"2020, Año de Leona Vicario, Benemérita Madre de la Patria"

SAN LUIS POTOSÍ. S.L.P. 13/Agosto/2020
SECCIÓN: DIV. ESTUDIOS DE POSGRADO E INVESTIGACIÓN
OFICIO: D.E.P.I. /059
ASUNTO: AUTORIZACIÓN DE IMPRESIÓN.

ING. IVÁN ALBERTO CRUZ GARCÍA
ALUMNO DEL PROGRAMA DE MAESTRÍA EN ING. ELECTRÓNICA
PRESENTE.

El que suscribe, Jefe de la División de Estudios de Posgrado e Investigación, por este medio me permito comunicar a Usted que se autoriza la impresión de su trabajo de tesis titulado: ***"OPTIMIZACIÓN DE NANOESTRUCTURAS PARA EL COSECHADO DE ENERGÍA SOLAR."***, para que con ello pueda usted continuar con los trámites para la obtención de su grado de Maestro en Ingeniería Electrónica.

Lo anterior de conformidad con el dictamen emitido favorablemente por los integrantes del H. Jurado Revisor, integrado por:

DR. RAMÓN DÍAZ DE LEÓN ZAPATA
Presidente
DR. EFRÉN FLORES GARCÍA
Secretario
DR. SAÚL ALMAZÁN CUÉLLAR
Vocal Propietario
DR. ISMAEL LARA VELÁZQUEZ
Vocal Suplente

ATENTAMENTE.
"CON TECNOLOGÍA Y ESPÍRITU UNA PATRIA FORJARÉ" ®

M.C. ARIEL BENJAMÍN DE LA ROSA ZAPATA
JEFE DEL DEPTO. DE LA DIVISIÓN DE ESTUDIOS DE POSGRADO E INVESTIGACIÓN DEL ITSLP

SECRETARÍA DE EDUCACIÓN PÚBLICA
TECNOLÓGICO NACIONAL DE MÉXICO
INSTITUTO TECNOLÓGICO DE SAN LUIS POTOSÍ
DIVISIÓN DE ESTUDIOS DE POSGRADO E INVESTIGACIÓN

Instituto Tecnológico de San Luis Potosí

Posgrado en Ingeniería Electrónica

Optimización de nanoestructuras para el cosechado de energía solar

TESIS

Que por optar por el grado de

MAESTRIA EN INGENIERIA ELECTRONICA

PRESENTA:

Ing. Iván Alberto Cruz García

Asesor de tesis:

Dr. Ramon Diaz de León Zapata

San Luis Potosí, S.L.P. agosto 2020

Agradecimientos

A mis padres y hermanas por su incondicional apoyo

A mi hija, es quien me motiva

A mis amigos y compañeros de estudio

A mis maestros por su asesoría, consejos y paciencia

Contenido

Capitulo 1. Introducción

El campo electromagnético ha sido causa de estudio desde hace siglos, anteriormente no se sabía de su existencia sin embargo ya se hacían observaciones acerca de la respuesta de los diferentes materiales a este campo electromagnético por ejemplo los primeros espejos que se hacían con metales pulidos como obsidiana y más tarde con cobre, existen historias sobre cómo Arquímedes logró diseñar unos espejos cóncavos con fines bélicos, para quemar barcos, está demostrado que prácticamente en sus tiempo no pudo construir espejos con las características necesarias para concentrar la energía con la cual se logrará este hecho, sin embargo se mostró que estos espejos concentran energía proveniente del sol, con lo cual se empezaba a experimentar la resonancia de los metales a nivel macroscópico.

Más adelante los experimentos de Faraday, Henry, Gauss y Ampere entre otros junto con las ecuaciones del matemático James Maxwell que ya en su tiempo caracterizaron el comportamiento básico del electromagnetismo en una teoría simétrica y amplia [1] [2].

Faraday experimentó la conversión de magnetismo en electricidad, con dichos experimentos obtuvo que un campo magnético variante en el tiempo tiene un campo magnético asociado a él, Gauss encontró una relación del flujo del campo eléctrico y las fuentes de este flujo de carga, esta es la ley de Gauss Electica, se sabe que no hay una contraparte magnética de la carga eléctrica, no existen los monopolos magnéticos por lo cual el flujo del campo magnético es cero, esta es la ley de Gauss magnética.

Los campos eléctrico y magnético se consideran dos características de un solo fenómeno físico el campo electromagnético, el cual es una onda que se mueve más allá de la fuente que la genero e independiente de la misma, estos campos eléctrico y magnético se regeneran mutuamente y sin fin. En la actualidad el ahorro en el consumo energético es de vital importancia, debido que se prevé que a futuro se agotaran lo recursos no renovables, como son el petróleo y sus derivados por lo cual se ha optado por la generación de energía limpia, barata y renovable como lo es captar la luz que proviene del sol convirtiéndola en energía eléctrica, por lo cual se ha buscado otras formas de captar dicha energía, en los últimos años el uso de dispositivos como antenas termopares ópticas para el sensado de luz y cosechado de energía solar, que ha tenido mucha atención debido a que es un medio para captar energía del infrarrojo cercano, en [3] se

demuestra numéricamente que estas antenas generan pequeñas señales de corriente directa, y se evalúa teóricamente la eficiencia de conversión óptica a eléctrica mediante la irradiación de las ondas electromagnéticas a frecuencias ópticas, además se aprovecha el efecto seebeck que se conoce de los materiales para incrementar la conversión a electricidad.

A proporciones nanométricas se ha descubierto que algunos metales como la plata con cierta geometría, por ejemplo antenas planas tienen una resonancia a frecuencias de tera Hertz, estas nano antenas tienen la capacidad de concentrar luz más allá del límite de difracción, este efecto se puede ver también en partículas nanométricas que soportan modos de resonancia plasmonica en anchos de onda ópticos haciéndolos antenas ópticas naturales, incluso algunas nanoestructuras auto ensambladas que tienen formas de antenas ya conocidas son usadas como dispositivos para recibir y emitir en frecuencias de THz

En [4] se muestra que esta estructura nanométricas en forma de estrella con nano cables de plata y nanotubos de óxido de Zinc, se muestran en simulaciones hechas en el software COMSOL resonancia de 10 THz a 120 THz con una resonancia principal a 60 THz, también hacen experimentaciones mediante espectroscopia Raman y espectroscopia infrarroja por transformada de Fourier de lo cual obtuvieron mediciones de 9 THz a 103 THz con lo cual se demuestra que los resultados coinciden con las simulaciones.

En este artículo la estructura en forma de estrella está basada en el diseño clásico de antenas el cual tiene un gran costo en recursos de tiempo y complejidad por lo que se ha buscado otros métodos de optimización como lo son los algoritmos evolutivos.

Algoritmos Genéticos (AG)

Algunos conceptos han sido inspirados en las investigaciones que Darwin realizó acerca de la evolución para resolver problemas del mundo real, esto generó un paradigma para resolver algún problema complejo.

Los AG son algoritmos numéricos para optimización basados en la naturaleza selección genética, es un algoritmo fácil de entender y programar en cualquier lenguaje de propósito general [5].

Un algoritmo genético simple consiste de las siguientes características:

1. Una población de individuos que suponen una solución a el problema usualmente aleatorios.
2. Una manera de calcular que tan bueno o malo (o que tan saludable) es cada individuo de la población.

3. Un método para mezclar fragmentos de los mejores individuos para formar uno nuevo.
4. Un operador mutación para evitar una pérdida permanente de diversidad de los individuos.

Típicamente los individuos (Cromosoma o genotipo) propuestos se codifican en binario de las variables del mundo real, esta población inicial será procesada por estos tres principales operadores.

Selección: realiza una función similar a la selección natural encontradas en sistemas biológicos, los individuos con un desempeño pobre son eliminados y los que tienen el mejor desempeño o que se encuentran en mejor forma siguen adelante, y son los que pasan la información de las características deseadas a generaciones futuras.

Cruce: Cambia información de una manera similar a como lo realiza un organismo natural mediante reproducción sexual, se toman un par de individuos escogidos por el operador selección, de las cadenas de binarios se selecciona un lugar de forma aleatoria y se intercambia toda la información a la derecha de este lugar entre los dos individuos y así nacerá un nuevo individuo.

Mutación: Cambia aleatoriamente el valor de un solo bit dentro de la cadena de un individuo, la mutación se usa escasamente.

Este proceso de selección, cruce y mutación continua hasta que un número fijo de generaciones han transcurrido o que se cumpla algún criterio de convergencia

Haciendo una analogía con el diseño de antenas hay que proponer varias geometrías diferentes, medir las ganancias de cada geometría, elegir las que tienen mayores ganancias (o visto desde otro punto de vista, menores pérdidas), hacer fusiones de 2 geometrías de las cuales se creará una nueva generación de nano antenas, de las cuales se medirán sus ganancias y así generación tras generación en las cuales la operación de mutación sería un cambio aleatorio de la geometría independiente de los demás individuos que sirve para aumentar la diversidad de geometrías.

En [6] se propone el uso de algoritmos genéticos para el diseño de antenas en lugar de las técnicas clásicas de diseño lo que requiere un proceso costoso que incluye el tiempo en diseño y la dificultad de construirlas, un software de simulación permite modelar una antena en específico y medir su eficiencia en cuestión de segundos.

Se presentan un tipo de antena conocido como (croocked-wire antena), una antena de cables torcidos, cuya aplicación es para comunicaciones tierra a un satélite, son antenas omnidireccionales.

En estas antenas se busca que tengan varias cualidades, una debe ser que solo debe tener un punto de alimentación en la base de la antena, además se espera que sea relativamente pequeña para que sea una antena barata y robusta pero también para que el espacio de soluciones del problema sea bastante pequeño; es deseable que la antena tiene una serie de cables rectos interconectados; de 5 a 8 cables se probaron para investigación, se utilizaron 7 cables debido a que se dieron mejores resultados preliminares.

Para codificar cada cromosoma se tienen 3 coordenadas x,y,z para cada punto inicial y final de cada cable (7 cables en total) lo que hace que se tengan 21 parámetros donde cada uno está representado con un número binario de 5 bits, entonces el cromosoma completo contiene 105 bits, las pruebas se hicieron con 1600 MHz, con la ecuación 1 se obtiene la función de salud de cada individuo propuesto, se inició con 500 cromosomas con un 50% de superposición en la operación de cruce y una taza de 1% de mutación, una vez converge el algoritmo en la computadora obtuvieron una antena con una forma inusual.

En los cálculos dio un excelente desempeño y se pudo construir a mano de manera sencilla, se le hicieron pruebas y se obtuvieron resultados de resonancia a frecuencias de 1300MHz y 1900MHz lo cual es esperado debido a las diferencias entre la simulación donde algunas características son ideales.

En la revisión se ha visto la efectividad de los algoritmos evolutivos frente a los métodos clásicos de diseño de antenas de tamaño macroscópico, además de que también se muestran experimentos y simulaciones la excelente respuesta que tienen algunas geometrías de tamaño nanométrico a la incidencia de ondas electromagnéticas a frecuencias cercanas a la de la luz visible incluyendo el infrarrojo cercano y lejano, por lo que en los siguientes artículos se presentara los resultados para experimentos y simulaciones de geometrías de tamaño nanométricos cuyo optimización es realizada mediante algoritmos genéticos.

En [7] y [8] se discute la diferencia entre la respuesta que tienen antenas de radiofrecuencia (R.F.) de tamaño milimétrico y las antenas ópticas de tamaño nanométrico, el diseño de antenas ópticas con el diseño clásico de antenas R.F. no contempla las propiedades de dispersión del material, esto se puede traducir en una antena óptica no optima, se argumenta el uso de algoritmos genéticos debido a que por su naturaleza permite explorar geometrías no convencionales y llegar a un óptimo local y en ocasiones a una buena aproximación de un óptimo

general, usan un AG para obtener una geometría optima que concentre el campo electromagnético de una nano-antena tipo dipolo a una frecuencia de resonancia de 500 THz, se utiliza COMSOL live link para Matlab para obtener simulaciones y compararlos con una geometría clásica de un dipolo.
En los resultados presentan dicha comparación en donde se ve fácilmente que el dipo generado con el algoritmo genético tiene menos perdidas que el dipolo clásico, además se realiza un conjunto de simulaciones para un rango de valores de frecuencia que van desde 200 Thz hasta 700 THz en donde se mide la concentración del campo electromagnético donde los resultados muestran una mayor respuesta en las frecuencias cercanas a los 500 THz, el dipolo clásico tiene una respuesta de aproximadamente 0.4 y el dipolo del AG de aproximadamente 0.95. con estos resultados concluyen que para encontrar una antena optima se debe tener en consideración los efectos del fotón que en principio son diferentes a los del electrón y que por lo tanto no se debe asumir que el proceso de diseño de antenas en el rango de radiofrecuencia es compatible con las antenas ópticas.
Los polinomios de Bernstein [9] son bien conocido en la aproximación matemática de funciones, se usan para aproximar tanto para funciones conocidas como desconocidas con un grado de exactitud deseado. Bezier desarrolló un modelo de spline que es bastante utilizado en graficación por computadora y diseño asistido por computadora. Este modelo ayuda a diseñar curvas suaves y superficies de diferentes formas y tamaños arbitrarios basados en un conjunto de puntos de control, el polinomio de Bezier-Bernstein representa una curva que puede ser generada de un conjunto ordenado de puntos representativos, llamados puntos de control, la línea que une esos puntos de control es llamada la línea de control del polinomio, con esto se tiene la forma de la curva que se quiere dibujar, dicha curva tiene las siguientes propiedades:

- Siempre interpolan los puntos de control final, además la línea que une los dos puntos consecutivos es la tangente a la curva en ese punto final.

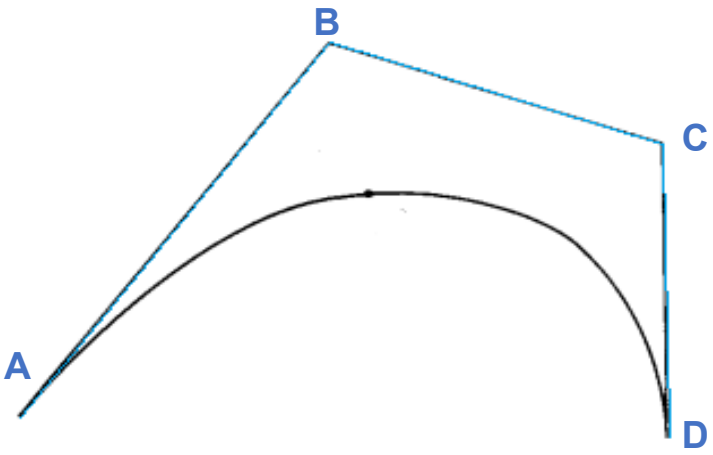


Figura 1.1 Curva Bézier con puntos de control A, B, C y D.

- La curva permanece dentro de las líneas que unen los puntos de control Figura 1.1
- Las curvas no exhiben un comportamiento oscilante sobre cualquier línea de mayor frecuencia que las líneas que unen los puntos de control.
- El dibujo de la curva no depende de ningún eje.
- La determinación del orden del polinomio es el numero de vértices del polinomio de control menos uno.

Cuando se tiene un solo objetivo es sencillo determinar cuáles de los individuos es que tiene menor campo eléctrico normalizado solo hay que ordenar con respecto a la variable objetivo, sin embargo, cuando son dos o más los objetivos y tienen diferentes mínimos o máximos si ordenamos por un objetivo en específico, los demás objetivos no quedan ordenados y no sería justa la selección de individuos visto desde el punto de vista del algoritmo genético.

Para optimizar geometrías y/o materiales se pueden utilizar variables eléctricas como son: el campo eléctrico normalizado, densidad de corriente, disipación de potencia, etc., cada una de estas se pueden utilizar como funciones de salud e incluso tener varias funciones que tengan mínimos o máximos independientes para cada individuo del AG. En [10] se presenta una metodología para optimizar geometrías mediante algoritmos evolutivos multiobjetivo es decir más de una variable física a medir, usan tres funciones, f1 la geometría, f2 las pérdidas de potencia del campo eléctrico y f3 la función de densidad de corriente, a menores perdidas de potencia y mayor densidad de corriente, se encuentra la mayor eficiencia de los plasmones.

Como resultados se obtuvo que la primera iteración del algoritmo genético multiobjetivo (AGM) tiene como respuesta un plasmón ineficiente comparado con un plasmón con una geometría con forma de gota sin embargo ya en la iteración 22 se obtiene un plasmón cuya geometría es más eficiente.
Un plasmón es una estructura metálica nanométrica que tiene una respuesta a una onda electromagnética con una longitud de onda cercana al tamaño de la estructura. Esta respuesta son electrones en movimiento sobre la superficie plasmonica. [11].
Para los algoritmos multiobjetivo se propone el método de optimización de Pareto, el cual define las condiciones necesarias y suficientes para resolver un problema matemático, típicamente en problemas de optimización multiobjetivo cada función tiene objetivos en conflicto, por un valor pequeño en un objetivo, es un valor alto en otro objetivo, por lo que la definición de optimalidad no es obvio como en el caso de un solo objetivo.
Uno de los métodos para resolver multiobjetivo [12] consiste en encontrar en el espacio de objetivos, una solución de Pareto que minimiza la distancia con un punto de referencia, el cual generalmente es el punto ideal.

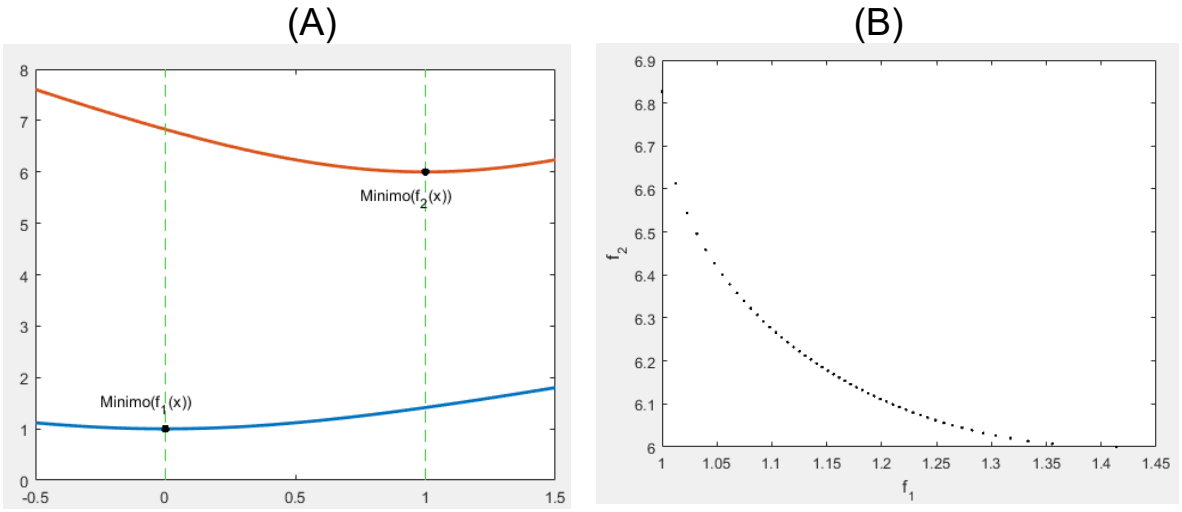


Figura 1.2 (A) Frente de Pareto de las dos funciones a minimizar (B) Funciones objetivo a minimizar,

En la Figura 1.2 (derecha) se observan dos funciones $f_1(x)\ y\ f_2(x)$ de las cuales se busca el punto x optimo, son restricciones de igualdad, en las cuales se calculó el mínimo: $\min\ (f_1(x)) = 1\ y\ \min\big(f_2(x)\big) = 6$,los cuales se encuentran en diferentes puntos de x por lo cual no es obvio cual sería el mínimo que satisfaga amabas restricciones para lo cual se

utiliza la optimalidad de Pareto, los mínimos anteriormente calculados son el punto ideal y los valores que se encuentran 0 < x < 1 para $f_1(x)\ y\ f_2(x)$ se grafica la Figura 1.2 (izquierda) conocida como el frente de Pareto, así se convierte en un problema de una sola restricción el mínimo de todas las distancias al punto ideal (1,6).

El método de elemento finito (FEM) permite resolver problemas de matemática física mediante métodos numéricos, comparándolo con métodos exactos tiene un error tan pequeño que para cuestiones prácticas es despreciable, además de que los métodos clásicos para cuestiones computacionales y para algunos problemas en 3 dimensiones son bastante complejos y en ocasiones imposibles de resolver. En la actualidad FEM es utilizado en una amplia variedad de problemas de ingeniería y física cuya característica principal es que tienen valores de frontera. El principio del FEM es cambiar todo un dominio continuo por varios subdominios en el cual la función incógnita es representada por una simple interpolación de funciones con coeficientes desconocidos. El problema original con valores de frontera con un numero infinito de grados de libertad se convierte a un problema con numero finito de grados de libertad. El principal paso de este método es la discretización del dominio debido a que la manera en que se discretiza afectara el tiempo de cómputo y la exactitud de los resultados numéricos. Los resultados de discretizar el dominio completo se les llama elementos, un dominio de una dimensión una línea recta o curva *Figura 1.3* (A), los elementos serian también líneas interconectadas para formar aproximadamente la línea original, para un dominio de dos dimensiones los elementos normalmente son pequeños triángulos y rectángulos *Figura 1.3* (B), en un dominio tridimensional se hacen divisiones en tetraedros y prismas triangulares o bloques rectangulares *Figura 1.3* (C).

En la mayoría de las soluciones de elemento finito, el problema se formula en términos de la función desconocida theta de los nodos asociados con los elementos. Una descripción completa de un nodo contiene sus valores de coordenada, un numero local y un número global, estos datos terminan en una matriz la cual dependiendo del número de elementos tiene su costo computacional el cual se puede reducir haciendo un apropiado numerado de los elementos y sus nodos.

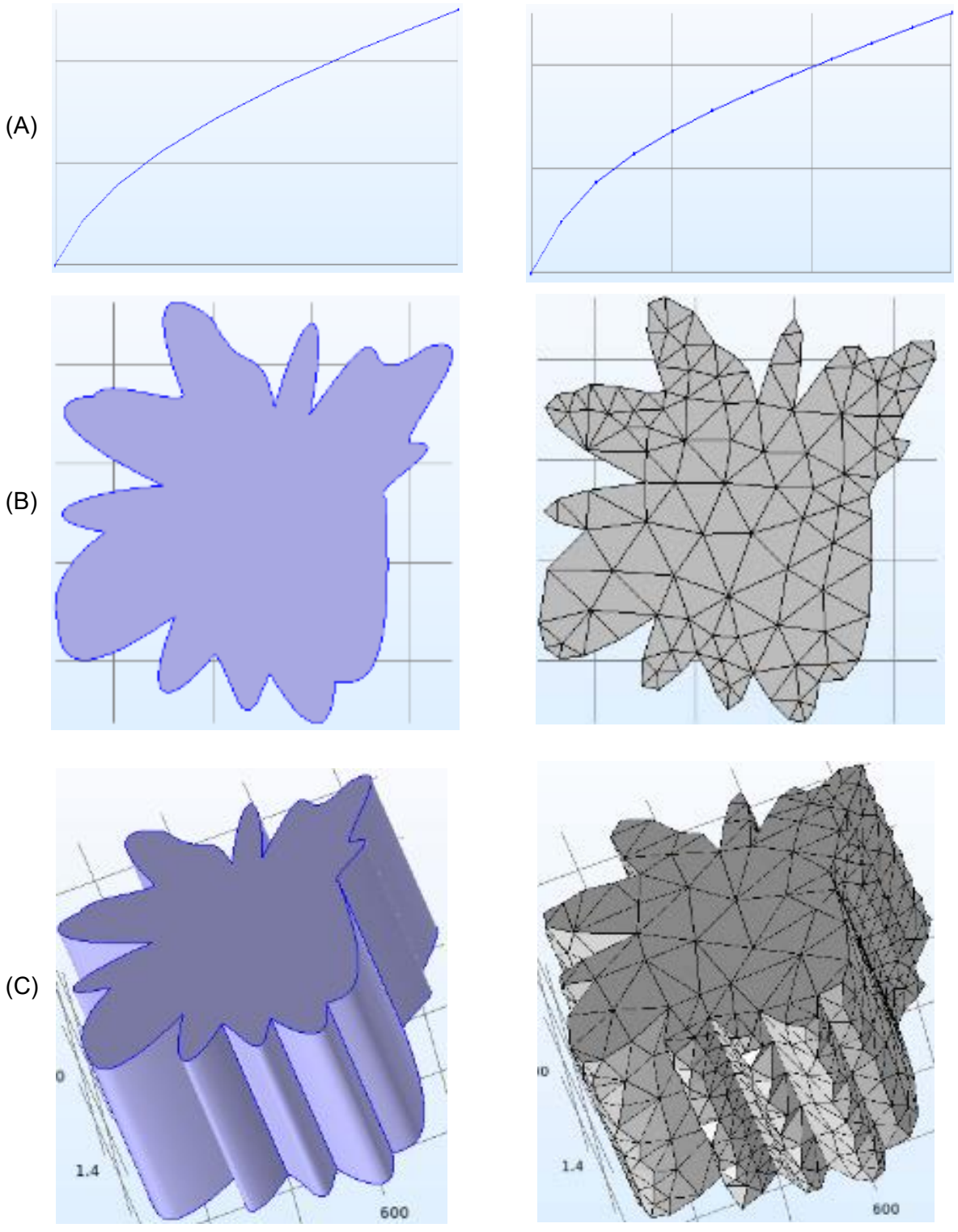


Figura 1.3 (A) ***Izquierda*** *dominio completo en 1D,* ***Derecha*** *conjunto de subdominios que forman el dominio completo en 1D, (B)* ***Izquierda*** *dominio completo en 2D,* ***Derecha*** *conjunto de subdominios que forman el dominio completo en 2D, (C)* ***Izquierda*** *dominio completo en 3D,* ***Derecha*** *conjunto de subdominios que forman el dominio completo en 3D.*

Objetivo General

Encontrar mediante algoritmos evolutivos la estructura de tamaño nanométrico con materiales metálicos que tenga la mejor respuesta a una onda electromagnética con frecuencia en el orden de las unidades de Tera Hertz.

Capitulo 2. Algoritmo Genético Simple (AGS)

Objetivo específico

Encontrar mediante el algoritmo genético mono objetivo la geometría de tamaño nanométrico con material Oro que tenga la menor magnitud del campo eléctrico normalizado en respuesta a una onda electromagnética con frecuencia de 1 THz.

Optimizando una geometría con material Oro.

Debe de ser una geometría de 2 dimensiones y un área máxima de un cuadrado de lado 1024nm, con esto se limita el espacio de soluciones del problema a optimizar.

Para la construcción de esta geometría se propone el algoritmo de curvas de Bézier el cual permite construir una curva suavizada con pocos parámetros de configuración 4 puntos de control, punto inicial y final, y dos puntos de inflexión, al unir varias curvas se forma un cuerpo cerrado el cual mediante el cambio de sus puntos de control permite que la geometría sea flexible, el número de curvas que se utilizan en cada geometría definirá el grado de flexibilidad en la geometría.

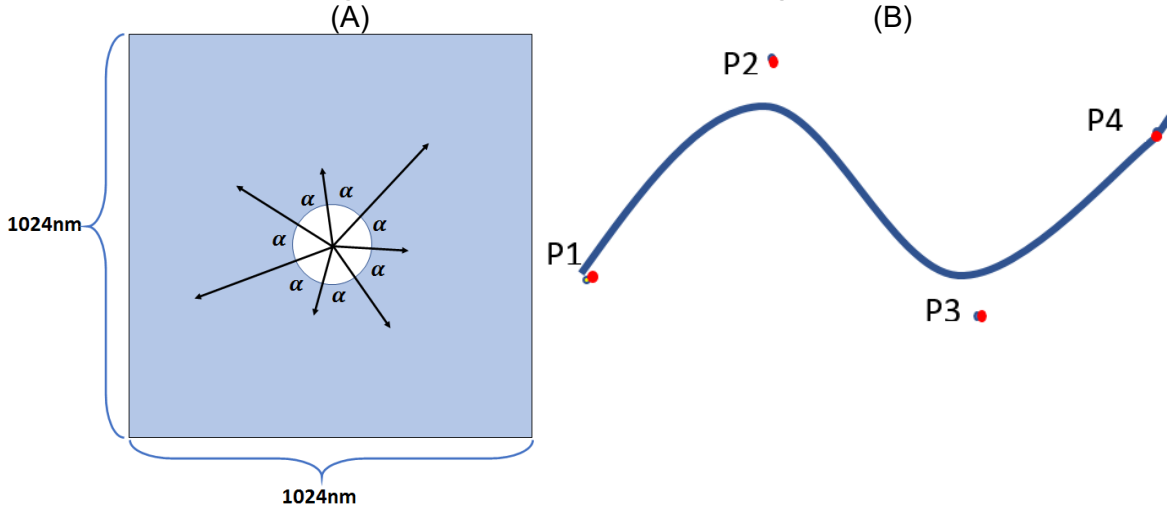


Figura 2.1 (A) Conjunto de vectores con magnitud aleatoria y ángulo de separación constante, (B) Curva Bézier grado 3.

*Cada punto de las curvas Bézier está definido por un vector (Magnitud y Angulo) cuyo origen para todos los vectores es el centro (512,512) del espacio de soluciones, como se puede ver en la (ref) todos los vectores tienen el mismo ángulo α entre sí, la magnitud de cada vector cambia aleatoriamente entre 1nm y 512nm, NTP = (NC*3)+1 donde NTP es el número total de puntos necesarios para formar una geometría, NC es el número de curvas y el ángulo α = 360/NTP.*

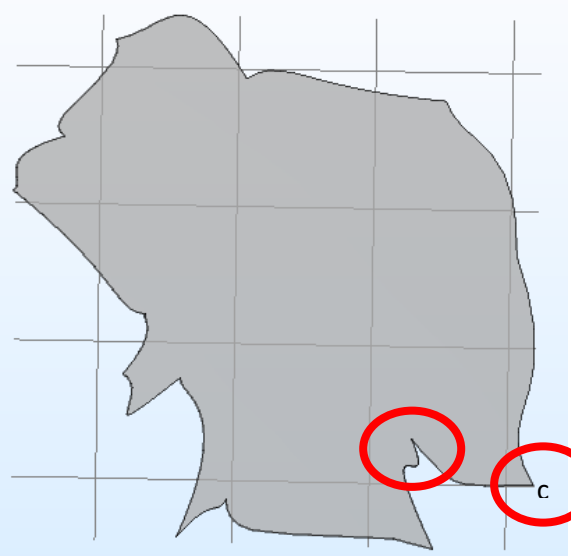


Figura 2.2 Figura generada por COMSOL, en óvalos rojos picos no deseados

Con las restricciones mencionadas se generan geometrías como la que se ve en la Figura 2.2 se pueden observar algunos picos no deseados debido a que estos causan errores en COMSOL, por lo que para resolver esto se debe cumplir Ecuación 2.1 que para los puntos cercanos a la unión de dos curvas generen un cambio suave

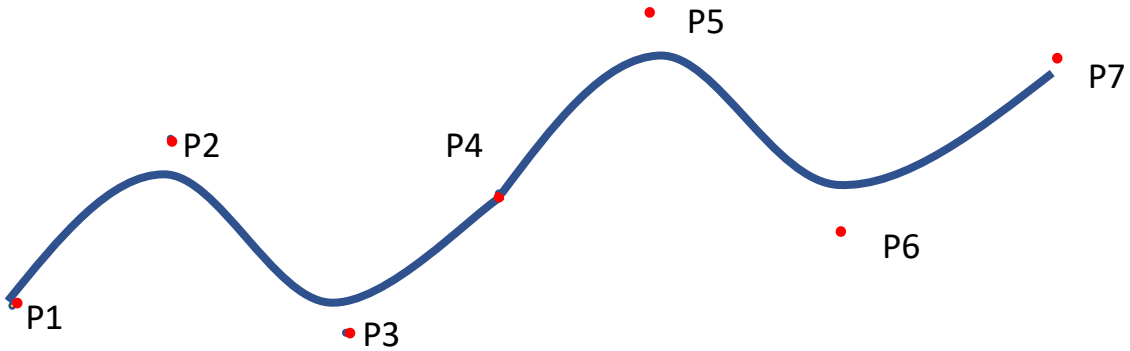


Figura 2.3 Unión de 2 curvas Bézier, primera curva (P1,P2,P3,P4), segunda curva (P4,P5,P6,P7)

Ecuación 2.1 $$|P3| \leq |P4| \leq |P5|$$

Población inicial, se generarán 40 geometrías y cada geometría cuenta con 17 curvas Bézier unidas entre sí, cada curva se define con vectores de control debido a que el 4to vector se comparte con la siguiente curva, por geometría son 52 vectores o cromosomas.

Cada cromosoma se representa en un número binario de 9 bits (512 combinaciones),

Selección: El software COMSOL regresa un escalar que es el campo eléctrico normalizado $norm(\vec{E})$ como respuesta a una onda electromagnética de 1Thz que incide sobre la geometría propuesta, Este escalar es lo que llamamos función de salud, FF (Fitness Function), se busca las mínimas pérdidas en el campo eléctrico normalizado por lo que de las 40 geometrías iniciales solo los 20 con menor campo eléctrico normalizado seguirán para la siguiente generación , las otras 20 geometrías desaparecen del algoritmo.
Una geometría contiene 52 cromosomas se tiene la misma FF debido que COMSOL calcula el campo eléctrico normalizado para toda el área incluyendo los puntos de las curvas Bézier.

Cruce: Esta operación es similar a la meiosis que realizan algunas células con un par de cromosomas se divide para formar dos células con la mitad de cromosomas. En AGS la magnitud del vector es el cromosoma, se puede ver en la Tabla 2.1 la madre hereda al hijo los 3 bits más significativos (azul) y el hijo intercambia de manera aleatoria los 6 bits menos significativos(amarillo), con esta operación el hijo será muy parecido a la madre y además los cambios evolutivos entre generaciones no son tan bruscos debido a que solo cambian los bits menos significativos

Madre									
Información del cromosoma	0	1	0	1	0	1	0	1	0

Hijo									
Información del cromosoma	0	1	0	0	0	1	1	1	1

Tabla 2.1 En azul la información que hereda la madre al hijo, en amarillo información que se genera aleatoriamente en el hijo

Mutación: En la mutación se realiza después de la operación cruce, cuando nace del nuevo individuo y consiste en cambiar cualquiera de los 9 bits que conforman un cromosoma, como se ve en la Tabla 2.1 el bit que se elegido es el 8 de manera aleatoria, si tiene un 0 lo cambia por 1 y si tiene un 1 lo cambia por 0, con esto nos permite que la geometría tenga variaciones mucho más bruscas con una probabilidad de 0.01% de ocurrencia.

Hijo									
Información del cromosoma	0	1	0	0	0	1	1	1	1

Tabla 2.2 En verde el bit que muta del cromosoma

COMSOL LiveLink para Matlab

Este software permite simular una gran variedad de físicas en cualquier geometría diseñada en software CAD, para realizar esto es necesario las siguientes configuraciones.

Geometría:

El primer parámetro es la unidad de longitud para nuestro caso de estudio nanómetros

Con la operación de polígono Bézier se van eligiendo cada punto de control de la geometría y el algoritmo Bézier va construyendo las curvas, todas estas curvas están unidas para formar una figura cerrada Figura 2.4

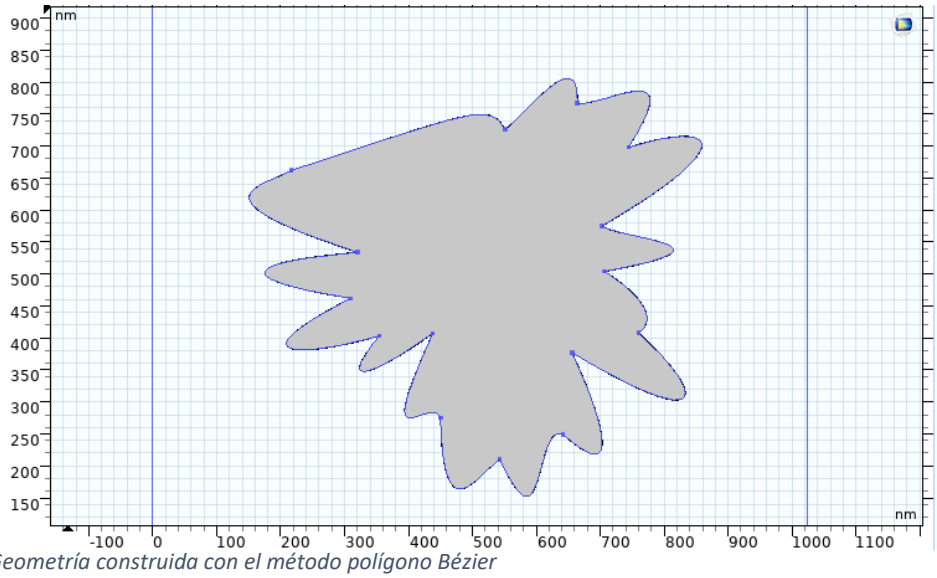


Figura 2.4 Geometría construida con el método polígono Bézier

La siguiente operación es extruir se ve en la Figura 2.5 esto debido a que la simulación debe ser realista por lo tanto la geometría, aunque pensada en 2 dimensiones debe tener un grosor mínimo 50nm.

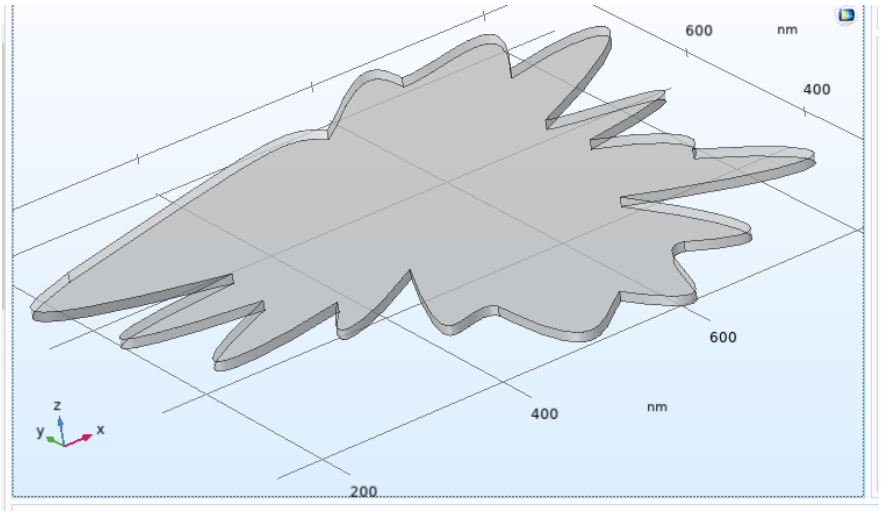


Figura 2.5 Operación extruir

Una vez que se tiene la geometría en 2 dimensiones con un grosor mínimo, se deposita dentro de un par de bloques de material aire.

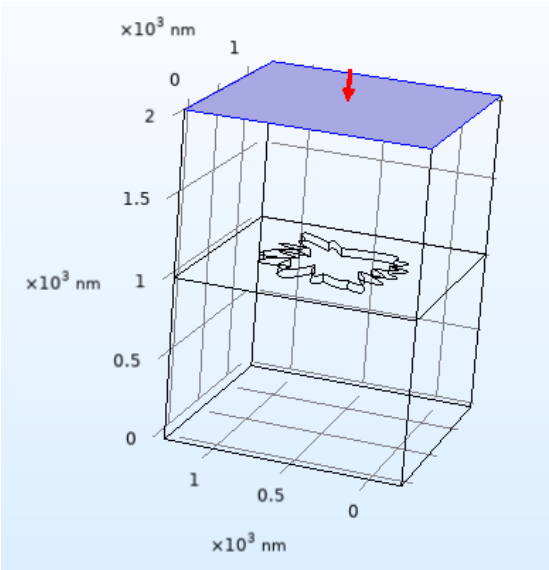


Figura 2.6 Operación Bloque

Materiales:

Se define el material el par de bloques (Aire) es el medio de dispersión de la onda electromagnética, los cubos de aire determinan las fronteras de la simulación, un par de caras de dichos cubos definen el sentido en el que viaja la onda electromagnética Figura 2.6, donde empieza, donde

termina y se genera un plano en donde se deposita la geometría a evaluar. Esta geometría para este caso de estudio del capítulo 1 es de material oro debido a sus características eléctricas como buen conductor, una de las características importantes que se configura en COMSOL es el índice de refracción del material, el cual cambia respecto a la longitud de onda que incide en el material para este caso 1THz, en [13] se obtiene un índice de refracción del oro de 345+j379, dato experimental con una longitud de onda de 248um. La física elegida para estas pruebas son Ondas Electromagnéticas, Dominio de la frecuencia, para realizar FEM de la física se define un mallado que cubre la geometría Figura 2.7 en COMSOL se tiene la flexibilidad de controlar el tamaño de cada uno de los elementos lo cual repercute en el tiempo de procesamiento y tamaño de los archivos generados, mientras más pequeño sean los elementos mayor costo computacional se tiene, sin embargo un tamaño muy grande de los elementos reduce la exactitud de los cálculos, por lo cual se eligió tamaño fino lo cual es un nivel más pequeño que normal esto genera elementos con un rango de tamaños que va desde 20 nm hasta 164nm, en cuanto la forma de los elementos COMSOL permite personalizarlo, se eligió tetraedro libre debido que como se revisó en la literatura el tetraedro se usa cuando la geometría tiene forma irregular, en nuestro caso es una geometría aleatoria.

(A) (B)

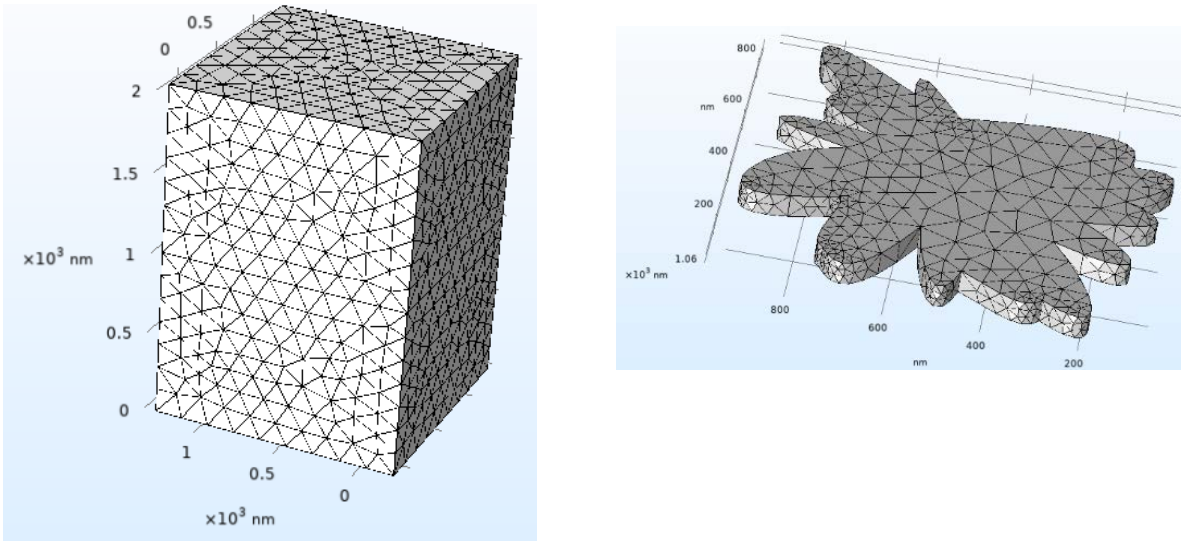


Figura 2.7 En el software COMSOL (A) Aire mallado, (B) geometría generada mallada.

Dentro de la física Ondas Electromagnéticas, Dominio de la frecuencia se crea un estudio en el cual se definen la variable a medir y la frecuencia de la onda electromagnética, la variable a medir es el campo eléctrico normalizado cuya unidad se mide en volts por metro, V/m, y la frecuencia se optó por 1THz.

(A) (B)

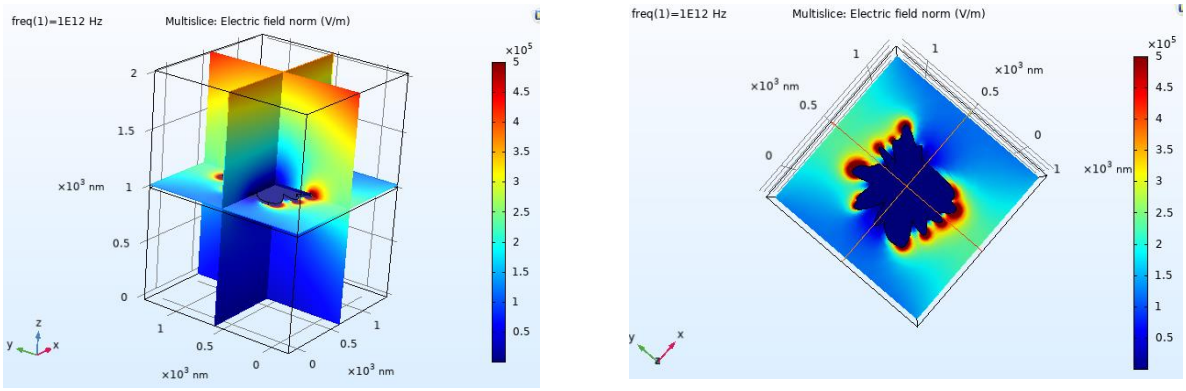


Figura 2.8 Representación del campo eléctrico normalizado a una onda electromagnética de 1Thz. (A) Vista de la onda electromagnética y el plano de la geometría, (B) vista del plano X Y donde está la geometría.

Una de las opciones de COMSOL para mostrar los resultados es sobre la geometría, los planos verticales simulan la señal electromagnética los colores indican que la señal se va atenuando conforme viaja por el aire, sobre el plano horizontal se deposita la geometría propuesta y se observa la respuesta de la onda electromagnética. Esta forma gráfica de representar la respuesta al ser sobre todo el área no es factible para poder obtener un escalar de la función de salud, otra forma de representar un resultado escalar es mediante una tabla la cual calcula mediante una integral doble de superficie del campo eléctrico normalizado (V*m)

Columna	Encabezado
1	Frecuencia (Hz)
2	Campo Eléctrico Normalizado (V/m)

Tabla 2.3 Resultado que nos proporciona COMSOL de la física aplicada a la geometría.

En la Figura 2.9 se observa el proceso en cual Matlab ejecuta directivas del software COMSOL mediante la función de LiveLink a su vez permite intercambiar parámetros por ejemplo los valores X, Y de cada punto de la geometría y la función de salud, el campo eléctrico normalizado.

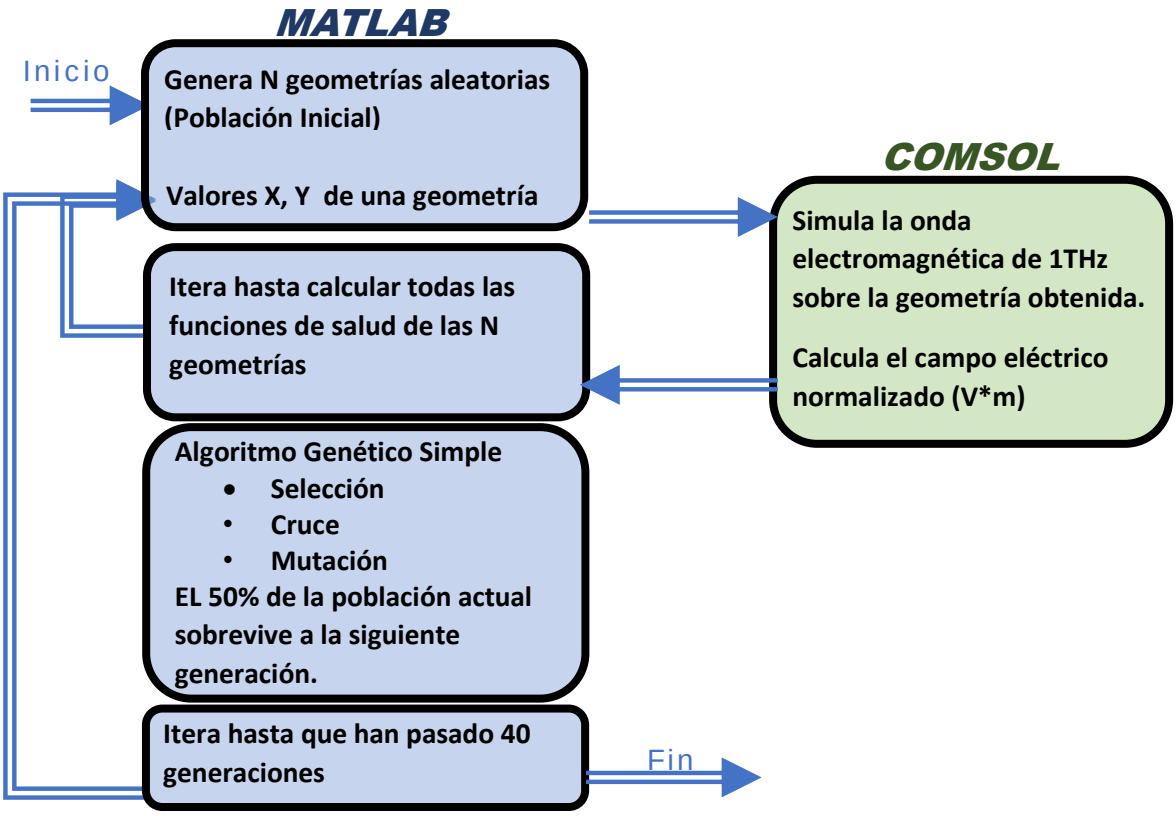


Figura 2.9 Proceso de comunicación de MATLAB con COMSOL mediante LiveLInk

El bloque que se resuelve en Matlab es un algoritmo que se propuso medida en Anexo A para este estudio el cual nos permite configurar los siguientes parámetros:

Tamaño del espacio de soluciones de lo cual depende el tamaño máximo de las geometrías, el centro de la geometría, el número de curvas Bézier de cada geometría, grado de cada curva Bézier, cantidad de geometrías propuestas como población inicial, numero de iteraciones del algoritmo genético simple, numero de bits a intercambiar en la operación cruce, el porcentaje de individuos que sobreviven a la siguiente iteración en la operación de selección, porcentaje de ocurrencia de la operación mutación.

Con todos estos parámetros configurables permiten realizar múltiples estudios con solo cambiar valores de dichos parámetros entre ellos proponer un algoritmo para dos geometrías independientes lo que se propone en los siguientes capítulos.

Capitulo 3. Optimizando dos geometrías con material Plata.

Objetivo específico

Encontrar mediante el algoritmo genético mono objetivo el par de geometrías de tamaño nanométrico con material plata que tenga la menor magnitud del campo eléctrico normalizado en respuesta a una onda electromagnética con frecuencia de 1 THz.

Optimizando dos geometrías con material Plata.

Para este caso de estudio el universo de soluciones cambia debido a que ahora se construirán 2 geometrías independientes y aleatorias, cada una estará limitada a un tamaño máximo de 1023 de ancho y altura, el área total de construcción de las 2 geometrías es de 1024 de altura y 2048 de ancho, el centro de las geometrías es (512,512) para la geometría de la derecha y (1536,512) para la geometría de la izquierda Figura 3.1.

La línea roja es la zona en donde los vectores más cercanos de las 2 geometrías se acercarán a esta zona, se genera una unión entre las 2 geometrías figura III.2 (A), esta unión no forma parte del algoritmo genético simple debido a que se busca que siempre tenga el mismo tamaño independientemente de la forma y tamaño de las 2 geometrías.

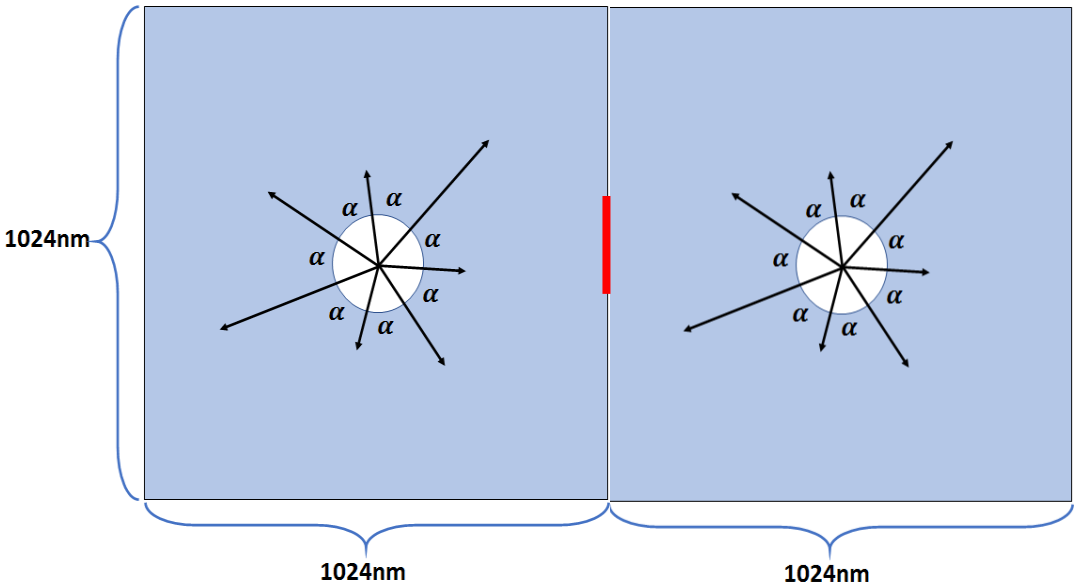
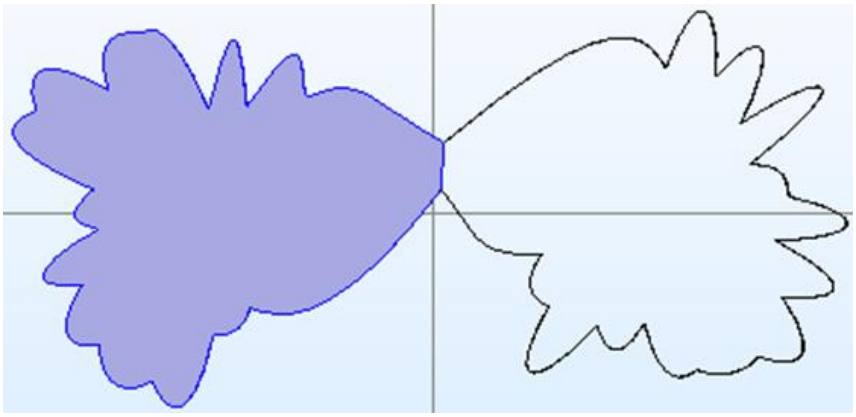


Figura 3.1 Espacio de soluciones para 2 geometrías, y los vectores que representan cada uno de los puntos Bézier

Esta unión figura 3.2 (B) se observa un área de unión, esta área es importante asegurar que el algoritmo no la disminuya debido a que si la concentración de energía en un área muy pequeña puede ocasionar calentamiento en esta zona.

(A)



(B)

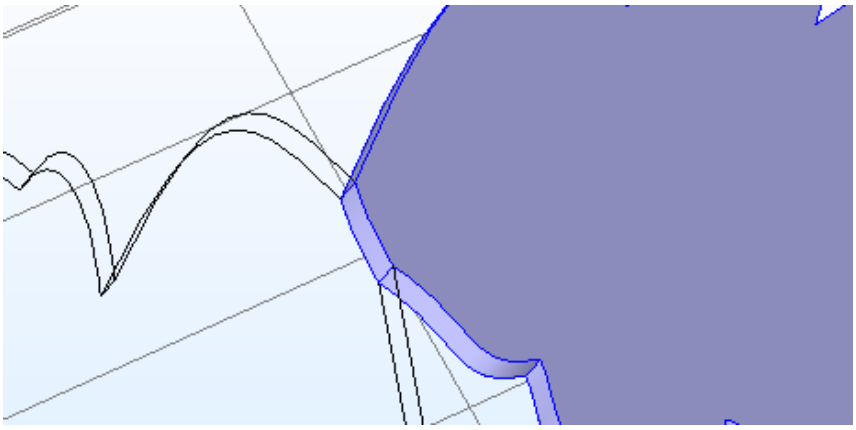


Figura 3.2 Unión entre dos geometrías aleatorias

Comparando con el algoritmo de construcción de la geometría del capítulo anterior el cálculo de cada uno de los valores de los puntos de estas geometrías es similar, únicamente los 4 vectores más cercanos a la zona de unión se les dan valores constantes durante todo el algoritmo, es decir todos los demás puntos irán evolucionando como si se tratase de una sola geometría. Aunque son dos geometrías independientes el AGS ve cada punto como un individuo y todos en conjunto como una sola población inicial por lo que las operaciones del algoritmo genético simple (selección, cruce, mutación) se puede reutilizar.

COMSOL LiveLink para Matlab

Las 2 geometrías son reconocidas como figuras independientes hay que especificar a cada una sus características por ejemplo el material que para este caso de estudio se eligió plata para ambas, una de las características importantes que se configura en COMSOL es el índice de refracción del material, el cual cambia respecto a la longitud de onda que incide en el material para este caso 1THz, en [13] se obtiene un índice de refracción de 531+j689, dato experimental con una longitud de onda de 248um.

El estudio de la física onda electromagnética de 1THz en el dominio de la frecuencia sobre un par de geometrías de material plata y sobre un medio que es el aire, este software permite observar la respuesta, el campo eléctrico normalizado, (V*m), esta variable es la que define la función de salud.

En la Figura 3.3 se ve el mallado de dos geometrías propuestas con una unión en el centro del universo de soluciones, con la cual COMSOL calculara mediante FEM el campo eléctrico normalizado.

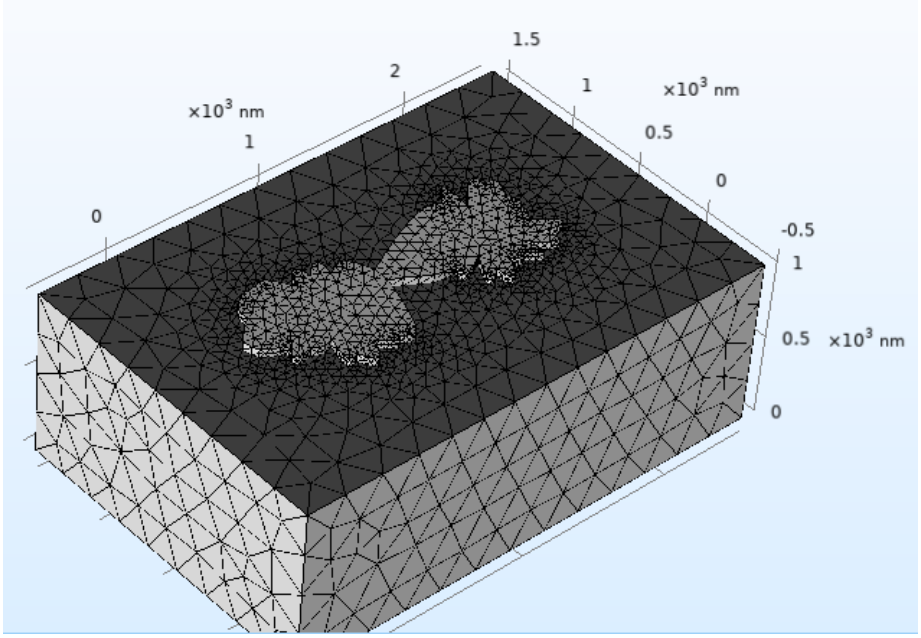


Figura 3.3 Geometrías vista del mallado

En la Figura 3.4 se ve el resultado de calcular el campo eléctrico normalizado sobre las dos geometrías propuestas, en esta vista aérea desde arriba como referencia desde la fuente de la onda

electromagnética, lo que se ve alrededor de las geometrías en color rojo intenso, amarillo y azul claro son los diferentes valores del campo eléctrico donde el rojo es la mayor magnitud de esta variable.

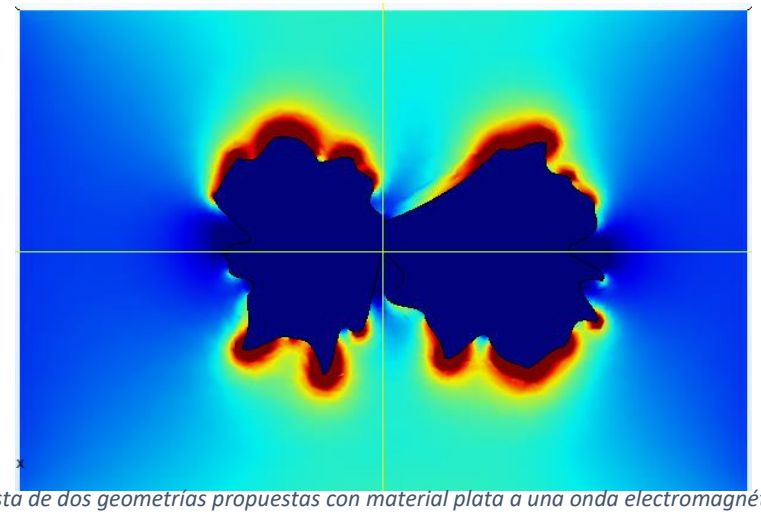


Figura 3.4 Respuesta de dos geometrías propuestas con material plata a una onda electromagnética con una frecuencia de 1Thz.

El resultado de calcular la integrar de superficie de la geometría obtenida por el AGS COMSOL lo pasa a Matlab mediante una tabla, dicho dato se utiliza para definir la función de salud dentro de la operación de selección, como se comentó en el capítulo anterior Anexo A, el algoritmo es similar al visto en el diagrama de bloques de la Figura 2.9, las diferencias en el código del Anexo B, uno de los principales cambios es que se usan dos funciones cada una genera una de las dos geometrías, esto es para poder generar diferentes tipos de figuras por ejemplo una genera una geometría aleatoria lo cual con las curvas Bézier se ve como una gota de agua al golpear una superficie y la otra geometría puede generar figuras deterministas como por ejemplo un círculo o elipse lo cual se genera dando la misma magnitud a todos los vectores que forman los puntos de control, con lo cual nos permite tener un par de geometrías como en la Figura 3.5.

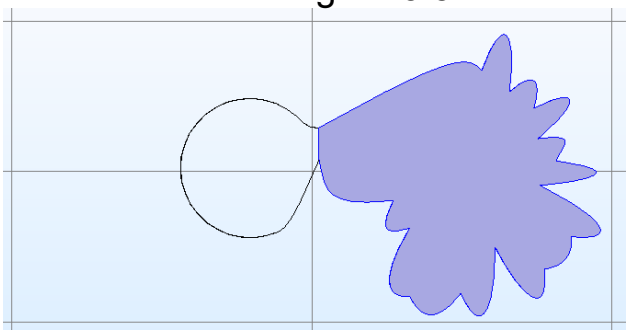


Figura 3.5 Par de geometrías, una determinista (circulo) y otra aleatoria (azul)

Capitulo 4. Algoritmo Genético Multiobjetivo

Objetivo específico

Encontrar mediante el algoritmo genético multi objetivo el par de geometrías de tamaño nanométrico con combinación de materiales con coeficiente seebeck de diferente signo que tenga el menor campo eléctrico normalizado y densidad de disipación de potencia total respuesta a una onda electromagnética con frecuencia de 1 THz.

Optimizando 2 geometrías y 2 materiales, con dos funciones objetivo

En este capítulo se muestra la metodología que se sigue para optimizar dos geometrías cuyos materiales muestren como resultado un efecto seebeck eficiente, se eligieron 6 materiales, 3 materiales con coeficiente seebeck negativo (CSN) y tres materiales con coeficiente seebeck positivo (CSP) ver en la Tabla 4.1

ID	Material	Coeficiente Seebeck
1	Tungsteno	7.5
2	Plata	6.5
3	Tantalio	4.5
4	Potasio	-9.0
5	Bismuto	-72
6	Nickel	-15

Tabla 4.1 Valores de coeficiente seebeck de los materiales: Tungsteno, plata, tantalio, potasio, bismuto y nickel

Para lograr un efecto seebeck eficiente se logra con unión de dos materiales uno con CSP y el otro con CSN por lo que las posibles combinaciones se ven en la Tabla 4.2.

Combinaciones	
Tungsteno	Potasio
Tungsteno	Bismuto
Tungsteno	Nickel
Plata	Potasio
Plata	Bismuto
Plata	Nickel
Tantalio	Potasio
Tantalio	Bismuto
Tantalio	Nickel

Tabla 4.2 Posibles combinaciones entre materiales

Estas combinaciones de materiales definen parte del universo de soluciones, en conjunto con las 2 geometrías como en el capítulo anterior nos queda un espacio de 1024nm de altura y 2048 nm de longitud, para este caso de estudio también se requiere que entre las dos geometrías tenga un área de unión.

COMSOL LiveLink para Matlab

Al ser geometrías independientes unidas en una cierta área COMSOL permite asignarle un material diferente a cada geometría, por lo que se investigó el índice de refracción de cada material y que además el material este dentro de la base de datos de COMSOL en [13], [14] se obtienen los índices de refracción como se ve en la Tabla 4.3.

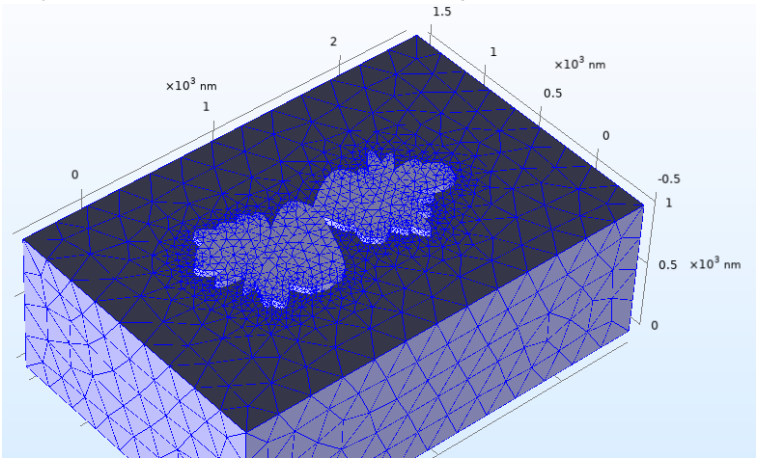
ID	Material	Índice de refracción
1	Tungsteno	242.14+j332.82
2	Plata	531+j689
3	Tantalio	184.5+j196.87
4	Potasio	4.77+j28.2
5	Bismuto	6.908+j1.47
6	Nickel	286+j375

Tabla 4.3 Índices de refracción de W, Au, Ta, K, Bi, Ni.

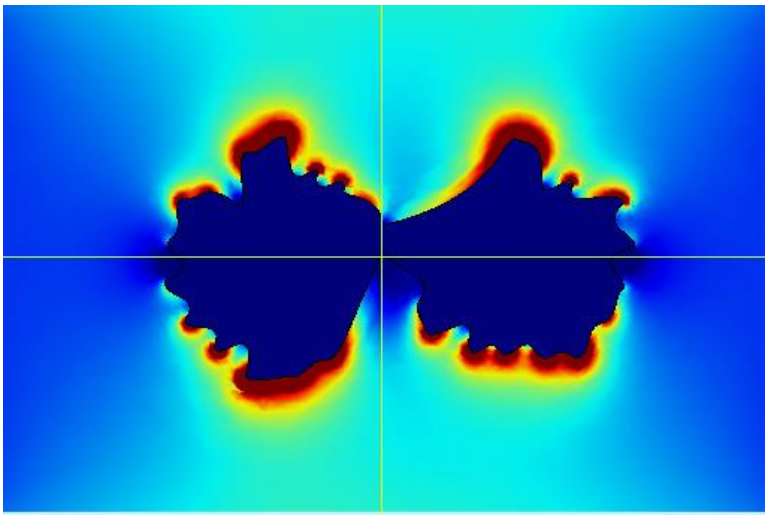
La física que se eligió es ondas electromagnéticas, dominio de la frecuencia debido que dentro se encuentran las siguientes dos variables: Campo eléctrico normalizado y Densidad de disipación de potencia total se mide en W/m^3.

En la Figura 4.1 (A) se presenta el mallado de dos geometrías y dos materiales, en si no hay diferencia de obteníamos el mismo tipo de mallado para dos geometrías y un material, para este caso de estudio se configuraron dos respuestas graficas una para la respuesta del campo eléctrico normalizado y como se ve en la Figura 4.1 (B) no es claro si existe un diferencial de voltaje o temperatura entre los dos materiales con diferente coeficiente seebeck por lo que se optó por la segunda respuesta grafica la densidad de disipación de potencia total en la Figura 4.1 (C) se ve que esta variable si mide una diferencia entre los dos materiales por lo cual se puede utilizar para determinar que par de geometrías y dos materiales es la mejor.

(A)



(B)



(C)

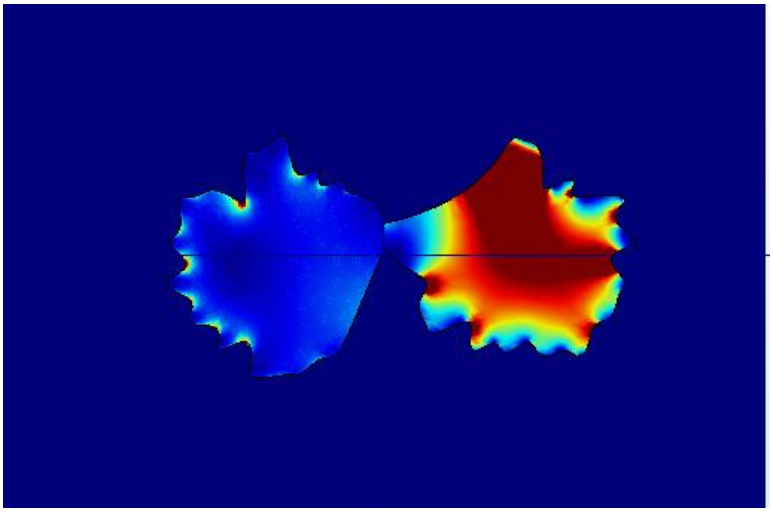


Figura 4.1Par de geometrías con 2 materiales: (A) Mallado, (B) Respuesta campo eléctrico normalizado V/m, (C) Densidad de la disipación de potencia total W/m^3.

Medir la disipación de potencia sobre las dos geometrías es una forma indirecta de medir el diferencial de potencial a través de la unión de dos metales, efecto seebeck, al mismo tiempo se busca medir las pérdidas de energía mediante el campo eléctrico por lo que para el AGM se utilizaron estas dos medidas las cuales COMSOL las calcula y regresa mediante la Tabla 4.4.

Columna	Encabezado
1	Frecuencia (Hz)
2	Campo Eléctrico Normalizado (V/m)
3	Densidad de disipación de potencia total (W/m^3)

Tabla 4.4 Variables respuesta que regresa COMSOL a Matlab.

Estas dos variables se calculan con una integral de superficie por lo que esta tabla regresa un escalar en cada variable, en la Figura 4.3 se muestra el proceso en donde después de generar las N par de geometrías aleatorias también se calculan las combinaciones de materiales de 9 diferentes posibilidades, esta información se pasa a COMSOL donde se calcula las dos variables antes mencionadas las cuales MATLAB mediante la optimalidad de Pareto logra transformar dos funciones objetivo f1(Campo eléctrico) y f2(disipación de potencia) que tienen la característica de no tener un mínimo en común, por lo que no es muy claro cuál de los puntos es el mínimo.

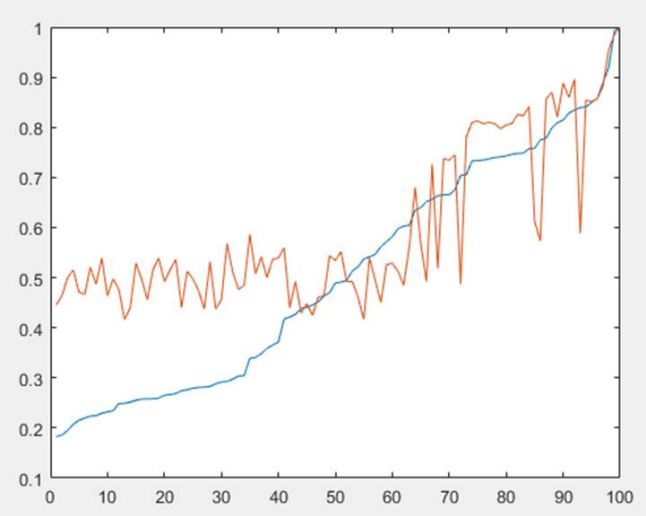


Figura 4.2 Grafica Campo eléctrico normalizado (azul) y la densidad de disipación de potencia total(rojo) de 100 par de geometrías.

De las dos se busca un punto ideal que es el mínimo de cada función (min(f1),min(f2)), cada una de las N geometrías tienen un punto (f1(x),f2(x)) por lo que se busca las N/2 geometrías con la menor distancia al punto ideal con lo que el problema Multiobjetivo se transforma en un problema de un objetivo, un conjunto de distancias al punto ideal. Entonces hay que encontrar el o los mínimos de este conjunto de distancias. Una ventaja de usar la optimalidad de Pareto es que es que se acopla bastante bien a la definición de multiobjetivo y lo simplifica a un solo objetivo eso es debido a que la distancia al punto ideal es una métrica que funciona para cualquier espacio $\mathbb{R}^m$, es decir, aunque se tengan m restricciones a minimizar en nuestro problema Pareto lo transformara a un problema en $\mathbb{R}^1$.

El diagrama que se presenta en la Figura 4.3 todos sus bloques en color azul están codificados en el leguaje de Matlab y mediante el Live link se ejecutan los bloques en verde que son funciones especiales de COMSOL, al primer bloque de Matlab se le agrego que generara aleatoriamente que combinación de material iba a tener cada par de geometrías como población inicial, al bloque de COMSOL se le agrego que midiera la densidad de disipación de potencia total cuyo resultado lo pasa por medio de una tabla al tercer bloque de Matlab el cual por medio de la optimalidad de Pareto determina e mínimo campo eléctrico normalizado y la menor densidad de disipación de potencia total y pueda a su vez llevar a cabo las operaciones del Algoritmo Genético Multiobjetivo; selección, cruce y mutación.

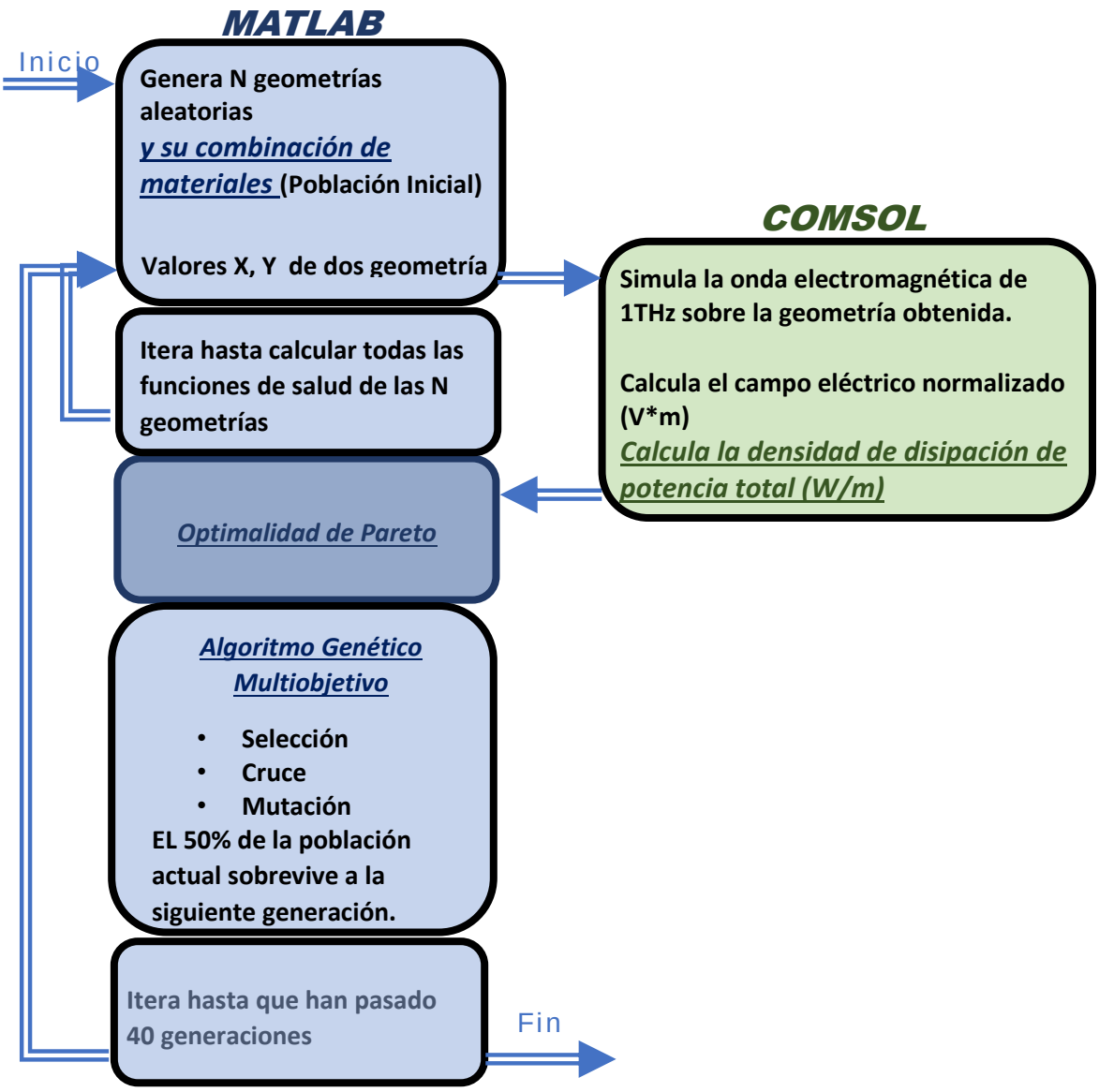


Figura 4.3 Proceso de comunicación de MATLAB con COMSOL mediante LiveLInk del AGM

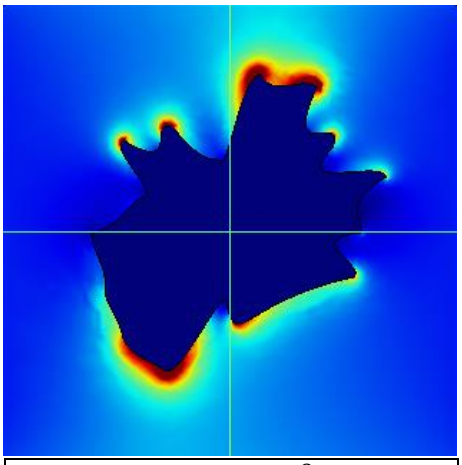
En el Anexo C todas las funciones necesarias para correr el AGM, cada función de las combinaciones de materiales se realizó a mano, esto debido a que cada material tiene varios vectores paramétricos que permiten definir las propiedades de los materiales, estos vectores son bastante largos y debido a esto es complicado que se pueda automatizar esta parte, se tiene que realizar cada combinación en COMSOL, aquí existe una opción que convierte el proyecto de COMSOL a código de Matlab, y de ahí separar la parte de código que pertenece solamente a las propiedades de los materiales y ponerlas en una función que será llamada cada que el algoritmo cambie de material.

Capitulo 5.Resultados, Discusión y Conclusiones

Resultados del Capítulo 2

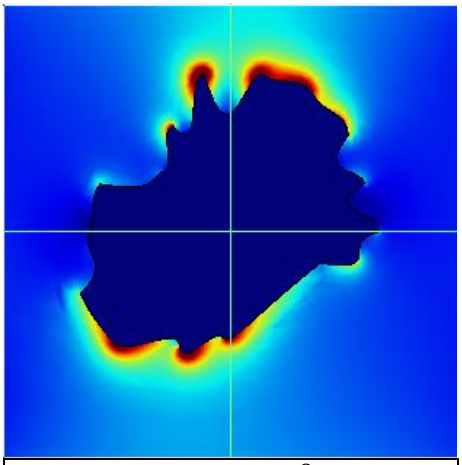
Inicialmente se realizó una prueba en donde se propuso una población inicial de 4 individuos, en este caso es pequeña la población para poder seguir de manera sencilla la evolución entre generaciones, en la Figura 5.1 se muestran 4 geometrías que se generaron aleatoriamente se observa la respuesta gráfica y su respectivo escalar el campo eléctrico normalizado.

(A)



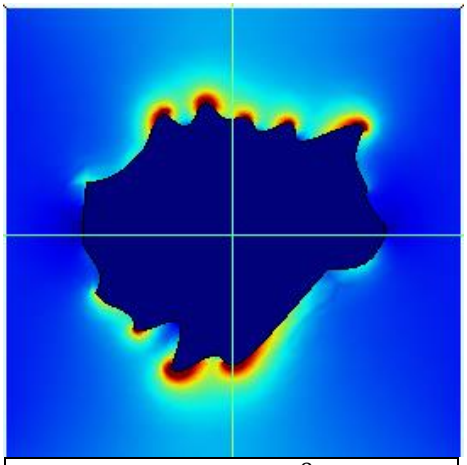
$4.6747x10^{-8}$

(B)



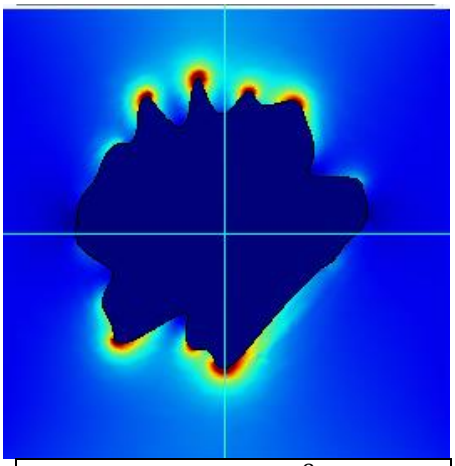
$5.2714x10^{-8}$

(C)



$4.9836x10^{-8}$

(D)



$5.2145x10^{-8}$

Figura 5.1 Población Inicial de 4 geometrías aleatorias, representando la respuesta campo eléctrico normalizado v/m

En la operación de selección, de estas 4 geometrías el algoritmo escogió el 50% con mayores pérdidas, Figura 5.1 ,los incisos (B) y (D), se descartan y no sobreviven, los incisos (A) y (C) sobreviven a la siguiente generación, por lo que estas dos geometrías por medio de la operación de cruce tienen 2 hijos, los cuales se observan en la Figura 5.2 donde cada geometría tiene el mismo inciso que la madre, si comparamos el hijo con la madre, en el caso del inciso (A) el hijo tuvo un mayor campo electromagnético normalizado, lo que implica que no necesariamente los hijos son mejores, esto es porque los cambios en la operación cruce son aleatorios, sin embargo en el caso del inciso (D) se puede observar que el hijo si tuvo mejoras con respecto a la madre esto es la parte del algoritmo que nos permite ir minimizando el campo eléctrico normalizado mientras van avanzando las generaciones.

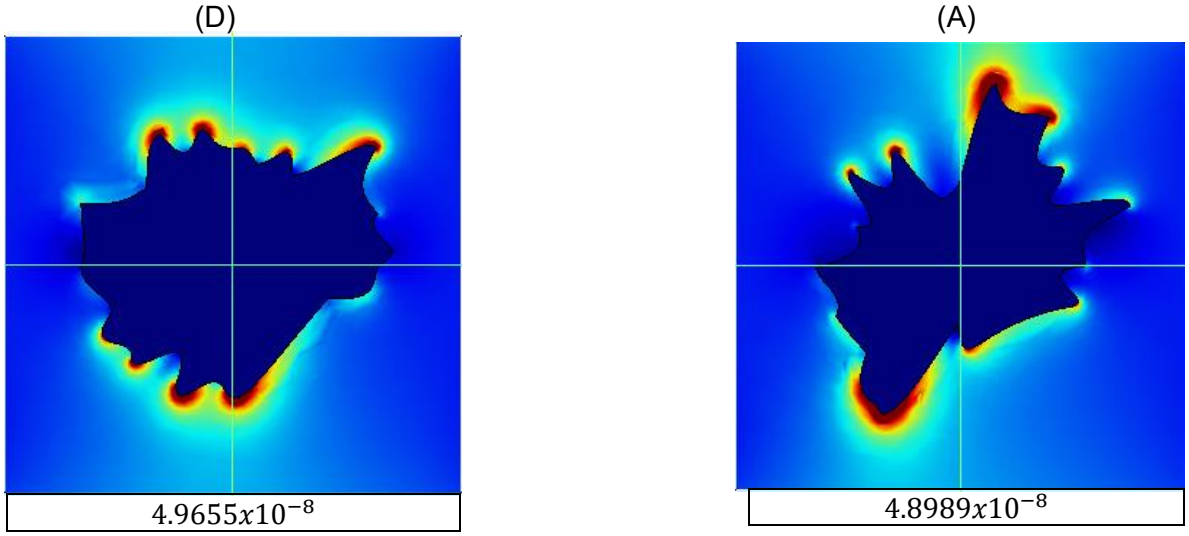


Figura 5.2 Hijos de los sobrevivientes de la población inicial de 4 geometrías aleatorias, representando la respuesta campo eléctrico normalizado v/m

Otra prueba que se hizo fue ya con 10 individuos y durante 30 Generaciones, un total de 155 geometrías, esto para tener una mayor variabilidad genética, es decir mientras más combinaciones de individuos en combinación con una mayor cantidad de generaciones para realizar cambio entre madre e hijo las geometrías tendrán más combinaciones para poder minimizar el campo eléctrico normalizado. En la Figura 5.3 se ven dos listas, la de la derecha muestras las 10 geometrías optimas y a la izquierda las 10 geometrías con peor respuesta.

Figura 5.3 Lista de archivos generados en una corrida con una población inicial de 10 individuos y 30 generaciones

El formato de estas listas es como sigue:
f_xxxxx indica la magnitud del campo eléctrico normalizado.
Inx indica una etiqueta que se le asigna a cada individuo donde la madre e hijo tienen la misma etiqueta.
Gex indica la generación de dicho individuo.
Con esto podemos observar que como se esperaría algunos de los individuos de la primera generación se encuentran en a la izquierda que son los archivos de mayor pérdida de energía, sin embargo también de las últimas generaciones como la 28 se encuentra dentro de las peores geometrías, esto debido a los cambios aleatorios, en la lista de las mejores geometrías (derecha) también se observan de varias generaciones por lo que se optó por hacer más pruebas esto para comprobar si con más generaciones y con poblaciones iniciales más grandes, se llegó a 100 individuos y 100 generaciones, para esta cantidad la respuesta entre generaciones es muy pequeña y no se mejoró la respuesta en las últimas iteraciones, para este caso de estudio se eligió 40 geometrías y 40 generaciones como la cantidad ideal para llegar a un óptimo, como se muestra en la Figura 5.4 se llegó a un mejor resultado minimizando el campo eléctrico normalizado comparado con las pruebas anteriores donde se usaron menos generaciones e individuos.

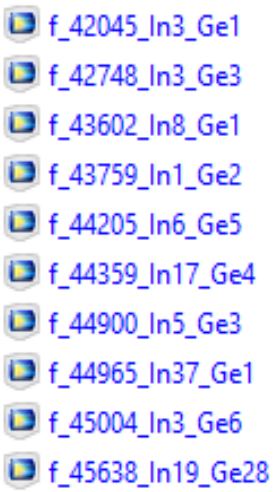


Figura 5.4 Lista de archivos generados en una corrida con una población inicial de 40 individuos y 40 generaciones.

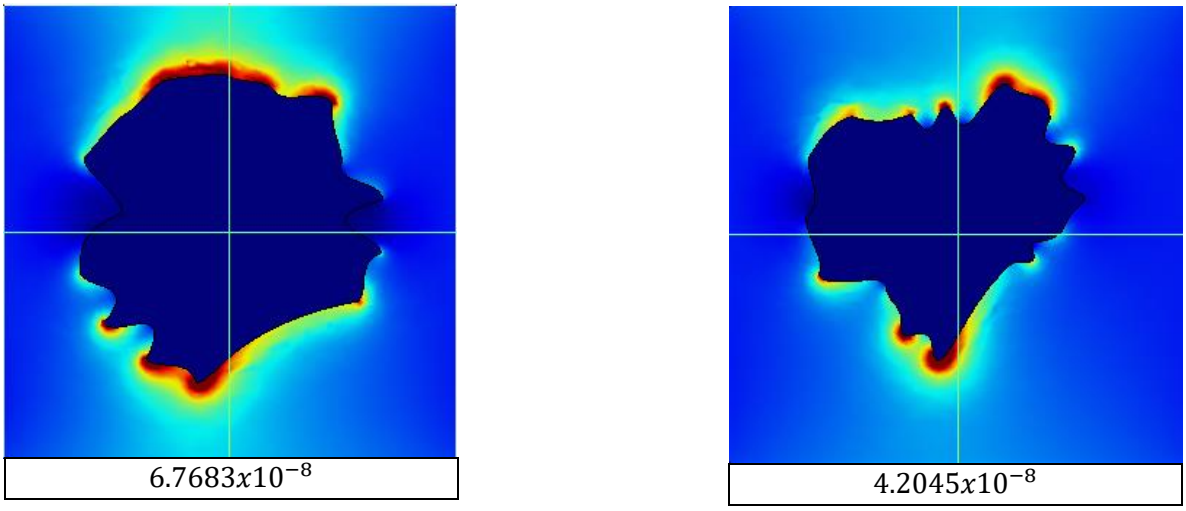


Figura 5.5 (Derecha) Geometria de material oro con menor campo eléctrico nomalizado v/m, (Izquierda) geometria de material oro con mayor campo eléctrico nomalizado en v/m.

En la Figura 5.5 se presenta la mejor geometría (izquierda) y la peor geometría (derecha), en relación a la magnitud del campo eléctrico normalizado resultado de correr el AGS con una población inicial de 40 individuos y 40 generaciones.

Como prueba adicional se propuso que las geometrías de la población inicial comenzaran con una forma geométrica determinista, un círculo, y a partir de ese punto comenzaran a evolucionar en la Figura 5.6 se muestra una población inicial de 4 geometrías de las cuales los individuos (C) y (D) son los que tienen menor campo eléctrico normalizado, lo que significa que sobrevivirán a las siguientes generaciones, las otras dos (A),(B) serán eliminadas, una vez que se

realiza la operación cruce se tiene como resultado las geometrías en la Figura 5.7, en donde los cambios en las geometrías son aleatorios por lo que a partir de la segunda generación dejan de ser formas circulares, al ser cambios pequeños nos da como resultado otro tipo de geometrías aleatorias.

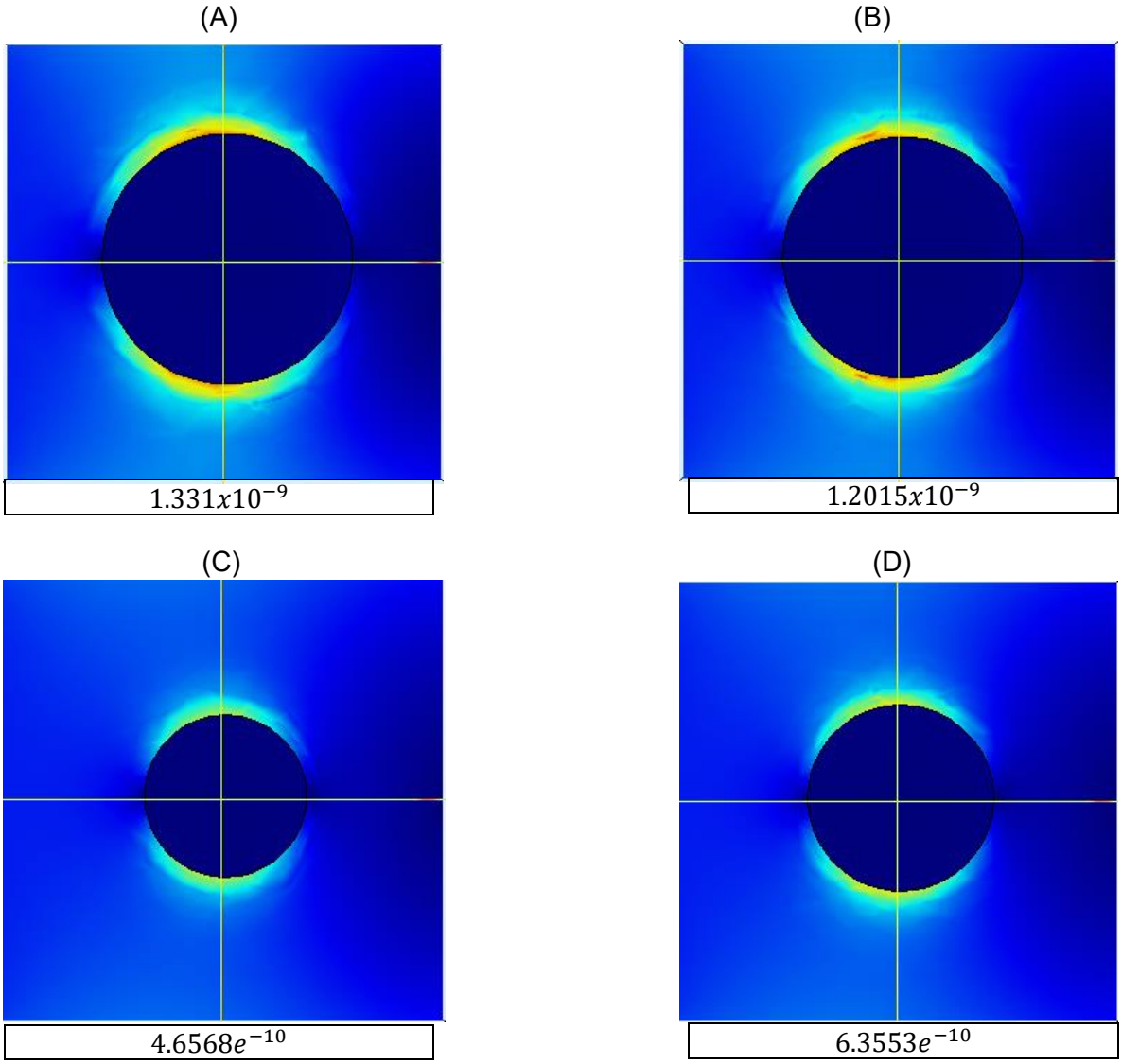


Figura 5.6 Prueba con Geometrias circulares con una poblacion inicial de 4 individuos y 4 generaciones

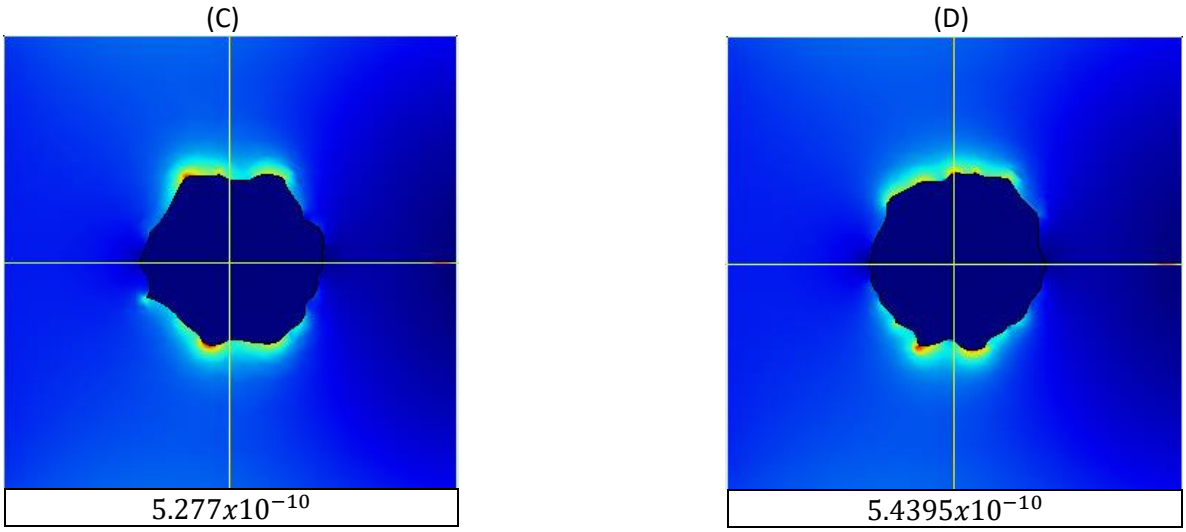


Figura 5.7 Hijos de los sobrevivientes de la población inicial de 4 geometrías circulares, representando la respuesta campo eléctrico normalizado v/m

Como se discutió anteriormente se requiere de pruebas en las cuales el número de individuos sea al menos 40 y las generaciones hasta 40, el resultado de esta prueba se ve en la Figura 5.8 en donde dentro de las dos mejores geometrías, las que tienen menor campo eléctrico normalizado, al ser de la 1 primer generación se sabe que son círculos, sin embargo también hay geometrías que son de generaciones más avanzadas se sabe que han evolucionado por lo tanto ya no son círculos perfectos.

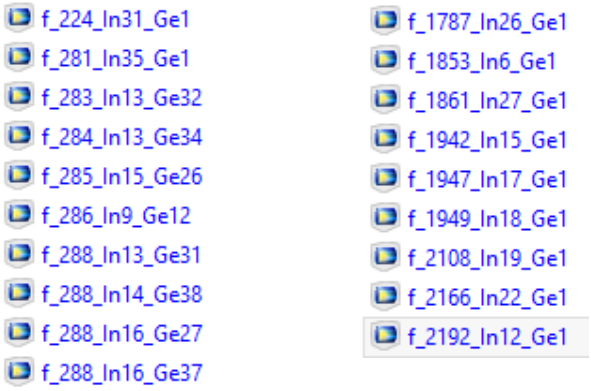


Figura 5.8 Listas de Prueba con población Inicial de 40 Individuos y 40 generaciones, derecha: lista de las 10 mejores geometrías, Izquierda: Lista de las 10 peores geometrías.

En la Figura 5.9 se presenta la mejor geometría circular (izquierda) y la peor geometría circular (derecha), en relación a la magnitud del campo eléctrico normalizado resultado de correr el AGS con una población inicial de 40 individuos y 40 generaciones.

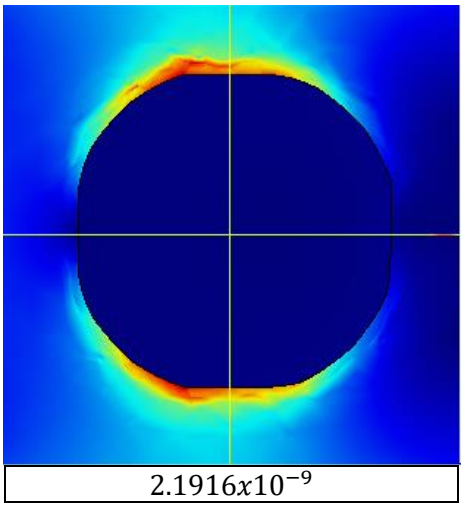


Figura 5.9 (Derecha) Geometria de material oro con menor campo eléctrico nomalizado v/m, (izquierda) geometria de material oro con mayor campo eléctrico nomalizado en v/m.

Resultados del Capítulo 3

En este capítulo se usan dos geometrías con una unión entre ellas, de material oro, una onda electromagnética con una frecuencia de 1Thz, se realizaron 3 diferentes pruebas:

1. Dos geometrías generadas aleatoriamente.
2. Una geometría generada aleatoriamente y una geometría circular
3. Dos geometrías circulares.

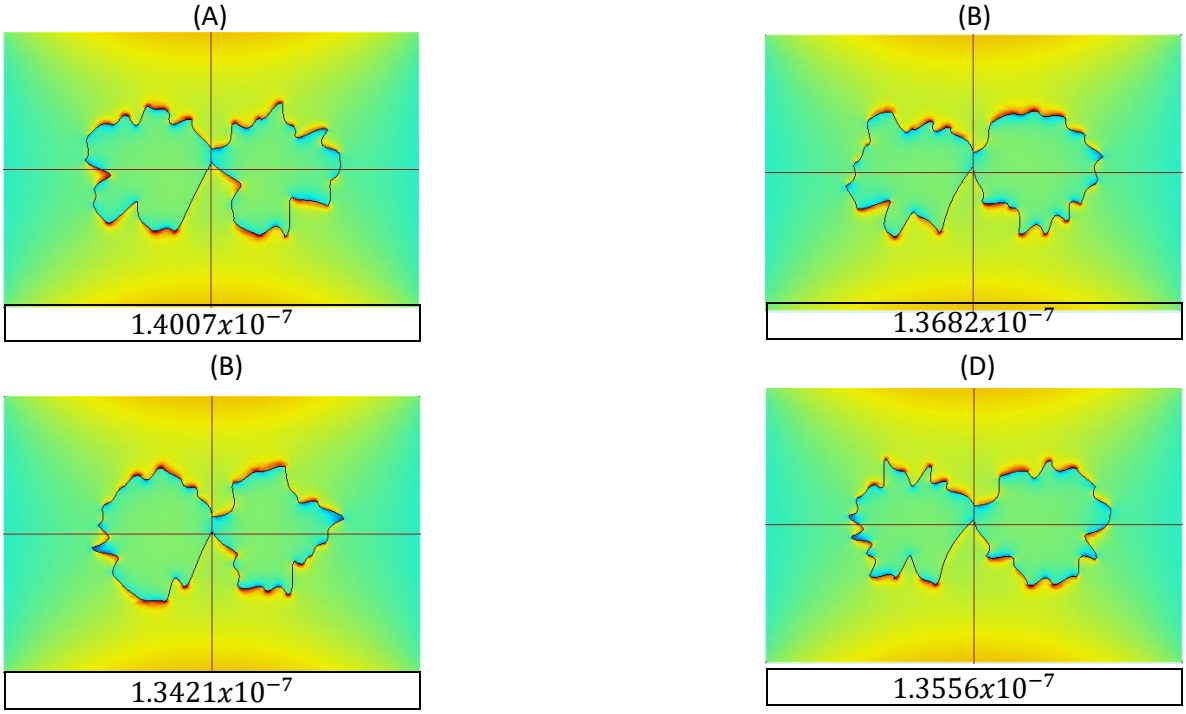


Figura 5.10 Primer prueba con una población inicial de 4 generaciones y 4 pares de individuos, representando la respuesta campo eléctrico normalizado v/m

Para la prueba 1 se corre el algoritmo genético simple dos geometrías generadas aleatoriamente, en la Figura 5.10 se tienen 4 geometrías como población inicial, de las cuales las geometrías con menores pérdidas son la (C) y (D), y sus hijos con igual etiqueta en la Figura 5.11 se ve que son muy parecidos a sus padres, esto nos sirve para observar que tan bruscos son los cambios entre madre e hijo y dependiendo de la aplicación aumentar o disminuir esos cambios, tomando en cuenta que cambios muy grandes de información genética en la operación de cruce se reflejan en cambios bruscos en las curvas de la geometría, y cambios de información genética muy pequeños, harán que la evolución de la geometría sea demasiado tardado.

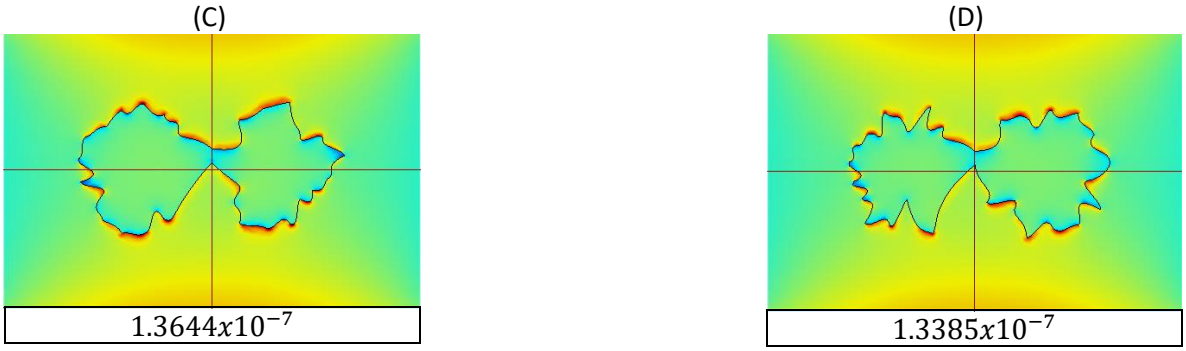


Figura 5.11 Hijos de los sobrevivientes de la población inicial de 4 pares geometrías aleatoria, representando la respuesta campo eléctrico normalizado v/m

Se termino esta primera prueba de dos geometrías generadas aleatoriamente con una corrida de 40 generaciones y 40 individuos, los resultados se observan en las listas de la *Figura 5.12*, la lista de los mejores resultados muestran que dentro de los 10 primeros se pueden encontrar geometrías generadas incluso en la primer generación, sin embargo como se esperaría en la lista de los peores resultados se encuentras solamente geometrías de las primeras generaciones.

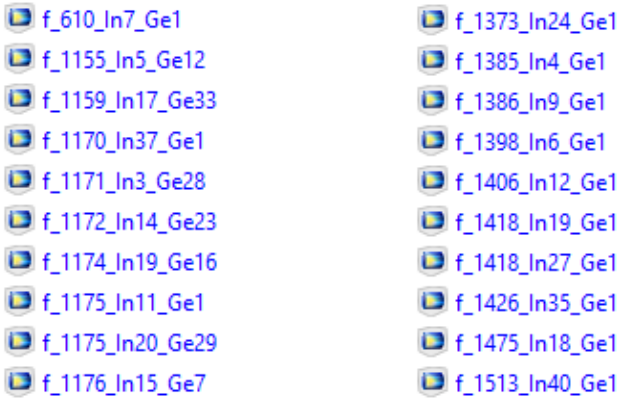


Figura 5.12 Listas de la primera prueba con población Inicial de 40 Individuos y 40 generaciones, derecha: lista de las 10 mejores geometrías, Izquierda: Lista de las 10 peores geometrías.

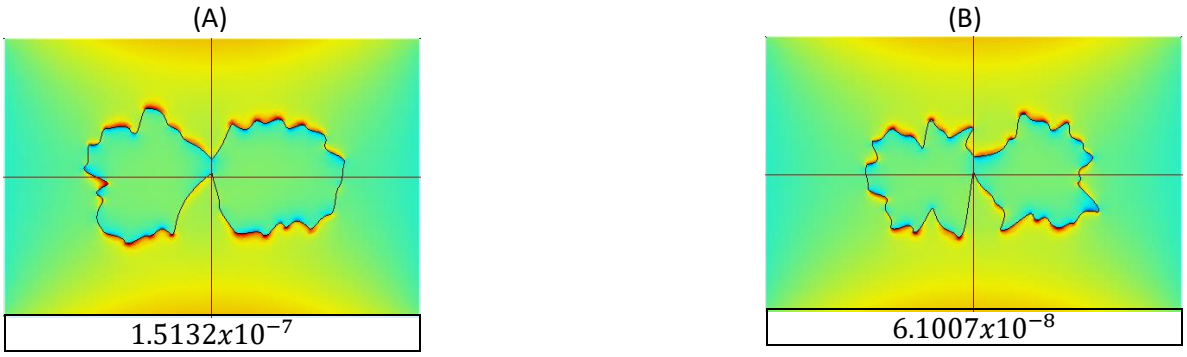


Figura 5.13 (A) par de geometrias de material oro con mayor campo eléctrico nomalizado en v/m. (B) Par de geometrias de material oro con menor campo eléctrico nomalizado v/m,

En la Figura 5.13 se presenta el mejor par de geometrías (A) y el peor par de geometrías (B), en relación a la magnitud del campo eléctrico normalizado resultado de correr el AGS con una población inicial de 40 individuos y 40 generaciones.

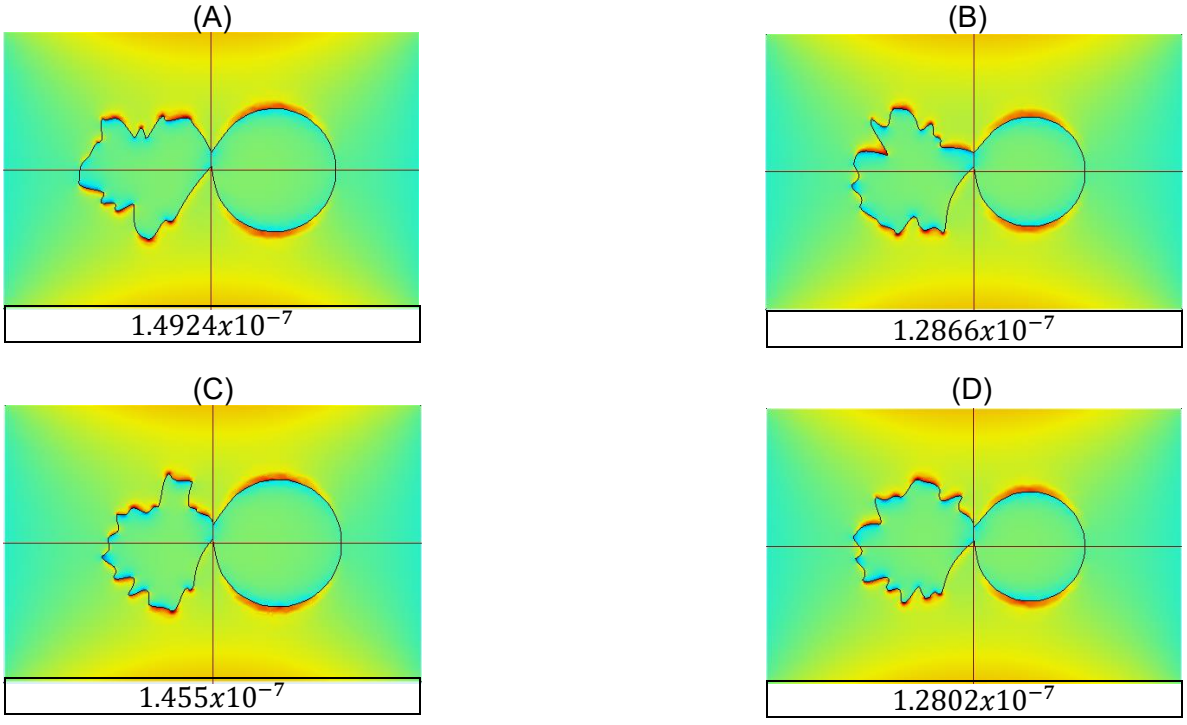


Figura 5.14 Primer prueba con una población inicial de 4 generaciones y 4 pares de individuos representando la respuesta campo eléctrico normalizado en v/m.

Como se vio en el capítulo anterior generar círculos como geometría inicial dio buenos resultados por lo que se optó como segunda prueba una geometría aleatoria y una geometría circular por que se realizado una corrida de 4 individuos y 4 generaciones como en la Figura 5.14 donde el par de geometrías (B) y (C) son las que tienen menor campo

eléctrico normalizado por lo que son las que sobrevivirán a la siguiente generación y se reproducirán, mediante la operación de cruce se obtienen los hijos en la Figura 5.15 donde uno de los parámetros que observo es el área de la unión entre geometrías, no cambia entre generaciones, esto es debido a que ya no forma parte de la evolución de ambas geometrías, es constante a través del paso de las generaciones.

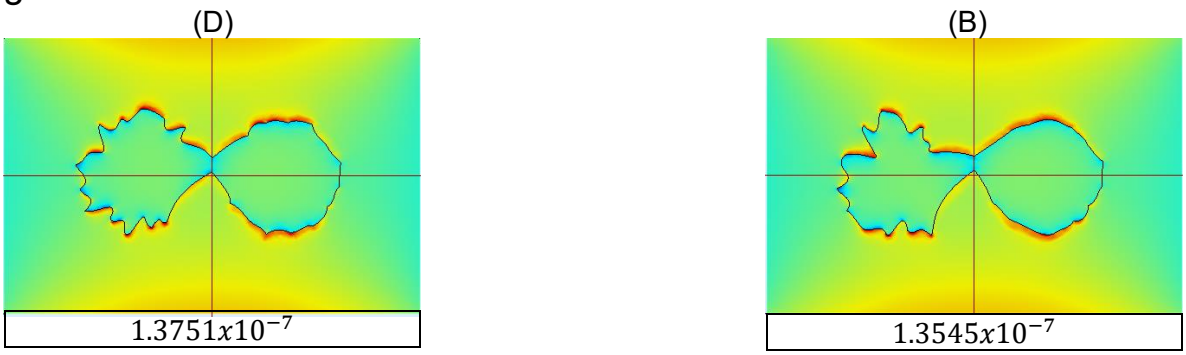


Figura 5.15 Hijos de los sobrevivientes de la población inicial de 4 par de geometrías, una aleatoria y una circular, representando la respuesta campo eléctrico normalizado v/m

Como se definió anteriormente se corrió una prueba con 40 individuos y 40 generaciones, se tiene un par de listas

Figura 5.16, a la derecha las geometrías con la mayor magnitud del campo eléctrico normalizado y se observa que todas son de la primera generación, la lista de la derecha son las geometrías con menor campo eléctrico normalizado.

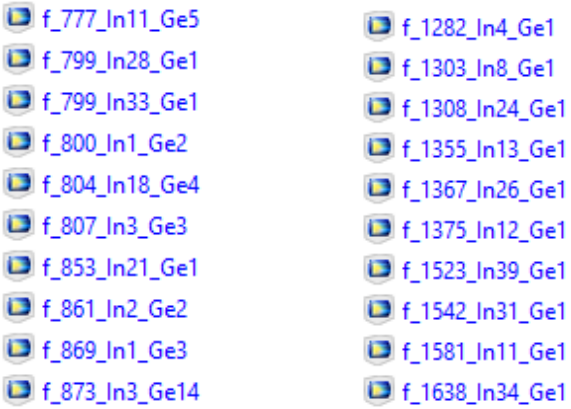
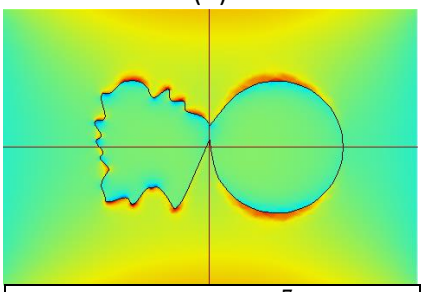


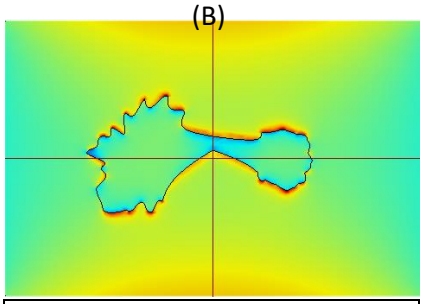
Figura 5.16 Listas de la segunda prueba con población Inicial de 40 Individuos y 40 generaciones, derecha: lista de las 10 mejores geometrías, Izquierda: Lista de las 10 peores geometrías.

(A)



$1.6384x10^{-7}$

(B)



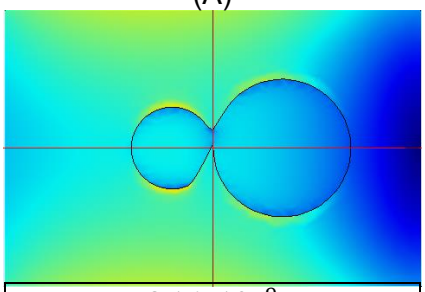
$7.7651x10^{-8}$

Figura 5.17 (derecha) Par de geometrias de material oro con menor campo eléctrico nomalizado v/m, (izquierda) par de geometrias de material oro con mayor campo eléctrico nomalizado en v/m.

En la Figura 5.17 se presenta el mejor par de geometrías (B) y el peor par de geometrías (A), en relación a la magnitud del campo eléctrico normalizado resultado de correr el AGS con una población inicial de 40 individuos y 40 generaciones.

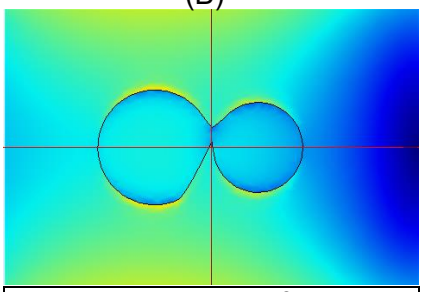
La tercera prueba que se realizo es con dos círculos como geometría si observamos las primeras geometrías de esta corrida Figura 5.18 son 4 individuos de los cuales una vez que se les aplico la onda electromagnética a 1THz su respuesta el campo eléctrico normalizado en v/m tienen mejor respuesta al inicio del algoritmo comparándolo con las 2 pruebas anteriores, después de la selección natural, los individuos que sobreviven (C) y (D) los cuales se cruzan, se obtienen los hijos con las mismos incisos en la Figura 5.19 los cuales se observan que no mejoran con respecto a la madre sin embargo, geometría (C), la cual para la tercera generación en la operación de selección si sobrevivirá porque tiene mejor respuesta que los 4 individuos de la segunda generación, los 2 madres y sus 2 hijos, las geometrías hijo tienen una forma diferente a los tipos de geometrías que se forman con respecto a las pruebas anteriores, se nota que tienen cambios pequeños entre cada curva Bézier.

(A)



$2.16x10^{-9}$

(B)



$4.2416x10^{-9}$

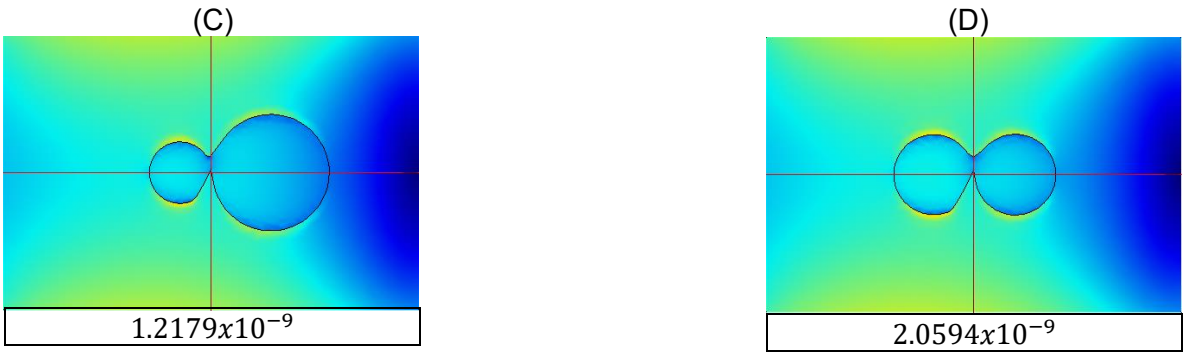


Figura 5.18 Primer prueba con una población inicial de 4 generaciones y 4 pares de individuos representando la respuesta campo eléctrico normalizado en v/m

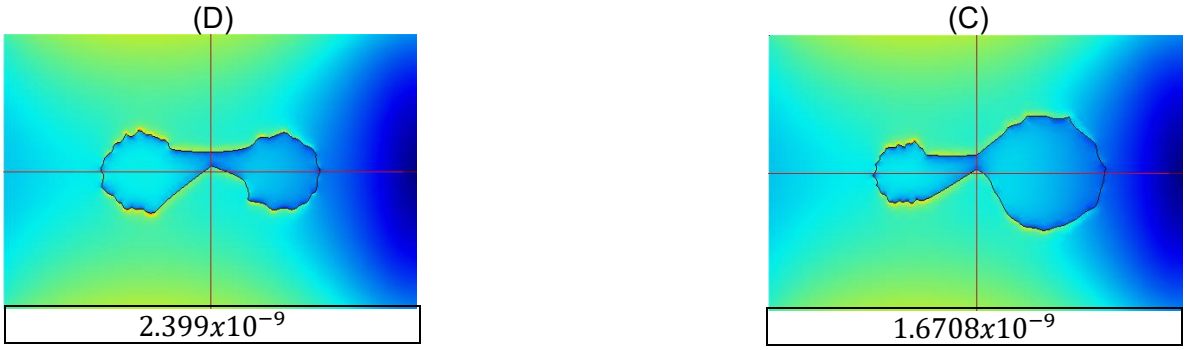


Figura 5.19 Hijos de los sobrevivientes de la población inicial de 4 par de geometrías, una aleatoria y una circular, representando la respuesta campo eléctrico normalizado v/m.

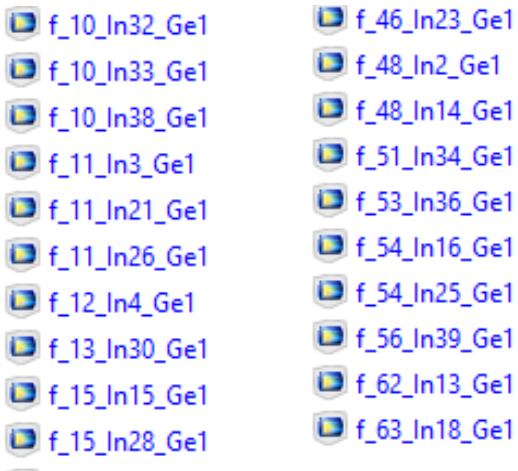


Figura 5.20 Listas de la tercera prueba con población Inicial de 40 Individuos y 40 generaciones, derecha: lista de las 10 mejores geometrías, Izquierda: Lista de las 10 peores geometrías.

Se corrió una prueba con 40 individuos y 40 generaciones, en la

Figura 5.16, a la derecha el par de geometrías con la mayor magnitud del campo eléctrico normalizado, la lista de la derecha son las geometrías con la menor magnitud del campo eléctrico normalizado.

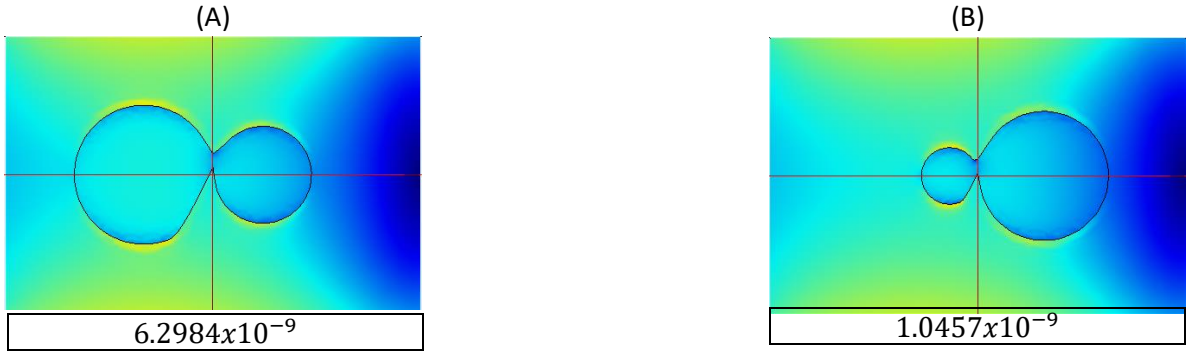


Figura 5.21 (Izquierda) Par de geometrias de material oro con menor campo eléctrico nomalizado v/m, (Derecha) par de geometrias de material oro con mayor campo eléctrico nomalizado en v/m.

En la Figura 5.21 se presenta el mejor par de geometrías (B) y el peor par de geometrías (A), en relación a la magnitud del campo eléctrico normalizado resultado de correr el AGS con una población inicial de 40 individuos y 40 generaciones.

Resultados del Capítulo 4

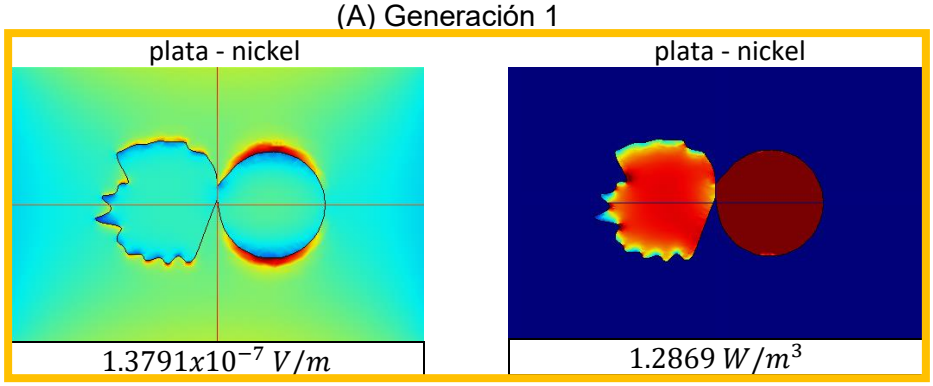
En este capítulo se muestran los resultados de tener 2 geometrías con una unión, diferentes materiales, estos materiales deben tener diferente signo en su coeficiente seebeck, se aplica una onda electromagnética con frecuencia de 1THz, para este caso de estudio se tienen 2 mediciones, el campo eléctrico normalizado que es con el que ya se ha estado trabajando y la densidad de disipación de potencia total en W/m^3 esto para encontrar la mejor par de geometrías con su combinación de material, para el cual se tenga el mejor efecto seebeck, para determinarlo se utiliza un Algoritmo Genético Multiobjetivo es cual mediante la optimalidad de Pareto transforma un problema de N objetivos a un problema de un solo objetivo.

En este caso de estudio se propone aumentar el número de individuos de la población inicial esto debido a que el espacio de soluciones aumento de manera significativa al incrementar de un material a nueve combinaciones de materiales por lo cual para tener una mayor diversidad genética se utilizan 100 individuos, la cantidad de generaciones se deja en 40.

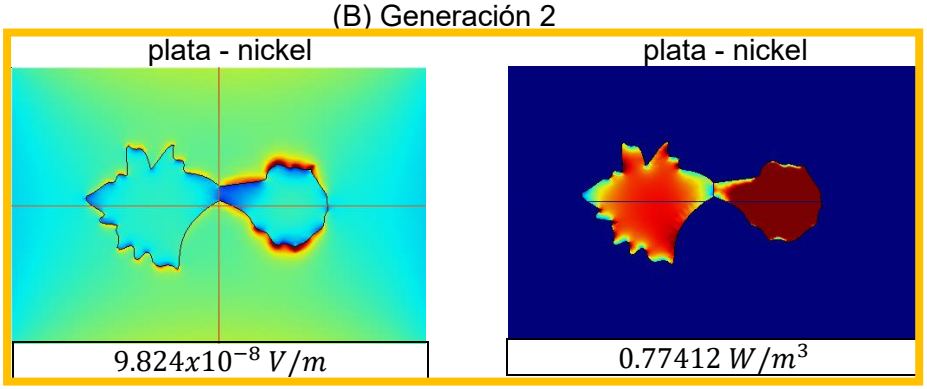
Se realizo una corrida con estas características, en la Figura 5.22 se observa el seguimiento la descendencia de un individuo, un par de

geometrías unidas, haciendo una analogía seria la madre, el hijo, el nieto y bisnieto.

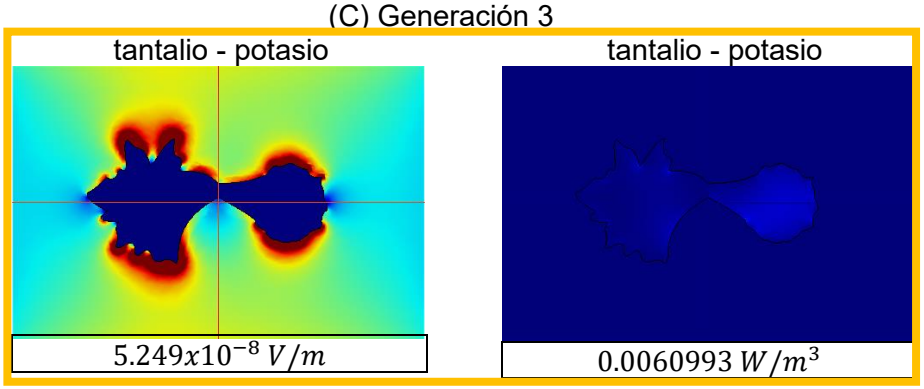
(A) Generación 1



(B) Generación 2



(C) Generación 3



(D) Generación 4

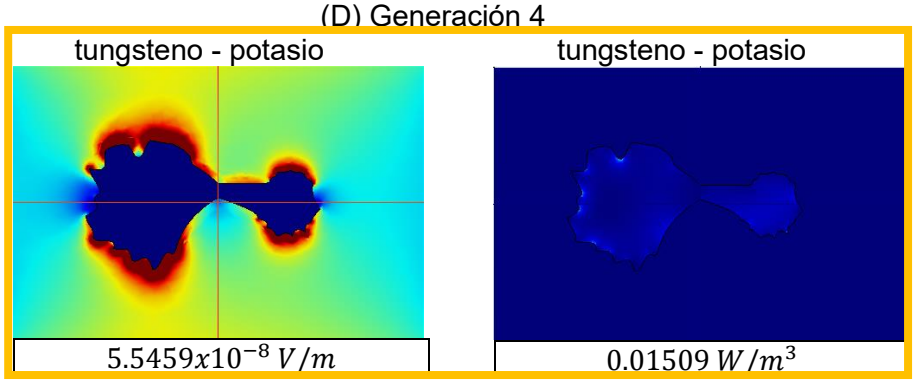


Figura 5.22 Decendencia de un individuo durante 4 generaciones, a la derecha: la representación de la respuesta densidad de disipación de potencia total, a la izquierda: la representación de la respuesta campo eléctrico normalizado.

En la Figura 5.22 se observa que entre la generación 1(A) y la generación 2(B) el cambio es únicamente en la forma de las geometrías ya que la combinación de materiales en las dos primeras generaciones plata y níquel no cambio, y se da el caso donde el hijo (B) minimizo ambas respuestas el campo eléctrico normalizado y la densidad de disipación de potencia total con respecto a su madre (A), sin embargo para la generación 3 (C) ya hay un cambio en los materiales tantalio y potasio y a su vez una reducción importante en las dos respuestas, incluso cuando los cambios en la forma de las geometrías son pequeños.

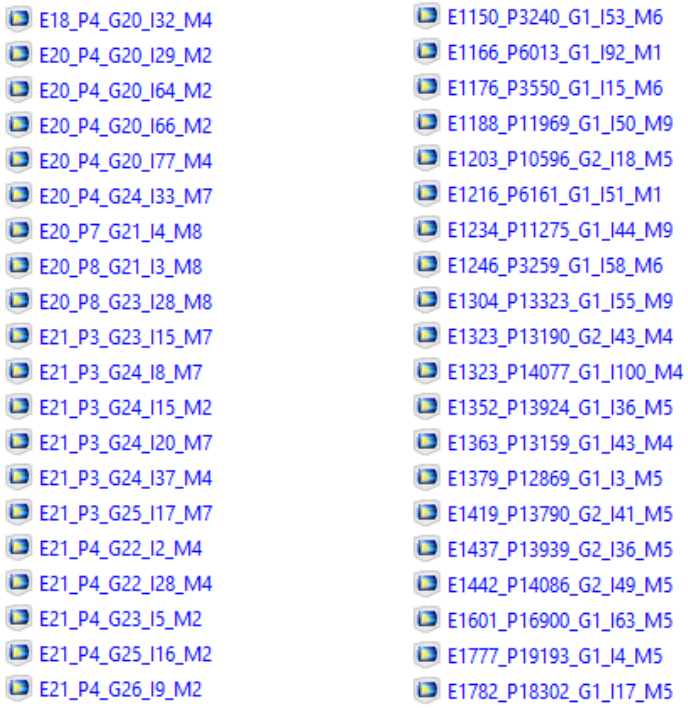


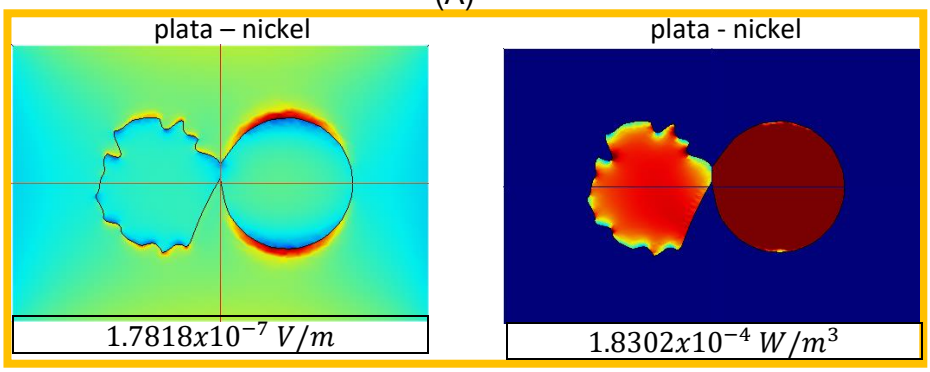
Figura 5.23 Derecha: Lista de las 20 par de geometrias con el menor campo eléctrico normalizado, Izquierda: Lista de las 20 par de geometrias con mayor campo eléctrico normalizado.

En la Figura 5.23 se tienen dos listas, las cuales tienen el siguiente formato:
Exxxx la magnitud del campo eléctrico normalizado.
Pxxxx la magnitud de la densidad de disipación de potencia total.
Gxx Numero de generación.
Ixx Identificador de cada individuo, madre e hijo tienen el mismo identificador.

Mx Combinación de materiales un total de 9, el 0 indica que no hay cambio de material.

La lista de la derecha que son los que menor campo eléctrico normalizado tienen se puede observar los materiales dentro de dicha lista, en orden ascendente son: 5 individuos con materiales níquel - tantalio (M4), 7 individuos con materiales bizmuto - tantalio (M2), 5 individuos con materiales potasio – tantalio (M7), 3 individuos con materiales potasio – tungsteno (M8). En la lista de la izquierda que es la que tiene mayor campo eléctrico normalizado de energía, las combinaciones de materiales que se encuentran aquí en orden descendentes: 9 individuos con materiales niquel – plata (M5),3 individuos con materiales niquel – tantalio (M4),3 individuos con materiales niquel – potasio (M9),3 individuos con materiales potasio plata (M6),2 individuos con materiales bizmuto – plata (M1). Con lo que se nota una clara discriminación de los materiales con respecto a las respuestas medidas a excepción del material M4 niquel -tantalio el cual se encuentra en ambas listas.

(A)



(B)

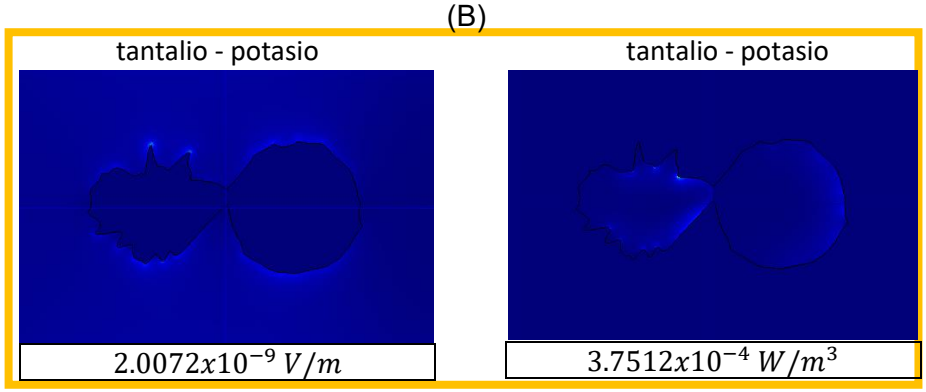


Figura 5.24 (A) Par de geometrias de material plata – niquel, (B) Par de geometrias de material tantalio – potasio; representando el campo eléctrico normalizado rectangulo de la derecha, representando la densidad de disipacion de potencia total rectangulo de la izquierda.

En la Figura 5.24 se nota el contraste de las respuestas entre el par de geometrías con menor campo eléctrico normalizado y densidad de disipación de potencia total (B) y el par de geometrías con mayor campo eléctrico normalizado y densidad de disipación de potencia total (A) después de correr el algoritmo genético multiobjetivo con población inicial de 100 individuos y 40 generaciones.

Conclusiones

En los resultados del capítulo 2 se obtiene la geometría de material oro con menor campo eléctrico normalizado para dos casos, uno con la población inicial generando geometrías aleatorias, el segundo caso con la población inicial con geometrías circulares, para lo cual se obtuvieron menor campo eléctrico normalizado con las geometrías circulares esto debido a que en este caso al generar los círculos solo varia el radio y esto nos da una mayor diversidad de tamaños a diferencia de las geometrías aleatorias sus tamaños son uniformes la mayor variación está en la forma, es por eso que vemos mejores resultados en las geometrías pequeñas.

En el capítulo 3 se hicieron 3 pruebas: 1) las dos geometrías generadas aleatoriamente, 2) una geometría aleatoria y una geometría circular, 3) ambas geometrías circulares. En este caso de estudio dieron mejores resultados la prueba 3) esto debido a que el circulo su periferia no tiene cambios bruscos, además si ambas geometrías son simétricas las pérdidas de energía son aun menores, esto se logra con mayor recurrencia con un par de círculos que con geometrías aleatorias.

Capítulo 4 una de las principales características de este caso de estudio es que van cambiando los materiales y en los resultados que se obtuvieron justo esta característica define en mayor parte si la función de salud disminuya o aumente su magnitud considerablemente, con lo cual se puede discriminar que combinación de materiales nos permiten minimizar el campo eléctrico normalizado y la densidad de disipación de potencia total, los materiales que dieron los resultados son (níquel – tantalio), (bizmuto – tantalio), (potasio – tantalio) (potasio – tungsteno), además el algoritmo también encuentra las geometrías para las cuales se cumplen los objetivos planteados.

En términos generales según los resultados que se obtuvieron en los 3 capítulos de este trabajo es claro que el algoritmo obtiene geometrías que van minimizando el campo eléctrico normalizado o la densidad de

disipación de potencia total mientras van avanzando las generaciones, en determinado número de generaciones, generalmente 40 iteraciones, se llega un punto en donde los resultados oscilan entre mejorar y empeorar, por lo que se toma como convergencia del algoritmo, y por lo tanto se llega a una geometría optima, como no es una técnica clásica de diseño de nano antenas, no se tiene un punto de partida ni de comparación, por lo que se propone como trabajo a futuro construir algunas de las geometrías propuestas por nuestro algoritmo como optimas, para experimentalmente determinar su respuesta a una onda electromagnética del orden de 1THz para comparar con los resultados de simulación, a su vez proponer diferentes algoritmos que generen otro tipo geometrías, también proponer más combinaciones de materiales y otras variables a medir como por ejemplo temperatura y diferencia de potencial, aprovechado la flexibilidad del código que se propuso en este trabajo.

Una aplicación propuesta como trabajo a futuro para dar seguimiento a estos resultados que se comentaron puede ser los orificios plasmonicos, los cuales son estructuras que tienen una resonancia nula al aplicarles una onda electromagnética de frecuencias del orden de 1THz.

Capitulo 6. Referencias

[1] D. Halladay y R. Resnik, Fisica Vol. 2, Mexico: Compañia Editorial Continental, S.A. DE C.V., 1999.

[2] E. Hetch, OPTICS, San Francisco: Addison Wesley, 2002.

[3] E. Briones, J. Briones, A. Cuadrado, J. C. Martinez-Anton, S. McMurtry, M. Hehn, F. Montaigne, J. Alda y F. Gonzalez, «Seebeck Nanoantennas for solar energy Harvesting,» *Applied Physics Letters,* 2014.

[4] J. E. Sanchez, R. D. d. L. Zapata, F. M. Santoyo, G. Gonzales, M. J. Yacaman, A. Ponce y F. J. Gonzales, «Resonance properties of Ag-ZnO nanoestructures at terahertz frecuencies,» *OPTIC EXPRESS,* vol. 23, nº 19, 2015.

[5] D. A. Coley, An introduccion to Genetic Algorithms for Scientist and engineers, Singapore: World scientific publishing Co. Pte. Ltd., 1999.

[6] D. S. Linden, «Antenna desing using genetic algorithms,» *Proceedings of the 4th Annual Conference on Genetic and Evolutionary Computation,* pp. 1133-1140, 2002.

[7] R. D. d. Leon-Zapata, G. Gonzalez, E. Flores-Garcia, A. G. Rodriguez y F. J. Gonzalez, «Evolutionary Algorithm Geometry Optimization of Optical antenas,» *International Journal of Antennas and Propagation,* p. 7, 2016.

[8] R. D. d. L. Zapata, G. Gonzaalez, E. F. A. G. Rodriguez y F. Gonzalez, «Genetic algorithm for geometry optimization of optical antennas,» *COMSOL Conference in Boston,* 2016.

[9] S. Biswas y B. C. Lovell, Bezier and Splines in Image Processing and Machine Vision, Londres: Springer, 2008.

[10] R. D. d. Leon-Zapata, J. V. Gonzalez-Fernandez, E. Flores-Garcia, A. B. d. l. R. Zapata y I. Lara.Velzquez, «Multi-objetive evolucionary algorithm as a method to obtain optimized nanoestructures,» 2017.

[11] S. A. Maier, Plasmonics, Fudamentals and applications, Bath: Springer, 2007.

[12] A. Chinchuluun, P. M. Pardalos, A. Migdalas y L. Pitsouis, Pareto Optimality, Game theorry and equilibria, Springer, 2008.

[13] H.-J. Hagemann, W. Gudat y C. Kunz, «Optical constants from the far infrared to the x-ray region: Mg, Al, Cu, Ag, Au, Bi, C and AL2O3,» *Journal of the Optical Society of America,* vol. 65, nº 6, pp. 742-744, 1975.

[14] M. A. Ordal, R. J. Bell, R. W. Alexander, L. L. Long y M. R. Querry, «Optical properties of fourteen metals in the infrared and far infrared: Al, Co, Cu, Au, Fe, Pb, Mo, Ni, Pd, Pt, Ag, Ti, V, and W.,» *Optical Socity of America,* vol. 24, nº 24, pp. 4493-4499, 1985.

[15] J. Jin, The finite element method in electromagnetics, New York: John Wiley and sons, Inc., 2002.

[16] P. Bezier, The mathematical basis of the UNISURF CAD system, Londres: BUTTERWORTHS and Co., 1986.

Capitulo 7. Anexos

Anexo A Código Algoritmo Genético Simple para una geometría y material oro.

```
function [] = AGS
clear all;
format long;
import com.comsol.model.*
import com.comsol.model.util.*
%%%%%%%%%%%%%%%%%PARAMETROS%%%%%%%%%%%%%%%%%%%%
% % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % %
NumGeo = 40;  %Poblacion Inicial
NumCur = 17;   %Numero de curvas en una geoetrias
Grado = 3;         % Grado de cada curva bezier
NC = (NumCur*Grado)+1;
TAMX = 1024; %Tamanio del universo de soluciones en X
TAMY = 1024;
Centro = [TAMX/2,TAMY/2]; %Centro de la geometria
Radiomin = 200; % Tamaño minimo de los vectores con respecto al centro
if(TAMX <= TAMY)
   RadioMax = (TAMX/2)+50; %Radio maximo de los vectores
else
   RadioMax = (TAMY/2)+50;
end
%%%%%%%%%%%%%%Funcion que genera N geometrias%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
[X, Y,Magnitud,Angulo] = GenMulGeo(NumGeo,NC,Grado,TAMX,TAMY,Radiomin,RadioMax,Centro);
Z = zeros(1,(NumCur*Grado)+1);
G = ones(1,NumCur).*Grado;
P = ones(1,NumCur*(Grado+1));
tbl_data = zeros(NumGeo,1);
fit = [];
band = 0;
%%%%%%%%%%%%%%%%%%Iteraciones (Numero de generaciones) %%%%%%%%%%%%%%5
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for j=1:40
   if band == 0
      numgeome = NumGeo;
   else
      numgeome = NumGeo/2;
   end
for i=1:numgeome%Numero de geometrias
   %%%%%%%%%%%%%Codigo generado por COMSOL%%%%%%%%%%%%%%%%%%%
   %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
   model = ModelUtil.create('Model');
   model.modelPath('D:\fisicas');
   model.label('fisicaA.mph');
   model.comments(['FisicaA\n\nExtruir1\n\nDos\n\nUntitled\n\n']);
   model.component.create('comp1', true);
   model.component('comp1').geom.create('geom1', 3);
   model.result.table.create('tbl1', 'Table');
   model.component('comp1').mesh.create('mesh1');
   model.component('comp1').geom('geom1').lengthUnit('nm');
   model.component('comp1').geom('geom1').create('wp1', 'WorkPlane');
   model.component('comp1').geom('geom1').feature('wp1').set('quickz', 1024);
   model.component('comp1').geom('geom1').feature('wp1').set('unite', true);
   model.component('comp1').geom('geom1').feature('wp1').geom.create('b1', 'BezierPolygon');
   %%%%%%%% valores de X, Y para generar las curvas bezier%%%%%%%%%%%%%%%%
   model.component('comp1').geom('geom1').feature('wp1').geom.feature('b1').set('p', [X(i,:);Y(i,:)]);
   model.component('comp1').geom('geom1').feature('wp1').geom.feature('b1').set('degree',G);
   model.component('comp1').geom('geom1').feature('wp1').geom.feature('b1').set('w', P);
   model.component('comp1').geom('geom1').feature('wp1').geom.feature('b1').set('type', 'open');
   model.component('comp1').geom('geom1').feature('wp1').geom.create('csol1', 'ConvertToSolid');
   model.component('comp1').geom('geom1').feature('wp1').geom.feature('csol1').selection('input').set({'b1'});
   model.component('comp1').geom('geom1').create('ext1', 'Extrude');
   model.component('comp1').geom('geom1').feature('ext1').setIndex('distance', '50', 0);
   model.component('comp1').geom('geom1').feature('ext1').selection('input').set({'wp1'});
   model.component('comp1').geom('geom1').create('blk1', 'Block');
   model.component('comp1').geom('geom1').feature('blk1').set('pos', [-250 -250 0]);
   model.component('comp1').geom('geom1').feature('blk1').set('size', [1500 1500 2048]);
   model.component('comp1').geom('geom1').feature('blk1').set('layername', {'Layer 1'});
   model.component('comp1').geom('geom1').feature('blk1').setIndex('layer', '1024', 0);
   model.component('comp1').geom('geom1').run;
   model.component('comp1').geom('geom1').run('fin');
   %%%%%%%%%%Funcion que abarca las caracteristicas que genera COMSOL para
   %%%%%%%%%%material plata
    fAu(model)
```

```
74        %%%%%%%%%%%%%%%Funcion que regresa desde COMSOL la respuesta en una
75        %%%%%%%%%%%%%%tabla
76 -      str = mphtable(model,'tbl1');
77 -      tbl_data(i) = str.data(2);
78 -      format SHORT
79 -      letras = 'D:/1fisica1Objetivo/f_';
80        %%%%%%%%%%%%%%%%%%%%Guarda el modelo anteriormente generado en formato
81        %%%%%%%%%%%%%%%%%%%mph e COMSOL
82 -      mphsave(model,strcat(letras,num2str(uint16(tbl_data(i)*1e12)),'_In',num2str(i),'_Ge',num2str(j)));
83 -      format LONG
84 -    end
85 -    max = [tbl_data ; fit]
86 -    tbl_data = [];
87 -    if band == 1
88 -      Magnitud = [newM;M];
89 -      Angulo = [newA;A];
90 -    else
91 -      band = 1;
92 -    end
93      % % % % % % % % % Operaciones del algoritmo Genetico SImple %%%%%%
94 -    [M,A,fit] = Seleccion(Magnitud,Angulo,max,NumGeo,NC);
95 -    [newM,newA] = Cruce(M,A,NumGeo,NC);
96 -    [newM] = Mutar(newM,NumGeo,NC,Radiomin,RadioMax);
97      %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
98 -    X = double(uint16((newM.*cos(newA))+Centro(1)));
99 -    Y = double(uint16((newM.*sin(newA))+Centro(2)));
100 -   iteracion = j
101 -   end
102 -   out = model;
```

Funciones secundarias

```
1   function [PuntosX, PuntosY,Radio,Angulo] = GenMulGeo(NG,NC,Grado,TAMX,TAMY,Radiomin,RadioMax,Centro)
2 -   AnguloMax = 2*pi;
3 -   rng('shuffle');
4 -   phase = (rand(1,1)*AnguloMax);
5 -   Angulomin = (360/NC)-1;
6 -   radgrad = 2*pi/360;
7     %%%%%%%%%%Asigna el valor a cada angulo%%%%%%%%%%%%%%%%
8 -   for i = 1:NC
9 -     for j = 1:NG
10 -      Angulo(j,i) = (Angulomin*i*radgrad) + phase;
11 -    end
12 -  end
13    %%%%%%%%%Construye las magnitudes de los vectores%%%%%%%%%%%%%%
14 -    Radio = ones(NG,NC);
15 -    rng('shuffle');
16 -    R = RadioMax-Radiomin;
17 -    Radio(:,1) = (rand(NG,1).*R)+Radiomin;
18 -    band = 1;
19 -    i = 2;
20 -    while i < NC
21 -      if band == 1 %%Mayor
22 -        rng('shuffle');
23 -        R = RadioMax-Radio(:,i-1);
24 -        Radio(:,i) = (rand(NG,1).*R)+Radio(:,i-1);
25 -        i=i+1;
26 -        rng('shuffle');
27 -        R = RadioMax-Radio(:,i-1);
28 -        Radio(:,i) = (rand(NG,1).*R)+Radio(:,i-1);
29 -        band = 0;
30 -      else
31 -        rng('shuffle');
32 -        R = Radio(:,i-1)-Radiomin;
33 -        Radio(:,i) = (rand(NG,1).*R)+Radiomin;
34 -        band = 1;
35 -      end
36 -      i=i+1;
37 -   end
38    %%%%%%%%%%%%%%%%%%%%%%%%Ajuste a los puntos X,Y para no sobrepasar el limite
39    %%%%%%%%%%%%%%%%%%%%%%%%de tamaño
40 -   PuntosX = uint16((Radio.*cos(Angulo))+Centro(1));
41 -   PuntosY = uint16((Radio.*sin(Angulo))+Centro(2));
42 -   PuntosX = double(PuntosX);
43 -   PuntosY = double(PuntosY);
44 -   PuntosX(:,NC) = PuntosX(:,1);
45 -   PuntosY(:,NC) = PuntosY(:,1);
```

```
for i = 1:NC
    for j = 1:NG
        if PuntosX(j,i) < 1
            PuntosX(j,i) = 1;
        end
        if PuntosX(j,i) >= 1024
            PuntosX(j,i) = 1024;
        end
        if PuntosY(j,i) >= 1024
            PuntosY(j,i) = 1024;
        end
        if PuntosY(j,i) < 1
            PuntosY(j,i) = 1;
        end
    end
end
end
```

```
function [] = fAu(model)
model.component('comp1').material.create('mat1', 'Common');
model.component('comp1').material.create('mat2', 'Common');
model.component('comp1').material('mat1').selection.set([1 2]);
model.component('comp1').material('mat1').propertyGroup('def').func.create('eta', 'Piecewise');
model.component('comp1').material('mat1').propertyGroup('def').func.create('Cp', 'Piecewise');
model.component('comp1').material('mat1').propertyGroup('def').func.create('rho', 'Analytic');
model.component('comp1').material('mat1').propertyGroup('def').func.create('k', 'Piecewise');
model.component('comp1').material('mat1').propertyGroup('def').func.create('cs', 'Analytic');
model.component('comp1').material('mat1').propertyGroup.create('RefractiveIndex', 'Refractive index');
model.component('comp1').material('mat2').selection.set([3]);
model.component('comp1').material('mat2').propertyGroup('def').func.create('k_solid_1', 'Piecewise');
model.component('comp1').material('mat2').propertyGroup('def').func.create('res_solid_1', 'Piecewise');
model.component('comp1').material('mat2').propertyGroup('def').func.create('alpha', 'Piecewise');
model.component('comp1').material('mat2').propertyGroup('def').func.create('C_solid_1', 'Piecewise');
model.component('comp1').material('mat2').propertyGroup('def').func.create('sigma_solid_1', 'Piecewise');
model.component('comp1').material('mat2').propertyGroup('def').func.create('HC_solid_1', 'Piecewise');
model.component('comp1').material('mat2').propertyGroup('def').func.create('VP_solid_1', 'Piecewise');
model.component('comp1').material('mat2').propertyGroup('def').func.create('rho', 'Piecewise');
model.component('comp1').material('mat2').propertyGroup.create('ThermalExpansion', 'Thermal expansion');
model.component('comp1').material('mat2').propertyGroup('ThermalExpansion').func.create('dL', 'Piecewise');
model.component('comp1').material('mat2').propertyGroup.create('Enu', 'Young''s modulus and Poisson''s ratio');
model.component('comp1').material('mat2').propertyGroup('Enu').func.create('E', 'Piecewise');
model.component('comp1').material('mat2').propertyGroup('Enu').func.create('nu', 'Piecewise');
model.component('comp1').material('mat2').propertyGroup.create('KG', 'Bulk modulus and shear modulus');
model.component('comp1').material('mat2').propertyGroup('KG').func.create('mu', 'Piecewise');
model.component('comp1').material('mat2').propertyGroup('KG').func.create('kappa', 'Piecewise');
model.component('comp1').material('mat2').propertyGroup.create('RefractiveIndex', 'Refractive index');
model.component('comp1').physics.create('ewfd', 'ElectromagneticWavesFrequencyDomain', 'geom1');
model.component('comp1').physics('ewfd').create('sctr1', 'Scattering', 2);
model.component('comp1').physics('ewfd').feature('sctr1').selection.set([1 2 3 4 5 7 8 9 27 28]);
model.component('comp1').physics('ewfd').create('port1', 'Port', 2);
model.component('comp1').physics('ewfd').feature('port1').selection.set([7]);
model.component('comp1').physics('ewfd').create('port2', 'Port', 2);
model.component('comp1').physics('ewfd').feature('port2').selection.set([3]);
model.component('comp1').mesh('mesh1').create('ftet1', 'FreeTet');
model.result.table('tbl1').comments('Surface Integration 1 (ewfd.normE)');
model.component('comp1').view('view1').set('renderwireframe', true);
model.component('comp1').view('view2').axis.set('xmin', -9.413238525390625);
model.component('comp1').view('view2').axis.set('xmax', 528.186767578125);
model.component('comp1').view('view2').axis.set('ymin', 0);
model.component('comp1').view('view2').axis.set('ymax', 588.8841552734375);
model.component('comp1').material('mat1').label('Air');
model.component('comp1').material('mat1').set('family', 'air');
model.component('comp1').material('mat1').propertyGroup('def').func('eta').set('arg', 'T');
model.component('comp1').material('mat1').propertyGroup('def').func('eta').set('pieces', {'200.0' '1600.0' '-8.38278E-7+8.35717342E-8*T^1-7.6942
model.component('comp1').material('mat1').propertyGroup('def').func('Cp').set('arg', 'T');
model.component('comp1').material('mat1').propertyGroup('def').func('Cp').set('pieces', {'200.0' '1600.0' '1047.63657-0.372589265*T^1+9.453042
model.component('comp1').material('mat1').propertyGroup('def').func('rho').set('expr', 'pA*0.02897/8.314/T');
model.component('comp1').material('mat1').propertyGroup('def').func('rho').set('args', {'pA' 'T'});
model.component('comp1').material('mat1').propertyGroup('def').func('rho').set('dermethod', 'manual');
model.component('comp1').material('mat1').propertyGroup('def').func('rho').set('argders', {'pA' 'd(pA*0.02897/8.314/T,pA)'; 'T' 'd(pA*0.02897/8.314/
model.component('comp1').material('mat1').propertyGroup('def').func('rho').set('plotargs', {'pA' '0' '1'; 'T' '0' '1'});
model.component('comp1').material('mat1').propertyGroup('def').func('k').set('arg', 'T');
model.component('comp1').material('mat1').propertyGroup('def').func('k').set('pieces', {'200.0' '1600.0' '-0.00227583562+1.15480022E-4*T^1-7.90
model.component('comp1').material('mat1').propertyGroup('def').func('cs').set('expr', 'sqrt(1.4*287*T)');
model.component('comp1').material('mat1').propertyGroup('def').func('cs').set('args', {'T'});
model.component('comp1').material('mat1').propertyGroup('def').func('cs').set('dermethod', 'manual');
model.component('comp1').material('mat1').propertyGroup('def').func('cs').set('argders', {'T' 'd(sqrt(1.4*287*T),T)'});
model.component('comp1').material('mat1').propertyGroup('def').func('cs').set('plotargs', {'T' '0' '1'});
model.component('comp1').material('mat1').propertyGroup('def').set('relpermeability', {'1' '0' '0' '0' '1' '0' '0' '0' '1'});
model.component('comp1').material('mat1').propertyGroup('def').set('relpermittivity', {'1' '0' '0' '0' '1' '0' '0' '0' '1'});
model.component('comp1').material('mat1').propertyGroup('def').set('dynamicviscosity', 'eta(T[1/K])[Pa*s]');
model.component('comp1').material('mat1').propertyGroup('def').set('ratioofspecificheat', '1.4');
model.component('comp1').material('mat1').propertyGroup('def').set('electricconductivity', {'0[S/m]' '0' '0' '0' '0[S/m]' '0' '0' '0' '0[S/m]'});
```

```
model.component('comp1').material('mat1').propertyGroup('def').set('heatcapacity', 'Cp(T[1/K])[J/(kg*K)]');
model.component('comp1').material('mat1').propertyGroup('def').set('density', 'rho(pA[1/Pa],T[1/K])[kg/m^3]');
model.component('comp1').material('mat1').propertyGroup('def').set('thermalconductivity', {'k(T[1/K])[W/(m*K)]' '0' '0' '0' 'k(T[1/K])[W/(m*K)]' '0' '0' '0'
model.component('comp1').material('mat1').propertyGroup('def').set('soundspeed', 'cs(T[1/K])[m/s]');
model.component('comp1').material('mat1').propertyGroup('def').addInput('temperature');
model.component('comp1').material('mat1').propertyGroup('def').addInput('pressure');
model.component('comp1').material('mat1').propertyGroup('RefractiveIndex').set('n', '');
model.component('comp1').material('mat1').propertyGroup('RefractiveIndex').set('ki', '');
model.component('comp1').material('mat1').propertyGroup('RefractiveIndex').set('n', {'1' '0' '0' '0' '1' '0' '0' '0' '1'});
model.component('comp1').material('mat1').propertyGroup('RefractiveIndex').set('ki', {'0' '0' '0' '0' '0' '0' '0' '0' '0'});
model.component('comp1').material('mat2').label('Gold [solid]');
model.component('comp1').material('mat2').set('family', 'gold');
model.component('comp1').material('mat2').propertyGroup('def').func('k_solid_1').set('arg', 'T');
model.component('comp1').material('mat2').propertyGroup('def').func('k_solid_1').set('pieces', {'0.0' '14.0' '496.8118*T^1+41.00857*T^2-10.46095*
'14.0' '45.0' '7977.526-554.7566*T^1+14.56751*T^2-0.1468484*T^3+3.231278E-4*T^4';  ...
'45.0' '85.0' '3568.586-199.4435*T^1+5.171042*T^2-0.06880706*T^3+4.612091E-4*T^4-1.23215E-6*T^5';  ...
'85.0' '1338.0' '330.6431-0.02536626*T^1-8.191375E-5*T^2+6.792908E-8*T^3-2.15362E-11*T^4'});
model.component('comp1').material('mat2').propertyGroup('def').func('res_solid_1').set('arg', 'T');
model.component('comp1').material('mat2').propertyGroup('def').func('res_solid_1').set('pieces', {'1.0' '25.0' '2.194815E-10+1.027514E-12*T^1-3.4
'25.0' '60.0' '1.390641E-9-1.228888E-10*T^1+4.182944E-12*T^2-2.773737E-14*T^3';  ...
'60.0' '400.0' '-2.210068E-9+9.057611E-11*T^1-4.632985E-14*T^2+6.950205E-17*T^3';  ...
'400.0' '1338.0' '-1.145028E-9+7.877041E-11*T^1-4.720065E-16*T^2+1.275961E-17*T^3'});
model.component('comp1').material('mat2').propertyGroup('def').func('alpha').set('arg', 'T');
model.component('comp1').material('mat2').propertyGroup('def').func('alpha').set('pieces', {'0.0' '74.0' '1.109296E-5+4.023822E-8*T^1-2.476496E-
model.component('comp1').material('mat2').propertyGroup('def').func('C_solid_1').set('arg', 'T');
model.component('comp1').material('mat2').propertyGroup('def').func('C_solid_1').set('pieces', {'293.0' '1338.0' '399352.2*T^-2+114.8987+0.0322
model.component('comp1').material('mat2').propertyGroup('def').func('sigma_solid_1').set('arg', 'T');
model.component('comp1').material('mat2').propertyGroup('def').func('sigma_solid_1').set('pieces', {'1.0' '25.0' '1/(3.010164E-14*T^3-3.409278E-1
'25.0' '60.0' '1/(-2.773737E-14*T^3+4.182944E-12*T^2-1.228888E-10*T+1.390641E-09)';  ...
'60.0' '400.0' '1/(6.950205E-17*T^3-4.632985E-14*T^2+9.057611E-11*T-2.210068E-09)';  ...
'400.0' '1338.0' '1/(1.275961E-17*T^3-4.720065E-16*T^2+7.877041E-11*T-1.145028E-09)'});
model.component('comp1').material('mat2').propertyGroup('def').func('HC_solid_1').set('arg', 'T');
model.component('comp1').material('mat2').propertyGroup('def').func('HC_solid_1').set('pieces', {'293.0' '1338.0' '78659.2*T^-2+22.63126+0.0063
model.component('comp1').material('mat2').propertyGroup('def').func('VP_solid_1').set('arg', 'T');
model.component('comp1').material('mat2').propertyGroup('def').func('VP_solid_1').set('pieces', {'293.0' '1337.0' '(exp((-1.934300e+04/T-7.47900(
model.component('comp1').material('mat2').propertyGroup('def').func('rho').set('arg', 'T');
model.component('comp1').material('mat2').propertyGroup('def').func('rho').set('pieces', {'0.0' '86.0' '19471.18+0.1076816*T^1-0.00817626*T^2+3.
model.component('comp1').material('mat2').propertyGroup('def').set('thermalconductivity', {'k_solid_1(T[1/K])[W/(m*K)]' '0' '0' '0' 'k_solid_1(T[1/K])[W
model.component('comp1').material('mat2').propertyGroup('def').set('resistivity', {'res_solid_1(T[1/K])[ohm*m]' '0' '0' '0' 'res_solid_1(T[1/K])[ohm*m]'
model.component('comp1').material('mat2').propertyGroup('def').set('thermalexpansioncoefficient', {'(alpha(T[1/K])[1/K]+(Tempref-293[K])*if(abs(T-T
model.component('comp1').material('mat2').propertyGroup('def').set('heatcapacity', 'C_solid_1(T[1/K])[J/(kg*K)]');
model.component('comp1').material('mat2').propertyGroup('def').set('electricconductivity', {'sigma_solid_1(T[1/K])[S/m]' '0' '0' '0' 'sigma_solid_1(T[1.
model.component('comp1').material('mat2').propertyGroup('def').set('HC', 'HC_solid_1(T[1/K])[J/(mol*K)]');
model.component('comp1').material('mat2').propertyGroup('def').set('VP', 'VP_solid_1(T[1/K])[Pa]');
model.component('comp1').material('mat2').propertyGroup('def').set('density', 'rho(T[1/K])[kg/m^3]');
model.component('comp1').material('mat2').propertyGroup('def').addInput('temperature');
model.component('comp1').material('mat2').propertyGroup('def').addInput('strainreferencetemperature');
model.component('comp1').material('mat2').propertyGroup('ThermalExpansion').func('dL').set('arg', 'T');
model.component('comp1').material('mat2').propertyGroup('ThermalExpansion').func('dL').set('pieces', {'0.0' '92.0' '-0.003245794-2.191545E-7*T^1
model.component('comp1').material('mat2').propertyGroup('ThermalExpansion').func('CTE').set('arg', 'T');
model.component('comp1').material('mat2').propertyGroup('ThermalExpansion').func('CTE').set('pieces', {'86.0' '200.0' '4.040319E-6+1.106436E-7
model.component('comp1').material('mat2').propertyGroup('ThermalExpansion').set('alphatan', '');
model.component('comp1').material('mat2').propertyGroup('ThermalExpansion').set('dL', '');
model.component('comp1').material('mat2').propertyGroup('ThermalExpansion').set('alphatanIso', '');
model.component('comp1').material('mat2').propertyGroup('ThermalExpansion').set('dLIso', '');
model.component('comp1').material('mat2').propertyGroup('ThermalExpansion').set('alphatan', {'CTE(T[1/K])[1/K]' '0' '0' '0' 'CTE(T[1/K])[1/K]' '0' '0' '0
model.component('comp1').material('mat2').propertyGroup('ThermalExpansion').set('dL', {'(dL(T[1/K])-dL(Tempref[1/K]))/(1+dL(Tempref[1/K]))' '0' '0'
model.component('comp1').material('mat2').propertyGroup('ThermalExpansion').set('alphatanIso', 'CTE(T)');
model.component('comp1').material('mat2').propertyGroup('ThermalExpansion').set('dLIso', '(dL(T)-dL(Tempref))/(1+dL(Tempref))');
model.component('comp1').material('mat2').propertyGroup('ThermalExpansion').addInput('temperature');
model.component('comp1').material('mat2').propertyGroup('ThermalExpansion').addInput('strainreferencetemperature');
model.component('comp1').material('mat2').propertyGroup('Enu').func('E').set('arg', 'T');
model.component('comp1').material('mat2').propertyGroup('Enu').func('E').set('pieces', {'0.0' '93.0' '8.190431E10-2.049635E7*T^1-8937.063*T^2';
model.component('comp1').material('mat2').propertyGroup('Enu').func('nu').set('arg', 'T');
model.component('comp1').material('mat2').propertyGroup('Enu').func('nu').set('pieces', {'293.0' '1280.0' '0.4403494+8.884688E-6*T^1-1.158876E-
model.component('comp1').material('mat2').propertyGroup('Enu').set('youngsmodulus', 'E(T[1/K])[Pa]');
model.component('comp1').material('mat2').propertyGroup('Enu').set('poissonsratio', 'nu(T[1/K])');
model.component('comp1').material('mat2').propertyGroup('Enu').addInput('temperature');
model.component('comp1').material('mat2').propertyGroup('KG').func('mu').set('arg', 'T');
model.component('comp1').material('mat2').propertyGroup('KG').func('mu').set('pieces', {'293.0' '1280.0' '2.682905E10+142041.4*T^1-7037.486*T
model.component('comp1').material('mat2').propertyGroup('KG').func('kappa').set('arg', 'T');
model.component('comp1').material('mat2').propertyGroup('KG').func('kappa').set('pieces', {'293.0' '1280.0' '2.224306E11-913189.7*T^1-37420.01
model.component('comp1').material('mat2').propertyGroup('KG').set('K', '');
model.component('comp1').material('mat2').propertyGroup('KG').set('G', '');
model.component('comp1').material('mat2').propertyGroup('KG').set('K', 'kappa(T[1/K])[Pa]');
model.component('comp1').material('mat2').propertyGroup('KG').set('G', 'mu(T[1/K])[Pa]');
model.component('comp1').material('mat2').propertyGroup('KG').addInput('temperature');
model.component('comp1').material('mat2').propertyGroup('RefractiveIndex').set('n', '');
model.component('comp1').material('mat2').propertyGroup('RefractiveIndex').set('ki', '');
model.component('comp1').material('mat2').propertyGroup('RefractiveIndex').set('n', {'345' '0' '0' '0' '345' '0' '0' '0' '345'});
model.component('comp1').material('mat2').propertyGroup('RefractiveIndex').set('ki', {'379' '0' '0' '0' '379' '0' '0' '0' '379'});
model.component('comp1').physics('ewfd').prop('EquationForm').set('freq', '1[GHz]');
model.component('comp1').physics('ewfd').prop('EquationForm').set('modeFreq', '1[GHz]');
model.component('comp1').physics('ewfd').prop('MeshControl').set('EnableMeshControl', false);
model.component('comp1').physics('ewfd').feature('port1').set('Pin', '1[mW]');
model.component('comp1').physics('ewfd').feature('port1').set('beta', 'abs(ewfd.k0)');
model.component('comp1').physics('ewfd').feature('port1').set('E0', {'0'; 'exp(-j*ewfd.k0*z)'; '0'});
```

```
model.component('comp1').physics('ewfd').feature('port2').set('beta', 'abs(ewfd.k0)');
model.component('comp1').physics('ewfd').feature('port2').set('E0', {'0'; 'exp(-j*ewfd.k0*z)'; '0'});
model.component('comp1').mesh('mesh1').feature('size').set('hauto', 4);
model.component('comp1').mesh('mesh1').run;
model.study.create('std1');
model.study('std1').create('freq', 'Frequency');
model.study('std1').feature('freq').set('activate', {'ewfd' 'on'});
model.sol.create('sol1');
model.sol('sol1').study('std1');
model.sol('sol1').attach('std1');
model.sol('sol1').create('st1', 'StudyStep');
model.sol('sol1').create('v1', 'Variables');
model.sol('sol1').create('s1', 'Stationary');
model.sol('sol1').feature('s1').create('p1', 'Parametric');
model.sol('sol1').feature('s1').create('fc1', 'FullyCoupled');
model.sol('sol1').feature('s1').create('i1', 'Iterative');
model.sol('sol1').feature('s1').feature('i1').create('mg1', 'Multigrid');
model.sol('sol1').feature('s1').feature('i1').feature('mg1').feature('pr').create('sv1', 'SORVector');
model.sol('sol1').feature('s1').feature('i1').feature('mg1').feature('po').create('sv1', 'SORVector');
model.sol('sol1').feature('s1').feature.remove('fcDef');
model.result.numerical.create('int1', 'IntSurface');
model.result.numerical('int1').selection.set([13]);
model.result.numerical('int1').set('probetag', 'none');
model.result.create('pg1', 'PlotGroup3D');
model.result.create('pg2', 'PlotGroup3D');
model.result('pg1').create('mslc1', 'Multislice');
model.result('pg1').feature('mslc1').set('data', 'dset1');
model.result('pg2').create('slc1', 'Slice');
model.result('pg2').feature('slc1').create('def1', 'Deform');
model.study('std1').feature('freq').set('punit', 'Hz');
model.study('std1').feature('freq').set('plist', '1e12');
model.study('std1').feature('freq').set('discretization', {'ewfd' 'physics'});
model.sol('sol1').attach('std1');
model.sol('sol1').feature('v1').set('clistctrl', {'p1'});
model.sol('sol1').feature('v1').set('cname', {'freq'});
model.sol('sol1').feature('v1').set('clist', {'1e12[Hz]'});
model.sol('sol1').feature('s1').feature('p1').set('pname', {'freq'});
model.sol('sol1').feature('s1').feature('p1').set('plistarr', {'1e12'});
model.sol('sol1').feature('s1').feature('p1').set('punit', {'Hz'});
model.sol('sol1').feature('s1').feature('p1').set('pcontinuationmode', 'no');
model.sol('sol1').feature('s1').feature('p1').set('preusesol', 'auto');
model.sol('sol1').feature('s1').feature('i1').label('Suggested Iterative Solver (ewfd)');
model.sol('sol1').feature('s1').feature('i1').set('linsolver', 'bicgstab');
model.sol('sol1').feature('s1').feature('i1').feature('mg1').feature('pr').feature('sv1').set('sorvecdof', {'comp1_E'});
model.sol('sol1').feature('s1').feature('i1').feature('mg1').feature('po').feature('sv1').set('sorvecdof', {'comp1_E'});
model.sol('sol1').runAll;
model.result.numerical('int1').set('table', 'tbl1');
model.result.numerical('int1').set('expr', {'ewfd.normE'});
model.result.numerical('int1').set('unit', {'V*m'});
model.result.numerical('int1').set('descr', {'Electric field norm'});
model.result.numerical('int1').setResult;
model.result('pg1').label('Electric Field (ewfd)');
model.result('pg1').set('frametype', 'spatial');
model.result('pg1').feature('mslc1').set('multiplanezmethod', 'coord');
model.result('pg1').feature('mslc1').set('zcoord', 1050);
model.result('pg1').feature('mslc1').set('rangecoloractive', true);
model.result('pg1').feature('mslc1').set('rangecolormin', 1.638211753582661);
model.result('pg1').feature('mslc1').set('rangecolormax', 1000000);
model.result('pg1').feature('mslc1').set('smooth', 'internal');
model.result('pg1').feature('mslc1').set('resolution', 'normal');
model.result('pg2').feature('slc1').set('quickplane', 'xy');
model.result('pg2').feature('slc1').set('quickzmethod', 'coord');
model.result('pg2').feature('slc1').set('quickz', 1050);
model.result('pg2').feature('slc1').set('smooth', 'internal');
model.result('pg2').feature('slc1').set('resolution', 'normal');
model.result('pg2').feature('slc1').feature('def1').active(false);
model.result('pg2').feature('slc1').feature('def1').set('expr', {'+ewfd.Ex' '+ewfd.Ey' '+ewfd.Ez'});
model.result('pg2').feature('slc1').feature('def1').set('descr', ' + Electric field');
model.result('pg2').feature('slc1').feature('def1').set('scale', 5.335142543557717E-4);
model.result('pg2').feature('slc1').feature('def1').set('scaleactive', false);
end
```

```
function [newM, newA] = Cruce(M,A,NG,NC)
  N = 9;
  for i = 1:NG/2
        for j = 1:NC-1
              binM = dec2bin(M(i,j),N);
              [binM] = dividirse(binM,N);
              newM(i,j) = bin2dec(binM);
        end
        newM(i,NC) = newM(i,1);
        newA(i,:) = A(i,:);
        newA(i,NC) = newA(i,1);
    end
  end
```

```
function [M,A,fitness] = Seleccion(Magnitud,Angulo,tbl_data,NG,NC)
    tbl = tbl_data;
    M = zeros(NG/2,NC);
    A = zeros(NG/2,NC);
    fitness = ones(1,NG/2);
    for j = 1:NG/2
        for i = 1:NG
            if( fitness(j) > tbl(i) )
                fitness(j) = tbl(i);
                indice = i;
            end
        end
        tbl(indice) = 1;
        M(j,:) = Magnitud(indice,:);
        A(j,:) = Angulo(indice,:);
    end
end
```

```
function [binM] = dividirse(binM,N)
    a = ((N+1)/2)-1:N;
    rng('shuffle');
    b=dec2bin(uint16(rand(1,1)*(2^(1+(N+1)/2))-1),N);
    binM(a)=b(a);
end
```

```
function [newM] = Mutar(newM,NG,NC,Radiomin,RadioMax)
RadioMax = RadioMax -2;
Radiomin = Radiomin +2;
for i = 1:NG/2
    for j = 1:NC-1
        rng('shuffle');
        al = rand(1,1)*1000;
        if al >= 500 && al <= 501
            R = RadioMax-(Radiomin+1);
            rng('shuffle');
            newM(i,j) =  (rand(1,1)*R)+Radiomin;
        end
    end
    newM(i,NC) = newM(i,1);
end
end
```

Anexo B código AGS para dos geometrías y material plata

```
function out = model
clear all;
format long;
import com.comsol.model.*
import com.comsol.model.util.*
%%%%%%%%%%%%%%%%PARAMETROS%%%%%%%%%%%%%%%%%%%%
% % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % %
NumGeo = 40;%Poblacion Inicial
NumCur = 17;%Numero de curvas en una geometrias
Grado = 3;% Grado de cada curva bezier
NC = (NumCur*Grado)+1;
TAMX = 2048;%Tamanio del universo de soluciones en X
TAMY = 1024;
Radiomin = 200; % Tamaño minimo de los vectores con respecto al centro
RadioMax = (TAMY/2); %Radio maximo de los vectores
%%%%%%%%%%%%%%Funcion que genera N geometrias 1%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Centro1 = [(TAMX/4),TAMY/2]; %Centro de la geometria 1
[X1, Y1,Magnitud1,Angulo1] = GenMulGeo1(NumGeo,NC,Grado,TAMX,TAMY,Radiomin,RadioMax,Centro1);
G1 = ones(1,NumCur).*Grado;
P1 = ones(1,NumCur*(Grado+1));
%%%%%%%%%%%%%%Funcion que genera N geometrias 2%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Centro2 = [(TAMX*3/4),TAMY/2];  %Centro de la geometria2
[X2, Y2,Magnitud2,Angulo2] = GenMulGeo2(NumGeo,NC,Grado,TAMX,TAMY,Radiomin,RadioMax,Centro2,X1, Y1);
G2 = ones(1,NumCur).*Grado;
P2 = ones(1,NumCur*(Grado+1));
rng('shuffle');
tbl_data = zeros(NumGeo,1);
fit = [];
band = 0;
%%%%%%%%%%%%%%%%%%Iteraciones (Numero de generaciones) %%%%%%%%%%%%%%5
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for j=1:40
   if band == 0
      numgeome = NumGeo;
   else
      numgeome = NumGeo/2;
   end
for i=1:numgeome%Numero de geometrias
%%%%%%%%%%%Codigo generado por COMSOL%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
model = ModelUtil.create('Model');
model.modelPath('D:\');
model.label('f_18009_2_2_Ramon - copia.mph');
model.comments(['F 18009 2 2\n\nF B\n\nF 30669 B5\n\nFisicaA\n\nExtruir1\n\nDos\n\nUntitled\n\n']);
model.component.create('comp1', true);
model.component('comp1').geom.create('geom1', 3);
model.result.table.create('tbl1', 'Table');
model.result.table.create('evl3', 'Table');
model.component('comp1').mesh.create('mesh1');
model.component('comp1').geom('geom1').lengthUnit('nm');
model.component('comp1').geom('geom1').create('wp1', 'WorkPlane');
model.component('comp1').geom('geom1').feature('wp1').set('quickz', 1024);
model.component('comp1').geom('geom1').feature('wp1').set('unite', true);
model.component('comp1').geom('geom1').feature('wp1').geom.create('b1', 'BezierPolygon');
%%%%% valores de X, Y para generar las curvas bezier de la geometria 1%%%%%%%%%%%%%%%%
model.component('comp1').geom('geom1').feature('wp1').geom.feature('b1').set('p', [X1(i,:);Y1(i,:)]);
model.component('comp1').geom('geom1').feature('wp1').geom.feature('b1').set('degree', G1);
model.component('comp1').geom('geom1').feature('wp1').geom.feature('b1').set('w', P1);
model.component('comp1').geom('geom1').feature('wp1').geom.feature('b1').set('type', 'open');
model.component('comp1').geom('geom1').feature('wp1').geom.create('csol1', 'ConvertToSolid');
```

```
63 -    model.component('comp1').geom('geom1').feature('wp1').geom.feature('csol1').selection('input').set({'b1'});
64 -    model.component('comp1').geom('geom1').feature('wp1').geom.create('b2', 'BezierPolygon');
65      %%%%% valores de X, Y para generar las curvas bezier de la geometria 1%%%%%%%%%%%%%%%%
66 -    model.component('comp1').geom('geom1').feature('wp1').geom.feature('b2').set('p', [X2(i,:);Y2(i,:)]);
67 -    model.component('comp1').geom('geom1').feature('wp1').geom.feature('b2').set('degree', G2);
68 -    model.component('comp1').geom('geom1').feature('wp1').geom.feature('b2').set('w', P2);
69 -    model.component('comp1').geom('geom1').feature('wp1').geom.feature('b2').set('type', 'open');
70 -    model.component('comp1').geom('geom1').feature('wp1').geom.create('csol2', 'ConvertToSolid');
71 -    model.component('comp1').geom('geom1').feature('wp1').geom.feature('csol2').selection('input').set({'b2'});
72 -    model.component('comp1').geom('geom1').create('ext1', 'Extrude');
73 -    model.component('comp1').geom('geom1').feature('ext1').setIndex('distance', '50', 0);
74 -    model.component('comp1').geom('geom1').feature('ext1').selection('input').set({'wp1'});
75 -    model.component('comp1').geom('geom1').create('blk1', 'Block');
76 -    model.component('comp1').geom('geom1').feature('blk1').set('pos', [-500 -500 0]);
77 -    model.component('comp1').geom('geom1').feature('blk1').set('size', [3048 2024 2048]);
78 -    model.component('comp1').geom('geom1').feature('blk1').set('layername', {'Layer 1'});
79 -    model.component('comp1').geom('geom1').feature('blk1').setIndex('layer', '1024', 0);
80 -    model.component('comp1').geom('geom1').run;
81 -    model.component('comp1').geom('geom1').run('fin');
82 -    model.component('comp1').view('view1').hideEntities.create('hide1');
83 -    model.component('comp1').view('view1').hideEntities('hide1').geom('geom1', 2);
84 -    model.component('comp1').view('view1').hideEntities('hide1').set([7]);
85      %%%%%%%%%%Funcion que abarca las caracteristicas que genera COMSOL para
86       %%%%%%%%%%material oro
87 -    fAgAg(model);
88      %%%%%%%%%%%%%%%%Funcion que regresa desde COMSOL la respuesta en una
89      %%%%%%%%%%%%%%%%tabla
90 -    str = mphtable(model,'tbl1');
91 -    tbl_data(i) = str.data(2);
92 -    format SHORT
93 -    letras = 'D:/2fisicas1Objetivo/f_';
94      %%%%%%%%%%%%%%%%%%%%Guarda el modelo anteriormente generado en formato
95      %%%%%%%%%%%%%%%%%%%%mph e COMSOL
96 -    mphsave(model,strcat(letras,num2str(uint16(tbl_data(i)*1e12)), ...
97      '_In',num2str(i),'_Ge',num2str(j)));
98 -    format LONG
99 -    geometria = i
100 -   end
101 -   max = [tbl_data ; fit]
102 -   tbl_data = [];
103 -   if band == 1
104 -      Magnitud1 = [newM1;M1];
105 -      Angulo1 = [newA1;A1];
106 -      Magnitud2 = [newM2;M2];
107 -      Angulo2 = [newA2;A2];
108 -   else
109 -      band = 1;
110 -   end
111     % % % % % % % % % Operaciones del algoritmo Genetico SImple %%%%%%
112 -   [M1,A1,M2,A2,fit,NMat] = Seleccion2(Magnitud1,Angulo1,Magnitud2,Angulo2,max,NumGeo,NC,[]);
113 -   [newM1,newA1,newM2,newA2,NMateriales] = Cruce2(M1,A1,M2,A2,NumGeo,NC,[]);
114 -   [newM1,newM2] = Mutar2(newM1,newM2,NumGeo,NC,Radiomin,RadioMax);
115     %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
116 -   for l = 1:NumGeo/2
117 -      for k = 5:NC-1
118 -         X1(l,k) = double(uint16((newM1(l,k)*cos(newA1(l,k)))+Centro1(1)));
119 -         Y1(l,k) = double(uint16((newM1(l,k)*sin(newA1(l,k)))+Centro1(2)));
120 -         X2(l,k) = double(uint16((newM2(l,k)*cos(newA2(l,k)))+Centro2(1)));
121 -         Y2(l,k) = double(uint16((newM2(l,k)*sin(newA2(l,k)))+Centro2(2)));
122 -      end
123 -   end
124 -   iteracion = j
125 -   end
126 -   out = model;
```

```
1       function [PuntosX, PuntosY,Radio,Angulo] = GenMulGeo1(NG,NC,Grado,TAMX,TAMY,Radiomin,RadioMax,Centro)
2 -     AnguloMax = 2*pi;
3 -     Angulomin = (360/NC)-1;
4 -     radgrad = 2*pi/360;
5 -     phase = 0;
6       %%%%%%%%%%%Asigna el valor a cada angulo%%%%%%%%%%%%%%%%%%
7 -     for i = 1:NC
8 -        for j = 1:NG
9 -           Angulo(j,i) = (Angulomin*i*radgrad) + phase;
10 -       end
11 -    end
```

```
%%%%%%%%%Construye las magnitudes de los vectores%%%%%%%%%%%%%%%
   Radio = ones(NG,NC);
   rng('shuffle');
   R = RadioMax;
   Radio(:,1) = (rand(NG,1).*R);
   band = 1;
   i = 2;
   while i < NC
      if band == 1 %%Mayor
         rng('shuffle');
         R = RadioMax-Radio(:,i-1);
         Radio(:,i) = (rand(NG,1).*R)+Radio(:,i-1);
         i=i+1;
         rng('shuffle');
         R = RadioMax-Radio(:,i-1);
         Radio(:,i) = (rand(NG,1).*R)+Radio(:,i-1);
         band = 0;
      else
         rng('shuffle');
         R = Radio(:,i-1)-Radiomin;
         Radio(:,i) = (rand(NG,1).*R)+Radiomin;
         band = 1;
      end
      i=i+1;
   end
Radio = double(uint16(Radio));
PuntosX = double(uint16((Radio.*cos(Angulo))+Centro(1)));
PuntosY = double(uint16((Radio.*sin(Angulo))+Centro(2)));
% % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % %
% % % % % Codigo que genera la union de la geometria 1%%%%%%%%%%%%
for j = 1:NG
   [indice] = mayor(PuntosX(j,:),NC);
   moverX = 1024-PuntosX(j,indice);
   for i = 1:NC
      PuntosX(j,i) = PuntosX(j,i)+moverX;
   end
end

for j = 1:NG
   for i = 1:4
      PuntosX(j,i) = 1024;
   end
end
tamUnion = 34;
PuntosY(:,2) = PuntosY(:,1)+tamUnion*1;
PuntosY(:,3) = PuntosY(:,1)+tamUnion*2;
PuntosY(:,4) = PuntosY(:,1)+tamUnion*3;
PuntosX(:,NC) = PuntosX(:,1);
PuntosY(:,NC) = PuntosY(:,1);
% % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % %
%%%%%%%%%%%%%%%%%%%%%%%%%Ajuste a los puntos X,Y para no sobrepasar el limite
%%%%%%%%%%%%%%%%%%%%%%%%%de tamaño
for i = 1:NC
   for j = 1:NG
      if PuntosX(j,i) < 1
         PuntosX(j,i) = 1;
      end
      if PuntosX(j,i) >= 1024
         PuntosX(j,i) = 1024;
      end
      if PuntosY(j,i) >= 1024
         PuntosY(j,i) = 1023;
      end
      if PuntosY(j,i) < 1
         PuntosY(j,i) = 1;
      end
   end
end
end
```

```
function [PuntosX, PuntosY,Radio,Angulo] = GenMulGeo2(NG,NC,Grado,TAMX,TAMY,Radiomin,RadioMax, Centro,PX1, PY1)
AnguloMax = 2*pi;
Angulomin = (360/NC)-1;
radgrad = 2*pi/360;
phase = Angulomin*radgrad*(NC-1)/2;
%%%%%%%%%%%%Asigna el valor a cada angulo%%%%%%%%%%%%%%%%%%
 Angulomin = (360/NC);
for i = 1:NC
   for j = 1:NG
      Angulo(j,i) = (Angulomin*i*radgrad) + phase;
   end
end
```

```
%%%%%%%%%Construye las magnitudes de los vectores%%%%%%%%%%%%%%%
Radio = ones(NG,NC).*(rand(NG,1)*(RadioMax-Radiomin))+Radiomin;  %% EL mismo radio para todos los puntos
Radio = double(uint16(Radio));
PuntosX = uint16((Radio.*cos(Angulo))+Centro(1));
PuntosY = uint16((Radio.*sin(Angulo))+Centro(2));
PuntosX = double(PuntosX);
PuntosY = double(PuntosY);
% % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % %
% % % % % Codigo que genera la union de la geometria 1%%%%%%%%%%%%
for j = 1:NG
    [indice] = menor(PuntosX(j,:),NC);
    for i = 1:4
        PuntosX(j,i) = PuntosX(j,indice)-10;
    end
end
PuntosX(:,NC) = PuntosX(:,1);
PuntosY(:,NC) = PuntosY(:,1);
for j = 1:NG
    indice = 1;
    indice2 = 1;
    distanciaX = PuntosX(j,indice)- PX1(j,indice2);
    PuntosX(j,:) = PuntosX(j,:) - abs(distanciaX);
end
for i = 1:4
    PuntosY(:,5-i) = PY1(:,i);
    PuntosX(:,5-i) = PX1(:,i);
end
PuntosX(:,NC) = PuntosX(:,1);
PuntosY(:,NC) = PuntosY(:,1);
% % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % %
%%%%%%%%%%Ajuste a los puntos X,Y para no sobrepasar el limite de tamaño%%%%
for i = 1:NC
    for j = 1:NG
        if PuntosX(j,i) > 2047
            PuntosX(j,i) = 2047;
        end
        if PuntosY(j,i) >= 1024
            PuntosY(j,i) = 1024;
        end
        if PuntosY(j,i) < 1
            PuntosY(j,i) = 1;
        end
    end
end
end
```

```
function [M1,A1,M2,A2,fitness,NMat] = Seleccion2(Magnitud1,Angulo1,Magnitud2,Angulo2,tbl_data,NG,NC,Materiales)
    tbl = tbl_data;
    M1 = zeros(NG/2,NC);
    A1 = zeros(NG/2,NC);
    M2 = zeros(NG/2,NC);
    A2 = zeros(NG/2,NC);
    NMat = zeros(NG/2);
    fitness = ones(1,NG/2);
    for j = 1:NG/2
        for i = 1:NG
            if( fitness(j) > tbl(i) )
                fitness(j) = tbl(i);
                indice = i;
            end
        end
        tbl(indice) = 1;
        M1(j,:) = Magnitud1(indice,:);
        A1(j,:) = Angulo1(indice,:);
        M2(j,:) = Magnitud2(indice,:);
        A2(j,:) = Angulo2(indice,:);
        if isempty(Materiales) == 0
            NMat(j) = Materiales(indice);
        end
    end
    if isempty(Materiales) == 1
      NMat = [];
    end
end
```

```
function [newM1, newA1,newM2, newA2,NMateriales] = Cruce2(M1,A1,M2,A2,NG,NC,NMat)
N = 9;
for i = 1:NG/2
    for j = 5:NC-1
        binM1 = dec2bin(M1(i,j),N);
        [binM1] = dividirse(binM1,N);
        newM1(i,j) = bin2dec(binM1);
    end
    newM1(i,NC) = newM1(i,1);
    newA1(i,:) = A1(i,:);
    newA1(i,NC) = newA1(i,1);
end
for i = 1:NG/2
    for j = 5:NC-1
        binM2 = dec2bin(M2(i,j),N);
        [binM2] = dividirse(binM2,N);
        newM2(i,j) = bin2dec(binM2);
    end
    newM2(i,NC) = newM2(i,1);
    newA2(i,:) = A2(i,:);
    newA2(i,NC) = newA2(i,1);
end
sum = int16(rand(NG/2,1)*2.5)-1;
if isempty(NMat) == 0
    for i = 1:NG/2
        NMateriales(i) = NMat(i) + sum(i);
        if NMateriales(i) < 1
            NMateriales(i) = NMateriales(i) + 1;
        end
        if NMateriales(i) > 9
            NMateriales(i) = NMateriales(i) - 1;
        end
    end
end
if isempty(NMat) == 1
    NMateriales = [];
end
end
```

```
function [newM1,newM2] = Mutar2(newM1,newM2,NG,NC,Radiomin,RadioMax)
RadioMax = RadioMax - 2;
Radiomin = Radiomin + 2;
for i = 1:NG/2
    for j = 5:NC-1
        rng('shuffle');
        al = rand(1,1)*1000;
        if al > 500 && al < 501
            R = RadioMax-(Radiomin+1);
            rng('shuffle');
            newM1(i,j) =  (rand(1,1)*R)+Radiomin;
            rng('shuffle');
            newM2(i,j) =  (rand(1,1)*R)+Radiomin;
        end
    end
    newM1(i,NC) = newM1(i,1);
    newM2(i,NC) = newM2(i,1);
end
end
```

Anexo C código Algoritmo Genético Multiobjetivo para dos geometrías y dos materiales.

```
function out = AGMPareto
clear all;
format long;
import com.comsol.model.*
import com.comsol.model.util.*
%%%%%%%%%%%%%%%%PARAMETROS%%%%%%%%%%%%%%%%%%%%
% % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % %
NumGeo = 100;%Poblacion Inicial
NumCur = 15;%Numero de curvas en una geometrias
Grado = 3;% Grado de cada curva bezier
NC = (NumCur*Grado)+1;
TAMX = 2048;%Tamanio del universo de soluciones en X
TAMY = 1024;
Radiomin = 200;% Tamaño minimo de los vectores con respecto al centro
RadioMax = (TAMY/2);%Radio maximo de los vectores
%%%%%%%%%%%%%Funcion que genera N geometrias 1%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Centro1 = [(TAMX/4),TAMY/2];%Centro de la geometria 1
[X1, Y1,Magnitud1,Angulo1] = GenMulGeo1(NumGeo,NC,Grado,TAMX,TAMY,Radiomin,RadioMax,Centro1);
G1 = ones(1,NumCur).*Grado;
P1 = ones(1,NumCur*(Grado+1));
%%%%%%%%%%%%%Funcion que genera N geometrias 2%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Centro2 = [(TAMX*3/4),TAMY/2];%Centro de la geometria2
[X2, Y2,Magnitud2,Angulo2] = GenMulGeo2(NumGeo,NC,Grado,TAMX,TAMY,Radiomin,RadioMax,Centro2,X1, Y1);
G2 = ones(1,NumCur).*Grado;
P2 = ones(1,NumCur*(Grado+1));
rng('shuffle');
Materiales = int16(rand(NumGeo,1)*8.5)+1;
tbl_data = zeros(NumGeo,1);
tbl_data2 = zeros(NumGeo,1);
fit = [];
band = 0;
%%%%%%%%%%%%%%%%%Iteraciones (Numero de generaciones) %%%%%%%%%%%%%5
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for j=1:40
    if band == 0
        numgeome = NumGeo;
    else
        numgeome = NumGeo/2;
    end
    tbl_data = [];
    tbl_data2 = [];
for i=1:numgeome%Numero de geometrias de cada generacion
%%%%%%%%%%%%Codigo generado por COMSOL%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
model = ModelUtil.create('Model');
model.modelPath('G:\MIE 2018\Cuarto semestre\Nueva carpeta');
model.label('f_BiAg.mph');
model.comments('F 18009 2 2\n\nF B\n\nF 30669 B5\n\nFisicaA\n\nExtruir1\n\nDos\n\nUntitled\n\n');
model.component.create('comp1', true);
model.component('comp1').geom.create('geom1', 3);
model.result.table.create('tbl1', 'Table');
model.result.table.create('evl3', 'Table');
model.result.table.create('tbl2', 'Table');
model.component('comp1').mesh.create('mesh1');
model.component('comp1').geom('geom1').lengthUnit('nm');
model.component('comp1').geom('geom1').create('wp1', 'WorkPlane');
model.component('comp1').geom('geom1').feature('wp1').set('quickz', 1024);
model.component('comp1').geom('geom1').feature('wp1').set('unite', true);
model.component('comp1').geom('geom1').feature('wp1').geom.create('b1', 'BezierPolygon');
%%%%% valores de X, Y para generar las curvas bezier de la geometria 1%%%%%%%%%%%%%%%
model.component('comp1').geom('geom1').feature('wp1').geom.feature('b1').set('p', [X1(i,:);Y1(i,:)]);
model.component('comp1').geom('geom1').feature('wp1').geom.feature('b1').set('degree', G1);
model.component('comp1').geom('geom1').feature('wp1').geom.feature('b1').set('w', P1);
model.component('comp1').geom('geom1').feature('wp1').geom.feature('b1').set('type', 'open');
model.component('comp1').geom('geom1').feature('wp1').geom.create('csol1', 'ConvertToSolid');
model.component('comp1').geom('geom1').feature('wp1').geom.feature('csol1').selection('input').set({'b1'});
model.component('comp1').geom('geom1').feature('wp1').geom.create('b2', 'BezierPolygon');
%%%%% valores de X, Y para generar las curvas bezier de la geometria 2%%%%%%%%%%%%%%%
model.component('comp1').geom('geom1').feature('wp1').geom.feature('b2').set('p', [X2(i,:);Y2(i,:)]);
model.component('comp1').geom('geom1').feature('wp1').geom.feature('b2').set('degree', G2);
model.component('comp1').geom('geom1').feature('wp1').geom.feature('b2').set('w', P2);
model.component('comp1').geom('geom1').feature('wp1').geom.feature('b2').set('type', 'open');
model.component('comp1').geom('geom1').feature('wp1').geom.create('csol2', 'ConvertToSolid');
model.component('comp1').geom('geom1').feature('wp1').geom.feature('csol2').selection('input').set({'b2'});
model.component('comp1').geom('geom1').create('ext1', 'Extrude');
model.component('comp1').geom('geom1').feature('ext1').setIndex('distance', '50', 0);
model.component('comp1').geom('geom1').feature('ext1').selection('input').set({'wp1'});
model.component('comp1').geom('geom1').create('blk1', 'Block');
model.component('comp1').geom('geom1').feature('blk1').set('pos', [-500 -500 0]);
model.component('comp1').geom('geom1').feature('blk1').set('size', [3048 2024 2048]);
```

```
    model.component('comp1').geom('geom1').feature('blk1').set('layername', {'Layer 1'});
    model.component('comp1').geom('geom1').feature('blk1').setIndex('layer', '1024', 0);
    model.component('comp1').geom('geom1').run;
    model.component('comp1').geom('geom1').run('fin');
    model.component('comp1').view('view1').hideEntities.create('hide1');
    model.component('comp1').view('view1').hideEntities('hide1').set([2]);
    %%%%%%%%%%%%%%%Seleccion de combinacion de materiales%%%%%%%%%%%%%
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    if Materiales(i) == 1
        f_BiAg(model); %%%Bizmuto Plata
    elseif Materiales(i) == 2
        f_BiTa(model)   %%%Bizmuto Tantalio
    elseif Materiales(i) == 3
        f_BiW(model);  %%%Bizmuto Tungsteno
    elseif Materiales(i) == 4
        f_NiTa(model); %%%Niquel Tantalio
    elseif Materiales(i) == 5
        f_NiAg(model);%%%Niquel Plata
    elseif Materiales(i) == 6
        f_KAg(model);%%%Potasio Plata
    elseif Materiales(i) == 7
        f_KTa(model);%%%Potasio Tantalio
    elseif Materiales(i) == 8
        f_KW(model);%%%Potasio Tungsteno
    else %if Materiales(i) == 9
        f_NiW(model);%%% Niquel Potasio
    end
    %%%%%%%%%%%%%%%Funcion que regresa desde COMSOL las respuesta en dos
    %%%%%%%%%%%%%%%%tablas, tbl1 campo electrico nomalizado y tbl2 densidad de
    %%%%%%%%%%%%%%%disipacion de potencia total
    str = mphtable(model,'tbl1');
    str2 = mphtable(model,'tbl2');
    tbl_data(i) = str.data(2);
    tbl_data2(i) = str2.data(2);
    format SHORT
    letras = 'G:/MIE 2018/Cuarto semestre/ULTIMO/f_';
    %%%%%%%%%%%%%%%%%%%Guarda el modelo anteriormente generado en formato
    %%%%%%%%%%%%%%%%%%mph e COMSOL
    if j > 1
        mphsave(model,strcat(letras,'_E',num2str(uint16(tbl_data(i)*1e11)), ...
        '_P',num2str(uint16(tbl_data2(i)*1e5)),'_G',num2str(j)));
    end
    format LONG
    end
    frentepareto =[];
    fp = [];
    %%%%%%%%%%%%%Optimalidad de Pareto%%%%%%%%%%%%%%
    maxf1 = max(tbl_data);
    maxf2 = max(tbl_data2);
    tbl_data = tbl_data/maxf1;
    tbl_data2 = tbl_data2/maxf2;
    minf1 = min(tbl_data);
    minf2 = min(tbl_data2);
    %%%Calcula la distancia entre un indivduo optimo y cada uno de los
    %%%individuos del algoritmo
    frentepareto = sqrt((tbl_data-minf1).^2) + ((tbl_data2-minf2).^2));    %
    %save(strcat(letras,'Tablas.mat'),'tbl_data','tbl_data2');
    minimizar = [frentepareto ; fit]
    if band == 1
        Magnitud1 = [newM1;M1];
        Angulo1 = [newA1;A1];
        Magnitud2 = [newM2;M2];
        Angulo2 = [newA2;A2];
        Materiales = [NMateriales;NMat];
    else
        band = 1;
    end
    % % % % % % % % % Operaciones del algoritmo Genetico Multiobjetivo %%%%%%
    [M1,A1,M2,A2,fit,NMat] = Seleccion2(Magnitud1,Angulo1,Magnitud2,Angulo2,minimizar,NumGeo,NC,Materiales);
    [newM1,newA1,newM2,newA2,NMateriales] = Cruce2(M1,A1,M2,A2,NumGeo,NC,NMat);
    [newM1,newM2] = Mutar2(newM1,newM2,NumGeo,NC,Radiomin,RadioMax);
    for l = 1:NumGeo/2
        for k = 5:NC-1
            X1(l,k) = double(uint16((newM1(l,k)*cos(newA1(l,k)))+Centro1(1)));
            Y1(l,k) = double(uint16((newM1(l,k)*sin(newA1(l,k)))+Centro1(2)));
            X2(l,k) = double(uint16((newM2(l,k)*cos(newA2(l,k)))+Centro2(1)));
            Y2(l,k) = double(uint16((newM2(l,k)*sin(newA2(l,k)))+Centro2(2)));
        end
    end
    end
    out = model;
```

```
function [] = f_KW(model)
model.component('comp1').material.create('mat1', 'Common');
model.component('comp1').material.create('mat5', 'Common');
model.component('comp1').material.create('mat7', 'Common');
model.component('comp1').material('mat1').propertyGroup('def').func.create('eta', 'Piecewise');
model.component('comp1').material('mat1').propertyGroup('def').func.create('Cp', 'Piecewise');
model.component('comp1').material('mat1').propertyGroup('def').func.create('rho', 'Analytic');
model.component('comp1').material('mat1').propertyGroup('def').func.create('k', 'Piecewise');
model.component('comp1').material('mat1').propertyGroup('def').func.create('cs', 'Analytic');
model.component('comp1').material('mat1').propertyGroup.create('RefractiveIndex', 'Refractive index');
model.component('comp1').material('mat5').selection.set([3]);
model.component('comp1').material('mat5').propertyGroup.create('RefractiveIndex', 'Refractive index');
model.component('comp1').material('mat5').propertyGroup('RefractiveIndex').func.create('int1', 'Interpolation');
model.component('comp1').material('mat5').propertyGroup('RefractiveIndex').func.create('int2', 'Interpolation');
model.component('comp1').material('mat7').selection.set([4]);
model.component('comp1').material('mat7').propertyGroup('def').func.create('k_solid_1', 'Piecewise');
model.component('comp1').material('mat7').propertyGroup('def').func.create('res_solid_1', 'Piecewise');
model.component('comp1').material('mat7').propertyGroup('def').func.create('Syfunc_solid_tested_at_195K_8', 'Piecewise');
model.component('comp1').material('mat7').propertyGroup('def').func.create('Syt', 'Piecewise');
model.component('comp1').material('mat7').propertyGroup('def').func.create('C_solid_1', 'Piecewise');
model.component('comp1').material('mat7').propertyGroup('def').func.create('sigma_solid_1', 'Piecewise');
model.component('comp1').material('mat7').propertyGroup('def').func.create('HC_solid_1', 'Piecewise');
model.component('comp1').material('mat7').propertyGroup('def').func.create('VP_solid_1', 'Piecewise');
model.component('comp1').material('mat7').propertyGroup('def').func.create('elong', 'Piecewise');
model.component('comp1').material('mat7').propertyGroup.create('Enu', 'Young''s modulus and Poisson''s ratio');
model.component('comp1').material('mat7').propertyGroup('Enu').func.create('E', 'Piecewise');
model.component('comp1').material('mat7').propertyGroup('Enu').func.create('nu', 'Piecewise');
model.component('comp1').material('mat7').propertyGroup.create('KG', 'Bulk modulus and shear modulus');
model.component('comp1').material('mat7').propertyGroup('KG').func.create('mu', 'Piecewise');
model.component('comp1').material('mat7').propertyGroup('KG').func.create('kappa_solid_1', 'Piecewise');
model.component('comp1').material('mat7').propertyGroup.create('ElastoplasticModel', 'Elastoplastic material model');
model.component('comp1').material('mat7').propertyGroup('ElastoplasticModel').func.create('Sys', 'Piecewise');
model.component('comp1').material('mat7').propertyGroup.create('RefractiveIndex', 'Refractive index');
model.component('comp1').physics.create('ewfd', 'ElectromagneticWavesFrequencyDomain', 'geom1');
model.component('comp1').physics('ewfd').create('sctr1', 'Scattering', 2);
model.component('comp1').physics('ewfd').feature('sctr1').selection.set([1 2 3 4 5 7 8 9 43 44]);
model.component('comp1').physics('ewfd').create('port1', 'Port', 2);
model.component('comp1').physics('ewfd').feature('port1').selection.set([7]);
model.component('comp1').physics('ewfd').create('port2', 'Port', 2);
model.component('comp1').physics('ewfd').feature('port2').selection.set([3]);
model.component('comp1').mesh('mesh1').create('ftet1', 'FreeTet');
model.result.table('tbl1').comments('Surface Integration 1 (ewfd.normE)');
model.result.table('evl3').label('Evaluation 3D');
model.result.table('evl3').comments('Interactive 3D values');
model.result.table('tbl2').comments('Surface Integration 2 (ewfd.Qh)');
model.component('comp1').view('view1').set('renderwireframe', true);
model.component('comp1').view('view2').axis.set('xmin', -37.6529541015625);
model.component('comp1').view('view2').axis.set('xmax', 2112.7470703125);
model.component('comp1').view('view2').axis.set('ymin', 0);
model.component('comp1').view('view2').axis.set('ymax', 1651.199951171875);
model.component('comp1').material('mat1').label('Air');
model.component('comp1').material('mat1').set('family', 'air');
model.component('comp1').material('mat1').propertyGroup('def').func('eta').set('arg', 'T');
model.component('comp1').material('mat1').propertyGroup('def').func('eta').set('pieces', {'200.0' '1600.0' '-8.38278E-7+8.35717342E-8*T^1-7.6942
model.component('comp1').material('mat1').propertyGroup('def').func('Cp').set('arg', 'T');
model.component('comp1').material('mat1').propertyGroup('def').func('Cp').set('pieces', {'200.0' '1600.0' '1047.63657-0.372589265*T^1+9.453042
model.component('comp1').material('mat1').propertyGroup('def').func('rho').set('expr', 'pA*0.02897/8.314/T');
model.component('comp1').material('mat1').propertyGroup('def').func('rho').set('args', {'pA' 'T'});
model.component('comp1').material('mat1').propertyGroup('def').func('rho').set('dermethod', 'manual');
model.component('comp1').material('mat1').propertyGroup('def').func('rho').set('argders', {'pA' 'd(pA*0.02897/8.314/T,pA)'; 'T' 'd(pA*0.02897/8.314/
model.component('comp1').material('mat1').propertyGroup('def').func('rho').set('plotargs', {'pA' '0' '1'; 'T' '0' '1'});
model.component('comp1').material('mat1').propertyGroup('def').func('k').set('arg', 'T');
model.component('comp1').material('mat1').propertyGroup('def').func('k').set('pieces', {'200.0' '1600.0' '-0.00227583562+1.15480022E-4*T^1-7.90
model.component('comp1').material('mat1').propertyGroup('def').func('cs').set('expr', 'sqrt(1.4*287*T)');
model.component('comp1').material('mat1').propertyGroup('def').func('cs').set('args', {'T'});
model.component('comp1').material('mat1').propertyGroup('def').func('cs').set('dermethod', 'manual');
model.component('comp1').material('mat1').propertyGroup('def').func('cs').set('argders', {'T' 'd(sqrt(1.4*287*T),T)'});
model.component('comp1').material('mat1').propertyGroup('def').func('cs').set('plotargs', {'T' '0' '1'});
model.component('comp1').material('mat1').propertyGroup('def').set('relpermeability', {'1' '0' '0' '0' '1' '0' '0' '0' '1'});
model.component('comp1').material('mat1').propertyGroup('def').set('relpermittivity', {'1' '0' '0' '0' '1' '0' '0' '0' '1'});
model.component('comp1').material('mat1').propertyGroup('def').set('dynamicviscosity', 'eta(T[1/K])[Pa*s]');
model.component('comp1').material('mat1').propertyGroup('def').set('ratioofspecificheat', '1.4');
model.component('comp1').material('mat1').propertyGroup('def').set('electricconductivity', {'0[S/m]' '0' '0' '0' '0[S/m]' '0' '0' '0' '0[S/m]'});
model.component('comp1').material('mat1').propertyGroup('def').set('heatcapacity', 'Cp(T[1/K])[J/(kg*K)]');
model.component('comp1').material('mat1').propertyGroup('def').set('density', 'rho(pA[1/Pa],T[1/K])[kg/m^3]');
model.component('comp1').material('mat1').propertyGroup('def').set('thermalconductivity', {'k(T[1/K])[W/(m*K)]' '0' '0' '0' 'k(T[1/K])[W/(m*K)]' '0' '0' '0'
model.component('comp1').material('mat1').propertyGroup('def').set('soundspeed', 'cs(T[1/K])[m/s]');
model.component('comp1').material('mat1').propertyGroup('def').addInput('temperature');
model.component('comp1').material('mat1').propertyGroup('def').addInput('pressure');
model.component('comp1').material('mat1').propertyGroup('RefractiveIndex').set('n', '');
model.component('comp1').material('mat1').propertyGroup('RefractiveIndex').set('ki', '');
model.component('comp1').material('mat1').propertyGroup('RefractiveIndex').set('n', {'1' '0' '0' '0' '1' '0' '0' '0' '1'});
model.component('comp1').material('mat1').propertyGroup('RefractiveIndex').set('ki', {'0' '0' '0' '0' '0' '0' '0' '0' '0'});
model.component('comp1').material('mat5').label(['W (Tungsten) (Ordal et al. 1988: n,k 0.667-200 ' native2unicode(hex2dec({'00' 'b5'}), 'unicode') 'm
model.component('comp1').material('mat5').set('groups', {'inorganic' 'Inorganic Materials'; 'w_tungsten_and_tungstates' 'W - Tungsten and tungstates
model.component('comp1').material('mat5').propertyGroup('RefractiveIndex').func('int1').label('Refractive index');
```

```
model.component('comp1').material('mat5').propertyGroup('RefractiveIndex').func('int1').set('funcname', 'n_interp');
model.component('comp1').material('mat5').propertyGroup('RefractiveIndex').func('int1').set('table', {'6.67E-7' '3.8312601'; ...
'7.14E-7' '3.9313491';'7.69E-7' '3.778336';'8.3299999999999999E-7' '3.5190631';'9.09E-7' '3.2814572';'1.0E-6' '3.0826871'; ...
'1.05E-6' '3.0570934';'1.11E-6' '3.0821649';'1.18E-6' '3.1599449';'1.2499999999999999E-6' '3.1960757'; ...
'1.33E-6' '3.1990778';'1.4299999999999999E-6' '3.0278249';'1.5399999999999999E-6' '2.2587782'; ...
'1.6699999999999999E-6' '1.8828191';'1.82E-6' '1.5455798';'2.0E-6' '1.2992808';'2.1099999999999997E-6' '1.1946455'; ...
'2.22E-6' '1.221203';'2.35E-6' '1.2886548';'2.4999999999999998E-6' '1.4507921';'2.67E-6' '1.6746349'; ...
'2.8599999999999997E-6' '1.8748013';'3.0799999999999997E-6' '1.91692';'3.33E-6' '1.8505739';'3.64E-6' '1.8774806'; ...
'4.0E-6' '1.9949739';'4.44E-6' '2.2388765';'4.9999999999999996E-6' '2.6672954';'5.7099999999999995E-6' '3.3712766'; ...
'6.67E-6' '4.4961625';'8.0E-6' '6.2868052';'9.9999999999999999E-6' '9.1935519';'1.1099999999999999E-5' '10.843912'; ...
'1.2499999999999999E-5' '12.935818';'1.43E-5' '15.634532';'1.67E-5' '19.231714';'1.9999999999999998E-5' '24.256111'; ...
'2.2199999999999998E-5' '27.599188';'2.4999999999999998E-5' '31.77255';'2.86E-5' '37.127733'; ...
'3.3299999999999996E-5' '44.209069'; '3.9999999999999996E-5' '54.191208';'4.4399999999999995E-5' '61.731428'; ...
'4.9999999999999996E-5' '68.665703';'5.71E-5' '76.505227';'6.67E-5' '87.148396';'7.999999999999999E-5' '103.53408'; ...
'9.999999999999999E-5' '130.219';'1.25E-4' '163.02849'; '1.54E-4' '196.84283'; '1.9999999999999998E-4' '242.14161'});
model.component('comp1').material('mat5').propertyGroup('RefractiveIndex').func('int1').set('interp', 'piecewisecubic');
model.component('comp1').material('mat5').propertyGroup('RefractiveIndex').func('int1').set('argunit', 'm');
model.component('comp1').material('mat5').propertyGroup('RefractiveIndex').func('int1').set('fununit', '1');
model.component('comp1').material('mat5').propertyGroup('RefractiveIndex').func('int2').label('Refractive index, imaginary part');
model.component('comp1').material('mat5').propertyGroup('RefractiveIndex').func('int2').set('funcname', 'k_interp');
model.component('comp1').material('mat5').propertyGroup('RefractiveIndex').func('int2').set('table', {'6.67E-7' '2.9042727'; ...
'7.14E-7' '2.7924078';'7.69E-7' '2.6322954';'8.3299999999999999E-7' '2.76706';'9.09E-7' '3.0100921'; ...
'1.0E-6' '3.4208368';'1.05E-6' '3.722401';'1.11E-6' '3.9693342';'1.18E-6' '4.1791582';'1.2499999999999999E-6' '4.3367596'; ...
'1.33E-6' '4.4460869';'1.4299999999999999E-6' '4.3901355';'1.5399999999999999E-6' '4.7741297'; ...
'1.6699999999999999E-6' '5.4881071';'1.82E-6' '6.4453363';'2.0E-6' '7.5659499';'2.1099999999999997E-6' '8.2613235'; ...
'2.22E-6' '9.0476312';'2.35E-6' '9.8569158';'2.4999999999999998E-6' '10.710594';'2.67E-6' '11.554665'; ...
'2.8599999999999997E-6' '12.375593';'3.0799999999999997E-6' '13.261664';'3.33E-6' '14.40046'; ...
'3.64E-6' '15.887186';'4.0E-6' '17.69477';'4.44E-6' '19.89851';'4.9999999999999996E-6' '22.612178'; ...
'5.7099999999999995E-6' '26.005432';'6.67E-6' '30.343034'; '8.0E-6' '36.072044'; ...
'9.9999999999999999E-6' '44.019189';'1.1099999999999999E-5' '48.146015';'1.2499999999999999E-5' '53.112521'; ...
'1.43E-5' '59.208636';'1.67E-5' '66.923752';'1.9999999999999998E-5' '77.087223'; ...
'2.2199999999999998E-5' '83.517367';'2.4999999999999998E-5' '91.217414'; '2.86E-5' '100.62038'; ...
'3.3299999999999996E-5' '112.34062';'3.9999999999999996E-5' '128.08146';'4.4399999999999995E-5' '136.67195'; ...
'4.9999999999999996E-5' '146.58093';'5.71E-5' '159.83902';'6.67E-5' '177.92054'; ...
'7.999999999999999E-5' '202.21197';'9.999999999999999E-5' '234.03939';'1.25E-4' '265.86909'; ...
'1.54E-4' '294.94073';'1.9999999999999998E-4' '332.81796'});
model.component('comp1').material('mat5').propertyGroup('RefractiveIndex').func('int2').set('interp', 'piecewisecubic');
model.component('comp1').material('mat5').propertyGroup('RefractiveIndex').func('int2').set('argunit', 'm');
model.component('comp1').material('mat5').propertyGroup('RefractiveIndex').func('int2').set('fununit', '1');
model.component('comp1').material('mat5').propertyGroup('RefractiveIndex').set('n', '');
model.component('comp1').material('mat5').propertyGroup('RefractiveIndex').set('ki', '');
model.component('comp1').material('mat5').propertyGroup('RefractiveIndex').set('n', '');
model.component('comp1').material('mat5').propertyGroup('RefractiveIndex').set('ki', '');
model.component('comp1').material('mat5').propertyGroup('RefractiveIndex').set('n', {'n_interp(c_const/freq)' '0' '0' '0' 'n_interp(c_const/freq)' '0' '0' '
model.component('comp1').material('mat5').propertyGroup('RefractiveIndex').set('ki', {'k_interp(c_const/freq)' '0' '0' '0' 'k_interp(c_const/freq)' '0' '0' '
model.component('comp1').material('mat5').propertyGroup('RefractiveIndex').addInput('frequency');
model.component('comp1').material('mat7').label('Potassium [solid,tested at -78C (195K)]');
model.component('comp1').material('mat7').set('family', 'custom');
model.component('comp1').material('mat7').set('specular', 'custom');
model.component('comp1').material('mat7').set('customspecular', [0.7843137254901961 1 1]);
model.component('comp1').material('mat7').set('diffuse', 'custom');
model.component('comp1').material('mat7').set('customdiffuse', [0.7843137254901961 0.7843137254901961 0.7843137254901961]);
model.component('comp1').material('mat7').set('ambient', 'custom');
model.component('comp1').material('mat7').set('customambient', [0.7843137254901961 0.7843137254901961 0.7843137254901961]);
model.component('comp1').material('mat7').set('noise', true);
model.component('comp1').material('mat7').set('noisefreq', 1);
model.component('comp1').material('mat7').set('lighting', 'cooktorrance');
model.component('comp1').material('mat7').set('fresnel', 0.9);
model.component('comp1').material('mat7').propertyGroup('def').func('k_solid_1').set('arg', 'T');
model.component('comp1').material('mat7').propertyGroup('def').func('k_solid_1').set('pieces', {'0.0' '6.0' '1122.085*T^1+263.7268*T^2-227.3021*T
'6.0' '20.0' '6265.378-1779.74*T^1+224.4572*T^2-14.64801*T^3+0.4827659*T^4-0.006357878*T^5'; ...
'20.0' '50.0' '1340.183-148.6854*T^1+7.279199*T^2-0.1780719*T^3+0.002163925*T^4-1.042377E-5*T^5'; ...
'50.0' '100.0' '131.9865-0.6112663*T^1+0.004307492*T^2+2.385629E-6*T^3-9.319928E-8*T^4'; ...
'100.0' '273.0' '114.4634-0.07385023*T^1-3.37167E-4*T^2+4.432826E-6*T^3-1.303897E-8*T^4+1.09916E-11*T^5'; ...
'273.0' '337.0' '296.1808-2.610472*T^1+0.01292855*T^2-2.714074E-5*T^3+1.957806E-8*T^4'});
model.component('comp1').material('mat7').propertyGroup('def').func('res_solid_1').set('arg', 'T');
model.component('comp1').material('mat7').propertyGroup('def').func('res_solid_1').set('pieces', {'1.0' '5.6' '6.94E-12+2.721429E-12*T^1-1.428571
'5.6' '17.0' '7.187017E-11-2.871008E-11*T^1+3.196071E-12*T^2+5.622896E-14*T^3'; ...
'17.0' '58.6' '-5.804623E-10+2.361562E-11*T^1+3.744807E-12*T^2-2.436224E-14*T^3'; ...
'58.6' '336.0' '9.85994E-11+8.21212E-11*T^1+1.427422E-12*T^2-5.793066E-15*T^3+9.642156E-18*T^4'});
model.component('comp1').material('mat7').propertyGroup('def').func('Syfunc_solid_tested_at_195K_8').set('arg', 'epe');
model.component('comp1').material('mat7').propertyGroup('def').func('Syfunc_solid_tested_at_195K_8').set('pieces', {'0.0' '0.26' '310926.5+24785
model.component('comp1').material('mat7').propertyGroup('def').func('Syt').set('arg', 'T');
model.component('comp1').material('mat7').propertyGroup('def').func('Syt').set('pieces', {'5.0' '10.0' '1.057693E7-2491807.0*T^1+534385.5*T^2-52
'10.0' '20.0' '-2.309365E7+1.063972E7*T^1-1414767.0*T^2+85199.73*T^3-2383.933*T^4+25.21833*T^5'; ...
'20.0' '65.0' '-1856289.0+621900.6*T^1-18706.67*T^2+211.8753*T^3-0.8293287*T^4'; ...
'65.0' '195.0' '6823541.0-85137.6*T^1+442.1314*T^2-0.8750862*T^3'});
model.component('comp1').material('mat7').propertyGroup('def').func('C_solid_1').set('arg', 'T');
model.component('comp1').material('mat7').propertyGroup('def').func('C_solid_1').set('pieces', {'0.0' '1.8' '0.0468054*T^1+3.193572E-11*T^2+0.06
'1.8' '3.0' '-2.595532+3.512074*T^1-1.146209*T^2-0.02025602*T^3+0.05830843*T^4'; ...
'3.0' '5.0' '7.58581-6.425737*T^1+1.585278*T^2-0.02790238*T^3-2.555793E-4*T^4-5.115287E-6*T^5'; ...
'5.0' '40.0' '21.7639-12.58592*T^1+2.517692*T^2-0.08921844*T^3+0.001310724*T^4-6.945858E-6*T^5'; ...
'40.0' '140.0' '-38.20099+21.76232*T^1-0.2807147*T^2+0.001673195*T^3-3.737921E-6*T^4'; ...
'140.0' '336.6' '737.0564-2.830118*T^1+0.02801372*T^2-1.024661E-4*T^3+1.381151E-7*T^4'});
model.component('comp1').material('mat7').propertyGroup('def').func('sigma_solid_1').set('arg', 'T');
model.component('comp1').material('mat7').propertyGroup('def').func('sigma_solid_1').set('pieces', {'1.0' '5.6' '1/(2.500000E-13*T^3-1.428571E-12
'5.6' '17.0' '1/(5.622896E-14*T^3+3.196071E-12*T^2-2.871008E-11*T+7.187017E-11)'; ...
```

```
    '17.0' '58.6' '1/(-2.436224E-14*T^3+3.744807E-12*T^2+2.361562E-11*T-5.804623E-10)';  ...
    '58.6' '336.0' '1/(9.642156E-18*T^4-5.793066E-15*T^3+1.427422E-12*T^2+8.212120E-11*T+0.985994E-10)'});
model.component('comp1').material('mat7').propertyGroup('def').func('HC_solid_1').set('arg', 'T');
model.component('comp1').material('mat7').propertyGroup('def').func('HC_solid_1').set('pieces', {'0.0' '1.8' '0.001829998*T^1+1.248623E-12*T^2+(
    '1.8' '3.0' '-0.1014801+0.1373151*T^1-0.04481449*T^2-7.919697E-4*T^3+0.002279742*T^4';  ...
    '3.0' '5.0' '0.29659-0.2512335*T^1+0.06198119*T^2-0.001090927*T^3-9.992639E-6*T^4-1.999975E-7*T^5';  ...
    '5.0' '40.0' '0.8509252-0.4920844*T^1+0.09843672*T^2-0.003488262*T^3+5.124668E-5*T^4-2.715692E-7*T^5';  ...
    '40.0' '140.0' '-1.493582+0.8508633*T^1-0.01097539*T^2+6.54186E-5*T^3-1.461452E-7*T^4';  ...
    '140.0' '336.6' '28.81743-0.1106519*T^1+0.00109528*T^2-4.006218E-6*T^3+5.400025E-9*T^4'});
model.component('comp1').material('mat7').propertyGroup('def').func('VP_solid_1').set('arg', 'T');
model.component('comp1').material('mat7').propertyGroup('def').func('VP_solid_1').set('pieces', {'293.0' '337.0' '(exp((-4.646000e+03/T+7.841810(
model.component('comp1').material('mat7').propertyGroup('def').func('elong').set('arg', 'T');
model.component('comp1').material('mat7').propertyGroup('def').func('elong').set('pieces', {'5.0' '195.0' '42.51027+0.09061633*T^1-0.001950349*T
model.component('comp1').material('mat7').propertyGroup('def').set('thermalconductivity', {'k_solid_1(T[1/K])[W/(m*K)]' '0' '0' '0' 'k_solid_1(T[1/K])[W
model.component('comp1').material('mat7').propertyGroup('def').set('resistivity', {'res_solid_1(T[1/K])[ohm*m]' '0' '0' '0' 'res_solid_1(T[1/K])[ohm*m]'
model.component('comp1').material('mat7').propertyGroup('def').set('Syfunc', 'Syfunc_solid_tested_at_195K_8(epe)[Pa]');
model.component('comp1').material('mat7').propertyGroup('def').set('Syt', 'Syt(T[1/K])[Pa]');
model.component('comp1').material('mat7').propertyGroup('def').set('heatcapacity', 'C_solid_1(T[1/K])[J/(kg*K)]');
model.component('comp1').material('mat7').propertyGroup('def').set('electricconductivity', {'sigma_solid_1(T[1/K])[S/m]' '0' '0' '0' 'sigma_solid_1(T[1
model.component('comp1').material('mat7').propertyGroup('def').set('HC', 'HC_solid_1(T[1/K])[J/(mol*K)]');
model.component('comp1').material('mat7').propertyGroup('def').set('VP', 'VP_solid_1(T[1/K])[Pa]');
model.component('comp1').material('mat7').propertyGroup('def').set('elong', 'elong(T[1/K])');
model.component('comp1').material('mat7').propertyGroup('def').addInput('temperature');
model.component('comp1').material('mat7').propertyGroup('def').addInput('effectiveplasticstrain');
model.component('comp1').material('mat7').propertyGroup('Enu').func('E').set('arg', 'T');
model.component('comp1').material('mat7').propertyGroup('Enu').func('E').set('pieces', {'4.0' '295.0' '3.36682E9-3351970.0*T^1'});
model.component('comp1').material('mat7').propertyGroup('Enu').func('nu').set('arg', 'T');
model.component('comp1').material('mat7').propertyGroup('Enu').func('nu').set('pieces', {'4.0' '295.0' '0.3274145+1.473721E-4*T^1'});
model.component('comp1').material('mat7').propertyGroup('Enu').set('youngsmodulus', 'E(T[1/K])[Pa]');
model.component('comp1').material('mat7').propertyGroup('Enu').set('poissonsratio', 'nu(T[1/K])');
model.component('comp1').material('mat7').propertyGroup('Enu').addInput('temperature');
model.component('comp1').material('mat7').propertyGroup('KG').func('mu').set('arg', 'T');
model.component('comp1').material('mat7').propertyGroup('KG').func('mu').set('pieces', {'4.0' '295.0' '1.266401E9-1360805.0*T^1'});
model.component('comp1').material('mat7').propertyGroup('KG').func('kappa_solid_1').set('arg', 'T');
model.component('comp1').material('mat7').propertyGroup('KG').func('kappa_solid_1').set('pieces', {'4.0' '295.0' '3.265582E9-654512.0*T^1'});
model.component('comp1').material('mat7').propertyGroup('KG').set('K', '');
model.component('comp1').material('mat7').propertyGroup('KG').set('G', '');
model.component('comp1').material('mat7').propertyGroup('KG').set('K', 'kappa_solid_1(T[1/K])[Pa]');
model.component('comp1').material('mat7').propertyGroup('KG').set('G', 'mu(T[1/K])[Pa]');
model.component('comp1').material('mat7').propertyGroup('KG').addInput('temperature');
model.component('comp1').material('mat7').propertyGroup('ElastoplasticModel').func('Sys').set('arg', 'T');
model.component('comp1').material('mat7').propertyGroup('ElastoplasticModel').func('Sys').set('pieces', {'5.0' '25.0' '5003534.0-276355.7*T^1+395
model.component('comp1').material('mat7').propertyGroup('ElastoplasticModel').set('sigmags', '');
model.component('comp1').material('mat7').propertyGroup('ElastoplasticModel').set('Et', '');
model.component('comp1').material('mat7').propertyGroup('ElastoplasticModel').set('Ek', '');
model.component('comp1').material('mat7').propertyGroup('ElastoplasticModel').set('sigmagh', '');
model.component('comp1').material('mat7').propertyGroup('ElastoplasticModel').set('Hillcoefficients', '');
model.component('comp1').material('mat7').propertyGroup('ElastoplasticModel').set('ys', '');
model.component('comp1').material('mat7').propertyGroup('ElastoplasticModel').set('sigmags', 'Sys(T[1/K])[Pa]');
model.component('comp1').material('mat7').propertyGroup('ElastoplasticModel').set('Hillcoefficients', {'0' '0' '0' '0' '0' '0'});
model.component('comp1').material('mat7').propertyGroup('ElastoplasticModel').set('ys', {'0' '0' '0' '0' '0' '0'});
model.component('comp1').material('mat7').propertyGroup('ElastoplasticModel').addInput('temperature');
model.component('comp1').material('mat7').propertyGroup('RefractiveIndex').set('n', '');
model.component('comp1').material('mat7').propertyGroup('RefractiveIndex').set('ki', '');
model.component('comp1').material('mat7').propertyGroup('RefractiveIndex').set('n', {'4.77' '0' '0' '0' '4.77' '0' '0' '0' '4.77'});
model.component('comp1').material('mat7').propertyGroup('RefractiveIndex').set('ki', {'28.2' '0' '0' '0' '28.2' '0' '0' '0' '28.2'});
model.component('comp1').physics('ewfd').prop('EquationForm').set('freq', '1[GHz]');
model.component('comp1').physics('ewfd').prop('EquationForm').set('modeFreq', '1[GHz]');
model.component('comp1').physics('ewfd').prop('MeshControl').set('EnableMeshControl', false);
model.component('comp1').physics('ewfd').feature('port1').set('Pin', '1[mW]');
model.component('comp1').physics('ewfd').feature('port1').set('beta', 'abs(ewfd.k0)');
model.component('comp1').physics('ewfd').feature('port1').set('E0', {'0'; 'exp(-j*ewfd.k0*z)'; '0'});
model.component('comp1').physics('ewfd').feature('port2').set('beta', 'abs(ewfd.k0)');
model.component('comp1').physics('ewfd').feature('port2').set('E0', {'0'; 'exp(-j*ewfd.k0*z)'; '0'});
model.component('comp1').mesh('mesh1').feature('size').set('hauto', 4);
model.component('comp1').mesh('mesh1').run;
model.study.create('std1');
model.study('std1').create('freq', 'Frequency');
model.sol.create('sol1');
model.sol('sol1').study('std1');
model.sol('sol1').attach('std1');
model.sol('sol1').create('st1', 'StudyStep');
model.sol('sol1').create('v1', 'Variables');
model.sol('sol1').create('s1', 'Stationary');
model.sol('sol1').feature('s1').create('p1', 'Parametric');
model.sol('sol1').feature('s1').create('fc1', 'FullyCoupled');
model.sol('sol1').feature('s1').create('i1', 'Iterative');
model.sol('sol1').feature('s1').feature('i1').create('mg1', 'Multigrid');
model.sol('sol1').feature('s1').feature('i1').feature('mg1').feature('pr').create('sv1', 'SORVector');
model.sol('sol1').feature('s1').feature('i1').feature('mg1').feature('po').create('sv1', 'SORVector');
model.sol('sol1').feature('s1').feature.remove('fcDef');
model.result.numerical.create('int1', 'IntSurface');
model.result.numerical.create('int2', 'IntSurface');
model.result.numerical('int1').selection.set([13 29]);
model.result.numerical('int1').set('probetag', 'none');
model.result.numerical('int2').selection.set([13 29]);
```

```
model.result.numerical('int2').set('probetag', 'none');
model.result.create('pg1', 'PlotGroup3D');
model.result.create('pg2', 'PlotGroup3D');
model.result.create('pg3', 'PlotGroup1D');
model.result.create('pg4', 'PlotGroup3D');
model.result('pg1').create('mslc1', 'Multislice');
model.result('pg2').create('slc1', 'Slice');
model.result('pg2').feature('slc1').create('def1', 'Deform');
model.result('pg3').create('tblp1', 'Table');
model.result('pg4').create('mslc1', 'Multislice');
model.study('std1').feature('freq').set('punit', 'Hz');
model.study('std1').feature('freq').set('plist', '1.0e12');
model.sol('sol1').attach('std1');
model.sol('sol1').feature('v1').set('clistctrl', {'p1'});
model.sol('sol1').feature('v1').set('cname', {'freq'});
model.sol('sol1').feature('v1').set('clist', {'1.0e12[Hz]'});
model.sol('sol1').feature('s1').feature('pDef').set('pname', {'freq'});
model.sol('sol1').feature('s1').feature('pDef').set('plistarr', {'1.0e12'});
model.sol('sol1').feature('s1').feature('pDef').set('punit', {'Hz'});
model.sol('sol1').feature('s1').feature('pDef').set('pcontinuationmode', 'no');
model.sol('sol1').feature('s1').feature('pDef').set('preusesol', 'auto');
model.sol('sol1').feature('s1').feature('pDef').set('useIsqdata', false);
model.sol('sol1').feature('s1').feature('p1').set('pname', {'freq'});
model.sol('sol1').feature('s1').feature('p1').set('plistarr', {'1.0e12'});
model.sol('sol1').feature('s1').feature('p1').set('punit', {'Hz'});
model.sol('sol1').feature('s1').feature('p1').set('pcontinuationmode', 'no');
model.sol('sol1').feature('s1').feature('p1').set('preusesol', 'auto');
model.sol('sol1').feature('s1').feature('i1').label('Suggested Iterative Solver (ewfd)');
model.sol('sol1').feature('s1').feature('i1').set('linsolver', 'bicgstab');
model.sol('sol1').feature('s1').feature('i1').feature('mg1').feature('pr').feature('sv1').set('sorvecdof', {'comp1_E'});
model.sol('sol1').feature('s1').feature('i1').feature('mg1').feature('po').feature('sv1').set('sorvecdof', {'comp1_E'});
model.sol('sol1').runAll;
model.result.numerical('int1').set('table', 'tbl1');
model.result.numerical('int1').set('expr', {'ewfd.normE'});
model.result.numerical('int1').set('unit', {'V*m'});
model.result.numerical('int1').set('descr', {'Electric field norm'});
model.result.numerical('int2').set('table', 'tbl2');
model.result.numerical('int2').set('expr', {'ewfd.Qh'});
model.result.numerical('int2').set('unit', {'W/m'});
model.result.numerical('int2').set('descr', {'Total power dissipation density'});
model.result.numerical('int1').setResult;
model.result.numerical('int2').setResult;
model.result('pg1').label('Electric Field (ewfd)');
model.result('pg1').set('frametype', 'spatial');
model.result('pg1').feature('mslc1').set('multiplanezmethod', 'coord');
model.result('pg1').feature('mslc1').set('zcoord', '1050.0');
model.result('pg1').feature('mslc1').set('rangecoloractive', true);
model.result('pg1').feature('mslc1').set('rangecolormin', 41923.82486663746);
model.result('pg1').feature('mslc1').set('rangecolormax', 400000);
model.result('pg1').feature('mslc1').set('smooth', 'internal');
model.result('pg1').feature('mslc1').set('resolution', 'normal');
model.result('pg2').feature('slc1').set('quickplane', 'xy');
model.result('pg2').feature('slc1').set('quickzmethod', 'coord');
model.result('pg2').feature('slc1').set('quickz', '1033.0');
model.result('pg2').feature('slc1').set('rangecoloractive', true);
model.result('pg2').feature('slc1').set('rangecolormin', 9292.73707567522);
model.result('pg2').feature('slc1').set('rangecolormax', 250000);
model.result('pg2').feature('slc1').set('smooth', 'internal');
model.result('pg2').feature('slc1').set('resolution', 'normal');
model.result('pg2').feature('slc1').feature('def1').active(false);
model.result('pg2').feature('slc1').feature('def1').set('expr', {'+ewfd.Ex' '+ewfd.Ey' '+ewfd.Ez'});
model.result('pg2').feature('slc1').feature('def1').set('descr', ' + Electric field');
model.result('pg2').feature('slc1').feature('def1').set('scale', 2.085880595474555E-4);
model.result('pg2').feature('slc1').feature('def1').set('scaleactive', false);
model.result('pg3').set('data', 'none');
model.result('pg3').set('xlabel', 'freq (Hz)');
model.result('pg3').set('ylabel', 'Electric field norm (V*m)');
model.result('pg3').set('xlabelactive', false);
model.result('pg3').set('ylabelactive', false);
model.result('pg3').feature('tblp1').set('linewidth', 3);
model.result('pg3').feature('tblp1').set('linemarker', 'cycle');
model.result('pg3').feature('tblp1').set('markerpos', 'datapoints');
model.result('pg4').set('frametype', 'spatial');
model.result('pg4').feature('mslc1').set('expr', 'ewfd.Qh');
model.result('pg4').feature('mslc1').set('unit', 'W/m^3');
model.result('pg4').feature('mslc1').set('descr', 'Total power dissipation density');
model.result('pg4').feature('mslc1').set('multiplanezmethod', 'coord');
model.result('pg4').feature('mslc1').set('zcoord', 1050);
model.result('pg4').feature('mslc1').set('rangecoloractive', true);
model.result('pg4').feature('mslc1').set('rangecolormin', 1);
model.result('pg4').feature('mslc1').set('rangecolormax', '8E10');
model.result('pg4').feature('mslc1').set('resolution', 'normal');
end
```

```
function [] = fBiAg(model)
model.component('comp1').material.create('mat1', 'Common');
model.component('comp1').material.create('mat3', 'Common');
model.component('comp1').material.create('mat5', 'Common');
model.component('comp1').material('mat1').propertyGroup('def').func.create('eta', 'Piecewise');
model.component('comp1').material('mat1').propertyGroup('def').func.create('Cp', 'Piecewise');
model.component('comp1').material('mat1').propertyGroup('def').func.create('rho', 'Analytic');
model.component('comp1').material('mat1').propertyGroup('def').func.create('k', 'Piecewise');
model.component('comp1').material('mat1').propertyGroup('def').func.create('cs', 'Analytic');
model.component('comp1').material('mat1').propertyGroup.create('RefractiveIndex', 'Refractive index');
model.component('comp1').material('mat3').selection.set([3]);
model.component('comp1').material('mat3').propertyGroup('def').func.create('k_solid_1', 'Piecewise');
model.component('comp1').material('mat3').propertyGroup('def').func.create('res_solid_1', 'Piecewise');
model.component('comp1').material('mat3').propertyGroup('def').func.create('alpha', 'Piecewise');
model.component('comp1').material('mat3').propertyGroup('def').func.create('C_solid_1', 'Piecewise');
model.component('comp1').material('mat3').propertyGroup('def').func.create('sigma_solid_1', 'Piecewise');
model.component('comp1').material('mat3').propertyGroup('def').func.create('HC_solid_1', 'Piecewise');
model.component('comp1').material('mat3').propertyGroup('def').func.create('VP_solid_1', 'Piecewise');
model.component('comp1').material('mat3').propertyGroup('def').func.create('rho', 'Piecewise');
model.component('comp1').material('mat3').propertyGroup.create('ThermalExpansion', 'Thermal expansion');
model.component('comp1').material('mat3').propertyGroup('ThermalExpansion').func.create('dL', 'Piecewise');
model.component('comp1').material('mat3').propertyGroup('ThermalExpansion').func.create('CTE', 'Piecewise');
model.component('comp1').material('mat3').propertyGroup.create('Enu', 'Young''s modulus and Poisson''s ratio');
model.component('comp1').material('mat3').propertyGroup('Enu').func.create('E', 'Piecewise');
model.component('comp1').material('mat3').propertyGroup('Enu').func.create('nu', 'Piecewise');
model.component('comp1').material('mat3').propertyGroup.create('KG', 'Bulk modulus and shear modulus');
model.component('comp1').material('mat3').propertyGroup('KG').func.create('mu', 'Piecewise');
model.component('comp1').material('mat3').propertyGroup('KG').func.create('kappa', 'Piecewise');
model.component('comp1').material('mat3').propertyGroup.create('RefractiveIndex', 'Refractive index');
model.component('comp1').material('mat5').selection.set([4]);
model.component('comp1').material('mat5').propertyGroup.create('RefractiveIndex', 'Refractive index');
model.component('comp1').material('mat5').propertyGroup('RefractiveIndex').func.create('int1', 'Interpolation');
model.component('comp1').material('mat5').propertyGroup('RefractiveIndex').func.create('int2', 'Interpolation');
model.component('comp1').physics.create('ewfd', 'ElectromagneticWavesFrequencyDomain', 'geom1');
model.component('comp1').physics('ewfd').create('sctr1', 'Scattering', 2);
model.component('comp1').physics('ewfd').feature('sctr1').selection.set([1 2 3 4 5 7 8 9 43 44]);
model.component('comp1').physics('ewfd').create('port1', 'Port', 2);
model.component('comp1').physics('ewfd').feature('port1').selection.set([7]);
model.component('comp1').physics('ewfd').create('port2', 'Port', 2);
model.component('comp1').physics('ewfd').feature('port2').selection.set([3]);
model.component('comp1').mesh('mesh1').create('ftet1', 'FreeTet');
model.result.table('tbl1').comments('Surface Integration 1 (ewfd.normE)');
model.result.table('evl3').label('Evaluation 3D');
model.result.table('evl3').comments('Interactive 3D values');
model.component('comp1').view('view1').set('renderwireframe', true);
model.component('comp1').view('view2').axis.set('xmin', -37.6529541015625);
model.component('comp1').view('view2').axis.set('xmax', 2112.7470703125);
model.component('comp1').view('view2').axis.set('ymin', 0);
model.component('comp1').view('view2').axis.set('ymax', 1651.199951171875);
model.component('comp1').material('mat1').label('Air');
model.component('comp1').material('mat1').set('family', 'air');
model.component('comp1').material('mat1').propertyGroup('def').func('eta').set('arg', 'T');
model.component('comp1').material('mat1').propertyGroup('def').func('eta').set('pieces', {'200.0' '1600.0' '-8.38278E-7+8.35717342E-8*T^1-7.6942
model.component('comp1').material('mat1').propertyGroup('def').func('Cp').set('arg', 'T');
model.component('comp1').material('mat1').propertyGroup('def').func('Cp').set('pieces', {'200.0' '1600.0' '1047.63657-0.372589265*T^1+9.453042
model.component('comp1').material('mat1').propertyGroup('def').func('rho').set('expr', 'pA*0.02897/8.314/T');
model.component('comp1').material('mat1').propertyGroup('def').func('rho').set('args', {'pA' 'T'});
model.component('comp1').material('mat1').propertyGroup('def').func('rho').set('dermethod', 'manual');
model.component('comp1').material('mat1').propertyGroup('def').func('rho').set('argders', {'pA' 'd(pA*0.02897/8.314/T,pA)'; 'T' 'd(pA*0.02897/8.314/
model.component('comp1').material('mat1').propertyGroup('def').func('rho').set('plotargs', {'pA' '0' '1'; 'T' '0' '1'});
model.component('comp1').material('mat1').propertyGroup('def').func('k').set('arg', 'T');
model.component('comp1').material('mat1').propertyGroup('def').func('k').set('pieces', {'200.0' '1600.0' '-0.00227583562+1.15480022E-4*T^1-7.90
model.component('comp1').material('mat1').propertyGroup('def').func('cs').set('expr', 'sqrt(1.4*287*T)');
model.component('comp1').material('mat1').propertyGroup('def').func('cs').set('args', {'T'});
model.component('comp1').material('mat1').propertyGroup('def').func('cs').set('dermethod', 'manual');
model.component('comp1').material('mat1').propertyGroup('def').func('cs').set('argders', {'T' 'd(sqrt(1.4*287*T),T)'});
model.component('comp1').material('mat1').propertyGroup('def').func('cs').set('plotargs', {'T' '0' '1'});
model.component('comp1').material('mat1').propertyGroup('def').set('relpermeability', {'1' '0' '0' '0' '1' '0' '0' '0' '1'});
model.component('comp1').material('mat1').propertyGroup('def').set('relpermittivity', {'1' '0' '0' '0' '1' '0' '0' '0' '1'});
model.component('comp1').material('mat1').propertyGroup('def').set('dynamicviscosity', 'eta(T[1/K])[Pa*s]');
model.component('comp1').material('mat1').propertyGroup('def').set('ratioofspecificheat', '1.4');
model.component('comp1').material('mat1').propertyGroup('def').set('electricconductivity', {'0[S/m]' '0' '0' '0' '0[S/m]' '0' '0' '0' '0[S/m]'});
model.component('comp1').material('mat1').propertyGroup('def').set('heatcapacity', 'Cp(T[1/K])[J/(kg*K)]');
model.component('comp1').material('mat1').propertyGroup('def').set('density', 'rho(pA[1/Pa],T[1/K])[kg/m^3]');
model.component('comp1').material('mat1').propertyGroup('def').set('thermalconductivity', {'k(T[1/K])[W/(m*K)]' '0' '0' '0' 'k(T[1/K])[W/(m*K)]' '0' '0' '0'
model.component('comp1').material('mat1').propertyGroup('def').set('soundspeed', 'cs(T[1/K])[m/s]');
model.component('comp1').material('mat1').propertyGroup('def').addInput('temperature');
model.component('comp1').material('mat1').propertyGroup('def').addInput('pressure');
model.component('comp1').material('mat1').propertyGroup('RefractiveIndex').set('n', '');
model.component('comp1').material('mat1').propertyGroup('RefractiveIndex').set('ki', '');
model.component('comp1').material('mat1').propertyGroup('RefractiveIndex').set('n', {'1' '0' '0' '0' '1' '0' '0' '0' '1'});
model.component('comp1').material('mat1').propertyGroup('RefractiveIndex').set('ki', {'0' '0' '0' '0' '0' '0' '0' '0' '0'});
model.component('comp1').material('mat3').label('Silver [solid]');
model.component('comp1').material('mat3').set('family', 'custom');
model.component('comp1').material('mat3').set('specular', 'custom');
model.component('comp1').material('mat3').set('customspecular', [0.7843137254901961 1 1]);
```

```
model.component('comp1').material('mat3').set('diffuse', 'custom');
model.component('comp1').material('mat3').set('customdiffuse', [0.7843137254901961 0.7843137254901961 0.7843137254901961]);
model.component('comp1').material('mat3').set('ambient', 'custom');
model.component('comp1').material('mat3').set('customambient', [0.7843137254901961 0.7843137254901961 0.7843137254901961]);
model.component('comp1').material('mat3').set('noise', true);
model.component('comp1').material('mat3').set('noisefreq', 1);
model.component('comp1').material('mat3').set('lighting', 'cooktorrance');
model.component('comp1').material('mat3').set('fresnel', 0.9);
model.component('comp1').material('mat3').propertyGroup('def').func('k_solid_1').set('arg', 'T');
model.component('comp1').material('mat3').propertyGroup('def').func('k_solid_1').set('pieces', {'0.0' '12.0' '3587.152*T^1+348.8866*T^2-101.463*T
'12.0' '35.0' '50522.05-4753.283*T^1+172.2387*T^2-2.689213*T^3+0.01433909*T^4';  ...
'35.0' '100.0' '10048.41-524.6271*T^1+11.6041*T^2-0.1287378*T^3+7.123183E-4*T^4-1.566974E-6*T^5';  ...
'100.0' '250.0' '759.2297-7.120703*T^1+0.06197118*T^2-2.715928E-4*T^3+6.016276E-7*T^4-5.424229E-10*T^5';  ...
'250.0' '1235.0' '419.8682+0.09979317*T^1-2.937158E-4*T^2+2.109166E-7*T^3-5.786644E-11*T^4'});
model.component('comp1').material('mat3').propertyGroup('def').func('res_solid_1').set('arg', 'T');
model.component('comp1').material('mat3').propertyGroup('def').func('res_solid_1').set('pieces', {'1.0' '15.8' '9.822048E-12+2.259567E-13*T^1-6.6
'15.8' '27.5' '-2.33E-11+7.473333E-12*T^1-6.16E-13*T^2+2.026667E-14*T^3';  ...
'27.5' '60.0' '8.015476E-10-7.586429E-11*T^1+2.210952E-12*T^2-1.2E-14*T^3';  ...
'60.0' '200.0' '-2.428741E-9+6.974508E-11*T^1-4.447028E-14*T^2+6.841184E-17*T^3';  ...
'200.0' '1235.0' '-1.812752E-9+6.074742E-11*T^1-3.077059E-15*T^2+8.269045E-18*T^3'});
model.component('comp1').material('mat3').propertyGroup('def').func('alpha').set('arg', 'T');
model.component('comp1').material('mat3').propertyGroup('def').func('alpha').set('pieces', {'0.0' '92.0' '1.397008E-5+6.293815E-8*T^1-3.772802E-
model.component('comp1').material('mat3').propertyGroup('def').func('C_solid_1').set('arg', 'T');
model.component('comp1').material('mat3').propertyGroup('def').func('C_solid_1').set('pieces', {'1.0' '12.3' '0.01224227-0.01394369*T^1+0.01009
'12.3' '75.0' '24.24847-4.669813*T^1+0.2956444*T^2-0.004853545*T^3+3.380225E-5*T^4-8.439015E-8*T^5';  ...
'75.0' '300.0' '-63.85884+5.177265*T^1-0.03961478*T^2+1.570454E-4*T^3-3.105375E-7*T^4+2.411435E-10*T^5';  ...
'300.0' '1235.0' '225.7065+0.01705702*T^1+5.007143E-5*T^2-1.768498E-8*T^3'});
model.component('comp1').material('mat3').propertyGroup('def').func('sigma_solid_1').set('arg', 'T');
model.component('comp1').material('mat3').propertyGroup('def').func('sigma_solid_1').set('pieces', {'1.0' '15.8' '1/(6.144183E-15*T^3-6.690094E-1
'15.8' '27.5' '1/(2.026667E-14*T^3-6.160000E-13*T^2+7.473333E-12*T-2.330000E-11)';  ...
'27.5' '60.0' '1/(-1.200000E-14*T^3+2.210952E-12*T^2-7.586429E-11*T+8.015476E-10)';  ...
'60.0' '200.0' '1/(6.841184E-17*T^3-4.447028E-14*T^2+6.974508E-11*T-2.428741E-09)';  ...
'200.0' '1235.0' '1/(8.269045E-18*T^3-3.077059E-15*T^2+6.074742E-11*T-1.812752E-09)'});
model.component('comp1').material('mat3').propertyGroup('def').func('HC_solid_1').set('arg', 'T');
model.component('comp1').material('mat3').propertyGroup('def').func('HC_solid_1').set('pieces', {'1.0' '12.3' '0.001320549-0.001504078*T^1+0.00
'12.3' '75.0' '2.615634-0.5037235*T^1+0.03189057*T^2-5.235422E-4*T^3+3.646182E-6*T^4-9.102999E-9*T^5';  ...
'75.0' '300.0' '-6.888324+0.5584615*T^1-0.004273165*T^2+1.694017E-5*T^3-3.349706E-8*T^4+2.601167E-11*T^5';  ...
'300.0' '1235.0' '24.34652+0.001839907*T^1+5.401105E-6*T^2-1.907643E-9*T^3'});
model.component('comp1').material('mat3').propertyGroup('def').func('VP_solid_1').set('arg', 'T');
model.component('comp1').material('mat3').propertyGroup('def').func('VP_solid_1').set('pieces', {'293.0' '1235.0' '(exp((-1.499900e+04/T-7.845000
model.component('comp1').material('mat3').propertyGroup('def').func('rho').set('arg', 'T');
model.component('comp1').material('mat3').propertyGroup('def').func('rho').set('pieces', {'0.0' '30.0' '10630.38'; '30.0' '140.0' '10630.25+0.1394367
model.component('comp1').material('mat3').propertyGroup('def').set('thermalconductivity', {'k_solid_1(T[1/K])[W/(m*K)]' '0' '0' '0' 'k_solid_1(T[1/K])[W
model.component('comp1').material('mat3').propertyGroup('def').set('resistivity', {'res_solid_1(T[1/K])[ohm*m]' '0' '0' '0' 'res_solid_1(T[1/K])[ohm*m]'
model.component('comp1').material('mat3').propertyGroup('def').set('thermalexpansioncoefficient', {'(alpha(T[1/K])[1/K]+(Tempref-293[K])*if(abs(T-T
model.component('comp1').material('mat3').propertyGroup('def').set('heatcapacity', 'C_solid_1(T[1/K])[J/(kg*K)]');
model.component('comp1').material('mat3').propertyGroup('def').set('electricconductivity', {'sigma_solid_1(T[1/K])[S/m]' '0' '0' '0' 'sigma_solid_1(T[1/
model.component('comp1').material('mat3').propertyGroup('def').set('HC', 'HC_solid_1(T[1/K])[J/(mol*K)]');
model.component('comp1').material('mat3').propertyGroup('def').set('VP', 'VP_solid_1(T[1/K])[Pa]');
model.component('comp1').material('mat3').propertyGroup('def').set('density', 'rho(T[1/K])[kg/m^3]');
model.component('comp1').material('mat3').propertyGroup('def').addInput('temperature');
model.component('comp1').material('mat3').propertyGroup('def').addInput('strainreferencetemperature');
model.component('comp1').material('mat3').propertyGroup('ThermalExpansion').func('dL').set('arg', 'T');
model.component('comp1').material('mat3').propertyGroup('ThermalExpansion').func('dL').set('pieces', {'0.0' '10.0' '-0.004134001';  ...
'10.0' '30.0' '-0.00415083+2.566303E-6*T^1-1.154799E-7*T^2+2.71357E-9*T^3';  ...
'30.0' '87.0' '-0.004065065-6.856586E-6*T^1+2.120401E-7*T^2-9.814541E-10*T^3+2.367498E-12*T^4-2.275841E-15*T^5';  ...
'87.0' '873.0' '-0.004745213+1.182195E-5*T^1+1.97186E-8*T^2-1.837579E-11*T^3+6.889976E-15*T^4'});
model.component('comp1').material('mat3').propertyGroup('ThermalExpansion').func('CTE').set('arg', 'T');
model.component('comp1').material('mat3').propertyGroup('ThermalExpansion').func('CTE').set('pieces', {'30.0' '300.0' '-8.786433E-6+5.111237E-7
model.component('comp1').material('mat3').propertyGroup('ThermalExpansion').set('alphatan', '');
model.component('comp1').material('mat3').propertyGroup('ThermalExpansion').set('dL', '');
model.component('comp1').material('mat3').propertyGroup('ThermalExpansion').set('alphatanIso', '');
model.component('comp1').material('mat3').propertyGroup('ThermalExpansion').set('dLIso', '');
model.component('comp1').material('mat3').propertyGroup('ThermalExpansion').set('alphatan', {'CTE(T[1/K])[1/K]' '0' '0' '0' 'CTE(T[1/K])[1/K]' '0' '0' '0
model.component('comp1').material('mat3').propertyGroup('ThermalExpansion').set('dL', {'(dL(T[1/K])-dL(Tempref[1/K]))/(1+dL(Tempref[1/K]))' '0' '0'
model.component('comp1').material('mat3').propertyGroup('ThermalExpansion').set('alphatanIso', 'CTE(T)');
model.component('comp1').material('mat3').propertyGroup('ThermalExpansion').set('dLIso', '(dL(T)-dL(Tempref))/(1+dL(Tempref))');
model.component('comp1').material('mat3').propertyGroup('ThermalExpansion').addInput('temperature');
model.component('comp1').material('mat3').propertyGroup('ThermalExpansion').addInput('strainreferencetemperature');
model.component('comp1').material('mat3').propertyGroup('Enu').func('E').set('arg', 'T');
model.component('comp1').material('mat3').propertyGroup('Enu').func('E').set('pieces', {'0.0' '1173.0' '9.143965E10-2.775728E7*T^1-38123.31*T^2
model.component('comp1').material('mat3').propertyGroup('Enu').func('nu').set('arg', 'T');
model.component('comp1').material('mat3').propertyGroup('Enu').func('nu').set('pieces', {'0.0' '1173.0' '0.360203+1.499369E-5*T^1+4.008534E-8*T
model.component('comp1').material('mat3').propertyGroup('Enu').set('youngsmodulus', 'E(T[1/K])[Pa]');
model.component('comp1').material('mat3').propertyGroup('Enu').set('poissonsratio', 'nu(T[1/K])');
model.component('comp1').material('mat3').propertyGroup('Enu').addInput('temperature');
model.component('comp1').material('mat3').propertyGroup('KG').func('mu').set('arg', 'T');
model.component('comp1').material('mat3').propertyGroup('KG').func('mu').set('pieces', {'0.0' '1173.0' '3.362582E10-6732560.0*T^1-31364.68*T^2
model.component('comp1').material('mat3').propertyGroup('KG').func('kappa').set('arg', 'T');
model.component('comp1').material('mat3').propertyGroup('KG').func('kappa').set('pieces', {'0.0' '1173.0' '1.088261E11-3077357.0*T^1-87636.32*
model.component('comp1').material('mat3').propertyGroup('KG').set('K', '');
model.component('comp1').material('mat3').propertyGroup('KG').set('G', '');
model.component('comp1').material('mat3').propertyGroup('KG').set('K', 'kappa(T[1/K])[Pa]');
model.component('comp1').material('mat3').propertyGroup('KG').set('G', 'mu(T[1/K])[Pa]');
model.component('comp1').material('mat3').propertyGroup('KG').addInput('temperature');
model.component('comp1').material('mat3').propertyGroup('RefractiveIndex').set('n', '');
```

```
model.component('comp1').material('mat3').propertyGroup('RefractiveIndex').set('ki', '');
model.component('comp1').material('mat3').propertyGroup('RefractiveIndex').set('n', {'1.6090' '0' '0' '0' '1.6090' '0' '0' '0' '1.6090'});
model.component('comp1').material('mat3').propertyGroup('RefractiveIndex').set('ki', {'0.91260' '0' '0' '0' '0.91260' '0' '0' '0' '0.91260'});
model.component('comp1').material('mat5').label(['Bi (Bismuth) (Hagemann et al. 1974: n,k 0.00155-6.199 ' native2unicode(hex2dec({'00' 'b5'}), 'uni
model.component('comp1').material('mat5').set('groups', {'inorganic' 'Inorganic Materials' 'bi_bismuth' 'Bi - Bismuth' 'experimental_data' 'Experimen
model.component('comp1').material('mat5').propertyGroup('RefractiveIndex').func('int1').label('Refractive index');
model.component('comp1').material('mat5').propertyGroup('RefractiveIndex').func('int1').set('funcname', 'n_interp');
model.component('comp1').material('mat5').propertyGroup('RefractiveIndex').func('int1').set('table', {'2.48E-12' '1.0';  ...
'4.133E-12' '1.0';  '1.2399999999999999E-11' '1.0';  '1.378E-11' '1.0';  '1.4589999999999998E-11' '1.0';  '2.4799999999999998E-11' '1.0';  ...
'4.1329999999999995E-11' '1.0'; '6.199E-11' '1.0'; '8.265999999999999E-11' '1.0';  '8.856E-11' '1.0';  '9.537E-11' '1.0';  '1.033E-10' '1.0';  ...
'1.24E-10' '1.0'; '1.5499999999999998E-10' '1.0'; '2.066E-10' '1.0'; '3.0999999999999996E-10' '1.0'; '3.542E-10' '1.0';  ...
'3.757E-10' '1.0'; '3.8749999999999955E-10' '1.0'; '3.999E-10' '1.0'; '4.133E-10' '1.0'; '4.4279999999999996E-10' '1.0'; '4.769E-10' '1.0';  ...
'4.958999999999999E-10' '1.0'; '5.0609999999999999E-10' '1.0'; '5.166E-10' '1.0'; '5.636E-10' '1.0'; '6.199E-10' '1.0';  ...
'6.888E-10' '1.0'; '7.749E-10' '1.0'; '8.8559999999999999E-10' '0.999';'9.537E-10' '0.999';'1.033E-9' '0.999'; '1.127E-9' '0.999';  ...
'1.2399999999999999E-9' '0.999'; '1.305E-9' '0.999';'1.378E-9' '0.999';'1.459E-9' '0.999';'1.5499999999999998E-9' '0.999';  ...
'1.653E-9' '0.998'; '1.771E-9' '0.998'; '1.907E-9' '0.998'; '2.066E-9' '0.998';'2.254E-9' '0.998'; '2.4799999999999997E-9' '0.998';  ...
'2.583E-9' '0.998'; '2.695E-9' '0.997'; '2.818E-9' '0.997'; '2.952E-9' '0.997';'3.0999999999999996E-9' '0.997';'3.179E-9' '0.997';  ...
'3.2629999999999996E-9' '0.997';'3.3509999999999997E-9' '0.997'; '3.4439999999999997E-9' '0.997'; '3.542E-9' '0.997';  ...
'3.647E-9' '0.997';'3.757E-9' '0.997';'3.875E-9' '0.997';'3.999E-9' '0.997';'4.1330000000000005E-9' '0.997';  ...
'4.275E-9' '0.997';'4.4279999999999999E-9' '0.996'; '4.592E-9' '0.996';  '4.7689999999999996E-9' '0.996';  ...
'4.959E-9' '0.996'; '5.166E-9' '0.995'; '5.391E-9' '0.994'; '5.636E-9' '0.993'; '5.904E-9' '0.991'; '6.1989999999999999E-9' '0.99';  ...
'6.5250000000000005E-9' '0.988'; '6.887999999999999E-9' '0.986'; '7.2929999999999995E-9' '0.984';  '7.748999999999999E-9' '0.98';  ...
'8.266000000000001E-9' '0.976'; '8.551E-9' '0.974'; '8.855999999999998E-9' '0.972';'9.184E-9' '0.969'; '9.537E-9' '0.966';  ...
'9.919000000000001E-9' '0.964';'1.033E-8' '0.96'; '1.078E-8' '0.956'; '1.127E-8' '0.953'; '1.181E-8' '0.949'; '1.2399999999999999E-8' '0.945';  ...
'1.305000000000001E-8' '0.94';'1.378E-8' '0.935';'1.459E-8' '0.933';'1.55E-8' '0.932'; '1.653E-8' '0.933';'1.7709999999999998E-8' '0.942';  ...
'1.9069999999999998E-8' '0.955'; '2.066E-8' '0.966';'2.101E-8' '0.97';'2.138E-8' '0.975';'2.1749999999999996E-8' '0.981'; '2.214E-8' '0.987';  ...
'2.254E-8' '0.993';'2.2960000000000002E-8' '1.0';'2.339E-8' '1.007'; '2.384E-8' '1.014';'2.4309999999999997E-8' '1.022';  ...
'2.479999999999997E-8' '1.027';'2.5829999999999997E-8' '1.033';'2.695E-8' '1.036';'2.8179999999999998E-8' '1.041';  ...
'2.952E-8' '1.048';'3.1E-8' '1.051';'3.2629999999999995E-8' '1.046';'3.4439999999999996E-8' '1.038';'3.647E-8' '1.027'; '3.757E-8' '1.02';  ...
'3.875E-8' '1.009';'3.9989999999999996E-8' '0.997';'4.1329999999999994E-8' '0.987';'4.202999999999999E-8' '0.988';'4.275E-8' '0.991';  ...
'4.3499999999999999E-8' '0.987';'4.428E-8' '0.998'; '4.5089999999999995E-8' '1.035';'4.5920000000000004E-8' '1.011';'4.679E-8' '0.998';  ...
'4.769E-8' '0.999';'4.8619999999999995E-8' '1.001'; '4.959E-8' '1.004';'5.061E-8' '1.07';'5.1659999999999994E-8' '1.065';  ...
'5.276E-8' '1.013';'5.391E-8' '0.981';'5.6359999999999996E-8' '0.936';'6.199E-8' '0.856';'6.8879999999999999E-8' '0.763';  ...
'7.749E-8' '0.692';'8.856E-8' '0.64';'1.0329999999999999E-7' '0.627'; '1.24E-7' '0.71'; '1.55E-7' '0.904'; '2.0659999999999998E-7' '1.364';  ...
'2.48E-7' '1.482';'3.1E-7' '1.517';'3.542E-7' '1.541';'4.133E-7' '1.556'; '4.959E-7' '1.865';'6.199E-7' '2.244';'8.266E-7' '2.861';  ...
'1.24E-6' '5.036';'1.55E-6' '6.247';'2.0659999999999998E-6' '7.466'; '3.1E-6' '8.572';'6.1989999999999994E-6' '6.908'});
model.component('comp1').material('mat5').propertyGroup('RefractiveIndex').func('int1').set('interp', 'piecewisecubic');
model.component('comp1').material('mat5').propertyGroup('RefractiveIndex').func('int1').set('argunit', 'm');
model.component('comp1').material('mat5').propertyGroup('RefractiveIndex').func('int1').set('fununit', '1');
model.component('comp1').material('mat5').propertyGroup('RefractiveIndex').func('int2').label('Refractive index, imaginary part');
model.component('comp1').material('mat5').propertyGroup('RefractiveIndex').func('int2').set('funcname', 'k_interp');
model.component('comp1').material('mat5').propertyGroup('RefractiveIndex').func('int2').set('table', {'2.48E-12' '2.02E-11';  ...
'4.133E-12' '1.19E-10';'1.2399999999999999E-11' '5.71E-9';'1.378E-11' '7.13E-9';'1.4589999999999998E-11' '2.56E-9';  ...
'2.4799999999999998E-11' '1.6E-8';'4.1329999999999995E-11' '9.48E-8';'6.199E-11' '4.13E-7'; '8.265999999999999E-11' '7.04E-7';  ...
'8.856E-11' '9.36E-7'; '9.537E-11' '4.95E-7';'1.033E-10' '6.63E-7';'1.24E-10' '1.21E-6';'1.5499999999999998E-10' '2.64E-6';  ...
'2.066E-10' '7.69E-6';'3.0999999999999996E-10' '2.93E-5'; '3.542E-10' '4.5E-5';'3.757E-10' '5.23E-5';'3.8749999999999955E-10' '5.42E-5';  ...
'3.999E-10' '5.18E-5';'4.133E-10' '5.57E-5'; '4.4279999999999996E-10' '6.92E-5';'4.769E-10' '6.67E-5'; '4.9589999999999999E-10' '7.39E-5';  ...
'5.0609999999999999E-10' '6.45E-5';'5.166E-10' '3.34E-5';'5.636E-10' '4.54E-5';'6.199E-10' '6.27E-5'; '6.888E-10' '8.82E-5';  ...
'7.749E-10' '1.29E-4';'8.8559999999999999E-10' '1.98E-4'; '9.537E-10' '2.56E-4';'1.033E-9' '3.29E-4'; '1.127E-9' '4.34E-4';  ...
'1.2399999999999999E-9' '5.79E-4';'1.305E-9' '7.01E-4'; '1.378E-9' '8.39E-4';'1.459E-9' '9.73E-4';'1.5499999999999998E-9' '0.00112';  ...
'1.653E-9' '0.00126'; '1.771E-9' '0.00145'; '1.907E-9' '0.00164';'2.066E-9' '0.00188';'2.254E-9' '0.00214';'2.4799999999999997E-9' '0.00245';  ...
'2.583E-9' '0.00259'; '2.695E-9' '0.00276';'2.818E-9' '0.00294';'2.952E-9' '0.00314';'3.0999999999999996E-9' '0.00335';'3.179E-9' '0.00345';  ...
'3.2629999999999996E-9' '0.00355';'3.3509999999999997E-9' '0.00365';'3.4439999999999997E-9' '0.00376';'3.542E-9' '0.00387';  ...
'3.647E-9' '0.00398';'3.757E-9' '0.00405';'3.875E-9' '0.00411';'3.999E-9' '0.00416';'4.1330000000000005E-9' '0.00418';  ...
'4.275E-9' '0.00417';'4.4279999999999999E-9' '0.00412';'4.592E-9' '0.00403';'4.7689999999999996E-9' '0.00364';'4.959E-9' '0.00315';  ...
'5.166E-9' '0.00293';'5.391E-9' '0.00285';'5.636E-9' '0.00279';'5.904E-9' '0.00279';'6.1989999999999999E-9' '0.00283';  ...
'6.5250000000000005E-9' '0.00299';'6.8879999999999999E-9' '0.00331';'7.2929999999999995E-9' '0.00381';  ...
'7.748999999999999E-9' '0.00448'; '8.266000000000001E-9' '0.00603';'8.551E-9' '0.00726';'8.855999999999998E-9' '0.00867';  ...
'9.184E-9' '0.0104';'9.537E-9' '0.0122';'9.919000000000001E-9' '0.0143';'1.033E-8' '0.017';'1.078E-8' '0.0205';'1.127E-8' '0.0251';  ...
'1.181E-8' '0.03';'1.2399999999999999E-8' '0.0359';'1.3050000000000001E-8' '0.0437';'1.378E-8' '0.0536';'1.459E-8' '0.0671';  ...
'1.55E-8' '0.0807';'1.653E-8' '0.0985';'1.7709999999999998E-8' '0.12';'1.9069999999999998E-8' '0.132';'2.066E-8' '0.151';  ...
'2.101E-8' '0.155';'2.138E-8' '0.159';'2.1749999999999996E-8' '0.162';'2.214E-8' '0.165'; '2.254E-8' '0.167';'2.2960000000000002E-8' '0.167';  ...
'2.339E-8' '0.168';'2.384E-8' '0.167';'2.4309999999999997E-8' '0.167';'2.4799999999999997E-8' '0.16'; '2.5829999999999997E-8' '0.152';  ...
'2.695E-8' '0.149';'2.8179999999999998E-8' '0.144';'2.952E-8' '0.142';'3.1E-8' '0.131';'3.2629999999999995E-8' '0.12';  ...
'3.4439999999999996E-8' '0.114';'3.647E-8' '0.109';'3.757E-8' '0.108';'3.875E-8' '0.108';'3.9989999999999996E-8' '0.115';  ...
'4.1329999999999994E-8' '0.127';'4.202999999999999E-8' '0.138'; '4.275E-8' '0.143';'4.3499999999999999E-8' '0.149'; '4.428E-8' '0.172';  ...
'4.5089999999999995E-8' '0.15'; '4.5920000000000004E-8' '0.128'; '4.679E-8' '0.138'; '4.769E-8' '0.146';'4.8619999999999995E-8' '0.152';  ...
'4.959E-8' '0.168'; '5.061E-8' '0.183';'5.1659999999999994E-8' '0.0739';'5.276E-8' '0.0609';'5.391E-8' '0.0622';  ...
'5.6359999999999996E-8' '0.0717';'6.199E-8' '0.103';'6.887999999999999E-8' '0.168';'7.749E-8' '0.293'; '8.856E-8' '0.456';  ...
'1.0329999999999999E-7' '0.681';'1.24E-7' '0.968';'1.55E-7' '1.3';'2.0659999999999998E-7' '1.54';'2.48E-7' '1.52';'3.1E-7' '1.7';  ...
'3.542E-7' '1.85';'4.133E-7' '2.27';'4.959E-7' '2.67';'6.199E-7' '3.1';'8.266E-7' '3.89';'1.24E-6' '4.84';'1.55E-6' '4.52';  ...
'2.0659999999999998E-6' '3.87';'3.1E-6' '2.21';'6.1989999999999994E-6' '1.47'});
model.component('comp1').material('mat5').propertyGroup('RefractiveIndex').func('int2').set('interp', 'piecewisecubic');
model.component('comp1').material('mat5').propertyGroup('RefractiveIndex').func('int2').set('argunit', 'm');
model.component('comp1').material('mat5').propertyGroup('RefractiveIndex').func('int2').set('fununit', '1');
model.component('comp1').material('mat5').propertyGroup('RefractiveIndex').set('n', '');
model.component('comp1').material('mat5').propertyGroup('RefractiveIndex').set('ki', '');
model.component('comp1').material('mat5').propertyGroup('RefractiveIndex').set('n', '');
model.component('comp1').material('mat5').propertyGroup('RefractiveIndex').set('ki', '');
model.component('comp1').material('mat5').propertyGroup('RefractiveIndex').set('n', {'n_interp(c_const/freq)' '0' '0' '0' 'n_interp(c_const/freq)' '0' '0' '
model.component('comp1').material('mat5').propertyGroup('RefractiveIndex').set('ki', {'k_interp(c_const/freq)' '0' '0' '0' 'k_interp(c_const/freq)' '0' '0' '
model.component('comp1').material('mat5').propertyGroup('RefractiveIndex').addInput('frequency');
model.component('comp1').physics('ewfd').prop('EquationForm').set('freq', '1[GHz]');
model.component('comp1').physics('ewfd').prop('EquationForm').set('modeFreq', '1[GHz]');
model.component('comp1').physics('ewfd').prop('MeshControl').set('EnableMeshControl', false);
model.component('comp1').physics('ewfd').feature('port1').set('Pin', '1[mW]');
```

```
model.component('comp1').physics('ewfd').feature('port1').set('beta', 'abs(ewfd.k0)');
model.component('comp1').physics('ewfd').feature('port1').set('E0', {'0'; 'exp(-j*ewfd.k0*z)'; '0'});
model.component('comp1').physics('ewfd').feature('port2').set('beta', 'abs(ewfd.k0)');
model.component('comp1').physics('ewfd').feature('port2').set('E0', {'0'; 'exp(-j*ewfd.k0*z)'; '0'});
model.component('comp1').mesh('mesh1').feature('size').set('hauto', 4);
model.component('comp1').mesh('mesh1').run;
model.study.create('std1');
model.study('std1').create('freq', 'Frequency');
model.sol.create('sol1');
model.sol('sol1').study('std1');
model.sol('sol1').attach('std1');
model.sol('sol1').create('st1', 'StudyStep');
model.sol('sol1').create('v1', 'Variables');
model.sol('sol1').create('s1', 'Stationary');
model.sol('sol1').feature('s1').create('p1', 'Parametric');
model.sol('sol1').feature('s1').create('fc1', 'FullyCoupled');
model.sol('sol1').feature('s1').create('i1', 'Iterative');
model.sol('sol1').feature('s1').feature('i1').create('mg1', 'Multigrid');
model.sol('sol1').feature('s1').feature('i1').feature('mg1').feature('pr').create('sv1', 'SORVector');
model.sol('sol1').feature('s1').feature('i1').feature('mg1').feature('po').create('sv1', 'SORVector');
model.sol('sol1').feature('s1').feature.remove('fcDef');
model.result.numerical.create('int1', 'IntSurface');
model.result.numerical('int1').selection.set([13 29]);
model.result.numerical('int1').set('probetag', 'none');
model.result.create('pg1', 'PlotGroup3D');
model.result.create('pg2', 'PlotGroup3D');
model.result.create('pg3', 'PlotGroup1D');
model.result('pg1').create('mslc1', 'Multislice');
model.result('pg2').create('slc1', 'Slice');
model.result('pg2').feature('slc1').create('def1', 'Deform');
model.result('pg3').create('tblp1', 'Table');
model.study('std1').feature('freq').set('punit', 'Hz');
model.study('std1').feature('freq').set('plist', '1.0e12');
model.sol('sol1').attach('std1');
model.sol('sol1').feature('v1').set('clistctrl', {'p1'});
model.sol('sol1').feature('v1').set('cname', {'freq'});
model.sol('sol1').feature('v1').set('clist', {'1.0e12[Hz]'});
model.sol('sol1').feature('s1').feature('p1').set('pname', {'freq'});
model.sol('sol1').feature('s1').feature('p1').set('plistarr', {'1.0e12'});
model.sol('sol1').feature('s1').feature('p1').set('punit', {'Hz'});
model.sol('sol1').feature('s1').feature('p1').set('pcontinuationmode', 'no');
model.sol('sol1').feature('s1').feature('p1').set('preusesol', 'auto');
model.sol('sol1').feature('s1').feature('i1').label('Suggested Iterative Solver (ewfd)');
model.sol('sol1').feature('s1').feature('i1').set('linsolver', 'bicgstab');
model.sol('sol1').feature('s1').feature('i1').feature('mg1').feature('pr').feature('sv1').set('sorvecdof', {'comp1_E'});
model.sol('sol1').feature('s1').feature('i1').feature('mg1').feature('po').feature('sv1').set('sorvecdof', {'comp1_E'});
model.sol('sol1').runAll;
model.result.numerical('int1').set('table', 'tbl1');
model.result.numerical('int1').set('expr', {'ewfd.normE'});
model.result.numerical('int1').set('unit', {'V*m'});
model.result.numerical('int1').set('descr', {'Electric field norm'});
model.result.numerical('int1').setResult;
model.result('pg1').label('Electric Field (ewfd)');
model.result('pg1').set('frametype', 'spatial');
model.result('pg1').feature('mslc1').set('multiplanezmethod', 'coord');
model.result('pg1').feature('mslc1').set('zcoord', '1050.0');
model.result('pg1').feature('mslc1').set('rangecoloractive', true);
model.result('pg1').feature('mslc1').set('rangecolormin', 41923.82486663746);
model.result('pg1').feature('mslc1').set('rangecolormax', 400000);
model.result('pg1').feature('mslc1').set('smooth', 'internal');
model.result('pg1').feature('mslc1').set('resolution', 'normal');
model.result('pg2').feature('slc1').set('quickplane', 'xy');
model.result('pg2').feature('slc1').set('quickzmethod', 'coord');
model.result('pg2').feature('slc1').set('quickz', '1033.0');
model.result('pg2').feature('slc1').set('rangecoloractive', true);
model.result('pg2').feature('slc1').set('rangecolormin', 9292.73707567522);
model.result('pg2').feature('slc1').set('rangecolormax', 250000);
model.result('pg2').feature('slc1').set('smooth', 'internal');
model.result('pg2').feature('slc1').set('resolution', 'normal');
model.result('pg2').feature('slc1').feature('def1').active(false);
model.result('pg2').feature('slc1').feature('def1').set('expr', {'+ewfd.Ex' '+ewfd.Ey' '+ewfd.Ez'});
model.result('pg2').feature('slc1').feature('def1').set('descr', ' + Electric field');
model.result('pg2').feature('slc1').feature('def1').set('scale', 2.085880595474555E-4);
model.result('pg2').feature('slc1').feature('def1').set('scaleactive', false);
model.result('pg3').set('data', 'none');
model.result('pg3').set('xlabel', 'freq (Hz)');
model.result('pg3').set('ylabel', 'Electric field norm (V*m)');
model.result('pg3').set('xlabelactive', false);
model.result('pg3').set('ylabelactive', false);
model.result('pg3').feature('tblp1').set('linewidth', 3);
model.result('pg3').feature('tblp1').set('linemarker', 'cycle');
model.result('pg3').feature('tblp1').set('markerpos', 'datapoints');
end
```

```
function [] = fBiTa(model)
model.component('comp1').material.create('mat1', 'Common');
model.component('comp1').material.create('mat5', 'Common');
model.component('comp1').material.create('mat6', 'Common');
model.component('comp1').material('mat1').propertyGroup('def').func.create('eta', 'Piecewise');
model.component('comp1').material('mat1').propertyGroup('def').func.create('Cp', 'Piecewise');
model.component('comp1').material('mat1').propertyGroup('def').func.create('rho', 'Analytic');
model.component('comp1').material('mat1').propertyGroup('def').func.create('k', 'Piecewise');
model.component('comp1').material('mat1').propertyGroup('def').func.create('cs', 'Analytic');
model.component('comp1').material('mat1').propertyGroup.create('RefractiveIndex', 'Refractive index');
model.component('comp1').material('mat5').selection.set([3]);
model.component('comp1').material('mat5').propertyGroup.create('RefractiveIndex', 'Refractive index');
model.component('comp1').material('mat5').propertyGroup('RefractiveIndex').func.create('int1', 'Interpolation');
model.component('comp1').material('mat5').propertyGroup('RefractiveIndex').func.create('int2', 'Interpolation');
model.component('comp1').material('mat6').selection.set([4]);
model.component('comp1').material('mat6').propertyGroup.create('RefractiveIndex', 'Refractive index');
model.component('comp1').material('mat6').propertyGroup('RefractiveIndex').func.create('int1', 'Interpolation');
model.component('comp1').material('mat6').propertyGroup('RefractiveIndex').func.create('int2', 'Interpolation');
model.component('comp1').physics.create('ewfd', 'ElectromagneticWavesFrequencyDomain', 'geom1');
model.component('comp1').physics('ewfd').create('sctr1', 'Scattering', 2);
model.component('comp1').physics('ewfd').feature('sctr1').selection.set([1 2 3 4 5 7 8 9 43 44]);
model.component('comp1').physics('ewfd').create('port1', 'Port', 2);
model.component('comp1').physics('ewfd').feature('port1').selection.set([7]);
model.component('comp1').physics('ewfd').create('port2', 'Port', 2);
model.component('comp1').physics('ewfd').feature('port2').selection.set([3]);
model.component('comp1').mesh('mesh1').create('ftet1', 'FreeTet');
model.result.table('tbl1').comments('Surface Integration 1 (ewfd.normE)');
model.result.table('evl3').label('Evaluation 3D');
model.result.table('evl3').comments('Interactive 3D values');
model.component('comp1').view('view1').set('renderwireframe', true);
model.component('comp1').view('view2').axis.set('xmin', -37.6529541015625);
model.component('comp1').view('view2').axis.set('xmax', 2112.7470703125);
model.component('comp1').view('view2').axis.set('ymin', 0);
model.component('comp1').view('view2').axis.set('ymax', 1651.199951171875);
model.component('comp1').material('mat1').label('Air');
model.component('comp1').material('mat1').set('family', 'air');
model.component('comp1').material('mat1').propertyGroup('def').func('eta').set('arg', 'T');
model.component('comp1').material('mat1').propertyGroup('def').func('eta').set('pieces', {'200.0'...
'1600.0' '-8.38278E-7+8.35717342E-8*T^1-7.69429583E-11*T^2+4.6437266E-14*T^3-1.06585607E-17*T^4'});
model.component('comp1').material('mat1').propertyGroup('def').func('Cp').set('arg', 'T');
model.component('comp1').material('mat1').propertyGroup('def').func('Cp').set('pieces', {'200.0'...
'1600.0' '1047.63657-0.372589265*T^1+9.45304214E-4*T^2-6.02409443E-7*T^3+1.2858961E-10*T^4'});
model.component('comp1').material('mat1').propertyGroup('def').func('rho').set('expr', 'pA*0.02897/8.314/T');
model.component('comp1').material('mat1').propertyGroup('def').func('rho').set('args', {'pA' 'T'});
model.component('comp1').material('mat1').propertyGroup('def').func('rho').set('dermethod', 'manual');
model.component('comp1').material('mat1').propertyGroup('def').func('rho').set('argders', {'pA' 'd(pA*0.02897/8.314/T,pA)'; ...
'T' 'd(pA*0.02897/8.314/T,T)'});
model.component('comp1').material('mat1').propertyGroup('def').func('rho').set('plotargs', {'pA' '0' '1'; 'T' '0' '1'});
model.component('comp1').material('mat1').propertyGroup('def').func('k').set('arg', 'T');
model.component('comp1').material('mat1').propertyGroup('def').func('k').set('pieces', {'200.0' '1600.0'...
    '-0.00227583562+1.15480022E-4*T^1-7.90252856E-8*T^2+4.11702505E-11*T^3-7.43864331E-15*T^4'});
model.component('comp1').material('mat1').propertyGroup('def').func('cs').set('expr', 'sqrt(1.4*287*T)');
model.component('comp1').material('mat1').propertyGroup('def').func('cs').set('args', {'T'});
model.component('comp1').material('mat1').propertyGroup('def').func('cs').set('dermethod', 'manual');
model.component('comp1').material('mat1').propertyGroup('def').func('cs').set('argders', {'T' 'd(sqrt(1.4*287*T),T)'});
model.component('comp1').material('mat1').propertyGroup('def').func('cs').set('plotargs', {'T' '0' '1'});
model.component('comp1').material('mat1').propertyGroup('def').set('relpermeability', {'1' '0' '0' '0' '1' '0' '0' '0' '1'});
model.component('comp1').material('mat1').propertyGroup('def').set('relpermittivity', {'1' '0' '0' '0' '1' '0' '0' '0' '1'});
model.component('comp1').material('mat1').propertyGroup('def').set('dynamicviscosity', 'eta(T[1/K])[Pa*s]');
model.component('comp1').material('mat1').propertyGroup('def').set('ratioofspecificheat', '1.4');
model.component('comp1').material('mat1').propertyGroup('def').set('electricconductivity', {'0[S/m]' '0' '0' '0' '0[S/m]' '0' '0' '0' '0[S/m]'});
model.component('comp1').material('mat1').propertyGroup('def').set('heatcapacity', 'Cp(T[1/K])[J/(kg*K)]');
model.component('comp1').material('mat1').propertyGroup('def').set('density', 'rho(pA[1/Pa],T[1/K])[kg/m^3]');
model.component('comp1').material('mat1').propertyGroup('def').set('thermalconductivity', {'k(T[1/K])[W/(m*K)]' '0' '0' '0' 'k(T[1/K])[W/(m*K)]' '0' '0' ...
    '0' 'k(T[1/K])[W/(m*K)]'});
model.component('comp1').material('mat1').propertyGroup('def').set('soundspeed', 'cs(T[1/K])[m/s]');
model.component('comp1').material('mat1').propertyGroup('def').addInput('temperature');
model.component('comp1').material('mat1').propertyGroup('def').addInput('pressure');
model.component('comp1').material('mat1').propertyGroup('RefractiveIndex').set('n', '');
model.component('comp1').material('mat1').propertyGroup('RefractiveIndex').set('ki', '');
model.component('comp1').material('mat1').propertyGroup('RefractiveIndex').set('n', {'1' '0' '0' '0' '1' '0' '0' '0' '1'});
model.component('comp1').material('mat1').propertyGroup('RefractiveIndex').set('ki', {'0' '0' '0' '0' '0' '0' '0' '0' '0'});
model.component('comp1').material('mat5').label(['Ta (Tantalum) (Ordal et al. 1988: n,k 0.667-125 ' native2unicode(hex2dec({'00' 'b5'}), 'unicode') ...
    'm)']);
model.component('comp1').material('mat5').set('groups', {'inorganic' 'Inorganic Materials'; 'ta_tantalum_and_tantalates'...
    'Ta - Tantalum and Tantalates'; 'experimental_data' 'Experimental data'});
model.component('comp1').material('mat5').propertyGroup('RefractiveIndex').func('int1').label('Refractive index');
model.component('comp1').material('mat5').propertyGroup('RefractiveIndex').func('int1').set('funcname', 'n_interp');
model.component('comp1').material('mat5').propertyGroup('RefractiveIndex').func('int1').set('table', {'6.67E-7' '1.4478379';  ...
'7.14E-7' '1.2685432'; '7.69E-7' '1.1466787'; '8.329999999999999E-7' '1.073087'; '9.09E-7' '1.0150307'; '1.0E-6' '0.9821499';  ...
'1.05E-6' '0.96032';'1.11E-6' '0.9253053';'1.18E-6' '0.9012078';'1.2499999999999999E-6' '0.8740663';'1.33E-6' '0.8414308';  ...
'1.4299999999999999E-6' '0.8354757';'1.5399999999999999E-6' '0.8564461';'1.6699999999999999E-6' '0.9019896';  ...
'1.82E-6' '0.960826';'2.0E-6' '1.0326738';'2.1099999999999997E-6' '1.1067824';'2.22E-6' '1.192353'; '2.35E-6' '1.2678805';  ...
'2.4999999999999998E-6' '1.3499202';'2.67E-6' '1.4802727';'2.8599999999999997E-6' '1.6348171';'3.0799999999999997E-6' '1.8209204';  ...
'3.33E-6' '2.0587589';'3.64E-6' '2.3469537';'4.0E-6' '2.6799222';'4.44E-6' '3.1932058';'4.9999999999999996E-6' '3.9020861';  ...
'5.7099999999999995E-6' '4.9008382';'6.67E-6' '6.4122619';'8.0E-6' '8.7691085';'9.999999999999999E-6' '12.931507';  ...
```

```
'1.1099999999999999E-5' '15.61576';'1.2499999999999999E-5' '19.058763';'1.43E-5' '24.23852';'1.67E-5' '31.493393'; ...
'1.9999999999999998E-5' '41.721911';'2.2199999999999998E-5' '48.392026';'2.4999999999999998E-5' '56.42849'; '2.86E-5' '66.1388'; ...
'3.3299999999999996E-5' '77.867703';'3.9999999999999996E-5' '92.016211'; '4.4399999999999995E-5' '100.17673'; ...
'4.9999999999999996E-5' '109.23359';'5.71E-5' '119.43744';'6.67E-5' '131.25365';'7.999999999999999E-5' '145.5884'; ...
'9.999999999999999E-5' '164.07966';'1.25E-4' '184.49663'});
model.component('comp1').material('mat5').propertyGroup('RefractiveIndex').func('int1').set('interp', 'piecewisecubic');
model.component('comp1').material('mat5').propertyGroup('RefractiveIndex').func('int1').set('argunit', 'm');
model.component('comp1').material('mat5').propertyGroup('RefractiveIndex').func('int1').set('fununit', '1');
model.component('comp1').material('mat5').propertyGroup('RefractiveIndex').func('int2').label('Refractive index, imaginary part');
model.component('comp1').material('mat5').propertyGroup('RefractiveIndex').func('int2').set('funcname', 'k_interp');
model.component('comp1').material('mat5').propertyGroup('RefractiveIndex').func('int2').set('table', {'6.67E-7' '2.3306224'; ...
'7.14E-7' '2.7919609';'7.69E-7' '3.2384897';'8.329999999999999E-7' '3.7362305';'9.09E-7' '4.2736932'; '1.0E-6' '4.8873028'; ...
'1.05E-6' '5.2167868';'1.11E-6' '5.5894492';'1.18E-6' '6.0121558';'1.2499999999999999E-6' '6.4776281';'1.33E-6' '7.0171487'; ...
'1.4299999999999999E-6' '7.6548582';'1.5399999999999999E-6' '8.3686838';'1.6699999999999999E-6' '9.1771344'; '1.82E-6' '10.114648'; ...
'2.0E-6' '11.243595';'2.1099999999999997E-6' '11.896174';'2.22E-6' '12.595572';'2.35E-6' '13.363144';'2.4999999999999998E-6' '14.23958'; ...
'2.67E-6' '15.240124';'2.8599999999999997E-6' '16.359682';'3.07999999999997E-6' '17.64005';'3.33E-6' '19.120563';'3.64E-6' '20.845754'; ...
'4.0E-6' '22.915456';'4.44E-6' '25.461819';'4.999999999999996E-6' '28.57704';'5.709999999999995E-6' '32.510917';'6.67E-6' '37.643719'; ...
'8.0E-6' '44.613332';'9.999999999999999E-6' '54.673721';'1.1099999999999999E-5' '59.926459';'1.2499999999999999E-5' '66.350003'; ...
'1.43E-5' '73.949845';'1.67E-5' '82.79423';'1.9999999999999998E-5' '93.122866';'2.2199999999999998E-5' '98.921392'; ...
'2.4999999999999998E-5' '105.16669';'2.86E-5' '111.86857';'3.3299999999999996E-5' '119.08288';'3.9999999999999996E-5' '127.08528'; ...
'4.4399999999999995E-5' '131.63028';'4.9999999999999996E-5' '136.82885';'5.71E-5' '143.08774';'6.67E-5' '151.08524'; ...
'7.999999999999999E-5' '161.9815';'9.999999999999999E-5' '177.86128';'1.25E-4' '196.87085'});
model.component('comp1').material('mat5').propertyGroup('RefractiveIndex').func('int2').set('interp', 'piecewisecubic');
model.component('comp1').material('mat5').propertyGroup('RefractiveIndex').func('int2').set('argunit', 'm');
model.component('comp1').material('mat5').propertyGroup('RefractiveIndex').func('int2').set('fununit', '1');
model.component('comp1').material('mat5').propertyGroup('RefractiveIndex').set('n', '');
model.component('comp1').material('mat5').propertyGroup('RefractiveIndex').set('ki', '');
model.component('comp1').material('mat5').propertyGroup('RefractiveIndex').set('n', '');
model.component('comp1').material('mat5').propertyGroup('RefractiveIndex').set('ki', '');
model.component('comp1').material('mat5').propertyGroup('RefractiveIndex').set('n', {'n_interp(c_const/freq)' '0' '0' '0' 'n_interp(c_const/freq)' '0' ...
   '0' '0' 'n_interp(c_const/freq)'});
model.component('comp1').material('mat5').propertyGroup('RefractiveIndex').set('ki', {'k_interp(c_const/freq)' '0' '0' '0' 'k_interp(c_const/freq)' '0' ...
   '0' '0' 'k_interp(c_const/freq)'});
model.component('comp1').material('mat5').propertyGroup('RefractiveIndex').addInput('frequency');
model.component('comp1').material('mat6').label(['Bi (Bismuth) (Hagemann et al. 1974: n,k 0.00155-6.199 ' native2unicode(hex2dec({'00' 'b5'}), ...
   'unicode') 'm)']);
model.component('comp1').material('mat6').set('groups', {'inorganic' 'Inorganic Materials'; 'bi_bismuth' 'Bi - Bismuth'; 'experimental_data' ...
   'Experimental data'});
model.component('comp1').material('mat6').propertyGroup('RefractiveIndex').func('int1').label('Refractive index');
model.component('comp1').material('mat6').propertyGroup('RefractiveIndex').func('int1').set('funcname', 'n_interp');
model.component('comp1').material('mat6').propertyGroup('RefractiveIndex').func('int1').set('table', {'2.48E-12' '1.0'; ...
'4.133E-12' '1.0';'1.2399999999999999E-11' '1.0';'1.378E-11' '1.0';'1.4589999999999998E-11' '1.0';'2.4799999999999998E-11' '1.0'; ...
'4.1329999999999995E-11' '1.0';'6.199E-11' '1.0';'8.265999999999999E-11' '1.0';'8.856E-11' '1.0';'9.537E-11' '1.0';'1.033E-10' '1.0'; ...
'1.24E-10' '1.0';'1.5499999999999998E-10' '1.0';'2.066E-10' '1.0';'3.0999999999999996E-10' '1.0';'3.542E-10' '1.0';'3.757E-10' '1.0'; ...
'3.8749999999999995E-10' '1.0';'3.999E-10' '1.0';'4.133E-10' '1.0';'4.4279999999999996E-10' '1.0';'4.769E-10' '1.0'; ...
'4.9589999999999999E-10' '1.0';'5.060999999999999E-10' '1.0';'5.166E-10' '1.0';'5.636E-10' '1.0';'6.199E-10' '1.0';'6.888E-10' '1.0'; ...
'7.749E-10' '1.0';'8.855999999999999E-10' '0.999';'9.537E-10' '0.999';'1.033E-9' '0.999';'1.127E-9' '0.999';'1.2399999999999999E-9' '0.999'; ...
'1.305E-9' '0.999';'1.378E-9' '0.999';'1.459E-9' '0.999';'1.5499999999999998E-9' '0.999';'1.653E-9' '0.998';'1.771E-9' '0.998';'1.907E-9' '0.998'; ...
'2.066E-9' '0.998';'2.254E-9' '0.998';'2.4799999999999997E-9' '0.998';'2.583E-9' '0.998';'2.695E-9' '0.997';'2.818E-9' '0.997'; ...
'2.952E-9' '0.997';'3.0999999999999996E-9' '0.997';'3.179E-9' '0.997';'3.2629999999999996E-9' '0.997';'3.3509999999999997E-9' '0.997'; ...
'3.4439999999999997E-9' '0.997';'3.542E-9' '0.997';'3.647E-9' '0.997';'3.757E-9' '0.997';'3.875E-9' '0.997';'3.999E-9' '0.997'; ...
'4.1330000000000005E-9' '0.997';'4.275E-9' '0.997';'4.4279999999999999E-9' '0.996';'4.592E-9' '0.996';'4.7689999999999996E-9' '0.996'; ...
'4.959E-9' '0.996';'5.166E-9' '0.995';'5.391E-9' '0.994';'5.636E-9' '0.993';'5.904E-9' '0.991';'6.1989999999999999E-9' '0.99'; ...
'6.5250000000000005E-9' '0.988';'6.887999999999999E-9' '0.986';'7.2929999999999995E-9' '0.984';'7.748999999999999E-9' '0.98'; ...
'8.266000000000001E-9' '0.976';'8.551E-9' '0.974';'8.855999999999998E-9' '0.972';'9.184E-9' '0.969';'9.537E-9' '0.966'; ...
'9.919000000000001E-9' '0.964';'1.033E-8' '0.96';'1.078E-8' '0.956';'1.127E-8' '0.953';'1.181E-8' '0.949';'1.2399999999999999E-8' '0.945'; ...
'1.3050000000000001E-8' '0.94';'1.378E-8' '0.935';'1.459E-8' '0.933';'1.55E-8' '0.932';'1.653E-8' '0.933';'1.7709999999999998E-8' '0.942'; ...
'1.9069999999999998E-8' '0.955';'2.066E-8' '0.966';'2.101E-8' '0.97';'2.138E-8' '0.975';'2.1749999999999996E-8' '0.981'; ...
'2.214E-8' '0.987';'2.254E-8' '0.993';'2.2960000000000002E-8' '1.0';'2.339E-8' '1.007';'2.384E-8' '1.014';'2.4309999999999997E-8' '1.022'; ...
'2.4799999999999997E-8' '1.027';'2.5829999999999997E-8' '1.033';'2.695E-8' '1.036';'2.8179999999999998E-8' '1.041';'2.952E-8' '1.048'; ...
'3.1E-8' '1.051';'3.2629999999999995E-8' '1.046';'3.4439999999999996E-8' '1.038';'3.647E-8' '1.027';'3.757E-8' '1.02';'3.875E-8' '1.009'; ...
'3.9989999999999996E-8' '0.997';'4.1329999999999994E-8' '0.987';'4.2029999999999999E-8' '0.988';'4.275E-8' '0.991'; ...
'4.3499999999999998E-8' '0.987';'4.428E-8' '0.998';'4.5089999999999995E-8' '1.035';'4.5920000000000004E-8' '1.011'; ...
'4.679E-8' '0.998';'4.769E-8' '0.999';'4.8619999999999995E-8' '1.001';'4.959E-8' '1.004';'5.061E-8' '1.07';'5.1659999999999994E-8' '1.065'; ...
'5.276E-8' '1.013';'5.391E-8' '0.981';'5.6359999999999996E-8' '0.936';'6.199E-8' '0.856';'6.8879999999999998E-8' '0.763';'7.749E-8' '0.692';
'3.1E-7' '1.517';'3.542E-7' '1.541';'4.133E-7' '1.556';'4.959E-7' '1.865';'6.199E-7' '2.244';'8.266E-7' '2.861';'1.24E-6' '5.036'; ...
'1.55E-6' '6.247';'2.0659999999999998E-6' '7.466';'3.1E-6' '8.572';'6.1989999999999994E-6' '6.908'});
model.component('comp1').material('mat6').propertyGroup('RefractiveIndex').func('int1').set('interp', 'piecewisecubic');
model.component('comp1').material('mat6').propertyGroup('RefractiveIndex').func('int1').set('argunit', 'm');
model.component('comp1').material('mat6').propertyGroup('RefractiveIndex').func('int1').set('fununit', '1');
model.component('comp1').material('mat6').propertyGroup('RefractiveIndex').func('int2').label('Refractive index, imaginary part');
model.component('comp1').material('mat6').propertyGroup('RefractiveIndex').func('int2').set('funcname', 'k_interp');
model.component('comp1').material('mat6').propertyGroup('RefractiveIndex').func('int2').set('table', {'2.48E-12' '2.02E-11'; ...
'4.133E-12' '1.19E-10';'1.2399999999999999E-11' '5.71E-9';'1.378E-11' '7.13E-9';'1.4589999999999998E-11' '2.56E-9'; ...
'2.4799999999999998E-11' '1.6E-8';'4.1329999999999995E-11' '9.48E-8';'6.199E-11' '4.13E-7';'8.2659999999999999E-11' '7.04E-7'; ...
'8.856E-11' '9.36E-7';'9.537E-11' '4.95E-7';'1.033E-10' '6.63E-7';'1.24E-10' '1.21E-6';'1.54999999999998E-10' '2.64E-6';'2.066E-10' '7.69E-6'; ...
'3.0999999999999996E-10' '2.93E-5';'3.542E-10' '4.5E-5';'3.757E-10' '5.23E-5';'3.8749999999999995E-10' '5.42E-5';'3.999E-10' '5.18E-5'; ...
'4.133E-10' '5.57E-5';'4.4279999999999996E-10' '6.92E-5';'4.769E-10' '6.67E-5';'4.9589999999999999E-10' '7.39E-5'; ...
'5.0609999999999999E-10' '6.45E-5';'5.166E-10' '3.34E-5'; '5.636E-10' '4.54E-5'; '6.199E-10' '6.27E-5'; '6.888E-10' '8.82E-5'; ...
'7.749E-10' '1.29E-4';'8.855999999999999E-10' '1.98E-4';'9.537E-10' '2.56E-4';'1.033E-9' '3.29E-4';'1.127E-9' '4.34E-4'; ...
'1.2399999999999999E-9' '5.79E-4';'1.305E-9' '7.01E-4';'1.378E-9' '8.39E-4';'1.459E-9' '9.73E-4';'1.5499999999999998E-9' '0.00112'; ...
'1.653E-9' '0.00126';'1.771E-9' '0.00145';'1.907E-9' '0.00164';'2.066E-9' '0.00188';'2.254E-9' '0.00214';'2.4799999999999997E-9' '0.00245'; ...
'2.583E-9' '0.00259';'2.695E-9' '0.00276';'2.818E-9' '0.00294';'2.952E-9' '0.00314';'3.0999999999999996E-9' '0.00335';'3.179E-9' '0.00345'; ...
'3.2629999999999996E-9' '0.00355';'3.3509999999999997E-9' '0.00365';'3.4439999999999997E-9' '0.00376';'3.542E-9' '0.00387'; ...
'3.647E-9' '0.00398';'3.757E-9' '0.00405';'3.875E-9' '0.00411';'3.999E-9' '0.00416';'4.1330000000000005E-9' '0.00418';'4.275E-9' '0.00417'; ...
'4.4279999999999999E-9' '0.00412';'4.592E-9' '0.00403';'4.7689999999999996E-9' '0.00364';'4.959E-9' '0.00315'; '5.166E-9' '0.00293'; ...
'5.391E-9' '0.00285';'5.636E-9' '0.00279';'5.904E-9' '0.00279'; '6.1989999999999999E-9' '0.00283';'6.5250000000000005E-9' '0.00299'; ...
```

```
'6.8879999999999999E-9' '0.00331';'7.2929999999999995E-9' '0.00381';'7.7489999999999999E-9' '0.00448'; ...
'8.266000000000001E-9' '0.00603';'8.551E-9' '0.00726';'8.855999999999998E-9' '0.00867';'9.184E-9' '0.0104';'9.537E-9' '0.0122'; ...
'9.919000000000001E-9' '0.0143';'1.033E-8' '0.017';'1.078E-8' '0.0205';'1.127E-8' '0.0251';'1.181E-8' '0.03';'1.2399999999999999E-8' '0.0359'; ...
'1.3050000000000001E-8' '0.0437';'1.378E-8' '0.0536';'1.459E-8' '0.0671';'1.55E-8' '0.0807';'1.653E-8' '0.0985';'1.770999999999998E-8' '0.12'; ...
'1.9069999999999998E-8' '0.132';'2.066E-8' '0.151';'2.101E-8' '0.155';'2.138E-8' '0.159';'2.1749999999999996E-8' '0.162'; '2.214E-8' '0.165'; ...
'2.254E-8' '0.167';'2.2960000000000002E-8' '0.167';'2.339E-8' '0.168';'2.384E-8' '0.167'; '2.4309999999999997E-8' '0.167'; ...
'2.4799999999999997E-8' '0.16';'2.5829999999999997E-8' '0.152';'2.695E-8' '0.149';'2.8179999999999998E-8' '0.144';'2.952E-8' '0.142'; ...
'3.1E-8' '0.131';'3.2629999999999995E-8' '0.12';'3.4439999999999996E-8' '0.114';'3.647E-8' '0.109';'3.757E-8' '0.108';'3.875E-8' '0.108'; ...
'3.9989999999999996E-8' '0.115';'4.1329999999999994E-8' '0.127';'4.202999999999999E-8' '0.138';'4.275E-8' '0.143'; ...
'4.349999999999999E-8' '0.149';'4.428E-8' '0.172';'4.5089999999999995E-8' '0.15';'4.5920000000000004E-8' '0.128'; '4.679E-8' '0.138'; ...
'4.769E-8' '0.146';'4.8619999999999995E-8' '0.152';'4.959E-8' '0.168';'5.061E-8' '0.183';'5.1659999999999994E-8' '0.0739'; ...
'5.276E-8' '0.0609';'5.391E-8' '0.0622';'5.6359999999999996E-8' '0.0717';'6.199E-8' '0.103';'6.887999999999999E-8' '0.168'; ...
'7.749E-8' '0.293';'8.856E-8' '0.456';'1.0329999999999999E-7' '0.681';'1.24E-7' '0.968';'1.55E-7' '1.3';'2.0659999999999998E-7' '1.54'; ...
'2.48E-7' '1.52';'3.1E-7' '1.7';'3.542E-7' '1.85';'4.133E-7' '2.27';'4.959E-7' '2.67';'6.199E-7' '3.1';'8.266E-7' '3.89'; '1.24E-6' '4.84'; ...
'1.55E-6' '4.52';'2.0659999999999998E-6' '3.87';'3.1E-6' '2.21';'6.1989999999999994E-6' '1.47'});
model.component('comp1').material('mat6').propertyGroup('RefractiveIndex').func('int2').set('interp', 'piecewisecubic');
model.component('comp1').material('mat6').propertyGroup('RefractiveIndex').func('int2').set('argunit', 'm');
model.component('comp1').material('mat6').propertyGroup('RefractiveIndex').func('int2').set('fununit', '1');
model.component('comp1').material('mat6').propertyGroup('RefractiveIndex').set('n', '');
model.component('comp1').material('mat6').propertyGroup('RefractiveIndex').set('ki', '');
model.component('comp1').material('mat6').propertyGroup('RefractiveIndex').set('n', '');
model.component('comp1').material('mat6').propertyGroup('RefractiveIndex').set('ki', '');
model.component('comp1').material('mat6').propertyGroup('RefractiveIndex').set('n', {'n_interp(c_const/freq)' '0' '0' '0' 'n_interp(c_const/freq)' '0' ...
    '0' '0' 'n_interp(c_const/freq)'});
model.component('comp1').material('mat6').propertyGroup('RefractiveIndex').set('ki', {'k_interp(c_const/freq)' '0' '0' '0' 'k_interp(c_const/freq)' '0'...
    '0' '0' 'k_interp(c_const/freq)'});
model.component('comp1').material('mat6').propertyGroup('RefractiveIndex').addInput('frequency');
model.component('comp1').physics('ewfd').prop('EquationForm').set('freq', '1[GHz]');
model.component('comp1').physics('ewfd').prop('EquationForm').set('modeFreq', '1[GHz]');
model.component('comp1').physics('ewfd').prop('MeshControl').set('EnableMeshControl', false);
model.component('comp1').physics('ewfd').feature('port1').set('Pin', '1[mW]');
model.component('comp1').physics('ewfd').feature('port1').set('beta', 'abs(ewfd.k0)');
model.component('comp1').physics('ewfd').feature('port1').set('E0', {'0'; 'exp(-j*ewfd.k0*z)'; '0'});
model.component('comp1').physics('ewfd').feature('port2').set('beta', 'abs(ewfd.k0)');
model.component('comp1').physics('ewfd').feature('port2').set('E0', {'0'; 'exp(-j*ewfd.k0*z)'; '0'});
model.component('comp1').mesh('mesh1').feature('size').set('hauto', 4);
model.component('comp1').mesh('mesh1').run;
model.study.create('std1');
model.study('std1').create('freq', 'Frequency');
model.sol.create('sol1');
model.sol('sol1').study('std1');
model.sol('sol1').attach('std1');
model.sol('sol1').create('st1', 'StudyStep');
model.sol('sol1').create('v1', 'Variables');
model.sol('sol1').create('s1', 'Stationary');
model.sol('sol1').feature('s1').create('p1', 'Parametric');
model.sol('sol1').feature('s1').create('fc1', 'FullyCoupled');
model.sol('sol1').feature('s1').create('i1', 'Iterative');
model.sol('sol1').feature('s1').feature('i1').create('mg1', 'Multigrid');
model.sol('sol1').feature('s1').feature('i1').feature('mg1').feature('pr').create('sv1', 'SORVector');
model.sol('sol1').feature('s1').feature('i1').feature('mg1').feature('po').create('sv1', 'SORVector');
model.sol('sol1').feature('s1').feature.remove('fcDef');
model.result.numerical.create('int1', 'IntSurface');
model.result.numerical('int1').selection.set([13 29]);
model.result.numerical('int1').set('probetag', 'none');
model.result.create('pg1', 'PlotGroup3D');
model.result.create('pg2', 'PlotGroup3D');
model.result.create('pg3', 'PlotGroup1D');
model.result('pg1').create('mslc1', 'Multislice');
model.result('pg2').create('slc1', 'Slice');
model.result('pg2').feature('slc1').create('def1', 'Deform');
model.result('pg3').create('tblp1', 'Table');
model.study('std1').feature('freq').set('punit', 'Hz');
model.study('std1').feature('freq').set('plist', '1.0e12');
model.sol('sol1').attach('std1');
model.sol('sol1').feature('v1').set('clistctrl', {'p1'});
model.sol('sol1').feature('v1').set('cname', {'freq'});
model.sol('sol1').feature('v1').set('clist', {'1.0e12[Hz]'});
model.sol('sol1').feature('s1').feature('p1').set('pname', {'freq'});
model.sol('sol1').feature('s1').feature('p1').set('plistarr', {'1.0e12'});
model.sol('sol1').feature('s1').feature('p1').set('punit', {'Hz'});
model.sol('sol1').feature('s1').feature('p1').set('pcontinuationmode', 'no');
model.sol('sol1').feature('s1').feature('p1').set('preusesol', 'auto');
model.sol('sol1').feature('s1').feature('i1').label('Suggested Iterative Solver (ewfd)');
model.sol('sol1').feature('s1').feature('i1').set('linsolver', 'bicgstab');
model.sol('sol1').feature('s1').feature('i1').feature('mg1').feature('pr').feature('sv1').set('sorvecdof', {'comp1_E'});
model.sol('sol1').feature('s1').feature('i1').feature('mg1').feature('po').feature('sv1').set('sorvecdof', {'comp1_E'});
model.sol('sol1').runAll;
model.result.numerical('int1').set('table', 'tbl1');
model.result.numerical('int1').set('expr', {'ewfd.normE'});
model.result.numerical('int1').set('unit', {'V*m'});
model.result.numerical('int1').set('descr', {'Electric field norm'});
model.result.numerical('int1').setResult;
model.result('pg1').label('Electric Field (ewfd)');
model.result('pg1').set('frametype', 'spatial');
model.result('pg1').feature('mslc1').set('multiplanezmethod', 'coord');
```

```
model.result('pg1').feature('mslc1').set('zcoord', '1050.0');
model.result('pg1').feature('mslc1').set('rangecoloractive', true);
model.result('pg1').feature('mslc1').set('rangecolormin', 41923.82486663746);
model.result('pg1').feature('mslc1').set('rangecolormax', 400000);
model.result('pg1').feature('mslc1').set('smooth', 'internal');
model.result('pg1').feature('mslc1').set('resolution', 'normal');
model.result('pg2').feature('slc1').set('quickplane', 'xy');
model.result('pg2').feature('slc1').set('quickzmethod', 'coord');
model.result('pg2').feature('slc1').set('quickz', '1033.0');
model.result('pg2').feature('slc1').set('rangecoloractive', true);
model.result('pg2').feature('slc1').set('rangecolormin', 9292.73707567522);
model.result('pg2').feature('slc1').set('rangecolormax', 250000);
model.result('pg2').feature('slc1').set('smooth', 'internal');
model.result('pg2').feature('slc1').set('resolution', 'normal');
model.result('pg2').feature('slc1').feature('def1').active(false);
model.result('pg2').feature('slc1').feature('def1').set('expr', {'+ewfd.Ex' '+ewfd.Ey' '+ewfd.Ez'});
model.result('pg2').feature('slc1').feature('def1').set('descr', ' + Electric field');
model.result('pg2').feature('slc1').feature('def1').set('scale', 2.085880595474555E-4);
model.result('pg2').feature('slc1').feature('def1').set('scaleactive', false);
model.result('pg3').set('data', 'none');
model.result('pg3').set('xlabel', 'freq (Hz)');
model.result('pg3').set('ylabel', 'Electric field norm (V*m)');
model.result('pg3').set('xlabelactive', false);
model.result('pg3').set('ylabelactive', false);
model.result('pg3').feature('tblp1').set('linewidth', 3);
model.result('pg3').feature('tblp1').set('linemarker', 'cycle');
model.result('pg3').feature('tblp1').set('markerpos', 'datapoints');
end
```

```
function [] = fBiW(model)
model.component('comp1').material.create('mat1', 'Common');
model.component('comp1').material.create('mat5', 'Common');
model.component('comp1').material.create('mat6', 'Common');
model.component('comp1').material('mat1').propertyGroup('def').func.create('eta', 'Piecewise');
model.component('comp1').material('mat1').propertyGroup('def').func.create('Cp', 'Piecewise');
model.component('comp1').material('mat1').propertyGroup('def').func.create('rho', 'Analytic');
model.component('comp1').material('mat1').propertyGroup('def').func.create('k', 'Piecewise');
model.component('comp1').material('mat1').propertyGroup('def').func.create('cs', 'Analytic');
model.component('comp1').material('mat1').propertyGroup.create('RefractiveIndex', 'Refractive index');
model.component('comp1').material('mat5').selection.set([3]);
model.component('comp1').material('mat5').propertyGroup.create('RefractiveIndex', 'Refractive index');
model.component('comp1').material('mat5').propertyGroup('RefractiveIndex').func.create('int1', 'Interpolation');
model.component('comp1').material('mat5').propertyGroup('RefractiveIndex').func.create('int2', 'Interpolation');
model.component('comp1').material('mat6').selection.set([4]);
model.component('comp1').material('mat6').propertyGroup.create('RefractiveIndex', 'Refractive index');
model.component('comp1').material('mat6').propertyGroup('RefractiveIndex').func.create('int1', 'Interpolation');
model.component('comp1').material('mat6').propertyGroup('RefractiveIndex').func.create('int2', 'Interpolation');
model.component('comp1').physics.create('ewfd', 'ElectromagneticWavesFrequencyDomain', 'geom1');
model.component('comp1').physics('ewfd').create('sctr1', 'Scattering', 2);
model.component('comp1').physics('ewfd').feature('sctr1').selection.set([1 2 3 4 5 7 8 9 43 44]);
model.component('comp1').physics('ewfd').create('port1', 'Port', 2);
model.component('comp1').physics('ewfd').feature('port1').selection.set([7]);
model.component('comp1').physics('ewfd').create('port2', 'Port', 2);
model.component('comp1').physics('ewfd').feature('port2').selection.set([3]);
model.component('comp1').mesh('mesh1').create('ftet1', 'FreeTet');
model.result.table('tbl1').comments('Surface Integration 1 (ewfd.normE)');
model.result.table('evl3').label('Evaluation 3D');
model.result.table('evl3').comments('Interactive 3D values');
model.component('comp1').view('view1').set('renderwireframe', true);
model.component('comp1').view('view2').axis.set('xmin', -37.6529541015625);
model.component('comp1').view('view2').axis.set('xmax', 2112.7470703125);
model.component('comp1').view('view2').axis.set('ymin', 0);
model.component('comp1').view('view2').axis.set('ymax', 1651.199951171875);
model.component('comp1').material('mat1').label('Air');
model.component('comp1').material('mat1').set('family', 'air');
model.component('comp1').material('mat1').propertyGroup('def').func('eta').set('arg', 'T');
model.component('comp1').material('mat1').propertyGroup('def').func('eta').set('pieces', {'200.0' '1600.0' ...
    '-8.38278E-7+8.35717342E-8*T^1-7.69429583E-11*T^2+4.6437266E-14*T^3-1.06585607E-17*T^4'});
model.component('comp1').material('mat1').propertyGroup('def').func('Cp').set('arg', 'T');
model.component('comp1').material('mat1').propertyGroup('def').func('Cp').set('pieces', {'200.0' '1600.0' ...
    '1047.63657-0.372589265*T^1+9.45304214E-4*T^2-6.02409443E-7*T^3+1.2858961E-10*T^4'});
model.component('comp1').material('mat1').propertyGroup('def').func('rho').set('expr', 'pA*0.02897/8.314/T');
model.component('comp1').material('mat1').propertyGroup('def').func('rho').set('args', {'pA' 'T'});
model.component('comp1').material('mat1').propertyGroup('def').func('rho').set('dermethod', 'manual');
model.component('comp1').material('mat1').propertyGroup('def').func('rho').set('argders', {'pA' 'd(pA*0.02897/8.314/T,pA)'; 'T' ...
```

```
    'd(pA*0.02897/8.314/T,T)'});
model.component('comp1').material('mat1').propertyGroup('def').func('rho').set('plotargs', {'pA' '0' '1'; 'T' '0' '1'});
model.component('comp1').material('mat1').propertyGroup('def').func('k').set('arg', 'T');
model.component('comp1').material('mat1').propertyGroup('def').func('k').set('pieces', {'200.0' '1600.0' ...
    '-0.00227583562+1.15480022E-4*T^1-7.90252856E-8*T^2+4.11702505E-11*T^3-7.43864331E-15*T^4'});
model.component('comp1').material('mat1').propertyGroup('def').func('cs').set('expr', 'sqrt(1.4*287*T)');
model.component('comp1').material('mat1').propertyGroup('def').func('cs').set('args', {'T'});
model.component('comp1').material('mat1').propertyGroup('def').func('cs').set('dermethod', 'manual');
model.component('comp1').material('mat1').propertyGroup('def').func('cs').set('argders', {'T' 'd(sqrt(1.4*287*T),T)'});
model.component('comp1').material('mat1').propertyGroup('def').func('cs').set('plotargs', {'T' '0' '1'});
model.component('comp1').material('mat1').propertyGroup('def').set('relpermeability', {'1' '0' '0' '0' '1' '0' '0' '0' '1'});
model.component('comp1').material('mat1').propertyGroup('def').set('relpermittivity', {'1' '0' '0' '0' '1' '0' '0' '0' '1'});
model.component('comp1').material('mat1').propertyGroup('def').set('dynamicviscosity', 'eta(T[1/K])[Pa*s]');
model.component('comp1').material('mat1').propertyGroup('def').set('ratioofspecificheat', '1.4');
model.component('comp1').material('mat1').propertyGroup('def').set('electricconductivity', {'0[S/m]' '0' '0' '0' '0[S/m]' '0' '0' '0' '0[S/m]'});
model.component('comp1').material('mat1').propertyGroup('def').set('heatcapacity', 'Cp(T[1/K])[J/(kg*K)]');
model.component('comp1').material('mat1').propertyGroup('def').set('density', 'rho(pA[1/Pa],T[1/K])[kg/m^3]');
model.component('comp1').material('mat1').propertyGroup('def').set('thermalconductivity', {'k(T[1/K])[W/(m*K)]' '0' '0' '0' 'k(T[1/K])[W/(m*K)]' '0' '0' ...
    '0' 'k(T[1/K])[W/(m*K)]'});
model.component('comp1').material('mat1').propertyGroup('def').set('soundspeed', 'cs(T[1/K])[m/s]');
model.component('comp1').material('mat1').propertyGroup('def').addInput('temperature');
model.component('comp1').material('mat1').propertyGroup('def').addInput('pressure');
model.component('comp1').material('mat1').propertyGroup('RefractiveIndex').set('n', '');
model.component('comp1').material('mat1').propertyGroup('RefractiveIndex').set('ki', '');
model.component('comp1').material('mat1').propertyGroup('RefractiveIndex').set('n', {'1' '0' '0' '0' '1' '0' '0' '0' '1'});
model.component('comp1').material('mat1').propertyGroup('RefractiveIndex').set('ki', {'0' '0' '0' '0' '0' '0' '0' '0' '0'});
model.component('comp1').material('mat5').label(['W (Tungsten) (Ordal et al. 1988: n,k 0.667-200 ' native2unicode(hex2dec({'00' 'b5'}), ...
    'unicode') 'm)']);
model.component('comp1').material('mat5').set('groups', {'inorganic' 'Inorganic Materials'; 'w_tungsten_and_tungstates' ...
    'W - Tungsten and tungstates'; 'experimental_data' 'Experimental data'});
model.component('comp1').material('mat5').propertyGroup('RefractiveIndex').func('int1').label('Refractive index');
model.component('comp1').material('mat5').propertyGroup('RefractiveIndex').func('int1').set('funcname', 'n_interp');
model.component('comp1').material('mat5').propertyGroup('RefractiveIndex').func('int1').set('table', {'6.67E-7' '3.8312601';'7.14E-7' '3.9313491';  ...
'7.69E-7' '3.778336';'8.3299999999999997E-7' '3.5190631';'9.09E-7' '3.2814572';'1.0E-6' '3.0826871';'1.05E-6' '3.0570934';  ...
'1.11E-6' '3.0821649';'1.18E-6' '3.1599449';'1.2499999999999999E-6' '3.1960757';'1.33E-6' '3.1990778';'1.4299999999999999E-6' '3.0278249';  ...
'1.5399999999999999E-6' '2.2587782';'1.6699999999999999E-6' '1.8828191';'1.82E-6' '1.5455798';'2.0E-6' '1.2992808';  ...
'2.1099999999999997E-6' '1.1946455';'2.22E-6' '1.221203';'2.35E-6' '1.2886548';'2.4999999999999998E-6' '1.4507921';  ...
'2.67E-6' '1.6746349';'2.8599999999999997E-6' '1.8748013';'3.0799999999999997E-6' '1.91692';'3.33E-6' '1.8505739';'3.64E-6' '1.8774806';  ...
'4.0E-6' '1.9949739';'4.44E-6' '2.2388765';'4.9999999999999996E-6' '2.6672954';'5.7099999999999995E-6' '3.3712766';'6.67E-6' '4.4961625';  ...
'8.0E-6' '6.2868052';'9.9999999999999999E-6' '9.1935519';'1.1099999999999999E-5' '10.843912';'1.2499999999999999E-5' '12.935818';  ...
'1.43E-5' '15.634532';'1.67E-5' '19.231714';'1.9999999999999998E-5' '24.256111';'2.2199999999999998E-5' '27.599188';  ...
'2.4999999999999998E-5' '31.77255';'2.86E-5' '37.127733';'3.3299999999999996E-5' '44.209069';'3.9999999999999996E-5' '54.191208';  ...
'4.4399999999999995E-5' '61.731428';'4.9999999999999996E-5' '68.665703';'5.71E-5' '76.505227';'6.67E-5' '87.148396';  ...
'7.9999999999999999E-5' '103.53408';'9.9999999999999999E-5' '130.219';'1.25E-4' '163.02849';'1.54E-4' '196.84283';  ...
'1.9999999999999998E-4' '242.14161'});
model.component('comp1').material('mat5').propertyGroup('RefractiveIndex').func('int1').set('interp', 'piecewisecubic');
model.component('comp1').material('mat5').propertyGroup('RefractiveIndex').func('int1').set('argunit', 'm');
model.component('comp1').material('mat5').propertyGroup('RefractiveIndex').func('int1').set('fununit', '1');
model.component('comp1').material('mat5').propertyGroup('RefractiveIndex').func('int2').label('Refractive index, imaginary part');
model.component('comp1').material('mat5').propertyGroup('RefractiveIndex').func('int2').set('funcname', 'k_interp');
model.component('comp1').material('mat5').propertyGroup('RefractiveIndex').func('int2').set('table', {'6.67E-7' '2.9042727';  ...
'7.14E-7' '2.7924078';'7.69E-7' '2.6322954';'8.32999999999999E-7' '2.76706';'9.09E-7' '3.0100921';'1.0E-6' '3.4208368';'1.05E-6' '3.722401';  ...
'1.11E-6' '3.9693342';'1.18E-6' '4.1791582';'1.2499999999999999E-6' '4.3367596';'1.33E-6' '4.4460869';'1.4299999999999999E-6' '4.3901355';  ...
'1.5399999999999999E-6' '4.7741297';'1.6699999999999999E-6' '5.4881071';'1.82E-6' '6.4453363';'2.0E-6' '7.5659499';  ...
'2.1099999999999997E-6' '8.2613235';'2.22E-6' '9.0476312';'2.35E-6' '9.8569158';'2.4999999999999998E-6' '10.710594';  ...
'2.67E-6' '11.554665';'2.8599999999999997E-6' '12.375593';'3.0799999999999997E-6' '13.261664';'3.33E-6' '14.40046';'3.64E-6' '15.887186';  ...
'4.0E-6' '17.69477';'4.44E-6' '19.89851';'4.9999999999999996E-6' '22.612178';'5.7099999999999995E-6' '26.005432';'6.67E-6' '30.343034';  ...
'8.0E-6' '36.072044';'9.9999999999999999E-6' '44.019189';'1.1099999999999999E-5' '48.146015';'1.2499999999999999E-5' '53.112521';  ...
'1.43E-5' '59.208636';'1.67E-5' '66.923752';'1.9999999999999998E-5' '77.087223';  ...
'2.2199999999999998E-5' '83.517367';'2.4999999999999998E-5' '91.217414';'2.86E-5' '100.62038';'3.3299999999999996E-5' '112.34062';  ...
'3.9999999999999996E-5' '128.08146';'4.4399999999999995E-5' '136.67195';'4.9999999999999996E-5' '146.58093';'5.71E-5' '159.83902';  ...
'6.67E-5' '177.92054';'7.9999999999999999E-5' '202.21197';'9.9999999999999999E-5' '234.03939';'1.25E-4' '265.86909';'1.54E-4' '294.94073';  ...
'1.9999999999999998E-4' '332.81796'});
model.component('comp1').material('mat5').propertyGroup('RefractiveIndex').func('int2').set('interp', 'piecewisecubic');
model.component('comp1').material('mat5').propertyGroup('RefractiveIndex').func('int2').set('argunit', 'm');
model.component('comp1').material('mat5').propertyGroup('RefractiveIndex').func('int2').set('fununit', '1');
model.component('comp1').material('mat5').propertyGroup('RefractiveIndex').set('n', '');
model.component('comp1').material('mat5').propertyGroup('RefractiveIndex').set('ki', '');
model.component('comp1').material('mat5').propertyGroup('RefractiveIndex').set('n', '');
model.component('comp1').material('mat5').propertyGroup('RefractiveIndex').set('ki', '');
model.component('comp1').material('mat5').propertyGroup('RefractiveIndex').set('n', {'n_interp(c_const/freq)' '0' '0' '0' 'n_interp(c_const/freq)' '0' ...
    '0' '0' 'n_interp(c_const/freq)'});
model.component('comp1').material('mat5').propertyGroup('RefractiveIndex').set('ki', {'k_interp(c_const/freq)' '0' '0' '0' 'k_interp(c_const/freq)' '0' ...
    '0' '0' 'k_interp(c_const/freq)'});
model.component('comp1').material('mat5').propertyGroup('RefractiveIndex').addInput('frequency');
model.component('comp1').material('mat6').label(['Bi (Bismuth) (Hagemann et al. 1974: n,k 0.00155-6.199 ' native2unicode(hex2dec({'00' 'b5'}), ...
    'unicode') 'm)']);
model.component('comp1').material('mat6').set('groups', {'inorganic' 'Inorganic Materials'; 'bi_bismuth' 'Bi - Bismuth'; 'experimental_data' ...
    'Experimental data'});
model.component('comp1').material('mat6').propertyGroup('RefractiveIndex').func('int1').label('Refractive index');
model.component('comp1').material('mat6').propertyGroup('RefractiveIndex').func('int1').set('funcname', 'n_interp');
model.component('comp1').material('mat6').propertyGroup('RefractiveIndex').func('int1').set('table', {'2.48E-12' '1.0';'4.133E-12' '1.0';  ...
'1.2399999999999999E-11' '1.0';'1.378E-11' '1.0';'1.4589999999999998E-11' '1.0';'2.4799999999999998E-11' '1.0';  ...
'4.1329999999999995E-11' '1.0';'6.199E-11' '1.0';'8.2659999999999999E-11' '1.0';'8.856E-11' '1.0';'9.537E-11' '1.0';'1.033E-10' '1.0';  ...
'1.24E-10' '1.0';'1.5499999999999998E-10' '1.0';'2.066E-10' '1.0';'3.0999999999999996E-10' '1.0';'3.542E-10' '1.0';'3.757E-10' '1.0';  ...
'3.8749999999999995E-10' '1.0';'3.999E-10' '1.0';'4.133E-10' '1.0';'4.4279999999999996E-10' '1.0';'4.769E-10' '1.0';  ...
```

```
    '4.9589999999999999E-10' '1.0';'5.0609999999999999E-10' '1.0';'5.166E-10' '1.0';'5.636E-10' '1.0';'6.199E-10' '1.0';'6.888E-10' '1.0'; ...
    '7.749E-10' '1.0';'8.8559999999999999E-10' '0.999';'9.537E-10' '0.999';'1.033E-9' '0.999';'1.127E-9' '0.999'; ...
    '1.2399999999999999E-9' '0.999';'1.305E-9' '0.999';'1.378E-9' '0.999';'1.459E-9' '0.999';'1.5499999999999998E-9' '0.999'; ...
    '1.653E-9' '0.998';'1.771E-9' '0.998';'1.907E-9' '0.998';'2.066E-9' '0.998';'2.254E-9' '0.998'; '2.4799999999999997E-9' '0.998'; ...
    '2.583E-9' '0.998';'2.695E-9' '0.997';'2.818E-9' '0.997';'2.952E-9' '0.997';'3.0999999999999996E-9' '0.997';'3.179E-9' '0.997'; ...
    '3.2629999999999996E-9' '0.997';'3.3509999999999997E-9' '0.997';'3.4439999999999997E-9' '0.997';'3.542E-9' '0.997'; ...
    '3.647E-9' '0.997';'3.757E-9' '0.997';'3.875E-9' '0.997';'3.999E-9' '0.997';'4.1330000000000005E-9' '0.997';'4.275E-9' '0.997'; ...
    '4.4279999999999999E-9' '0.996';'4.592E-9' '0.996';'4.7689999999999996E-9' '0.996';'4.959E-9' '0.996';'5.166E-9' '0.995'; ...
    '5.391E-9' '0.994';'5.636E-9' '0.993';'5.904E-9' '0.991';'6.1989999999999999E-9' '0.99';'6.5250000000000005E-9' '0.988'; ...
    '6.8879999999999999E-9' '0.986';'7.2929999999999995E-9' '0.984';'7.7489999999999999E-9' '0.98';'8.2660000000000001E-9' '0.976'; ...
    '8.551E-9' '0.974';'8.8559999999999998E-9' '0.972';'9.184E-9' '0.969';'9.537E-9' '0.966';'9.9190000000000001E-9' '0.964'; ...
    '1.033E-8' '0.96';'1.078E-8' '0.956';'1.127E-8' '0.953';'1.181E-8' '0.949';'1.2399999999999999E-8' '0.945';'1.3050000000000001E-8' '0.94'; ...
    '1.378E-8' '0.935';'1.459E-8' '0.933';'1.55E-8' '0.932';'1.653E-8' '0.933';'1.7709999999999998E-8' '0.942';'1.9069999999999998E-8' '0.955'; ...
    '2.066E-8' '0.966';'2.101E-8' '0.97';'2.138E-8' '0.975';'2.1749999999999996E-8' '0.981';'2.214E-8' '0.987'; '2.254E-8' '0.993'; ...
    '2.2960000000000002E-8' '1.0';'2.339E-8' '1.007';'2.384E-8' '1.014';'2.4309999999999997E-8' '1.022'; '2.4799999999999997E-8' '1.027'; ...
    '2.5829999999999997E-8' '1.033';'2.695E-8' '1.036';'2.8179999999999998E-8' '1.041';'2.952E-8' '1.048';'3.1E-8' '1.051'; ...
    '3.2629999999999995E-8' '1.046';'3.4439999999999996E-8' '1.038';'3.647E-8' '1.027';'3.757E-8' '1.02';'3.875E-8' '1.009'; ...
    '3.9989999999999996E-8' '0.997';'4.1329999999999994E-8' '0.987';'4.2029999999999999E-8' '0.988';'4.275E-8' '0.991'; ...
    '4.349999999999999E-8' '0.987';'4.428E-8' '0.998';'4.5089999999999995E-8' '1.035';'4.5920000000000004E-8' '1.011'; ...
    '4.679E-8' '0.998';'4.769E-8' '0.999';'4.8619999999999995E-8' '1.001';'4.959E-8' '1.004';'5.061E-8' '1.07';'5.1659999999999994E-8' '1.065'; ...
    '5.276E-8' '1.013';'5.391E-8' '0.981';'5.6359999999999996E-8' '0.936';'6.199E-8' '0.856';'6.8879999999999999E-8' '0.763';'7.749E-8' '0.692'; ...
    '8.856E-8' '0.64';'1.0329999999999999E-7' '0.627';'1.24E-7' '0.71';'1.55E-7' '0.904';'2.0659999999999998E-7' '1.364';'2.48E-7' '1.482'; ...
    '3.1E-7' '1.517';'3.542E-7' '1.541';'4.133E-7' '1.556';'4.959E-7' '1.865';'6.199E-7' '2.244';'8.266E-7' '2.861';'1.24E-6' '5.036'; ...
    '1.55E-6' '6.247';'2.0659999999999998E-6' '7.466';'3.1E-6' '8.572';'6.1989999999999994E-6' '6.908'});
model.component('comp1').material('mat6').propertyGroup('RefractiveIndex').func('int1').set('interp', 'piecewisecubic');
model.component('comp1').material('mat6').propertyGroup('RefractiveIndex').func('int1').set('argunit', 'm');
model.component('comp1').material('mat6').propertyGroup('RefractiveIndex').func('int1').set('fununit', '1');
model.component('comp1').material('mat6').propertyGroup('RefractiveIndex').func('int2').label('Refractive index, imaginary part');
model.component('comp1').material('mat6').propertyGroup('RefractiveIndex').func('int2').set('funcname', 'k_interp');
model.component('comp1').material('mat6').propertyGroup('RefractiveIndex').func('int2').set('table', {'2.48E-12' '2.02E-11';'4.133E-12' '1.19E-10'; ...
    '1.2399999999999999E-11' '5.71E-9';'1.378E-11' '7.13E-9';'1.4589999999999998E-11' '2.56E-9';'2.4799999999999998E-11' '1.6E-8'; ...
    '4.1329999999999995E-11' '9.48E-8';'6.199E-11' '4.13E-7';'8.2659999999999999E-11' '7.04E-7';'8.856E-11' '9.36E-7';'9.537E-11' '4.95E-7'; ...
    '1.033E-10' '6.63E-7';'1.24E-10' '1.21E-6';'1.5499999999999998E-10' '2.64E-6';'2.066E-10' '7.69E-6';'3.0999999999999996E-10' '2.93E-5'; ...
    '3.542E-10' '4.5E-5';'3.757E-10' '5.23E-5';'3.8749999999999995E-10' '5.42E-5';'3.999E-10' '5.18E-5';'4.133E-10' '5.57E-5'; ...
    '4.4279999999999996E-10' '6.92E-5';'4.769E-10' '6.67E-5';'4.9589999999999999E-10' '7.39E-5';'5.0609999999999999E-10' '6.45E-5'; ...
    '5.166E-10' '3.34E-5';'5.636E-10' '4.54E-5';'6.199E-10' '6.27E-5';'6.888E-10' '8.82E-5';'7.749E-10' '1.29E-4';'8.8559999999999E-10' '1.98E-4'; ...
    '9.537E-10' '2.56E-4';'1.033E-9' '3.29E-4';'1.127E-9' '4.34E-4';'1.2399999999999999E-9' '5.79E-4';'1.305E-9' '7.01E-4';'1.378E-9' '8.39E-4'; ...
    '1.459E-9' '9.73E-4';'1.5499999999999998E-9' '0.00112';'1.653E-9' '0.00126';'1.771E-9' '0.00145';'1.907E-9' '0.00164';'2.066E-9' '0.00188'; ...
    '2.254E-9' '0.00214';'2.4799999999999997E-9' '0.00245';'2.583E-9' '0.00259';'2.695E-9' '0.00276';'2.818E-9' '0.00294';'2.952E-9' '0.00314'; ...
    '3.0999999999999996E-9' '0.00335';'3.179E-9' '0.00345';'3.2629999999999996E-9' '0.00355';'3.3509999999999997E-9' '0.00365'; ...
    '3.4439999999999997E-9' '0.00376';'3.542E-9' '0.00387';'3.647E-9' '0.00398';'3.757E-9' '0.00405'; '3.875E-9' '0.00411';'3.999E-9' '0.00416'; ...
    '4.1330000000000005E-9' '0.00418';'4.275E-9' '0.00417';'4.4279999999999999E-9' '0.00412';'4.592E-9' '0.00403'; ...
    '4.7689999999999996E-9' '0.00364';'4.959E-9' '0.00315';'5.166E-9' '0.00293'; '5.391E-9' '0.00285'; '5.636E-9' '0.00279'; ...
    '5.904E-9' '0.00279';'6.1989999999999999E-9' '0.00283';'6.5250000000000005E-9' '0.00299';'6.8879999999999999E-9' '0.00331'; ...
    '7.2929999999999995E-9' '0.00381';'7.7489999999999999E-9' '0.00448';'8.2660000000000001E-9' '0.00603';'8.551E-9' '0.00726'; ...
    '8.8559999999999998E-9' '0.00867';'9.184E-9' '0.0104';'9.537E-9' '0.0122';'9.9190000000000001E-9' '0.0143';'1.033E-8' '0.017'; ...
    '1.078E-8' '0.0205';'1.127E-8' '0.0251';'1.181E-8' '0.03';'1.2399999999999999E-8' '0.0359';'1.3050000000000001E-8' '0.0437'; ...
    '1.378E-8' '0.0536';'1.459E-8' '0.0671';'1.55E-8' '0.0807';'1.653E-8' '0.0985';'1.7709999999999998E-8' '0.12';'1.9069999999999998E-8' '0.132'; ...
    '2.066E-8' '0.151';'2.101E-8' '0.155';'2.138E-8' '0.159';'2.1749999999999996E-8' '0.162';'2.214E-8' '0.165';'2.254E-8' '0.167'; ...
    '2.2960000000000002E-8' '0.167';'2.339E-8' '0.168';'2.384E-8' '0.167';'2.4309999999999997E-8' '0.167';'2.4799999999999997E-8' '0.16'; ...
    '2.5829999999999997E-8' '0.152';'2.695E-8' '0.149';'2.8179999999999998E-8' '0.144';'2.952E-8' '0.142';'3.1E-8' '0.131'; ...
    '3.2629999999999995E-8' '0.12';'3.4439999999999996E-8' '0.114';'3.647E-8' '0.109';'3.757E-8' '0.108';'3.875E-8' '0.108'; ...
    '3.9989999999999996E-8' '0.115';'4.1329999999999994E-8' '0.127';'4.2029999999999999E-8' '0.138';'4.275E-8' '0.143'; ...
    '4.349999999999999E-8' '0.149';'4.428E-8' '0.172';'4.5089999999999995E-8' '0.15';'4.5920000000000004E-8' '0.128';'4.679E-8' '0.138'; ...
    '4.769E-8' '0.146';'4.8619999999999995E-8' '0.152';'4.959E-8' '0.168';'5.061E-8' '0.183';'5.1659999999999994E-8' '0.0739'; ...
    '5.276E-8' '0.0609';'5.391E-8' '0.0622';'5.6359999999999996E-8' '0.0717';'6.199E-8' '0.103';'6.8879999999999999E-8' '0.168'; ...
    '7.749E-8' '0.293';'8.856E-8' '0.456';'1.0329999999999999E-7' '0.681';'1.24E-7' '0.968';'1.55E-7' '1.3';'2.0659999999999998E-7' '1.54'; ...
    '2.48E-7' '1.52';'3.1E-7' '1.7';'3.542E-7' '1.85';'4.133E-7' '2.27';'4.959E-7' '2.67';'6.199E-7' '3.1';'8.266E-7' '3.89';'1.24E-6' '4.84';'1.55E-6' '4.52'; ...
    '2.0659999999999998E-6' '3.87';'3.1E-6' '2.21';'6.1989999999999994E-6' '1.47'});
model.component('comp1').material('mat6').propertyGroup('RefractiveIndex').func('int2').set('interp', 'piecewisecubic');
model.component('comp1').material('mat6').propertyGroup('RefractiveIndex').func('int2').set('argunit', 'm');
model.component('comp1').material('mat6').propertyGroup('RefractiveIndex').func('int2').set('fununit', '1');
model.component('comp1').material('mat6').propertyGroup('RefractiveIndex').set('n', '');
model.component('comp1').material('mat6').propertyGroup('RefractiveIndex').set('ki', '');
model.component('comp1').material('mat6').propertyGroup('RefractiveIndex').set('n', '');
model.component('comp1').material('mat6').propertyGroup('RefractiveIndex').set('ki', '');
model.component('comp1').material('mat6').propertyGroup('RefractiveIndex').set('n', {'n_interp(c_const/freq)' '0' '0' '0' 'n_interp(c_const/freq)' '0' ...
    '0' '0' 'n_interp(c_const/freq)'});
model.component('comp1').material('mat6').propertyGroup('RefractiveIndex').set('ki', {'k_interp(c_const/freq)' '0' '0' '0' 'k_interp(c_const/freq)' '0' ...
    '0' '0' 'k_interp(c_const/freq)'});
model.component('comp1').material('mat6').propertyGroup('RefractiveIndex').addInput('frequency');
model.component('comp1').physics('ewfd').prop('EquationForm').set('freq', '1[GHz]');
model.component('comp1').physics('ewfd').prop('EquationForm').set('modeFreq', '1[GHz]');
model.component('comp1').physics('ewfd').prop('MeshControl').set('EnableMeshControl', false);
model.component('comp1').physics('ewfd').feature('port1').set('Pin', '1[mW]');
model.component('comp1').physics('ewfd').feature('port1').set('beta', 'abs(ewfd.k0)');
model.component('comp1').physics('ewfd').feature('port1').set('E0', {'0'; 'exp(-j*ewfd.k0*z)'; '0'});
model.component('comp1').physics('ewfd').feature('port2').set('beta', 'abs(ewfd.k0)');
model.component('comp1').physics('ewfd').feature('port2').set('E0', {'0'; 'exp(-j*ewfd.k0*z)'; '0'});
model.component('comp1').mesh('mesh1').feature('size').set('hauto', 4);
model.component('comp1').mesh('mesh1').run;
model.study.create('std1');
model.study('std1').create('freq', 'Frequency');
model.sol.create('sol1');
model.sol('sol1').study('std1');
```

```
model.sol('sol1').attach('std1');
model.sol('sol1').create('st1', 'StudyStep');
model.sol('sol1').create('v1', 'Variables');
model.sol('sol1').create('s1', 'Stationary');
model.sol('sol1').feature('s1').create('p1', 'Parametric');
model.sol('sol1').feature('s1').create('fc1', 'FullyCoupled');
model.sol('sol1').feature('s1').create('i1', 'Iterative');
model.sol('sol1').feature('s1').feature('i1').create('mg1', 'Multigrid');
model.sol('sol1').feature('s1').feature('i1').feature('mg1').feature('pr').create('sv1', 'SORVector');
model.sol('sol1').feature('s1').feature('i1').feature('mg1').feature('po').create('sv1', 'SORVector');
model.sol('sol1').feature('s1').feature.remove('fcDef');
model.result.numerical.create('int1', 'IntSurface');
model.result.numerical('int1').selection.set([13 29]);
model.result.numerical('int1').set('probetag', 'none');
model.result.create('pg1', 'PlotGroup3D');
model.result.create('pg2', 'PlotGroup3D');
model.result.create('pg3', 'PlotGroup1D');
model.result('pg1').create('mslc1', 'Multislice');
model.result('pg2').create('slc1', 'Slice');
model.result('pg2').feature('slc1').create('def1', 'Deform');
model.result('pg3').create('tblp1', 'Table');
model.study('std1').feature('freq').set('punit', 'Hz');
model.study('std1').feature('freq').set('plist', '1.0e12');
model.sol('sol1').attach('std1');
model.sol('sol1').feature('v1').set('clistctrl', {'p1'});
model.sol('sol1').feature('v1').set('cname', {'freq'});
model.sol('sol1').feature('v1').set('clist', {'1.0e12[Hz]'});
model.sol('sol1').feature('s1').feature('p1').set('pname', {'freq'});
model.sol('sol1').feature('s1').feature('p1').set('plistarr', {'1.0e12'});
model.sol('sol1').feature('s1').feature('p1').set('punit', {'Hz'});
model.sol('sol1').feature('s1').feature('p1').set('pcontinuationmode', 'no');
model.sol('sol1').feature('s1').feature('p1').set('preusesol', 'auto');
model.sol('sol1').feature('s1').feature('i1').label('Suggested Iterative Solver (ewfd)');
model.sol('sol1').feature('s1').feature('i1').set('linsolver', 'bicgstab');
model.sol('sol1').feature('s1').feature('i1').feature('mg1').feature('pr').feature('sv1').set('sorvecdof', {'comp1_E'});
model.sol('sol1').feature('s1').feature('i1').feature('mg1').feature('po').feature('sv1').set('sorvecdof', {'comp1_E'});
model.sol('sol1').runAll;
model.result.numerical('int1').set('table', 'tbl1');
model.result.numerical('int1').set('expr', {'ewfd.normE'});
model.result.numerical('int1').set('unit', {'V*m'});
model.result.numerical('int1').set('descr', {'Electric field norm'});
model.result.numerical('int1').setResult;
model.result('pg1').label('Electric Field (ewfd)');
model.result('pg1').set('frametype', 'spatial');
model.result('pg1').feature('mslc1').set('multiplanezmethod', 'coord');
model.result('pg1').feature('mslc1').set('zcoord', '1050.0');
model.result('pg1').feature('mslc1').set('rangecoloractive', true);
model.result('pg1').feature('mslc1').set('rangecolormin', 41923.82486663746);
model.result('pg1').feature('mslc1').set('rangecolormax', 400000);
model.result('pg1').feature('mslc1').set('smooth', 'internal');
model.result('pg1').feature('mslc1').set('resolution', 'normal');
model.result('pg2').feature('slc1').set('quickplane', 'xy');
model.result('pg2').feature('slc1').set('quickzmethod', 'coord');
model.result('pg2').feature('slc1').set('quickz', '1033.0');
model.result('pg2').feature('slc1').set('rangecoloractive', true);
model.result('pg2').feature('slc1').set('rangecolormin', 9292.73707567522);
model.result('pg2').feature('slc1').set('rangecolormax', 250000);
model.result('pg2').feature('slc1').set('smooth', 'internal');
model.result('pg2').feature('slc1').set('resolution', 'normal');
model.result('pg2').feature('slc1').feature('def1').active(false);
model.result('pg2').feature('slc1').feature('def1').set('expr', {'+ewfd.Ex' '+ewfd.Ey' '+ewfd.Ez'});
model.result('pg2').feature('slc1').feature('def1').set('descr', ' + Electric field');
model.result('pg2').feature('slc1').feature('def1').set('scale', 2.085880595474555E-4);
model.result('pg2').feature('slc1').feature('def1').set('scaleactive', false);
model.result('pg3').set('data', 'none');
model.result('pg3').set('xlabel', 'freq (Hz)');
model.result('pg3').set('ylabel', 'Electric field norm (V*m)');
model.result('pg3').set('xlabelactive', false);
model.result('pg3').set('ylabelactive', false);
model.result('pg3').feature('tblp1').set('linewidth', 3);
model.result('pg3').feature('tblp1').set('linemarker', 'cycle');
model.result('pg3').feature('tblp1').set('markerpos', 'datapoints');
end
```

```
function [] = fKAg(model)
model.component('comp1').material.create('mat1', 'Common');
model.component('comp1').material.create('mat3', 'Common');
model.component('comp1').material.create('mat6', 'Common');
model.component('comp1').material('mat1').propertyGroup('def').func.create('eta', 'Piecewise');
model.component('comp1').material('mat1').propertyGroup('def').func.create('Cp', 'Piecewise');
model.component('comp1').material('mat1').propertyGroup('def').func.create('rho', 'Analytic');
model.component('comp1').material('mat1').propertyGroup('def').func.create('k', 'Piecewise');
model.component('comp1').material('mat1').propertyGroup('def').func.create('cs', 'Analytic');
model.component('comp1').material('mat1').propertyGroup.create('RefractiveIndex', 'Refractive index');
model.component('comp1').material('mat3').selection.set([3]);
model.component('comp1').material('mat3').propertyGroup('def').func.create('k_solid_1', 'Piecewise');
model.component('comp1').material('mat3').propertyGroup('def').func.create('res_solid_1', 'Piecewise');
model.component('comp1').material('mat3').propertyGroup('def').func.create('alpha', 'Piecewise');
model.component('comp1').material('mat3').propertyGroup('def').func.create('C_solid_1', 'Piecewise');
model.component('comp1').material('mat3').propertyGroup('def').func.create('sigma_solid_1', 'Piecewise');
model.component('comp1').material('mat3').propertyGroup('def').func.create('HC_solid_1', 'Piecewise');
model.component('comp1').material('mat3').propertyGroup('def').func.create('VP_solid_1', 'Piecewise');
model.component('comp1').material('mat3').propertyGroup('def').func.create('rho', 'Piecewise');
model.component('comp1').material('mat3').propertyGroup.create('ThermalExpansion', 'Thermal expansion');
model.component('comp1').material('mat3').propertyGroup('ThermalExpansion').func.create('dL', 'Piecewise');
model.component('comp1').material('mat3').propertyGroup('ThermalExpansion').func.create('CTE', 'Piecewise');
model.component('comp1').material('mat3').propertyGroup.create('Enu', 'Young''s modulus and Poisson''s ratio');
model.component('comp1').material('mat3').propertyGroup('Enu').func.create('E', 'Piecewise');
model.component('comp1').material('mat3').propertyGroup('Enu').func.create('nu', 'Piecewise');
model.component('comp1').material('mat3').propertyGroup.create('KG', 'Bulk modulus and shear modulus');
model.component('comp1').material('mat3').propertyGroup('KG').func.create('mu', 'Piecewise');
model.component('comp1').material('mat3').propertyGroup('KG').func.create('kappa', 'Piecewise');
model.component('comp1').material('mat3').propertyGroup.create('RefractiveIndex', 'Refractive index');
model.component('comp1').material('mat6').selection.set([4]);
model.component('comp1').material('mat6').propertyGroup('def').func.create('k_solid_1', 'Piecewise');
model.component('comp1').material('mat6').propertyGroup('def').func.create('res_solid_1', 'Piecewise');
model.component('comp1').material('mat6').propertyGroup('def').func.create('Syfunc_solid_tested_at_195K_8', 'Piecewise');
model.component('comp1').material('mat6').propertyGroup('def').func.create('Syt', 'Piecewise');
model.component('comp1').material('mat6').propertyGroup('def').func.create('C_solid_1', 'Piecewise');
model.component('comp1').material('mat6').propertyGroup('def').func.create('sigma_solid_1', 'Piecewise');
model.component('comp1').material('mat6').propertyGroup('def').func.create('HC_solid_1', 'Piecewise');
model.component('comp1').material('mat6').propertyGroup('def').func.create('VP_solid_1', 'Piecewise');
model.component('comp1').material('mat6').propertyGroup('def').func.create('elong', 'Piecewise');
model.component('comp1').material('mat6').propertyGroup.create('Enu', 'Young''s modulus and Poisson''s ratio');
model.component('comp1').material('mat6').propertyGroup('Enu').func.create('E', 'Piecewise');
model.component('comp1').material('mat6').propertyGroup('Enu').func.create('nu', 'Piecewise');
model.component('comp1').material('mat6').propertyGroup.create('KG', 'Bulk modulus and shear modulus');
model.component('comp1').material('mat6').propertyGroup('KG').func.create('mu', 'Piecewise');
model.component('comp1').material('mat6').propertyGroup('KG').func.create('kappa_solid_1', 'Piecewise');
model.component('comp1').material('mat6').propertyGroup.create('ElastoplasticModel', 'Elastoplastic material model');
model.component('comp1').material('mat6').propertyGroup('ElastoplasticModel').func.create('Sys', 'Piecewise');
model.component('comp1').material('mat6').propertyGroup.create('RefractiveIndex', 'Refractive index');
model.component('comp1').physics.create('ewfd', 'ElectromagneticWavesFrequencyDomain', 'geom1');
model.component('comp1').physics('ewfd').create('sctr1', 'Scattering', 2);
model.component('comp1').physics('ewfd').feature('sctr1').selection.set([1 2 3 4 5 7 8 9 43 44]);
model.component('comp1').physics('ewfd').create('port1', 'Port', 2);
model.component('comp1').physics('ewfd').feature('port1').selection.set([7]);
model.component('comp1').physics('ewfd').create('port2', 'Port', 2);
model.component('comp1').physics('ewfd').feature('port2').selection.set([3]);
model.component('comp1').mesh('mesh1').create('ftet1', 'FreeTet');
model.result.table('tbl1').comments('Surface Integration 1 (ewfd.normE)');
model.result.table('evl3').label('Evaluation 3D');
model.result.table('evl3').comments('Interactive 3D values');
model.component('comp1').view('view1').set('renderwireframe', true);
model.component('comp1').view('view2').axis.set('xmin', -37.6529541015625);
model.component('comp1').view('view2').axis.set('xmax', 2112.7470703125);
model.component('comp1').view('view2').axis.set('ymin', 0);
model.component('comp1').view('view2').axis.set('ymax', 1651.199951171875);
model.component('comp1').material('mat1').label('Air');
model.component('comp1').material('mat1').set('family', 'air');
model.component('comp1').material('mat1').propertyGroup('def').func('eta').set('arg', 'T');
model.component('comp1').material('mat1').propertyGroup('def').func('eta').set('pieces', {'200.0' '1600.0' ...
    '-8.38278E-7+8.35717342E-8*T^1-7.69429583E-11*T^2+4.6437266E-14*T^3-1.06585607E-17*T^4'});
model.component('comp1').material('mat1').propertyGroup('def').func('Cp').set('arg', 'T');
model.component('comp1').material('mat1').propertyGroup('def').func('Cp').set('pieces', {'200.0' '1600.0' ...
    '1047.63657-0.372589265*T^1+9.45304214E-4*T^2-6.02409443E-7*T^3+1.2858961E-10*T^4'});
model.component('comp1').material('mat1').propertyGroup('def').func('rho').set('expr', 'pA*0.02897/8.314/T');
model.component('comp1').material('mat1').propertyGroup('def').func('rho').set('args', {'pA' 'T'});
model.component('comp1').material('mat1').propertyGroup('def').func('rho').set('dermethod', 'manual');
model.component('comp1').material('mat1').propertyGroup('def').func('rho').set('argders', {'pA' 'd(pA*0.02897/8.314/T,pA)'; 'T' ...
    'd(pA*0.02897/8.314/T,T)'});
model.component('comp1').material('mat1').propertyGroup('def').func('rho').set('plotargs', {'pA' '0' '1'; 'T' '0' '1'});
model.component('comp1').material('mat1').propertyGroup('def').func('k').set('arg', 'T');
model.component('comp1').material('mat1').propertyGroup('def').func('k').set('pieces', {'200.0' '1600.0' ...
    '-0.00227583562+1.15480022E-4*T^1-7.90252856E-8*T^2+4.11702505E-11*T^3-7.43864331E-15*T^4'});
model.component('comp1').material('mat1').propertyGroup('def').func('cs').set('expr', 'sqrt(1.4*287*T)');
model.component('comp1').material('mat1').propertyGroup('def').func('cs').set('args', {'T'});
model.component('comp1').material('mat1').propertyGroup('def').func('cs').set('dermethod', 'manual');
model.component('comp1').material('mat1').propertyGroup('def').func('cs').set('argders', {'T' 'd(sqrt(1.4*287*T),T)'});
model.component('comp1').material('mat1').propertyGroup('def').func('cs').set('plotargs', {'T' '0' '1'});
```

```
model.component('comp1').material('mat1').propertyGroup('def').set('relpermeability', {'1' '0' '0' '0' '1' '0' '0' '0' '1'});
model.component('comp1').material('mat1').propertyGroup('def').set('relpermittivity', {'1' '0' '0' '0' '1' '0' '0' '0' '1'});
model.component('comp1').material('mat1').propertyGroup('def').set('dynamicviscosity', 'eta(T[1/K])[Pa*s]');
model.component('comp1').material('mat1').propertyGroup('def').set('ratioofspecificheat', '1.4');
model.component('comp1').material('mat1').propertyGroup('def').set('electricconductivity', {'0[S/m]' '0' '0' '0' '0[S/m]' '0' '0' '0' '0[S/m]'});
model.component('comp1').material('mat1').propertyGroup('def').set('heatcapacity', 'Cp(T[1/K])[J/(kg*K)]');
model.component('comp1').material('mat1').propertyGroup('def').set('density', 'rho(pA[1/Pa],T[1/K])[kg/m^3]');
model.component('comp1').material('mat1').propertyGroup('def').set('thermalconductivity', {'k(T[1/K])[W/(m*K)]' '0' '0' '0' 'k(T[1/K])[W/(m*K)]' ...
    '0' '0' '0' 'k(T[1/K])[W/(m*K)]'});
model.component('comp1').material('mat1').propertyGroup('def').set('soundspeed', 'cs(T[1/K])[m/s]');
model.component('comp1').material('mat1').propertyGroup('def').addInput('temperature');
model.component('comp1').material('mat1').propertyGroup('def').addInput('pressure');
model.component('comp1').material('mat1').propertyGroup('RefractiveIndex').set('n', '');
model.component('comp1').material('mat1').propertyGroup('RefractiveIndex').set('ki', '');
model.component('comp1').material('mat1').propertyGroup('RefractiveIndex').set('n', {'1' '0' '0' '0' '1' '0' '0' '0' '1'});
model.component('comp1').material('mat1').propertyGroup('RefractiveIndex').set('ki', {'0' '0' '0' '0' '0' '0' '0' '0' '0'});
model.component('comp1').material('mat3').label('Silver [solid]');
model.component('comp1').material('mat3').set('family', 'custom');
model.component('comp1').material('mat3').set('specular', 'custom');
model.component('comp1').material('mat3').set('customspecular', [0.7843137254901961 1 1]);
model.component('comp1').material('mat3').set('diffuse', 'custom');
model.component('comp1').material('mat3').set('customdiffuse', [0.7843137254901961 0.7843137254901961 0.7843137254901961]);
model.component('comp1').material('mat3').set('ambient', 'custom');
model.component('comp1').material('mat3').set('customambient', [0.7843137254901961 0.7843137254901961 0.7843137254901961]);
model.component('comp1').material('mat3').set('noise', true);
model.component('comp1').material('mat3').set('noisefreq', 1);
model.component('comp1').material('mat3').set('lighting', 'cooktorrance');
model.component('comp1').material('mat3').set('fresnel', 0.9);
model.component('comp1').material('mat3').propertyGroup('def').func('k_solid_1').set('arg', 'T');
model.component('comp1').material('mat3').propertyGroup('def').func('k_solid_1').set('pieces', {'0.0' '12.0' ...
    '3587.152*T^1+348.8866*T^2-101.463*T^3+5.370411*T^4-0.06180556*T^5'; ...
  '12.0' '35.0' '50522.05-4753.283*T^1+172.2387*T^2-2.689213*T^3+0.01433909*T^4'; ...
  '35.0' '100.0' '10048.41-524.6271*T^1+11.6041*T^2-0.1287378*T^3+7.123183E-4*T^4-1.566974E-6*T^5'; ...
  '100.0' '250.0' '759.2297-7.120703*T^1+0.06197118*T^2-2.715928E-4*T^3+6.016276E-7*T^4-5.424229E-10*T^5'; ...
  '250.0' '1235.0' '419.8682+0.09979317*T^1-2.937158E-4*T^2+2.109166E-7*T^3-5.786644E-11*T^4'});
model.component('comp1').material('mat3').propertyGroup('def').func('res_solid_1').set('arg', 'T');
model.component('comp1').material('mat3').propertyGroup('def').func('res_solid_1').set('pieces', {'1.0' '15.8' ...
    '9.822048E-12+2.259567E-13*T^1-6.690094E-14*T^2+6.144183E-15*T^3'; ...
  '15.8' '27.5' '-2.33E-11+7.473333E-12*T^1-6.16E-13*T^2+2.026667E-14*T^3'; ...
  '27.5' '60.0' '8.015476E-10-7.586429E-11*T^1+2.210952E-12*T^2-1.2E-14*T^3'; ...
  '60.0' '200.0' '-2.428741E-9+6.974508E-11*T^1-4.447028E-14*T^2+6.841184E-17*T^3'; ...
  '200.0' '1235.0' '-1.812752E-9+6.074742E-11*T^1-3.077059E-15*T^2+8.269045E-18*T^3'});
model.component('comp1').material('mat3').propertyGroup('def').func('alpha').set('arg', 'T');
model.component('comp1').material('mat3').propertyGroup('def').func('alpha').set('pieces', {'0.0' '92.0' ...
    '1.397008E-5+6.293815E-8*T^1-3.772802E-10*T^2+1.128415E-12*T^3-1.223488E-15*T^4'; '92.0' '873.0' ...
    '1.604749E-5+1.576798E-8*T^1-1.719191E-11*T^2+6.931419E-15*T^3'});
model.component('comp1').material('mat3').propertyGroup('def').func('C_solid_1').set('arg', 'T');
model.component('comp1').material('mat3').propertyGroup('def').func('C_solid_1').set('pieces', {'1.0' '12.3' ...
    '0.01224227-0.01394369*T^1+0.01009593*T^2-4.957659E-4*T^3+1.66168E-4*T^4-3.512349E-6*T^5'; ...
  '12.3' '75.0' '24.24847-4.669813*T^1+0.2956444*T^2-0.004853545*T^3+3.380225E-5*T^4-8.439015E-8*T^5'; ...
  '75.0' '300.0' '-63.85884+5.177265*T^1-0.03961478*T^2+1.570454E-4*T^3-3.105375E-7*T^4+2.411435E-10*T^5'; ...
  '300.0' '1235.0' '225.7065+0.01705702*T^1+5.007143E-5*T^2-1.768498E-8*T^3'});
model.component('comp1').material('mat3').propertyGroup('def').func('sigma_solid_1').set('arg', 'T');
model.component('comp1').material('mat3').propertyGroup('def').func('sigma_solid_1').set('pieces', {'1.0' '15.8' ...
    '1/(6.144183E-15*T^3-6.690094E-14*T^2+2.259567E-13*T+9.822048E-12)'; ...
  '15.8' '27.5' '1/(2.026667E-14*T^3-6.160000E-13*T^2+7.473333E-12*T-2.330000E-11)'; ...
  '27.5' '60.0' '1/(-1.200000E-14*T^3+2.210952E-12*T^2-7.586429E-11*T+8.015476E-10)'; ...
  '60.0' '200.0' '1/(6.841184E-17*T^3-4.447028E-14*T^2+6.974508E-11*T-2.428741E-09)'; ...
  '200.0' '1235.0' '1/(8.269045E-18*T^3-3.077059E-15*T^2+6.074742E-11*T-1.812752E-09)'});
model.component('comp1').material('mat3').propertyGroup('def').func('HC_solid_1').set('arg', 'T');
model.component('comp1').material('mat3').propertyGroup('def').func('HC_solid_1').set('pieces', {'1.0' '12.3' ...
    '0.001320549-0.001504078*T^1+0.001089028*T^2-5.347725E-5*T^3+1.792421E-5*T^4-3.788701E-7*T^5'; ...
  '12.3' '75.0' '2.615634-0.5037235*T^1+0.03189057*T^2-5.235422E-4*T^3+3.646182E-6*T^4-9.102999E-9*T^5'; ...
  '75.0' '300.0' '-6.888324+0.5584615*T^1-0.004273165*T^2+1.694017E-5*T^3-3.349706E-8*T^4+2.601167E-11*T^5'; ...
  '300.0' '1235.0' '24.34652+0.001839907*T^1+5.401105E-6*T^2-1.907643E-9*T^3'});
model.component('comp1').material('mat3').propertyGroup('def').func('VP_solid_1').set('arg', 'T');
model.component('comp1').material('mat3').propertyGroup('def').func('VP_solid_1').set('pieces', {'293.0' '1235.0' ...
    '(exp((-1.499900e+04/T-7.845000e-01*log10(T)+1.200781e+01)*log(10.0)))*1.333200e+02'});
model.component('comp1').material('mat3').propertyGroup('def').func('rho').set('arg', 'T');
model.component('comp1').material('mat3').propertyGroup('def').func('rho').set('pieces', {'0.0' '30.0' '10630.38'; '30.0' '140.0' ...
    '10630.25+0.1394367*T^1-0.0048655*T^2+1.188653E-5*T^3'; '140.0' '873.0' '10658.96-0.4692536*T^1-2.976784E-4*T^2+1.470941E-7*T^3'});
model.component('comp1').material('mat3').propertyGroup('def').set('thermalconductivity', {'k_solid_1(T[1/K])[W/(m*K)]' '0' '0' '0' ...
    'k_solid_1(T[1/K])[W/(m*K)]' '0' '0' '0' 'k_solid_1(T[1/K])[W/(m*K)]'});
model.component('comp1').material('mat3').propertyGroup('def').set('resistivity', {'res_solid_1(T[1/K])[ohm*m]' '0' '0' '0' ...
    'res_solid_1(T[1/K])[ohm*m]' '0' '0' '0' 'res_solid_1(T[1/K])[ohm*m]'});
model.component('comp1').material('mat3').propertyGroup('def').set('thermalexpansioncoefficient', ...
{'(alpha(T[1/K])[1/K]+(Tempref-293[K])*if(abs(T-Tempref)>1e-3,(alpha(T[1/K])[1/K]-alpha(Tempref[1/K])[1/K])/(T-Tempref),d(alpha(T[1/K])[1/K],T)))/(1
model.component('comp1').material('mat3').propertyGroup('def').set('heatcapacity', 'C_solid_1(T[1/K])[J/(kg*K)]');
model.component('comp1').material('mat3').propertyGroup('def').set('electricconductivity', {'sigma_solid_1(T[1/K])[S/m]' '0' '0' '0' ...
    'sigma_solid_1(T[1/K])[S/m]' '0' '0' '0' 'sigma_solid_1(T[1/K])[S/m]'});
model.component('comp1').material('mat3').propertyGroup('def').set('HC', 'HC_solid_1(T[1/K])[J/(mol*K)]');
model.component('comp1').material('mat3').propertyGroup('def').set('VP', 'VP_solid_1(T[1/K])[Pa]');
model.component('comp1').material('mat3').propertyGroup('def').set('density', 'rho(T[1/K])[kg/m^3]');
model.component('comp1').material('mat3').propertyGroup('def').addInput('temperature');
model.component('comp1').material('mat3').propertyGroup('def').addInput('strainreferencetemperature');
model.component('comp1').material('mat3').propertyGroup('ThermalExpansion').func('dL').set('arg', 'T');
```

```
'10.0' '30.0' '-0.00415083+2.566303E-6*T^1-1.154799E-7*T^2+2.71357E-9*T^3';  ...
'30.0' '87.0' '-0.004065065-6.856586E-6*T^1+2.120401E-7*T^2-9.814541E-10*T^3+2.367498E-12*T^4-2.275841E-15*T^5';  ...
'87.0' '873.0' '-0.004745213+1.182195E-5*T^1+1.97186E-8*T^2-1.837579E-11*T^3+6.889976E-15*T^4'});
model.component('comp1').material('mat3').propertyGroup('ThermalExpansion').func('CTE').set('arg', 'T');
model.component('comp1').material('mat3').propertyGroup('ThermalExpansion').func('CTE').set('pieces', {'30.0' '300.0' ...
    '-8.786433E-6+5.111237E-7*T^1-4.275991E-9*T^2+1.8602E-11*T^3-4.019836E-14*T^4+3.403595E-17*T^5'});
model.component('comp1').material('mat3').propertyGroup('ThermalExpansion').set('alphatan', '');
model.component('comp1').material('mat3').propertyGroup('ThermalExpansion').set('dL', '');
model.component('comp1').material('mat3').propertyGroup('ThermalExpansion').set('alphatanIso', '');
model.component('comp1').material('mat3').propertyGroup('ThermalExpansion').set('dLIso', '');
model.component('comp1').material('mat3').propertyGroup('ThermalExpansion').set('alphatan', {'CTE(T[1/K])[1/K]' '0' '0' '0' 'CTE(T[1/K])[1/K]' '0' ...
    '0' '0' 'CTE(T[1/K])[1/K]'});
model.component('comp1').material('mat3').propertyGroup('ThermalExpansion').set('dL', {'(dL(T[1/K])-dL(Tempref[1/K]))/(1+dL(Tempref[1/K]))' ...
    '0' '0' '0' '(dL(T[1/K])-dL(Tempref[1/K]))/(1+dL(Tempref[1/K]))' '0' '0' '0' '(dL(T[1/K])-dL(Tempref[1/K]))/(1+dL(Tempref[1/K]))'});
model.component('comp1').material('mat3').propertyGroup('ThermalExpansion').set('alphatanIso', 'CTE(T)');
model.component('comp1').material('mat3').propertyGroup('ThermalExpansion').set('dLIso', '(dL(T)-dL(Tempref))/(1+dL(Tempref))');
model.component('comp1').material('mat3').propertyGroup('ThermalExpansion').addInput('temperature');
model.component('comp1').material('mat3').propertyGroup('ThermalExpansion').addInput('strainreferencetemperature');
model.component('comp1').material('mat3').propertyGroup('Enu').func('E').set('arg', 'T');
model.component('comp1').material('mat3').propertyGroup('Enu').func('E').set('pieces', {'0.0' '1173.0' ...
    '9.143965E10-2.775728E7*T^1-38123.31*T^2+49.99535*T^3-0.02009828*T^4'});
model.component('comp1').material('mat3').propertyGroup('Enu').func('nu').set('arg', 'T');
model.component('comp1').material('mat3').propertyGroup('Enu').func('nu').set('pieces', {'0.0' '1173.0' ...
    '0.360203+1.499369E-5*T^1+4.008534E-8*T^2-1.475466E-11*T^3-1.181682E-15*T^4'});
model.component('comp1').material('mat6').set('specular', 'custom');
model.component('comp1').material('mat6').set('customspecular', [0.7843137254901961 1 1]);
model.component('comp1').material('mat6').set('diffuse', 'custom');
model.component('comp1').material('mat6').set('customdiffuse', [0.7843137254901961 0.7843137254901961 0.7843137254901961]);
model.component('comp1').material('mat6').set('ambient', 'custom');
model.component('comp1').material('mat6').set('customambient', [0.7843137254901961 0.7843137254901961 0.7843137254901961]);
model.component('comp1').material('mat6').set('noise', true);
model.component('comp1').material('mat6').set('noisefreq', 1);
model.component('comp1').material('mat6').set('lighting', 'cooktorrance');
model.component('comp1').material('mat6').set('fresnel', 0.9);
model.component('comp1').material('mat6').propertyGroup('def').func('k_solid_1').set('arg', 'T');
model.component('comp1').material('mat6').propertyGroup('def').func('k_solid_1').set('pieces', {'0.0' '6.0' ...
    '1122.085*T^1+263.7268*T^2-227.3021*T^3+38.40137*T^4-2.033789*T^5';  ...
'6.0' '20.0' '6265.378-1779.74*T^1+224.4572*T^2-14.64801*T^3+0.4827659*T^4-0.006357878*T^5';  ...
'20.0' '50.0' '1340.183-148.6854*T^1+7.279199*T^2-0.1780719*T^3+0.002163925*T^4-1.042377E-5*T^5';  ...
'50.0' '100.0' '131.9865-0.6112663*T^1+0.004307492*T^2+2.385629E-6*T^3-9.319928E-8*T^4';  ...
'100.0' '273.0' '114.4634-0.07385023*T^1-3.37167E-4*T^2+4.432826E-6*T^3-1.303897E-8*T^4+1.09916E-11*T^5';  ...
'273.0' '337.0' '296.1808-2.610472*T^1+0.01292855*T^2-2.714074E-5*T^3+1.957806E-8*T^4'});
model.component('comp1').material('mat6').propertyGroup('def').func('res_solid_1').set('arg', 'T');
model.component('comp1').material('mat6').propertyGroup('def').func('res_solid_1').set('pieces', {'1.0' '5.6' ...
    '6.94E-12+2.721429E-12*T^1-1.428571E-12*T^2+2.5E-13*T^3';  ...
'5.6' '17.0' '7.187017E-11-2.871008E-11*T^1+3.196071E-12*T^2+5.622896E-14*T^3';  ...
'17.0' '58.6' '-5.804623E-10+2.361562E-11*T^1+3.744807E-12*T^2-2.436224E-14*T^3';  ...
'58.6' '336.0' '9.85994E-11+8.21212E-11*T^1+1.427422E-12*T^2-5.793066E-15*T^3+9.642156E-18*T^4'});
model.component('comp1').material('mat6').propertyGroup('def').func('Syfunc_solid_tested_at_195K_8').set('arg', 'epe');
model.component('comp1').material('mat6').propertyGroup('def').func('Syfunc_solid_tested_at_195K_8').set('pieces', {'0.0' '0.26' ...
    '310926.5+2478539.0*epe^1-3649675.0*epe^2+1533861.0*epe^3'});
model.component('comp1').material('mat6').propertyGroup('def').func('Syt').set('arg', 'T');
model.component('comp1').material('mat6').propertyGroup('def').func('Syt').set('pieces', {'5.0' '10.0' ...
    '1.057693E7-2491807.0*T^1+534385.5*T^2-52475.55*T^3+2255.125*T^4-34.64048*T^5';  ...
'10.0' '20.0' '-2.309365E7+1.063972E7*T^1-1414767.0*T^2+85199.73*T^3-2383.933*T^4+25.21833*T^5';  ...
'20.0' '65.0' '-1856289.0+621900.6*T^1-18706.67*T^2+211.8753*T^3-0.8293287*T^4';  ...
'65.0' '195.0' '6823541.0-85137.6*T^1+442.1314*T^2-0.8750862*T^3'});
model.component('comp1').material('mat6').propertyGroup('def').func('C_solid_1').set('arg', 'T');
model.component('comp1').material('mat6').propertyGroup('def').func('C_solid_1').set('pieces', {'0.0' '1.8' ...
    '0.0468054*T^1+3.193572E-11*T^2+0.06803414*T^3+1.216601E-11*T^4+0.001304416*T^5';  ...
'1.8' '3.0' '-2.595532+3.512074*T^1-1.146209*T^2-0.02025602*T^3+0.05830843*T^4';  ...
'3.0' '5.0' '7.58581-6.425737*T^1+1.585278*T^2-0.02790238*T^3-2.555793E-4*T^4-5.115287E-6*T^5';  ...
'5.0' '40.0' '21.7639-12.58592*T^1+2.517692*T^2-0.08921844*T^3+0.001310724*T^4-6.945858E-6*T^5';  ...
'40.0' '140.0' '-38.20099+21.76232*T^1-0.2807147*T^2+0.001673195*T^3-3.737921E-6*T^4';  ...
'140.0' '336.6' '737.0564-2.830118*T^1+0.02801372*T^2-1.024661E-4*T^3+1.381151E-7*T^4'});
model.component('comp1').material('mat6').propertyGroup('def').func('sigma_solid_1').set('arg', 'T');
model.component('comp1').material('mat6').propertyGroup('def').func('sigma_solid_1').set('pieces', {'1.0' '5.6' ...
    '1/(2.500000E-13*T^3-1.428571E-12*T^2+2.721429E-12*T+6.940000E-12)';  ...
'5.6' '17.0' '1/(5.622896E-14*T^3+3.196071E-12*T^2-2.871008E-11*T+7.187017E-11)';  ...
'17.0' '58.6' '1/(-2.436224E-14*T^3+3.744807E-12*T^2+2.361562E-11*T-5.804623E-10)';  ...
'58.6' '336.0' '1/(9.642156E-18*T^4-5.793066E-15*T^3+1.427422E-12*T^2+8.212120E-11*T+0.985994E-10)'});
model.component('comp1').material('mat6').propertyGroup('def').func('HC_solid_1').set('arg', 'T');
model.component('comp1').material('mat6').propertyGroup('def').func('HC_solid_1').set('pieces', {'0.0' '1.8' ...
    '0.001829998*T^1+1.248623E-12*T^2+0.002659999*T^3+4.756664E-13*T^4+5.100003E-5*T^5';  ...
'1.8' '3.0' '-0.1014801+0.1373151*T^1-0.04481449*T^2-7.919697E-4*T^3+0.002279742*T^4';  ...
'3.0' '5.0' '0.29659-0.2512335*T^1+0.06198119*T^2-0.001090927*T^3-9.992639E-6*T^4-1.999975E-7*T^5';  ...
'5.0' '40.0' '0.8509252-0.4920844*T^1+0.09843672*T^2-0.003488262*T^3+5.124668E-5*T^4-2.715692E-7*T^5';  ...
'40.0' '140.0' '-1.493582+0.8508633*T^1-0.01097539*T^2+6.54186E-5*T^3-1.461452E-7*T^4';  ...
'140.0' '336.6' '28.81743-0.1106519*T^1+0.00109528*T^2-4.006218E-6*T^3+5.400025E-9*T^4'});
model.component('comp1').material('mat6').propertyGroup('def').func('VP_solid_1').set('arg', 'T');
model.component('comp1').material('mat6').propertyGroup('def').func('VP_solid_1').set('pieces', {'293.0' '337.0' ...
    '(exp((-4.646000e+03/T+7.841810e+00)*log(10.0)))*1.333200e+02'});
model.component('comp1').material('mat6').propertyGroup('def').func('elong').set('arg', 'T');
model.component('comp1').material('mat6').propertyGroup('def').func('elong').set('pieces', {'5.0' '195.0' ...
    '42.51027+0.09061633*T^1-0.001950349*T^2+1.267534E-5*T^3'});
model.component('comp1').material('mat6').propertyGroup('def').set('thermalconductivity', {'k_solid_1(T[1/K])[W/(m*K)]' '0' '0' '0' ...
```

```
    'k_solid_1(T[1/K])[W/(m*K)]' '0' '0' '0' 'k_solid_1(T[1/K])[W/(m*K)]'});
model.component('comp1').material('mat6').propertyGroup('def').set('resistivity', {'res_solid_1(T[1/K])[ohm*m]' '0' '0' '0' ...
    'res_solid_1(T[1/K])[ohm*m]' '0' '0' '0' 'res_solid_1(T[1/K])[ohm*m]'});
model.component('comp1').material('mat6').propertyGroup('def').set('Syfunc', 'Syfunc_solid_tested_at_195K_8(epe)[Pa]');
model.component('comp1').material('mat6').propertyGroup('def').set('Syt', 'Syt(T[1/K])[Pa]');
model.component('comp1').material('mat6').propertyGroup('def').set('heatcapacity', 'C_solid_1(T[1/K])[J/(kg*K)]');
model.component('comp1').material('mat6').propertyGroup('def').set('electricconductivity', {'sigma_solid_1(T[1/K])[S/m]' '0' '0' '0' ...
    'sigma_solid_1(T[1/K])[S/m]' '0' '0' '0' 'sigma_solid_1(T[1/K])[S/m]'});
model.component('comp1').material('mat6').propertyGroup('def').set('HC', 'HC_solid_1(T[1/K])[J/(mol*K)]');
model.component('comp1').material('mat6').propertyGroup('def').set('VP', 'VP_solid_1(T[1/K])[Pa]');
model.component('comp1').material('mat6').propertyGroup('def').set('elong', 'elong(T[1/K])');
model.component('comp1').material('mat6').propertyGroup('def').addInput('temperature');
model.component('comp1').material('mat6').propertyGroup('def').addInput('effectiveplasticstrain');
model.component('comp1').material('mat6').propertyGroup('Enu').func('E').set('arg', 'T');
model.component('comp1').material('mat6').propertyGroup('Enu').func('E').set('pieces', {'4.0' '295.0' '3.36682E9-3351970.0*T^1'});
model.component('comp1').material('mat6').propertyGroup('Enu').func('nu').set('arg', 'T');
model.component('comp1').material('mat6').propertyGroup('Enu').func('nu').set('pieces', {'4.0' '295.0' '0.3274145+1.473721E-4*T^1'});
model.component('comp1').material('mat6').propertyGroup('Enu').set('youngsmodulus', 'E(T[1/K])[Pa]');
model.component('comp1').material('mat6').propertyGroup('Enu').set('poissonsratio', 'nu(T[1/K])');
model.component('comp1').material('mat6').propertyGroup('Enu').addInput('temperature');
model.component('comp1').material('mat6').propertyGroup('KG').func('mu').set('arg', 'T');
model.component('comp1').material('mat6').propertyGroup('KG').func('mu').set('pieces', {'4.0' '295.0' '1.266401E9-1360805.0*T^1'});
model.component('comp1').material('mat6').propertyGroup('KG').func('kappa_solid_1').set('arg', 'T');
model.component('comp1').material('mat6').propertyGroup('KG').func('kappa_solid_1').set('pieces', {'4.0' '295.0' '3.265582E9-654512.0*T^1'});
model.component('comp1').material('mat6').propertyGroup('KG').set('K', '');
model.component('comp1').material('mat6').propertyGroup('KG').set('G', '');
model.component('comp1').material('mat6').propertyGroup('KG').set('K', 'kappa_solid_1(T[1/K])[Pa]');
model.component('comp1').material('mat6').propertyGroup('KG').set('G', 'mu(T[1/K])[Pa]');
model.component('comp1').material('mat6').propertyGroup('KG').addInput('temperature');
model.component('comp1').material('mat6').propertyGroup('ElastoplasticModel').func('Sys').set('arg', 'T');
model.component('comp1').material('mat6').propertyGroup('ElastoplasticModel').func('Sys').set('pieces', {'5.0' '25.0' ...
    '5003534.0-276355.7*T^1+3953.191*T^2'; '25.0' '28.0' '1.656172E7-1554593.0*T^1+49919.8*T^2-533.2133*T^3'; '28.0' '195.0' ...
    '487303.7-794.9799*T^1'});
model.component('comp1').material('mat6').propertyGroup('ElastoplasticModel').set('sigmags', '');
model.component('comp1').material('mat6').propertyGroup('ElastoplasticModel').set('Et', '');
model.component('comp1').material('mat6').propertyGroup('ElastoplasticModel').set('Ek', '');
model.component('comp1').material('mat6').propertyGroup('ElastoplasticModel').set('sigmagh', '');
model.component('comp1').material('mat6').propertyGroup('ElastoplasticModel').set('Hillcoefficients', '');
model.component('comp1').material('mat6').propertyGroup('ElastoplasticModel').set('ys', '');
model.component('comp1').material('mat6').propertyGroup('ElastoplasticModel').set('sigmags', 'Sys(T[1/K])[Pa]');
model.component('comp1').material('mat6').propertyGroup('ElastoplasticModel').set('Hillcoefficients', {'0' '0' '0' '0' '0' '0'});
model.component('comp1').material('mat6').propertyGroup('ElastoplasticModel').set('ys', {'0' '0' '0' '0' '0' '0'});
model.component('comp1').material('mat6').propertyGroup('ElastoplasticModel').addInput('temperature');
model.component('comp1').material('mat6').propertyGroup('RefractiveIndex').set('n', '');
model.component('comp1').material('mat6').propertyGroup('RefractiveIndex').set('ki', '');
model.component('comp1').material('mat6').propertyGroup('RefractiveIndex').set('n', {'4.77' '0' '0' '0' '4.77' '0' '0' '0' '4.77'});
model.component('comp1').material('mat6').propertyGroup('RefractiveIndex').set('ki', {'28.2' '0' '0' '0' '28.2' '0' '0' '0' '28.2'});
model.component('comp1').physics('ewfd').prop('EquationForm').set('freq', '1[GHz]');
model.component('comp1').physics('ewfd').prop('EquationForm').set('modeFreq', '1[GHz]');
model.component('comp1').physics('ewfd').prop('MeshControl').set('EnableMeshControl', false);
model.component('comp1').physics('ewfd').feature('port1').set('Pin', '1[mW]');
model.component('comp1').physics('ewfd').feature('port1').set('beta', 'abs(ewfd.k0)');
model.component('comp1').physics('ewfd').feature('port1').set('E0', {'0'; 'exp(-j*ewfd.k0*z)'; '0'});
model.component('comp1').physics('ewfd').feature('port2').set('beta', 'abs(ewfd.k0)');
model.component('comp1').physics('ewfd').feature('port2').set('E0', {'0'; 'exp(-j*ewfd.k0*z)'; '0'});
model.component('comp1').mesh('mesh1').feature('size').set('hauto', 4);
model.component('comp1').mesh('mesh1').run;
model.study.create('std1');
model.study('std1').create('freq', 'Frequency');
model.sol.create('sol1');
model.sol('sol1').study('std1');
model.sol('sol1').attach('std1');
model.sol('sol1').create('st1', 'StudyStep');
model.sol('sol1').create('v1', 'Variables');
model.sol('sol1').create('s1', 'Stationary');
model.sol('sol1').feature('s1').create('p1', 'Parametric');
model.sol('sol1').feature('s1').create('fc1', 'FullyCoupled');
model.sol('sol1').feature('s1').create('i1', 'Iterative');
model.sol('sol1').feature('s1').feature('i1').create('mg1', 'Multigrid');
model.sol('sol1').feature('s1').feature('i1').feature('mg1').feature('pr').create('sv1', 'SORVector');
model.sol('sol1').feature('s1').feature('i1').feature('mg1').feature('po').create('sv1', 'SORVector');
model.sol('sol1').feature('s1').feature.remove('fcDef');
model.result.numerical.create('int1', 'IntSurface');
model.result.numerical('int1').selection.set([13 29]);
model.result.numerical('int1').set('probetag', 'none');
model.result.create('pg1', 'PlotGroup3D');
model.result.create('pg2', 'PlotGroup3D');
model.result.create('pg3', 'PlotGroup1D');
model.result('pg1').create('mslc1', 'Multislice');
model.result('pg2').create('slc1', 'Slice');
model.result('pg2').feature('slc1').create('def1', 'Deform');
model.result('pg3').create('tblp1', 'Table');
model.study('std1').feature('freq').set('punit', 'Hz');
model.study('std1').feature('freq').set('plist', '1.0e12');
model.sol('sol1').attach('std1');
model.sol('sol1').feature('v1').set('clistctrl', {'p1'});
```

```
model.sol('sol1').feature('v1').set('cname', {'freq'});
model.sol('sol1').feature('v1').set('clist', {'1.0e12[Hz]'});
model.sol('sol1').feature('s1').feature('p1').set('pname', {'freq'});
model.sol('sol1').feature('s1').feature('p1').set('plistarr', {'1.0e12'});
model.sol('sol1').feature('s1').feature('p1').set('punit', {'Hz'});
model.sol('sol1').feature('s1').feature('p1').set('pcontinuationmode', 'no');
model.sol('sol1').feature('s1').feature('p1').set('preusesol', 'auto');
model.sol('sol1').feature('s1').feature('i1').label('Suggested Iterative Solver (ewfd)');
model.sol('sol1').feature('s1').feature('i1').set('linsolver', 'bicgstab');
model.sol('sol1').feature('s1').feature('i1').feature('mg1').feature('pr').feature('sv1').set('sorvecdof', {'comp1_E'});
model.sol('sol1').feature('s1').feature('i1').feature('mg1').feature('po').feature('sv1').set('sorvecdof', {'comp1_E'});
model.sol('sol1').runAll;
model.result.numerical('int1').set('table', 'tbl1');
model.result.numerical('int1').set('expr', {'ewfd.normE'});
model.result.numerical('int1').set('unit', {'V*m'});
model.result.numerical('int1').set('descr', {'Electric field norm'});
model.result.numerical('int1').setResult;
model.result('pg1').label('Electric Field (ewfd)');
model.result('pg1').set('frametype', 'spatial');
model.result('pg1').feature('mslc1').set('multiplanezmethod', 'coord');
model.result('pg1').feature('mslc1').set('zcoord', '1050.0');
model.result('pg1').feature('mslc1').set('rangecoloractive', true);
model.result('pg1').feature('mslc1').set('rangecolormin', 41923.82486663746);
model.result('pg1').feature('mslc1').set('rangecolormax', 400000);
model.result('pg1').feature('mslc1').set('smooth', 'internal');
model.result('pg1').feature('mslc1').set('resolution', 'normal');
model.result('pg2').feature('slc1').set('quickplane', 'xy');
model.result('pg2').feature('slc1').set('quickzmethod', 'coord');
model.result('pg2').feature('slc1').set('quickz', '1033.0');
model.result('pg2').feature('slc1').set('rangecoloractive', true);
model.result('pg2').feature('slc1').set('rangecolormin', 9292.73707567522);
model.result('pg2').feature('slc1').set('rangecolormax', 250000);
model.result('pg2').feature('slc1').set('smooth', 'internal');
model.result('pg2').feature('slc1').set('resolution', 'normal');
model.result('pg2').feature('slc1').feature('def1').active(false);
model.result('pg2').feature('slc1').feature('def1').set('expr', {'+ewfd.Ex' '+ewfd.Ey' '+ewfd.Ez'});
model.result('pg2').feature('slc1').feature('def1').set('descr', ' + Electric field');
model.result('pg2').feature('slc1').feature('def1').set('scale', 2.085880595474555E-4);
model.result('pg2').feature('slc1').feature('def1').set('scaleactive', false);
model.result('pg3').set('data', 'none');
model.result('pg3').set('xlabel', 'freq (Hz)');
model.result('pg3').set('ylabel', 'Electric field norm (V*m)');
model.result('pg3').set('xlabelactive', false);
model.result('pg3').set('ylabelactive', false);
model.result('pg3').feature('tblp1').set('linewidth', 3);
model.result('pg3').feature('tblp1').set('linemarker', 'cycle');
model.result('pg3').feature('tblp1').set('markerpos', 'datapoints');
end
```

```
function [] = fKTa(model)
model.component('comp1').material.create('mat1', 'Common');
model.component('comp1').material.create('mat5', 'Common');
model.component('comp1').material.create('mat7', 'Common');
model.component('comp1').material('mat1').propertyGroup('def').func.create('eta', 'Piecewise');
model.component('comp1').material('mat1').propertyGroup('def').func.create('Cp', 'Piecewise');
model.component('comp1').material('mat1').propertyGroup('def').func.create('rho', 'Analytic');
model.component('comp1').material('mat1').propertyGroup('def').func.create('k', 'Piecewise');
model.component('comp1').material('mat1').propertyGroup('def').func.create('cs', 'Analytic');
model.component('comp1').material('mat1').propertyGroup.create('RefractiveIndex', 'Refractive index');
model.component('comp1').material('mat5').selection.set([3]);
model.component('comp1').material('mat5').propertyGroup.create('RefractiveIndex', 'Refractive index');
model.component('comp1').material('mat5').propertyGroup('RefractiveIndex').func.create('int1', 'Interpolation');
model.component('comp1').material('mat5').propertyGroup('RefractiveIndex').func.create('int2', 'Interpolation');
model.component('comp1').material('mat7').selection.set([4]);
model.component('comp1').material('mat7').propertyGroup('def').func.create('k_solid_1', 'Piecewise');
model.component('comp1').material('mat7').propertyGroup('def').func.create('res_solid_1', 'Piecewise');
model.component('comp1').material('mat7').propertyGroup('def').func.create('Syfunc_solid_tested_at_195K_8', 'Piecewise');
model.component('comp1').material('mat7').propertyGroup('def').func.create('Syt', 'Piecewise');
model.component('comp1').material('mat7').propertyGroup('def').func.create('C_solid_1', 'Piecewise');
model.component('comp1').material('mat7').propertyGroup('def').func.create('sigma_solid_1', 'Piecewise');
model.component('comp1').material('mat7').propertyGroup('def').func.create('HC_solid_1', 'Piecewise');
model.component('comp1').material('mat7').propertyGroup('def').func.create('VP_solid_1', 'Piecewise');
model.component('comp1').material('mat7').propertyGroup('def').func.create('elong', 'Piecewise');
model.component('comp1').material('mat7').propertyGroup.create('Enu', 'Young''s modulus and Poisson''s ratio');
model.component('comp1').material('mat7').propertyGroup('Enu').func.create('E', 'Piecewise');
model.component('comp1').material('mat7').propertyGroup('Enu').func.create('nu', 'Piecewise');
model.component('comp1').material('mat7').propertyGroup.create('KG', 'Bulk modulus and shear modulus');
model.component('comp1').material('mat7').propertyGroup('KG').func.create('mu', 'Piecewise');
model.component('comp1').material('mat7').propertyGroup('KG').func.create('kappa_solid_1', 'Piecewise');
model.component('comp1').material('mat7').propertyGroup.create('ElastoplasticModel', 'Elastoplastic material model');
model.component('comp1').material('mat7').propertyGroup('ElastoplasticModel').func.create('Sys', 'Piecewise');
model.component('comp1').material('mat7').propertyGroup.create('RefractiveIndex', 'Refractive index');
model.component('comp1').physics.create('ewfd', 'ElectromagneticWavesFrequencyDomain', 'geom1');
model.component('comp1').physics('ewfd').create('sctr1', 'Scattering', 2);
```

```
model.component('comp1').physics('ewfd').feature('sctr1').selection.set([1 2 3 4 5 7 8 9 43 44]);
model.component('comp1').physics('ewfd').create('port1', 'Port', 2);
model.component('comp1').physics('ewfd').feature('port1').selection.set([7]);
model.component('comp1').physics('ewfd').create('port2', 'Port', 2);
model.component('comp1').physics('ewfd').feature('port2').selection.set([3]);
model.component('comp1').mesh('mesh1').create('ftet1', 'FreeTet');
model.result.table('tbl1').comments('Surface Integration 1 (ewfd.normE)');
model.result.table('evl3').label('Evaluation 3D');
model.result.table('evl3').comments('Interactive 3D values');
model.component('comp1').view('view1').set('renderwireframe', true);
model.component('comp1').view('view2').axis.set('xmin', -37.6529541015625);
model.component('comp1').view('view2').axis.set('xmax', 2112.7470703125);
model.component('comp1').view('view2').axis.set('ymin', 0);
model.component('comp1').view('view2').axis.set('ymax', 1651.199951171875);
model.component('comp1').material('mat1').label('Air');
model.component('comp1').material('mat1').set('family', 'air');
model.component('comp1').material('mat1').propertyGroup('def').func('eta').set('arg', 'T');
model.component('comp1').material('mat1').propertyGroup('def').func('eta').set('pieces', {'200.0' '1600.0' ...
    '-8.38278E-7+8.35717342E-8*T^1-7.69429583E-11*T^2+4.6437266E-14*T^3-1.06585607E-17*T^4'});
model.component('comp1').material('mat1').propertyGroup('def').func('Cp').set('arg', 'T');
model.component('comp1').material('mat1').propertyGroup('def').func('Cp').set('pieces', {'200.0' '1600.0' ...
    '1047.63657-0.372589265*T^1+9.45304214E-4*T^2-6.02409443E-7*T^3+1.2858961E-10*T^4'});
model.component('comp1').material('mat1').propertyGroup('def').func('rho').set('expr', 'pA*0.02897/8.314/T');
model.component('comp1').material('mat1').propertyGroup('def').func('rho').set('args', {'pA' 'T'});
model.component('comp1').material('mat1').propertyGroup('def').func('rho').set('dermethod', 'manual');
model.component('comp1').material('mat1').propertyGroup('def').func('rho').set('argders', {'pA' 'd(pA*0.02897/8.314/T,pA)' 'T' ...
    'd(pA*0.02897/8.314/T,T)'});
model.component('comp1').material('mat1').propertyGroup('def').func('rho').set('plotargs', {'pA' '0' '1'; 'T' '0' '1'});
model.component('comp1').material('mat1').propertyGroup('def').func('k').set('arg', 'T');
model.component('comp1').material('mat1').propertyGroup('def').func('k').set('pieces', {'200.0' '1600.0' ...
    '-0.00227583562+1.15480022E-4*T^1-7.90252856E-8*T^2+4.11702505E-11*T^3-7.43864331E-15*T^4'});
model.component('comp1').material('mat1').propertyGroup('def').func('cs').set('expr', 'sqrt(1.4*287*T)');
model.component('comp1').material('mat1').propertyGroup('def').func('cs').set('args', {'T'});
model.component('comp1').material('mat1').propertyGroup('def').func('cs').set('dermethod', 'manual');
model.component('comp1').material('mat1').propertyGroup('def').func('cs').set('argders', {'T' 'd(sqrt(1.4*287*T),T)'});
model.component('comp1').material('mat1').propertyGroup('def').func('cs').set('plotargs', {'T' '0' '1'});
model.component('comp1').material('mat1').propertyGroup('def').set('relpermeability', {'1' '0' '0' '0' '1' '0' '0' '0' '1'});
model.component('comp1').material('mat1').propertyGroup('def').set('relpermittivity', {'1' '0' '0' '0' '1' '0' '0' '0' '1'});
model.component('comp1').material('mat1').propertyGroup('def').set('dynamicviscosity', 'eta(T[1/K])[Pa*s]');
model.component('comp1').material('mat1').propertyGroup('def').set('ratioofspecificheat', '1.4');
model.component('comp1').material('mat1').propertyGroup('def').set('electricconductivity', {'0[S/m]' '0' '0' '0' '0[S/m]' '0' '0' '0' '0[S/m]'});
model.component('comp1').material('mat1').propertyGroup('def').set('heatcapacity', 'Cp(T[1/K])[J/(kg*K)]');
model.component('comp1').material('mat1').propertyGroup('def').set('density', 'rho(pA[1/Pa],T[1/K])[kg/m^3]');
model.component('comp1').material('mat1').propertyGroup('def').set('thermalconductivity', {'k(T[1/K])[W/(m*K)]' '0' '0' '0' 'k(T[1/K])[W/(m*K)]' '0' '0' ...
    '0' 'k(T[1/K])[W/(m*K)]'});
model.component('comp1').material('mat1').propertyGroup('def').set('soundspeed', 'cs(T[1/K])[m/s]');
model.component('comp1').material('mat1').propertyGroup('def').addInput('temperature');
model.component('comp1').material('mat1').propertyGroup('def').addInput('pressure');
model.component('comp1').material('mat1').propertyGroup('RefractiveIndex').set('n', '');
model.component('comp1').material('mat1').propertyGroup('RefractiveIndex').set('ki', '');
model.component('comp1').material('mat1').propertyGroup('RefractiveIndex').set('n', {'1' '0' '0' '0' '1' '0' '0' '0' '1'});
model.component('comp1').material('mat1').propertyGroup('RefractiveIndex').set('ki', {'0' '0' '0' '0' '0' '0' '0' '0' '0'});
model.component('comp1').material('mat5').label(['Ta (Tantalum) (Ordal et al. 1988: n,k 0.667-125 ' native2unicode(hex2dec({'00' 'b5'}), ...
    'unicode') 'm)']);
model.component('comp1').material('mat5').set('groups', {'inorganic' 'Inorganic Materials'; 'ta_tantalum_and_tantalates' ...
    'Ta - Tantalum and Tantalates'; 'experimental_data' 'Experimental data'});
model.component('comp1').material('mat5').propertyGroup('RefractiveIndex').func('int1').label('Refractive index');
model.component('comp1').material('mat5').propertyGroup('RefractiveIndex').func('int1').set('funcname', 'n_interp');
model.component('comp1').material('mat5').propertyGroup('RefractiveIndex').func('int1').set('table', {'6.67E-7' '1.4478379';'7.14E-7' '1.2685432';  ...
'7.69E-7' '1.1466787';'8.329999999999999E-7' '1.073087';'9.09E-7' '1.0150307';'1.0E-6' '0.9821499';'1.05E-6' '0.96032';'1.11E-6' '0.9253053';  ...
'1.18E-6' '0.9012078';'1.2499999999999999E-6' '0.8740663';'1.33E-6' '0.8414308';'1.4299999999999999E-6' '0.8354757';  ...
'1.5399999999999999E-6' '0.8564461';'1.6699999999999999E-6' '0.9019896';'1.82E-6' '0.960826';'2.0E-6' '1.0326738';  ...
'2.1099999999999997E-6' '1.1067824';'2.22E-6' '1.192353';'2.35E-6' '1.2678805';'2.4999999999999998E-6' '1.3499202';  ...
'2.67E-6' '1.4802727';'2.8599999999999997E-6' '1.6348171';'3.0799999999999997E-6' '1.8209204';'3.33E-6' '2.0587589';  ...
'3.64E-6' '2.3469537';'4.0E-6' '2.6799222';'4.44E-6' '3.1932058';'4.999999999999996E-6' '3.9020861';'5.709999999999995E-6' '4.9008382';  ...
'6.67E-6' '6.4122619';'8.0E-6' '8.7691085';'9.999999999999999E-6' '12.931507';'1.1099999999999999E-5' '15.61576';  ...
'1.2499999999999999E-5' '19.058763';'1.43E-5' '24.23852';'1.67E-5' '31.493393';'1.9999999999999998E-5' '41.721911';  ...
'2.2199999999999998E-5' '48.392026';'2.4999999999999998E-5' '56.42849';'2.86E-5' '66.1388';'3.3299999999999996E-5' '77.867703';  ...
'3.9999999999999996E-5' '92.016211';'4.4399999999999995E-5' '100.17673';'4.9999999999999996E-5' '109.23359';  ...
'5.71E-5' '119.43744';'6.67E-5' '131.25365';'7.999999999999999E-5' '145.5884';'9.999999999999999E-5' '164.07966';'1.25E-4' '184.49663'});
model.component('comp1').material('mat5').propertyGroup('RefractiveIndex').func('int1').set('interp', 'piecewisecubic');
model.component('comp1').material('mat5').propertyGroup('RefractiveIndex').func('int1').set('argunit', 'm');
model.component('comp1').material('mat5').propertyGroup('RefractiveIndex').func('int1').set('fununit', '1');
model.component('comp1').material('mat5').propertyGroup('RefractiveIndex').func('int2').label('Refractive index, imaginary part');
model.component('comp1').material('mat5').propertyGroup('RefractiveIndex').func('int2').set('funcname', 'k_interp');
model.component('comp1').material('mat5').propertyGroup('RefractiveIndex').func('int2').set('table', {'6.67E-7' '2.3306224';'7.14E-7' '2.7919609';  ...
'7.69E-7' '3.2384897';'8.329999999999999E-7' '3.7362305';'9.09E-7' '4.2736932';'1.0E-6' '4.8873028';'1.05E-6' '5.2167868';  ...
'1.11E-6' '5.5894492';'1.18E-6' '6.0121558';'1.2499999999999999E-6' '6.4776281';'1.33E-6' '7.0171487';'1.429999999999999E-6' '7.6548582';  ...
'1.5399999999999999E-6' '8.3686838';'1.6699999999999999E-6' '9.1771344';'1.82E-6' '10.114648';'2.0E-6' '11.243595';  ...
'2.1099999999999997E-6' '11.896174';'2.22E-6' '12.595572';'2.35E-6' '13.363144';'2.4999999999999998E-6' '14.23958';  ...
'2.67E-6' '15.240124';'2.8599999999999997E-6' '16.359682';'3.0799999999999997E-6' '17.640057';'3.33E-6' '19.120563';  ...
'3.64E-6' '20.845754';'4.0E-6' '22.915456';'4.44E-6' '25.461819';'4.999999999999996E-6' '28.57704';'5.7099999999999995E-6' '32.510917';  ...
'6.67E-6' '37.643719';'8.0E-6' '44.613332';'9.999999999999999E-6' '54.673721';'1.1099999999999999E-5' '59.926459';  ...
'1.2499999999999999E-5' '66.350003';'1.43E-5' '73.949845';'1.67E-5' '82.79423';'1.9999999999999998E-5' '93.122866';  ...
'2.2199999999999998E-5' '98.921392';'2.4999999999999998E-5' '105.16669';'2.86E-5' '111.86857';'3.3299999999999996E-5' '119.08288';  ...
```

```
'3.9999999999999996E-5' '127.08528';'4.4399999999999995E-5' '131.63028';'4.9999999999999996E-5' '136.82885';  ...
'5.71E-5' '143.08774'; '6.67E-5' '151.08524';'7.999999999999999E-5' '161.9815';'9.999999999999999E-5' '177.86128';'1.25E-4' '196.87085'});
model.component('comp1').material('mat5').propertyGroup('RefractiveIndex').func('int2').set('interp', 'piecewisecubic');
model.component('comp1').material('mat5').propertyGroup('RefractiveIndex').func('int2').set('argunit', 'm');
model.component('comp1').material('mat5').propertyGroup('RefractiveIndex').func('int2').set('fununit', '1');
model.component('comp1').material('mat5').propertyGroup('RefractiveIndex').set('n', '');
model.component('comp1').material('mat5').propertyGroup('RefractiveIndex').set('ki', '');
model.component('comp1').material('mat5').propertyGroup('RefractiveIndex').set('n', '');
model.component('comp1').material('mat5').propertyGroup('RefractiveIndex').set('ki', '');
model.component('comp1').material('mat5').propertyGroup('RefractiveIndex').set('n', {'n_interp(c_const/freq)' '0' '0' '0' 'n_interp(c_const/freq)' '0' ...
    '0' '0' 'n_interp(c_const/freq)'});
model.component('comp1').material('mat5').propertyGroup('RefractiveIndex').set('ki', {'k_interp(c_const/freq)' '0' '0' '0' 'k_interp(c_const/freq)' '0' ...
    '0' '0' 'k_interp(c_const/freq)'});
model.component('comp1').material('mat5').propertyGroup('RefractiveIndex').addInput('frequency');
model.component('comp1').material('mat7').label('Potassium [solid,tested at -78C (195K)]');
model.component('comp1').material('mat7').set('family', 'custom');
model.component('comp1').material('mat7').set('specular', 'custom');
model.component('comp1').material('mat7').set('customspecular', [0.7843137254901961 1 1]);
model.component('comp1').material('mat7').set('diffuse', 'custom');
model.component('comp1').material('mat7').set('customdiffuse', [0.7843137254901961 0.7843137254901961 0.7843137254901961]);
model.component('comp1').material('mat7').set('ambient', 'custom');
model.component('comp1').material('mat7').set('customambient', [0.7843137254901961 0.7843137254901961 0.7843137254901961]);
model.component('comp1').material('mat7').set('noise', true);
model.component('comp1').material('mat7').set('noisefreq', 1);
model.component('comp1').material('mat7').set('lighting', 'cooktorrance');
model.component('comp1').material('mat7').set('fresnel', 0.9);
model.component('comp1').material('mat7').propertyGroup('def').func('k_solid_1').set('arg', 'T');
model.component('comp1').material('mat7').propertyGroup('def').func('k_solid_1').set('pieces', {'0.0' '6.0' ...
    '1122.085*T^1+263.7268*T^2-227.3021*T^3+38.40137*T^4-2.033789*T^5';  ...
'6.0' '20.0' '6265.378-1779.74*T^1+224.4572*T^2-14.64801*T^3+0.4827659*T^4-0.006357878*T^5';  ...
'20.0' '50.0' '1340.183-148.6854*T^1+7.279199*T^2-0.1780719*T^3+0.002163925*T^4-1.042377E-5*T^5';  ...
'50.0' '100.0' '131.9865-0.6112663*T^1+0.004307492*T^2+2.385629E-6*T^3-9.319928E-8*T^4';  ...
'100.0' '273.0' '114.4634-0.07385023*T^1-3.37167E-4*T^2+4.432826E-6*T^3-1.303897E-8*T^4+1.09916E-11*T^5';  ...
'273.0' '337.0' '296.1808-2.610472*T^1+0.01292855*T^2-2.714074E-5*T^3+1.957806E-8*T^4'});
model.component('comp1').material('mat7').propertyGroup('def').func('res_solid_1').set('arg', 'T');
model.component('comp1').material('mat7').propertyGroup('def').func('res_solid_1').set('pieces', {'1.0' '5.6' ...
    '6.94E-12+2.721429E-12*T^1-1.428571E-12*T^2+2.5E-13*T^3';  ...
'5.6' '17.0' '7.187017E-11-2.871008E-11*T^1+3.196071E-12*T^2+5.622896E-14*T^3';  ...
'17.0' '58.6' '-5.804623E-10+2.361562E-11*T^1+3.744807E-12*T^2-2.436224E-14*T^3';  ...
'58.6' '336.0' '9.85994E-11+8.21212E-11*T^1+1.427422E-12*T^2-5.793066E-15*T^3+9.642156E-18*T^4'});
model.component('comp1').material('mat7').propertyGroup('def').func('Syfunc_solid_tested_at_195K_8').set('arg', 'epe');
model.component('comp1').material('mat7').propertyGroup('def').func('Syfunc_solid_tested_at_195K_8').set('pieces', {'0.0' '0.26' ...
    '310926.5+2478539.0*epe^1-3649675.0*epe^2+1533861.0*epe^3'});
model.component('comp1').material('mat7').propertyGroup('def').func('Syt').set('arg', 'T');
model.component('comp1').material('mat7').propertyGroup('def').func('Syt').set('pieces', {'5.0' '10.0' ...
    '1.057693E7-2491807.0*T^1+534385.5*T^2-52475.55*T^3+2255.125*T^4-34.64048*T^5';  ...
'10.0' '20.0' '-2.309365E7+1.063972E7*T^1-1414767.0*T^2+85199.73*T^3-2383.933*T^4+25.21833*T^5';  ...
'20.0' '65.0' '-1856289.0+621900.6*T^1-18706.67*T^2+211.8753*T^3-0.8293287*T^4';  ...
'65.0' '195.0' '6823541.0-85137.6*T^1+442.1314*T^2-0.8750862*T^3'});
model.component('comp1').material('mat7').propertyGroup('def').func('C_solid_1').set('arg', 'T');
model.component('comp1').material('mat7').propertyGroup('def').func('C_solid_1').set('pieces', {'0.0' '1.8' ...
    '0.0468054*T^1+3.193572E-11*T^2+0.06803414*T^3+1.216601E-11*T^4+0.001304416*T^5';  ...
'1.8' '3.0' '-2.595532+3.512074*T^1-1.146209*T^2-0.02025602*T^3+0.05830843*T^4';  ...
'3.0' '5.0' '7.58581-6.425737*T^1+1.585278*T^2-0.02790238*T^3-2.555793E-4*T^4-5.115287E-6*T^5';  ...
'5.0' '40.0' '21.7639-12.58592*T^1+2.517692*T^2-0.08921844*T^3+0.001310724*T^4-6.945858E-6*T^5';  ...
'40.0' '140.0' '-38.20099+21.76232*T^1-0.2807147*T^2+0.001673195*T^3-3.737921E-6*T^4';  ...
'140.0' '336.6' '737.0564-2.830118*T^1+0.02801372*T^2-1.024661E-4*T^3+1.381151E-7*T^4'});
model.component('comp1').material('mat7').propertyGroup('def').func('sigma_solid_1').set('arg', 'T');
model.component('comp1').material('mat7').propertyGroup('def').func('sigma_solid_1').set('pieces', {'1.0' '5.6' ...
    '1/(2.500000E-13*T^3-1.428571E-12*T^2+2.721429E-12*T+6.940000E-12)';  ...
'5.6' '17.0' '1/(5.622896E-14*T^3+3.196071E-12*T^2-2.871008E-11*T+7.187017E-11)';  ...
'17.0' '58.6' '1/(-2.436224E-14*T^3+3.744807E-12*T^2+2.361562E-11*T-5.804623E-10)';  ...
'58.6' '336.0' '1/(9.642156E-18*T^4-5.793066E-15*T^3+1.427422E-12*T^2+8.212120E-11*T+0.985994E-10)'});
model.component('comp1').material('mat7').propertyGroup('def').func('HC_solid_1').set('arg', 'T');
model.component('comp1').material('mat7').propertyGroup('def').func('HC_solid_1').set('pieces', {'0.0' '1.8' ...
    '0.001829998*T^1+1.248623E-12*T^2+0.002659999*T^3+4.756664E-13*T^4+5.100003E-5*T^5';  ...
'1.8' '3.0' '-0.1014801+0.1373151*T^1-0.04481449*T^2-7.919697E-4*T^3+0.002279742*T^4';  ...
'3.0' '5.0' '0.29659-0.2512335*T^1+0.06198119*T^2-0.001090927*T^3-9.992639E-6*T^4-1.999975E-7*T^5';  ...
'5.0' '40.0' '0.8509252-0.4920844*T^1+0.09843672*T^2-0.003488262*T^3+5.124668E-5*T^4-2.715692E-7*T^5';  ...
'40.0' '140.0' '-1.493582+0.8508633*T^1-0.01097539*T^2+6.54186E-5*T^3-1.461452E-7*T^4';  ...
'140.0' '336.6' '28.81743-0.1106519*T^1+0.00109528*T^2-4.006218E-6*T^3+5.400025E-9*T^4'});
model.component('comp1').material('mat7').propertyGroup('def').func('VP_solid_1').set('arg', 'T');
model.component('comp1').material('mat7').propertyGroup('def').func('VP_solid_1').set('pieces', {'293.0' '337.0' ...
    '(exp((-4.646000e+03/T+7.841810e+00)*log(10.0)))*1.333200e+02'});
model.component('comp1').material('mat7').propertyGroup('def').func('elong').set('arg', 'T');
model.component('comp1').material('mat7').propertyGroup('def').func('elong').set('pieces', {'5.0' '195.0' ...
    '42.51027+0.09061633*T^1-0.001950349*T^2+1.267534E-5*T^3'});
model.component('comp1').material('mat7').propertyGroup('def').set('thermalconductivity', {'k_solid_1(T[1/K])[W/(m*K)]' '0' '0' '0' ...
    'k_solid_1(T[1/K])[W/(m*K)]' '0' '0' '0' 'k_solid_1(T[1/K])[W/(m*K)]'});
model.component('comp1').material('mat7').propertyGroup('def').set('resistivity', {'res_solid_1(T[1/K])[ohm*m]' '0' '0' '0' ...
    'res_solid_1(T[1/K])[ohm*m]' '0' '0' '0' 'res_solid_1(T[1/K])[ohm*m]'});
model.component('comp1').material('mat7').propertyGroup('def').set('Syfunc', 'Syfunc_solid_tested_at_195K_8(epe)[Pa]');
model.component('comp1').material('mat7').propertyGroup('def').set('Syt', 'Syt(T[1/K])[Pa]');
model.component('comp1').material('mat7').propertyGroup('def').set('heatcapacity', 'C_solid_1(T[1/K])[J/(kg*K)]');
model.component('comp1').material('mat7').propertyGroup('def').set('electricconductivity', {'sigma_solid_1(T[1/K])[S/m]' '0' '0' '0' ...
```

```
    'sigma_solid_1(T[1/K])[S/m]' '0' '0' '0' 'sigma_solid_1(T[1/K])[S/m]'});
model.component('comp1').material('mat7').propertyGroup('def').set('HC', 'HC_solid_1(T[1/K])[J/(mol*K)]');
model.component('comp1').material('mat7').propertyGroup('def').set('VP', 'VP_solid_1(T[1/K])[Pa]');
model.component('comp1').material('mat7').propertyGroup('def').set('elong', 'elong(T[1/K])');
model.component('comp1').material('mat7').propertyGroup('def').addInput('temperature');
model.component('comp1').material('mat7').propertyGroup('def').addInput('effectiveplasticstrain');
model.component('comp1').material('mat7').propertyGroup('Enu').func('E').set('arg', 'T');
model.component('comp1').material('mat7').propertyGroup('Enu').func('E').set('pieces', {'4.0' '295.0' '3.36682E9-3351970.0*T^1'});
model.component('comp1').material('mat7').propertyGroup('Enu').func('nu').set('arg', 'T');
model.component('comp1').material('mat7').propertyGroup('Enu').func('nu').set('pieces', {'4.0' '295.0' '0.3274145+1.473721E-4*T^1'});
model.component('comp1').material('mat7').propertyGroup('Enu').set('youngsmodulus', 'E(T[1/K])[Pa]');
model.component('comp1').material('mat7').propertyGroup('Enu').set('poissonsratio', 'nu(T[1/K])');
model.component('comp1').material('mat7').propertyGroup('Enu').addInput('temperature');
model.component('comp1').material('mat7').propertyGroup('KG').func('mu').set('arg', 'T');
model.component('comp1').material('mat7').propertyGroup('KG').func('mu').set('pieces', {'4.0' '295.0' '1.266401E9-1360805.0*T^1'});
model.component('comp1').material('mat7').propertyGroup('KG').func('kappa_solid_1').set('arg', 'T');
model.component('comp1').material('mat7').propertyGroup('KG').func('kappa_solid_1').set('pieces', {'4.0' '295.0' '3.265582E9-654512.0*T^1'});
model.component('comp1').material('mat7').propertyGroup('KG').set('K', '');
model.component('comp1').material('mat7').propertyGroup('KG').set('G', '');
model.component('comp1').material('mat7').propertyGroup('KG').set('K', 'kappa_solid_1(T[1/K])[Pa]');
model.component('comp1').material('mat7').propertyGroup('KG').set('G', 'mu(T[1/K])[Pa]');
model.component('comp1').material('mat7').propertyGroup('KG').addInput('temperature');
model.component('comp1').material('mat7').propertyGroup('ElastoplasticModel').func('Sys').set('arg', 'T');
model.component('comp1').material('mat7').propertyGroup('ElastoplasticModel').func('Sys').set('pieces', {'5.0' '25.0' ...
    '5003534.0-276355.7*T^1+3953.191*T^2'; '25.0' '28.0' '1.656172E7-1554593.0*T^1+49919.8*T^2-533.2133*T^3'; '28.0' '195.0' ...
    '487303.7-794.9799*T^1'});
model.component('comp1').material('mat7').propertyGroup('ElastoplasticModel').set('sigmags', '');
model.component('comp1').material('mat7').propertyGroup('ElastoplasticModel').set('Et', '');
model.component('comp1').material('mat7').propertyGroup('ElastoplasticModel').set('Ek', '');
model.component('comp1').material('mat7').propertyGroup('ElastoplasticModel').set('sigmagh', '');
model.component('comp1').material('mat7').propertyGroup('ElastoplasticModel').set('Hillcoefficients', '');
model.component('comp1').material('mat7').propertyGroup('ElastoplasticModel').set('ys', '');
model.component('comp1').material('mat7').propertyGroup('ElastoplasticModel').set('sigmags', 'Sys(T[1/K])[Pa]');
model.component('comp1').material('mat7').propertyGroup('ElastoplasticModel').set('Hillcoefficients', {'0' '0' '0' '0' '0' '0'});
model.component('comp1').material('mat7').propertyGroup('ElastoplasticModel').set('ys', {'0' '0' '0' '0' '0' '0'});
model.component('comp1').material('mat7').propertyGroup('ElastoplasticModel').addInput('temperature');
model.component('comp1').material('mat7').propertyGroup('RefractiveIndex').set('n', '');
model.component('comp1').material('mat7').propertyGroup('RefractiveIndex').set('ki', '');
model.component('comp1').material('mat7').propertyGroup('RefractiveIndex').set('n', {'4.77' '0' '0' '0' '4.77' '0' '0' '0' '4.77'});
model.component('comp1').material('mat7').propertyGroup('RefractiveIndex').set('ki', {'28.2' '0' '0' '0' '28.2' '0' '0' '0' '28.2'});
model.component('comp1').physics('ewfd').prop('EquationForm').set('freq', '1[GHz]');
model.component('comp1').physics('ewfd').prop('EquationForm').set('modeFreq', '1[GHz]');
model.component('comp1').physics('ewfd').prop('MeshControl').set('EnableMeshControl', false);
model.component('comp1').physics('ewfd').feature('port1').set('Pin', '1[mW]');
model.component('comp1').physics('ewfd').feature('port1').set('beta', 'abs(ewfd.k0)');
model.component('comp1').physics('ewfd').feature('port1').set('E0', {'0'; 'exp(-j*ewfd.k0*z)'; '0'});
model.component('comp1').physics('ewfd').feature('port2').set('beta', 'abs(ewfd.k0)');
model.component('comp1').physics('ewfd').feature('port2').set('E0', {'0'; 'exp(-j*ewfd.k0*z)'; '0'});
model.component('comp1').mesh('mesh1').feature('size').set('hauto', 4);
model.component('comp1').mesh('mesh1').run;
model.study.create('std1');
model.study('std1').create('freq', 'Frequency');
model.sol.create('sol1');
model.sol('sol1').study('std1');
model.sol('sol1').attach('std1');
model.sol('sol1').create('st1', 'StudyStep');
model.sol('sol1').create('v1', 'Variables');
model.sol('sol1').create('s1', 'Stationary');
model.sol('sol1').feature('s1').create('p1', 'Parametric');
model.sol('sol1').feature('s1').create('fc1', 'FullyCoupled');
model.sol('sol1').feature('s1').create('i1', 'Iterative');
model.sol('sol1').feature('s1').feature('i1').create('mg1', 'Multigrid');
model.sol('sol1').feature('s1').feature('i1').feature('mg1').feature('pr').create('sv1', 'SORVector');
model.sol('sol1').feature('s1').feature('i1').feature('mg1').feature('po').create('sv1', 'SORVector');
model.sol('sol1').feature('s1').feature.remove('fcDef');
model.result.numerical.create('int1', 'IntSurface');
model.result.numerical('int1').selection.set([13 29]);
model.result.numerical('int1').set('probetag', 'none');
model.result.create('pg1', 'PlotGroup3D');
model.result.create('pg2', 'PlotGroup3D');
model.result.create('pg3', 'PlotGroup1D');
model.result('pg1').create('mslc1', 'Multislice');
model.result('pg2').create('slc1', 'Slice');
model.result('pg2').feature('slc1').create('def1', 'Deform');
model.result('pg3').create('tblp1', 'Table');
model.study('std1').feature('freq').set('punit', 'Hz');
model.study('std1').feature('freq').set('plist', '1.0e12');
model.sol('sol1').attach('std1');
model.sol('sol1').feature('v1').set('clistctrl', {'p1'});
model.sol('sol1').feature('v1').set('cname', {'freq'});
model.sol('sol1').feature('v1').set('clist', {'1.0e12[Hz]'});
model.sol('sol1').feature('s1').feature('p1').set('pname', {'freq'});
model.sol('sol1').feature('s1').feature('p1').set('plistarr', {'1.0e12'});
model.sol('sol1').feature('s1').feature('p1').set('punit', {'Hz'});
model.sol('sol1').feature('s1').feature('p1').set('pcontinuationmode', 'no');
model.sol('sol1').feature('s1').feature('p1').set('preusesol', 'auto');
```

```
model.sol('sol1').feature('s1').feature('i1').label('Suggested Iterative Solver (ewfd)');
model.sol('sol1').feature('s1').feature('i1').set('linsolver', 'bicgstab');
model.sol('sol1').feature('s1').feature('i1').feature('mg1').feature('pr').feature('sv1').set('sorvecdof', {'comp1_E'});
model.sol('sol1').feature('s1').feature('i1').feature('mg1').feature('po').feature('sv1').set('sorvecdof', {'comp1_E'});
model.sol('sol1').runAll;
model.result.numerical('int1').set('table', 'tbl1');
model.result.numerical('int1').set('expr', {'ewfd.normE'});
model.result.numerical('int1').set('unit', {'V*m'});
model.result.numerical('int1').set('descr', {'Electric field norm'});
model.result.numerical('int1').setResult;
model.result('pg1').label('Electric Field (ewfd)');
model.result('pg1').set('frametype', 'spatial');
model.result('pg1').feature('mslc1').set('multiplanezmethod', 'coord');
model.result('pg1').feature('mslc1').set('zcoord', '1050.0');
model.result('pg1').feature('mslc1').set('rangecoloractive', true);
model.result('pg1').feature('mslc1').set('rangecolormin', 41923.82486663746);
model.result('pg1').feature('mslc1').set('rangecolormax', 400000);
model.result('pg1').feature('mslc1').set('smooth', 'internal');
model.result('pg1').feature('mslc1').set('resolution', 'normal');
model.result('pg2').feature('slc1').set('quickplane', 'xy');
model.result('pg2').feature('slc1').set('quickzmethod', 'coord');
model.result('pg2').feature('slc1').set('quickplane', 'xy');
model.result('pg2').feature('slc1').set('quickzmethod', 'coord');
model.result('pg2').feature('slc1').set('quickz', '1033.0');
model.result('pg2').feature('slc1').set('rangecoloractive', true);
model.result('pg2').feature('slc1').set('rangecolormin', 9292.73707567522);
model.result('pg2').feature('slc1').set('rangecolormax', 250000);
model.result('pg2').feature('slc1').set('smooth', 'internal');
model.result('pg2').feature('slc1').set('resolution', 'normal');
model.result('pg2').feature('slc1').feature('def1').active(false);
model.result('pg2').feature('slc1').feature('def1').set('expr', {'+ewfd.Ex' '+ewfd.Ey' '+ewfd.Ez'});
model.result('pg2').feature('slc1').feature('def1').set('descr', ' + Electric field');
model.result('pg2').feature('slc1').feature('def1').set('scale', 2.085880595474555E-4);
model.result('pg2').feature('slc1').feature('def1').set('scaleactive', false);
model.result('pg3').set('data', 'none');
model.result('pg3').set('xlabel', 'freq (Hz)');
model.result('pg3').set('ylabel', 'Electric field norm (V*m)');
model.result('pg3').set('xlabelactive', false);
model.result('pg3').set('ylabelactive', false);
model.result('pg3').feature('tblp1').set('linewidth', 3);
model.result('pg3').feature('tblp1').set('linemarker', 'cycle');
model.result('pg3').feature('tblp1').set('markerpos', 'datapoints');
end
```

```
function [] = fNiAg(model)
model.component('comp1').material.create('mat1', 'Common');
model.component('comp1').material.create('mat3', 'Common');
model.component('comp1').material.create('mat4', 'Common');
model.component('comp1').material('mat1').propertyGroup('def').func.create('eta', 'Piecewise');
model.component('comp1').material('mat1').propertyGroup('def').func.create('Cp', 'Piecewise');
model.component('comp1').material('mat1').propertyGroup('def').func.create('rho', 'Analytic');
model.component('comp1').material('mat1').propertyGroup('def').func.create('k', 'Piecewise');
model.component('comp1').material('mat1').propertyGroup('def').func.create('cs', 'Analytic');
model.component('comp1').material('mat1').propertyGroup.create('RefractiveIndex', 'Refractive index');
model.component('comp1').material('mat3').selection.set([3]);
model.component('comp1').material('mat3').propertyGroup('def').func.create('k_solid_1', 'Piecewise');
model.component('comp1').material('mat3').propertyGroup('def').func.create('res_solid_1', 'Piecewise');
model.component('comp1').material('mat3').propertyGroup('def').func.create('alpha', 'Piecewise');
model.component('comp1').material('mat3').propertyGroup('def').func.create('C_solid_1', 'Piecewise');
model.component('comp1').material('mat3').propertyGroup('def').func.create('sigma_solid_1', 'Piecewise');
model.component('comp1').material('mat3').propertyGroup('def').func.create('HC_solid_1', 'Piecewise');
model.component('comp1').material('mat3').propertyGroup('def').func.create('VP_solid_1', 'Piecewise');
model.component('comp1').material('mat3').propertyGroup('def').func.create('rho', 'Piecewise');
model.component('comp1').material('mat3').propertyGroup.create('ThermalExpansion', 'Thermal expansion');
model.component('comp1').material('mat3').propertyGroup('ThermalExpansion').func.create('dL', 'Piecewise');
model.component('comp1').material('mat3').propertyGroup('ThermalExpansion').func.create('CTE', 'Piecewise');
model.component('comp1').material('mat3').propertyGroup.create('Enu', 'Young''s modulus and Poisson''s ratio');
model.component('comp1').material('mat3').propertyGroup('Enu').func.create('E', 'Piecewise');
model.component('comp1').material('mat3').propertyGroup('Enu').func.create('nu', 'Piecewise');
model.component('comp1').material('mat3').propertyGroup.create('KG', 'Bulk modulus and shear modulus');
model.component('comp1').material('mat3').propertyGroup('KG').func.create('mu', 'Piecewise');
model.component('comp1').material('mat3').propertyGroup('KG').func.create('kappa', 'Piecewise');
model.component('comp1').material('mat3').propertyGroup.create('RefractiveIndex', 'Refractive index');
model.component('comp1').material('mat4').selection.set([4]);
model.component('comp1').material('mat4').info.create('Composition');
model.component('comp1').material('mat4').propertyGroup('def').func.create('k', 'Piecewise');
model.component('comp1').material('mat4').propertyGroup('def').func.create('alpha', 'Piecewise');
model.component('comp1').material('mat4').propertyGroup('def').func.create('rho', 'Piecewise');
model.component('comp1').material('mat4').propertyGroup.create('ThermalExpansion', 'Thermal expansion');
model.component('comp1').material('mat4').propertyGroup('ThermalExpansion').func.create('dL', 'Piecewise');
model.component('comp1').material('mat4').propertyGroup.create('Enu', 'Young''s modulus and Poisson''s ratio');
model.component('comp1').material('mat4').propertyGroup('Enu').func.create('E', 'Piecewise');
model.component('comp1').material('mat4').propertyGroup.create('RefractiveIndex', 'Refractive index');
model.component('comp1').physics.create('ewfd', 'ElectromagneticWavesFrequencyDomain', 'geom1');
model.component('comp1').physics('ewfd').create('sctr1', 'Scattering', 2);
model.component('comp1').physics('ewfd').feature('sctr1').selection.set([1 2 3 4 5 7 8 9 43 44]);
```

```
model.component('comp1').physics('ewfd').create('port1', 'Port', 2);
model.component('comp1').physics('ewfd').feature('port1').selection.set([7]);
model.component('comp1').physics('ewfd').create('port2', 'Port', 2);
model.component('comp1').physics('ewfd').feature('port2').selection.set([3]);
model.component('comp1').mesh('mesh1').create('ftet1', 'FreeTet');
model.result.table('tbl1').comments('Surface Integration 1 (ewfd.normE)');
model.result.table('evl3').label('Evaluation 3D');
model.result.table('evl3').comments('Interactive 3D values');
model.component('comp1').view('view1').set('renderwireframe', true);
model.component('comp1').view('view2').axis.set('xmin', -37.6529541015625);
model.component('comp1').view('view2').axis.set('xmax', 2112.7470703125);
model.component('comp1').view('view2').axis.set('ymin', 0);
model.component('comp1').view('view2').axis.set('ymax', 1651.199951171875);
model.component('comp1').material('mat1').label('Air');
model.component('comp1').material('mat1').set('family', 'air');
model.component('comp1').material('mat1').propertyGroup('def').func('eta').set('arg', 'T');
model.component('comp1').material('mat1').propertyGroup('def').func('eta').set('pieces', {'200.0' '1600.0' ...
    '-8.38278E-7+8.35717342E-8*T^1-7.69429583E-11*T^2+4.6437266E-14*T^3-1.06585607E-17*T^4'});
model.component('comp1').material('mat1').propertyGroup('def').func('Cp').set('arg', 'T');
model.component('comp1').material('mat1').propertyGroup('def').func('Cp').set('pieces', {'200.0' '1600.0' ...
    '1047.63657-0.372589265*T^1+9.45304214E-4*T^2-6.02409443E-7*T^3+1.2858961E-10*T^4'});
model.component('comp1').material('mat1').propertyGroup('def').func('rho').set('expr', 'pA*0.02897/8.314/T');
model.component('comp1').material('mat1').propertyGroup('def').func('rho').set('args', {'pA' 'T'});
model.component('comp1').material('mat1').propertyGroup('def').func('rho').set('dermethod', 'manual');
model.component('comp1').material('mat1').propertyGroup('def').func('rho').set('argders', {'pA' 'd(pA*0.02897/8.314/T,pA)'; 'T' ...
model.component('comp1').material('mat1').propertyGroup('def').addInput('temperature');
model.component('comp1').material('mat1').propertyGroup('def').addInput('pressure');
model.component('comp1').material('mat1').propertyGroup('RefractiveIndex').set('n', '');
model.component('comp1').material('mat1').propertyGroup('RefractiveIndex').set('ki', '');
model.component('comp1').material('mat1').propertyGroup('RefractiveIndex').set('n', {'1' '0' '0' '0' '1' '0' '0' '0' '1'});
model.component('comp1').material('mat1').propertyGroup('RefractiveIndex').set('ki', {'0' '0' '0' '0' '0' '0' '0' '0' '0'});
model.component('comp1').material('mat3').label('Silver [solid]');
model.component('comp1').material('mat3').set('family', 'custom');
model.component('comp1').material('mat3').set('specular', 'custom');
model.component('comp1').material('mat3').set('customspecular', [0.7843137254901961 1 1]);
model.component('comp1').material('mat3').set('diffuse', 'custom');
model.component('comp1').material('mat3').set('customdiffuse', [0.7843137254901961 0.7843137254901961 0.7843137254901961]);
model.component('comp1').material('mat3').set('ambient', 'custom');
model.component('comp1').material('mat3').set('customambient', [0.7843137254901961 0.7843137254901961 0.7843137254901961]);
model.component('comp1').material('mat3').set('noise', true);
model.component('comp1').material('mat3').set('noisefreq', 1);
model.component('comp1').material('mat3').set('lighting', 'cooktorrance');
model.component('comp1').material('mat3').set('fresnel', 0.9);
model.component('comp1').material('mat3').propertyGroup('def').func('k_solid_1').set('arg', 'T');
model.component('comp1').material('mat3').propertyGroup('def').func('k_solid_1').set('pieces', {'0.0' '12.0' ...
    '3587.152*T^1+348.8866*T^2-101.463*T^3+5.370411*T^4-0.06180556*T^5'; ...
  '12.0' '35.0' '50522.05-4753.283*T^1+172.2387*T^2-2.689213*T^3+0.01433909*T^4'; ...
  '35.0' '100.0' '10048.41-524.6271*T^1+11.6041*T^2-0.1287378*T^3+7.123183E-4*T^4-1.566974E-6*T^5'; ...
  '100.0' '250.0' '759.2297-7.120703*T^1+0.06197118*T^2-2.715928E-4*T^3+6.016276E-7*T^4-5.424229E-10*T^5'; ...
  '250.0' '1235.0' '419.8682+0.09979317*T^1-2.937158E-4*T^2+2.109166E-7*T^3-5.786644E-11*T^4'});
model.component('comp1').material('mat3').propertyGroup('def').func('res_solid_1').set('arg', 'T');
model.component('comp1').material('mat3').propertyGroup('def').func('res_solid_1').set('pieces', {'1.0' '15.8' ...
  '9.822048E-12+2.259567E-13*T^1-6.690094E-14*T^2+6.144183E-15*T^3'; ...
  '15.8' '27.5' '-2.33E-11+7.473333E-12*T^1-6.16E-13*T^2+2.026667E-14*T^3'; ...
  '27.5' '60.0' '8.015476E-10-7.586429E-11*T^1+2.210952E-12*T^2-1.2E-14*T^3'; ...
  '60.0' '200.0' '-2.428741E-9+6.974508E-11*T^1-4.447028E-14*T^2+6.841184E-17*T^3'; ...
  '200.0' '1235.0' '-1.812752E-9+6.074742E-11*T^1-3.077059E-15*T^2+8.269045E-18*T^3'});
model.component('comp1').material('mat3').propertyGroup('def').func('alpha').set('arg', 'T');
model.component('comp1').material('mat3').propertyGroup('def').func('alpha').set('pieces', {'0.0' '92.0' ...
    '1.397008E-5+6.293815E-8*T^1-3.772802E-10*T^2+1.128415E-12*T^3-1.223488E-15*T^4'; '92.0' '873.0' ...
    '1.604749E-5+1.576798E-8*T^1-1.719191E-11*T^2+6.931419E-15*T^3'});
model.component('comp1').material('mat3').propertyGroup('def').func('C_solid_1').set('arg', 'T');
model.component('comp1').material('mat3').propertyGroup('def').func('C_solid_1').set('pieces', {'1.0' '12.3' ...
    '0.01224227-0.01394369*T^1+0.01009593*T^2-4.957659E-4*T^3+1.66168E-4*T^4-3.512349E-6*T^5'; ...
  '12.3' '75.0' '24.24847-4.669813*T^1+0.2956444*T^2-0.004853545*T^3+3.380225E-5*T^4-8.439015E-8*T^5'; ...
  '75.0' '300.0' '-63.85884+5.177265*T^1-0.03961478*T^2+1.570454E-4*T^3-3.105375E-7*T^4+2.411435E-10*T^5'; ...
  '300.0' '1235.0' '225.7065+0.01705702*T^1+5.007143E-5*T^2-1.768498E-8*T^3'});
model.component('comp1').material('mat3').propertyGroup('def').func('sigma_solid_1').set('arg', 'T');
model.component('comp1').material('mat3').propertyGroup('def').func('sigma_solid_1').set('pieces', {'1.0' '15.8' ...
    '1/(6.144183E-15*T^3-6.690094E-14*T^2+2.259567E-13*T+9.822048E-12)'; ...
  '15.8' '27.5' '1/(2.026667E-14*T^3-6.160000E-13*T^2+7.473333E-12*T-2.330000E-11)'; ...
  '27.5' '60.0' '1/(-1.200000E-14*T^3+2.210952E-12*T^2-7.586429E-11*T+8.015476E-10)'; ...
  '60.0' '200.0' '1/(6.841184E-17*T^3-4.447028E-14*T^2+6.974508E-11*T-2.428741E-09)'; ...
  '200.0' '1235.0' '1/(8.269045E-18*T^3-3.077059E-15*T^2+6.074742E-11*T-1.812752E-09)'});
model.component('comp1').material('mat3').propertyGroup('def').func('HC_solid_1').set('arg', 'T');
model.component('comp1').material('mat3').propertyGroup('def').func('HC_solid_1').set('pieces', {'1.0' '12.3' ...
    '0.001320549-0.001504078*T^1+0.001089028*T^2-5.347725E-5*T^3+1.792421E-5*T^4-3.788701E-7*T^5'; ...
  '12.3' '75.0' '2.615634-0.5037235*T^1+0.03189057*T^2-5.235422E-4*T^3+3.646182E-6*T^4-9.102999E-9*T^5'; ...
  '75.0' '300.0' '-6.888324+0.5584615*T^1-0.004273165*T^2+1.694017E-5*T^3-3.349706E-8*T^4+2.601167E-11*T^5'; ...
  '300.0' '1235.0' '24.34652+0.001839907*T^1+5.401105E-6*T^2-1.907643E-9*T^3'});
model.component('comp1').material('mat3').propertyGroup('def').func('VP_solid_1').set('arg', 'T');
model.component('comp1').material('mat3').propertyGroup('def').func('VP_solid_1').set('pieces', {'293.0' '1235.0' ...
    '(exp((-1.499900e+04/T-7.845000e-01*log10(T)+1.200781e+01)*log(10.0)))*1.333200e+02'});
model.component('comp1').material('mat3').propertyGroup('def').func('rho').set('arg', 'T');
model.component('comp1').material('mat3').propertyGroup('def').func('rho').set('pieces', {'0.0' '30.0' '10630.38'; '30.0' '140.0' ...
    '10630.25+0.1394367*T^1-0.0048655*T^2+1.188653E-5*T^3'; '140.0' '873.0' '10658.96-0.4692536*T^1-2.976784E-4*T^2+1.470941E-7*T^3'});
```

```
    'k_solid_1(T[1/K])[W/(m*K)]' '0' '0' '0' 'k_solid_1(T[1/K])[W/(m*K)]'});
model.component('comp1').material('mat3').propertyGroup('def').set('resistivity', {'res_solid_1(T[1/K])[ohm*m]' '0' '0' '0' ...
    'res_solid_1(T[1/K])[ohm*m]' '0' '0' '0' 'res_solid_1(T[1/K])[ohm*m]'});
model.component('comp1').material('mat3').propertyGroup('def').set('thermalexpansioncoefficient', ...
    {'(alpha(T[1/K])[1/K]+(Tempref-293[K])*if(abs(T-Tempref)>1e-3,(alpha(T[1/K])[1/K]-alpha(Tempref[1/K])[1/K])/(T-Tempref),d(alpha(T[1/K])[1/K],T)))
model.component('comp1').material('mat3').propertyGroup('def').set('heatcapacity', 'C_solid_1(T[1/K])[J/(kg*K)]');
model.component('comp1').material('mat3').propertyGroup('def').set('electricconductivity', {'sigma_solid_1(T[1/K])[S/m]' '0' '0' '0' ...
    'sigma_solid_1(T[1/K])[S/m]' '0' '0' '0' 'sigma_solid_1(T[1/K])[S/m]'});
model.component('comp1').material('mat3').propertyGroup('def').set('HC', 'HC_solid_1(T[1/K])[J/(mol*K)]');
model.component('comp1').material('mat3').propertyGroup('def').set('VP', 'VP_solid_1(T[1/K])[Pa]');
model.component('comp1').material('mat3').propertyGroup('def').set('density', 'rho(T[1/K])[kg/m^3]');
model.component('comp1').material('mat3').propertyGroup('def').addInput('temperature');
model.component('comp1').material('mat3').propertyGroup('def').addInput('strainreferencetemperature');
model.component('comp1').material('mat3').propertyGroup('ThermalExpansion').func('dL').set('arg', 'T');
model.component('comp1').material('mat3').propertyGroup('ThermalExpansion').func('dL').set('pieces', {'0.0' '10.0' '-0.004134001';  ...
'10.0' '30.0' '-0.00415083+2.566303E-6*T^1-1.154799E-7*T^2+2.71357E-9*T^3';  ...
'30.0' '87.0' '-0.004065065-6.856586E-6*T^1+2.120401E-7*T^2-9.814541E-10*T^3+2.367498E-12*T^4-2.275841E-15*T^5';  ...
'87.0' '873.0' '-0.004745213+1.182195E-5*T^1+1.97186E-8*T^2-1.837579E-11*T^3+6.889976E-15*T^4'});
model.component('comp1').material('mat3').propertyGroup('ThermalExpansion').func('CTE').set('arg', 'T');
model.component('comp1').material('mat3').propertyGroup('ThermalExpansion').func('CTE').set('pieces', {'30.0' '300.0' ...
    '-8.786433E-6+5.111237E-7*T^1-4.275991E-9*T^2+1.8602E-11*T^3-4.019836E-14*T^4+3.403595E-17*T^5'});
model.component('comp1').material('mat3').propertyGroup('ThermalExpansion').set('alphatan', '');
model.component('comp1').material('mat3').propertyGroup('ThermalExpansion').set('dL', '');
model.component('comp1').material('mat3').propertyGroup('ThermalExpansion').set('alphatanIso', '');
model.component('comp1').material('mat3').propertyGroup('ThermalExpansion').set('dLIso', '');
model.component('comp1').material('mat3').propertyGroup('ThermalExpansion').set('alphatan', {'CTE(T[1/K])[1/K]' '0' '0' '0' 'CTE(T[1/K])[1/K]' '0' ...
    '0' '0' 'CTE(T[1/K])[1/K]'});
model.component('comp1').material('mat3').propertyGroup('ThermalExpansion').set('dL', {'(dL(T[1/K])-dL(Tempref[1/K]))/(1+dL(Tempref[1/K]))' ...
    '0' '0' '0' '(dL(T[1/K])-dL(Tempref[1/K]))/(1+dL(Tempref[1/K]))' '0' '0' '0' '(dL(T[1/K])-dL(Tempref[1/K]))/(1+dL(Tempref[1/K]))'});
model.component('comp1').material('mat3').propertyGroup('ThermalExpansion').set('alphatanIso', 'CTE(T)');
model.component('comp1').material('mat3').propertyGroup('ThermalExpansion').set('dLIso', '(dL(T)-dL(Tempref))/(1+dL(Tempref))');
model.component('comp1').material('mat3').propertyGroup('ThermalExpansion').addInput('temperature');
model.component('comp1').material('mat3').propertyGroup('ThermalExpansion').addInput('strainreferencetemperature');
model.component('comp1').material('mat3').propertyGroup('Enu').func('E').set('arg', 'T');
model.component('comp1').material('mat3').propertyGroup('Enu').func('E').set('pieces', {'0.0' '1173.0' ...
    '9.143965E10-2.775728E7*T^1-38123.31*T^2+49.99535*T^3-0.02009828*T^4'});
model.component('comp1').material('mat3').propertyGroup('Enu').func('nu').set('arg', 'T');
model.component('comp1').material('mat3').propertyGroup('Enu').func('nu').set('pieces', {'0.0' '1173.0' ...
    '0.360203+1.499369E-5*T^1+4.008534E-8*T^2-1.475466E-11*T^3-1.181682E-15*T^4'});
model.component('comp1').material('mat3').propertyGroup('Enu').set('youngsmodulus', 'E(T[1/K])[Pa]');
model.component('comp1').material('mat3').propertyGroup('Enu').set('poissonsratio', 'nu(T[1/K])');
model.component('comp1').material('mat3').propertyGroup('Enu').addInput('temperature');
model.component('comp1').material('mat3').propertyGroup('KG').func('mu').set('arg', 'T');
model.component('comp1').material('mat3').propertyGroup('KG').func('mu').set('pieces', {'0.0' '1173.0' ...
    '3.362582E10-6732560.0*T^1-31364.68*T^2+40.77008*T^3-0.01638472*T^4'});
model.component('comp1').material('mat3').propertyGroup('KG').func('kappa').set('arg', 'T');
model.component('comp1').material('mat3').propertyGroup('KG').func('kappa').set('pieces', {'0.0' '1173.0' ...
    '1.088261E11-3077357.0*T^1-87636.32*T^2+131.2545*T^3-0.05742809*T^4'});
model.component('comp1').material('mat3').propertyGroup('KG').set('K', '');
model.component('comp1').material('mat3').propertyGroup('KG').set('G', '');
model.component('comp1').material('mat3').propertyGroup('KG').set('K', 'kappa(T[1/K])[Pa]');
model.component('comp1').material('mat3').propertyGroup('KG').set('G', 'mu(T[1/K])[Pa]');
model.component('comp1').material('mat3').propertyGroup('KG').addInput('temperature');
model.component('comp1').material('mat3').propertyGroup('RefractiveIndex').set('n', '');
model.component('comp1').material('mat3').propertyGroup('RefractiveIndex').set('ki', '');
model.component('comp1').material('mat3').propertyGroup('RefractiveIndex').set('n', {'1.6090' '0' '0' '0' '1.6090' '0' '0' '0' '1.6090'});
model.component('comp1').material('mat3').propertyGroup('RefractiveIndex').set('ki', {'0.91260' '0' '0' '0' '0.91260' '0' '0' '0' '0.91260'});
model.component('comp1').material('mat4').label('Nickel (commercially pure) [solid]');
model.component('comp1').material('mat4').set('family', 'custom');
model.component('comp1').material('mat4').set('specular', 'custom');
model.component('comp1').material('mat4').set('customspecular', [0.7843137254901961 1 1]);
model.component('comp1').material('mat4').set('diffuse', 'custom');
model.component('comp1').material('mat4').set('customdiffuse', [0.7843137254901961 0.7843137254901961 0.7843137254901961]);
model.component('comp1').material('mat4').set('ambient', 'custom');
model.component('comp1').material('mat4').set('customambient', [0.7843137254901961 0.7843137254901961 0.7843137254901961]);
model.component('comp1').material('mat4').set('noise', true);
model.component('comp1').material('mat4').set('noisefreq', 1);
model.component('comp1').material('mat4').set('lighting', 'cooktorrance');
model.component('comp1').material('mat4').set('fresnel', 0.9);
model.component('comp1').material('mat4').set('info', {'Composition' '' 'commerically pure'});
model.component('comp1').material('mat4').propertyGroup('def').func('k').set('arg', 'T');
model.component('comp1').material('mat4').propertyGroup('def').func('k').set('pieces', {'273.0' '673.0' ...
    '82.43111-0.05351518*T^1-6.532678E-5*T^2+9.112509E-8*T^3'; '673.0' '1173.0' ...
    '237.7946-0.6995247*T^1+8.086325E-4*T^2-2.907407E-7*T^3'});
model.component('comp1').material('mat4').propertyGroup('def').func('alpha').set('arg', 'T');
model.component('comp1').material('mat4').propertyGroup('def').func('alpha').set('pieces', {'0.0' '20.0' '8.085811E-6+2.790925E-8*T^1'; '20.0' ...
    '300.0' '8.041785E-6+3.13522E-8*T^1-5.800962E-11*T^2-1.094272E-13*T^3+3.207968E-16*T^4'});
model.component('comp1').material('mat4').propertyGroup('def').func('rho').set('arg', 'T');
model.component('comp1').material('mat4').propertyGroup('def').func('rho').set('pieces', {'0.0' '20.0' ...
    '8953.509-0.008017787*T^1-4.251331E-4*T^2'; '20.0' ...
    '300.0' '8952.813+0.04935669*T^1-0.001622488*T^2+3.023595E-6*T^3-1.90504E-9*T^4'});
model.component('comp1').material('mat4').propertyGroup('def').set('thermalconductivity', {'k(T[1/K])[W/(m*K)]' '0' '0' '0' 'k(T[1/K])[W/(m*K)]' '0' ...
    '0' '0' 'k(T[1/K])[W/(m*K)]'});
model.component('comp1').material('mat4').propertyGroup('def').set('thermalexpansioncoefficient', ...
    {'(alpha(T[1/K])[1/K]+(Tempref-293[K])*if(abs(T-Tempref)>1e-3,(alpha(T[1/K])[1/K]-alpha(Tempref[1/K])[1/K])/(T-Tempref),d(alpha(T[1/K])[1/K],T)))
```

```
model.component('comp1').material('mat4').propertyGroup('def').set('density', 'rho(T[1/K])[kg/m^3]');
model.component('comp1').material('mat4').propertyGroup('def').addInput('temperature');
model.component('comp1').material('mat4').propertyGroup('def').addInput('strainreferencetemperature');
model.component('comp1').material('mat4').propertyGroup('ThermalExpansion').func('dL').set('arg', 'T');
model.component('comp1').material('mat4').propertyGroup('ThermalExpansion').func('dL').set('pieces', {'0.0' '20.0' ...
    '-0.00237+9.227418E-9*T^1+3.167763E-8*T^2'; '20.0' '300.0' ...
    '-0.002344272-1.825737E-6*T^1+6.010195E-8*T^2-1.109772E-10*T^3+6.933659E-14*T^4'});
model.component('comp1').material('mat4').propertyGroup('ThermalExpansion').set('alphatan', '');
model.component('comp1').material('mat4').propertyGroup('ThermalExpansion').set('dL', '');
model.component('comp1').material('mat4').propertyGroup('ThermalExpansion').set('alphatanIso', '');
model.component('comp1').material('mat4').propertyGroup('ThermalExpansion').set('dLIso', '');
model.component('comp1').material('mat4').propertyGroup('ThermalExpansion').set('alphatan', {'1' '0' '0' '0' '1' '0' '0' '0' '1'});
model.component('comp1').material('mat4').propertyGroup('ThermalExpansion').set('dL', {'(dL(T[1/K])-dL(Tempref[1/K]))/(1+dL(Tempref[1/K]))' ...
    '0' '0' '0' '(dL(T[1/K])-dL(Tempref[1/K]))/(1+dL(Tempref[1/K]))' '0' '0' '0' '(dL(T[1/K])-dL(Tempref[1/K]))/(1+dL(Tempref[1/K]))'});
model.component('comp1').material('mat4').propertyGroup('ThermalExpansion').set('alphatanIso', '1');
model.component('comp1').material('mat4').propertyGroup('ThermalExpansion').set('dLIso', '(dL(T)-dL(Tempref))/(1+dL(Tempref))');
model.component('comp1').material('mat4').propertyGroup('ThermalExpansion').addInput('temperature');
model.component('comp1').material('mat4').propertyGroup('ThermalExpansion').addInput('strainreferencetemperature');
model.component('comp1').material('mat4').propertyGroup('Enu').func('E').set('arg', 'T');
model.component('comp1').material('mat4').propertyGroup('Enu').func('E').set('pieces', {'20.0' '80.0' ...
    '2.128205E11+7.300654E7*T^1-2910689.0*T^2+44672.81*T^3-262.2094*T^4'; '80.0' '311.0' ....
    '2.03931E11+3.899186E8*T^1-5310091.0*T^2+25603.5*T^3-55.59948*T^4+0.04649125*T^5'});
model.component('comp1').material('mat4').propertyGroup('Enu').set('youngsmodulus', 'E(T[1/K])[Pa]');
model.component('comp1').material('mat4').propertyGroup('Enu').set('poissonsratio', '1');
model.sol('sol1').create('st1', 'StudyStep');
model.sol('sol1').create('v1', 'Variables');
model.sol('sol1').create('s1', 'Stationary');
model.sol('sol1').feature('s1').create('p1', 'Parametric');
model.sol('sol1').feature('s1').create('fc1', 'FullyCoupled');
model.sol('sol1').feature('s1').create('i1', 'Iterative');
model.sol('sol1').feature('s1').feature('i1').create('mg1', 'Multigrid');
model.sol('sol1').feature('s1').feature('i1').feature('mg1').feature('pr').create('sv1', 'SORVector');
model.sol('sol1').feature('s1').feature('i1').feature('mg1').feature('po').create('sv1', 'SORVector');
model.sol('sol1').feature('s1').feature.remove('fcDef');
model.result.numerical.create('int1', 'IntSurface');
model.result.numerical('int1').selection.set([13 29]);
model.result.numerical('int1').set('probetag', 'none');
model.result.create('pg1', 'PlotGroup3D');
model.result.create('pg2', 'PlotGroup3D');
model.result.create('pg3', 'PlotGroup1D');
model.result('pg1').create('mslc1', 'Multislice');
model.result('pg2').create('slc1', 'Slice');
model.result('pg2').feature('slc1').create('def1', 'Deform');
model.result('pg3').create('tblp1', 'Table');
model.study('std1').feature('freq').set('punit', 'Hz');
model.study('std1').feature('freq').set('plist', '1.0e12');
model.sol('sol1').attach('std1');
model.sol('sol1').feature('v1').set('clistctrl', {'p1'});
model.sol('sol1').feature('v1').set('cname', {'freq'});
model.sol('sol1').feature('v1').set('clist', {'1.0e12[Hz]'});
model.sol('sol1').feature('s1').feature('p1').set('pname', {'freq'});
model.sol('sol1').feature('s1').feature('p1').set('plistarr', {'1.0e12'});
model.sol('sol1').feature('s1').feature('p1').set('punit', {'Hz'});
model.sol('sol1').feature('s1').feature('p1').set('pcontinuationmode', 'no');
model.sol('sol1').feature('s1').feature('p1').set('preusesol', 'auto');
model.sol('sol1').feature('s1').feature('i1').label('Suggested Iterative Solver (ewfd)');
model.sol('sol1').feature('s1').feature('i1').set('linsolver', 'bicgstab');
model.sol('sol1').feature('s1').feature('i1').feature('mg1').feature('pr').feature('sv1').set('sorvecdof', {'comp1_E'});
model.sol('sol1').feature('s1').feature('i1').feature('mg1').feature('po').feature('sv1').set('sorvecdof', {'comp1_E'});
model.sol('sol1').runAll;
model.result.numerical('int1').set('table', 'tbl1');
model.result.numerical('int1').set('expr', {'ewfd.normE'});
model.result.numerical('int1').set('unit', {'V*m'});
model.result.numerical('int1').set('descr', {'Electric field norm'});
model.result.numerical('int1').setResult;
model.result('pg1').label('Electric Field (ewfd)');
model.result('pg1').set('frametype', 'spatial');
model.result('pg1').feature('mslc1').set('multiplanezmethod', 'coord');
model.result('pg1').feature('mslc1').set('zcoord', '1050.0');
model.result('pg1').feature('mslc1').set('rangecoloractive', true);
model.result('pg1').feature('mslc1').set('rangecolormin', 41923.82486663746);
model.result('pg1').feature('mslc1').set('rangecolormax', 400000);
model.result('pg1').feature('mslc1').set('smooth', 'internal');
model.result('pg1').feature('mslc1').set('resolution', 'normal');
model.result('pg2').feature('slc1').set('quickplane', 'xy');
model.result('pg2').feature('slc1').set('quickzmethod', 'coord');
model.result('pg2').feature('slc1').set('quickz', '1033.0');
model.result('pg2').feature('slc1').set('rangecoloractive', true);
model.result('pg2').feature('slc1').set('rangecolormin', 9292.73707567522);
model.result('pg2').feature('slc1').set('rangecolormax', 250000);
model.result('pg2').feature('slc1').set('smooth', 'internal');
model.result('pg2').feature('slc1').set('resolution', 'normal');
model.result('pg2').feature('slc1').feature('def1').active(false);
model.result('pg2').feature('slc1').feature('def1').set('expr', {'+ewfd.Ex' '+ewfd.Ey' '+ewfd.Ez'});
model.result('pg2').feature('slc1').feature('def1').set('descr', ' + Electric field');
model.result('pg2').feature('slc1').feature('def1').set('scale', 2.085880595474555E-4);
```

```
model.result('pg2').feature('slc1').feature('def1').set('scaleactive', false);
model.result('pg3').set('data', 'none');
model.result('pg3').set('xlabel', 'freq (Hz)');
model.result('pg3').set('ylabel', 'Electric field norm (V*m)');
model.result('pg3').set('xlabelactive', false);
model.result('pg3').set('ylabelactive', false);
model.result('pg3').feature('tblp1').set('linewidth', 3);
model.result('pg3').feature('tblp1').set('linemarker', 'cycle');
model.result('pg3').feature('tblp1').set('markerpos', 'datapoints');
end
```

```
function [] = fNiTa(model)
model.component('comp1').material.create('mat1', 'Common');
model.component('comp1').material.create('mat4', 'Common');
model.component('comp1').material.create('mat5', 'Common');
model.component('comp1').material('mat1').propertyGroup('def').func.create('eta', 'Piecewise');
model.component('comp1').material('mat1').propertyGroup('def').func.create('Cp', 'Piecewise');
model.component('comp1').material('mat1').propertyGroup('def').func.create('rho', 'Analytic');
model.component('comp1').material('mat1').propertyGroup('def').func.create('k', 'Piecewise');
model.component('comp1').material('mat1').propertyGroup('def').func.create('cs', 'Analytic');
model.component('comp1').material('mat1').propertyGroup.create('RefractiveIndex', 'Refractive index');
model.component('comp1').material('mat4').selection.set([4]);
model.component('comp1').material('mat4').info.create('Composition');
model.component('comp1').material('mat4').propertyGroup('def').func.create('k', 'Piecewise');
model.component('comp1').material('mat4').propertyGroup('def').func.create('alpha', 'Piecewise');
model.component('comp1').material('mat4').propertyGroup('def').func.create('rho', 'Piecewise');
model.component('comp1').material('mat4').propertyGroup.create('ThermalExpansion', 'Thermal expansion');
model.component('comp1').material('mat4').propertyGroup('ThermalExpansion').func.create('dL', 'Piecewise');
model.component('comp1').material('mat4').propertyGroup.create('Enu', 'Young''s modulus and Poisson''s ratio');
model.component('comp1').material('mat4').propertyGroup('Enu').func.create('E', 'Piecewise');
model.component('comp1').material('mat4').propertyGroup.create('RefractiveIndex', 'Refractive index');
model.component('comp1').material('mat5').selection.set([3]);
model.component('comp1').material('mat5').propertyGroup.create('RefractiveIndex', 'Refractive index');
model.component('comp1').material('mat5').propertyGroup('RefractiveIndex').func.create('int1', 'Interpolation');
model.component('comp1').material('mat5').propertyGroup('RefractiveIndex').func.create('int2', 'Interpolation');
model.component('comp1').physics.create('ewfd', 'ElectromagneticWavesFrequencyDomain', 'geom1');
model.component('comp1').physics('ewfd').create('sctr1', 'Scattering', 2);
model.component('comp1').physics('ewfd').feature('sctr1').selection.set([1 2 3 4 5 7 8 9 43 44]);
model.component('comp1').physics('ewfd').create('port1', 'Port', 2);
model.component('comp1').physics('ewfd').feature('port1').selection.set([7]);
model.component('comp1').physics('ewfd').create('port2', 'Port', 2);
model.component('comp1').physics('ewfd').feature('port2').selection.set([3]);
model.component('comp1').mesh('mesh1').create('ftet1', 'FreeTet');
model.result.table('tbl1').comments('Surface Integration 1 (ewfd.normE)');
model.result.table('evl3').label('Evaluation 3D');
model.result.table('evl3').comments('Interactive 3D values');
model.component('comp1').view('view1').set('renderwireframe', true);
model.component('comp1').view('view1').set('transparency', true);
model.component('comp1').view('view2').axis.set('xmin', -37.6529541015625);
model.component('comp1').view('view2').axis.set('xmax', 2112.7470703125);
model.component('comp1').view('view2').axis.set('ymin', 0);
model.component('comp1').view('view2').axis.set('ymax', 1651.199951171875);
model.component('comp1').material('mat1').label('Air');
model.component('comp1').material('mat1').set('family', 'air');
model.component('comp1').material('mat1').propertyGroup('def').func('eta').set('arg', 'T');
model.component('comp1').material('mat1').propertyGroup('def').func('eta').set('pieces', {'200.0' '1600.0' ...
    '-8.38278E-7+8.35717342E-8*T^1-7.69429583E-11*T^2+4.6437266E-14*T^3-1.06585607E-17*T^4'});
model.component('comp1').material('mat1').propertyGroup('def').func('Cp').set('arg', 'T');
model.component('comp1').material('mat1').propertyGroup('def').func('Cp').set('pieces', {'200.0' '1600.0' ...
    '1047.63657-0.372589265*T^1+9.45304214E-4*T^2-6.02409443E-7*T^3+1.2858961E-10*T^4'});
model.component('comp1').material('mat1').propertyGroup('def').func('rho').set('expr', 'pA*0.02897/8.314/T');
model.component('comp1').material('mat1').propertyGroup('def').func('rho').set('args', {'pA' 'T'});
model.component('comp1').material('mat1').propertyGroup('def').func('rho').set('dermethod', 'manual');
model.component('comp1').material('mat1').propertyGroup('def').func('rho').set('argders', {'pA' 'd(pA*0.02897/8.314/T,pA)'; 'T' ...
    'd(pA*0.02897/8.314/T,T)'});
model.component('comp1').material('mat1').propertyGroup('def').func('rho').set('plotargs', {'pA' '0' '1'; 'T' '0' '1'});
model.component('comp1').material('mat1').propertyGroup('def').func('k').set('arg', 'T');
model.component('comp1').material('mat1').propertyGroup('def').func('k').set('pieces', {'200.0' '1600.0' ...
    '-0.00227583562+1.15480022E-4*T^1-7.90252856E-8*T^2+4.11702505E-11*T^3-7.43864331E-15*T^4'});
model.component('comp1').material('mat1').propertyGroup('def').func('cs').set('expr', 'sqrt(1.4*287*T)');
model.component('comp1').material('mat1').propertyGroup('def').func('cs').set('args', {'T'});
model.component('comp1').material('mat1').propertyGroup('def').func('cs').set('dermethod', 'manual');
model.component('comp1').material('mat1').propertyGroup('def').func('cs').set('argders', {'T' 'd(sqrt(1.4*287*T),T)'});
model.component('comp1').material('mat1').propertyGroup('def').func('cs').set('plotargs', {'T' '0' '1'});
model.component('comp1').material('mat1').propertyGroup('def').set('relpermeability', {'1' '0' '0' '0' '1' '0' '0' '0' '1'});
model.component('comp1').material('mat1').propertyGroup('def').set('relpermittivity', {'1' '0' '0' '0' '1' '0' '0' '0' '1'});
model.component('comp1').material('mat1').propertyGroup('def').set('dynamicviscosity', 'eta(T[1/K])[Pa*s]');
model.component('comp1').material('mat1').propertyGroup('def').set('ratioofspecificheat', '1.4');
```

```
model.component('comp1').material('mat1').propertyGroup('def').set('electricconductivity', {'0[S/m]' '0' '0' '0' '0[S/m]' '0' '0' '0' '0[S/m]'});
model.component('comp1').material('mat1').propertyGroup('def').set('heatcapacity', 'Cp(T[1/K])[J/(kg*K)]');
model.component('comp1').material('mat1').propertyGroup('def').set('density', 'rho(pA[1/Pa],T[1/K])[kg/m^3]');
model.component('comp1').material('mat1').propertyGroup('def').set('thermalconductivity', {'k(T[1/K])[W/(m*K)]' '0' '0' '0' 'k(T[1/K])[W/(m*K)]' '0' '0' ...
    '0' 'k(T[1/K])[W/(m*K)]'});
model.component('comp1').material('mat1').propertyGroup('def').set('soundspeed', 'cs(T[1/K])[m/s]');
model.component('comp1').material('mat1').propertyGroup('def').addInput('temperature');
model.component('comp1').material('mat1').propertyGroup('def').addInput('pressure');
model.component('comp1').material('mat1').propertyGroup('RefractiveIndex').set('n', '');
model.component('comp1').material('mat1').propertyGroup('RefractiveIndex').set('ki', '');
model.component('comp1').material('mat1').propertyGroup('RefractiveIndex').set('n', {'1' '0' '0' '0' '1' '0' '0' '0' '1'});
model.component('comp1').material('mat1').propertyGroup('RefractiveIndex').set('ki', {'0' '0' '0' '0' '0' '0' '0' '0' '0'});
model.component('comp1').material('mat4').label('Nickel (commerically pure) [solid]');
model.component('comp1').material('mat4').set('family', 'custom');
model.component('comp1').material('mat4').set('specular', 'custom');
model.component('comp1').material('mat4').set('customspecular', [0.7843137254901961 1 1]);
model.component('comp1').material('mat4').set('diffuse', 'custom');
model.component('comp1').material('mat4').set('customdiffuse', [0.7843137254901961 0.7843137254901961 0.7843137254901961]);
model.component('comp1').material('mat4').set('ambient', 'custom');
model.component('comp1').material('mat4').set('customambient', [0.7843137254901961 0.7843137254901961 0.7843137254901961]);
model.component('comp1').material('mat4').set('noise', true);
model.component('comp1').material('mat4').set('noisefreq', 1);
```

```
model.component('comp1').material('mat4').set('lighting', 'cooktorrance');
model.component('comp1').material('mat4').set('fresnel', 0.9);
model.component('comp1').material('mat4').set('info', {'Composition' '' 'commerically pure'});
model.component('comp1').material('mat4').propertyGroup('def').func('k').set('arg', 'T');
model.component('comp1').material('mat4').propertyGroup('def').func('k').set('pieces', {'273.0' '673.0' ...
    '82.43111-0.05351518*T^1-6.532678E-5*T^2+9.112509E-8*T^3'; '673.0' ...
    '1173.0' '237.7946-0.6995247*T^1+8.086325E-4*T^2-2.907407E-7*T^3'});
model.component('comp1').material('mat4').propertyGroup('def').func('alpha').set('arg', 'T');
model.component('comp1').material('mat4').propertyGroup('def').func('alpha').set('pieces', {'0.0' '20.0' '8.085811E-6+2.790925E-8*T^1'; '20.0' ...
    '300.0' '8.041785E-6+3.13522E-8*T^1-5.800962E-11*T^2-1.094272E-13*T^3+3.207968E-16*T^4'});
model.component('comp1').material('mat4').propertyGroup('def').func('rho').set('arg', 'T');
model.component('comp1').material('mat4').propertyGroup('def').func('rho').set('pieces', {'0.0' '20.0' ...
    '8953.509-0.008017787*T^1-4.251331E-4*T^2'; '20.0' '300.0' ...
    '8952.813+0.04935669*T^1-0.001622488*T^2+3.023595E-6*T^3-1.90504E-9*T^4'});
model.component('comp1').material('mat4').propertyGroup('def').set('thermalconductivity', ....
    {'k(T[1/K])[W/(m*K)]' '0' '0' '0' 'k(T[1/K])[W/(m*K)]' '0' '0' '0' 'k(T[1/K])[W/(m*K)]'});
model.component('comp1').material('mat4').propertyGroup('def').set('thermalexpansioncoefficient', ...
    {'(alpha(T[1/K])[1/K]+(Tempref-293[K])*if(abs(T-Tempref)>1e-3,(alpha(T[1/K])[1/K]-alpha(Tempref[1/K])[1/K])/(T-Tempref),d(alpha(T[1/K])[1/K],T)))/
model.component('comp1').material('mat4').propertyGroup('def').set('density', 'rho(T[1/K])[kg/m^3]');
model.component('comp1').material('mat4').propertyGroup('def').addInput('temperature');
model.component('comp1').material('mat4').propertyGroup('def').addInput('strainreferencetemperature');
model.component('comp1').material('mat4').propertyGroup('ThermalExpansion').func('dL').set('arg', 'T');
model.component('comp1').material('mat4').propertyGroup('ThermalExpansion').func('dL').set('pieces', {'0.0' '20.0' ...
    '-0.00237+9.227418E-9*T^1+3.167763E-8*T^2'; '20.0' ...
    '300.0' '-0.002344272-1.825737E-6*T^1+6.010195E-8*T^2-1.109772E-10*T^3+6.933659E-14*T^4'});
model.component('comp1').material('mat4').propertyGroup('ThermalExpansion').set('alphatan', '');
model.component('comp1').material('mat4').propertyGroup('ThermalExpansion').set('dL', '');
model.component('comp1').material('mat4').propertyGroup('ThermalExpansion').set('alphatanIso', '');
model.component('comp1').material('mat4').propertyGroup('ThermalExpansion').set('dLIso', '');
model.component('comp1').material('mat4').propertyGroup('ThermalExpansion').set('alphatan', {'1' '0' '0' '0' '1' '0' '0' '0' '1'});
model.component('comp1').material('mat4').propertyGroup('ThermalExpansion').set('dL', {'(dL(T[1/K])-dL(Tempref[1/K]))/(1+dL(Tempref[1/K]))' ...
    '0' '0' '0' '(dL(T[1/K])-dL(Tempref[1/K]))/(1+dL(Tempref[1/K]))' '0' '0' '0' '(dL(T[1/K])-dL(Tempref[1/K]))/(1+dL(Tempref[1/K]))'});
model.component('comp1').material('mat4').propertyGroup('ThermalExpansion').set('alphatanIso', '1');
model.component('comp1').material('mat4').propertyGroup('ThermalExpansion').set('dLIso', '(dL(T)-dL(Tempref))/(1+dL(Tempref))');
model.component('comp1').material('mat4').propertyGroup('ThermalExpansion').addInput('temperature');
model.component('comp1').material('mat4').propertyGroup('ThermalExpansion').addInput('strainreferencetemperature');
model.component('comp1').material('mat4').propertyGroup('Enu').func('E').set('arg', 'T');
model.component('comp1').material('mat4').propertyGroup('Enu').func('E').set('pieces', {'20.0' '80.0' ...
    '2.128205E11+7.300654E7*T^1-2910689.0*T^2+44672.81*T^3-262.2094*T^4'; '80.0' '311.0' ...
    '2.03931E11+3.899186E8*T^1-5310091.0*T^2+25603.5*T^3-55.59948*T^4+0.04649125*T^5'});
model.component('comp1').material('mat4').propertyGroup('Enu').set('youngsmodulus', 'E(T[1/K])[Pa]');
model.component('comp1').material('mat4').propertyGroup('Enu').set('poissonsratio', '1');
model.component('comp1').material('mat4').propertyGroup('Enu').addInput('temperature');
model.component('comp1').material('mat4').propertyGroup('RefractiveIndex').set('n', '');
```

```
model.component('comp1').material('mat4').propertyGroup('RefractiveIndex').set('ki', '');
model.component('comp1').material('mat4').propertyGroup('RefractiveIndex').set('n', {'2.0211' '0' '0' '0' '2.0211' '0' '0' '0' '2.0211'});
model.component('comp1').material('mat4').propertyGroup('RefractiveIndex').set('ki', {'2.1822' '0' '0' '0' '2.1822' '0' '0' '0' '2.1822'});
model.component('comp1').material('mat5').label(['Ta (Tantalum) (Ordal et al. 1988: n,k 0.667-125 ' native2unicode(hex2dec({'00' 'b5'}), 'unicode') ...
    'm)']);
model.component('comp1').material('mat5').set('groups', {'inorganic' 'Inorganic Materials'; 'ta_tantalum_and_tantalates' ...
    'Ta - Tantalum and Tantalates'; 'experimental_data' 'Experimental data'});
model.component('comp1').material('mat5').propertyGroup('RefractiveIndex').func('int1').label('Refractive index');
model.component('comp1').material('mat5').propertyGroup('RefractiveIndex').func('int1').set('funcname', 'n_interp');
model.component('comp1').material('mat5').propertyGroup('RefractiveIndex').func('int1').set('table', {'6.67E-7' '1.4478379';'7.14E-7' '1.2685432';  ...
'7.69E-7' '1.1466787';'8.329999999999E-7' '1.073087';'9.09E-7' '1.0150307';'1.0E-6' '0.9821499';'1.05E-6' '0.96032';'1.11E-6' '0.9253053';  ...
'1.18E-6' '0.9012078';'1.2499999999999999E-6' '0.8740663';'1.33E-6' '0.8414308';'1.4299999999999999E-6' '0.8354757';  ...
'1.5399999999999999E-6' '0.8564461';'1.6699999999999999E-6' '0.9019896';'1.82E-6' '0.960826';'2.0E-6' '1.0326738';  ...
'2.1099999999999997E-6' '1.1067824';'2.22E-6' '1.192353';'2.35E-6' '1.2678805';'2.4999999999999998E-6' '1.3499202';'2.67E-6' '1.4802727';  ...
'2.8599999999999997E-6' '1.6348171';'3.0799999999999997E-6' '1.8209204';'3.33E-6' '2.0587589';'3.64E-6' '2.3469537';  ...
'4.0E-6' '2.6799222';'4.44E-6' '3.1932058';'4.999999999999996E-6' '3.9020861';'5.7099999999999995E-6' '4.9008382';'6.67E-6' '6.4122619';  ...
'8.0E-6' '8.7691085';'9.999999999999999E-6' '12.931507';'1.1099999999999999E-5' '15.61576';'1.2499999999999999E-5' '19.058763';  ...
'1.43E-5' '24.23852';'1.67E-5' '31.493393';'1.9999999999999998E-5' '41.721911';'2.2199999999999998E-5' '48.392026';  ...
'2.4999999999999998E-5' '56.42849';'2.86E-5' '66.1388';'3.3299999999999996E-5' '77.867703';'3.9999999999999996E-5' '92.016211';  ...
'4.4399999999999995E-5' '100.17673';'4.999999999999996E-5' '109.23359';'5.71E-5' '119.43744';'6.67E-5' '131.25365';  ...

'7.999999999999999E-5' '145.5884';'9.999999999999999E-5' '164.07966';'1.25E-4' '184.49663'});
model.component('comp1').material('mat5').propertyGroup('RefractiveIndex').func('int1').set('interp', 'piecewisecubic');
model.component('comp1').material('mat5').propertyGroup('RefractiveIndex').func('int1').set('argunit', 'm');
model.component('comp1').material('mat5').propertyGroup('RefractiveIndex').func('int1').set('fununit', '1');
model.component('comp1').material('mat5').propertyGroup('RefractiveIndex').func('int2').label('Refractive index, imaginary part');
model.component('comp1').material('mat5').propertyGroup('RefractiveIndex').func('int2').set('funcname', 'k_interp');
model.component('comp1').material('mat5').propertyGroup('RefractiveIndex').func('int2').set('table', {'6.67E-7' '2.3306224';'7.14E-7' '2.7919609';  ...
'7.69E-7' '3.2384897';'8.329999999999E-7' '3.7362305';'9.09E-7' '4.2736932';'1.0E-6' '4.8873028';'1.05E-6' '5.2167868';'1.11E-6' '5.5894492';  ...
'1.18E-6' '6.0121558';'1.2499999999999999E-6' '6.4776281';'1.33E-6' '7.0171487';'1.4299999999999999E-6' '7.6548582';  ...
'1.5399999999999999E-6' '8.3686838';'1.6699999999999999E-6' '9.1771344';'1.82E-6' '10.114648';'2.0E-6' '11.243595';  ...
'2.1099999999999997E-6' '11.896174';'2.22E-6' '12.595572';'2.35E-6' '13.363144';'2.4999999999999998E-6' '14.23958';  ...
'2.67E-6' '15.240124';'2.8599999999999997E-6' '16.359682';'3.0799999999999997E-6' '17.640057';'3.33E-6' '19.120563';  ...
'3.64E-6' '20.845754';'4.0E-6' '22.915456';'4.44E-6' '25.461819';'4.999999999999996E-6' '28.57704';'5.7099999999999995E-6' '32.510917';  ...
'6.67E-6' '37.643719';'8.0E-6' '44.613332';'9.999999999999999E-6' '54.673721';'1.1099999999999999E-5' '59.926459';  ...
'1.2499999999999999E-5' '66.350003';'1.43E-5' '73.949845';'1.67E-5' '82.79423';'1.9999999999999998E-5' '93.122866';  ...
'2.2199999999999998E-5' '98.921392';'2.4999999999999998E-5' '105.16669';'2.86E-5' '111.86857';'3.3299999999999996E-5' '119.08288';  ...
'3.9999999999999996E-5' '127.08528';'4.4399999999999995E-5' '131.63028';'4.999999999999996E-5' '136.82885';'5.71E-5' '143.08774';  ...
'6.67E-5' '151.08524';'7.999999999999999E-5' '161.9815';'9.999999999999999E-5' '177.86128';'1.25E-4' '196.87085'});
model.component('comp1').material('mat5').propertyGroup('RefractiveIndex').func('int2').set('interp', 'piecewisecubic');
model.component('comp1').material('mat5').propertyGroup('RefractiveIndex').func('int2').set('argunit', 'm');
model.component('comp1').material('mat5').propertyGroup('RefractiveIndex').func('int2').set('fununit', '1');
model.component('comp1').material('mat5').propertyGroup('RefractiveIndex').set('n', '');
model.component('comp1').material('mat5').propertyGroup('RefractiveIndex').set('ki', '');
model.component('comp1').material('mat5').propertyGroup('RefractiveIndex').set('n', '');
model.component('comp1').material('mat5').propertyGroup('RefractiveIndex').set('ki', '');
model.component('comp1').material('mat5').propertyGroup('RefractiveIndex').set('n', {'n_interp(c_const/freq)' '0' '0' '0' 'n_interp(c_const/freq)' '0' ...
    '0' '0' 'n_interp(c_const/freq)'});
model.component('comp1').material('mat5').propertyGroup('RefractiveIndex').set('ki', {'k_interp(c_const/freq)' '0' '0' '0' 'k_interp(c_const/freq)' '0' ...
    '0' '0' 'k_interp(c_const/freq)'});
model.component('comp1').material('mat5').propertyGroup('RefractiveIndex').addInput('frequency');
model.component('comp1').physics('ewfd').prop('EquationForm').set('freq', '1[GHz]');
model.component('comp1').physics('ewfd').prop('EquationForm').set('modeFreq', '1[GHz]');
model.component('comp1').physics('ewfd').prop('MeshControl').set('EnableMeshControl', false);
model.component('comp1').physics('ewfd').feature('port1').set('Pin', '1[mW]');
model.component('comp1').physics('ewfd').feature('port1').set('beta', 'abs(ewfd.k0)');
model.component('comp1').physics('ewfd').feature('port1').set('E0', {'0'; 'exp(-j*ewfd.k0*z)'; '0'});
model.component('comp1').physics('ewfd').feature('port2').set('beta', 'abs(ewfd.k0)');
model.component('comp1').physics('ewfd').feature('port2').set('E0', {'0'; 'exp(-j*ewfd.k0*z)'; '0'});
model.component('comp1').mesh('mesh1').feature('size').set('hauto', 4);
model.component('comp1').mesh('mesh1').run;
model.study.create('std1');
model.study('std1').create('freq', 'Frequency');
model.sol.create('sol1');
model.sol('sol1').study('std1');
```

```
model.sol('sol1').attach('std1');
model.sol('sol1').create('st1', 'StudyStep');
model.sol('sol1').create('v1', 'Variables');
model.sol('sol1').create('s1', 'Stationary');
model.sol('sol1').feature('s1').create('p1', 'Parametric');
model.sol('sol1').feature('s1').create('fc1', 'FullyCoupled');
model.sol('sol1').feature('s1').create('i1', 'Iterative');
model.sol('sol1').feature('s1').feature('i1').create('mg1', 'Multigrid');
model.sol('sol1').feature('s1').feature('i1').feature('mg1').feature('pr').create('sv1', 'SORVector');
model.sol('sol1').feature('s1').feature('i1').feature('mg1').feature('po').create('sv1', 'SORVector');
model.sol('sol1').feature('s1').feature.remove('fcDef');
model.result.numerical.create('int1', 'IntSurface');
model.result.numerical('int1').selection.set([13 29]);
model.result.numerical('int1').set('probetag', 'none');
model.result.create('pg1', 'PlotGroup3D');
model.result.create('pg2', 'PlotGroup3D');
model.result.create('pg3', 'PlotGroup1D');
model.result('pg1').create('mslc1', 'Multislice');
model.result('pg2').create('slc1', 'Slice');
model.result('pg2').feature('slc1').create('def1', 'Deform');
model.result('pg3').create('tblp1', 'Table');
model.study('std1').feature('freq').set('punit', 'Hz');
model.study('std1').feature('freq').set('plist', '1.0e12');
model.sol('sol1').attach('std1');
model.sol('sol1').feature('v1').set('clistctrl', {'p1'});
model.sol('sol1').feature('v1').set('cname', {'freq'});
model.sol('sol1').feature('v1').set('clist', {'1.0e12[Hz]'});
model.sol('sol1').feature('s1').feature('p1').set('pname', {'freq'});
model.sol('sol1').feature('s1').feature('p1').set('plistarr', {'1.0e12'});
model.sol('sol1').feature('s1').feature('p1').set('punit', {'Hz'});
model.sol('sol1').feature('s1').feature('p1').set('pcontinuationmode', 'no');
model.sol('sol1').feature('s1').feature('p1').set('preusesol', 'auto');
model.sol('sol1').feature('s1').feature('i1').label('Suggested Iterative Solver (ewfd)');
model.sol('sol1').feature('s1').feature('i1').set('linsolver', 'bicgstab');
model.sol('sol1').feature('s1').feature('i1').feature('mg1').feature('pr').feature('sv1').set('sorvecdof', {'comp1_E'});
model.sol('sol1').feature('s1').feature('i1').feature('mg1').feature('po').feature('sv1').set('sorvecdof', {'comp1_E'});
model.sol('sol1').runAll;
model.result.numerical('int1').set('table', 'tbl1');
model.result.numerical('int1').set('expr', {'ewfd.normE'});
model.result.numerical('int1').set('unit', {'V*m'});
model.result.numerical('int1').set('descr', {'Electric field norm'});
model.result.numerical('int1').setResult;

model.result('pg1').label('Electric Field (ewfd)');
model.result('pg1').set('frametype', 'spatial');
model.result('pg1').feature('mslc1').set('multiplanezmethod', 'coord');
model.result('pg1').feature('mslc1').set('zcoord', '1050.0');
model.result('pg1').feature('mslc1').set('rangecoloractive', true);
model.result('pg1').feature('mslc1').set('rangecolormin', 41923.82486663746);
model.result('pg1').feature('mslc1').set('rangecolormax', 400000);
model.result('pg1').feature('mslc1').set('smooth', 'internal');
model.result('pg1').feature('mslc1').set('resolution', 'normal');
model.result('pg2').feature('slc1').set('quickplane', 'xy');
model.result('pg2').feature('slc1').set('quickzmethod', 'coord');
model.result('pg2').feature('slc1').set('quickz', '1033.0');
model.result('pg2').feature('slc1').set('rangecoloractive', true);
model.result('pg2').feature('slc1').set('rangecolormin', 9292.73707567522);
model.result('pg2').feature('slc1').set('rangecolormax', 250000);
model.result('pg2').feature('slc1').set('smooth', 'internal');
model.result('pg2').feature('slc1').set('resolution', 'normal');
model.result('pg2').feature('slc1').feature('def1').active(false);
model.result('pg2').feature('slc1').feature('def1').set('expr', {'+ewfd.Ex' '+ewfd.Ey' '+ewfd.Ez'});
model.result('pg2').feature('slc1').feature('def1').set('descr', ' + Electric field');
model.result('pg2').feature('slc1').feature('def1').set('scale', 2.085880595474555E-4);
model.result('pg2').feature('slc1').feature('def1').set('scaleactive', false);
model.result('pg3').set('data', 'none');
model.result('pg3').set('xlabel', 'freq (Hz)');
model.result('pg3').set('ylabel', 'Electric field norm (V*m)');
model.result('pg3').set('xlabelactive', false);
model.result('pg3').set('ylabelactive', false);
model.result('pg3').feature('tblp1').set('linewidth', 3);
model.result('pg3').feature('tblp1').set('linemarker', 'cycle');
model.result('pg3').feature('tblp1').set('markerpos', 'datapoints');
end
```

```
function [] = fNiW(model)
model.component('comp1').material.create('mat1', 'Common');
model.component('comp1').material.create('mat4', 'Common');
model.component('comp1').material.create('mat5', 'Common');
model.component('comp1').material('mat1').propertyGroup('def').func.create('eta', 'Piecewise');
model.component('comp1').material('mat1').propertyGroup('def').func.create('Cp', 'Piecewise');
model.component('comp1').material('mat1').propertyGroup('def').func.create('rho', 'Analytic');
model.component('comp1').material('mat1').propertyGroup('def').func.create('k', 'Piecewise');
model.component('comp1').material('mat1').propertyGroup('def').func.create('cs', 'Analytic');
model.component('comp1').material('mat1').propertyGroup.create('RefractiveIndex', 'Refractive index');
model.component('comp1').material('mat4').selection.set([4]);
model.component('comp1').material('mat4').info.create('Composition');
model.component('comp1').material('mat4').propertyGroup('def').func.create('k', 'Piecewise');
model.component('comp1').material('mat4').propertyGroup('def').func.create('alpha', 'Piecewise');
model.component('comp1').material('mat4').propertyGroup('def').func.create('rho', 'Piecewise');
model.component('comp1').material('mat4').propertyGroup.create('ThermalExpansion', 'Thermal expansion');
model.component('comp1').material('mat4').propertyGroup('ThermalExpansion').func.create('dL', 'Piecewise');
model.component('comp1').material('mat4').propertyGroup.create('Enu', 'Young''s modulus and Poisson''s ratio');
model.component('comp1').material('mat4').propertyGroup('Enu').func.create('E', 'Piecewise');
model.component('comp1').material('mat4').propertyGroup.create('RefractiveIndex', 'Refractive index');
model.component('comp1').material('mat5').selection.set([3]);
model.component('comp1').material('mat5').propertyGroup.create('RefractiveIndex', 'Refractive index');
model.component('comp1').material('mat5').propertyGroup('RefractiveIndex').func.create('int1', 'Interpolation');
model.component('comp1').material('mat5').propertyGroup('RefractiveIndex').func.create('int2', 'Interpolation');
model.component('comp1').physics.create('ewfd', 'ElectromagneticWavesFrequencyDomain', 'geom1');
model.component('comp1').physics('ewfd').create('sctr1', 'Scattering', 2);
model.component('comp1').physics('ewfd').feature('sctr1').selection.set([1 2 3 4 5 7 8 9 43 44]);
model.component('comp1').physics('ewfd').create('port1', 'Port', 2);
model.component('comp1').physics('ewfd').feature('port1').selection.set([7]);
model.component('comp1').physics('ewfd').create('port2', 'Port', 2);
model.component('comp1').physics('ewfd').feature('port2').selection.set([3]);
model.component('comp1').mesh('mesh1').create('ftet1', 'FreeTet');
model.result.table('tbl1').comments('Surface Integration 1 (ewfd.normE)');
model.result.table('evl3').label('Evaluation 3D');
model.result.table('evl3').comments('Interactive 3D values');
model.component('comp1').view('view1').set('renderwireframe', true);
model.component('comp1').view('view2').axis.set('xmin', -37.6529541015625);
model.component('comp1').view('view2').axis.set('xmax', 2112.7470703125);
model.component('comp1').view('view2').axis.set('ymin', 0);
model.component('comp1').view('view2').axis.set('ymax', 1651.199951171875);
model.component('comp1').material('mat1').label('Air');
model.component('comp1').material('mat1').set('family', 'air');
model.component('comp1').material('mat1').propertyGroup('def').func('eta').set('arg', 'T');
model.component('comp1').material('mat1').propertyGroup('def').func('eta').set('pieces', {'200.0' '1600.0' ...
    '-8.38278E-7+8.35717342E-8*T^1-7.69429583E-11*T^2+4.6437266E-14*T^3-1.06585607E-17*T^4'});
model.component('comp1').material('mat1').propertyGroup('def').func('Cp').set('arg', 'T');
model.component('comp1').material('mat1').propertyGroup('def').func('Cp').set('pieces', {'200.0' '1600.0' ...
    '1047.63657-0.372589265*T^1+9.45304214E-4*T^2-6.02409443E-7*T^3+1.2858961E-10*T^4'});
model.component('comp1').material('mat1').propertyGroup('def').func('rho').set('expr', 'pA*0.02897/8.314/T');
model.component('comp1').material('mat1').propertyGroup('def').func('rho').set('args', {'pA' 'T'});
model.component('comp1').material('mat1').propertyGroup('def').func('rho').set('dermethod', 'manual');
model.component('comp1').material('mat1').propertyGroup('def').func('rho').set('argders', {'pA' 'd(pA*0.02897/8.314/T,pA)'; 'T' ...
    'd(pA*0.02897/8.314/T,T)'});
model.component('comp1').material('mat1').propertyGroup('def').func('rho').set('plotargs', {'pA' '0' '1'; 'T' '0' '1'});
model.component('comp1').material('mat1').propertyGroup('def').func('k').set('arg', 'T');
model.component('comp1').material('mat1').propertyGroup('def').func('k').set('pieces', {'200.0' '1600.0' ...
    '-0.00227583562+1.15480022E-4*T^1-7.90252856E-8*T^2+4.11702505E-11*T^3-7.43864331E-15*T^4'});
model.component('comp1').material('mat1').propertyGroup('def').func('cs').set('expr', 'sqrt(1.4*287*T)');
model.component('comp1').material('mat1').propertyGroup('def').func('cs').set('args', {'T'});
model.component('comp1').material('mat1').propertyGroup('def').func('cs').set('dermethod', 'manual');
model.component('comp1').material('mat1').propertyGroup('def').func('cs').set('argders', {'T' 'd(sqrt(1.4*287*T),T)'});
model.component('comp1').material('mat1').propertyGroup('def').func('cs').set('plotargs', {'T' '0' '1'});
model.component('comp1').material('mat1').propertyGroup('def').set('relpermeability', {'1' '0' '0' '0' '1' '0' '0' '0' '1'});
model.component('comp1').material('mat1').propertyGroup('def').set('relpermittivity', {'1' '0' '0' '0' '1' '0' '0' '0' '1'});
model.component('comp1').material('mat1').propertyGroup('def').set('dynamicviscosity', 'eta(T[1/K])[Pa*s]');
model.component('comp1').material('mat1').propertyGroup('def').set('ratioofspecificheat', '1.4');
model.component('comp1').material('mat1').propertyGroup('def').set('electricconductivity', {'0[S/m]' '0' '0' '0' '0[S/m]' '0' '0' '0' '0[S/m]'});
```

```
model.component('comp1').material('mat1').propertyGroup('def').set('heatcapacity', 'Cp(T[1/K])[J/(kg*K)]');
model.component('comp1').material('mat1').propertyGroup('def').set('density', 'rho(pA[1/Pa],T[1/K])[kg/m^3]');
model.component('comp1').material('mat1').propertyGroup('def').set('thermalconductivity', {'k(T[1/K])[W/(m*K)]' '0' '0' '0' 'k(T[1/K])[W/(m*K)]' '0' '0' ...
    '0' 'k(T[1/K])[W/(m*K)]'});
model.component('comp1').material('mat1').propertyGroup('def').set('soundspeed', 'cs(T[1/K])[m/s]');
model.component('comp1').material('mat1').propertyGroup('def').addInput('temperature');
model.component('comp1').material('mat1').propertyGroup('def').addInput('pressure');
model.component('comp1').material('mat1').propertyGroup('RefractiveIndex').set('n', '');
model.component('comp1').material('mat1').propertyGroup('RefractiveIndex').set('ki', '');
model.component('comp1').material('mat1').propertyGroup('RefractiveIndex').set('n', {'1' '0' '0' '0' '1' '0' '0' '0' '1'});
model.component('comp1').material('mat1').propertyGroup('RefractiveIndex').set('ki', {'0' '0' '0' '0' '0' '0' '0' '0' '0'});
model.component('comp1').material('mat4').label('Nickel (commerically pure) [solid]');
model.component('comp1').material('mat4').set('family', 'custom');
model.component('comp1').material('mat4').set('specular', 'custom');
model.component('comp1').material('mat4').set('customspecular', [0.7843137254901961 1 1]);
model.component('comp1').material('mat4').set('diffuse', 'custom');
model.component('comp1').material('mat4').set('customdiffuse', [0.7843137254901961 0.7843137254901961 0.7843137254901961]);
model.component('comp1').material('mat4').set('ambient', 'custom');
model.component('comp1').material('mat4').set('customambient', [0.7843137254901961 0.7843137254901961 0.7843137254901961]);
model.component('comp1').material('mat4').set('noise', true);
model.component('comp1').material('mat4').set('noisefreq', 1);
model.component('comp1').material('mat4').set('lighting', 'cooktorrance');
model.component('comp1').material('mat4').set('fresnel', 0.9);
model.component('comp1').material('mat4').set('info', {'Composition' '' 'commerically pure'});
model.component('comp1').material('mat4').propertyGroup('def').func('k').set('arg', 'T');
model.component('comp1').material('mat4').propertyGroup('def').func('k').set('pieces', {'273.0' ...
    '673.0' '82.43111-0.05351518*T^1-6.532678E-5*T^2+9.112509E-8*T^3'; '673.0' '1173.0' ...
    '237.7946-0.6995247*T^1+8.086325E-4*T^2-2.907407E-7*T^3'});
model.component('comp1').material('mat4').propertyGroup('def').func('alpha').set('arg', 'T');
model.component('comp1').material('mat4').propertyGroup('def').func('alpha').set('pieces', {'0.0' '20.0' '8.085811E-6+2.790925E-8*T^1'; '20.0' ...
    '300.0' '8.041785E-6+3.13522E-8*T^1-5.800962E-11*T^2-1.094272E-13*T^3+3.207968E-16*T^4'});
model.component('comp1').material('mat4').propertyGroup('def').func('rho').set('arg', 'T');
model.component('comp1').material('mat4').propertyGroup('def').func('rho').set('pieces', {'0.0' '20.0' ...
    '8953.509-0.008017787*T^1-4.251331E-4*T^2'; '20.0' '300.0' ...
    '8952.813+0.04935669*T^1-0.001622488*T^2+3.023595E-6*T^3-1.90504E-9*T^4'});
model.component('comp1').material('mat4').propertyGroup('def').set('thermalconductivity', {'k(T[1/K])[W/(m*K)]' '0' '0' '0' 'k(T[1/K])[W/(m*K)]' '0' '0' ...
    '0' 'k(T[1/K])[W/(m*K)]'});
model.component('comp1').material('mat4').propertyGroup('def').set('thermalexpansioncoefficient', ...
    {'(alpha(T[1/K])[1/K]+(Tempref-293[K])*if(abs(T-Tempref)>1e-3,(alpha(T[1/K])[1/K]-alpha(Tempref[1/K])[1/K])/(T-Tempref),d(alpha(T[1/K])[1/K],T))
model.component('comp1').material('mat4').propertyGroup('def').set('density', 'rho(T[1/K])[kg/m^3]');
model.component('comp1').material('mat4').propertyGroup('def').addInput('temperature');
model.component('comp1').material('mat4').propertyGroup('def').addInput('strainreferencetemperature');
model.component('comp1').material('mat4').propertyGroup('ThermalExpansion').func('dL').set('arg', 'T');
model.component('comp1').material('mat4').propertyGroup('ThermalExpansion').func('dL').set('pieces', {'0.0' '20.0' ...
    '-0.00237+9.227418E-9*T^1+3.167763E-8*T^2'; '20.0' '300.0' ...
    '-0.002344272-1.825737E-6*T^1+6.010195E-8*T^2-1.109772E-10*T^3+6.933659E-14*T^4'});
model.component('comp1').material('mat4').propertyGroup('ThermalExpansion').set('alphatan', '');
model.component('comp1').material('mat4').propertyGroup('ThermalExpansion').set('dL', '');
model.component('comp1').material('mat4').propertyGroup('ThermalExpansion').set('alphatanIso', '');
model.component('comp1').material('mat4').propertyGroup('ThermalExpansion').set('dLIso', '');
model.component('comp1').material('mat4').propertyGroup('ThermalExpansion').set('alphatan', {'1' '0' '0' '0' '1' '0' '0' '0' '1'});
model.component('comp1').material('mat4').propertyGroup('ThermalExpansion').set('dL', {'(dL(T[1/K])-dL(Tempref[1/K]))/(1+dL(Tempref[1/K]))' '0' ...
    '0' '0' '(dL(T[1/K])-dL(Tempref[1/K]))/(1+dL(Tempref[1/K]))' '0' '0' '0' '(dL(T[1/K])-dL(Tempref[1/K]))/(1+dL(Tempref[1/K]))'});
model.component('comp1').material('mat4').propertyGroup('ThermalExpansion').set('alphatanIso', '1');
model.component('comp1').material('mat4').propertyGroup('ThermalExpansion').set('dLIso', '(dL(T)-dL(Tempref))/(1+dL(Tempref))');
model.component('comp1').material('mat4').propertyGroup('ThermalExpansion').addInput('temperature');
model.component('comp1').material('mat4').propertyGroup('ThermalExpansion').addInput('strainreferencetemperature');
model.component('comp1').material('mat4').propertyGroup('Enu').func('E').set('arg', 'T');
model.component('comp1').material('mat4').propertyGroup('Enu').func('E').set('pieces', {'20.0' '80.0' ...
    '2.128205E11+7.300654E7*T^1-2910689.0*T^2+44672.81*T^3-262.2094*T^4'; '80.0' '311.0' ...
    '2.03931E11+3.899186E8*T^1-5310091.0*T^2+25603.5*T^3-55.59948*T^4+0.04649125*T^5'});
model.component('comp1').material('mat4').propertyGroup('Enu').set('youngsmodulus', 'E(T[1/K])[Pa]');
model.component('comp1').material('mat4').propertyGroup('Enu').set('poissonsratio', '1');
model.component('comp1').material('mat4').propertyGroup('Enu').addInput('temperature');
model.component('comp1').material('mat4').propertyGroup('RefractiveIndex').set('n', '');
model.component('comp1').material('mat4').propertyGroup('RefractiveIndex').set('ki', '');
model.component('comp1').material('mat4').propertyGroup('RefractiveIndex').set('n', {'2.0211' '0' '0' '0' '2.0211' '0' '0' '0' '2.0211'});
model.component('comp1').material('mat4').propertyGroup('RefractiveIndex').set('ki', {'2.1822' '0' '0' '0' '2.1822' '0' '0' '0' '2.1822'});
model.component('comp1').material('mat5').label(['W (Tungsten) (Ordal et al. 1988: n,k 0.667-200 ' native2unicode(hex2dec({'00' 'b5'}), ...
    'unicode') 'm)']);
model.component('comp1').material('mat5').set('groups', {'inorganic' 'Inorganic Materials'; 'w_tungsten_and_tungstates' ...
    'W - Tungsten and tungstates'; 'experimental_data' 'Experimental data'});
model.component('comp1').material('mat5').propertyGroup('RefractiveIndex').func('int1').label('Refractive index');
model.component('comp1').material('mat5').propertyGroup('RefractiveIndex').func('int1').set('funcname', 'n_interp');
model.component('comp1').material('mat5').propertyGroup('RefractiveIndex').func('int1').set('table', {'6.67E-7' '3.8312601';'7.14E-7' '3.9313491';  ...
'7.69E-7' '3.778336';'8.3299999999999999E-7' '3.5190631';'9.09E-7' '3.2814572';'1.0E-6' '3.0826871';'1.05E-6' '3.0570934';  ...
'1.11E-6' '3.0821649';'1.18E-6' '3.1599449';'1.2499999999999999E-6' '3.1960757';'1.33E-6' '3.1990778';'1.4299999999999999E-6' '3.0278249';  ...
'1.5399999999999999E-6' '2.2587782';'1.6699999999999999E-6' '1.8828191';'1.82E-6' '1.5455798';'2.0E-6' '1.2992808';  ...
'2.1099999999999997E-6' '1.1946455';'2.22E-6' '1.221203';'2.35E-6' '1.2886548';'2.4999999999999998E-6' '1.4507921';  ...
'2.67E-6' '1.6746349';'2.8599999999999997E-6' '1.8748013';'3.0799999999999997E-6' '1.91692';'3.33E-6' '1.8505739';'3.64E-6' '1.8774806';  ...
'4.0E-6' '1.9949739';'4.44E-6' '2.2388765';'4.9999999999999996E-6' '2.6672954';'5.7099999999999995E-6' '3.3712766';  ...
'6.67E-6' '4.4961625';'8.0E-6' '6.2868052';'9.999999999999999E-6' '9.1935519';'1.1099999999999999E-5' '10.843912';  ...
'1.2499999999999999E-5' '12.935818';'1.43E-5' '15.634532';'1.67E-5' '19.231714';'1.9999999999999998E-5' '24.256111';  ...
'2.2199999999999998E-5' '27.599188';'2.4999999999999998E-5' '31.77255';'2.86E-5' '37.127733';'3.3299999999999996E-5' '44.209069';  ...
'3.9999999999999996E-5' '54.191208';'4.4399999999999995E-5' '61.731428';'4.9999999999999996E-5' '68.665703';  ...
```

```
'5.71E-5' '76.505227';'6.67E-5' '87.148396';'7.999999999999999E-5' '103.53408';'9.999999999999999E-5' '130.219';'1.25E-4' '163.02849'; ...
'1.54E-4' '196.84283';'1.9999999999999998E-4' '242.14161'});
model.component('comp1').material('mat5').propertyGroup('RefractiveIndex').func('int1').set('interp', 'piecewisecubic');
model.component('comp1').material('mat5').propertyGroup('RefractiveIndex').func('int1').set('argunit', 'm');
model.component('comp1').material('mat5').propertyGroup('RefractiveIndex').func('int1').set('fununit', '1');
model.component('comp1').material('mat5').propertyGroup('RefractiveIndex').func('int2').label('Refractive index, imaginary part');
model.component('comp1').material('mat5').propertyGroup('RefractiveIndex').func('int2').set('funcname', 'k_interp');
model.component('comp1').material('mat5').propertyGroup('RefractiveIndex').func('int2').set('table', {'6.67E-7' '2.9042727';'7.14E-7' '2.7924078'; ...
'7.69E-7' '2.6322954';'8.329999999999999E-7' '2.76706';'9.09E-7' '3.0100921';'1.0E-6' '3.4208368';'1.05E-6' '3.722401'; ...
'1.11E-6' '3.9693342';'1.18E-6' '4.1791582';'1.2499999999999999E-6' '4.3367596';'1.33E-6' '4.4460869';'1.429999999999999E-6' '4.3901355'; ...
'1.5399999999999999E-6' '4.7741297';'1.6699999999999999E-6' '5.4881071';'1.82E-6' '6.4453363';'2.0E-6' '7.5659499'; ...
'2.1099999999999997E-6' '8.2613235';'2.22E-6' '9.0476312';'2.35E-6' '9.8569158';'2.4999999999999998E-6' '10.710594'; ...
'2.67E-6' '11.554665';'2.8599999999999997E-6' '12.375593';'3.0799999999999997E-6' '13.261664';'3.33E-6' '14.40046'; ...
'3.64E-6' '15.887186';'4.0E-6' '17.69477';'4.44E-6' '19.89851';'4.9999999999999996E-6' '22.612178';'5.7099999999999995E-6' '26.005432'; ...
'6.67E-6' '30.343034';'8.0E-6' '36.072044';'9.999999999999999E-6' '44.019189';'1.1099999999999999E-5' '48.146015'; ...
'1.2499999999999999E-5' '53.112521';'1.43E-5' '59.208636';'1.67E-5' '66.923752';'1.9999999999999998E-5' '77.087223'; ...
'2.2199999999999998E-5' '83.517367';'2.4999999999999998E-5' '91.217414';'2.86E-5' '100.62038';'3.3299999999999996E-5' '112.34062'; ...
'3.9999999999999996E-5' '128.08146';'4.4399999999999995E-5' '136.67195';'4.9999999999999996E-5' '146.58093'; ...
'5.71E-5' '159.83902';'6.67E-5' '177.92054';'7.999999999999999E-5' '202.21197';'9.999999999999999E-5' '234.03939'; ...
'1.25E-4' '265.86909';'1.54E-4' '294.94073';'1.9999999999999998E-4' '332.81796'});
model.component('comp1').material('mat5').propertyGroup('RefractiveIndex').func('int2').set('interp', 'piecewisecubic');
model.component('comp1').material('mat5').propertyGroup('RefractiveIndex').func('int2').set('argunit', 'm');
model.component('comp1').material('mat5').propertyGroup('RefractiveIndex').func('int2').set('fununit', '1');
model.component('comp1').material('mat5').propertyGroup('RefractiveIndex').set('n', '');
model.component('comp1').material('mat5').propertyGroup('RefractiveIndex').set('ki', '');
model.component('comp1').material('mat5').propertyGroup('RefractiveIndex').set('n', '');
model.component('comp1').material('mat5').propertyGroup('RefractiveIndex').set('ki', '');
model.component('comp1').material('mat5').propertyGroup('RefractiveIndex').set('n', {'n_interp(c_const/freq)' '0' '0' '0' 'n_interp(c_const/freq)' '0' ...
    '0' '0' 'n_interp(c_const/freq)'});
model.component('comp1').material('mat5').propertyGroup('RefractiveIndex').set('ki', {'k_interp(c_const/freq)' '0' '0' '0' 'k_interp(c_const/freq)' '0' ...
    '0' '0' 'k_interp(c_const/freq)'});
model.component('comp1').material('mat5').propertyGroup('RefractiveIndex').addInput('frequency');
model.component('comp1').physics('ewfd').prop('EquationForm').set('freq', '1[GHz]');
model.component('comp1').physics('ewfd').prop('EquationForm').set('modeFreq', '1[GHz]');
model.component('comp1').physics('ewfd').prop('MeshControl').set('EnableMeshControl', false);
model.component('comp1').physics('ewfd').feature('port1').set('Pin', '1[mW]');
model.component('comp1').physics('ewfd').feature('port1').set('beta', 'abs(ewfd.k0)');
model.component('comp1').physics('ewfd').feature('port1').set('E0', {'0'; 'exp(-j*ewfd.k0*z)'; '0'});
model.component('comp1').physics('ewfd').feature('port2').set('beta', 'abs(ewfd.k0)');
model.component('comp1').physics('ewfd').feature('port2').set('E0', {'0'; 'exp(-j*ewfd.k0*z)'; '0'});
model.component('comp1').mesh('mesh1').feature('size').set('hauto', 4);
model.component('comp1').mesh('mesh1').run;
model.study.create('std1');
model.study('std1').create('freq', 'Frequency');
model.sol.create('sol1');
model.sol('sol1').study('std1');
model.sol('sol1').attach('std1');
model.sol('sol1').create('st1', 'StudyStep');
model.sol('sol1').create('v1', 'Variables');
model.sol('sol1').create('s1', 'Stationary');
model.sol('sol1').feature('s1').create('p1', 'Parametric');
model.sol('sol1').feature('s1').create('fc1', 'FullyCoupled');
model.sol('sol1').feature('s1').create('i1', 'Iterative');
model.sol('sol1').feature('s1').feature('i1').create('mg1', 'Multigrid');
model.sol('sol1').feature('s1').feature('i1').feature('mg1').feature('pr').create('sv1', 'SORVector');
model.sol('sol1').feature('s1').feature('i1').feature('mg1').feature('po').create('sv1', 'SORVector');
model.sol('sol1').feature('s1').feature.remove('fcDef');
model.result.numerical.create('int1', 'IntSurface');
model.result.numerical('int1').selection.set([13 29]);
model.result.numerical('int1').set('probetag', 'none');
model.result.create('pg1', 'PlotGroup3D');
model.result.create('pg2', 'PlotGroup3D');
model.result.create('pg3', 'PlotGroup1D');
model.result('pg1').create('mslc1', 'Multislice');
model.result('pg2').create('slc1', 'Slice');
model.result('pg2').feature('slc1').create('def1', 'Deform');
```

```
model.result('pg3').create('tblp1', 'Table');
model.study('std1').feature('freq').set('punit', 'Hz');
model.study('std1').feature('freq').set('plist', '1.0e12');
model.sol('sol1').attach('std1');
model.sol('sol1').feature('v1').set('clistctrl', {'p1'});
model.sol('sol1').feature('v1').set('cname', {'freq'});
model.sol('sol1').feature('v1').set('clist', {'1.0e12[Hz]'});
model.sol('sol1').feature('s1').feature('p1').set('pname', {'freq'});
model.sol('sol1').feature('s1').feature('p1').set('plistarr', {'1.0e12'});
model.sol('sol1').feature('s1').feature('p1').set('punit', {'Hz'});
model.sol('sol1').feature('s1').feature('p1').set('pcontinuationmode', 'no');
model.sol('sol1').feature('s1').feature('p1').set('preusesol', 'auto');
model.sol('sol1').feature('s1').feature('i1').label('Suggested Iterative Solver (ewfd)');
model.sol('sol1').feature('s1').feature('i1').set('linsolver', 'bicgstab');
model.sol('sol1').feature('s1').feature('i1').feature('mg1').feature('pr').feature('sv1').set('sorvecdof', {'comp1_E'});
model.sol('sol1').feature('s1').feature('i1').feature('mg1').feature('po').feature('sv1').set('sorvecdof', {'comp1_E'});
model.sol('sol1').runAll;
model.result.numerical('int1').set('table', 'tbl1');
model.result.numerical('int1').set('expr', {'ewfd.normE'});
model.result.numerical('int1').set('unit', {'V*m'});
model.result.numerical('int1').set('descr', {'Electric field norm'});
model.result.numerical('int1').setResult;
model.result('pg1').label('Electric Field (ewfd)');
model.result('pg1').set('frametype', 'spatial');
model.result('pg1').feature('mslc1').set('multiplanezmethod', 'coord');
model.result('pg1').feature('mslc1').set('zcoord', '1050.0');
model.result('pg1').feature('mslc1').set('rangecoloractive', true);
model.result('pg1').feature('mslc1').set('rangecolormin', 41923.82486663746);
model.result('pg1').feature('mslc1').set('rangecolormax', 400000);
model.result('pg1').feature('mslc1').set('smooth', 'internal');
model.result('pg1').feature('mslc1').set('resolution', 'normal');
model.result('pg2').feature('slc1').set('quickplane', 'xy');
model.result('pg2').feature('slc1').set('quickzmethod', 'coord');
model.result('pg2').feature('slc1').set('quickz', '1033.0');
model.result('pg2').feature('slc1').set('rangecoloractive', true);
model.result('pg2').feature('slc1').set('rangecolormin', 9292.73707567522);
model.result('pg2').feature('slc1').set('rangecolormax', 250000);
model.result('pg2').feature('slc1').set('smooth', 'internal');
model.result('pg2').feature('slc1').set('resolution', 'normal');
model.result('pg2').feature('slc1').feature('def1').active(false);
model.result('pg2').feature('slc1').feature('def1').set('expr', {'+ewfd.Ex' '+ewfd.Ey' '+ewfd.Ez'});
model.result('pg2').feature('slc1').feature('def1').set('descr', ' + Electric field');
model.result('pg2').feature('slc1').feature('def1').set('scale', 2.085880595474555E-4);
model.result('pg2').feature('slc1').feature('def1').set('scaleactive', false);
model.result('pg3').set('data', 'none');
model.result('pg3').set('xlabel', 'freq (Hz)');
model.result('pg3').set('ylabel', 'Electric field norm (V*m)');
model.result('pg3').set('xlabelactive', false);
model.result('pg3').set('ylabelactive', false);
model.result('pg3').feature('tblp1').set('linewidth', 3);
model.result('pg3').feature('tblp1').set('linemarker', 'cycle');
model.result('pg3').feature('tblp1').set('markerpos', 'datapoints');
end
```