

# Análisis y Simulación Antena de Grafeno basada en el Triángulo Fractal de Sierpiński.

**Katya Máñez Marquina**

Grado en Ingeniería de Tecnologías y Servicios de Telecomunicación  
Análisis de Antenas

**Dr. Jaume Anguera Pros**

**Dra. Aurora Andújar Linares**

21 de Junio de 2022



Esta obra está sujeta a una licencia de [Reconocimiento-NoComercial-SinObraDerivada 3.0 España de Creative Commons](https://creativecommons.org/licenses/by-nc-nd/3.0/es/)

## **Dedicatoria**

A mis hijos, por hacerme ser mejor persona cada día.

A mi marido, por acompañarme incondicionalmente en este largo camino.

A mis padres, porque sin vosotros esto no hubiera sido posible y ojalá podáis saber de algún modo que llegué al final.

## **Agradecimientos**

A mi director de proyecto por su magistral asesoramiento, su coherencia y su optimismo. Gracias a sus buenos consejos hemos dado forma a este proyecto.

## FICHA DEL TRABAJO FINAL

<b>Título del trabajo:</b>	<i>Análisis y Simulación Antena de Grafeno basada en el Triángulo Fractal de Sierpiński</i>
<b>Nombre del autor:</b>	<i>Katya Máñez Marquina</i>
<b>Nombre del consultor/a:</b>	<i>Jaume Anguera Pros, Aurora Andújar Linares</i>
<b>Nombre del PRA:</b>	<i>Germán Cobo Rodríguez</i>
<b>Fecha de entrega (mm/aaaa):</b>	06/2022
<b>Titulación::</b>	<i>Grado en Ingeniería de Tecnologías y Servicios de Telecomunicación</i>
<b>Área del Trabajo Final:</b>	<i>Análisis de Antenas</i>
<b>Idioma del trabajo:</b>	<i>Castellano</i>
<b>Palabras clave</b>	<i>antenas, fractales, grafeno y terahercios.</i>
<p><b>Resumen del Trabajo (máximo 250 palabras):</b> <i>Con la finalidad, contexto de aplicación, metodología, resultados i conclusiones del trabajo.</i></p>	
<p>El importante crecimiento de los dispositivos y las redes de comunicaciones acontecido en los últimos tiempos junto con la necesidad de proveer de comunicaciones eficientes, hace que se busquen diseños de antenas miniaturizadas basados en nuevos materiales que provean de mayores anchos de banda y por lo tanto mejores velocidades de transmisión.</p> <p>En el presente proyecto se diseñan y analizan varias versiones de una antena fractal de grafeno. Por un lado se ha elegido este nanomaterial puesto que presenta una conductividad eléctrica muy alta y aplicado a una antena de dimensiones del orden de nanómetros, permite la propagación de las ondas electromagnéticas en las frecuencias de terahercios. Por otro lado, se ha seleccionado un diseño fractal porque éstos proporcionan un gran ancho de banda, comportamiento multibanda y una elevada ganancia.</p> <p>Para alcanzar el objetivo planteado se ha desarrollado el código de Matlab necesario para integrarse con CST Studio Suite, de modo que se pudiera diseñar la antena microstrip fractal para diversos órdenes, de forma rápida y eficiente. Así pues, se han podido comparar los distintos diseños implementados con diferentes materiales para obtener las conclusiones.</p> <p>Como conclusión se ha verificado lo que demostraban trabajos previos sobre la influencia del potencial químico del grafeno con respecto a la frecuencia de resonancia de la antena y con el cambio de sustrato.</p>	

**Abstract (in English, 250 words or less):**

The significant growth of communication devices and networks in recent times, together with the need to provide efficient communications, has led to the search for miniaturized antenna designs based on new materials that provide higher bandwidths and therefore better transmission speeds.

In the present project a graphene fractal antenna is designed and analyzed. On the one hand, this nanomaterial has been chosen because it has a very high electrical conductivity and when applied to an antenna of nanometer dimensions, it allows the propagation of electromagnetic waves at terahertz frequencies. On the other hand, a fractal design has been selected because it provides a high bandwidth, multiband behavior and high gain.

In order to achieve the proposed objective, the Matlab code needed to integrate with CST Studio Suite has been developed, so that the fractal microstrip antenna could be designed for different orders quickly and efficiently. Thus, it has been possible to compare the different designs implemented with different materials to obtain the conclusions.

In conclusion, previous work on the influence of the chemical potential of graphene with respect to the resonance frequency of the antenna and with the change of substrate has been verified.

# Índice

1. Introducción .....	1
1.1 Contexto y justificación del Trabajo .....	1
1.2 Objetivos del Trabajo.....	2
1.3 Enfoque y método seguido.....	2
1.4 Planificación del Trabajo.....	3
1.5 Breve resumen de productos obtenidos .....	6
1.6 Breve descripción de los otros capítulos de la memoria .....	6
2. Estado de la cuestión.....	7
2.1 Antenas fractales .....	7
2.1.1 Definición de fractal .....	7
2.1.1.1 Conjuntos de Cantor.....	8
2.1.1.2 Curvas de Koch .....	8
2.1.1.3 Curva de Peano .....	9
2.1.1.4 Curva de Hilbert.....	9
2.1.1.5 Triángulos de Sierpiński .....	9
2.1.2 Tipos de antenas fractales.....	10
2.1.2.1 Mandelbrot.....	10
2.1.2.2 Sierpiński .....	10
2.1.2.3 Koch .....	11
2.1.2.4 Hilbert .....	11
2.1.2.5 Cantor .....	12
2.1.2.6 Peano .....	12
2.2 Las propiedades del grafeno .....	13
2.2.1 Definición .....	13
2.2.2 Propiedades .....	13
2.2.3 Fabricación .....	13
2.3 Usos comerciales antenas grafeno THz .....	14
2.4 Las antenas de grafeno.....	15
2.4.1 Trabajo: <i>“Graphene-based nano-patch antenna for terahertz radiation”</i> .....	15
2.4.2 Trabajo: <i>“Terahertz Graphene-Based Reconfigurable Patch Antenna”</i> .....	17
2.4.3 Trabajo: <i>“Dual-Band Reconfigurable Graphene-Based Patch Antenna in Terahertz Band: Design, Analysis and Modeling Using WCIP Method”</i> .....	19
2.4.4 Trabajo: <i>“Reconfigurable Graphene Annular Ring Antenna for Medical and Imaging Applications”</i> .....	21
2.4.5 Trabajo: <i>“A Duo of Graphene-Copper Based Wideband Planar Plasmonic Antenna Analysis for Lower Region of Terahertz (THz) Communications”</i> .....	23
2.5 Las antenas fractales de grafeno .....	26
2.5.1 Trabajo: <i>“Design of a Novel Reconfigurable Sierpiński Fractal Graphene Antenna Operating at THz Band”</i> .....	26
2.5.2 Trabajo: <i>“A Preliminary Study of a Graphene Fractal Sierpiński Antenna”</i> .....	29
3. Diseño.....	32
3.1 Enfoque de la solución .....	32
3.2 Antena propuesta .....	32
3.3 Creación de material grafeno .....	33
3.4 Creación de sustrato <i>glass_epoxy</i> .....	35
3.5 Ejemplo de funcionamiento .....	35
3.6 Cálculos y simulaciones realizadas .....	41
3.6.1 Post-procesado en CST de Ganancia, Ganancia Realizable y Directividad .....	42
4. Modelos simulados.....	44
4.1. Modelo 1: Fractal orden 1 con grafeno y sustrato de silicio .....	44
4.1.1. Componente .....	44
4.1.2. Coeficiente de Reflexión $S_{11}$ .....	44

4.1.3.	<i>Farfields</i> para las frecuencia de resonancia .....	45
4.1.4.	Directividad .....	46
4.1.5.	Eficiencias de Radiación .....	46
4.1.6.	Ganancia .....	47
4.1.7.	Ganancia Realizable .....	47
4.1.8.	Potencia Total Radiada (TRP) .....	48
4.2.	Modelo 2: Fractal orden 1 con grafeno y sustrato fibra de vidrio .....	49
4.2.1.	Componente .....	49
4.2.2.	Coefficiente de Reflexión $S_{11}$ .....	49
4.2.3.	<i>Farfields</i> para las frecuencia de resonancia .....	50
4.2.4.	Directividad .....	50
4.2.5.	Eficiencias de Radiación .....	51
4.2.6.	Ganancia .....	51
4.2.7.	Ganancia Realizable .....	52
4.2.8.	Potencia Total Radiada (TRP) .....	52
4.3.	Modelo 3: Fractal orden 1 con grafeno y sustrato de silicio .....	53
4.3.1.	Componente .....	53
4.3.2.	Coefficiente de Reflexión $S_{11}$ .....	53
4.3.3.	<i>Farfields</i> para las frecuencia de resonancia .....	54
4.3.4.	Directividad .....	54
4.3.5.	Eficiencias de Radiación .....	55
4.3.6.	Ganancia .....	56
4.3.7.	Ganancia Realizable .....	56
4.3.8.	Potencia Total Radiada (TRP) .....	56
4.4.	Modelo 4: Fractal orden 1 con grafeno 2 y sustrato fibra de vidrio .....	57
4.4.1.	Componente .....	57
4.4.2.	Coefficiente de Reflexión $S_{11}$ .....	57
4.4.3.	<i>Farfields</i> para las frecuencia de resonancia .....	58
4.4.4.	Directividad .....	58
4.4.5.	Eficiencias de Radiación .....	59
4.4.6.	Ganancia .....	60
4.4.7.	Ganancia Realizable .....	60
4.4.8.	Potencia Total Radiada (TRP) .....	60
4.5.	Modelo 5: Fractal orden 2 con grafeno y sustrato silicio .....	61
4.5.1.	Componente .....	61
4.5.2.	Coefficiente de Reflexión $S_{11}$ .....	61
4.5.3.	<i>Farfields</i> para las frecuencia de resonancia .....	62
4.5.4.	Directividad .....	62
4.5.5.	Eficiencias de Radiación .....	63
4.5.6.	Ganancia .....	64
4.5.7.	Ganancia Realizable .....	64
4.5.8.	Potencia Total Radiada (TRP) .....	64
4.6.	Modelo 6: Fractal orden 2 con grafeno y sustrato fibra de vidrio .....	65
4.6.1.	Componente .....	65
4.6.2.	Coefficiente de Reflexión $S_{11}$ .....	65
4.6.3.	<i>Farfields</i> para las frecuencia de resonancia .....	66
4.6.4.	Directividad .....	66
4.6.5.	Eficiencias de Radiación .....	67
4.6.6.	Ganancia .....	67
4.6.7.	Ganancia Realizable .....	68
4.6.8.	Potencia Total Radiada (TRP) .....	68
4.7.	Modelo 7: Fractal orden 2 con grafeno 2 y sustrato silicio .....	69
4.7.1.	Componente .....	69
4.7.2.	Coefficiente de Reflexión $S_{11}$ .....	69
4.7.3.	<i>Farfields</i> para las frecuencia de resonancia .....	70
4.7.4.	Directividad .....	71
4.7.5.	Eficiencias de Radiación .....	71

4.7.6. Ganancia .....	72
4.7.7. Ganancia Realizable .....	72
4.7.8. Potencia Total Radiada (TRP).....	72
4.8. Modelo 8: Fractal orden 2 con grafeno y sustrato fibra de vidrio .....	73
4.8.1. Componente .....	73
4.8.2. Coeficiente de Reflexión $S_{11}$ .....	73
4.8.3. <i>Farfields</i> para las frecuencia de resonancia .....	74
4.8.4. Directividad .....	74
4.8.5. Eficiencias de Radiación .....	75
4.8.6. Ganancia .....	75
4.8.7. Ganancia Realizable .....	76
4.8.8. Potencia Total Radiada (TRP).....	76
4.9. Modelo 9: Fractal orden 2 con grafeno y sustrato silicio .....	77
4.9.1. Componente .....	77
4.9.2. Coeficiente de Reflexión $S_{11}$ .....	77
4.9.3. <i>Farfields</i> para las frecuencia de resonancia .....	78
4.9.4. Directividad .....	78
4.9.5. Eficiencias de Radiación .....	79
4.9.6. Ganancia .....	79
4.9.7. Ganancia Realizable .....	80
4.9.8. Potencia Total Radiada (TRP).....	80
4.10. Modelo 10: Fractal orden 2 con grafeno 4 y sustrato fibra de vidrio .....	81
4.10.1. Componente .....	81
4.10.2. Coeficiente de Reflexión $S_{11}$ .....	81
4.10.3. <i>Farfields</i> para las frecuencia de resonancia .....	82
4.10.4. Directividad .....	82
4.10.5. Eficiencias de Radiación .....	83
4.10.6. Ganancia .....	83
4.10.7. Ganancia Realizable .....	84
4.10.8. Potencia Total Radiada (TRP).....	84
4.11. Modelo 11: Fractal orden 3 con grafeno y sustrato silicio .....	85
4.11.1. Componente .....	85
4.11.2. Coeficiente de Reflexión $S_{11}$ .....	85
4.11.3. <i>Farfields</i> para las frecuencia de resonancia .....	86
4.11.4. Directividad .....	86
4.11.5. Eficiencias de Radiación .....	87
4.11.6. Ganancia .....	87
4.11.7. Ganancia Realizable .....	88
4.11.8. Potencia Total Radiada (TRP).....	88
4.12. Modelo 12: Fractal orden 3 con grafeno y sustrato fibra de vidrio .....	89
4.12.1. Componente .....	89
4.12.2. Coeficiente de Reflexión $S_{11}$ .....	89
4.12.3. <i>Farfields</i> para las frecuencia de resonancia .....	90
4.12.4. Directividad .....	90
4.12.5. Eficiencias de Radiación .....	91
4.12.6. Ganancia .....	91
4.12.7. Ganancia Realizable .....	92
4.12.8. Potencia Total Radiada (TRP).....	92
4.13. Modelo 13: Fractal orden 2 con grafeno 2 y sustrato silicio .....	93
4.13.1. Componente .....	93
4.13.2. Coeficiente de Reflexión $S_{11}$ .....	93
4.13.3. <i>Farfields</i> para las frecuencia de resonancia .....	94
4.13.4. Directividad .....	94
4.13.5. Eficiencias de Radiación .....	94
4.13.6. Ganancia .....	95
4.13.7. Ganancia Realizable .....	95
4.13.8. Potencia Total Radiada (TRP).....	95



4.14. Modelo 14: Fractal orden 3 con grafeno 2 y sustrato fibra de vidrio .....	96
4.14.1. Componente .....	96
4.14.2. Coeficiente de Reflexión $S_{11}$ .....	96
4.14.3. <i>Farfields</i> para las frecuencia de resonancia .....	97
4.14.4. Directividad .....	97
4.14.5. Eficiencias de Radiación .....	98
4.14.6. Ganancia .....	98
4.14.7. Ganancia Realizable .....	98
4.14.8. Potencia Total Radiada (TRP) .....	98
4.15 Resumen comparativo modelos .....	99
4.16 Mejor modelo orden 1 .....	100
4.17 Mejor modelo orden 2 .....	102
5. Conclusiones .....	104
6. Líneas futuras .....	106
7. Glosario .....	107
8. Bibliografía .....	108
9. Anexo I .....	111

# Lista de figuras

Figura 1. Diagrama de Gantt con la planificación .....	5
Figura 2. Conjunto de Mandelbrot .....	7
Figura 3. Conjunto de Cantor .....	8
Figura 4. Curva de Koch .....	8
Figura 5. Curva de Peano.....	9
Figura 6. Curva de Hilbert.....	9
Figura 7. Triángulo de Sierpiński.....	9
Figura 8. Antena parche microstrip basada en el fractal de Mandelbrot.....	10
Figura 9. Antena de parche microstrip basada en Sierpiński alimentada con sonda coaxial .....	11
Figura 10. Antena parche microstrip basada en la isla de Koch .....	11
Figura 11. Monopolo de Hilbert para varias iteraciones .....	11
Figura 12. Antena basada en la geometría de Cantor .....	12
Figura 13. Antena hilo de alta directividad y monopolo .....	12
Figura 14. Esquema nanoantena grafeno.....	15
Figura 15. Conductividad total vs conductividad intrabanda .....	15
Figura 16. Frecuencia vs ancho (W).....	16
Figura 17. Frecuencia resonancia vs polarización electrostática o temperatura .....	16
Figura 18. Esquema antena parche microstrip .....	17
Figura 19. Pérdidas de retorno antena parche microstrip de cobre .....	17
Figura 20. Diagrama radiación antena parche microstrip de cobre .....	18
Figura 21. Comparación pérdidas de retorno.....	18
Figura 22. Diagrama radiación antena parche microstrip de grafeno .....	18
Figura 23. Estructura de la antena propuesta .....	19
Figura 24. Coeficiente Reflexión configuraciones .....	19
Figura 25. Coeficiente Reflexión para distintos potenciales químicos de grafeno.....	20
Figura 26. Comparación distintos sustratos .....	20
Figura 27. Esquema de antena propuesta .....	21
Figura 28. Pérdidas de retorno para antena de grafeno y antena metálica.....	21
Figura 29. Pérdida de retorno para los diferentes valores de los potenciales químicos.....	22
Figura 30. Rendimiento de la antena propuesta para varios valores de potencial químico .....	22
Figura 31. Diagrama de radiación.....	22
Figura 32. Parte real e imaginaria de la conductividad superficial del grafeno.....	23
Figura 33. Diseño de antena de parche plasmónico de grafeno PPP .....	24
Figura 34. Comparación del $S_{11}$ simulado .....	24
Figura 35. Efecto del potencial químico .....	25
Figura 36. Eficiencia de radiación .....	25
Figura 37. Conductividad superficial del grafeno vs potencial químico o frecuencia .....	26
Figura 38. Esquemas antenas fractales de Sierpiński desde el orden cero al tercero.....	27
Figura 39. Simulación del parámetro $S_{11}$ para diferentes valores de $\mu_c$ .....	27
Figura 40. Diagramas de radiación para diferentes valores de $\mu_c$ .....	28
Figura 41. Modelo de antena Sierpiński .....	29
Figura 42. Parámetro $S_{11}$ global de Sierpiński vs la frecuencia con tres resonancias a 35, 110 y 215 THz. 30	
Figura 43. Patrones de coordenadas polares de campo lejano en las frecuencias de resonancia.....	30
Figura 44. Diagrama de radiación en 3D .....	31
Figura 45. Campos E superficiales de Sierpiński en las frecuencias de resonancia .....	31
Figura 46. Campos E de la sección media de Sierpiński en las frecuencias de resonancia .....	31
Figura 47. Dimensiones antena propuesta.....	32
Figura 48. Materiales del componente .....	33
Figura 49. Macro CST material grafeno .....	33
Figura 50. Parámetros material grafeno .....	34

Figura 51. Material grafeno .....	34
Figura 52. Añadir grafeno a la librería de materiales .....	34
Figura 53. Función fractal_sierpinski ejemplo 1 .....	36
Figura 54. Nuevo proyecto CST ejemplo 1 .....	36
Figura 55. Simulación CST ejemplo 1 .....	37
Figura 56. Simulación finalizada ejemplo 1 .....	37
Figura 57. Parámetro $S_{11}$ ejemplo 1 .....	38
Figura 58. Diagrama de radiación ejemplo 1 .....	38
Figura 59. Función fractal_sierpinski ejemplo 2 .....	39
Figura 60. Nuevo proyecto CST ejemplo 2 .....	39
Figura 61. Simulación CST ejemplo 2 .....	40
Figura 62. Simulación finalizada ejemplo 2 .....	40
Figura 63. Diagrama de radiación ejemplo 2 .....	40
Figura 64. Parche rectangular sobre sustrato dieléctrico .....	41
Figura 65. <i>Template Based Postprocessing</i> .....	42
Figura 66. <i>Template Based Postprocessing</i> monitor .....	42
Figura 67. <i>Template Based Postprocessing</i> parámetros .....	43
Figura 68. <i>Farfield</i> definiciones de CST .....	43
Figura 69. Modelo 1: Componente .....	44
Figura 70. Modelo 1: $S_{11}$ .....	44
Figura 71. Modelo 1: Diagrama radiación 3D $f_{r1}$ .....	45
Figura 72. Modelo 1: Diagrama radiación 3D $f_{r2}$ .....	45
Figura 73. Modelo 1: Diagrama radiación 3D $f_{r3}$ .....	45
Figura 74. Modelo 1: Directividad .....	46
Figura 75. Modelo 1: Eficiencia de Radiación .....	46
Figura 76. Modelo 1: Eficiencia Total .....	47
Figura 77. Modelo 1: Ganancia .....	47
Figura 78. Modelo 1: Ganancia Realizable .....	48
Figura 79. Modelo 2: Componente .....	49
Figura 80. Modelo 2: $S_{11}$ .....	49
Figura 81. Modelo 2: Diagrama radiación 3D $f_{r1}$ .....	50
Figura 82. Modelo 2: Diagrama radiación 3D $f_{r2}$ .....	50
Figura 83. Modelo 2: Directividad .....	50
Figura 84. Modelo 2: Eficiencia de Radiación .....	51
Figura 85. Modelo 2: Eficiencia Total .....	51
Figura 86. Modelo 2: Ganancia .....	52
Figura 87. Modelo 2: Ganancia Realizable .....	52
Figura 88. Modelo 3: Componente .....	53
Figura 89. Modelo 3: $S_{11}$ .....	53
Figura 90. Modelo 3: Diagrama radiación 3D $f_{r1}$ .....	54
Figura 91. Modelo 3: Diagrama radiación 3D $f_{r2}$ .....	54
Figura 92. Modelo 3: Diagrama radiación 3D $f_{r3}$ .....	54
Figura 93. Modelo 3: Directividad .....	55
Figura 94. Modelo 3: Eficiencia de Radiación .....	55
Figura 95. Modelo 3: Eficiencia Total .....	55
Figura 96. Modelo 3: Ganancia .....	56
Figura 97. Modelo 3: Ganancia Realizable .....	56
Figura 98. Modelo 4: Componente .....	57
Figura 99. Modelo 4: $S_{11}$ .....	57
Figura 100. Modelo 4: Diagrama radiación 3D $f_{r1}$ .....	58
Figura 101. Modelo 4: Diagrama radiación 3D $f_{r2}$ .....	58
Figura 102. Modelo 4: Directividad .....	59
Figura 103. Modelo 4: Eficiencia de Radiación .....	59
Figura 104. Modelo 4: Eficiencia Total .....	59
Figura 105. Modelo 4: Ganancia .....	60
Figura 106. Modelo 4: Ganancia Realizable .....	60
Figura 107. Modelo 5: Componente .....	61

Figura 108. Modelo 5: $S_{11}$ .....	61
Figura 109. Modelo 5: Diagrama radiación 3D $f_{r1}$ .....	62
Figura 110. Modelo 5: Diagrama radiación 3D $f_{r2}$ .....	62
Figura 111. Modelo 5: Diagrama radiación 3D $f_{r3}$ .....	62
Figura 112. Modelo 5: Directividad .....	63
Figura 113. Modelo 5: Eficiencia de Radiación .....	63
Figura 114. Modelo 5: Eficiencia Total .....	63
Figura 115. Modelo 5: Ganancia.....	64
Figura 116. Modelo 5: Ganancia Realizable .....	64
Figura 117. Modelo 6: Componente.....	65
Figura 118. Modelo 6: $S_{11}$ .....	65
Figura 119. Modelo 6: Diagrama de radiación 3D $f_{r1}$ .....	66
Figura 120. Modelo 6: Diagrama de radiación 3D $f_{r2}$ .....	66
Figura 121. Modelo 6: Directividad .....	66
Figura 122. Modelo 6: Eficiencia de Radiación .....	67
Figura 123. Modelo 6: Eficiencia Total .....	67
Figura 124. Modelo 6: Ganancia.....	68
Figura 125. Modelo 6: Ganancia Realizable .....	68
Figura 126. Modelo 7: Componente.....	69
Figura 127. Modelo 7: $S_{11}$ .....	69
Figura 128. Modelo 7: Diagrama radiación 3D $f_{r1}$ .....	70
Figura 129. Modelo 7: Diagrama radiación 3D $f_{r2}$ .....	70
Figura 130. Modelo 7: Diagrama radiación 3D $f_{r3}$ .....	70
Figura 131. Modelo 7: Directividad .....	71
Figura 132. Modelo 7: Eficiencia de Radiación .....	71
Figura 133. Modelo 7: Eficiencia Total .....	71
Figura 134. Modelo 7: Ganancia.....	72
Figura 135. Modelo 7: Ganancia Realizable .....	72
Figura 136. Modelo 8: Componente.....	73
Figura 137. Modelo 8: $S_{11}$ .....	73
Figura 138. Modelo 8: Diagrama radiación 3D $f_{r1}$ .....	74
Figura 139. Modelo 8: Diagrama radiación 3D $f_{r2}$ .....	74
Figura 140. Modelo 8: Directividad .....	74
Figura 141. Modelo 8: Eficiencia de Radiación .....	75
Figura 142. Modelo 8: Eficiencia Total .....	75
Figura 143. Modelo 8: Ganancia.....	76
Figura 144. Modelo 8: Ganancia Realizable .....	76
Figura 145. Modelo 9: Componente.....	77
Figura 146. Modelo 9: $S_{11}$ .....	77
Figura 147. Modelo 9: Diagrama radiación 3D $f_{r1}$ .....	78
Figura 148. Modelo 9: Diagrama radiación 3D $f_{r2}$ .....	78
Figura 149. Modelo 9: Directividad .....	78
Figura 150. Modelo 9: Eficiencia de Radiación .....	79
Figura 151. Modelo 9: Eficiencia Total .....	79
Figura 152. Modelo 9: Ganancia.....	80
Figura 153. Modelo 9: Ganancia Realizable .....	80
Figura 154. Modelo 10: Componente.....	81
Figura 155. Modelo 10: $S_{11}$ .....	81
Figura 156. Modelo 10: Diagrama radiación 3D $f_{r1}$ .....	82
Figura 157. Modelo 10: Diagrama radiación 3D $f_{r2}$ .....	82
Figura 158. Modelo 10: Directividad .....	82
Figura 159. Modelo 10: Eficiencia de Radiación .....	83
Figura 160. Modelo 10: Eficiencia Total .....	83
Figura 161. Modelo 10: Ganancia.....	84
Figura 162. Modelo 10: Ganancia Realizable .....	84
Figura 163. Modelo 11: Componente.....	85
Figura 164. Modelo 11: $S_{11}$ .....	85

Figura 165. Modelo 11: Diagrama radiación 3D $f_{r1}$ .....	86
Figura 166. Modelo 11: Diagrama radiación 3D $f_{r2}$ .....	86
Figura 167. Modelo 11: Directividad .....	86
Figura 168. Modelo 11: Eficiencia de Radiación.....	87
Figura 169. Modelo 11: Eficiencia Total .....	87
Figura 170. Modelo 11: Ganancia.....	88
Figura 171. Modelo 11: Ganancia Realizable .....	88
Figura 172. Modelo 12: Componente.....	89
Figura 173. Modelo 12: $S_{11}$ .....	89
Figura 174. Modelo 12: Diagrama radiación 3D $f_{r1}$ .....	90
Figura 175. Modelo 12: Diagrama radiación 3D $f_{r2}$ .....	90
Figura 176. Modelo 12: Directividad .....	91
Figura 177. Modelo 12: Ganancia.....	91
Figura 178. Modelo 12: Ganancia Realizable .....	92
Figura 179. Modelo 13: Componente.....	93
Figura 180. Modelo 13: $S_{11}$ .....	93
Figura 181. Modelo 13: Diagrama radiación 3D $f_{r1}$ .....	94
Figura 182. Modelo 13: Diagrama radiación 3D $f_{r2}$ .....	94
Figura 183. Modelo 13: Ganancia.....	95
Figura 184. Modelo 13: Ganancia Realizable .....	95
Figura 185. Modelo 14: Componente.....	96
Figura 186. Modelo 14: $S_{11}$ .....	96
Figura 187. Modelo 14: Diagrama radiación 3D $f_{r1}$ .....	97
Figura 188. Modelo 14: Diagrama radiación 3D $f_{r2}$ .....	97
Figura 189. Mejor modelo orden 1: Componente.....	100
Figura 190. Mejor modelo orden 1: $S_{11}$ .....	100
Figura 191. Mejor modelo orden 1: <i>Farfield</i> $f_{r1}$ .....	100
Figura 192. Mejor modelo orden 1: <i>Farfield</i> $f_{r2}$ .....	101
Figura 193. Mejor modelo orden 1: <i>Farfield</i> $f_{r3}$ .....	101
Figura 194. Mejor modelo orden 2: Componente.....	102
Figura 195. Mejor modelo orden 2: $S_{11}$ .....	102
Figura 196. Mejor modelo orden 2: <i>Farfield</i> $f_{r1}$ .....	103
Figura 197. Mejor modelo orden 2: <i>Farfield</i> $f_{r2}$ .....	103
Figura 198. Modelo fractal orden 5.....	105

# Lista de tablas

Tabla 1. Modelo 1: Coeficiente de Reflexión $S_{11}$	44
Tabla 2. Modelo 1: Directividad	46
Tabla 3. Modelo 1: Eficiencias de Radiación	46
Tabla 4. Modelo 1: Ganancia	47
Tabla 5. Modelo 1: Ganancia Realizable	47
Tabla 6. Modelo 1: Potencia Total Radiada	48
Tabla 7. Modelo 2: Coeficiente de Reflexión $S_{11}$	49
Tabla 8. Modelo 2: Directividad	50
Tabla 9. Modelo 2: Eficiencias de Radiación	51
Tabla 10. Modelo 2: Ganancia	51
Tabla 11. Modelo 2: Ganancia Realizable	52
Tabla 12. Modelo 2: Potencia Total Radiada	52
Tabla 13. Modelo 3: Coeficiente de Reflexión $S_{11}$	53
Tabla 14. Modelo 3: Directividad	54
Tabla 15. Modelo 3: Eficiencias de Radiación	55
Tabla 16. Modelo 3: Ganancia	56
Tabla 17. Modelo 3: Ganancia Realizable	56
Tabla 18. Modelo 3: Potencia Total Radiada	56
Tabla 19. Modelo 4: Coeficiente de Reflexión $S_{11}$	57
Tabla 20. Modelo 4: Directividad	58
Tabla 21. Modelo 4: Eficiencias de Radiación	59
Tabla 22. Modelo 4: Ganancia	60
Tabla 23. Modelo 4: Ganancia Realizable	60
Tabla 24. Modelo 4: Potencia Total Radiada	60
Tabla 25. Modelo 5: Coeficiente de Reflexión $S_{11}$	61
Tabla 26. Modelo 5: Directividad	62
Tabla 27. Modelo 5: Eficiencias de Radiación	63
Tabla 28. Modelo 5: Ganancia	64
Tabla 29. Modelo 5: Ganancia Realizable	64
Tabla 30. Modelo 5: Potencia Total Radiada	64
Tabla 31. Modelo 6: Coeficiente de Reflexión $S_{11}$	65
Tabla 32. Modelo 6: Directividad	66
Tabla 33. Modelo 6: Eficiencias de Radiación	67
Tabla 34. Modelo 6: Ganancia	67
Tabla 35. Modelo 6: Ganancia Realizable	68
Tabla 36. Modelo 6: Potencia Total Radiada	68
Tabla 37. Modelo 7: Coeficiente de Reflexión $S_{11}$	69
Tabla 38. Modelo 7: Directividad	71
Tabla 39. Modelo 7: Eficiencias de Radiación	71
Tabla 40. Modelo 7: Ganancia	72
Tabla 41. Modelo 7: Ganancia Realizable	72
Tabla 42. Modelo 7: Potencia Total Radiada	72
Tabla 43. Modelo 8: Coeficiente de Reflexión $S_{11}$	73
Tabla 44. Modelo 8: Directividad	74
Tabla 45. Modelo 8: Eficiencias de Radiación	75
Tabla 46. Modelo 8: Ganancia	75
Tabla 47. Modelo 8: Ganancia Realizable	76
Tabla 48. Modelo 8: Potencia Total Radiada	76
Tabla 49. Modelo 9: Coeficiente de Reflexión $S_{11}$	77
Tabla 50. Modelo 9: Directividad	78
Tabla 51. Modelo 9: Eficiencias de Radiación	79

Tabla 52. Modelo 9: Ganancia	79
Tabla 53. Modelo 9: Ganancia Realizable	80
Tabla 54. Modelo 9: Potencia Total Radiada	80
Tabla 55. Modelo 10: Coeficiente de Reflexión $S_{11}$	81
Tabla 56. Modelo 10: Directividad	82
Tabla 57. Modelo 10: Eficiencias de Radiación	83
Tabla 58. Modelo 10: Ganancia	83
Tabla 59. Modelo 10: Ganancia Realizable	84
Tabla 60. Modelo 10: Potencia Total Radiada	84
Tabla 61. Modelo 11: Coeficiente de Reflexión $S_{11}$	85
Tabla 62. Modelo 11: Directividad	86
Tabla 63. Modelo 11: Eficiencias de Radiación	87
Tabla 64. Modelo 11: Ganancia	87
Tabla 65. Modelo 11: Ganancia Realizable	88
Tabla 66. Modelo 11: Potencia Total Radiada	88
Tabla 67. Modelo 12: Coeficiente de Reflexión $S_{11}$	89
Tabla 68. Modelo 12: Directividad	90
Tabla 69. Modelo 12: Eficiencias de Radiación	91
Tabla 70. Modelo 12: Ganancia	91
Tabla 71. Modelo 12: Ganancia Realizable	92
Tabla 72. Modelo 12: Potencia Total Radiada	92
Tabla 73. Modelo 13: Coeficiente de Reflexión $S_{11}$	93
Tabla 74. Modelo 13: Directividad	94
Tabla 75. Modelo 13: Eficiencias de Radiación	94
Tabla 76. Modelo 13: Ganancia	95
Tabla 77. Modelo 13: Ganancia Realizable	95
Tabla 78. Modelo 13: Potencia Total Radiada	95
Tabla 79. Modelo 14: Coeficiente de Reflexión $S_{11}$	96
Tabla 80. Modelo 14: Directividad	97
Tabla 81. Modelo 14: Eficiencias de Radiación	98
Tabla 82. Modelo 14: Ganancia	98
Tabla 83. Modelo 14: Ganancia Realizable	98
Tabla 84. Modelo 14: Potencia Total Radiada	98
Tabla 85. Resumen comparativo modelos	99
Tabla 86. Mejor modelo orden 1: Parámetros	100
Tabla 87. Mejor modelo orden 2: Parámetros	102

# 1. Introducción

Benoît B. Mandelbrot define fractal como:

*“Que tiene una forma, bien sea sumamente irregular, sumamente interrumpida o fragmentada y sigue siendo así a cualquier escala que se produzca el examen” [1]*

## 1.1 Contexto y justificación del Trabajo

A través de este proyecto se pretende realizar el análisis y simulación de una nano antena implementada con un diseño fractal y con grafeno como material.

Las antenas fractales tienen unas características que las hacen muy interesantes como son: proporcionar un gran ancho de banda y tener comportamiento multibanda, dar una alta ganancia con poca dependencia de la frecuencia en un rango de frecuencias grande, proporcionar un patrón de radiación estable para también un amplio rango de frecuencias. Además con los diseños fractales es posible conseguir antenas con una alta directividad. [2]

Este tipo de antenas tienen una muy buena respuesta en frecuencia, completamente diferente a las antenas tradicionales, ya que son capaces de ofrecer excelentes ganancias en diferentes frecuencias de forma simultánea. También utilizan un diseño fractal para maximizar la distancia o el perímetro que puede recibir o transmitir. Por estos dos motivos, actualmente se están utilizando en diversas aplicaciones como son: sistemas móviles celulares, dispositivos de microondas, dispositivos IoT y otras aplicaciones aeronáuticas o militares. [3]

A su vez, el grafeno es un nanomaterial que presenta una conductividad eléctrica muy alta y aplicado a una antena de dimensiones de nanómetros permite la propagación de las ondas electromagnéticas en las frecuencias de terahercios. [14]

Por todo ello, resulta muy interesante la realización del estudio teórico y posterior simulación de una antena microstrip con geometría fractal realizada con grafeno como material del parche conductor.

En estos momentos existen empresas como Fractal Antenna System Inc o Fractus SA, dedicadas al desarrollo de antenas fractales para múltiples aplicaciones de telecomunicaciones pero que no han incluido el grafeno entre sus materiales de fabricación. La razón es que aunque es un material con características de superconductor, todavía se encuentra en fase experimental. También existen diversos estudios orientados al análisis y diseño de nuevos dispositivos capaces de operar directamente en el rango de terahercios y que pueden satisfacer las necesidades de velocidades de datos más altas en redes cada día más densas y con una tendencia futura creciente. [23] [24]

En el presente proyecto se pretende ampliar los estudios realizados hasta el momento y llegar a simular el comportamiento de una antena microstrip con topología fractal basada en el triángulo de Sierpiński con material del parche de grafeno para varias iteraciones y con distintos sustratos.



## 1.2 Objetivos del Trabajo

Los principales objetivos que se estiman alcanzar durante el proyecto son:

1. Adquirir los conocimientos necesarios sobre el diseño de antenas fractales.
2. Obtener los fundamentos teóricos sobre el nanomaterial grafeno.
3. Estudiar detenidamente las prestaciones del grafeno y su uso aplicado a las comunicaciones inalámbricas.
4. Desarrollar el código de Matlab necesario para comunicarse con CST Studio Suite SE para facilitar el diseño de una antena fractal basada en el triángulo de Sierpiński.
5. Analizar de forma teórica las prestaciones de una antena fractal basada en el triángulo de Sierpiński de grafeno sobre dos tipos sustrato: dióxido de silicio y fibra de vidrio.
6. Realizar la simulación mediante el software CST Studio Suite SE y validación de los resultados obtenidos de forma teórica.

## 1.3 Enfoque y método seguido

En el presente trabajo se pretende realizar el diseño y simulación de una antena multibanda basada en la forma fractal triángulo de Sierpiński. Para poder llevarlo a cabo, en primer lugar se ha definido el tipo de antena fractal a utilizar. En este caso se tratará de una antena microstrip con topología fractal triángulo de Sierpiński con la que se procederá al estudio de sus principales parámetros analizados de forma teórica y verificados mediante el software de simulación CST Studio Suite SE.

En segundo lugar, se realizará el análisis para distintas iteraciones de la misma antena fractal con material grafeno y sobre dos tipos distintos de sustrato: uno de dióxido de silicio y el otro sustrato de resina de vidrio. De este modo, se pretende llevar a término la comparación de los parámetros de los distintos modelos obtenidos y así poder seleccionar el modelo que mejores prestaciones proporcione.

Para diseñar los modelos de forma rápida se implementará una interfaz en Matlab que permita la comunicación con CST Studio Suite SE, de modo que mediante instrucciones de código Matlab se obtenga el proyecto CST con el modelo a simular. Una parte de la simulación se realizará de forma automática desde la propia API entre Matlab y CST; y otra parte de la simulaciones se realizarán con las herramientas de post-procesado de CST.

En ambos casos se ha seleccionado software con licencia de estudiante que no conlleve un coste adicional para el trabajo.

Por último, indicar que con el estudio realizado a lo largo del proyecto se pretende obtener diseños de antenas funcionales en la banda de los terahercios que permitan trabajar en múltiples bandas de frecuencia y que aporten ventajas sobre las actuales antenas fractales que tenemos en el mercado.

## 1.4 Planificación del Trabajo

Los recursos que se han utilizado para realizar el presente proyecto son los *papers* de investigación más recientes en el campo del diseño de antenas fractales, antenas de grafeno y antenas fractales de grafeno. De este último ámbito, existen pocos trabajos realizados pese a tratarse de un tema muy interesante.

Como herramientas de diseño y análisis se han utilizado los programas: **Matlab R2022a** (9.12.0.1884302) y **CST Studio Suite 2021 (Student Edition)** ambos con licencias de estudiante.

El detalle de cada una de las tareas es el siguiente:

### PEC1 Planificación e inicio del trabajo

Fecha inicio: 31/01/2022

Fecha fin: 22/02/2022

Tareas realizadas:

1. Propuesta TFG
2. Reunión inicial
3. Revisión *papers* fractales
4. Revisión *papers* fractales de grafeno
5. Presentación Estado de la cuestión
6. Reunión revisión Estado de la cuestión

### PEC2 Seguimiento del trabajo I:

Fecha inicio: 23/02/2022

Fecha fin: 15/03/2022

Tareas realizadas:

1. Aprendizaje Matlab y Antenna Toolbox Matlab
2. Aprendizaje CST Studio Suite 2021 SE
3. Diseño y desarrollo API Matlab
4. Creación y simulación modelos
5. Reunión de seguimiento

### PEC3 Seguimiento del trabajo II:

Fecha inicio: 15/03/2022

Fecha fin: 12/04/2022

Tareas realizadas:

1. Creación y simulación modelos grafeno
2. Mejoras API Matlab
3. Desarrollo nuevas funciones API Matlab
4. Reunión de seguimiento

**PEC4 Entrega preliminar del trabajo:**

Fecha inicio: 12/04/2022

Fecha fin: 15/05/2022

Tareas realizadas:

1. Comparación modelos
2. Presentación modelos
3. Reunión de seguimiento

**PEC5 Entrega final del trabajo:**

Fecha inicio: 15/05/2022

Fecha fin: 13/06/2022

Tareas realizadas:

1. Redacción presentación
2. Redacción memoria
3. Reunión final

En el diagrama de Gantt se muestra la planificación temporal de cada una de las tareas realizadas, con el detalle de los días y las horas de trabajo invertidas.

Análisis y simulación antena de grafeno basada en el triángulo fractal de Sierpiński

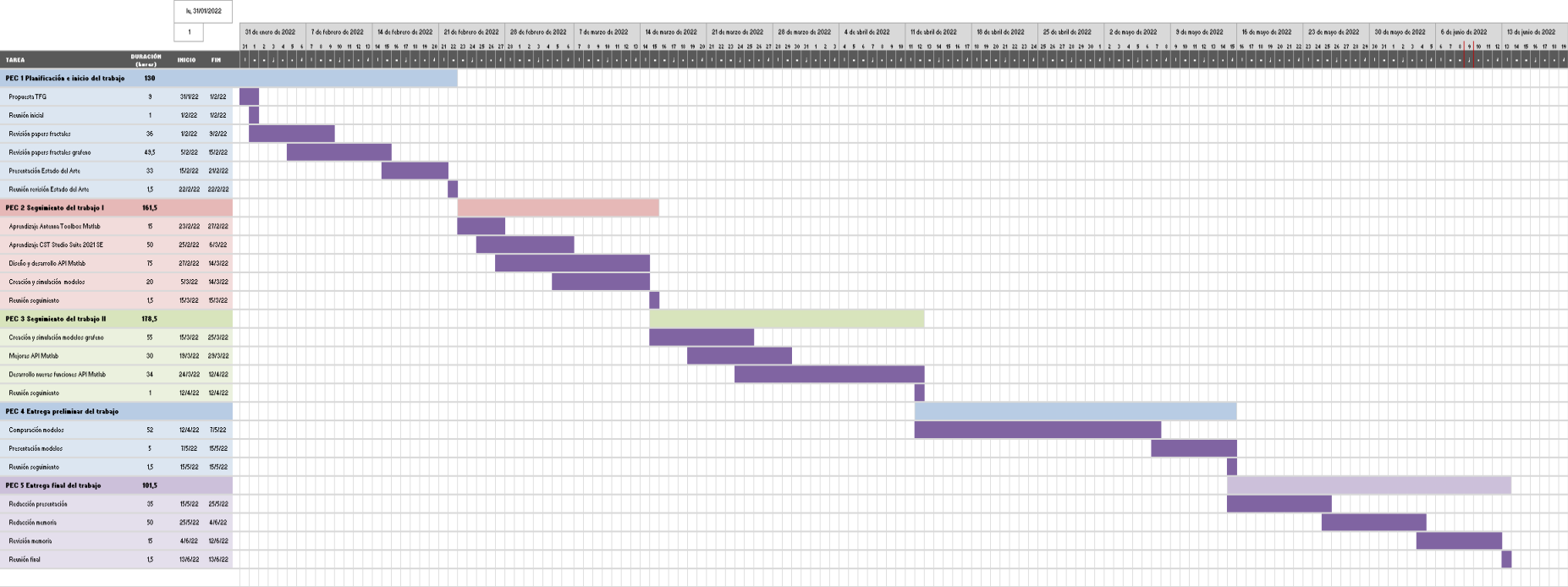


Figura 1. Diagrama de Gantt con la planificación.

## **1.5 Breve resumen de productos obtenidos**

En el trabajo se han programado las funciones de Matlab necesarias para automatizar una parte del diseño y simulación de la antena fractal de modo que se facilite el diseño en CST de los distintos modelos de la antena fractal en estudio de forma rápida y eficiente.

Adicionalmente se han obtenido en CST los modelos simulados y posteriormente se ha realizado una parte de post-procesado para completar el análisis de los diseños.

Finalmente se han conseguido dos antenas funcionales según los parámetros obtenidos en las distintas simulaciones.

## **1.6 Breve descripción de los otros capítulos de la memoria**

La memoria está dividida en cinco capítulos. En primer lugar se definen los conceptos básicos sobre fractales, antenas fractales y grafeno. En el mismo capítulo se realiza un recorrido por los distintos artículos de investigación que se centran en los temas indicados de modo que se realiza un completo estado de la cuestión.

En el siguiente capítulo se detalla el enfoque de la solución de la propuesta de trabajo. Se especifican los métodos utilizados para implementar los diseños.

En el tercer capítulo se incluyen los resultados de las simulaciones de todos los modelos generados y una tabla resumen de los mismos. Los últimos apartados de este capítulo son para remarcar los dos diseños que se consideran mejores de entre todos los obtenidos.

En el penúltimo capítulo se muestran las conclusiones obtenidas y en el último capítulo se proporcionan los planteamientos de futuras líneas de trabajo.

## 2. Estado de la cuestión

En este capítulo se realiza una rápida revisión del concepto de antena fractal y se comentan algunos tipos de antenas fractales que resultan más interesantes. Posteriormente se realiza un repaso por los distintos artículos de investigación publicados hasta el momento sobre el diseño de antenas fractales.

A continuación se profundiza en los fundamentos físicos del grafeno para continuar con una ampliación de información con referencias a las actuales líneas de trabajo en el diseño de antenas de grafeno en la región inferior de las comunicaciones, en las frecuencias de terahercios (THz).

Por último, se estudian los trabajos efectuados en el diseño de antenas fractales de grafeno en la banda de los terahercios y se mencionan los dos trabajos más relevantes.

### 2.1 Antenas fractales

#### 2.1.1 Definición de fractal

El matemático Benoît Mandelbrot fue el responsable de desarrollar, en 1975, el concepto de fractal, que proviene del vocablo latín *fractus* y que podemos traducir como quebrado o fracturado.

Según Mandelbrot un fractal se puede definir *“Que tiene una forma, bien sea sumamente irregular, sumamente interrumpida o fragmentada y sigue siendo así a cualquier escala que se produzca el examen”* [1]

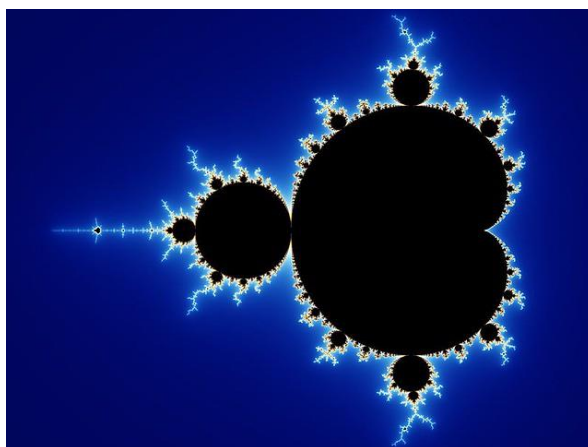


Figura 2. Conjunto de Mandelbrot

Por lo tanto, un fractal es un objeto semi geométrico en el que se repite el mismo patrón a diferentes escalas y con distinta orientación.

Existen numerosos tipos de conjuntos fractales: los conjuntos de Cantor, curvas de Koch, Peano, Hilbert, triángulos y alfombras de Sierpiński. A continuación se detallan algunos ejemplos.

### 2.1.1.1 Conjuntos de Cantor

El conjunto de Cantor está compuesto por subconjuntos de puntos dentro del intervalo  $[0,1]$ , construido de forma recursiva de modo que se parte del intervalo completo y se le quita su tercio interior. Se repite este proceso con los segmentos restantes infinitas veces:

$$C = \bigcap_{n=0}^{\infty} \bigcup_{i=1}^{2^n} I_n^i$$

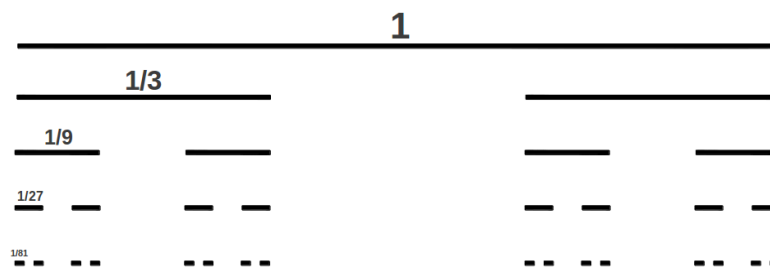


Figura 3. Conjunto de Cantor

### 2.1.1.2 Curvas de Koch

La curva de Koch se genera mediante un proceso recursivo en el que se divide un segmento unidad en tres partes y se sustituye el segmento central por dos segmentos de longitud  $1/3$  formando un ángulo de  $60^\circ$

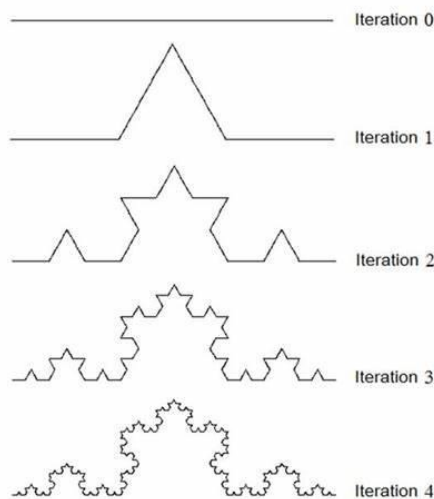


Figura 4. Curva de Koch

### 2.1.1.3 Curva de Peano

La curva de Peano es una curva continua que recubre todo el plano.

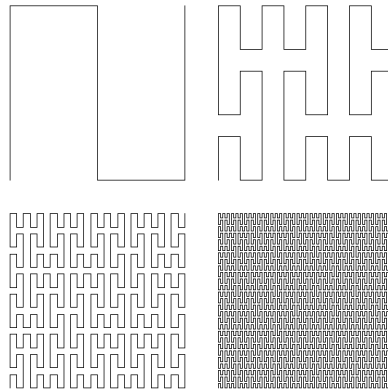


Figura 5. Curva de Peano

### 2.1.1.4 Curva de Hilbert

La curva de Hilbert es una curva fractal continua que recubre todo el plano y que deriva de la curva de Peano.

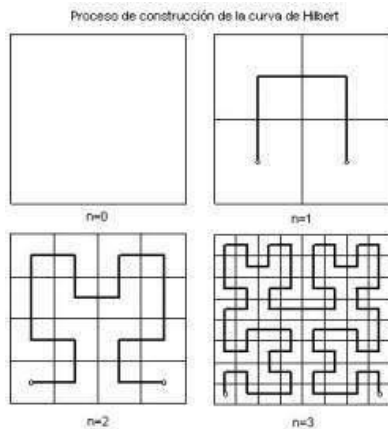


Figura 6. Curva de Hilbert

### 2.1.1.5 Triángulos de Sierpiński

El triángulo de Sierpiński se construye a partir de un triángulo equilátero al que se le resta el triángulo resultante de unir los puntos medios de sus lados y se repite este proceso de forma iterativa:



Figura 7. Triángulo de Sierpiński



### 2.1.2 Tipos de antenas fractales

Las antenas fractales están inspiradas en los diseños matemáticos definidos por los expertos de este campo como son: Benoît Mandelbrot, considerado el padre de la geometría fractal; W. Sierpiński, N. F. H. von Koch, D. Hilbert, G. F. L. P. Cantor, y G. Peano.

Los ingenieros comenzaron a diseñar antenas basadas en patrones fractales al descubrir de forma empírica que mejoraban considerablemente las prestaciones de sus diseños. Comprobaron que se obtenían antenas de dimensiones menores, multibanda y con una alta directividad. A continuación se describen brevemente los más notables. [2] [3]

#### 2.1.2.1 Mandelbrot

La antena de parche microstrip basada en el fractal de Mandelbrot se obtiene mediante una modificación de la ecuación de Mandelbrot. Resulta muy interesante por presentar un perfil rectangular y proporcionar una alta directividad a la frecuencia de 2,57 GHz, en concreto  $D = 11,5$  dB.

La principal ventaja que proporciona es conseguir una directividad muy alta con un solo elemento, de hecho se necesitarían varios parches microstrip que operasen en el modo fundamental para proporcionar la misma directividad. [3]

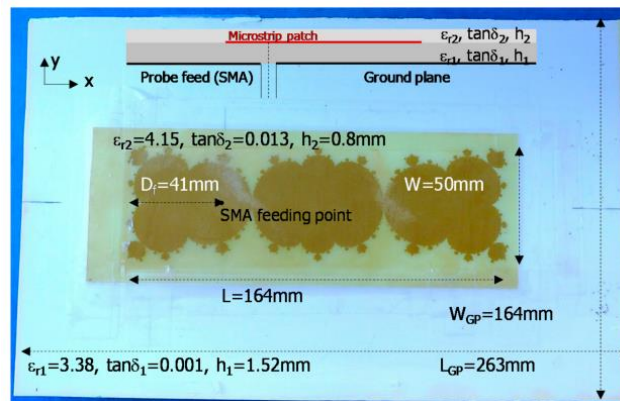


Figura 8. Antena parche microstrip basada en el fractal de Mandelbrot

#### 2.1.2.2 Sierpiński

La forma fractal triángulo de Sierpiński es una de las más utilizadas en el diseño de antenas fractales. Gracias a su autosimilitud resulta muy útil para construir antenas multibanda ya que los diseños consiguen patrones de radiación iguales o muy parecidos para distintas frecuencias de funcionamiento.

Al igual que las antenas de parche microstrip con forma fractal de Mandelbrot, este tipo de geometrías han demostrado ser atractivas para diseñar antenas que soportan modos de orden superior con alta directividad. [3]

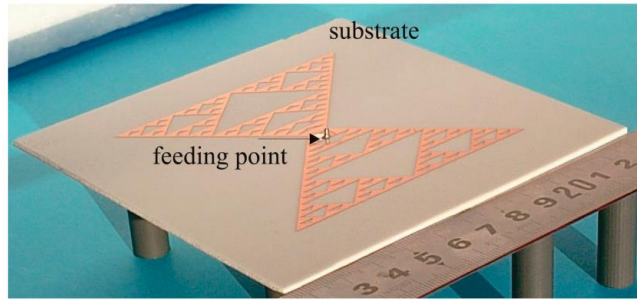


Figura 9. Antena de parche microstrip basada en Sierpiński alimentada con sonda coaxial

### 2.1.2.3 Koch

El diseño fractal de Koch aplicado a una antena de parche microstrip también proporciona alta directividad en el modo de orden superior a la frecuencia de 3,75GHz con una directividad de  $D= 13,5$  dB y con un patrón de radiación lateral estable. [3]

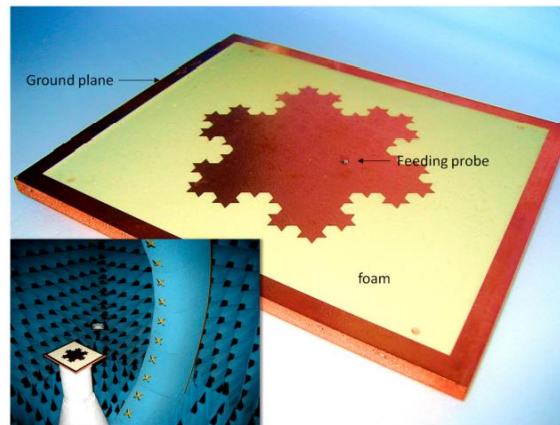


Figura 10. Antena parche microstrip basada en la isla de Koch

### 2.1.2.4 Hilbert

El fractal de Hilbert resulta muy útil en el diseño de antenas pequeñas puesto que una antena monopolo siguiendo un patrón de Hilbert muestra una reducción de la frecuencia de resonancia conforme aumenta el número de iteraciones. Este tipo de antenas resultan eficaces para las bandas de WiFi de 2,5 GHz y 5 GHz por lo que es una antena que se puede integrar en pequeños dispositivos inalámbricos *wereables*. [3]

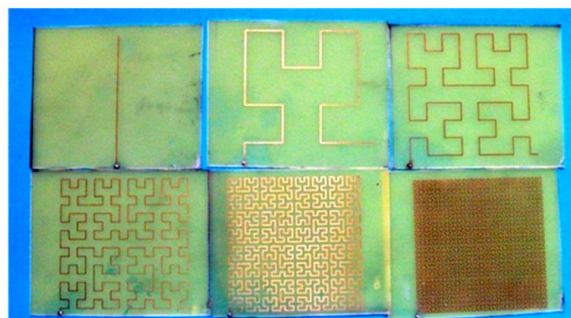


Figura 11. Monopolo de Hilbert para varias iteraciones

### 2.1.2.5 Cantor

Un diseño de antena basada en el conjunto de Cantor es el mostrado a continuación que se ha utilizado en un sistema de captación de energía RF. Las características autorepetitivas y autocomplementarias de este tipo de geometría permiten utilizarse en las tres bandas de frecuencia de GSM. [3]

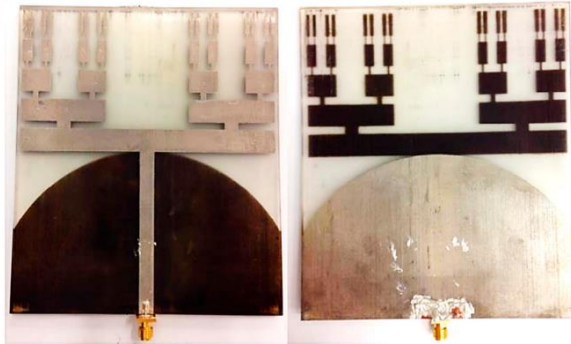


Figura 12. Antena basada en la geometría de Cantor

### 2.1.2.6 Peano

La curva de Peano permite diseñar antenas pequeñas, elementos de alta directividad, matrices y superficies selectivas de frecuencia. [3]

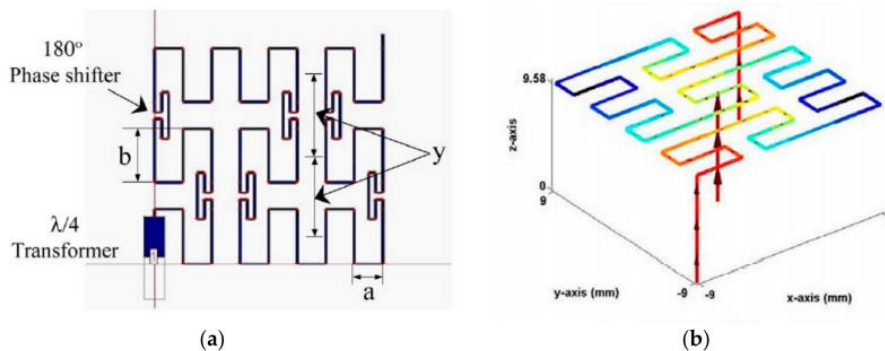


Figura 13. Antena hilo de alta directividad y monopolo

## 2.2 Las propiedades del grafeno

### 2.2.1 Definición

El grafeno es un material de carbono bidimensional que tiene los átomos organizados en base a un patrón regular hexagonal y que presenta unas propiedades eléctricas muy interesantes.

### 2.2.2 Propiedades

Este nanomaterial tiene una conductividad eléctrica muy alta y cuando se utiliza en una antena de nanómetros permite la propagación de las ondas electromagnéticas en las frecuencias de los terahercios (THz). [14]

La propagación de las ondas de polaritones de plasmón superficial que se producen en la capa de grafeno SPP permite la resonancia en la frecuencia de terahercios con una mayor eficiencia de radiación que el metal. [16].

Como breve explicación de este fenómeno físico se tiene que los polaritones de plasmón de superficie (SPP) son ondas electromagnéticas que se desplazan entre un metal-dieléctrico o entre metal-aire en la región del espectro electromagnético del infrarrojo o visible. Son ondas que implican movimiento de carga en el metal (plasmón de superficie) y ondas electromagnéticas en el aire o dieléctricas (polaritón).

Como notas aclaratorias de estos conceptos se indica a continuación la definición de polaritón y la definición de plasmón.

**Polaritón:** cuasipartícula resultado del fuerte acoplamiento de ondas electromagnéticas con la onda de polarización eléctrica o magnética

**Plasmón:** cuasipartícula que describe al campo EM producido por las oscilaciones colectivas de los electrones de conducción en un material. Los plasmones de superficie son aquellos que están confinados a las superficies y que forman un polaritón cuando interactúan con la luz.

### 2.2.3 Fabricación

Se puede obtener grafeno de área grande mediante deposición de vapor químico sobre cobre y posteriormente transferirlo a una oblea de Si/SiO<sub>2</sub>. Con esta técnica se puede utilizar para cultivar grafeno en películas de cobre de 300 mm sobre sustratos de Si. [15]

Existen muchas líneas de investigación de técnicas de obtención de grafeno en grandes cantidades pero exceden el objetivo de este trabajo.

## **2.3 Usos comerciales antenas grafeno THz**

Los usos comerciales más destacables para las antenas de grafeno en las frecuencias de los terahercios se han obtenido de los *papers* de investigación que se han utilizado en la presente memoria como referencia y son los siguientes:

### **2.3.1. Antena terahercios sintonizable**

Posibilidad de ser utilizada como una antena de terahercios sintonizable. Es posible cambiar la frecuencia de resonancia mediante el cambio de material o tamaño del sustrato o mediante la aplicación de un sesgo electrostático externo. [16]

### **2.3.2. Aplicaciones aeroespaciales**

La antena diseñada en [17] tiene una frecuencia de resonancia de 0.7 THz y se podría utilizar en aplicaciones aeroespaciales.

### **2.3.3. Antena reconfigurable**

Antenas sintonizables en terahercios. En [18] se demuestra que es posible realizar una antena sintonizable cambiando el potencial químico del grafeno.

### **2.3.4. Aplicaciones captación imagen**

La antena anular de grafeno diseñada en [19] se puede utilizar para aplicaciones médicas de captación de imagen.

### **2.3.5. Antenas fractales grafeno**

Se pueden utilizar como antenas reconfigurables para distintos órdenes con mayor ganancia que las antenas convencionales.

### **2.3.6. Estudios con tapiz y triángulo Sierpiński**

En ambos trabajos se comprueba la posibilidad de utilizarse estas configuraciones con grafeno para diseñar antenas reconfigurables. [23][24]

### **2.3.7. Aplicaciones de imagen médica**

La banda inferior de los THz, hasta los 0.310 THz, se pueden utilizar en aplicaciones de imagen médica. La radiación procedente de los terahercios no resulta perjudicial para los tejidos vivos puesto que solo es capaz de penetrar unos milímetros en la piel. [20]

## 2.4 Las antenas de grafeno

En el ámbito de estudio de las antenas de grafeno se localizan numerosos trabajos de investigación, de los que cabe especial mención los siguientes:

### 2.4.1 Trabajo: "Graphene-based nano-patch antenna for terahertz radiation"

En el trabajo "Antena Nano Parche Basada en Grafeno para la Radiación de Terahercios" [16] se realiza un análisis numérico de la dispersión de la radiación de terahercios en una nanoantena basada en grafeno.

En la Figura 14 se puede observar el esquema de la nanoantena considerada en el estudio:

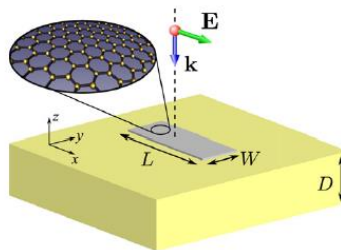


Figura 14. Esquema nanoantena grafeno

La antena diseñada de longitud  $L$  y anchura  $W$ , soportada por un sustrato dieléctrico de espesor  $D$  se excita por una onda plana polarizada linealmente a lo largo de la longitud del parche. La dispersión asociada se resuelve de forma numérica.

En la Figura 15 se puede observar la parte real e imaginaria de la conductividad total (líneas sólidas) y la conductividad intrabanda (líneas discontinuas) a temperatura ambiente ( $T = 300\text{K}$ ) y sesgo electrostático cero ( $\mu_c = 0\text{ eV}$ ). Podemos ver que la conductividad intrabanda es dominante en la región de interés (por debajo de los 5 THz), por esta razón la conductividad interbanda se desprecia y se presupone que la conductividad del parche de grafeno es puramente intrabanda.

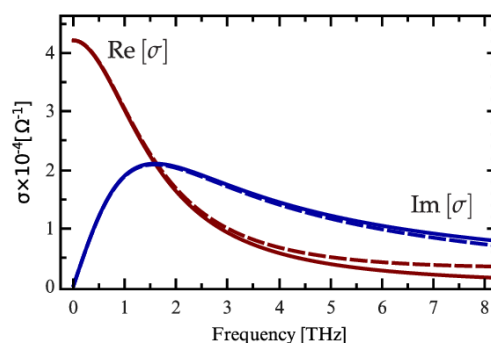


Figura 15. Conductividad total vs conductividad intrabanda

En el trabajo se muestran los métodos numéricos para encontrar la dispersión del campo electromagnético por una estructura de grafeno y acoplarlo con las ecuaciones de Maxwell, para ello se modela la capa de grafeno infinitesimal utilizando

una discretización del espacio de tamaño finito. El método elegido es modelar la lámina de grafeno como una superficie de impedancia equivalente.

En la Figura 16 se muestra la dependencia de la primera frecuencia de resonancia de la antena de grafeno para diferentes anchos del parche de grafeno ( $W$ ).

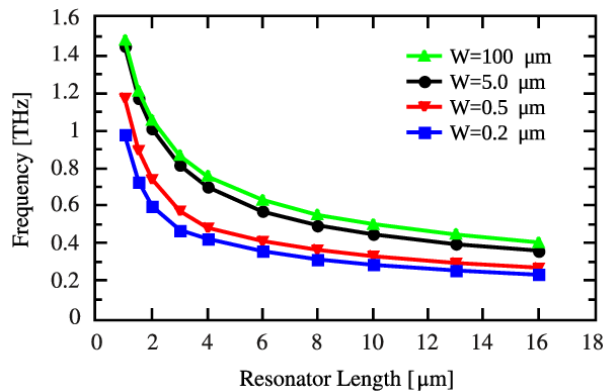


Figura 16. Frecuencia vs ancho ( $W$ )

En la Figura 17 se observa la dependencia de la frecuencia de resonancia de la antena con la polarización electrostática y con la temperatura.

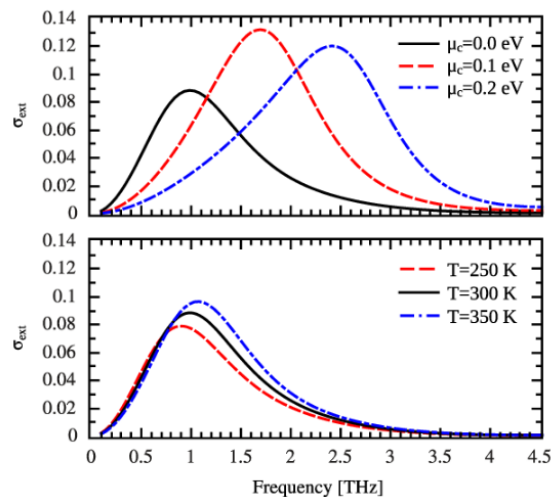


Figura 17. Frecuencia resonancia vs polarización electrostática o temperatura

En el trabajo se comprueba que la antena presenta un comportamiento resonante en la banda de los terahercios. El fenómeno que posibilita este hecho es la propagación de las ondas de polaritones de plasmón superficial (SPP) que se producen en la capa de grafeno. Además las simulaciones indican que se puede utilizar como una antena de terahercios sintonizable mediante el cambio del material, tamaño del sustrato o mediante la aplicación de un sesgo electrostático externo que modifique la polarización electrostática ( $\mu_c$ ). [16]

## 2.4.2 Trabajo: "Terahertz Graphene-Based Reconfigurable Patch Antenna"

En el trabajo "Antena de Parche Reconfigurable basada en Grafeno para Terahercios" [17] se presentan las propiedades de radiación de una antena de parche en la que el parche está hecho de cobre o grafeno y se comparan ambas. Se demuestra que el grafeno como material sintonizable permite modificar dinámicamente la frecuencia de funcionamiento de la antena al igual que su patrón de radiación.

El pico del valor de las pérdidas de retorno con la antena de cobre es de -29 dB a la frecuencia de funcionamiento 0,7 THz, lo que supone casi el doble del valor obtenido con el parche de cobre. Adicionalmente la ganancia de la antena pasa de ser de 5,73 dB para el cobre a 7,16 dB para el grafeno. El trabajo se centra en analizar las propiedades de radiación reconfigurables de la antena de parche con grafeno que están directamente relacionadas con la conductividad de la superficie del grafeno.

En la Figura 18 se puede ver la antena en estudio:

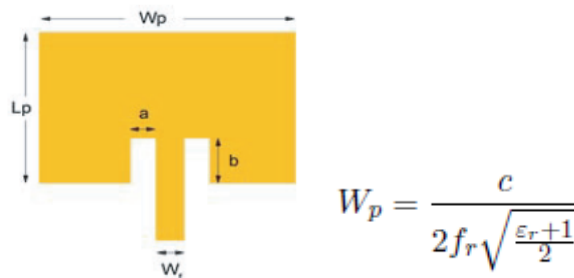


Figura 18. Esquema antena parche microstrip

En la Figura 19 se presentan las pérdidas de retorno para la antena de parche de cobre microstrip, se observa que la frecuencia de resonancia obtenida es 0,68 THz con una pérdida de retorno de tan sólo 15 dB.

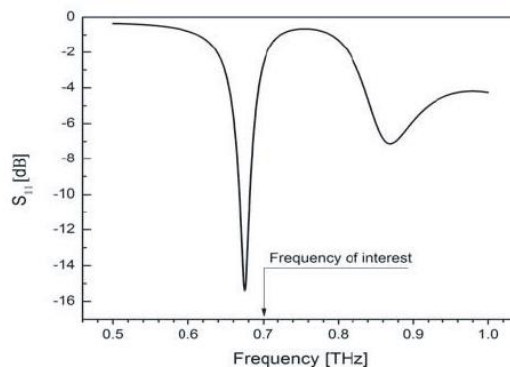


Figura 19. Pérdidas de retorno antena parche microstrip de cobre

En la Figura 20 se puede observar que la ganancia para la antena de cobre a la frecuencia de  $f=0,7$  THz es de unos 5,73 dB, que no es demasiado buena y se observa presencia de lóbulos laterales sobre -13,8 dB



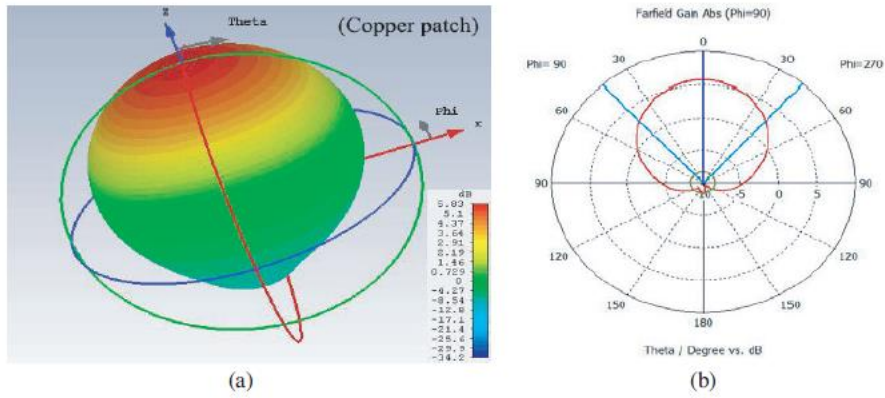


Figura 20. Diagrama radiación antena parche microstrip de cobre

A continuación se aplica grafeno como material de la antena y se consigue que la frecuencia de resonancia sea de 0,7 THz mediante la obtención de un nivel de Fermi de 0,76eV, para el que se obtienen unas pérdidas de retorno de -29 dB. Un resultado mucho mejor que el obtenido con el parche de cobre.

En la Figura 7 se puede ver la comparación de las pérdidas de retorno entre ambas antenas:

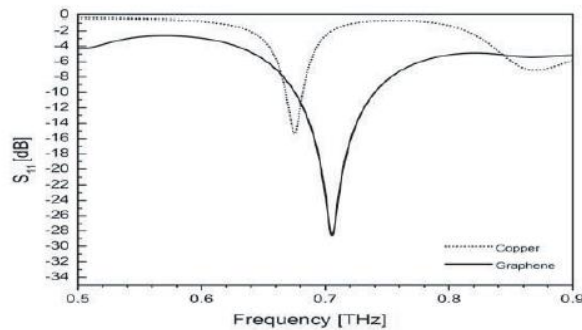


Figura 21. Comparación pérdidas de retorno

En la Figura 22 se puede observar que la ganancia para la antena de grafeno a la frecuencia de  $f=0,7$  THz es de unos 7,11 dB y nivel de lóbulo lateral de -16,8 dB

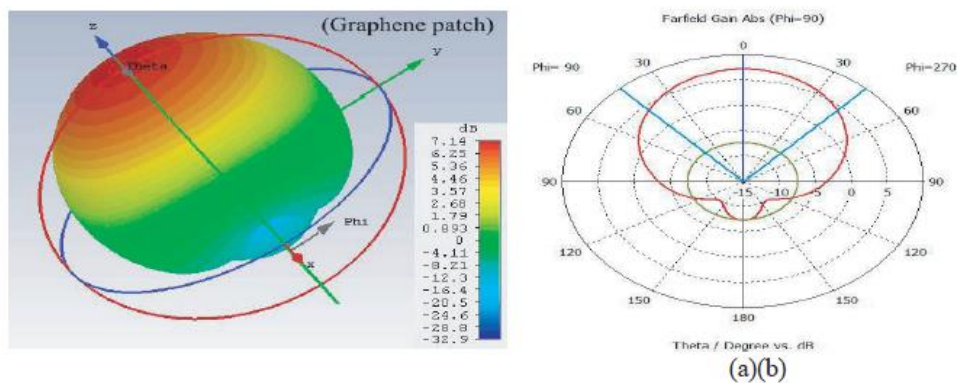


Figura 22. Diagrama radiación antena parche microstrip de grafeno

Esta antena se puede utilizar en aplicaciones aeroespaciales y su frecuencia de resonancia es de 0,7 THz. [17]

### 2.4.3 Trabajo: "Dual-Band Reconfigurable Graphene-Based Patch Antenna in Terahertz Band: Design, Analysis and Modeling Using WCIP Method"

En el estudio "Antena de Parche Reconfigurable de Doble Banda basada en Grafeno en la Banda de los Terahercios: Diseño, Análisis y Modelado mediante el Método WCIP" [18] se diseña una antena de parche de terahercios reconfigurable de doble banda con grafeno. El modelado se realiza mediante el uso de una nueva ecuación del método Wave Concept Iterative Process (WCIP) y los resultados se comparan con los obtenidos mediante la simulación con CST. En este trabajo se desarrolla una herramienta numérica eficiente y flexible para el modelado del grafeno.

En este trabajo se analiza el rendimiento del uso del grafeno en antenas reconfigurables en frecuencia en el régimen de terahercios para aplicaciones médicas y de imagen. Los resultados muestran claramente que la antena se puede sintonizar cambiando el potencial químico del grafeno.

El diseño de la antena de grafeno reconfigurable de doble banda se realiza mediante la comparación entre el grafeno no dopado y el metal en la misma estructura de antena. En la Figura 23 se observa la antena propuesta:

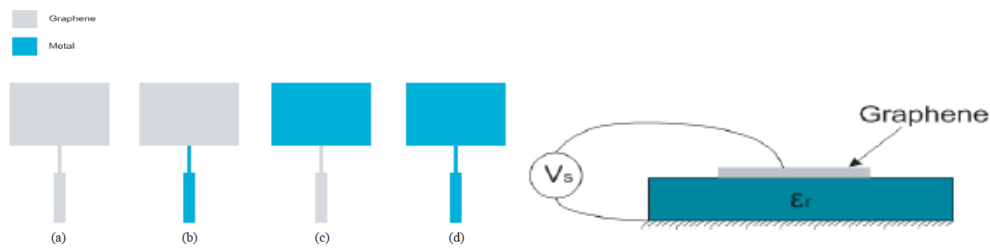


Figura 23. Estructura de la antena propuesta

En la Figura 24 observamos la variación del coeficiente de reflexión para las cuatro configuraciones propuestas anteriormente:

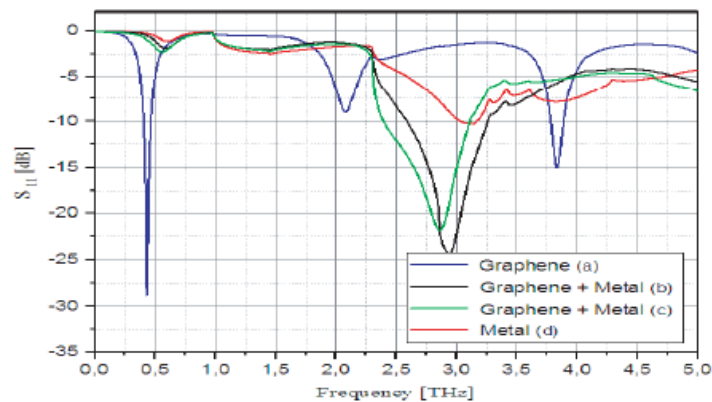


Figura 24. Coeficiente Reflexión configuraciones

Se puede ver que el coeficiente de reflexión de la antena basada en metal es el que peor se adapta en comparación con la antena basada en grafeno, la razón es que el

metal no es el material más adecuado para utilizarse en el rango de los terahercios. También se observa que las estructuras híbridas mejoran considerablemente el coeficiente de reflexión.

En la Figura 25 se muestran las pérdidas de retorno con la frecuencia para distintos potenciales químicos. Se puede observar que presenta dos bandas de frecuencia de resonancia en todo el rango de frecuencias 2,5-2,95 THz y 5,06-5,84 THz

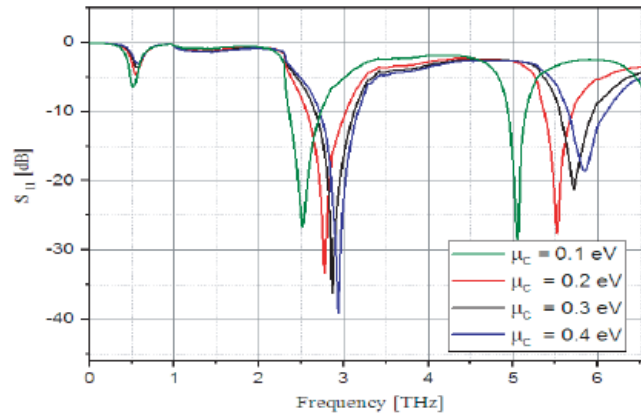


Figura 25. Coeficiente Reflexión para distintos potenciales químicos de grafeno

Por último, se compara el rendimiento de la antena de doble banda basada en grafeno propuesta para distintos materiales de sustrato. Los valores obtenidos se ven en la siguiente tabla:

Sustrato parámetros	Potencial químico [eV]	$F_{r1,2}$ [GHz]	Pérdida de retorno <sub>1,2</sub> [dB]	Ancho de banda <sub>1,2</sub> [GHz]	VSWR <sub>1,2</sub>
Duroid $\epsilon_r = 2.2$	$\mu_c = 0.1$	2500, 5061	-26. -57,28.52	386, 216	1.1, 1.08
	$\mu_c = 0.2$	2783, 5537	-33. -33,27.67	457, 341	1.04, 1.09
	$\mu_c = 0.3$	2861, 5729	-36. -12,21.30	467, 395	1.03, 1.19
	$\mu_c = 0.4$	2950, 5844	-39. -18,18.47	476, 463	1.02, 1.27
Arlon $\epsilon_r = 3$	$\mu_c = 0.1$	2250, 4460	-24. -41,36.93	340, 210	1.12, 1.34
	$\mu_c = 0.2$	2565, 5060	-42. -05,18.61	440, 425	1.01, 1.26
	$\mu_c = 0.3$	2570, 5070	-38. -34,17.98	450, 437	1.02, 1.29
	$\mu_c = 0.4$	2620, 5130	-44. -98,16.85	460, 448	1.01, 1.34
Poliimida $\epsilon_r = 3.5$	$\mu_c = 0.1$	2130, 4180	-21. -61,27.56	325, 225	1.18, 1.09
	$\mu_c = 0.2$	2310, 4560	-37. -22,20.56	425, 387	1.02, 1.20
	$\mu_c = 0.3$	2410, 4740	-42. -01,19.03	450, 435	1.01, 1.25
	$\mu_c = 0.4$	2460, 4850	-40. -63,17.81	495, 445	1.01, 1.29

Figura 26. Comparación distintos sustratos

El máximo ancho de banda de 495 GHz se obtiene para el sustrato de poliamida con  $\mu_c=0,4$  eV. También se observa que a medida que aumenta el valor del potencial químico aumenta el ancho de banda de la antena.

Los resultados obtenidos muestran que efectivamente la antena se puede sintonizar cambiando el potencial químico, por lo tanto el campo eléctrico de polarización es un factor importante para la antena de terahercios reconfigurable. [18]

#### 2.4.4 Trabajo: "Reconfigurable Graphene Annular Ring Antenna for Medical and Imaging Applications"

En el estudio "Antena Anular de Grafeno Reconfigurable para Aplicaciones Médicas y de Imagen" [19] se realiza el diseño de una antena de ranura de anillo concéntrico con grafeno y que dispone de un ancho de banda reconfigurable mediante la variación del potencial químico del grafeno. La antena presenta excelentes propiedades de rendimiento, un buen valor de pérdidas de retorno (-33.288 dB), un ancho de banda configurable desde los 255GHz hasta los 406GHz y una buena ganancia.

El diseño de la antena se realizó para reconfigurar y ampliar el ancho de banda de la misma, y para ello, se colocó un parche anular concéntrico de grafeno dentro de una ranura circular y una línea de alimentación de baja impedancia.

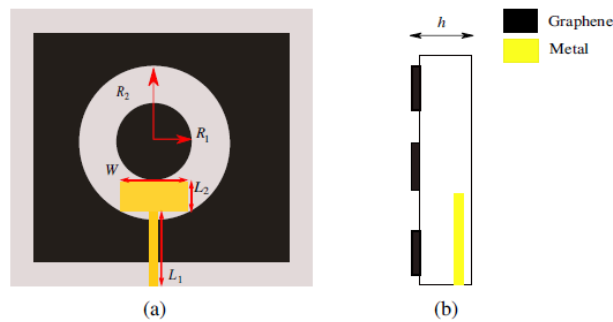


Figura 27. Esquema de antena propuesta

Los resultados de este estudio resultan prometedores para aplicaciones de imágenes médicas dentro del rango de los Terahercios. Se realiza una primera comparación de la antena propuesta con grafeno no dopado y de la misma estructura pero sustituyendo el grafeno por metal.

La Figura 28 muestra la pérdida de retorno para las dos configuraciones simuladas, se observa que los coeficientes de reflexión de la antena basada en metal se adaptan menos que los de grafeno. Además se puede observar que el grafeno ofrece un mejor rendimiento en términos del coeficiente de reflexión puesto que las pérdidas de retorno se reducen de -6.440dB hasta -15.125dB.

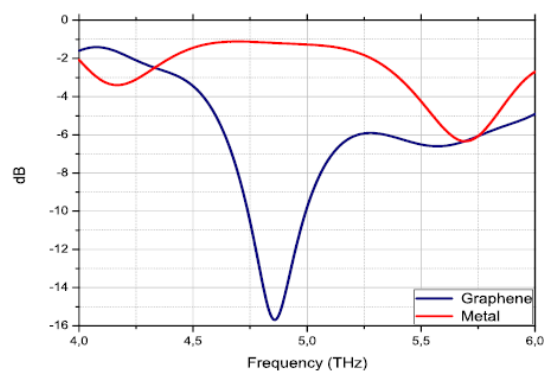


Figura 28. Pérdidas de retorno para antena de grafeno y antena metálica

En la Figura 29 se pueden observar los resultados del coeficiente de reflexión con la variación del potencial químico. Se pueden observar tres bandas de frecuencia de resonancia para los tres valores de  $\mu_c$  que cubren el rango de 4,667 THz hasta 5,055 THz. Para obtener estas tres reconfiguraciones del ancho de banda de la antena, es necesario variar el potencial químico.

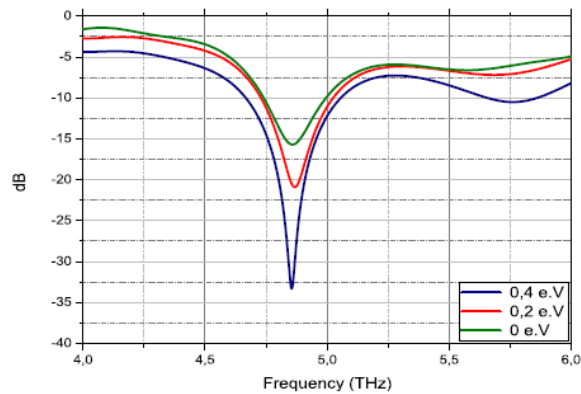


Figura 29. Pérdida de retorno para los diferentes valores de los potenciales químicos

En la siguiente tabla se indican los distintos parámetros del rendimiento de la antena para varios valores del potencial químico:

$\mu_c$ (e.V)	$f_r$ (THz)	BW (GHz)	VSWR	Gain (dB)	Return loss (dB)
$\mu_c = 0$	4.854	255	4.860	6.66	-15.125
$\mu_c = 0.2$	4.862	294	4.866	8.13	-20.896
$\mu_c = 0.4$	4.878	406	4.852	8.6	-33.288

Figura 30. Rendimiento de la antena propuesta para varios valores de potencial químico

Según los resultados obtenidos para un valor de  $\mu_c = 0.4 \text{ eV}$ , la antena presenta una mejor adaptación. Se puede observar el diagrama de radiación de la antena propuesta para un valor de  $\mu_c = 0.4 \text{ eV}$ , donde (a) Plano E y (b) Plano H:

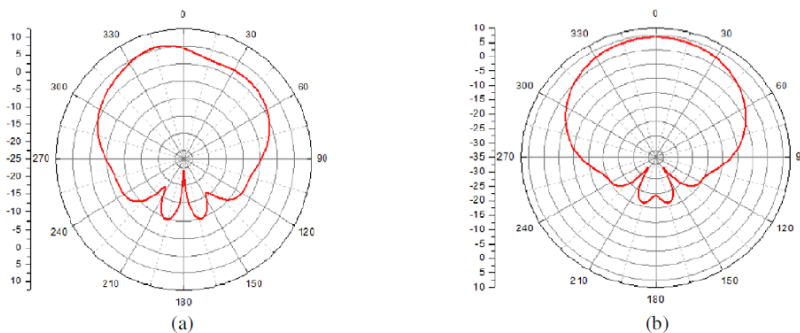


Figura 31. Diagrama de radiación

La antena anular de grafeno diseñada se puede destinar a aplicaciones médicas. Como conclusión de observa de las simulaciones que el grafeno muestra mejores resultados que el metal en términos de rendimiento con una frecuencia de resonancia bien adaptada. La antena diseñada permite reconfiguración del ancho de banda entre los 255 GHz y 406 GHz y dicha reconfiguración se puede realizar mediante la variación del potencial químico del grafeno ( $\mu_c$ ). [19]

### 2.4.5 Trabajo: "A Duo of Graphene-Copper Based Wideband Planar Plasmonic Antenna Analysis for Lower Region of Terahertz (THz) Communications"

En el trabajo "Análisis de un dúo de antenas plasmónicas planas de banda ancha basadas en grafeno y cobre para la región inferior de las comunicaciones en terahercios(THz)" [20], se plantea una idea de diseño para una antena de parche plasmónica planar basada en grafeno que funcionaría en el rango de los 0.1-10THz.

El diseño de la antena está compuesto por una sola capa de grafeno y cobre como material radiante sobre un sustrato de Silicio y con una masa parcial de cobre. También se utiliza únicamente el grafeno como material radiante para realizar la comparación entre ambas antenas.

En este trabajo se modela la conductividad del grafeno mediante la fórmula de Kubo que describe dos partes: la conductividad intrabanda y la conductividad interbanda. La primera, se debe principalmente a los fenómenos de generación y recombinación de pares electrón-hueco y la segunda, se corresponde a la conductividad de los portadores libres. Es sabido que si el rango de frecuencias está por debajo de la banda de los infrarrojos, en el rango inferior de los terahercios, la conductividad intrabanda es la dominante.

En la Figura 32 se observa el comportamiento decreciente de las partes real e imaginaria de la conductividad superficial del grafeno conforme aumenta la frecuencia, para un potencial químico de 0.4eV. Por esta razón, es esperable que con este potencial químico se produzcan menores pérdidas y por ello, se pueda lograr una alta eficiencia.

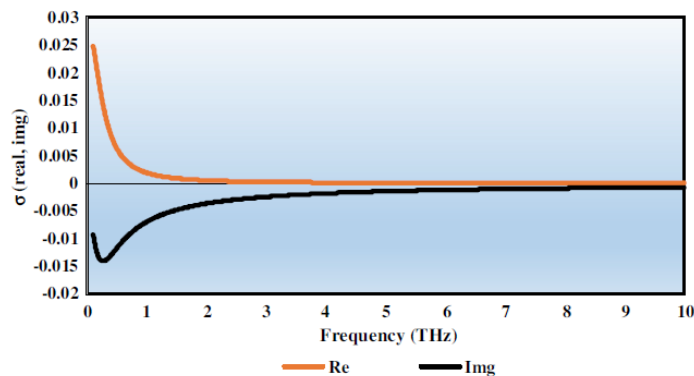


Figura 32. Parte real e imaginaria de la conductividad superficial del grafeno

En las láminas de grafeno se produce la propagación de las ondas SPP (ondas de plasmón superficial) cuyas propiedades de propagación se pueden determinar mediante la ecuación de dispersión del modo TM SPP. En este trabajo aparece la novedad de un diseño de antena basado en el efecto de aumentar el parche de arriba a abajo y viceversa con la intención de variar las impedancias y además generar bordes adicionales, ya que en el grafeno la radiación de los plasmones se produce en los bordes de la superficie radiante. La razón de introducir parches de cobre es para

reducir la velocidad de los electrones ya que es mayor en el grafeno que en el cobre y este efecto provoca un efecto condensador con una velocidad de carga más lenta.

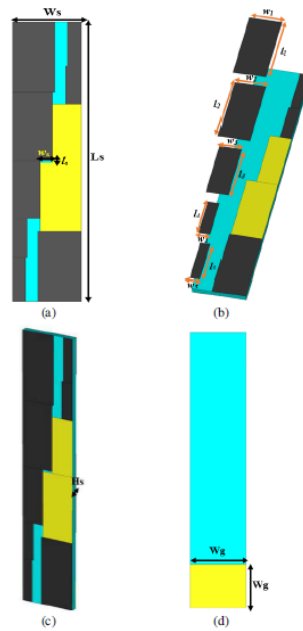


Figura 33. Diseño de antena de parche plasmónico de grafeno PPP

En la Figura 34 podemos ver el coeficiente de reflexión  $S_{11}(dB)$  simulado de una capa de grafeno como parche con el diseño propuesto que tiene un parche radiante combinando grafeno y cobre:

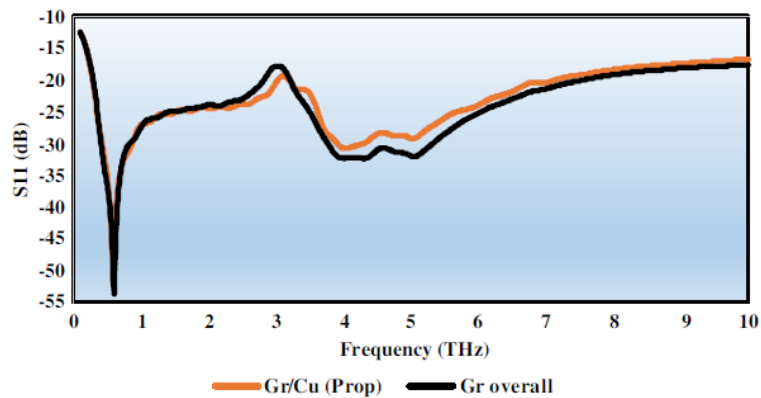


Figura 34. Comparación del  $S_{11}$  simulado

En la Figura 35 se observa el efecto de sintonía de la conductividad compleja del grafeno con respecto a la variación del potencial químico, y se puede verificar que los resultados mejoran cuanto mayor es el potencial químico:

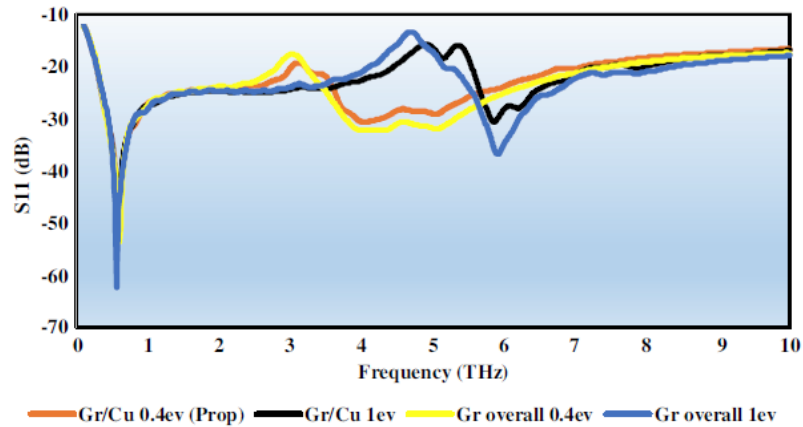


Figura 35. Efecto del potencial químico

En la Figura 36 se observa la Eficiencia de radiación de la antena es superior al 40% con un valor máximo es de casi el 48%:

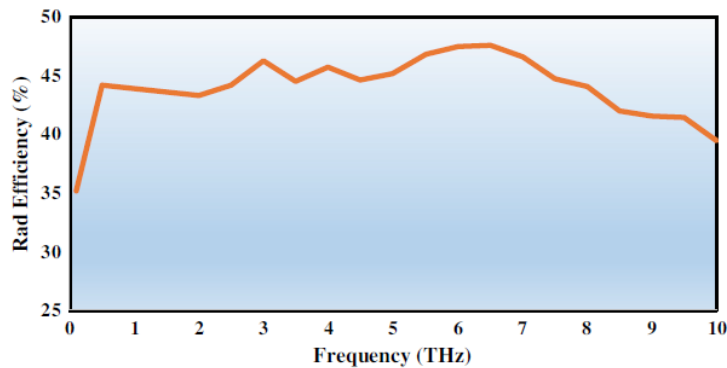


Figura 36. Eficiencia de radiación

Por último indicar que la radiación en la banda inferior de los terahercios o también denominada brecha de los THz, que iría hasta las frecuencias de los 0.310 THz, se pueden utilizar de forma segura en aplicaciones de imagen médica. La radiación procedente de los THz no resulta perjudicial para los tejidos vivos puesto que sólo es capaz de penetrar unos milímetros en la piel. [20]



## 2.5 Las antenas fractales de grafeno

No existen muchos trabajos de investigación sobre antenas fractales de grafeno, no obstante estos dos que se presentan se han considerado de interés y como punto de partida para el presente trabajo.

### 2.5.1 Trabajo: "Design of a Novel Reconfigurable Sierpiński Fractal Graphene Antenna Operating at THz Band"

En el trabajo de investigación "Diseño de una novedosa antena reconfigurable de grafeno fractal de Sierpiński que opera en la banda de THz" [23] se presenta la simulación del primer modelo de antena fractal de grafeno que permite sintonización en frecuencia gracias a la dependencia que tiene la frecuencia con la conductividad del grafeno.

Para obtener el modelo de conductividad del grafeno se deposita una hoja de grafeno en un sustrato compuesto por dióxido de silicio y silicio de alta resistividad. La forma de obtener la modulación de la conductividad es aplicar una tensión entre el grafeno y el electrodo inferior. Puesto que el potencial químico del grafeno varía con el voltaje aplicado, esto se traduce en un cambio en la conductividad superficial del grafeno, que se puede obtener mediante la ecuación de Kubo, y por lo tanto, un cambio en la frecuencia de resonancia de la antena.

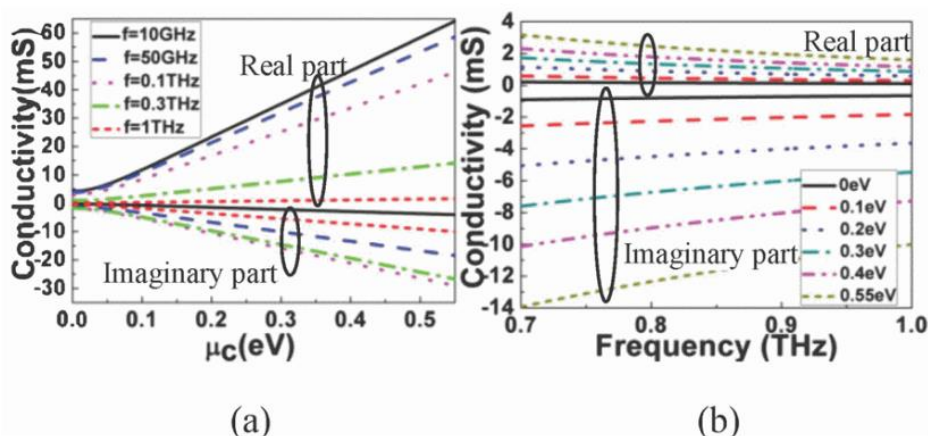


Figura 37. Conductividad superficial del grafeno vs potencial químico o frecuencia

Se observa que a mayores frecuencias de terahercios, el grafeno se comporta como semiconductor, mientras que a menor frecuencia muestra propiedades similares a las de un material metálico. Las antenas fractales simuladas en este trabajo son de tipología tapiz de Sierpiński hasta tercer orden y la capa de grafeno se deposita bajo la de cobre y todo el conjunto se simula mediante el software CST.

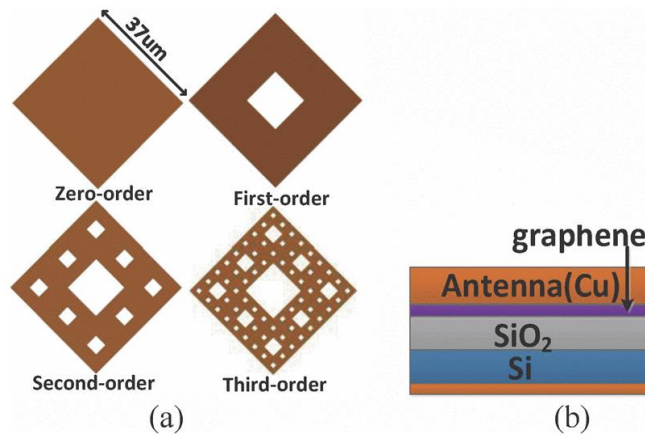


Figura 38. Esquemas antenas fractales de Sierpiński desde el orden cero al tercero

En la simulación del parámetro  $S_{11}$  se observa que cuanto mayor es el valor del potencial químico del grafeno  $\mu_c$  mayor será la frecuencia de resonancia de la antena. Para  $\mu_c = 0,55$  eV tenemos las frecuencias de resonancia  $f_1 = 0.802$  THz,  $f_2 = 0.822$  THz,  $f_3 = 0.836$  THz y  $f_4 = 0.843$  THz para los órdenes del fractal de cero a tres, respectivamente.

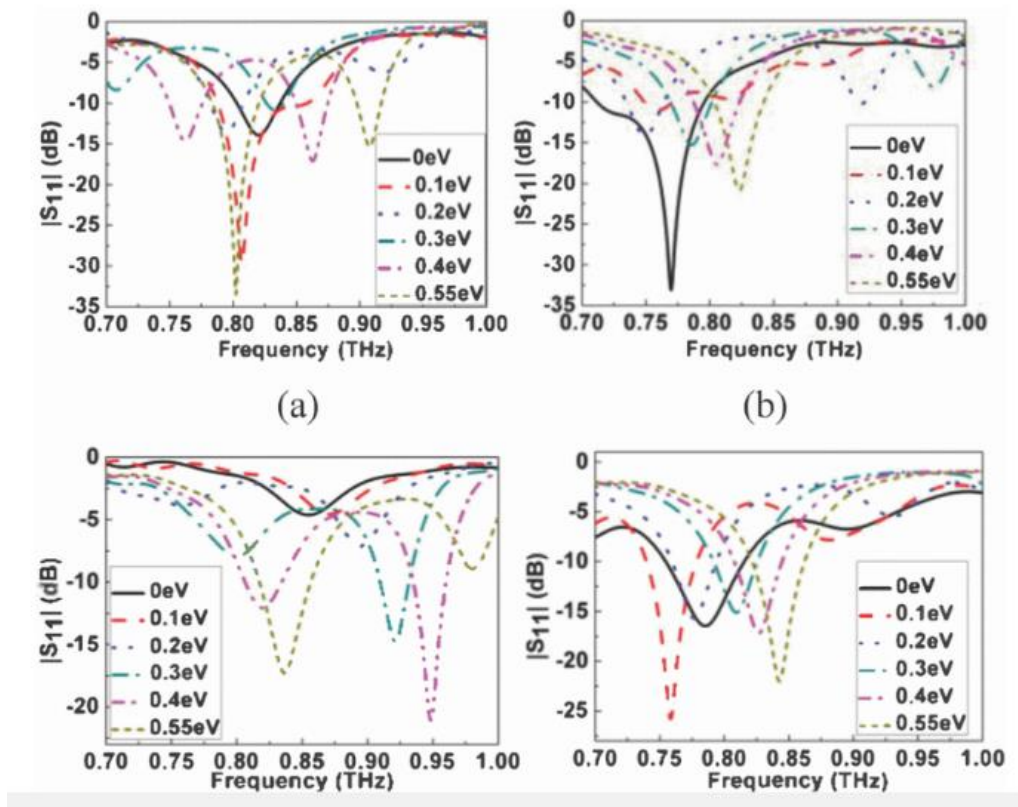


Figura 39. Simulación del parámetro  $S_{11}$  para diferentes valores de  $\mu_c$

En la simulación de los diagramas de radiación de la antena de grafeno donde  $f = 0.8 \text{ THz}$  y  $\varphi = 90^\circ$  para los distintos órdenes del fractal se observa que la ganancia de la antena se incrementa con el valor del potencial químico  $\mu_c$ . Se determina que la antena Sierpiński de segundo orden es la que tiene la ganancia más alta y por lo tanto, presentaría un rango de reconfiguración más amplio.

Se observa que el valor máximo de la ganancia es 2.1, 3.7, 4.1 y 3.7dB para los órdenes de cero a tres, respectivamente.

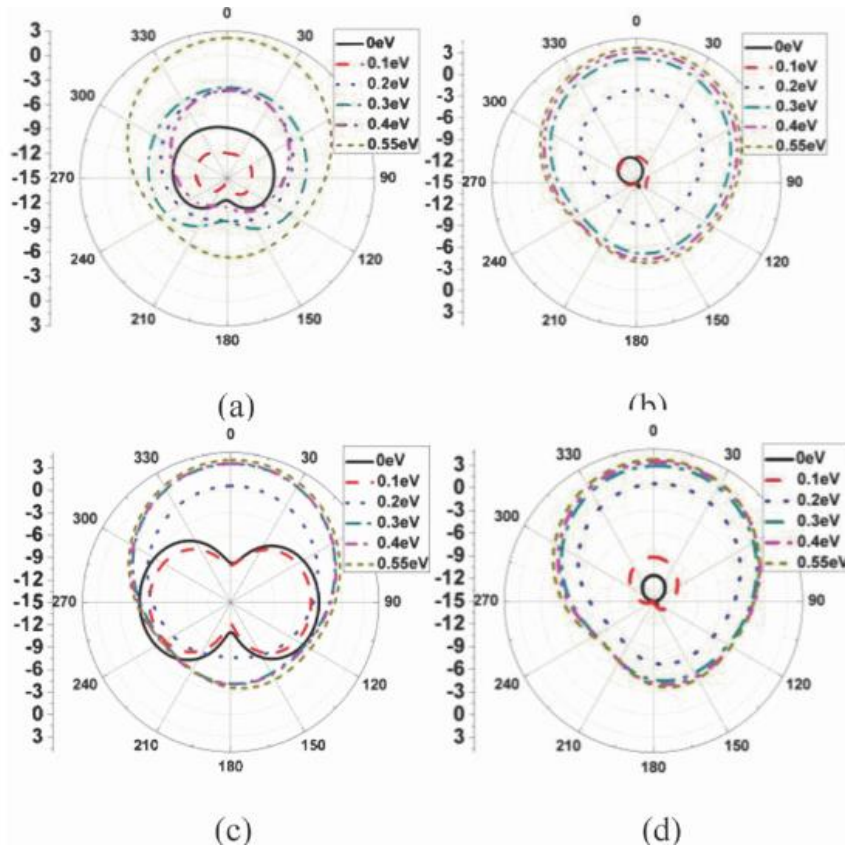


Figura 40. Diagramas de radiación para diferentes valores de  $\mu_c$

El trabajo concluye con la demostración y comparación de la característica reconfigurable de la antena de grafeno fractal para diferentes órdenes mediante simulación con el software CST y por lo tanto, la antena fractal de Sierpiński planteada podría funcionar bien en la banda de terahercios y lograr una mayor ganancia que las antenas de parche convencionales. Además su rango configurable podría ser más amplio. [23]

### 2.5.2 Trabajo: "A Preliminary Study of a Graphene Fractal Sierpiński Antenna"

Dentro de los últimos trabajos de investigación realizados en el diseño de antenas fractales de grafeno destaca el "*Estudio Preliminar de una Antena de Grafeno Fractal de Sierpiński*" [24] ya que presenta unos resultados muy prometedores. Se trata de un estudio inicial del comportamiento de una antena fractal de grafeno que opera en el rango de frecuencias de los terahercios con la posibilidad de modular la emisión, de modo que al tratarse de una antena sintonizable de alta frecuencia se posibilita su uso en comunicaciones de radio en el ámbito militar.

En dicho trabajo se estudia por primera vez la combinación de la plasmónica del grafeno y los fractales de tipo Sierpiński en una antena de banda ancha en el rango de frecuencias de los terahercios, con la intención de darle aplicación en las comunicaciones inalámbricas nanoelectrónicas. Se estudia una antena monopolo con topología de triángulo fractal de Sierpiński e implementada con oro, grafeno, silicio y dióxido de silicio mediante la resolución de la ecuación de Maxwell para un índice de refracción complejo en el dominio de la frecuencia en el rango de los 10-750 THz. La antena se construye depositando unas cuantas capas de grafeno sobre una capa nanométrica de silicio sobre un sustrato de dióxido de silicio y se realiza el modelado de cada material de la nanoestructura mediante el índice de refracción complejo. Los valores de  $n$  y  $k$ , que son la parte real e imaginaria del índice de refracción complejo se obtienen a partir de datos experimentales y modelos teóricos. La conductividad óptica del grafeno en la región de operación de 10-300 THz se puede modificar mediante la variación del voltaje de puerta y este diseño lo permite. De este modo, se comparan las prestaciones de la antena diseñada con grafeno y la antena diseñada con oro para diferentes voltajes de puerta.

Este diseño proporciona el control de la frecuencia de resonancia de la antena ya que una de las propiedades plasmónicas de las nanoantenas de grafeno es que proporcionan sintonización en el rango de frecuencias de los terahercios debido a que la conductividad superficial del grafeno puede modificarse aplicando un sesgo electrostático, que cambia el potencial químico (nivel de Fermi).

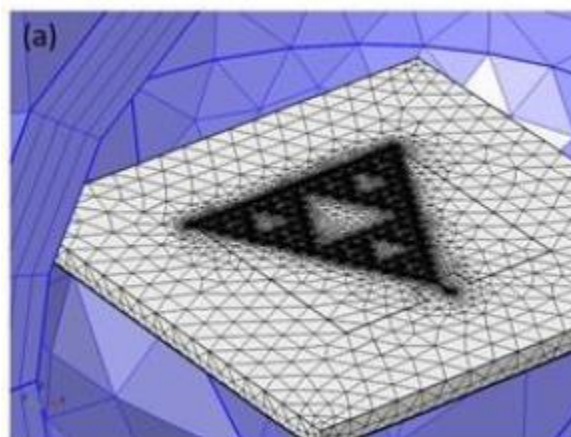


Figura 41. Modelo de antena Sierpiński

En este trabajo se modelan las propiedades del material para longitudes de onda de  $30\text{-}0.4\mu\text{m}$  (10-750 THz), se realizan simulaciones preliminares para una antena fractal monopolo de Sierpiński de tercer orden de dimensiones nanométricas y se obtienen tres frecuencias de resonancia  $f_1 = 35\text{ THz}$ ,  $f_2 = 110\text{ THz}$  y  $f_3 = 215\text{ THz}$ .

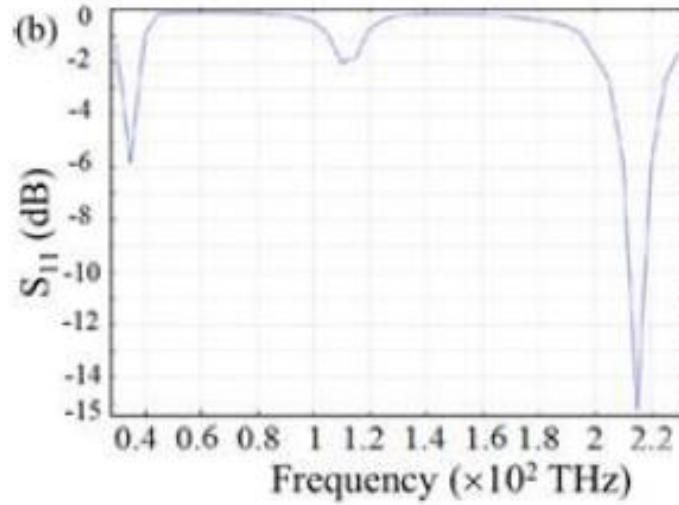


Figura 42. Parámetro  $S_{11}$  global de Sierpiński vs la frecuencia con tres resonancias a 35, 110 y 215 THz

En el resultado de la simulación, la emisión en campo lejano es similar a la que se podría conseguir un elemento radiante modelado como un PEC.

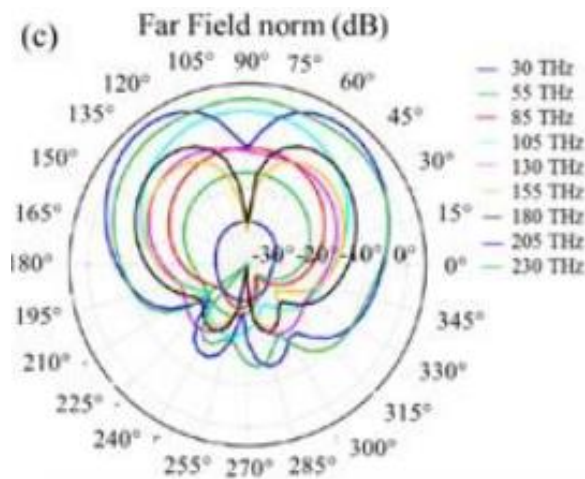


Figura 43. Patrones de coordenadas polares de campo lejano en las frecuencias de resonancia

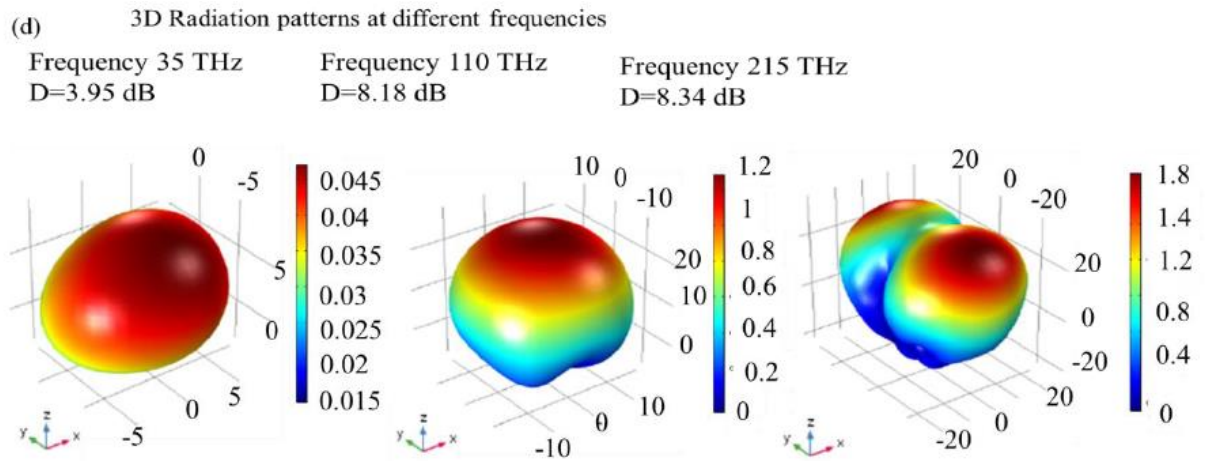


Figura 44. Diagrama de radiación en 3D

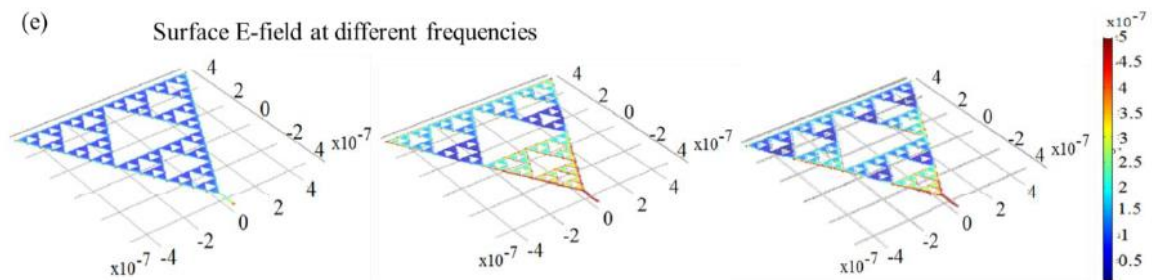


Figura 45. Campos E superficiales de Sierpiński en las frecuencias de resonancia

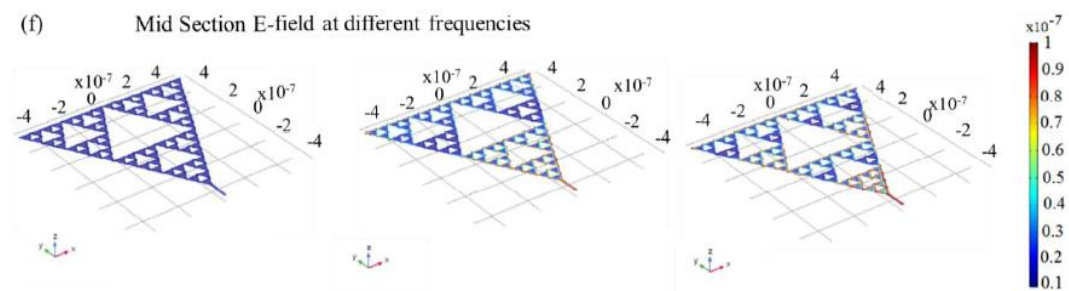


Figura 46. Campos E de la sección media de Sierpiński en las frecuencias de resonancia

El trabajo concluye indicando que se trata de un estudio preliminar de una antena fractal de grafeno que trabaja en las frecuencias de terahercios y que tiene la posibilidad de modular la emisión. Pero se indica que el modelo debe ser cuidadosamente revisado para simular la antena propuesta. [24]

## 3. Diseño

### 3.1 Enfoque de la solución

El presente trabajo consta de varias partes:

En primer lugar se plantea el diseño de antena propuesta.

En segundo lugar se implementa una interfaz en Matlab que permite la comunicación con CST Studio Suite SE utilizando la tecnología Windows COM. De modo que se genera una función principal en código Matlab y las funciones adicionales necesarias para poder crear un modelo de antena fractal completo en CST lanzando una instrucción con la parametrización desde la consola de comandos de Matlab.

En tercer lugar, una vez se ha conseguido automatizar la generación de modelos en CST, se realizan simulaciones de diversos órdenes del diseño fractal para material grafeno con distintos sustratos para el rango de frecuencias de terahercios.

En último lugar se realiza la comparación entre los distintos modelos obtenidos para finalmente elegir los dos mejores modelos.

### 3.2 Antena propuesta

El modelo de antena propuesto consiste en una antena fractal de grafeno basada en el triángulo de Sierpiński para distintos órdenes sobre dos tipos de sustrato: dióxido de silicio y fibra de vidrio. La antena es tipo microstrip y el parche está compuesto de grafeno y el plano de masa es un conductor perfecto. El diseño se basa en el modelo de antena planteado en [24] modificado.

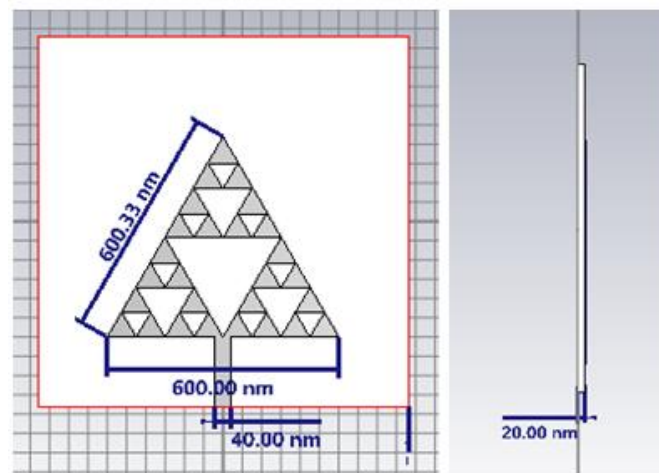


Figura 47. Dimensiones antena propuesta

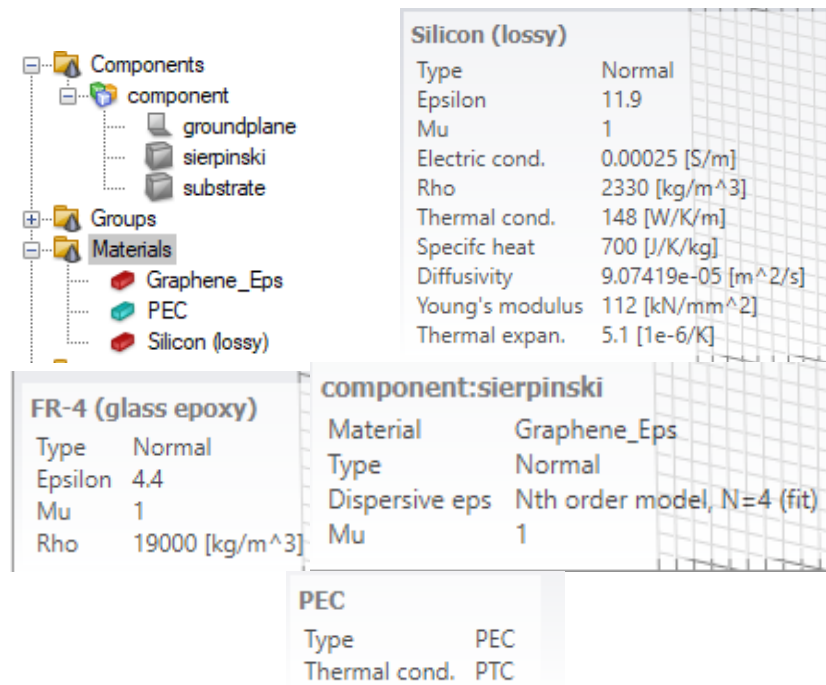


Figura 48. Materiales del componente

### 3.3 Creación de material grafeno

A continuación se detalla la forma de generar el material grafeno en CST para poder utilizarlo en los diseños. La forma de proceder es mediante la creación de la siguiente Macro en CST:

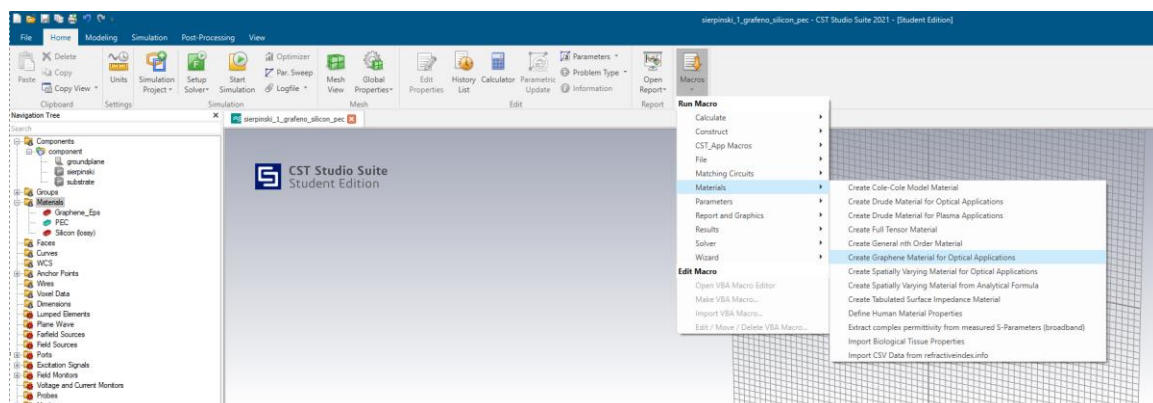


Figura 49. Macro CST material grafeno

Los parámetros que se deben especificar son los indicados en la siguiente imagen, donde el potencial químico del grafeno se ha ido modificando para obtener los tres tipos de material grafeno presentados en la memoria.



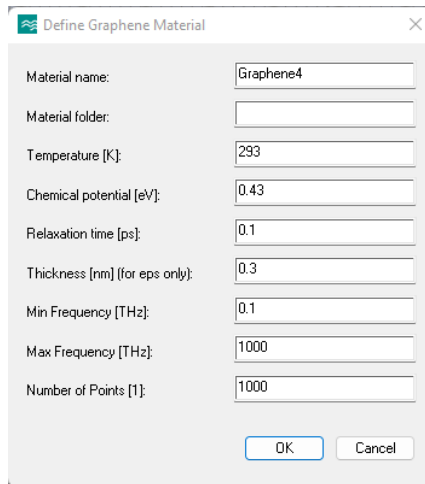


Figura 50. Parámetros material grafeno

Se generan dos materiales grafeno, uno sin espesor (Graphene4) y el otro con la especificada (Graphene4\_Eps). Este último será el que se utilizará en los cálculos.

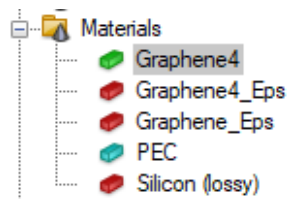


Figura 51. Material grafeno

Se hace doble click sobre cada uno de ellos y se añaden a la librería de materiales.

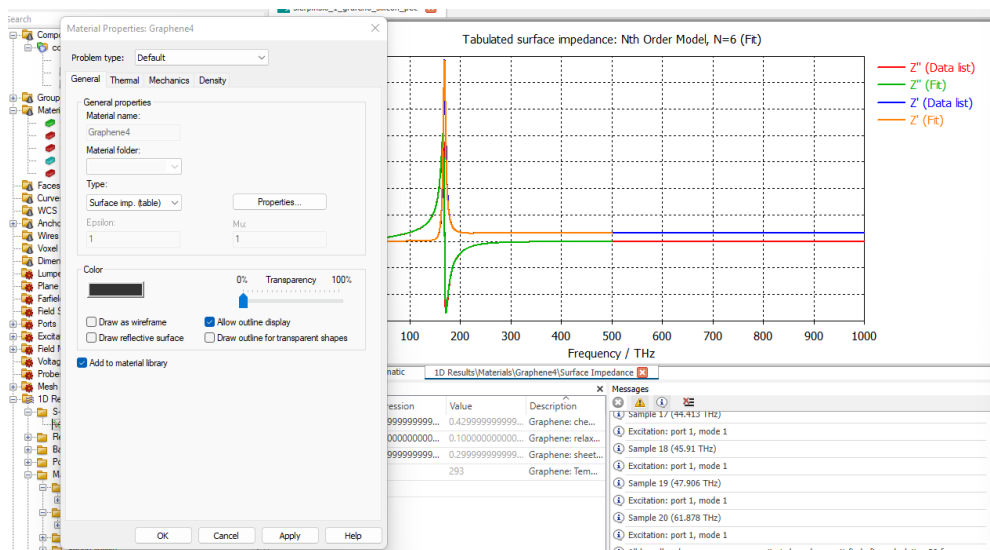


Figura 52. Añadir grafeno a la librería de materiales

Una vez creado el material grafeno, se extrae el fichero de configuración del mismo y se utiliza para programar la función de Matlab que genera dicho material de forma automática.

### 3.4 Creación de sustrato *glass\_epoxy*

Este sustrato tiene la siguiente configuración:

Sustrato FR4 glass epoxy  
Constante dieléctrica  $\epsilon_r = 4.4$ ,  
Tangente de pérdidas  $\delta_e = 0.02$   
Densidad de masa de  $19,000 \text{ kg/m}^3$

Al igual que con el material grafeno, una vez creado el sustrato FR4 glass epoxy, se extrae el fichero de configuración para crear la función de Matlab que genera dicho material de forma automática.

### 3.5 Ejemplo de funcionamiento

La interfaz de Matlab está compuesta de una función principal que llama al resto de funciones adicionales:

```
fractal_sierpinski(it, materialpatch, sustrato, groundplane)
```

Dado el número de iteraciones genera la antena de parche fractal de Sierpiński de orden  $n$  en CST, con el material, sustrato y plano de masa indicados. El programa genera un nuevo proyecto de CST con la parametrización: unidades, rango de frecuencia, dimensiones del elemento, puerto, monitores y realiza la simulación con el “FD Solver”.

Las funciones adicionales creadas son las indicadas con una breve descripción de lo que realizan:

`addmonitor.m`: función para definir el e-field, h-field y farfield monitor.  
`boolean_add.m`: función para realizar la adición booleana.  
`boolean_subs.m`: función para realizar la substracción booleana.  
`brick.m`: función para crear un bloque.  
`fdsolver.m`: función para configurar el tipo de *solver* y lanzar simulación.  
`frecuencia_range.m`: función para seleccionar el rango de frecuencias.  
`ground_gold.m`: función para definir el material oro.  
`mat_cooperpurelossy.m`: función para definir el material cobre.  
`mat_grapehe_eps.m`: función para definir el material grafeno  $\mu_c = 0.1 \text{ eV}$   
`mat_grapehe2_eps.m`: función para definir el material grafeno  $\mu_c = 0.2 \text{ eV}$   
`mat_grapehe4_eps.m`: función para definir el material grafeno  $\mu_c = 0.43 \text{ eV}$   
`pickface.m`: función para seleccionar una cara del objeto.  
`rectanglefeed.m`: función para dibujar un rectángulo.  
`subs_FR4lossy.m`: función para definir el material FR4.  
`subs_glassepoxy.m`: función para definir el material resina de vidrio.  
`subs_siliconlossy.m`: función para definir el material dióxido de silicio.  
`transform.m`: función para realizar la transformación de un componente.  
`triangle.m`: función para dibujar un rectángulo.

`units.m`: función para definir las unidades del proyecto.  
`waveguideport.m`: función para un puerto tipo guía de onda.

Para ver el funcionamiento se generan dos antenas:

- 1) Antena fractal de orden 2, con material grafeno, sustrato silicio y plano de masa conductor perfecto.

Desde Matlab se invoca a la función principal con los parámetros indicados:

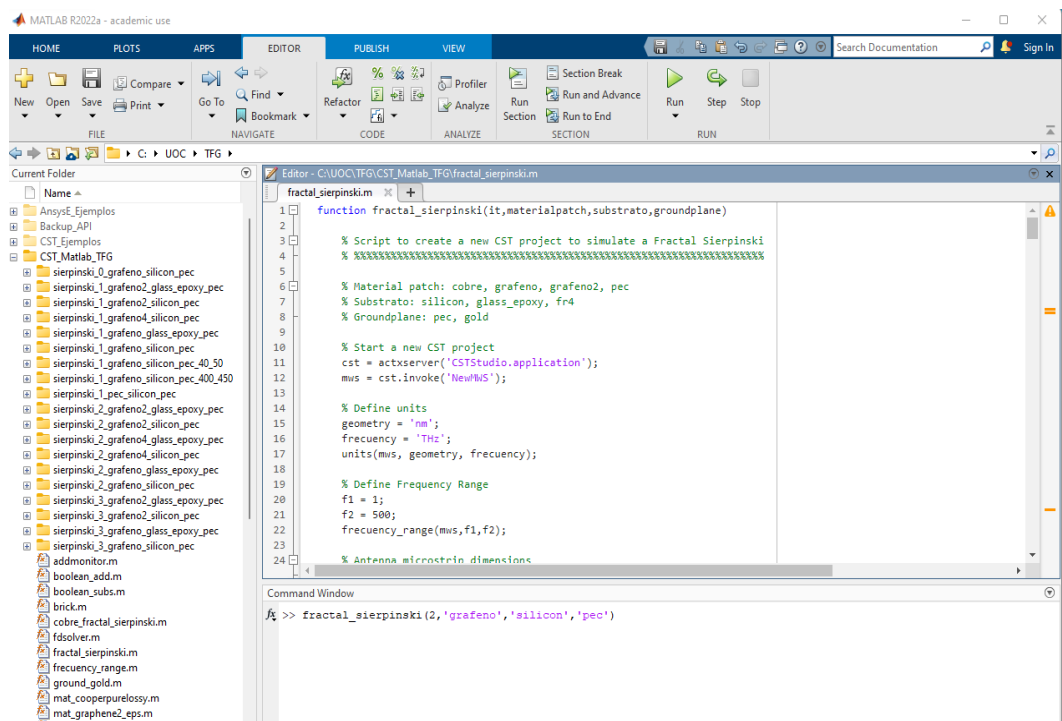


Figura 53. Función fractal\_sierpinski ejemplo 1

Se inicia la ejecución y se abre una nueva instancia de CST en la que se genera el nuevo proyecto con los parámetros indicados:

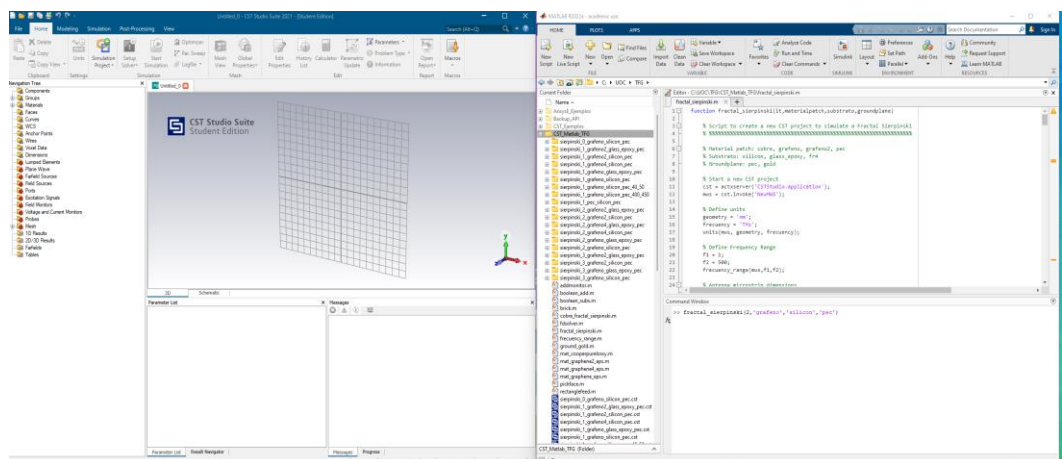


Figura 54. Nuevo proyecto CST ejemplo 1

La función principal muestra un log de los pasos que va ejecutando. Se asignan las unidades, se asigna el rango de frecuencias para la simulación, crea el diseño de fractal con el orden, material del parche, material del sustrato y material del plano de masa indicados. Le asigna el puerto para la simulación de tipo guía de onda y asigna el tipo de solucionador para la simulación, que en este caso es “*HF Frequency Domain*”.

Una vez generado el modelo, lanza la simulación:

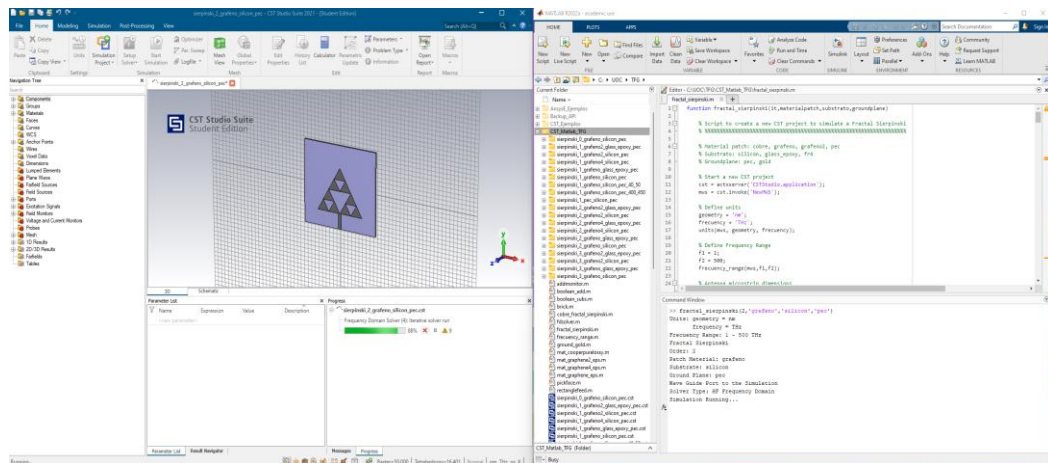


Figura 55. Simulación CST ejemplo 1

Una vez finalizada la simulación, la ejecución del programa termina y se puede ver el resultado de la misma.

Con esta ejecución se proporcionan: componente completamente diseñado, gráfica del parámetro  $S_{11}$ , diagramas de radiación para campo lejano a las frecuencias de resonancia y todos los parámetros que por defecto proporciona CST en este tipo de simulación.

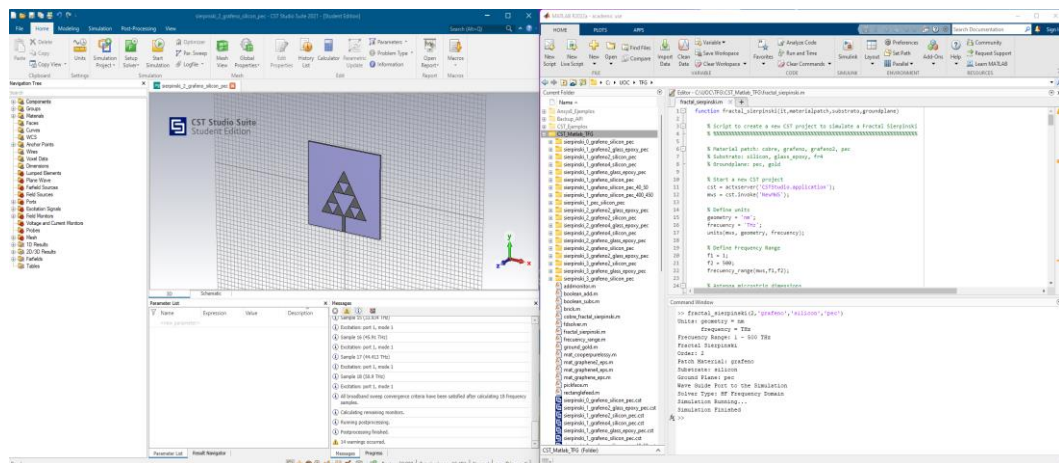


Figura 56. Simulación finalizada ejemplo 1

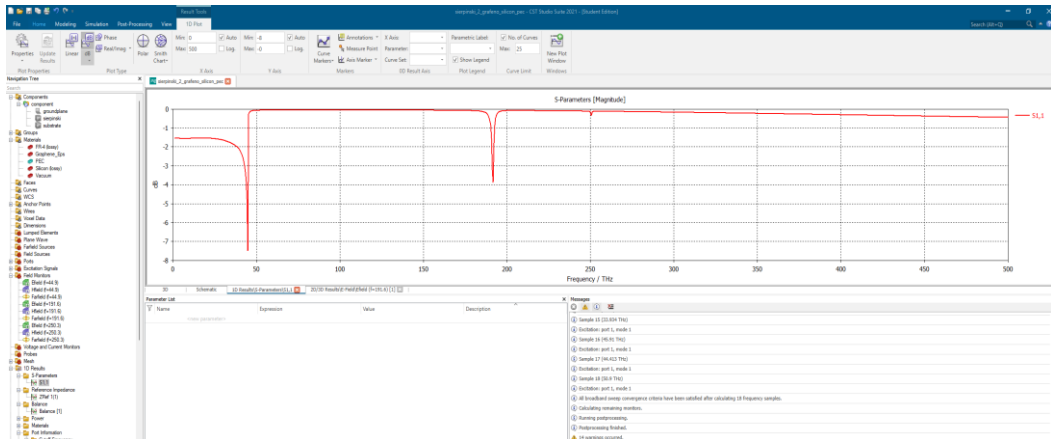


Figura 57. Parámetro  $S_{11}$  ejemplo 1

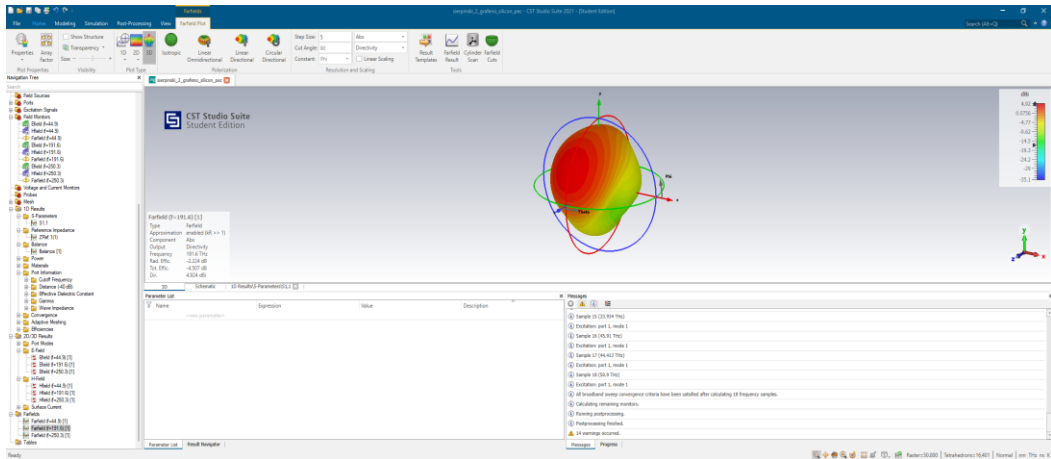


Figura 58. Diagrama de radiación ejemplo 1

2) Antena fractal de orden 3, con material grafeno4, sustrato glass\_epoxy y plano de masa pec.

Nuevamente desde Matlab se invoca a la función principal con los parámetros:

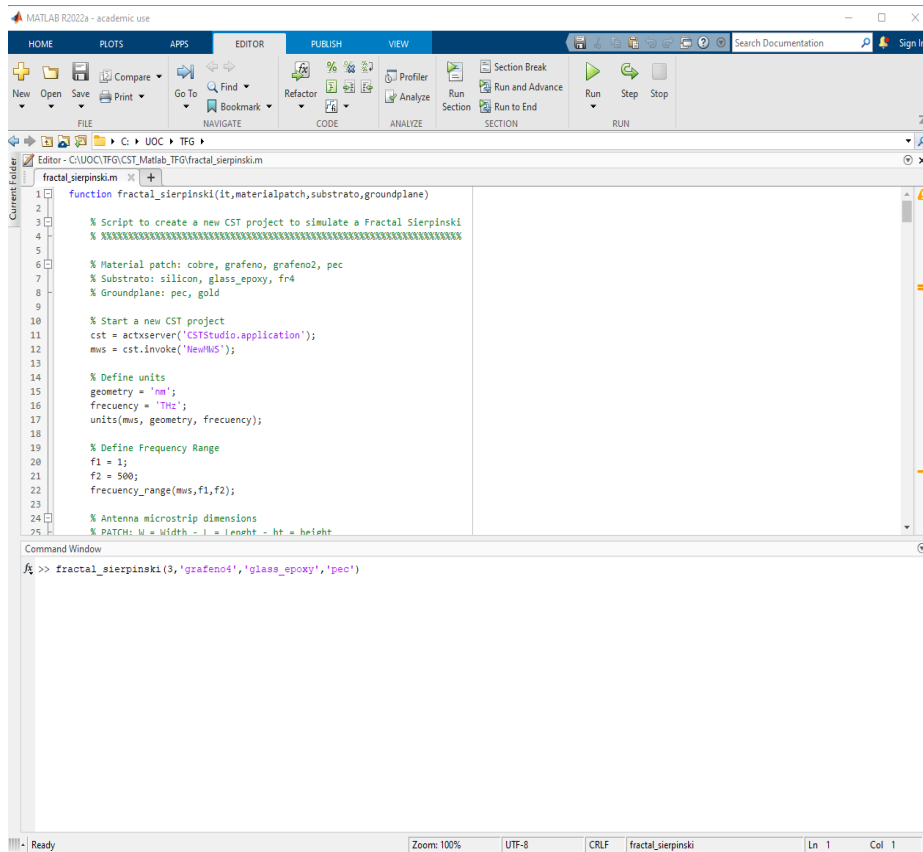


Figura 59. función fractal\_sierpinski ejemplo 2

Se inicia la ejecución y se abre una nueva instancia de CST en la que se genera el nuevo proyecto con los parámetros indicados:

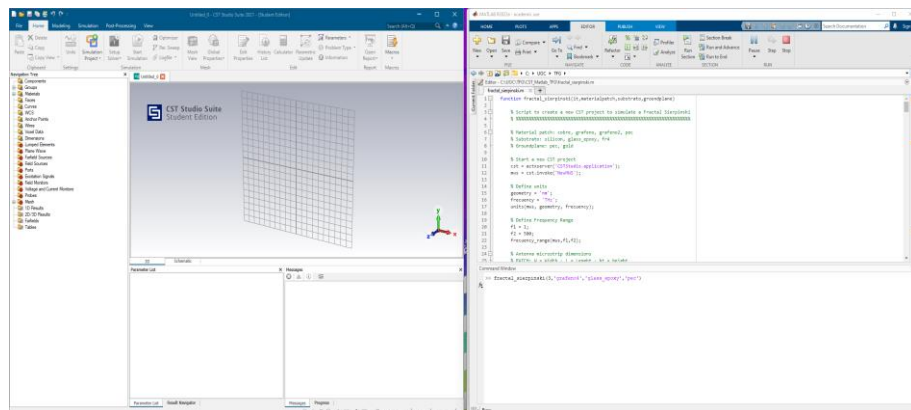


Figura 60. Nuevo proyecto CST ejemplo 2

Una vez generado el modelo, se lanza la simulación:

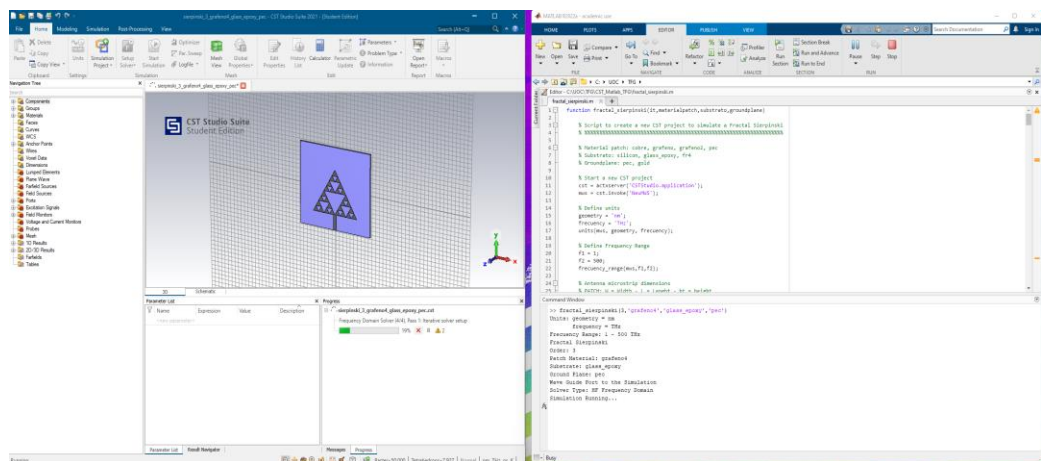


Figura 61. Simulación CST ejemplo 2

Una vez finalizada la simulación, se puede observar tanto el componente diseñado como los distintos parámetros que proporciona este tipo de simulación en CST.

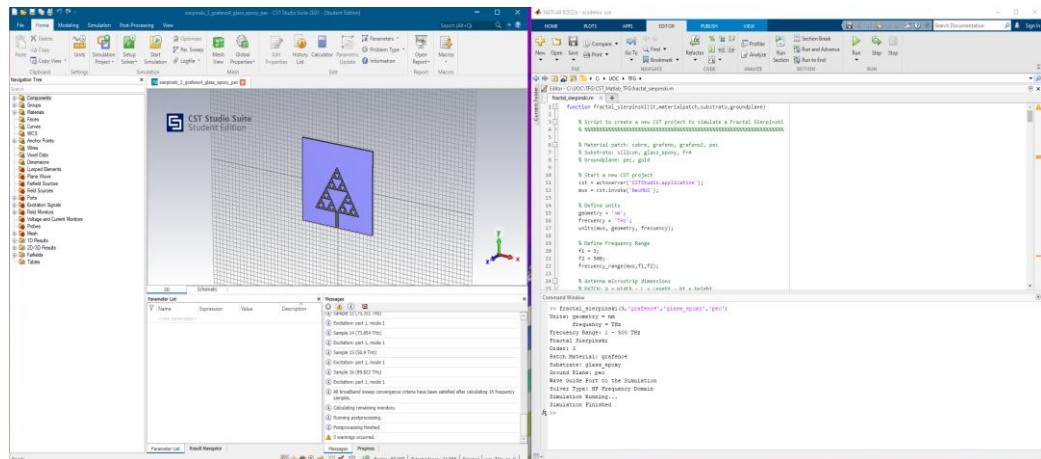


Figura 62. Simulación finalizada ejemplo 2

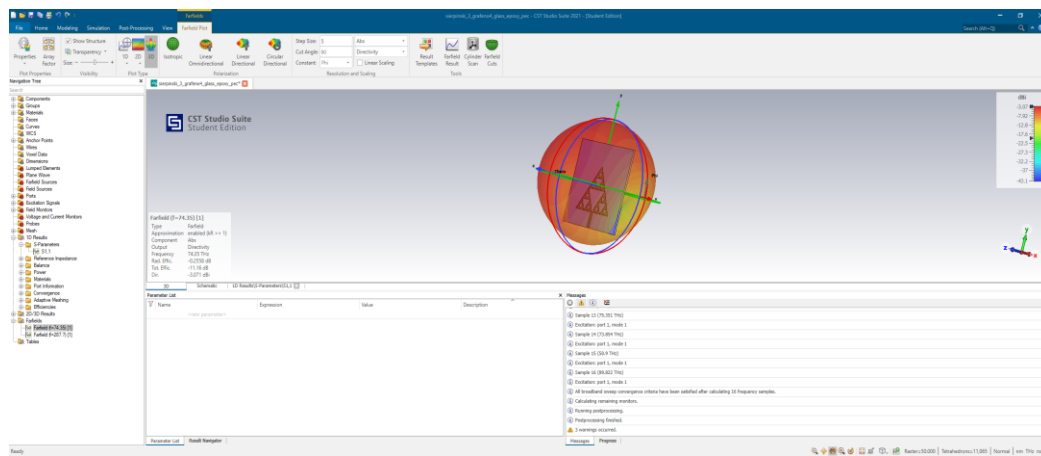


Figura 63. Diagrama de radiación ejemplo 2

### 3.6 Cálculos y simulaciones realizadas

Como idea inicial se parte de una antena de parche rectangular de dimensiones  $W \times L \times h$  sobre un sustrato dieléctrico de constante  $\epsilon_r$  de la que se obtiene la frecuencia de resonancia para el modo fundamental.

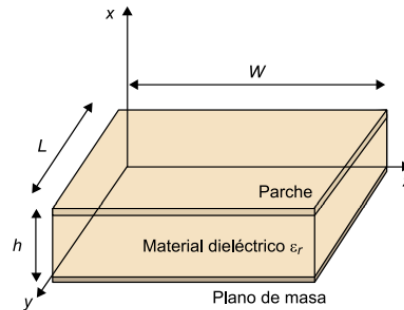


Figura 64. Parche rectangular sobre sustrato dieléctrico

Se pretende tener una aproximación de las frecuencias de resonancia que se obtendrán con la antena de parche fractal para las dimensiones de nanómetros propuestas. Posteriormente se comprueba que efectivamente se corresponden:

$$\text{Modo } TM_{10} : f_r = \frac{1}{2\pi\sqrt{\mu\epsilon}} \cdot \frac{\pi}{L}$$
$$L = 960\text{nm}$$

$$\text{Si } \epsilon_r = 11.9 \Rightarrow f_r = 45.2 \text{ THz}$$
$$\text{Si } \epsilon_r = 4.4 \Rightarrow f_r = 74.34 \text{ THz}$$

Los distintos parámetros obtenidos para cada modelo se consiguen de dos maneras: en primer lugar se lanza una simulación en el dominio de la frecuencia desde el código de Matlab y en segundo lugar se realiza una parte de post-procesado desde el propio programa CST. A continuación se detallan los parámetros obtenidos con cada uno de los métodos:

**Se calcula mediante la función fractal\_sierpinski:**

1. S<sub>11</sub>
2. *Farfields* para las frecuencias de resonancia.
3. Directividad
4. Eficiencias de Radiación

**Se calcula con post-procesado en CST:**

5. Ganancia.
6. Ganancia Realizable.
7. Potencia Total radiada (TRP)



### 3.6.1 Post-procesado en CST de Ganancia, Ganancia Realizable y Directividad

En los tres casos los parámetros se obtienen de la misma forma, mediante una template de *post-processing* en CST. Los pasos a seguir son los siguientes:

Después de la simulación para obtener la "*Realized Gain*" de la antena: En Post-Processing ->Template Based Postprocessing -> Farfield and Antenne Properties -> Farfield Results:

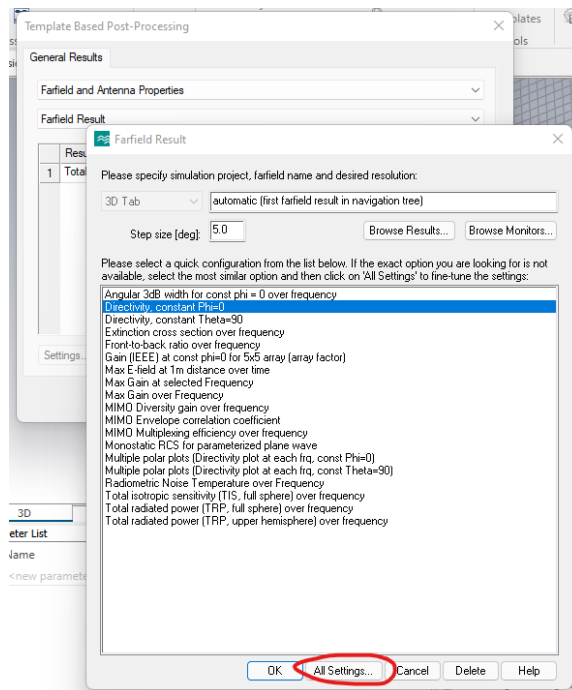


Figura 65. Template Based Postprocessing

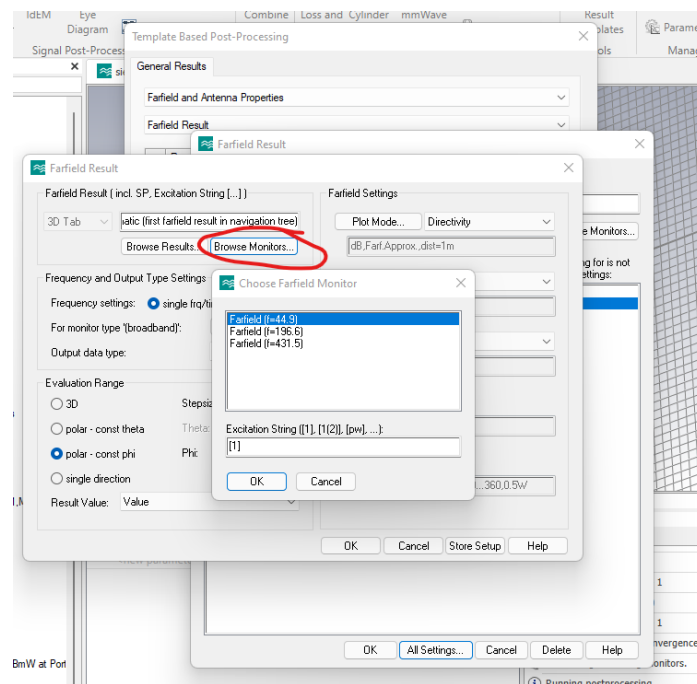


Figura 66. Template Based Postprocessing monitor

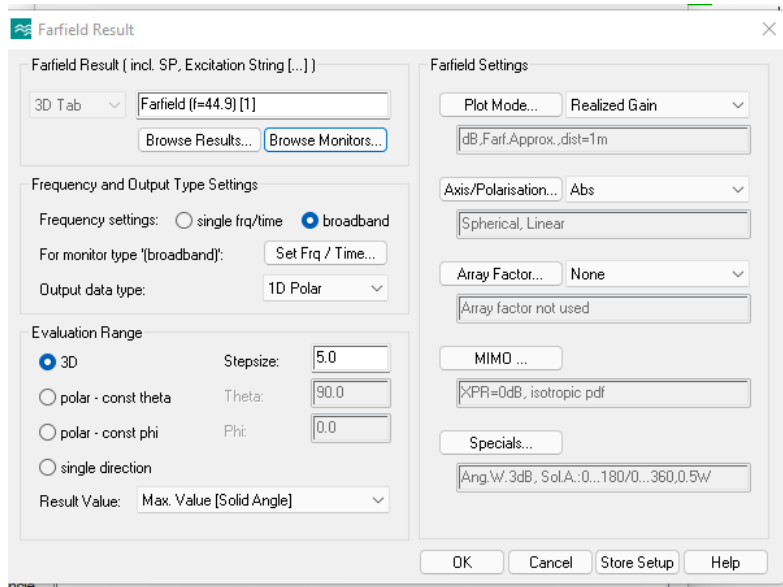


Figura 67. Template Based Postprocessing parámetros

Es necesario aclarar también el método utilizado para obtener el % de Eficiencia de Radiación. Se realiza con los tres pasos indicados a continuación:

1. Se define el *farfield monitor* para la frecuencia deseada.
2. Al dibujar el diagrama de radiación de campo lejano en 3D obtenemos la eficiencia radiada y la total en dB.
3. Se realiza la conversión de dB a porcentaje:

$$P = \left(10^{\frac{x}{10}}\right) \cdot 100 = \%$$

Donde X es la eficiencia en dB.

Por último, se ha considerado interesante incluir las definiciones del manual de CST para la Eficiencia de Radiación, la Eficiencia total y la Directividad:

### Farfield Definitions

**Approximation:**

$$E_{\text{tot}}(\text{enabled}) = \sqrt{|E_x|^2 + |E_y|^2}$$

$$E_{\text{tot}}(\text{disabled}) = \sqrt{|E_x|^2 + |E_y|^2 + |E_z|^2}$$

**Radiation efficiency:**

$$e_{\text{rad}} = \frac{\text{radiated power}}{\text{accepted (input) power}}$$

**Total efficiency:**

$$e_{\text{total}} = \frac{\text{radiated power}}{\text{stimulated power}}$$

**Directivity:**

$$D(\theta, \varphi) = 4\pi \frac{\text{power radiated per unit solid angle}}{\text{radiated power}}$$

**Component definition:**

Name	Height 1	Height 2	Component 1	Component 2
Spherical	Theta	Phi	Theta	Phi
Loading 2 AE	Elevation	Azimuth	Azimuth	Elevation
Loading 2 EA	Alpha	Epsilon	Alpha	Epsilon
Loading 3	Theta	Phi	Horizontal	Vertical
Circular			Left	Right
Start			Crosspolar	Coplanar

**Farfield Settings:**

Type: Farfield

Approximation: enabled (kR >> 1)

Monitor: Farfield (f=2000) [1]

Component: Abs

Output: Directivity

Frequency: 2000

Rad. effic.: -0.05585 dB

Tot. effic.: -0.2208 dB

Dir.: 10.87 dB

CST - COMPUTER SIMULATION TECHNOLOGY | www.cst.com

Figura 68. Farfield definiciones de CST

## 4. Modelos simulados

### 4.1. Modelo 1: Fractal orden 1 con grafeno y sustrato de silicio

#### 4.1.1. Componente

Nombre proyecto: sierpinski\_1\_grafeno\_silicon\_pec.cst

Orden fractal: 1

Materiales:

Parche grafeno:  $\mu_c = 0.1$  eV

Sustrato silicio:  $\epsilon_r = 11.9$

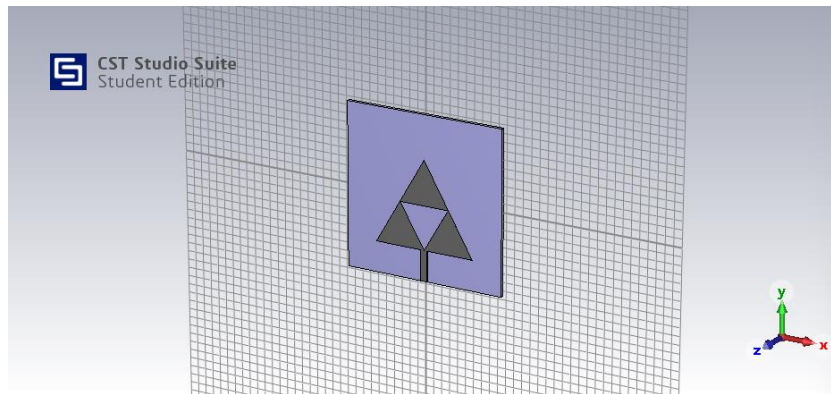


Figura 69. Modelo 1: Componente

#### 4.1.2. Coeficiente de Reflexión $S_{11}$

Frecuencias (THz)		$S_{11}$ (dB)
Fr1	44.9	-6.11
Fr2	196.6	-2.23
Fr3	422.2	-10.87

Tabla 1. Modelo 1 Coeficiente de Reflexión  $S_{11}$

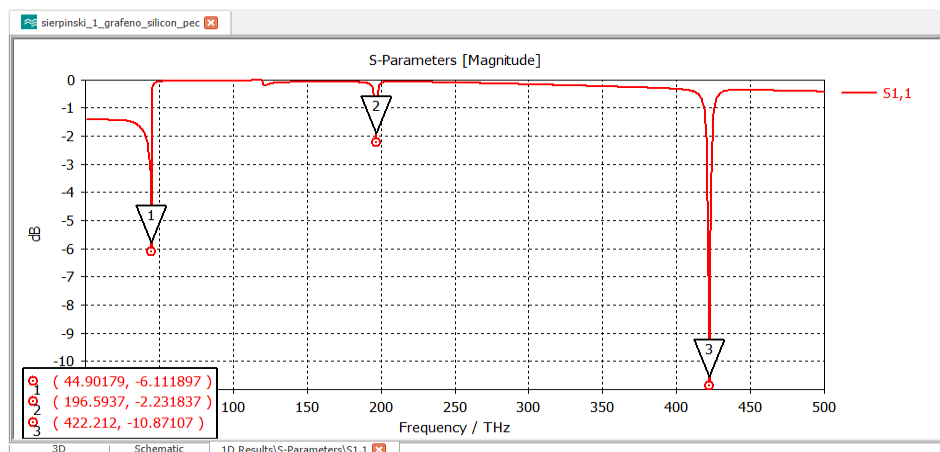


Figura 70. Modelo 1:  $S_{11}$

### 4.1.3. Farfields para las frecuencia de resonancia

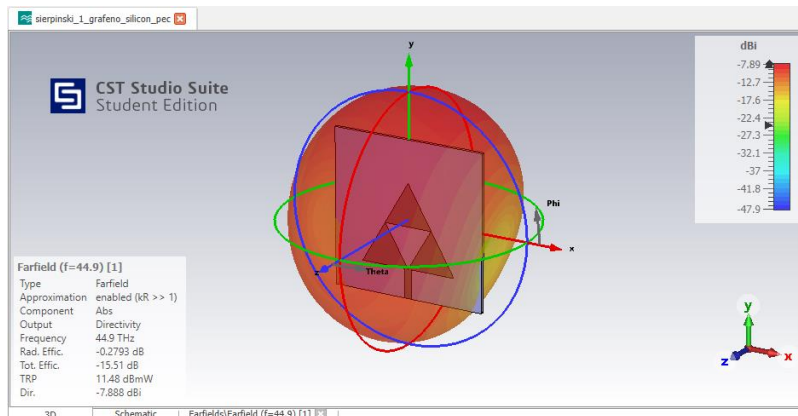


Figura 71. Modelo 1: Diagrama radiación 3D  $f_{r1}$

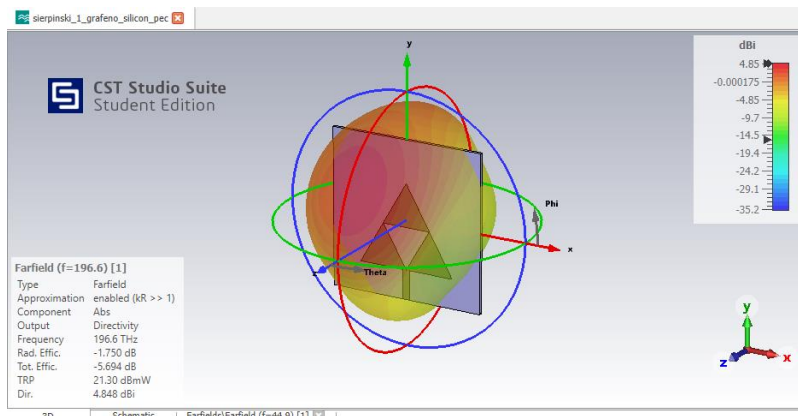


Figura 72. Modelo 1: Diagrama radiación 3D  $f_{r2}$

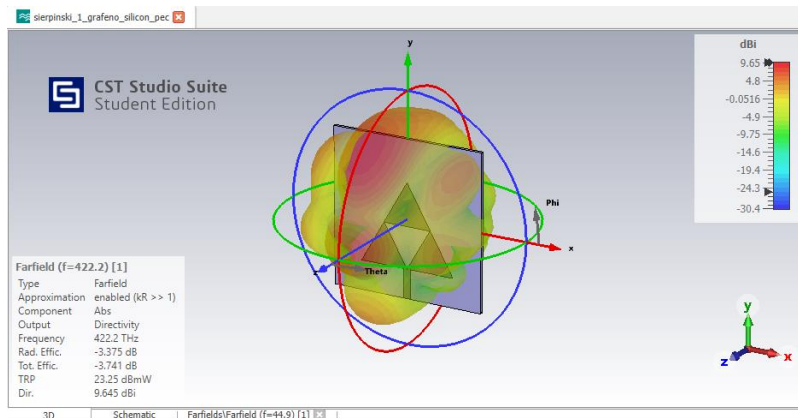


Figura 73. Modelo 1: Diagrama radiación 3D  $f_{r3}$

#### 4.1.4. Directividad

Frecuencias (THz)		Directividad (dBi)
Fr1	44.9	-7.888
Fr2	196.6	4.848
Fr3	422.2	9.645

Tabla 2. Modelo 1: Directividad

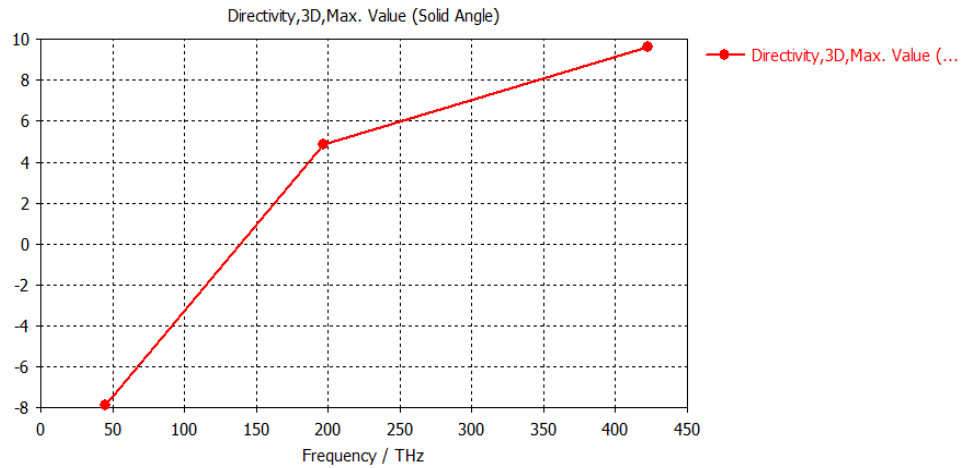


Figura 74. Modelo 1: Directividad

#### 4.1.5. Eficiencias de Radiación

Frecuencias (THz)		Rad.Effic (dB)	Rad.Effic (%)	Tot.Effic (dB)	Tot.Effic (%)
Fr1	44.9	-0.2793	93.77	-15.51	2.82
Fr2	196.6	-1.750	66.83	-5.694	26.95
Fr3	422.2	-3.375	45.97	-3.741	42.26

Tabla 3. Modelo 1: Eficiencias de Radiación

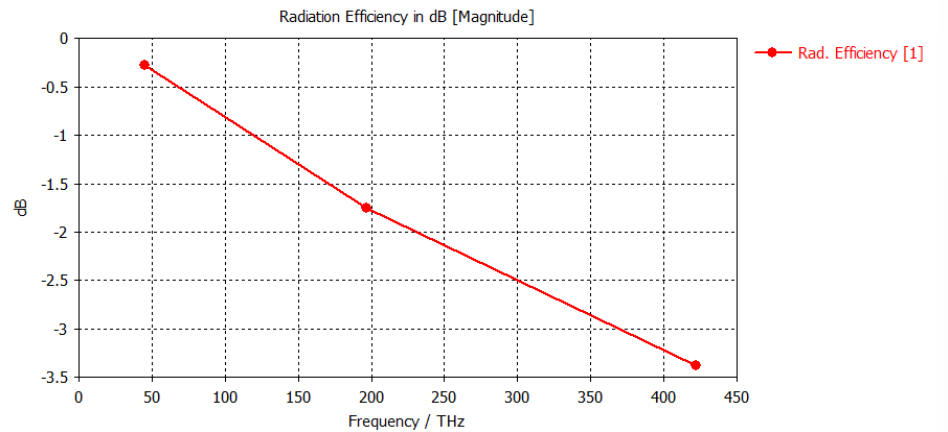


Figura 75. Modelo 1: Eficiencia de Radiación

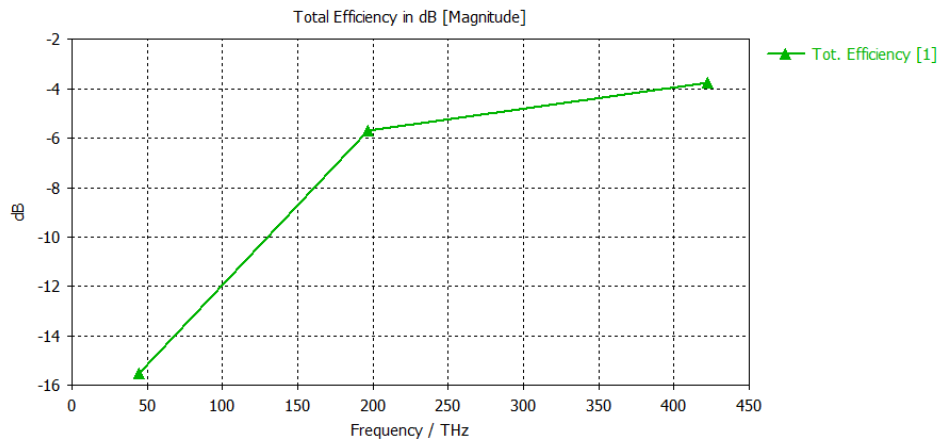


Figura 76. Modelo 1: Eficiencia Total

#### 4.1.6. Ganancia

Frecuencias (THz)		Gain (dB)
Fr1	44.9	-8.130
Fr2	196.6	3.059
Fr3	422.2	6.266

Tabla 4. Modelo 1: Ganancia

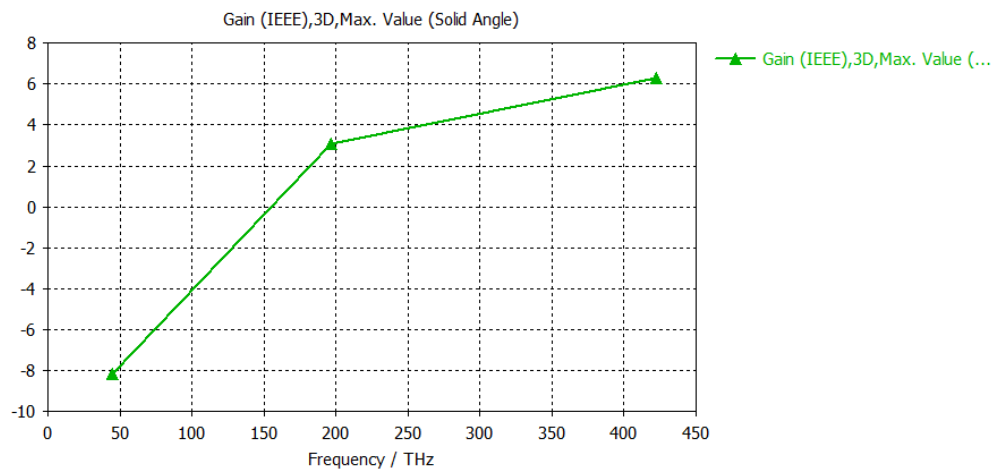


Figura 77. Modelo 1: Ganancia

#### 4.1.7. Ganancia Realizable

Frecuencias (THz)		Gain Realized (dB)
Fr1	44.9	-23.21
Fr2	196.6	-0.805
Fr3	422.2	5.856

Tabla 5. Modelo 1: Ganancia Realizable

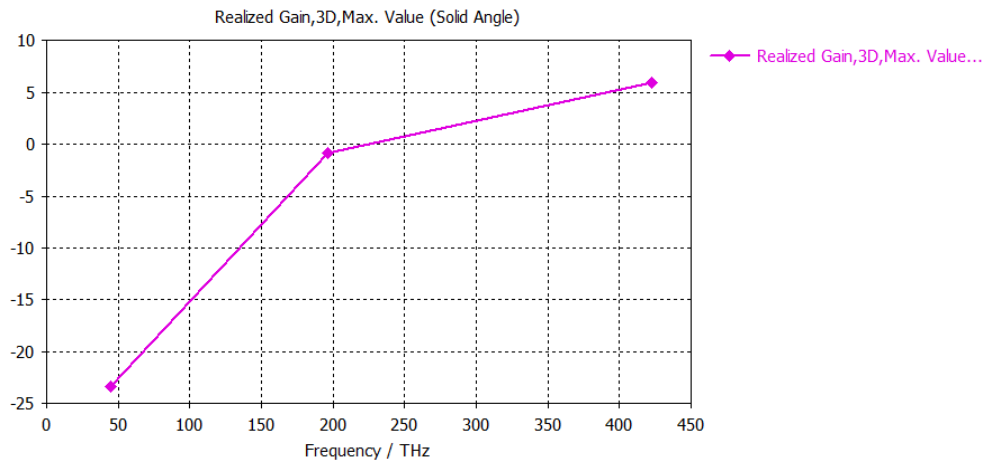


Figura 78. Modelo 1: Ganancia Realizable

#### 4.1.8. Potencia Total Radiada (TRP)

Frecuencias (THz)		TPR (3dBm)	TPR (13dBm)
Fr1	44.9	-12.51	-2.51
Fr2	196.6	-2.694	7.31
Fr3	422.2	-0.741	9.26

Tabla 6. Modelo 1: Potencia Total Radiada

## 4.2. Modelo 2: Fractal orden 1 con grafeno y sustrato fibra de vidrio

### 4.2.1. Componente

Nombre proyecto: sierpinski\_1\_grafeno\_glass\_epoxy\_pec.cst

Orden fractal: 1

Materiales:

Parche grafeno:  $\mu_c = 0.1$  eV

Sustrato glass\_epoxy:  $\epsilon_r = 4.4$

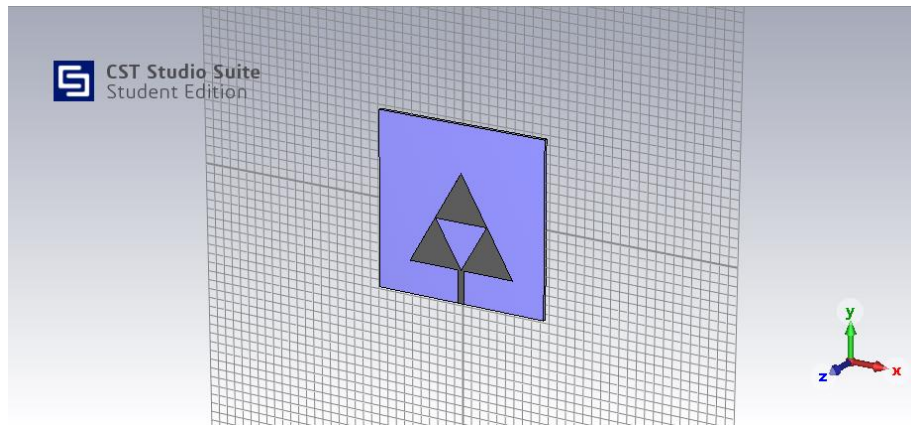


Figura 79. Modelo 2: Componente

### 4.2.2. Coeficiente de Reflexión $S_{11}$

Frecuencias (THz)		$S_{11}$ (dB)
Fr1	74.35	-21.16
Fr2	291.5	-2.99

Tabla 7. Modelo 2: Coeficiente de Reflexión  $S_{11}$

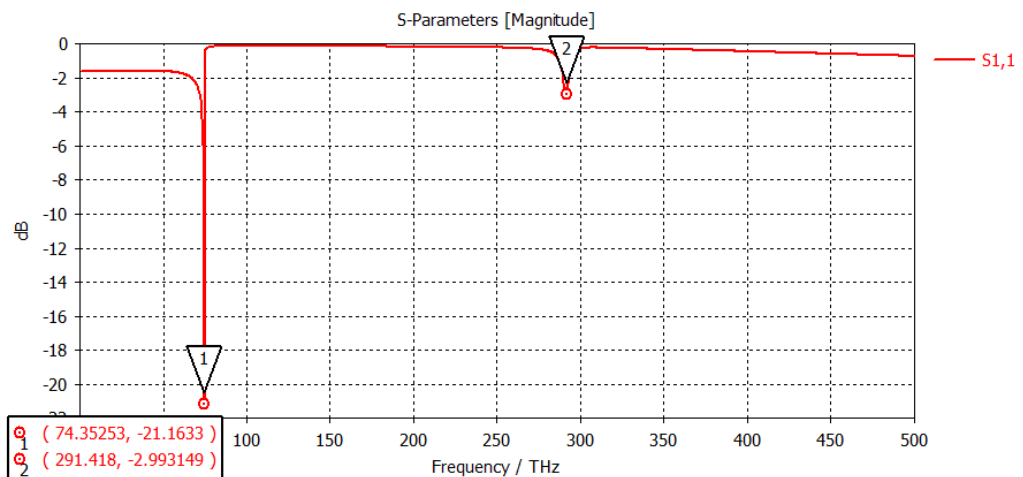


Figura 80. Modelo 2:  $S_{11}$



### 4.2.3. Farfields para las frecuencia de resonancia

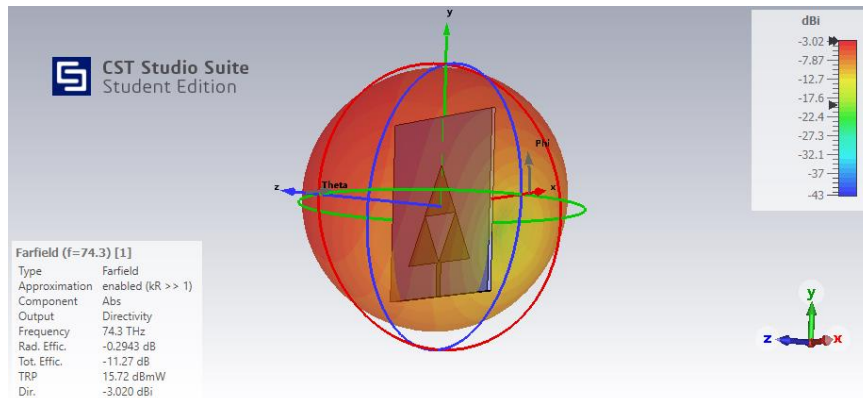


Figura 81. Modelo 2: Diagrama radiación 3D  $f_{r1}$

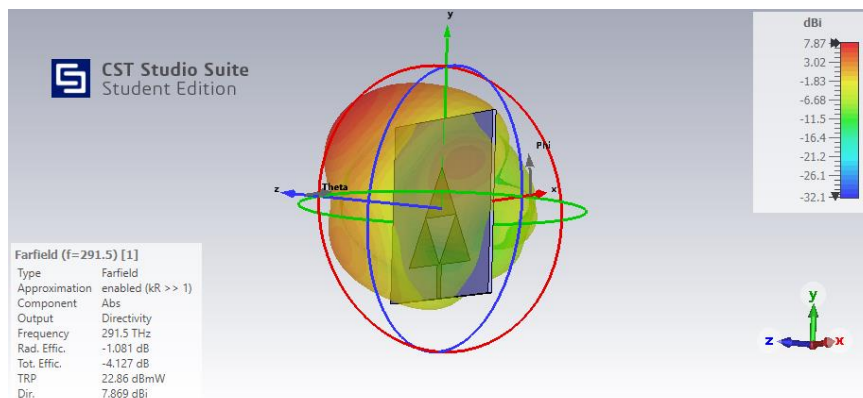


Figura 82. Modelo 2: Diagrama radiación 3D  $f_{r2}$

### 4.2.4. Directividad

Frecuencias (THz)		Directividad (dBi)
Fr1	74.35	-3.02
Fr2	291.5	7.87

Tabla 8. Modelo 2: Directividad

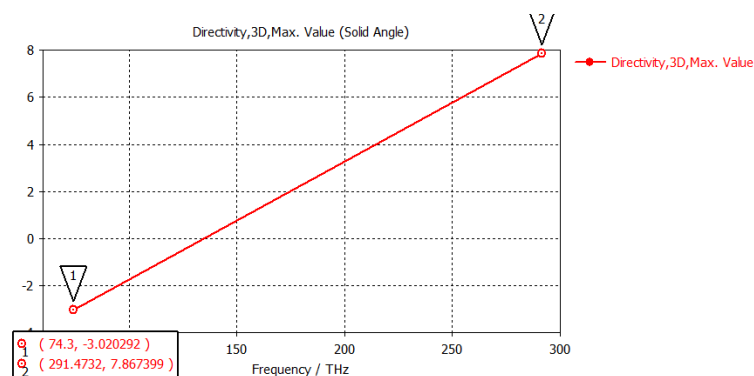


Figura 83. Modelo 2: Directividad

#### 4.2.5. Eficiencias de Radiación

Frecuencias (THz)		Rad.Effic (dB)	Rad.Effic (%)	Tot.Effic (dB)	Tot.Effic (%)
Fr1	74.35	-0.2942	93.45	-11.27	7.46
Fr2	291.5	-0.1081	97.542	-4.127	38.66

Tabla 9. Modelo 2: Eficiencias de Radiación

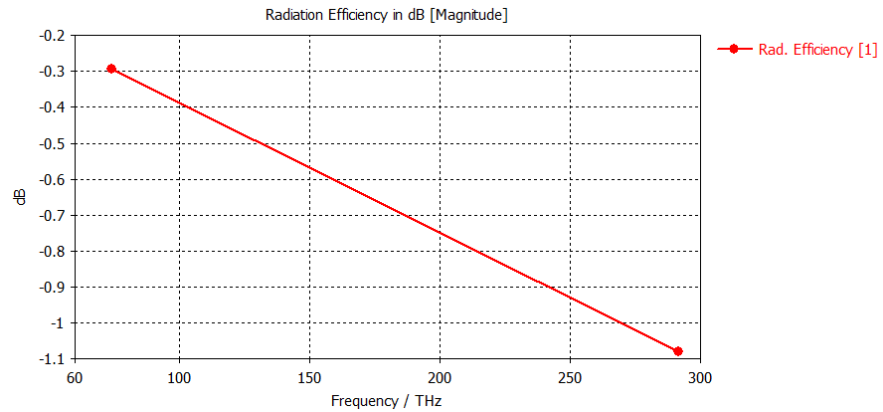


Figura 84. Modelo 2: Eficiencia de Radiación

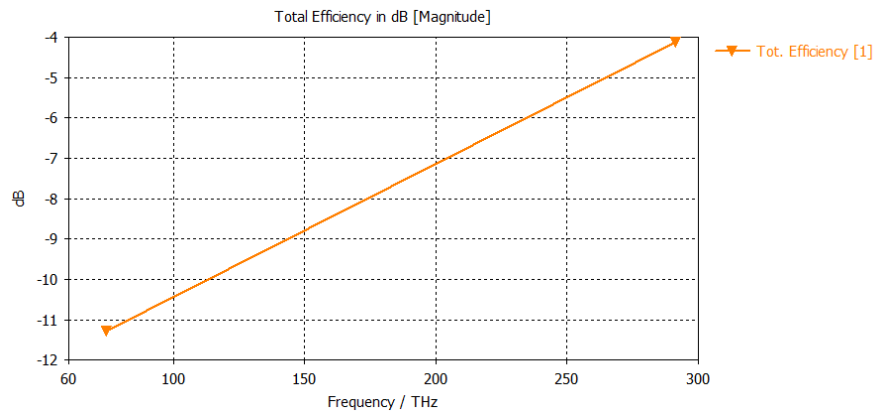


Figura 85. Modelo 2: Eficiencia Total

#### 4.2.6. Ganancia

Frecuencias (THz)		Gain (dB)
Fr1	74.35	-3.314
Fr2	291.5	6.776

Tabla 10. Modelo 2: Ganancia

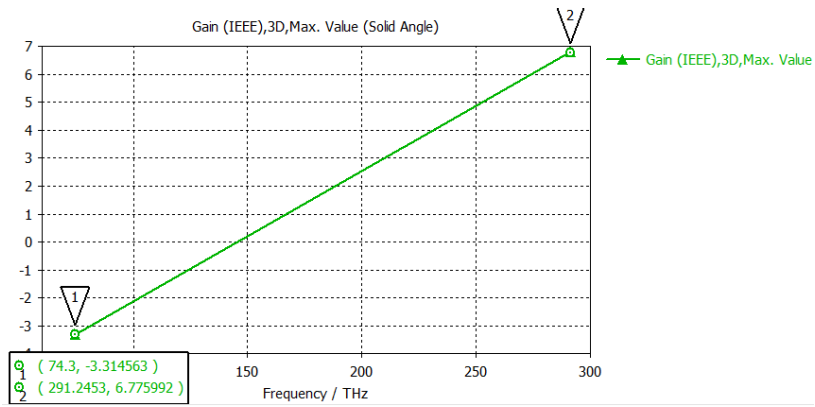


Figura 86. Modelo 2: Ganancia

#### 4.2.7. Ganancia Realizable

Frecuencias (THz)		Gain Realized (dB)
Fr1	74.35	-14.286
Fr2	291.5	3.731

Tabla 11. Modelo 2: Ganancia Realizable

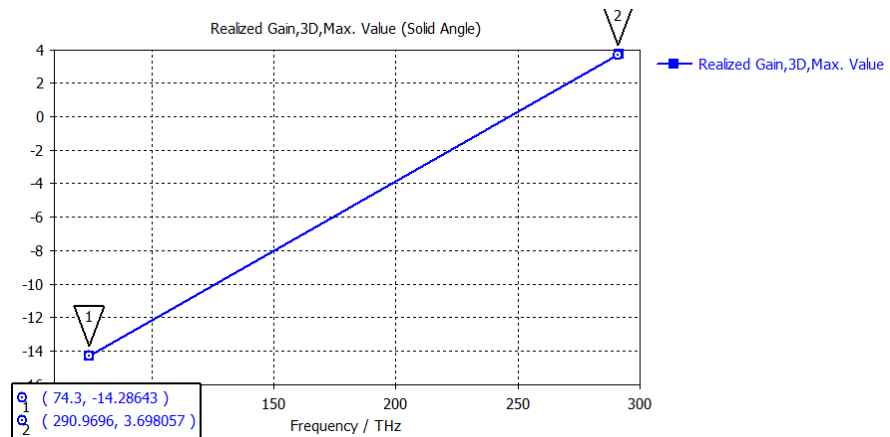


Figura 87. Modelo 2: Ganancia Realizable

#### 4.2.8. Potencia Total Radiada (TRP)

Frecuencias (THz)		TPR (3dBm)	TPR (13dBm)
Fr1	74.35	-8.266	1.734
Fr2	295.1	-1.127	8.873

Tabla 12. Modelo 2: Potencia Total Radiada

### 4.3. Modelo 3: Fractal orden 1 con grafeno y sustrato de silicio

#### 4.3.1. Componente

Nombre proyecto: sierpinski\_1\_grafeno2\_silicon\_pec.cst

Orden fractal: 1

Materiales:

Parche grafeno:  $\mu_c = 0.2 \text{ eV}$

Sustrato silicio:  $\epsilon_r = 11.9$

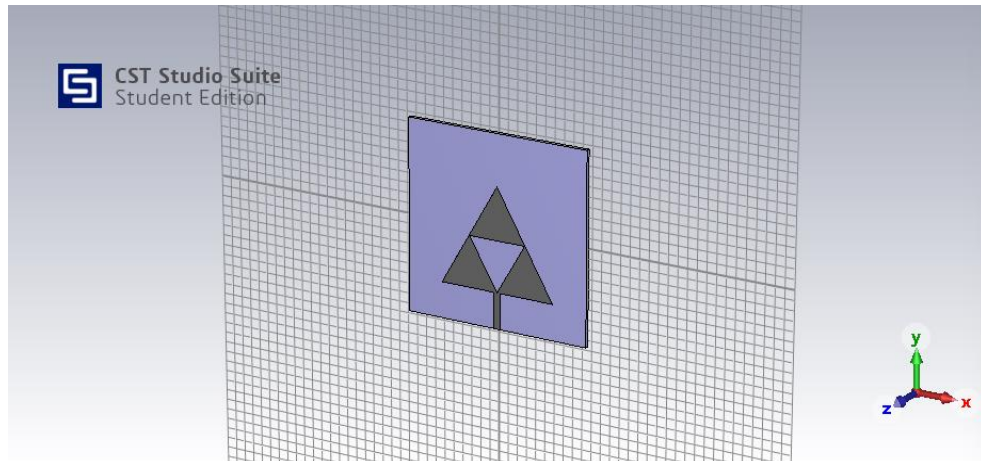


Figura 88. Modelo 3: Componente

#### 4.3.2. Coeficiente de Reflexión $S_{11}$

Frecuencias (THz)		$S_{11}$ (dB)
Fr1	44.9	-6.27
Fr2	196.6	-2.05
Fr3	422.2	-7.64

Tabla 13. Modelo 3: Coeficiente de Reflexión  $S_{11}$

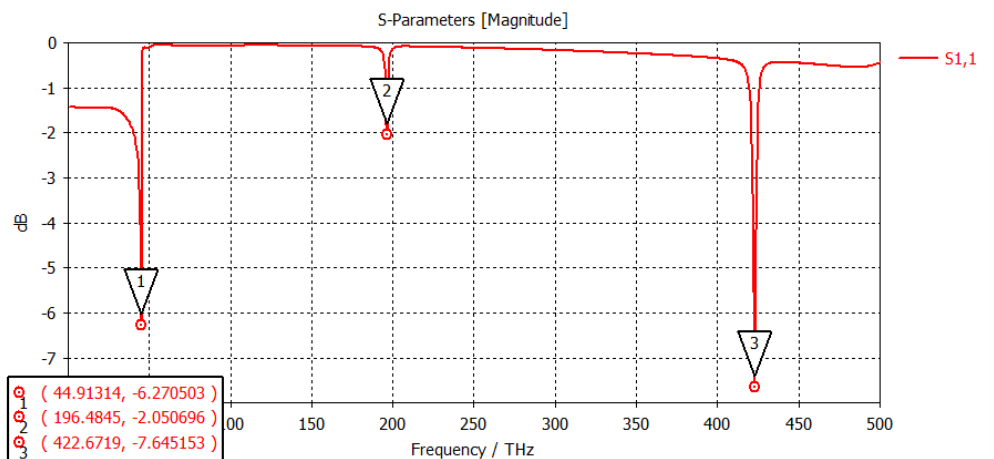


Figura 89. Modelo 3:  $S_{11}$

### 4.3.3. Farfields para las frecuencia de resonancia

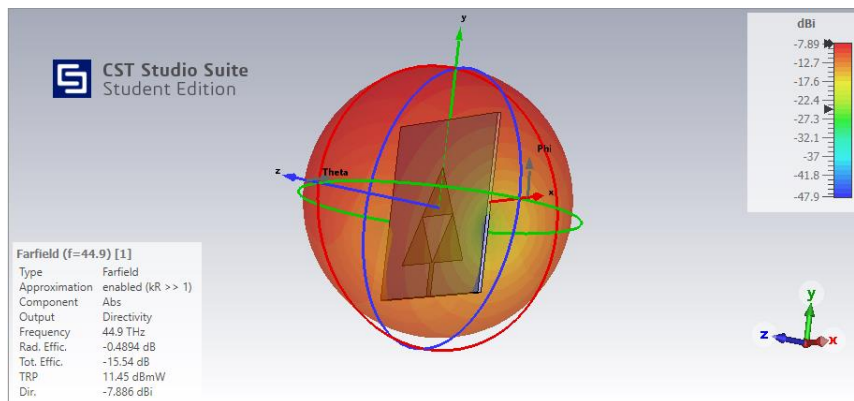


Figura 90. Modelo 3: Diagrama radiación 3D  $f_{r1}$

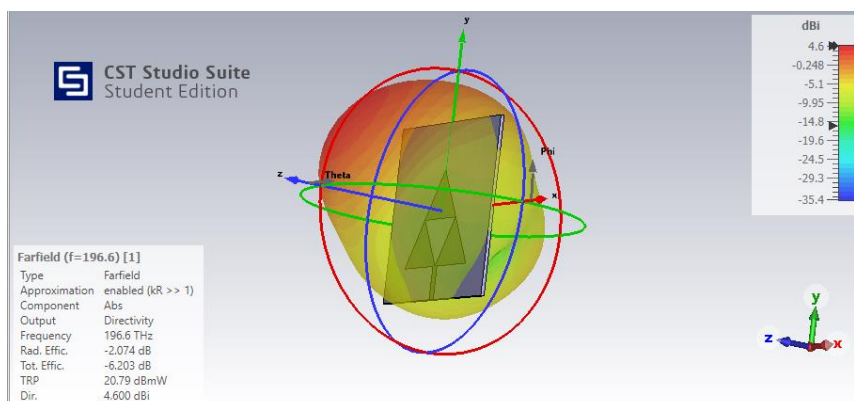


Figura 91. Modelo 3: Diagrama radiación 3D  $f_{r2}$

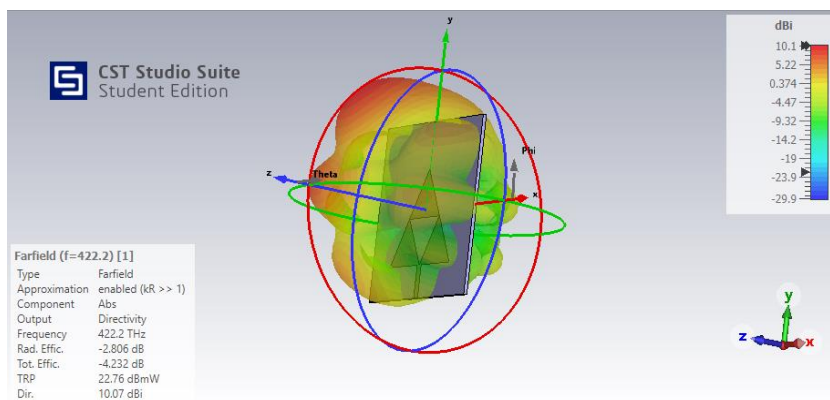


Figura 92. Modelo 3: Diagrama radiación 3D  $f_{r3}$

### 4.3.4. Directividad

Frecuencias (THz)		Directividad (dBi)
Fr1	44.9	-7.886
Fr2	196.6	4.600
Fr3	422.2	10.07

Tabla 14. Modelo 3: Directividad

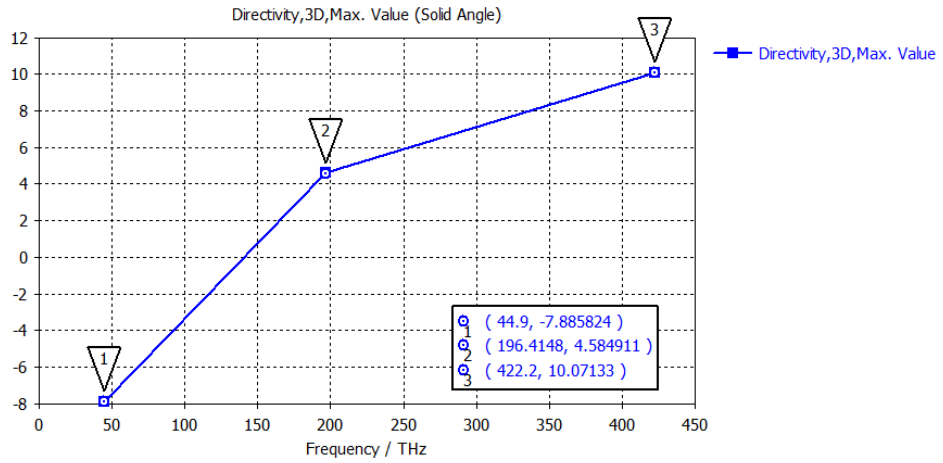


Figura 93. Modelo 3: Directividad

#### 4.3.5. Eficiencias de Radiación

Frecuencias (THz)		Rad.Effic (dB)	Rad.Effic (%)	Tot.Effic (dB)	Tot.Effic (%)
Fr1	44.9	-0.4894	89.34	-15.54	2.8
Fr2	196.6	-2.074	95.34	-6.203	23.97
Fr3	422.2	-2.806	52.41	-4.232	37.74

Tabla 15. Modelo 3: Eficiencias de Radiación

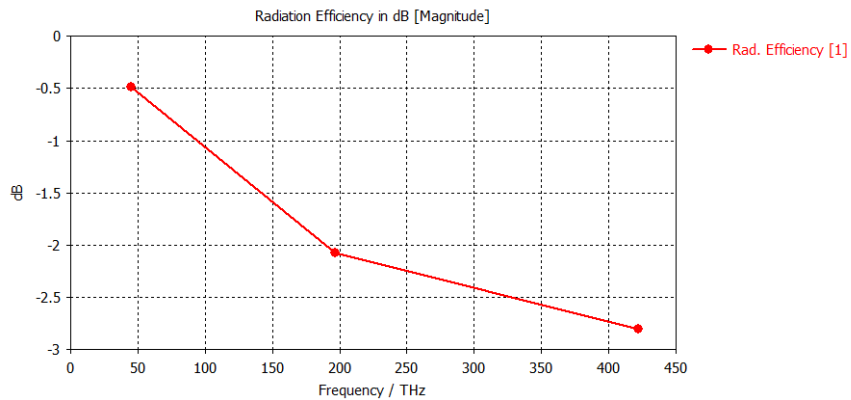


Figura 94. Modelo 3: Eficiencia de Radiación

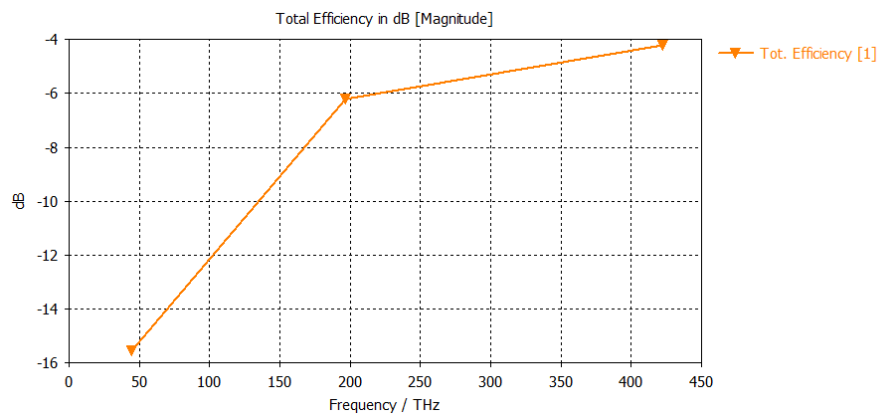


Figura 95. Modelo 3: Eficiencia Total

### 4.3.6. Ganancia

Frecuencias (THz)		Gain (dB)
Fr1	44.9	-8.300
Fr2	196.6	2.540
Fr3	422.2	7.239

Tabla 16. Modelo 3: Ganancia

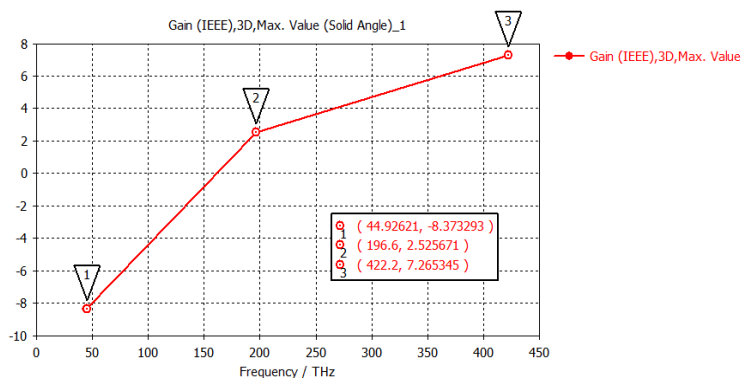


Figura 96. Modelo 3: Ganancia

### 4.3.7. Ganancia Realizable

Frecuencias (THz)		Gain Realized (dB)
Fr1	44.9	-23.280
Fr2	196.6	-1.591
Fr3	422.2	5.799

Tabla 17. Modelo 3: Ganancia Realizable

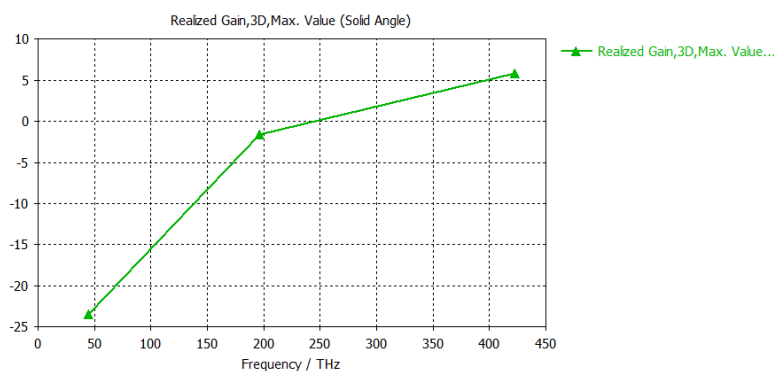


Figura 97. Modelo 3: Ganancia Realizable

### 4.3.8. Potencia Total Radiada (TRP)

Frecuencias (THz)		TPR (3dBm)	TPR (13dBm)
Fr1	44.9	-12.54	-2.536
Fr2	196.6	-3.203	6.797
Fr3	422.2	-1.232	8.768

Tabla 18. Modelo 3: Potencia Total Radiada

#### 4.4. Modelo 4: Fractal orden 1 con grafeno 2 y sustrato fibra de vidrio

##### 4.4.1. Componente

Nombre proyecto: sierpinski\_1\_grafeno2\_glass\_epoxy\_pec.cst

Orden fractal: 1

Materiales:

Parche grafeno:  $\mu_c = 0.2$  eV

Sustrato glass\_epoxy:  $\epsilon_r = 4.4$

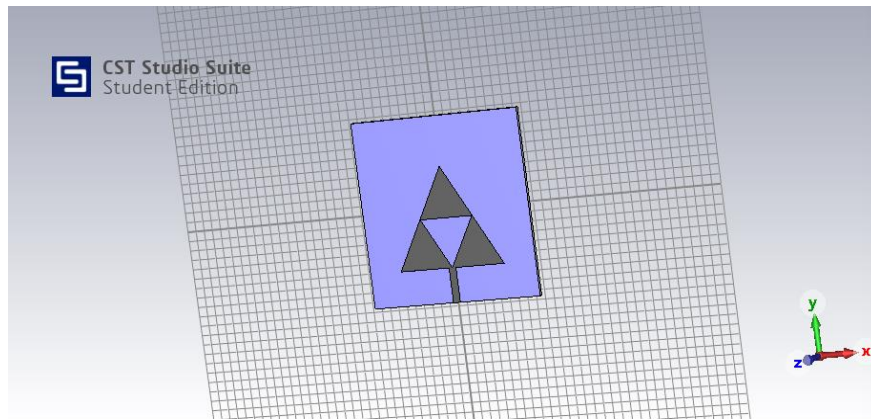


Figura 98. Modelo 4: Componente

##### 4.4.2. Coeficiente de Reflexión $S_{11}$

Frecuencias (THz)		$S_{11}$ (dB)
Fr1	74.35	-21.36
Fr2	292.3	-2.56

Tabla 19. Modelo 4: Coeficiente de Reflexión  $S_{11}$

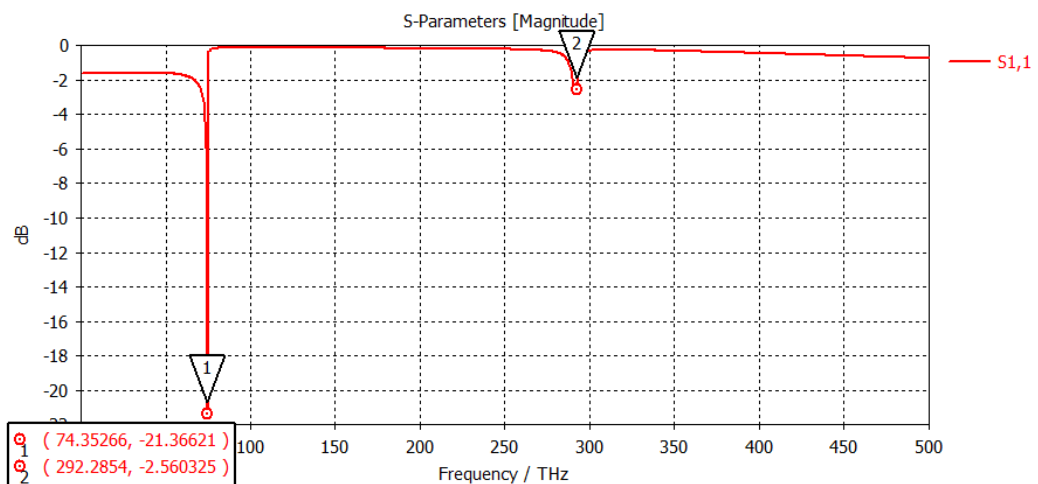


Figura 99. Modelo 4:  $S_{11}$



### 4.4.3. Farfields para las frecuencia de resonancia

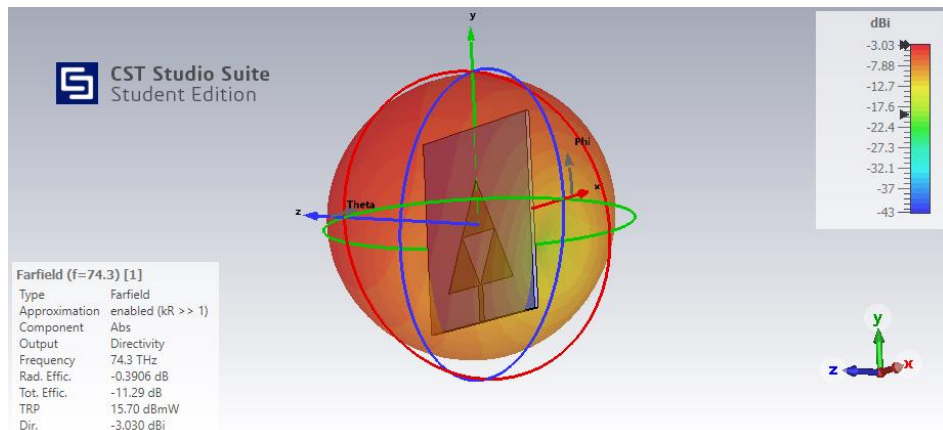


Figura 100. Modelo 4: Diagrama radiación 3D  $f_{r1}$

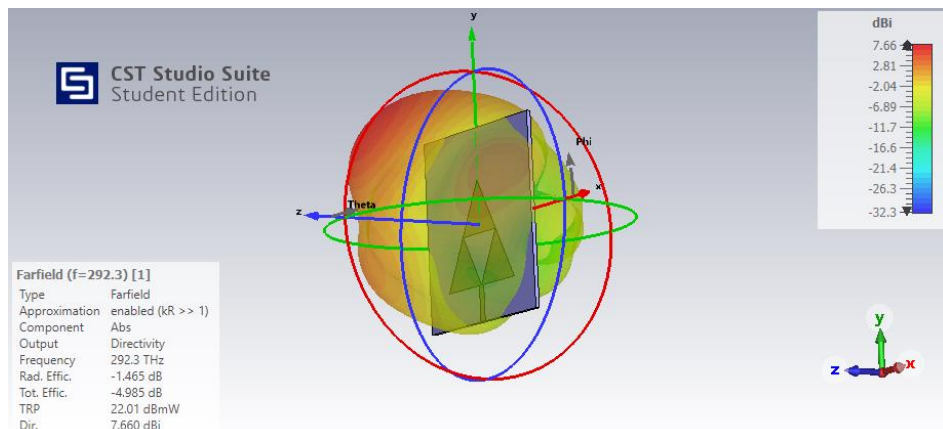


Figura 101. Modelo 4: Diagrama radiación 3D  $f_{r2}$

### 4.4.4. Directividad

Frecuencias (THz)		Directividad (dBi)
Fr1	74.35	-3.029
Fr2	292.3	7.647

Tabla 20. Modelo 4: Directividad

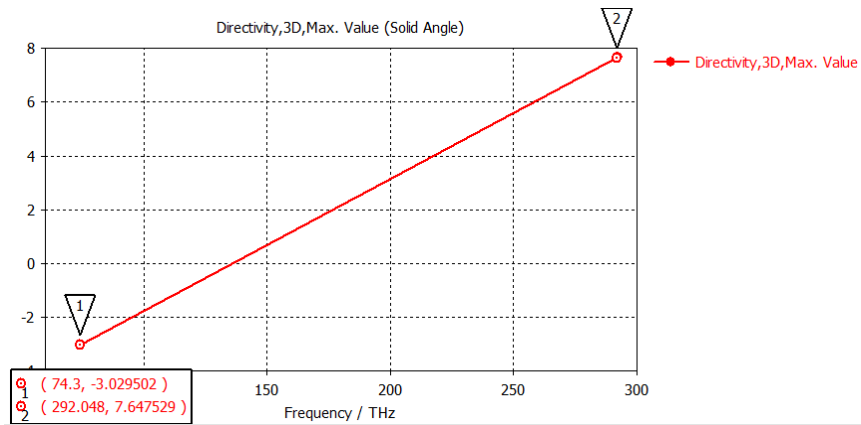


Figura 102. Modelo 4: Directividad

#### 4.4.5. Eficiencias de Radiación

Frecuencias (THz)		Rad.Effic (dB)	Rad.Effic (%)	Tot.Effic (dB)	Tot.Effic (%)
Fr1	74.35	-0.3906	91.40	-11.29	7.43
Fr2	292.3	-1.465	71.37	-4.985	31.73

Tabla 21. Modelo 4: Eficiencias de Radiación

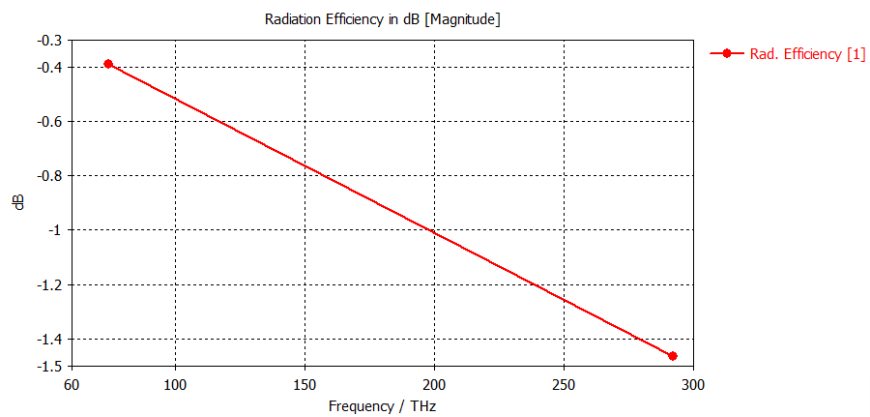


Figura 103. Modelo 4: Eficiencia de Radiación

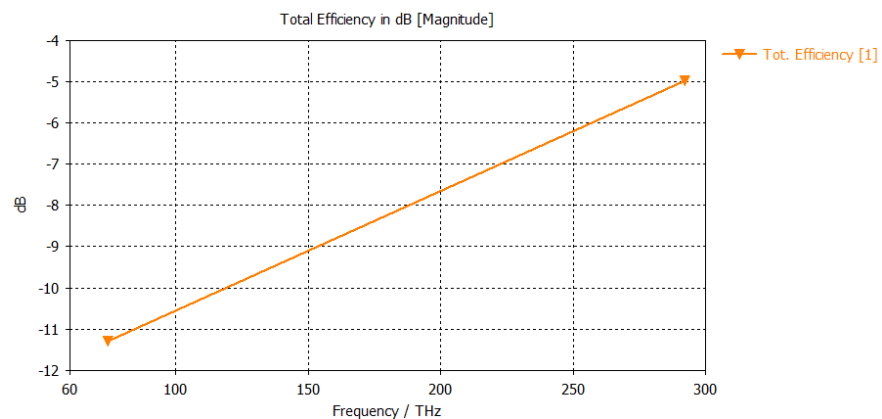


Figura 104. Modelo 4: Eficiencia Total

#### 4.4.6. Ganancia

Frecuencias (THz)		Gain (dB)
Fr1	74.35	-3.342
Fr2	292.3	6.195

Tabla 22. Modelo 4: Ganancia

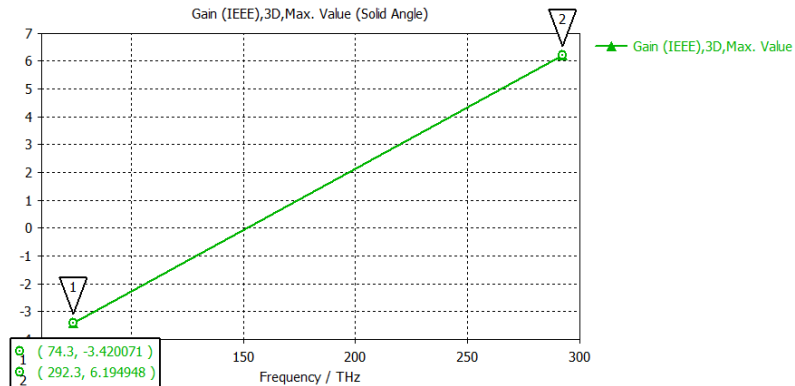


Figura 105. Modelo 4: Ganancia

#### 4.4.7. Ganancia Realizable

Frecuencias (THz)		Gain Realized (dB)
Fr1	74.35	-14.296
Fr2	292.3	2.675

Tabla 23. Modelo 4: Ganancia Realizable

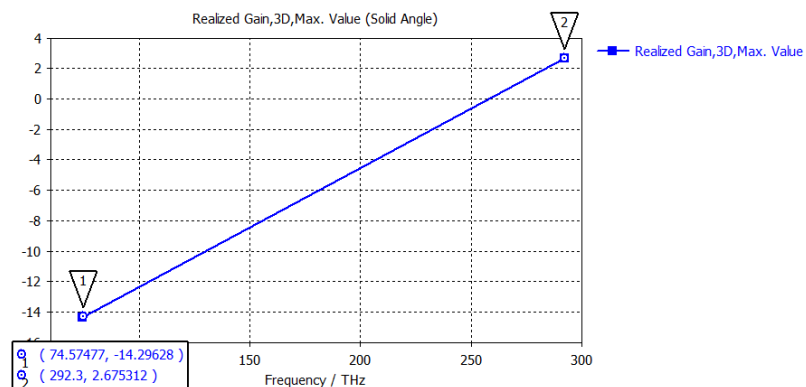


Figura 106. Modelo 4: Ganancia Realizable

#### 4.4.8. Potencia Total Radiada (TRP)

Frecuencias (THz)		TPR (3dBm)	TPR (13dBm)
Fr1	74.35	-8.288	1.712
Fr2	292.3	-1.985	8.015

Tabla 24. Modelo 4: Potencia Total Radiada

## 4.5. Modelo 5: Fractal orden 2 con grafeno y sustrato silicio

### 4.5.1. Componente

Nombre proyecto: sierpinski\_2\_grafeno\_silicon\_pec.cst

Orden fractal: 2

Materiales:

Parche grafeno:  $\mu_c = 0.1$  eV

Sustrato silicio:  $\epsilon_r = 11.9$

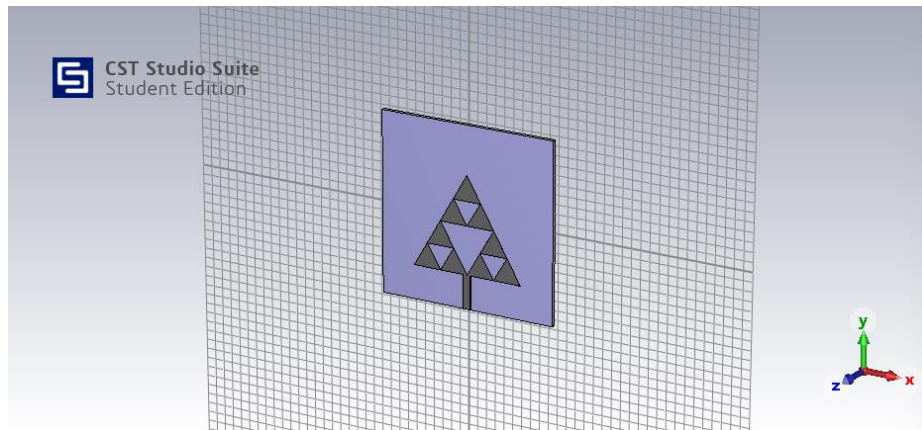


Figura 107. Modelo 5: Componente

### 4.5.2. Coeficiente de Reflexión S11

Frecuencias (THz)		S11 (dB)
Fr1	44.9	-7.45
Fr2	191.6	-3.87
Fr3	250.3	-0.33

Tabla 25. Modelo 5: Coeficiente de Reflexión S<sub>11</sub>

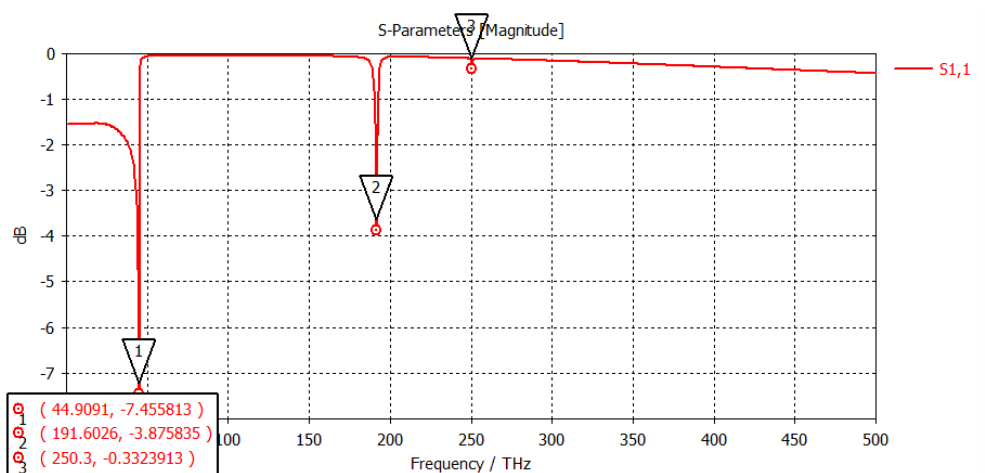


Figura 108. Modelo 5: S<sub>11</sub>

### 4.5.3. Farfields para las frecuencia de resonancia

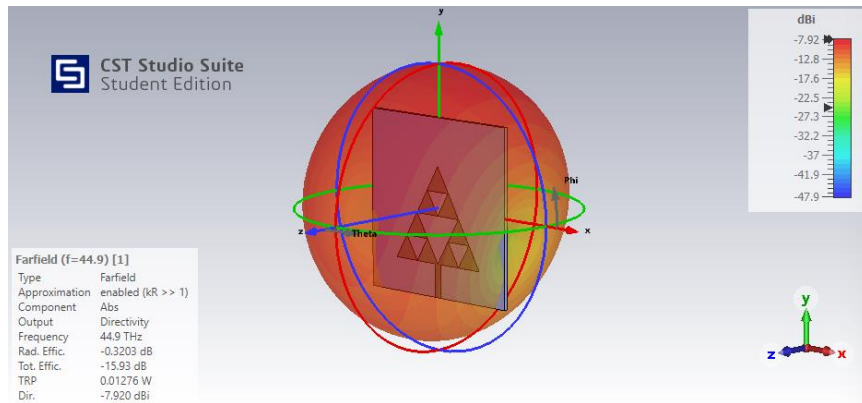


Figura 109. Modelo 5: Diagrama radiación 3D  $f_{r1}$

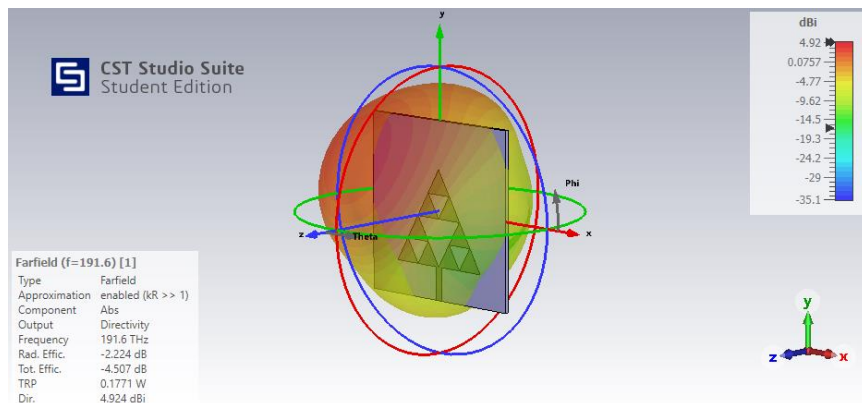


Figura 110. Modelo 5: Diagrama radiación 3D  $f_{r2}$

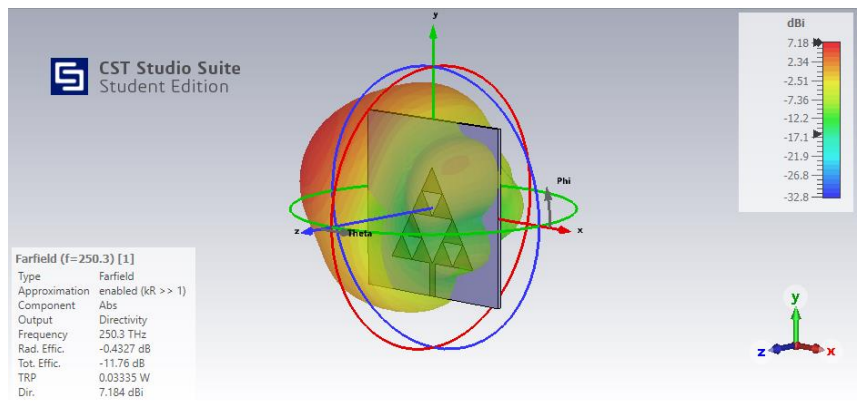


Figura 111. Modelo 5: Diagrama radiación 3D  $f_{r3}$

### 4.5.4. Directividad

Frecuencias (THz)		Directividad (dBi)
Fr1	44.9	-7.920
Fr2	191.6	4.924
Fr3	250.3	7.184

Tabla 26. Modelo 5: Directividad

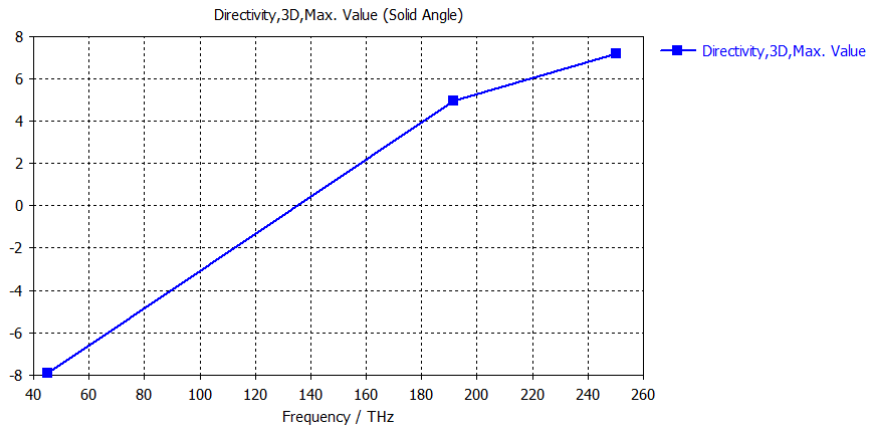


Figura 112. Modelo 5: Directividad

#### 4.5.5. Eficiencias de Radiación

Frecuencias (THz)		Rad.Effic (dB)	Rad.Effic (%)	Tot.Effic (dB)	Tot.Effic (%)
Fr1	44.9	-0.3203	92.89	-15.93	2.553
Fr2	191.6	-2.224	59.924	-4.507	35.424
Fr3	250.3	-0.4327	90.517	-11.76	6.668

Tabla 27. Modelo 5: Eficiencias de Radiación

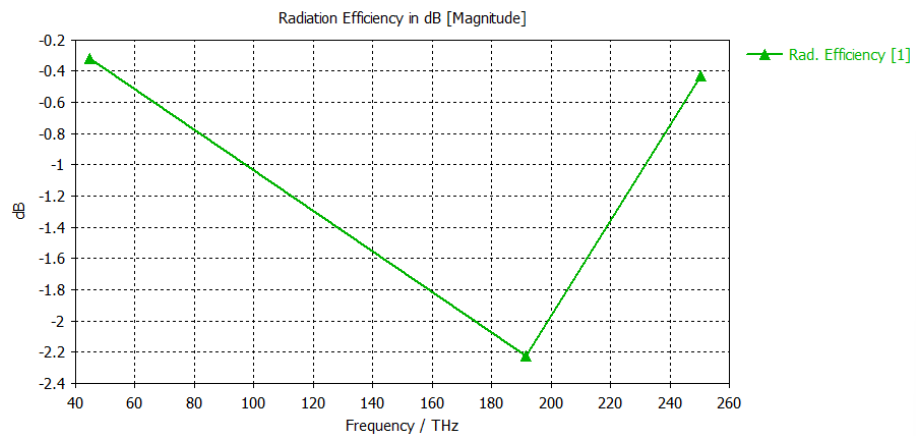


Figura 113. Modelo 5: Eficiencia de Radiación

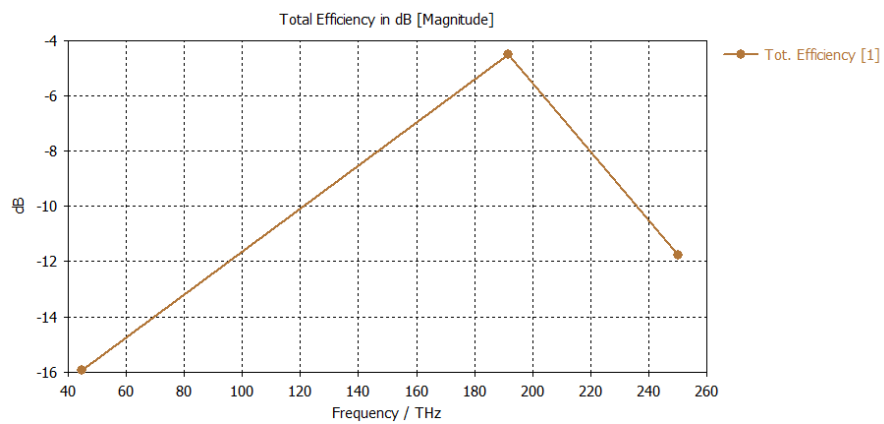


Figura 114. Modelo 5: Eficiencia Total

#### 4.5.6. Ganancia

Frecuencias (THz)		Gain (dB)
Fr1	44.9	-8.236
Fr2	191.6	2.702
Fr3	250.3	6.714

Tabla 28. Modelo 5: Ganancia

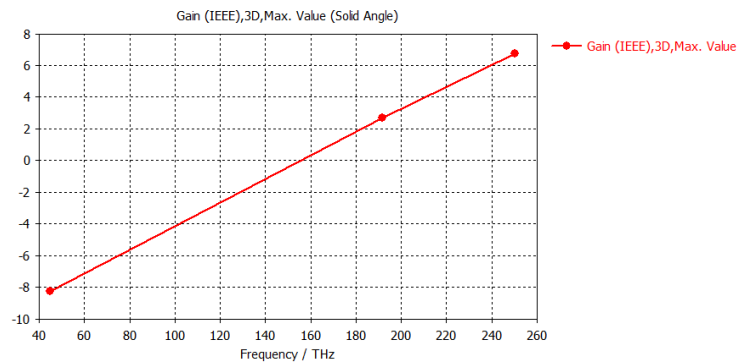


Figura 115. Modelo 5: Ganancia

#### 4.5.7. Ganancia Realizable

Frecuencias (THz)		Gain Realized (dB)
Fr1	44.9	-23.737
Fr2	191.6	0.408
Fr3	250.3	-4.529

Tabla 29. Modelo 5: Ganancia Realizable

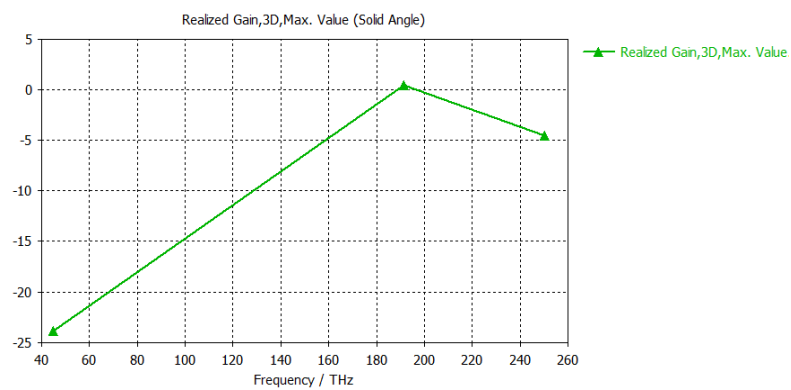


Figura 116. Modelo 5: Ganancia Realizable

#### 4.5.8. Potencia Total Radiada (TRP)

Frecuencias (THz)		TPR (3dBm)	TPR (13dBm)
Fr1	44.9	-12.93	-2.932
Fr2	191.6	-1.507	8.493
Fr3	250.3	-8.759	1.241

Tabla 30. Modelo 5: Potencia Total Radiada

## 4.6. Modelo 6: Fractal orden 2 con grafeno y sustrato fibra de vidrio

### 4.6.1. Componente

Nombre proyecto: sierpinski\_2\_grafeno\_glass\_epoxy\_pec.cst

Orden fractal: 2

Materiales:

Parche grafeno:  $\mu_c = 0.1 \text{ eV}$

Sustrato glass\_epoxy:  $\epsilon_r = 4.4$

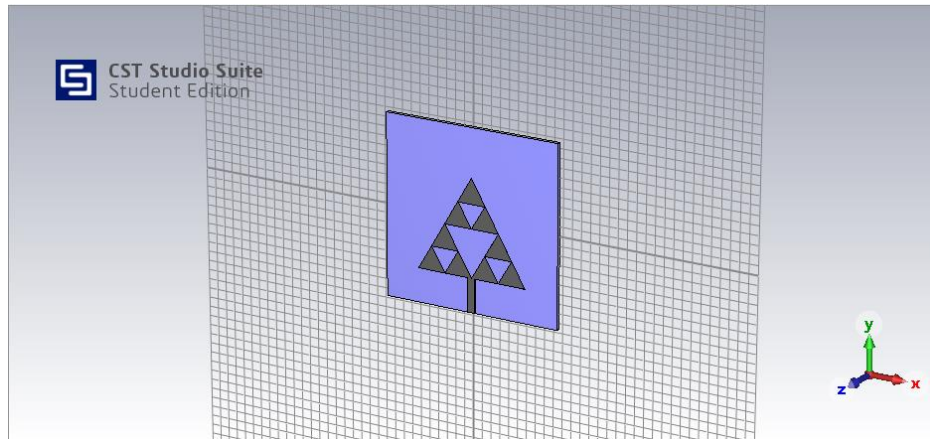


Figura 117. Modelo 6: Componente

### 4.6.2. Coeficiente de Reflexión $S_{11}$

Frecuencias (THz)		$S_{11}$ (dB)
Fr1	74.35	-20.58
Fr2	287.95	-3.61

Tabla 31. Modelo 6: Coeficiente de Reflexión  $S_{11}$

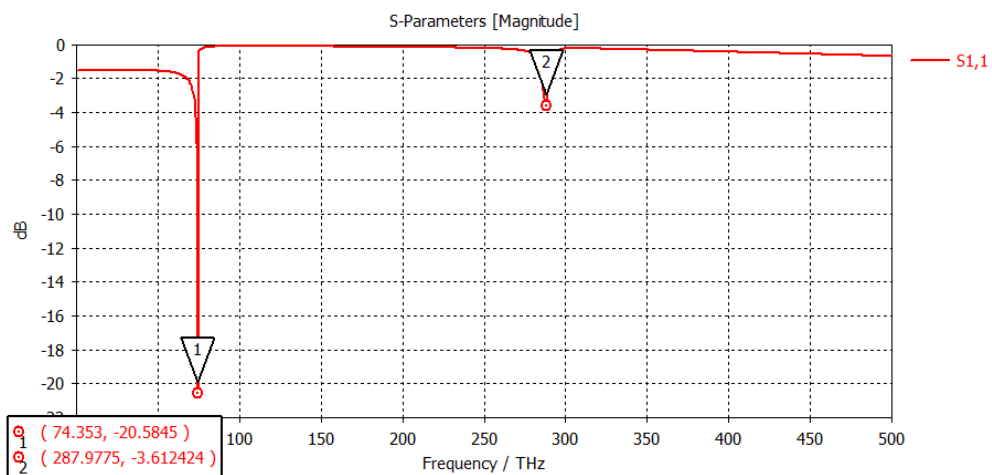


Figura 118. Modelo 6:  $S_{11}$



### 4.6.3. Farfields para las frecuencia de resonancia

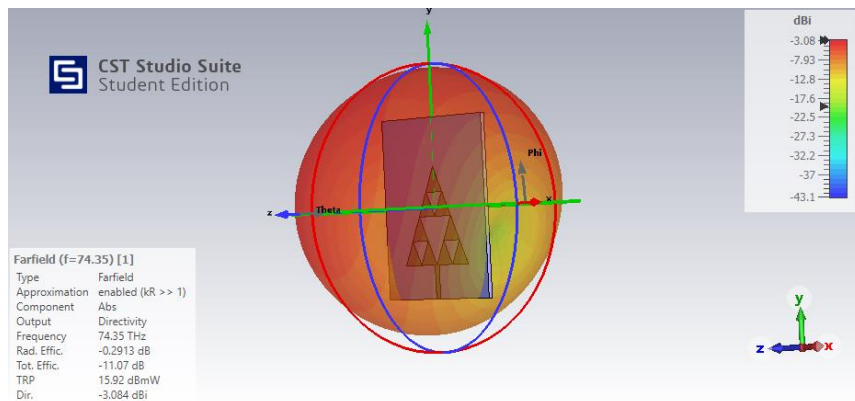


Figura 119. Modelo 6: Diagrama de radiación 3D  $f_{r1}$

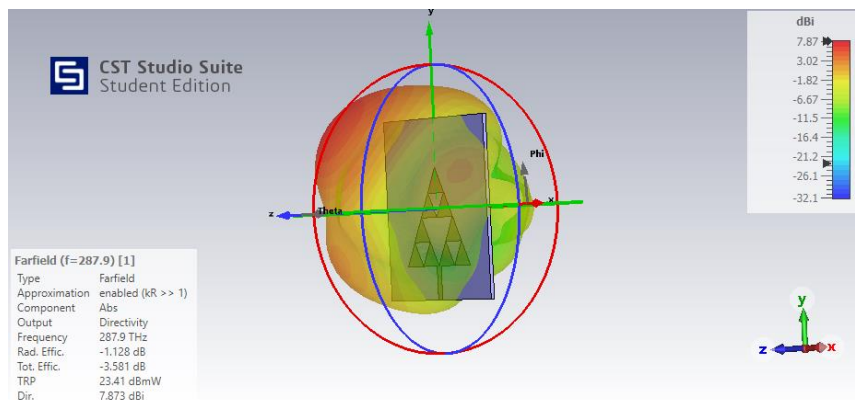


Figura 120. Modelo 6: Diagrama de radiación 3D  $f_{r2}$

### 4.6.4. Directividad

Frecuencias (THz)		Directividad (dBi)
Fr1	74,35	-3,084
Fr2	287,95	7,873

Tabla 32. Modelo 6: Directividad

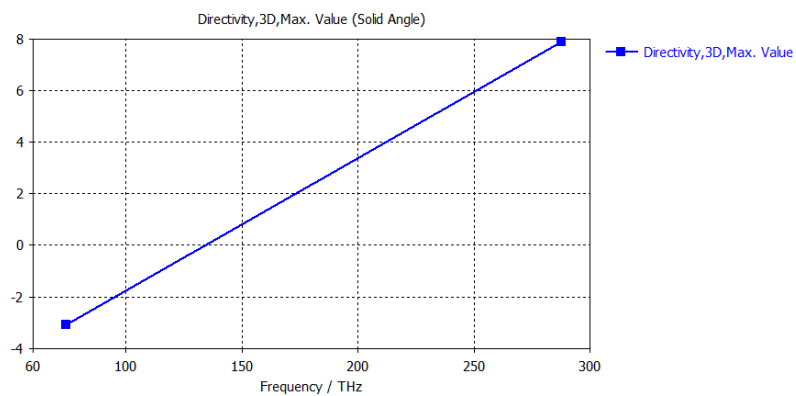


Figura 121. Modelo 6: Directividad

#### 4.6.5. Eficiencias de Radiación

Frecuencias (THz)		Rad.Effic (dB)	Rad.Effic (%)	Tot.Effic (dB)	Tot.Effic (%)
Fr1	74,35	-0,2913	93,513	-11,070	7,816
Fr2	287,95	-1,128	77,126	-3,581	43,843

Tabla 33. Modelo 6: Eficiencias de Radiación

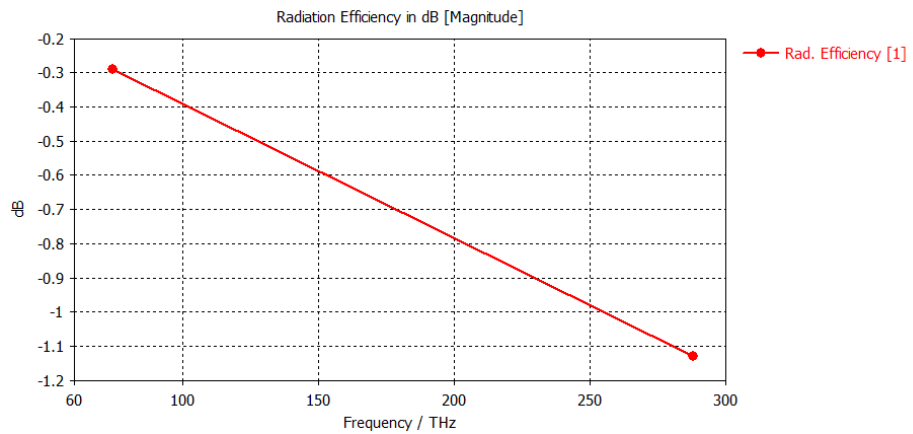


Figura 122. Modelo 6: Eficiencia de Radiación

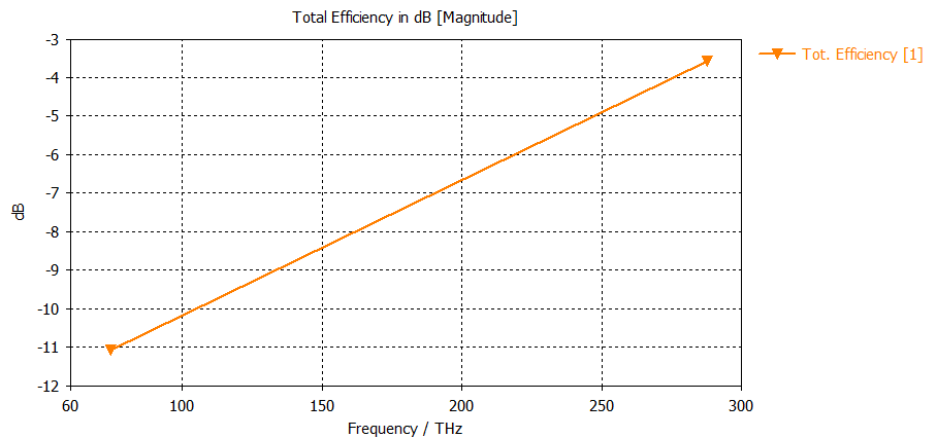


Figura 123. Modelo 6: Eficiencia Total

#### 4.6.6. Ganancia

Frecuencias (THz)		Gain (dB)
Fr1	74,35	-3,370
Fr2	287,95	6,744

Tabla 34. Modelo 6: Ganancia

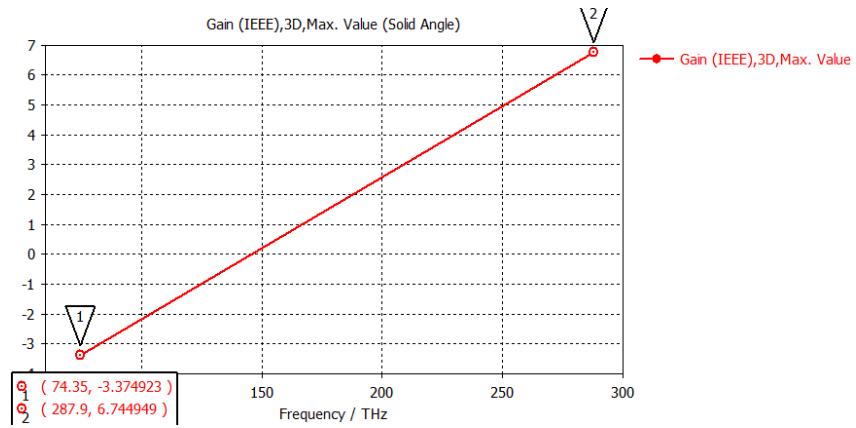


Figura 124. Modelo 6: Ganancia

#### 4.6.7. Ganancia Realizable

Frecuencias (THz)		Gain Realized (dB)
Fr1	74,35	-14,156
Fr2	287,95	4,232

Tabla 35. Modelo 6: Ganancia Realizable

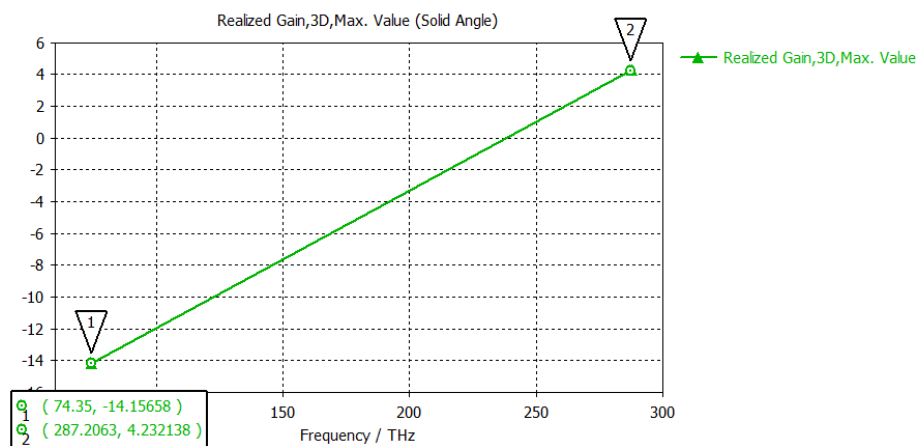


Figura 125. Modelo 6: Ganancia Realizable

#### 4.6.8. Potencia Total Radiada (TRP)

Frecuencias (THz)		TPR (3dBm)	TPR (13dBm)
Fr1	74,35	-8,073	1,927
Fr2	287,95	-0,581	9,419

Tabla 36. Modelo 6: Potencia Total Radiada

## 4.7. Modelo 7: Fractal orden 2 con grafeno 2 y sustrato silicio

### 4.7.1. Componente

Nombre proyecto: sierpinski\_2\_grafeno2\_silicon\_pec.cst

Orden fractal: 2

Materiales:

Parche grafeno:  $\mu_c = 0.2 \text{ eV}$

Sustrato silicio:  $\epsilon_r = 11.9$

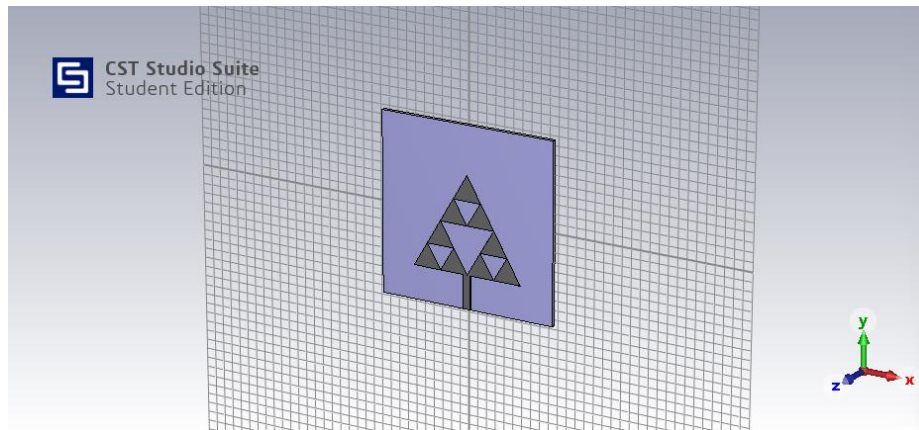


Figura 126. Modelo 7: Componente

### 4.7.2. Coeficiente de Reflexión $S_{11}$

Frecuencias (THz)		$S_{11}$ (dB)
Fr1	44,9	-7,48
Fr2	191,6	-3,85
Fr3	250,2	-0,29

Tabla 37. Modelo 7: Coeficiente de Reflexión  $S_{11}$

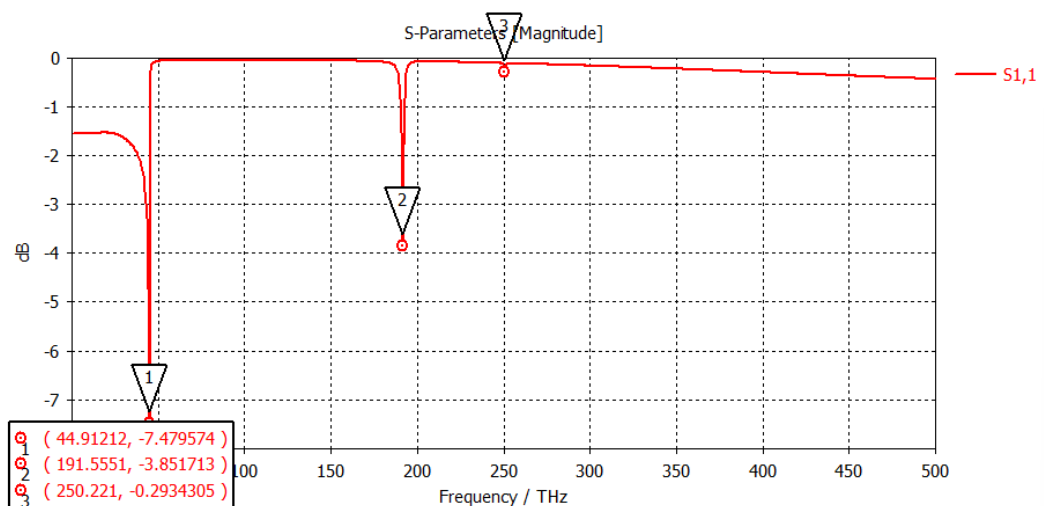


Figura 127. Modelo 7:  $S_{11}$

### 4.7.3. Farfields para las frecuencia de resonancia

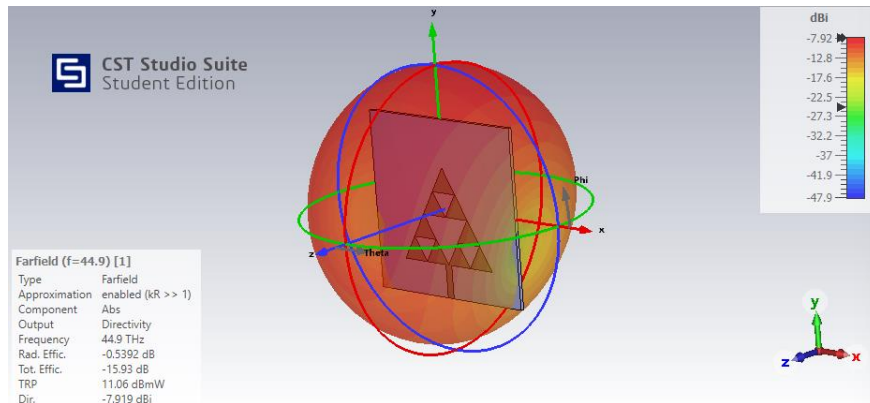


Figura 128. Modelo 7: Diagrama radiación 3D  $f_{r1}$

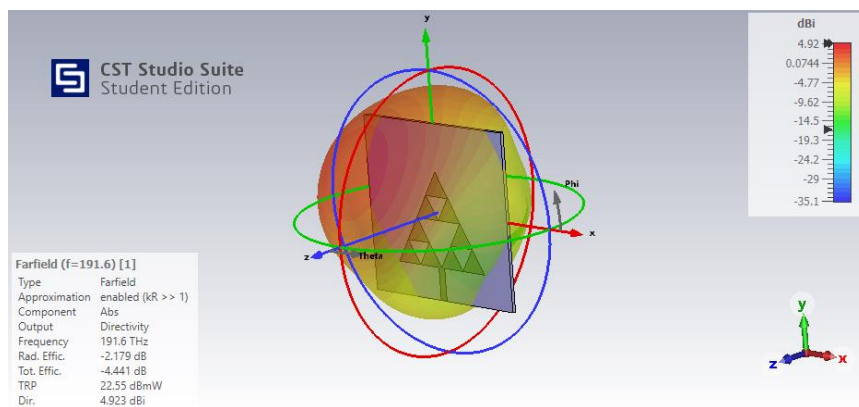


Figura 129. Modelo 7: Diagrama radiación 3D  $f_{r2}$

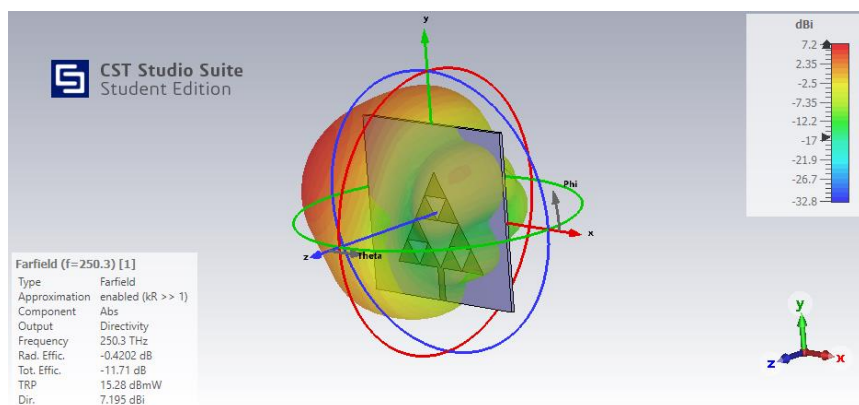


Figura 130. Modelo 7: Diagrama radiación 3D  $f_{r3}$

#### 4.7.4. Directividad

Frecuencias (THz)		Directividad (dBi)
Fr1	44,9	-7,919
Fr2	191,6	4,923
Fr3	250,2	7,195

Tabla 38. Modelo 7: Directividad

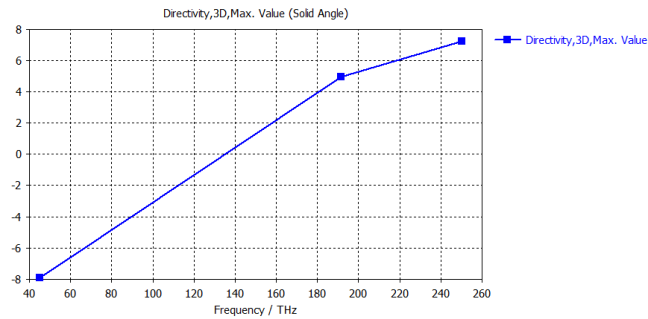


Figura 131. Modelo 7: Directividad

#### 4.7.5. Eficiencias de Radiación

Frecuencias (THz)		Rad.Effic (dB)	Rad.Effic (%)	Tot.Effic (dB)	Tot.Effic (%)
Fr1	44,9	-0,539	88,324	-15,930	2,553
Fr2	191,6	-2,179	60,548	-4,441	35,967
Fr3	250,2	-0,420	90,778	-11,710	6,745

Tabla 39. Modelo 7: Eficiencias de Radiación

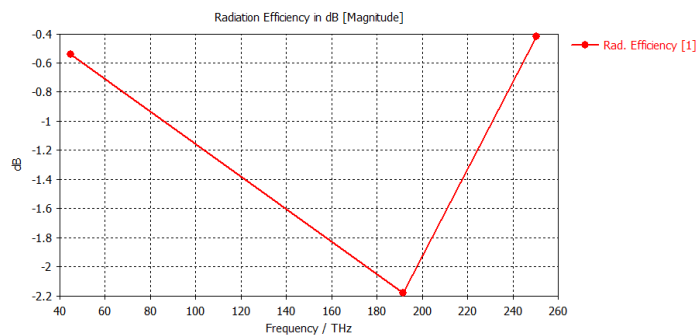


Figura 132. Modelo 7: Eficiencia de Radiación

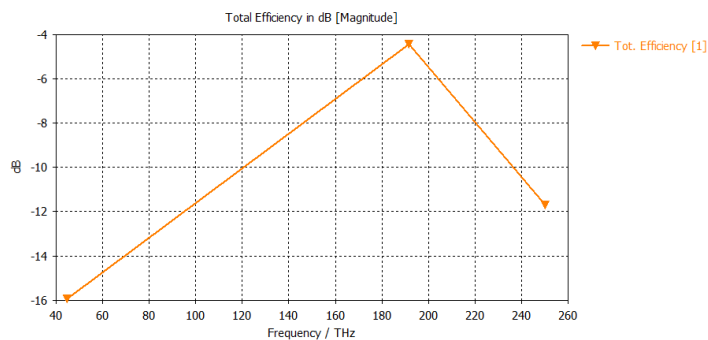


Figura 133. Modelo 7: Eficiencia Total

#### 4.7.6. Ganancia

Frecuencias (THz)		Gain (dB)
Fr1	44,9	-8,402
Fr2	191,6	2,748
Fr3	250,2	6,736

Tabla 40. Modelo 7: Ganancia

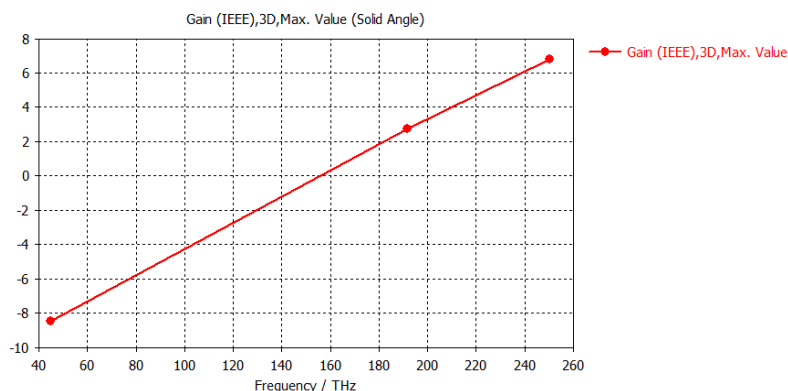


Figura 134. Modelo 7: Ganancia

#### 4.7.7. Ganancia Realizable

Frecuencias (THz)		Gain Realized (dB)
Fr1	44,9	-23,811
Fr2	191,6	0,475
Fr3	250,2	-4,469

Tabla 41. Modelo 7: Ganancia Realizable

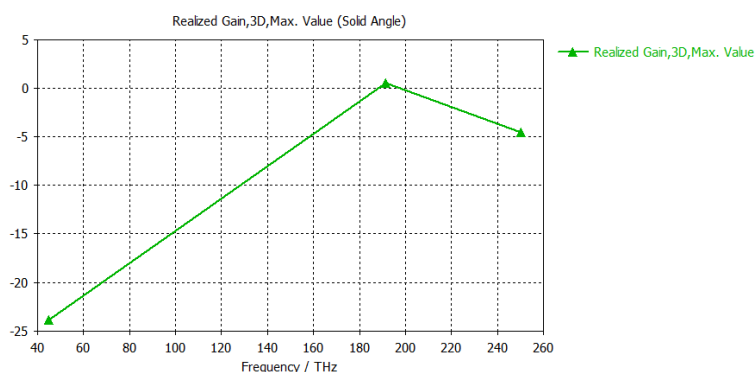


Figura 135. Modelo 7: Ganancia Realizable

#### 4.7.8. Potencia Total Radiada (TRP)

Frecuencias (THz)		TPR (3dBm)	TPR (13dBm)
Fr1	44,9	-12,930	-2,930
Fr2	191,6	-1,441	8,559
Fr3	250,2	-8,712	1,288

Tabla 42. Modelo 7: Potencia Total Radiada

## 4.8. Modelo 8: Fractal orden 2 con grafeno y sustrato fibra de vidrio

### 4.8.1. Componente

Nombre proyecto: sierpinski\_2\_grafeno2\_glass\_epoxy\_pec.cst

Orden fractal: 2

Materiales:

Parche grafeno:  $\mu_c = 0.2 \text{ eV}$

Sustrato glass\_epoxy:  $\epsilon_r = 4.4$

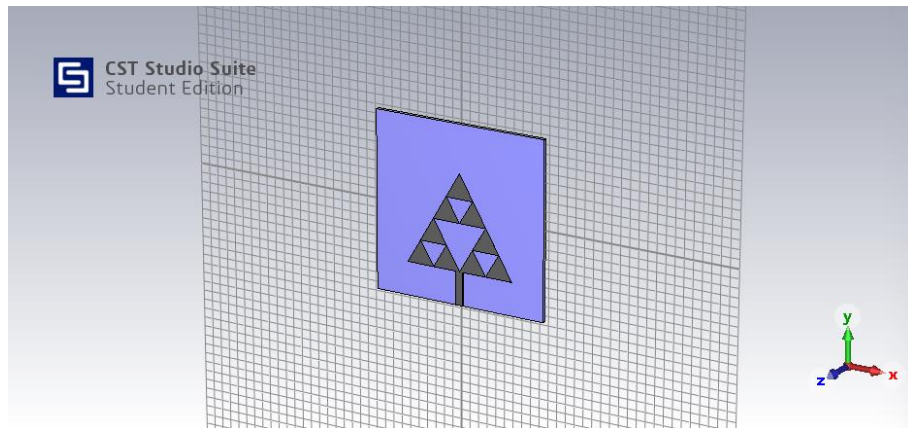


Figura 136. Modelo 8: Componente

### 4.8.2. Coeficiente de Reflexión $S_{11}$

Frecuencias (THz)		$S_{11}$ (dB)
Fr1	74,35	-20,70
Fr2	287,9	-3,68

Tabla 43. Modelo 8: Coeficiente de Reflexión  $S_{11}$

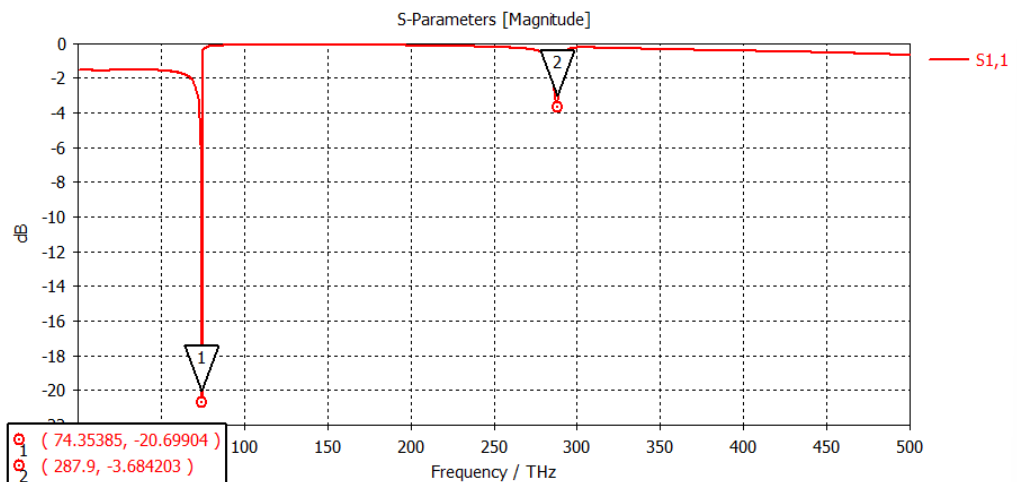


Figura 137. Modelo 8:  $S_{11}$



### 4.8.3. Farfields para las frecuencia de resonancia

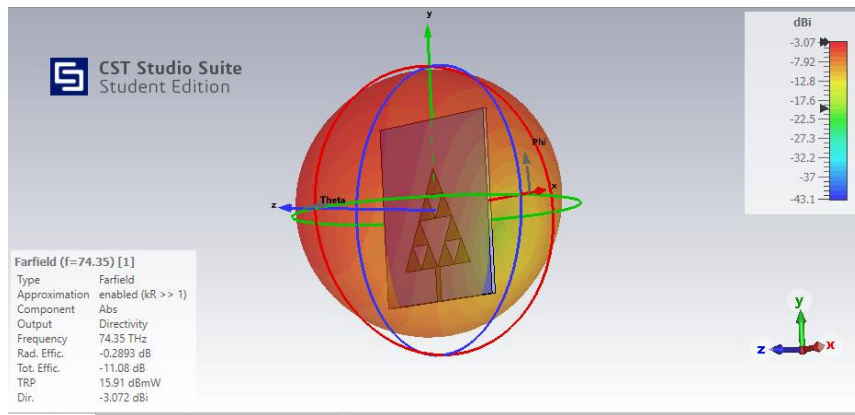


Figura 138. Modelo 8: Diagrama radiación 3D  $f_{r1}$

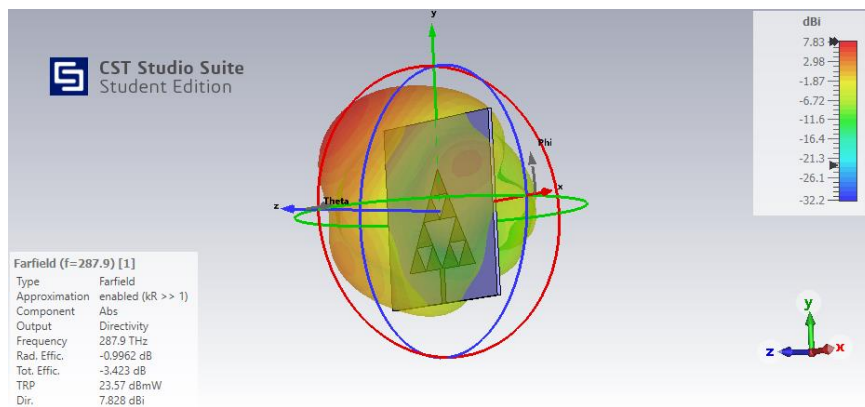


Figura 139. Modelo 8: Diagrama radiación 3D  $f_{r2}$

### 4.8.4. Directividad

Frecuencias (THz)		Directividad (dBi)
Fr1	74,35	-3,072
Fr2	287,9	7,828

Tabla 44. Modelo 8: Directividad

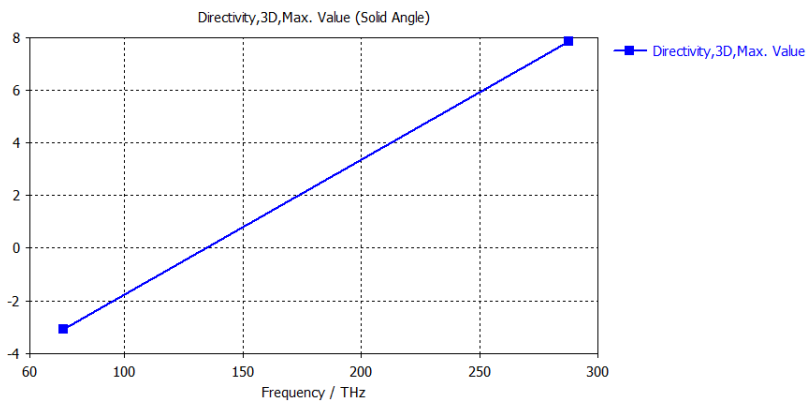


Figura 140. Modelo 8: Directividad

#### 4.8.5. Eficiencias de Radiación

Frecuencias (THz)		Rad.Effic (dB)	Rad.Effic (%)	Tot.Effic (dB)	Tot.Effic (%)
Fr1	74,35	-0,289	93,556	-11,080	7,798
Fr2	287,9	-0,996	79,502	-3,423	45,467

Tabla 45. Modelo 8: Eficiencias de Radiación

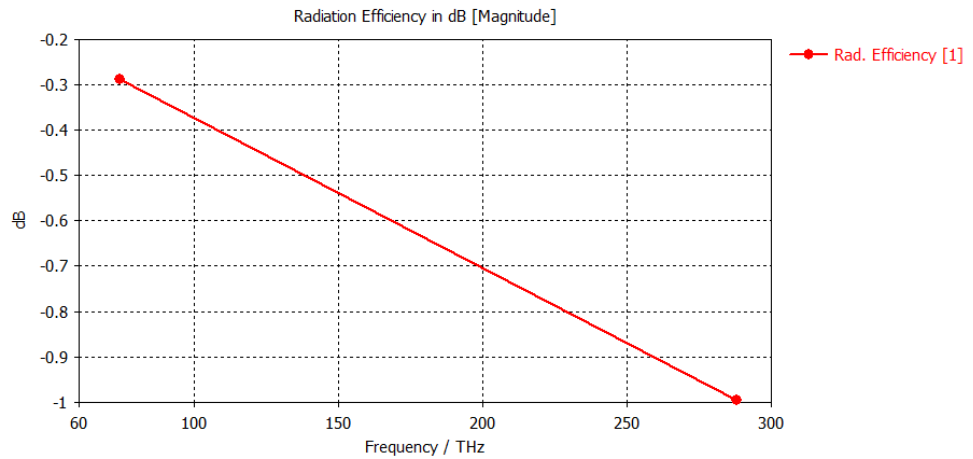


Figura 141. Modelo 8: Eficiencia de Radiación

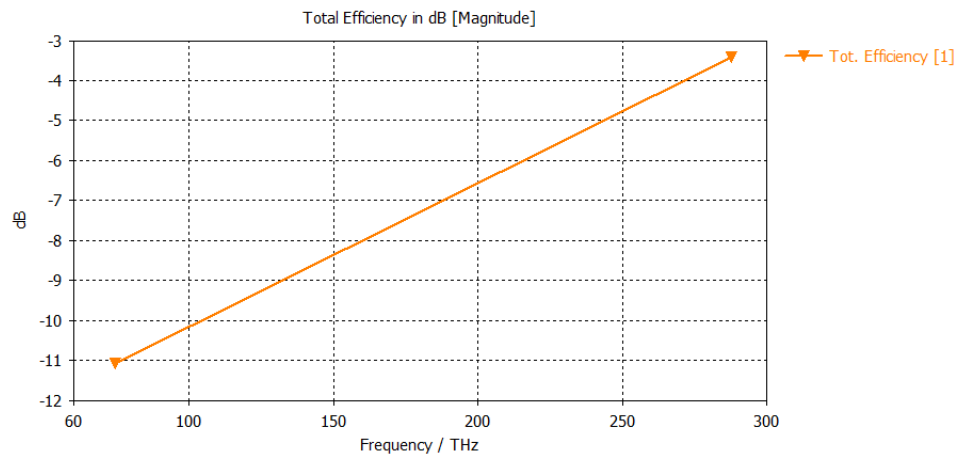


Figura 142. Modelo 8: Eficiencia Total

#### 4.8.6. Ganancia

Frecuencias (THz)		Gain (dB)
Fr1	74,35	-3,365
Fr2	287,9	6,831

Tabla 46. Modelo 8: Ganancia

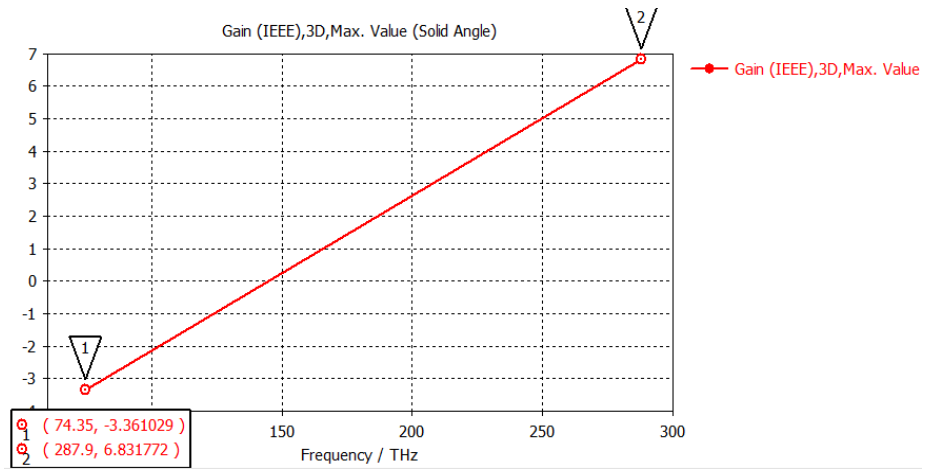


Figura 143. Modelo 8: Ganancia

#### 4.8.7. Ganancia Realizable

Frecuencias (THz)		Gain Realized (dB)
Fr1	74,35	-14,151
Fr2	287,9	4,336

Tabla 47. Modelo 8: Ganancia Realizable

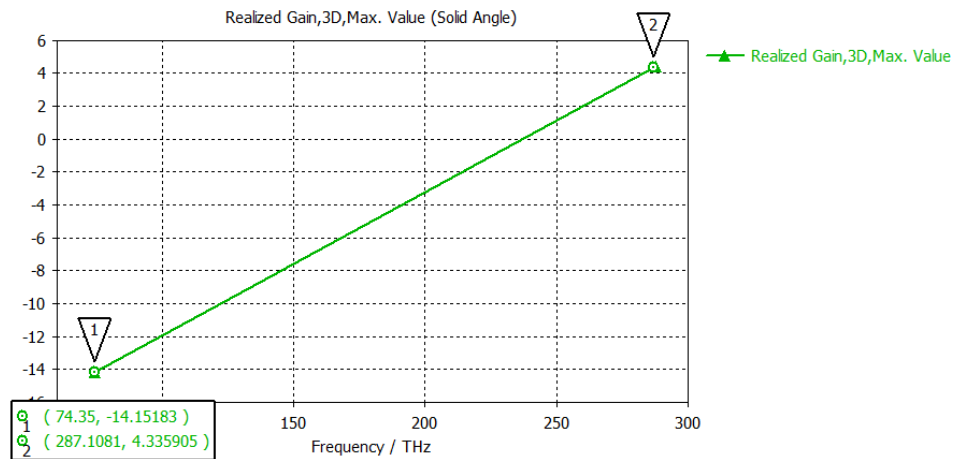


Figura 144. Modelo 8: Ganancia Realizable

#### 4.8.8. Potencia Total Radiada (TRP)

Frecuencias (THz)		TPR (3dBm)	TPR (13dBm)
Fr1	74,35	-8,080	-3,072
Fr2	287,9	-0,581	9,577

Tabla 48. Modelo 8: Potencia Total Radiada

## 4.9. Modelo 9: Fractal orden 2 con grafeno y sustrato silicio

### 4.9.1. Componente

Nombre proyecto: sierpinski\_2\_grafeno4\_silicon\_pec.cst

Orden fractal: 2

Materiales:

Parche grafeno:  $\mu_c = 0.2$  eV

Sustrato silicio:  $\epsilon_r = 11.9$

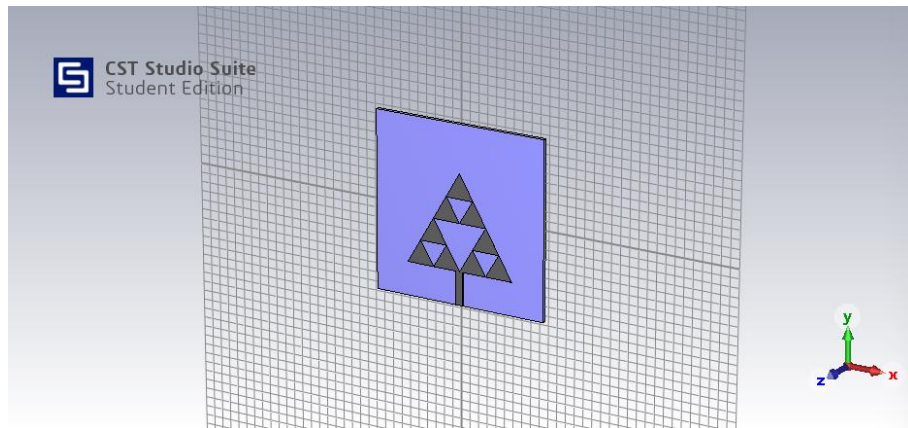


Figura 145. Modelo 9: Componente

### 4.9.2. Coeficiente de Reflexión $S_{11}$

Frecuencias (THz)		$S_{11}$ (dB)
Fr1	44,9	-7,78
Fr2	192,1	-4,39

Tabla 49. Modelo 9: Coeficiente de Reflexión  $S_{11}$

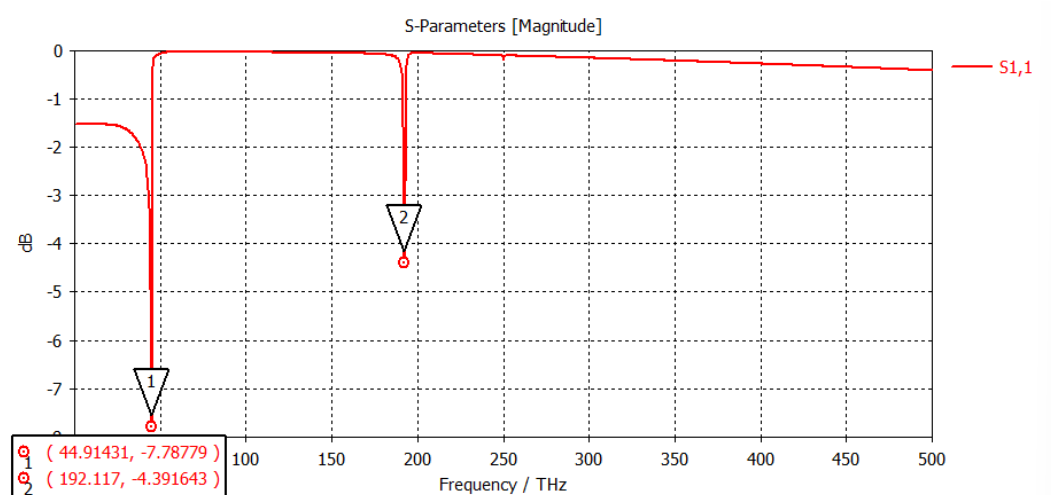


Figura 146. Modelo 9:  $S_{11}$

### 4.9.3. Farfields para las frecuencia de resonancia

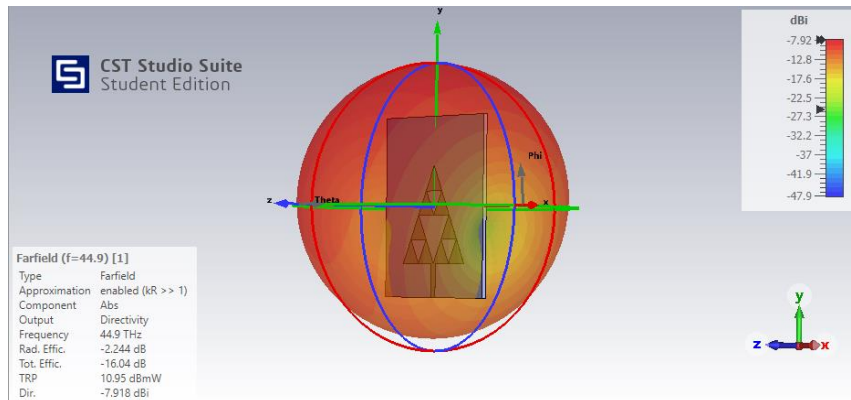


Figura 147. Modelo 9: Diagrama radiación 3D fr1

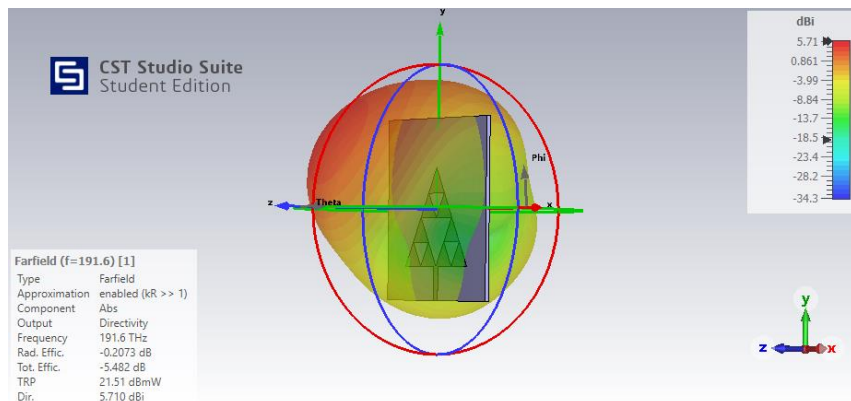


Figura 148. Modelo 9: Diagrama radiación 3D fr2

### 4.9.4. Directividad

Frecuencias (THz)		Directividad (dBi)
Fr1	44,9	-7,918
Fr2	192,1	5,710

Tabla 50. Modelo 9: Directividad

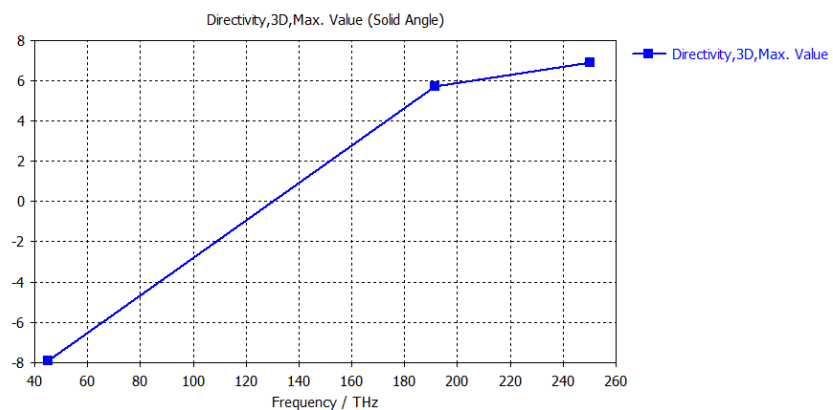


Figura 149. Modelo 9: Directividad

#### 4.9.5. Eficiencias de Radiación

Frecuencias (THz)		Rad.Effic (dB)	Rad.Effic (%)	Tot.Effic (dB)	Tot.Effic (%)
Fr1	44,9	-2,244	59,649	-16,04	2,489
Fr2	192,1	-0,207	95,345	-5,482	28,301

Tabla 51. Modelo 9: Eficiencias de Radiación

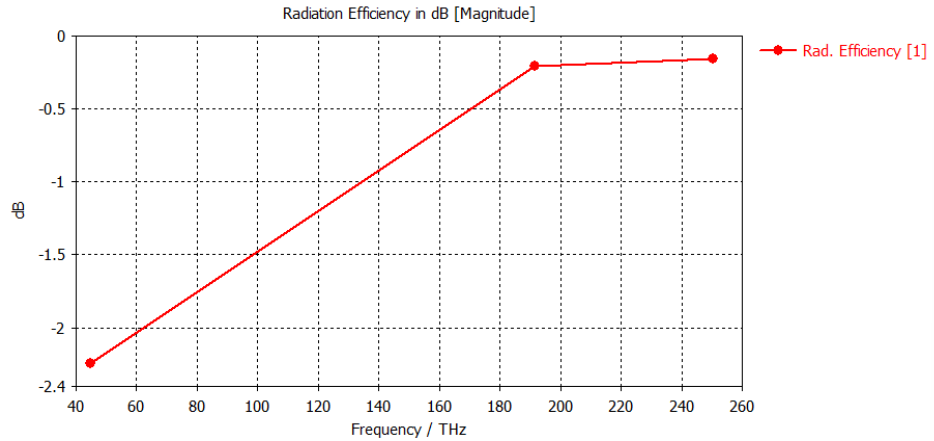


Figura 150. Modelo 9: Eficiencia de Radiación

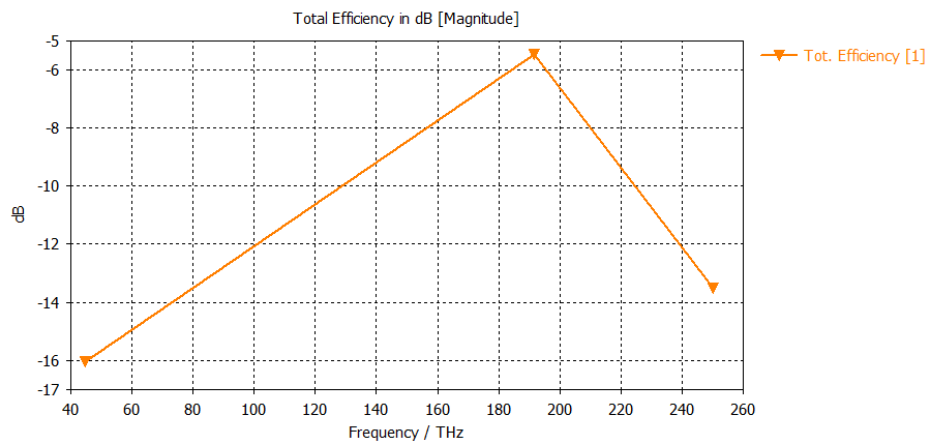


Figura 151. Modelo 9: Eficiencia Total

#### 4.9.6. Ganancia

Frecuencias (THz)		Gain (dB)
Fr1	44,9	10,156
Fr2	192,1	5,506

Tabla 52. Modelo 9: Ganancia

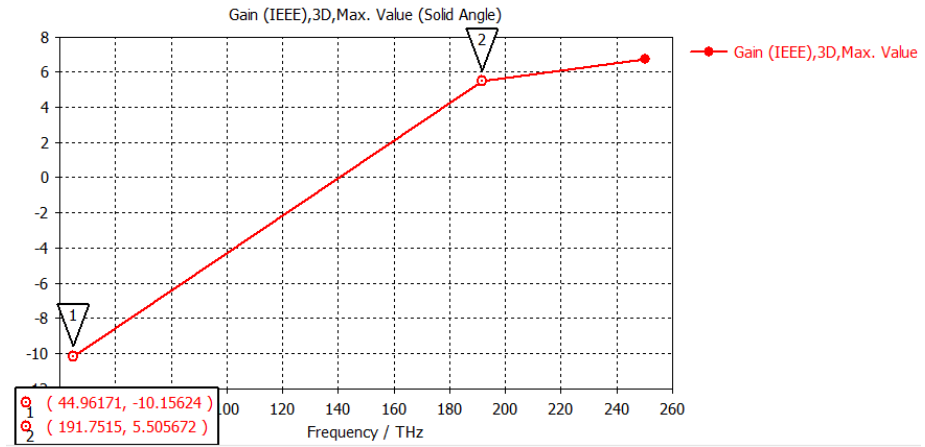


Figura 152. Modelo 9: Ganancia

#### 4.9.7. Ganancia Realizable

Frecuencias (THz)		Gain Realized (dB)
Fr1	44,9	-23,927
Fr2	192,1	0,213

Tabla 53. Modelo 9: Ganancia Realizable

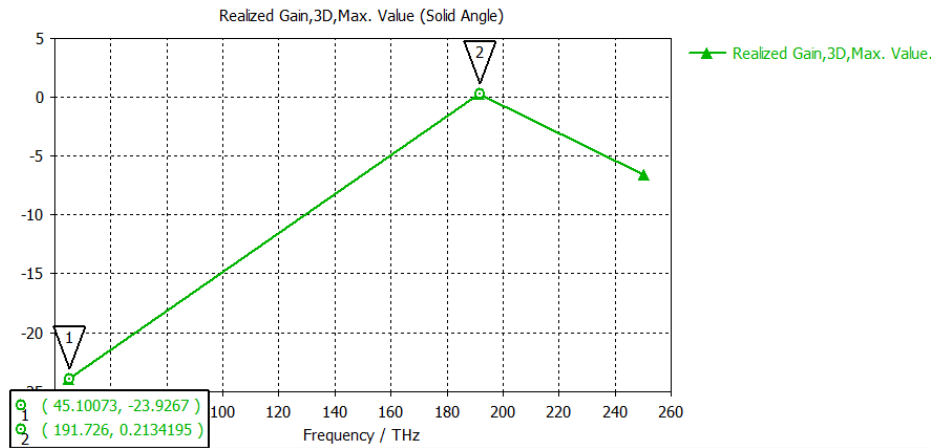


Figura 153. Modelo 9: Ganancia Realizable

#### 4.9.8. Potencia Total Radiada (TRP)

Frecuencias (THz)		TPR (3dBm)	TPR (13dBm)
Fr1	44,9	-13,04	-3,041
Fr2	192,1	-2,482	7,518

Tabla 54. Modelo 9: Potencia Total Radiada

## 4.10. Modelo 10: Fractal orden 2 con grafeno 4 y sustrato fibra de vidrio

### 4.10.1. Componente

Nombre proyecto: sierpinski\_2\_grafeno4\_glass\_epoxy\_pec.cst

Orden fractal: 2

Materiales:

Parche grafeno:  $\mu_c = 0.43$  eV

Sustrato glass\_epoxy:  $\epsilon_r = 4.4$

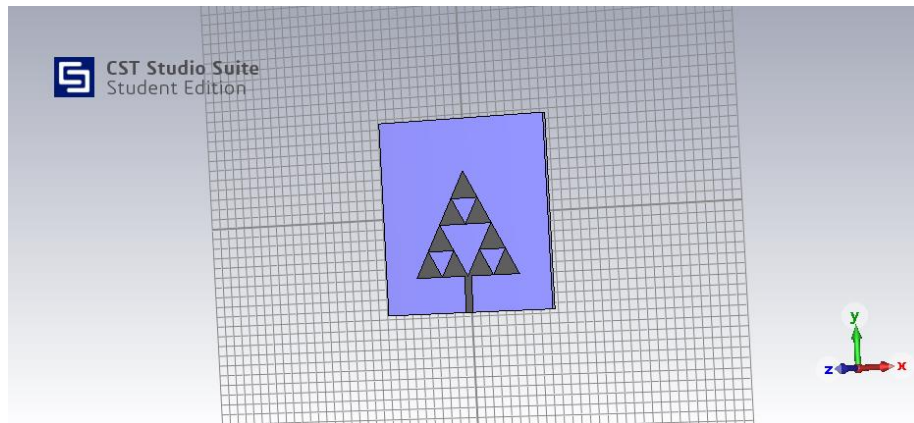


Figura 154. Modelo 10: Componente

### 4.10.2. Coeficiente de Reflexión $S_{11}$

Frecuencias (THz)		$S_{11}$ (dB)
Fr1	74,35	-20,44
Fr2	289,8	-2,102

Tabla 55. Modelo 10: Coeficiente de Reflexión  $S_{11}$

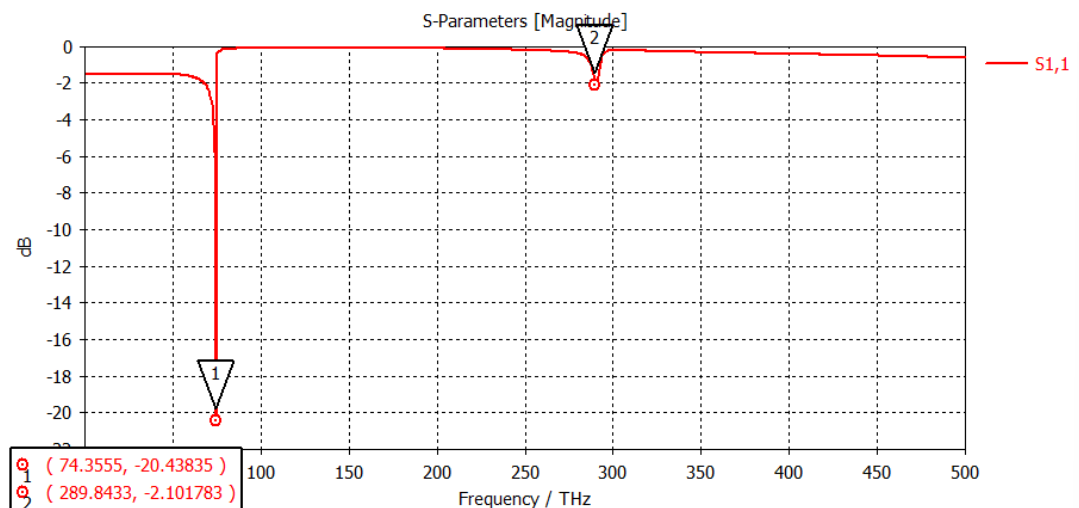


Figura 155. Modelo 10:  $S_{11}$



### 4.10.3. Farfields para las frecuencia de resonancia

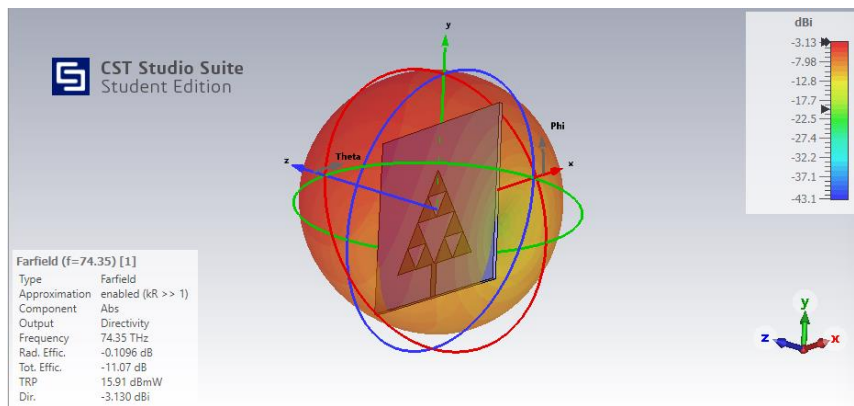


Figura 156. Modelo 10: Diagrama radiación 3D fr<sub>1</sub>

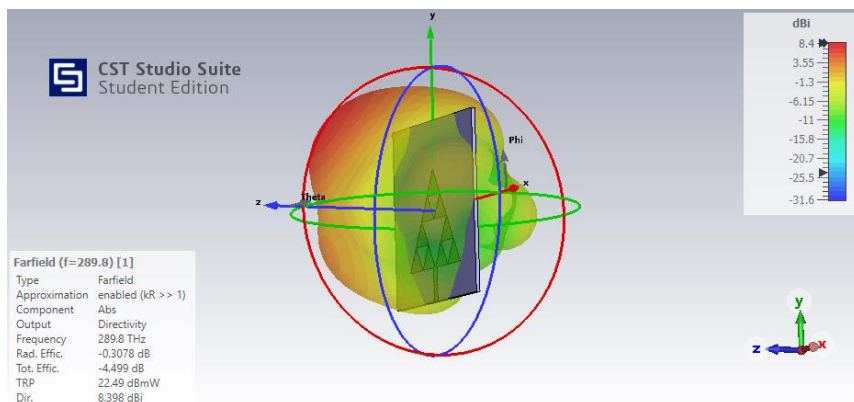


Figura 157. Modelo 10: Diagrama radiación 3D fr<sub>2</sub>

### 4.10.4. Directividad

Frecuencias (THz)		Directividad (dBi)
Fr1	74,35	-3,130
Fr2	289,8	8,398

Tabla 56. Modelo 10: Directividad

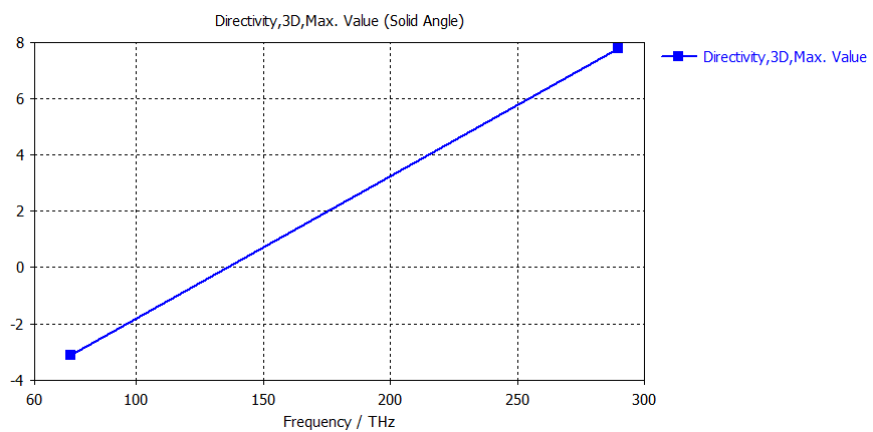


Figura 158. Modelo 10: Directividad

#### 4.10.5. Eficiencias de Radiación

Frecuencias (THz)		Rad.Effic (dB)	Rad.Effic (%)	Tot.Effic (dB)	Tot.Effic (%)
Fr1	74,35	-0,110	97,499	-11,070	7,816
Fr2	289,8	-0,308	93,154	-4,500	35,481

Tabla 57. Modelo 10: Eficiencias de Radiación

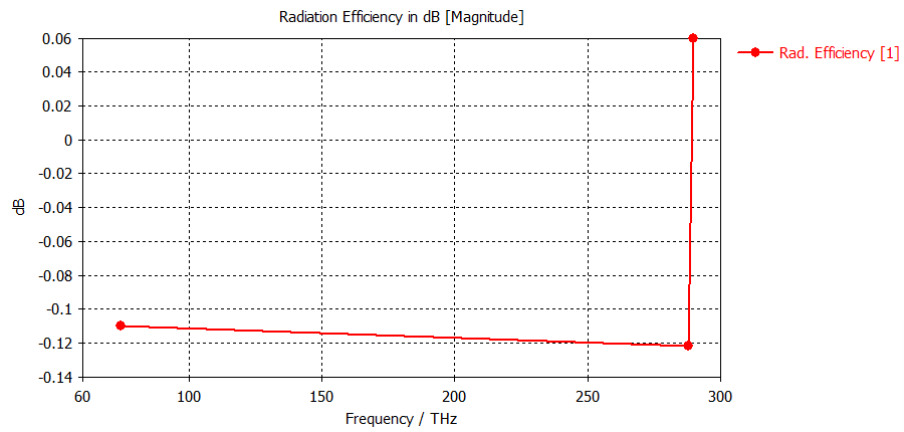


Figura 159. Modelo 10: Eficiencia de Radiación

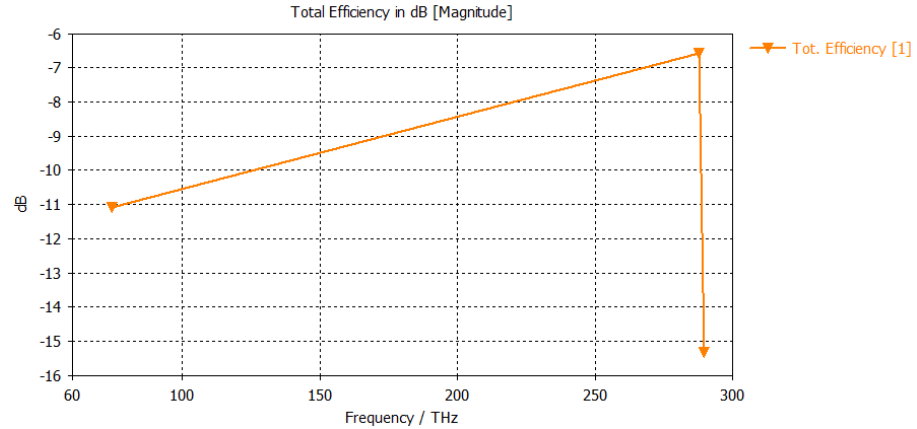


Figura 160. Modelo 10: Eficiencia Total

#### 4.10.6. Ganancia

Frecuencias (THz)		Gain (dB)
Fr1	74,35	-3,240
Fr2	289,8	8,062

Tabla 58. Modelo 10: Ganancia

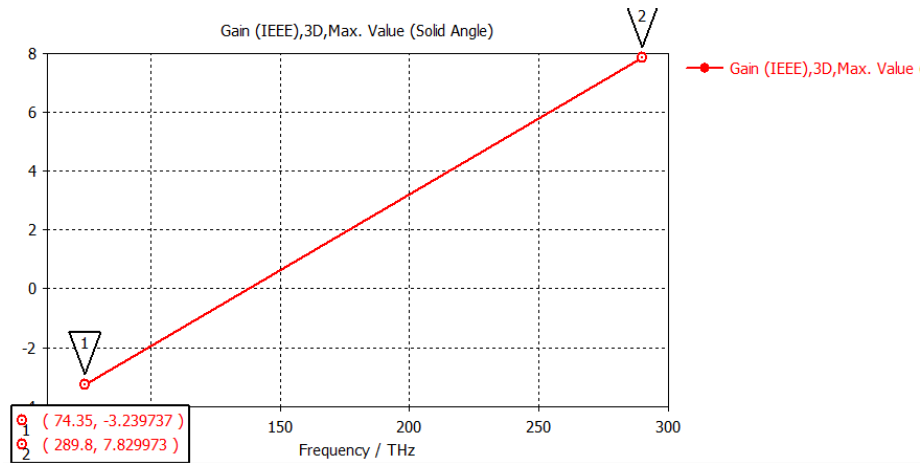


Figura 161. Modelo 10: Ganancia

#### 4.10.7. Ganancia Realizable

Frecuencias (THz)		Gain Realized (dB)
Fr1	74,35	-14,205
Fr2	289,8	3,899

Tabla 59. Modelo 10: Ganancia Realizable

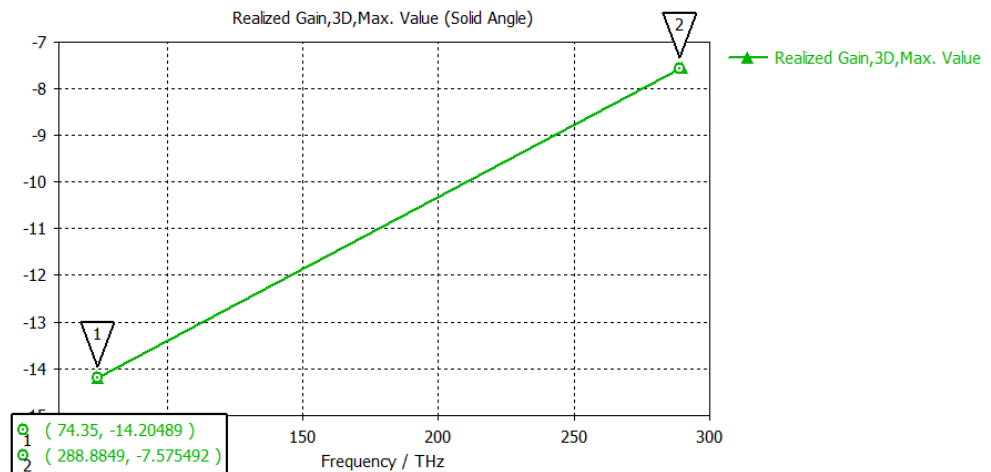


Figura 162. Modelo 10: Ganancia Realizable

#### 4.10.8. Potencia Total Radiada (TRP)

Frecuencias (THz)		TPR (3dBm)	TPR (13dBm)
Fr1	74,35	-8,075	1,925
Fr2	289,8	-12,320	-2,317

Tabla 60. Modelo 10: Potencia Total Radiada

## 4.11. Modelo 11: Fractal orden 3 con grafeno y sustrato silicio

### 4.11.1. Componente

Nombre proyecto: sierpinski\_3\_grafeno\_silicon\_pec.cst

Orden fractal: 3

Materiales:

Parche grafeno:  $\mu_c = 0.2 \text{ eV}$

Sustrato silicio:  $\epsilon_r = 11.9$

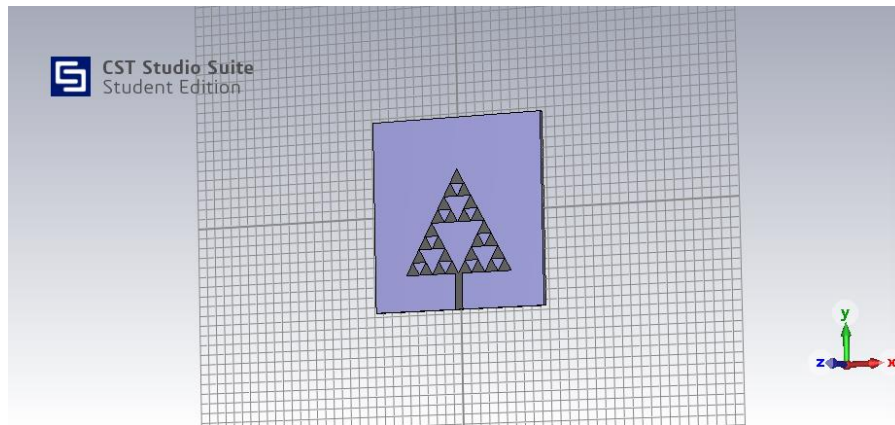


Figura 163. Modelo 11: Componente

### 4.11.2. Coeficiente de Reflexión $S_{11}$

Frecuencias (THz)		$S_{11}$ (dB)
Fr1	44,90	-7,620
Fr2	193,60	-3,530

Tabla 61. Modelo 11: Coeficiente de Reflexión  $S_{11}$

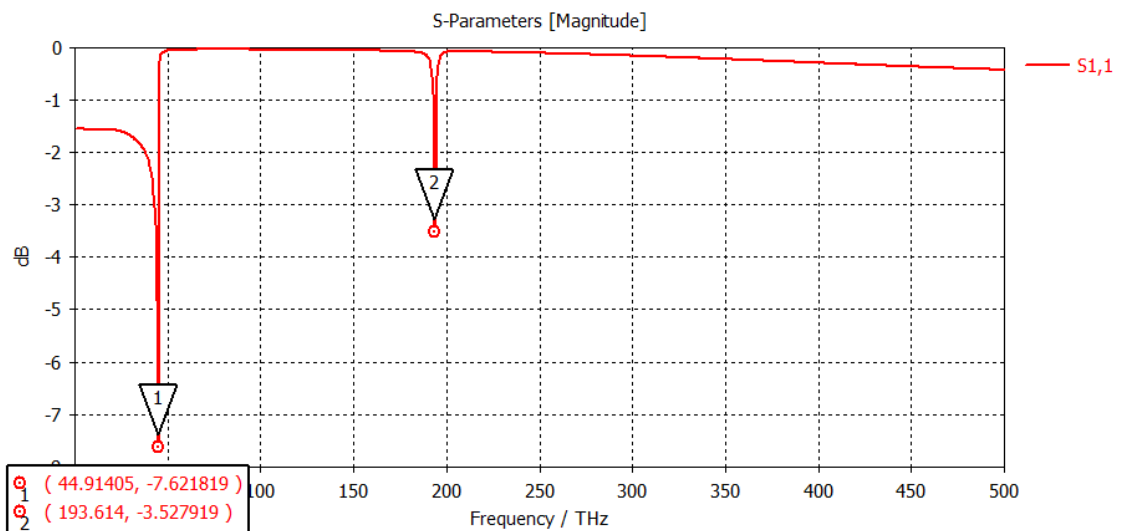


Figura 164. Modelo 11:  $S_{11}$

### 4.11.3. Farfields para las frecuencia de resonancia

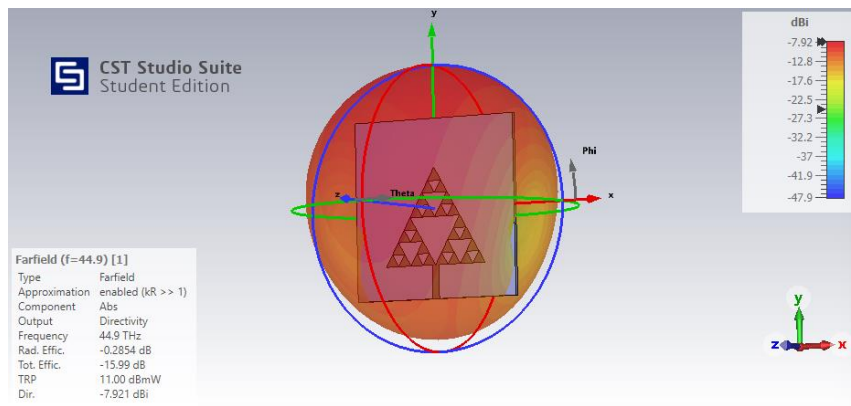


Figura 165. Modelo 11: Diagrama radiación 3D  $f_{r1}$

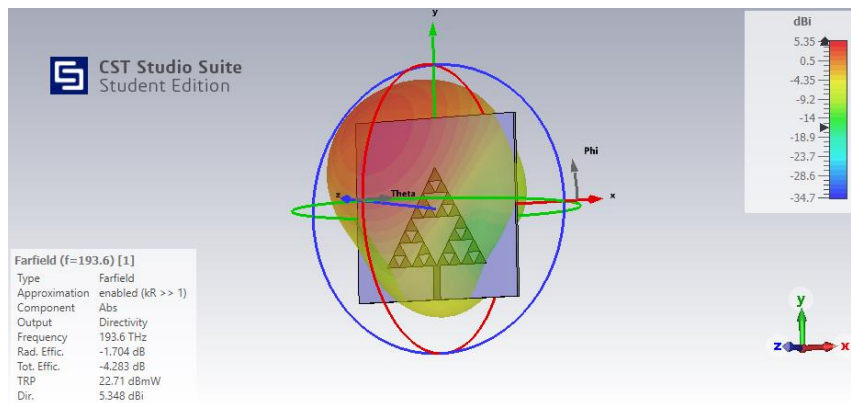


Figura 166. Modelo 11: Diagrama radiación 3D  $f_{r2}$

### 4.11.4. Directividad

Frecuencias (THz)		Directividad (dBi)
Fr1	44,90	-7,921
Fr2	193,60	5,348

Tabla 62. Modelo 11: Directividad

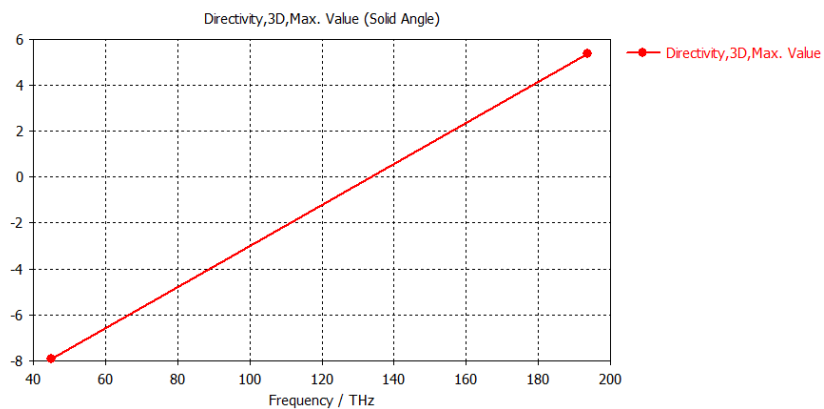


Figura 167. Modelo 11: Directividad

#### 4.11.5. Eficiencias de Radiación

Frecuencias (THz)		Rad.Effic (dB)	Rad.Effic (%)	Tot.Effic (dB)	Tot.Effic (%)
Fr1	44,9	-2,244	59,649	-16,04	2,489
Fr2	192,1	-0,207	95,345	-5,482	28,301

Tabla 63. Modelo 11: Eficiencias de Radiación

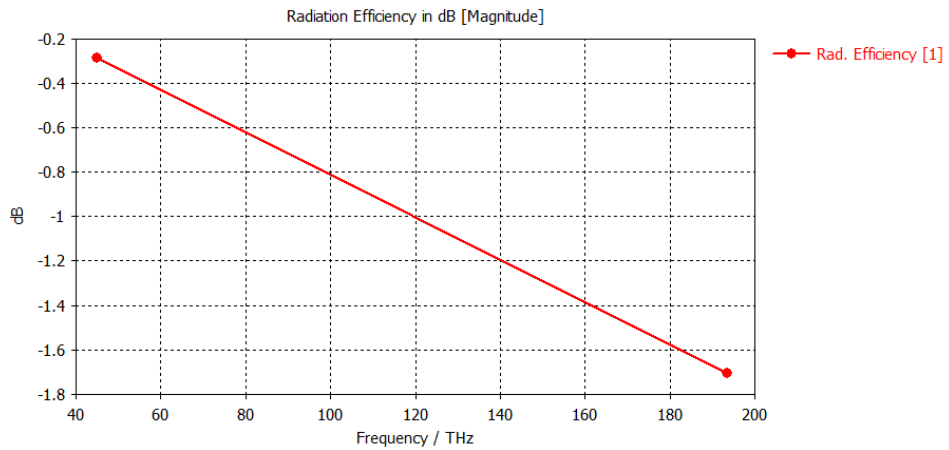


Figura 168. Modelo 11: Eficiencia de Radiación

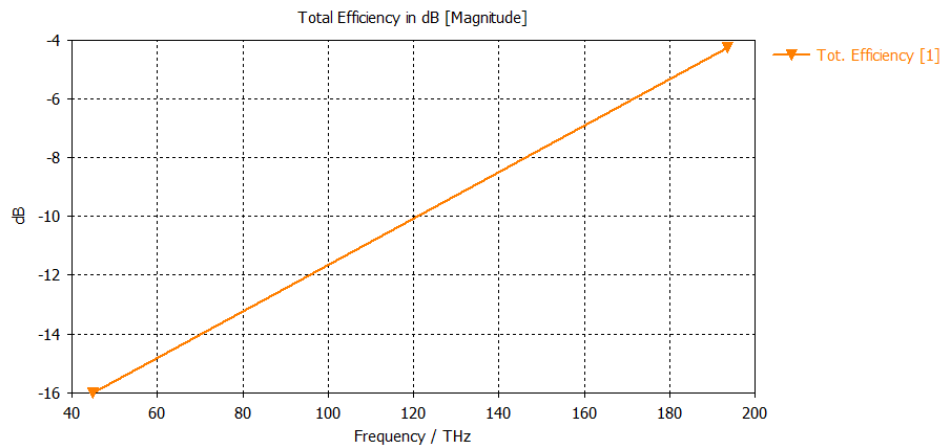


Figura 169. Modelo 11: Eficiencia Total

#### 4.11.6. Ganancia

Frecuencias (THz)		Gain (dB)
Fr1	44,90	-8,205
Fr2	193,60	3,588

Tabla 64. Modelo 11: Ganancia

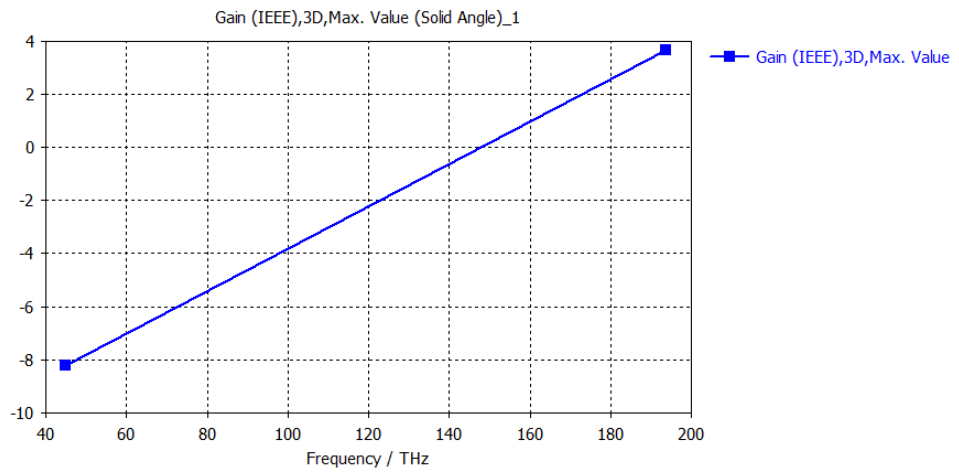


Figura 170. Modelo 11: Ganancia

#### 4.11.7. Ganancia Realizable

Frecuencias (THz)		Gain Realized (dB)
Fr1	44,90	-23,870
Fr2	193,60	0,947

Tabla 65. Modelo 11: Ganancia Realizable

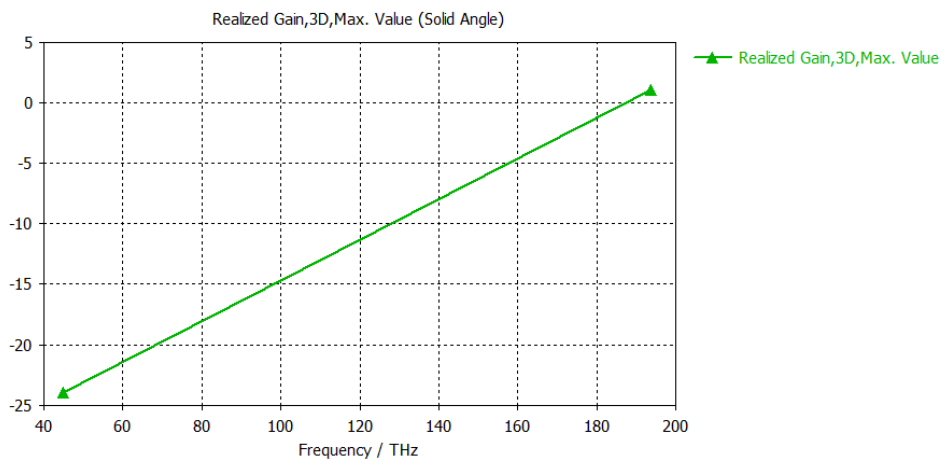


Figura 171. Modelo 11: Ganancia Realizable

#### 4.11.8. Potencia Total Radiada (TRP)

Frecuencias (THz)		TPR (3dBm)	TPR (13dBm)
Fr1	44,90	-12,990	-2,993
Fr2	193,60	-1,283	8,717

Tabla 66. Modelo 11: Potencia Total Radiada

## 4.12. Modelo 12: Fractal orden 3 con grafeno y sustrato fibra de vidrio

### 4.12.1. Componente

Nombre proyecto: sierpinski\_3\_grafeno\_glass\_epoxy\_pec.cst

Orden fractal: 3

Materiales:

Parche grafeno:  $\mu_c = 0.1 \text{ eV}$

Sustrato glass\_epoxy:  $\epsilon_r = 4.4$

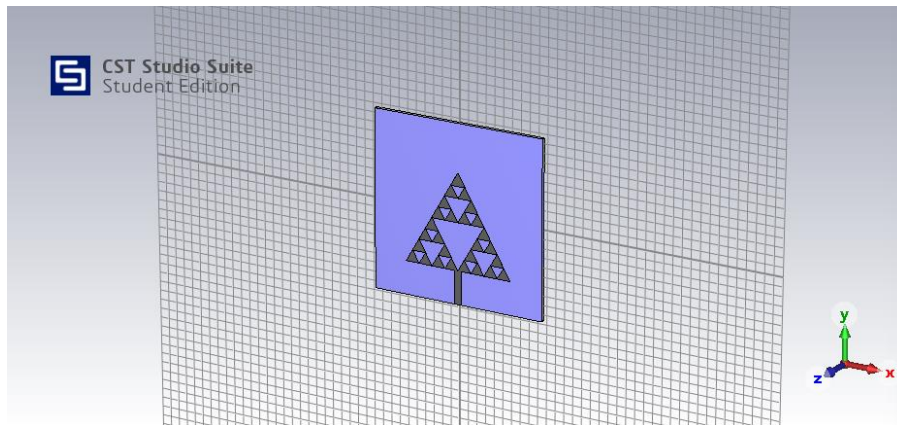


Figura 172. Modelo 12: Componente

### 4.12.2. Coeficiente de Reflexión $S_{11}$

Frecuencias (THz)		$S_{11}$ (dB)
Fr1	74,35	-19,825
Fr2	287,70	-2,571

Tabla 67. Modelo 12: Coeficiente de Reflexión  $S_{11}$

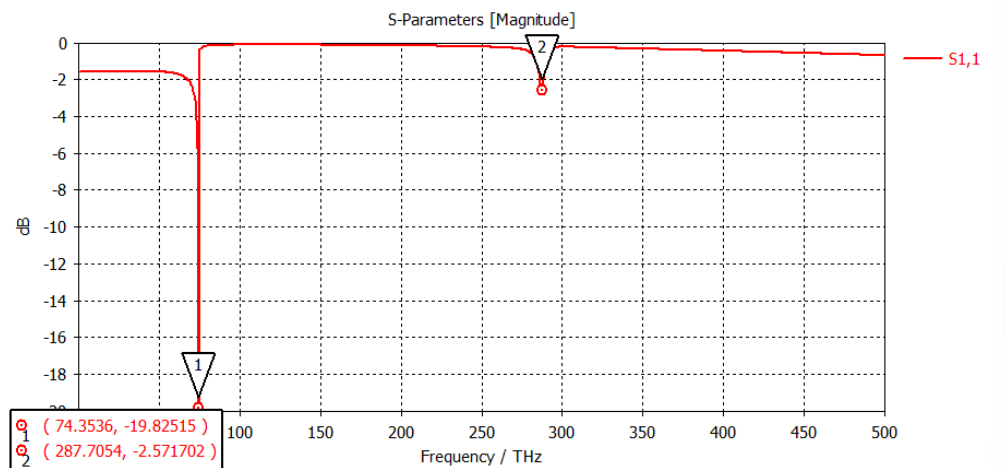


Figura 173. Modelo 12:  $S_{11}$



### 4.12.3. Farfields para las frecuencia de resonancia

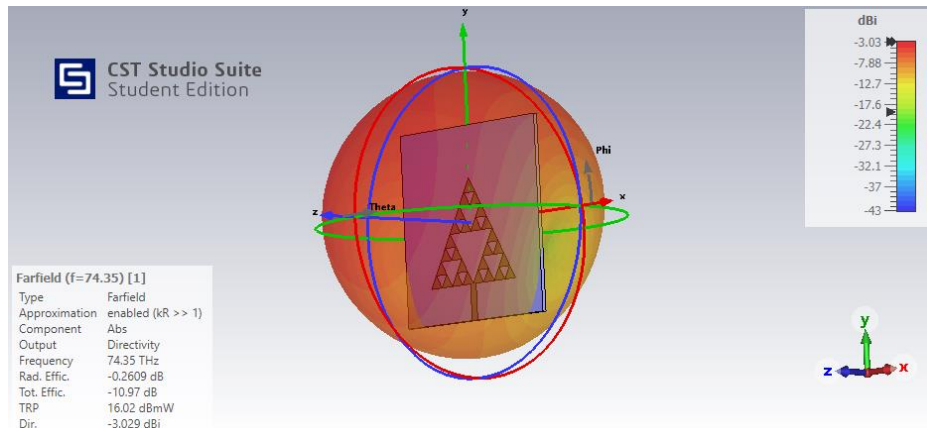


Figura 174. Modelo 12: Diagrama radiación 3D  $f_{r1}$

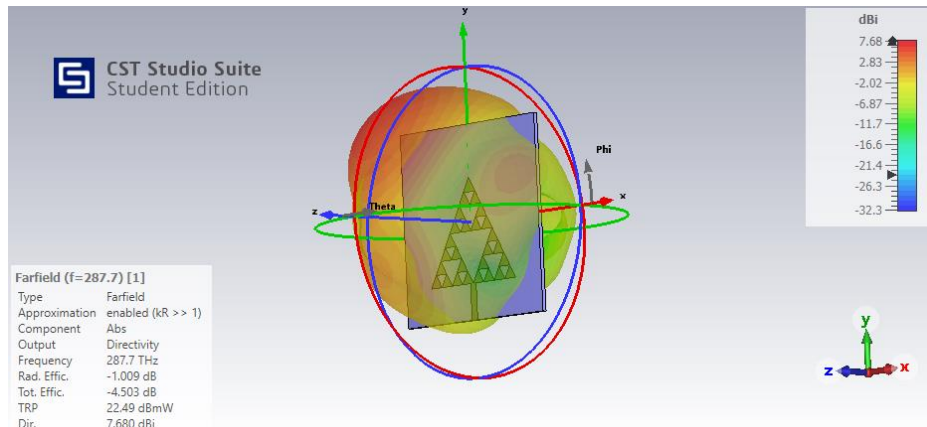


Figura 175. Modelo 12: Diagrama radiación 3D  $f_{r2}$

### 4.12.4. Directividad

Frecuencias (THz)		Directividad (dBi)
Fr1	74,35	-3,029
Fr2	287,70	7,680

Tabla 68. Modelo 12: Directividad

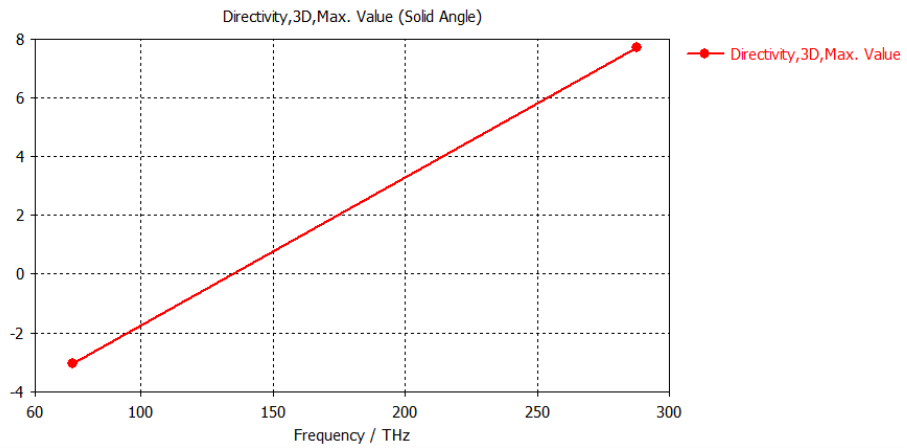


Figura 176. Modelo 12: Directividad

#### 4.12.5. Eficiencias de Radiación

Frecuencias (THz)		Rad.Effic (dB)	Rad.Effic (%)	Tot.Effic (dB)	Tot.Effic (%)
Fr1	74,35	-0,261	94,169	-10,970	7,998
Fr2	287,70	-1,009	79,268	-4,503	35,457

Tabla 69. Modelo 12: Eficiencias de Radiación

#### 4.12.6. Ganancia

Frecuencias (THz)		Gain (dB)
Fr1	74,35	-3,290
Fr2	287,70	6,641

Tabla 70. Modelo 12: Ganancia

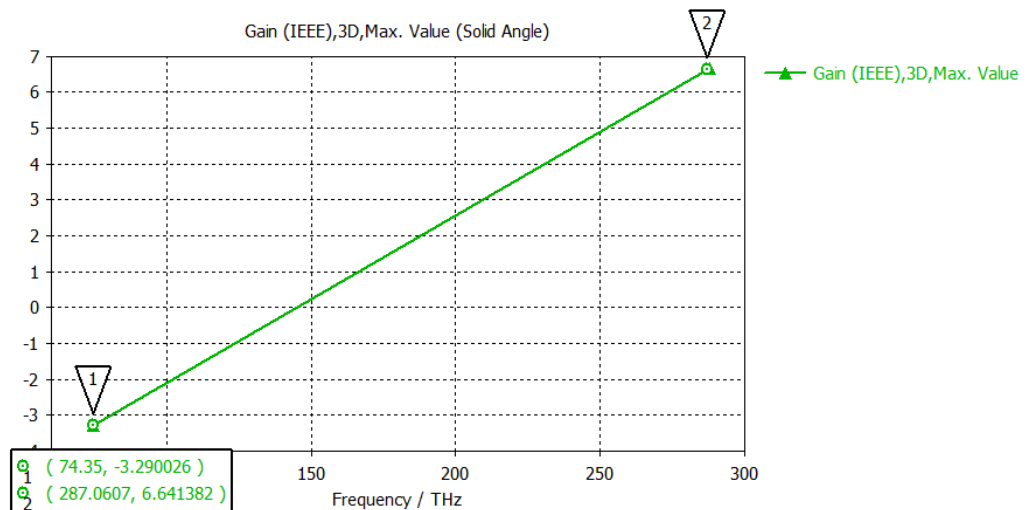


Figura 177. Modelo 12: Ganancia

#### 4.12.7. Ganancia Realizable

Frecuencias (THz)		Gain Realized (dB)
Fr1	74,35	-13,991
Fr2	287,70	3,177

Tabla 71. Modelo 12: Ganancia Realizable

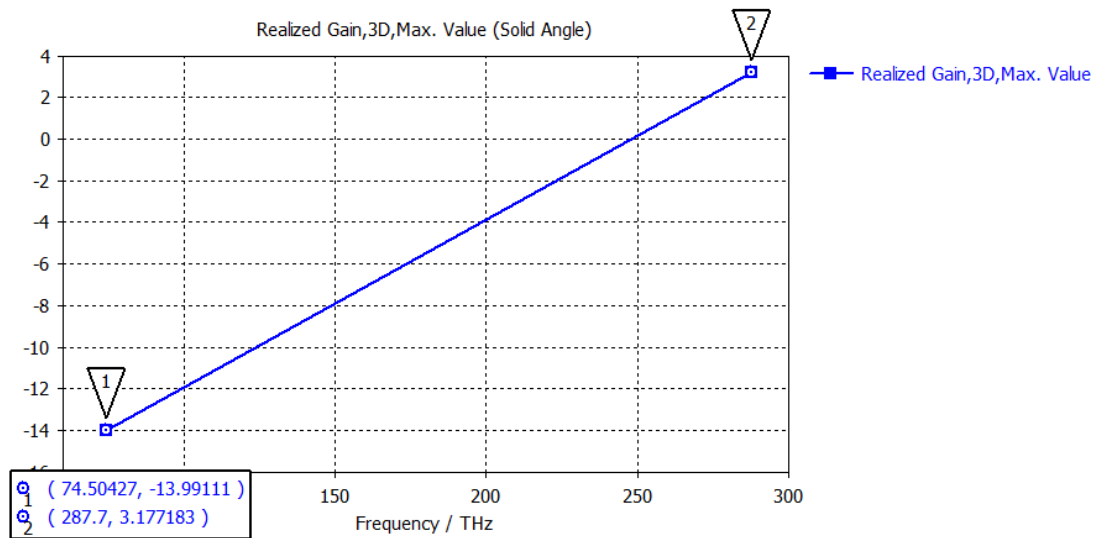


Figura 178. Modelo 12: Ganancia Realizable

#### 4.12.8. Potencia Total Radiada (TRP)

Frecuencias (THz)		TPR (3dBm)	TPR (13dBm)
Fr1	74,35	-7,974	2,026
Fr2	287,70	-1,503	8,497

Tabla 72. Modelo 12: Potencia Total Radiada

## 4.13. Modelo 13: Fractal orden 2 con grafeno 2 y sustrato silicio

### 4.13.1. Componente

Nombre proyecto: sierpinski\_3\_grafeno2\_silicon\_pec.cst

Orden fractal: 3

Materiales:

Parche grafeno:  $\mu_c = 0.2 \text{ eV}$

Sustrato silicio:  $\epsilon_r = 11.9$

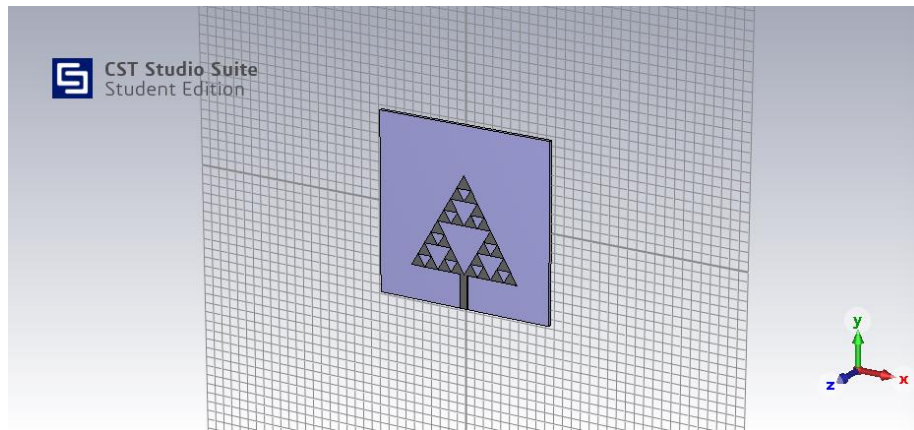


Figura 179. Modelo 13: Componente

### 4.13.2. Coeficiente de Reflexión $S_{11}$

Frecuencias (THz)		$S_{11}$ (dB)
Fr1	44,90	-7,650
Fr2	194,10	-3,490

Tabla 73. Modelo 13: Coeficiente de Reflexión  $S_{11}$

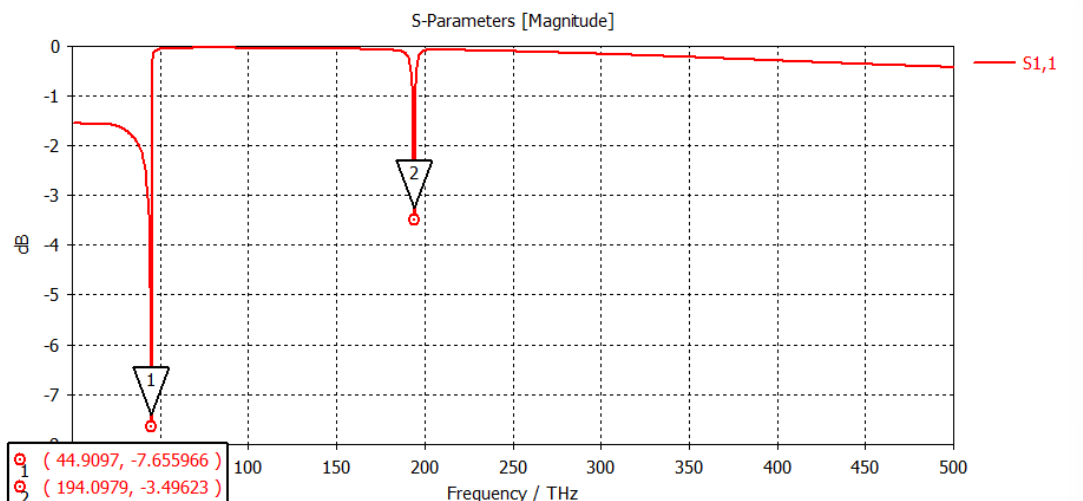


Figura 180. Modelo 13:  $S_{11}$

### 4.13.3. Farfields para las frecuencia de resonancia

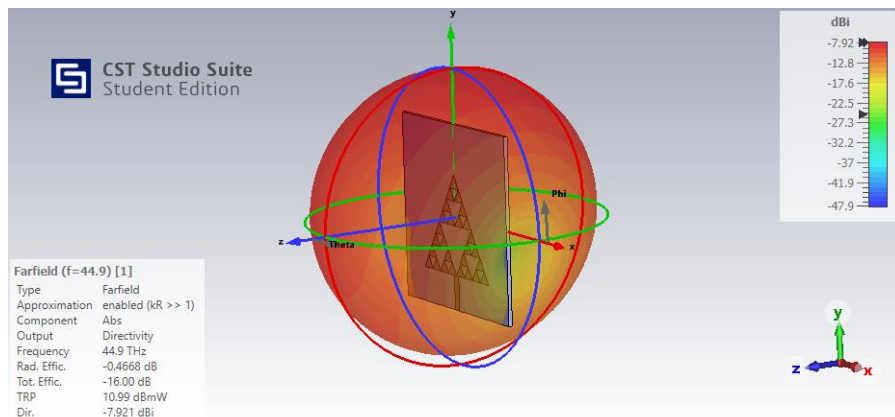


Figura 181. Modelo 13: Diagrama radiación 3D fr<sub>1</sub>

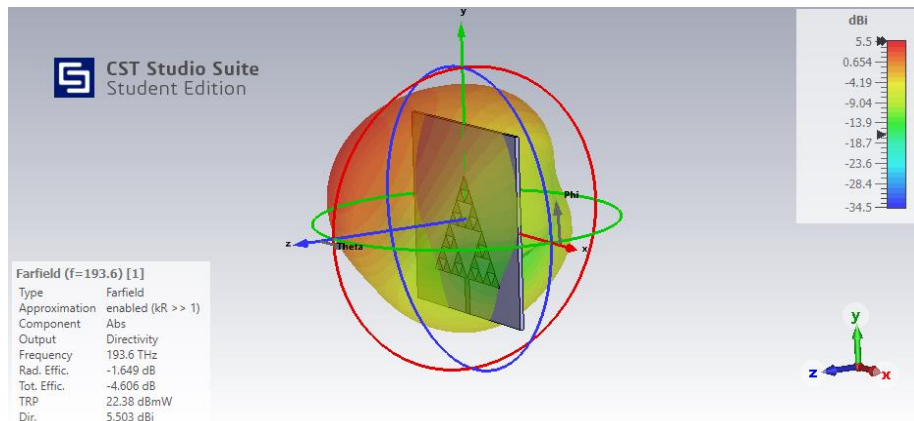


Figura 182. Modelo 13: Diagrama radiación 3D fr<sub>2</sub>

### 4.13.4. Directividad

Frecuencias (THz)		Directividad (dBi)
Fr1	44,90	-7,921
Fr2	194,10	5,503

Tabla 74. Modelo 13: Directividad

### 4.13.5. Eficiencias de Radiación

Frecuencias (THz)		Rad.Effic (dB)	Rad.Effic (%)	Tot.Effic (dB)	Tot.Effic (%)
Fr1	44,90	-0,467	89,809	-16,000	2,512
Fr2	194,10	-1,649	68,407	-4,606	34,626

Tabla 75. Modelo 13: Eficiencias de Radiación

#### 4.13.6. Ganancia

Frecuencias (THz)		Gain (dB)
Fr1	44,90	-8,390
Fr2	194,10	3,853

Tabla 76. Modelo 13: Ganancia

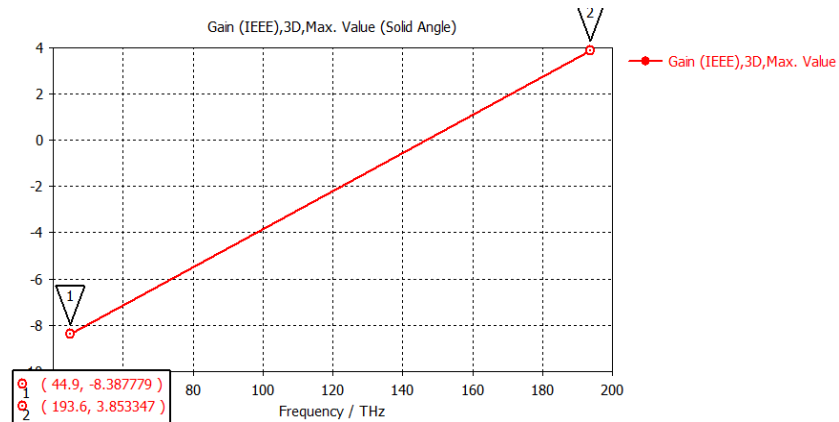


Figura 183. Modelo 13: Ganancia

#### 4.13.7. Ganancia Realizable

Frecuencias (THz)		Gain Realized (dB)
Fr1	44,90	-23,923
Fr2	194,10	0,871

Tabla 77. Modelo 13: Ganancia Realizable

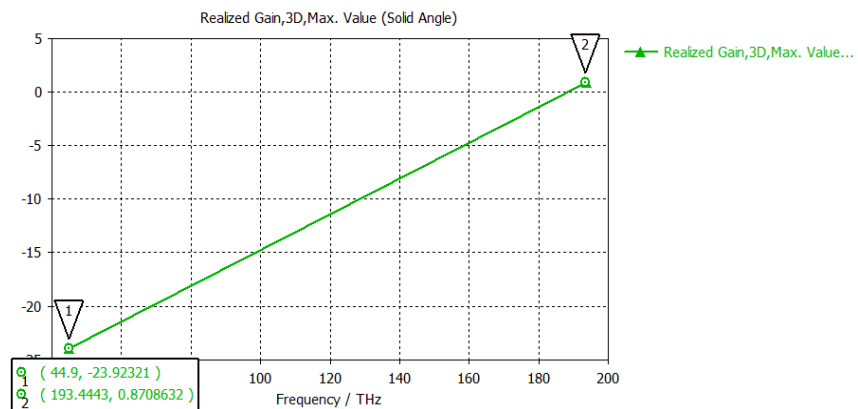


Figura 184. Modelo 13 Ganancia Realizable

#### 4.13.8. Potencia Total Radiada (TRP)

Frecuencias (THz)		TPR (3dBm)	TPR (13dBm)
Fr1	44,90	-13,000	-3,002
Fr2	194,10	-1,606	8,394

Tabla 78. Modelo 13: Potencia Total Radiada

## 4.14. Modelo 14: Fractal orden 3 con grafeno 2 y sustrato fibra de vidrio

### 4.14.1. Componente

Nombre proyecto: sierpinski\_3\_grafeno2\_glass\_epoxy\_pec.cst

Orden fractal: 3

Materiales:

Parche grafeno:  $\mu_c = 0.2 \text{ eV}$

Sustrato glass\_epoxy:  $\epsilon_r = 4.4$

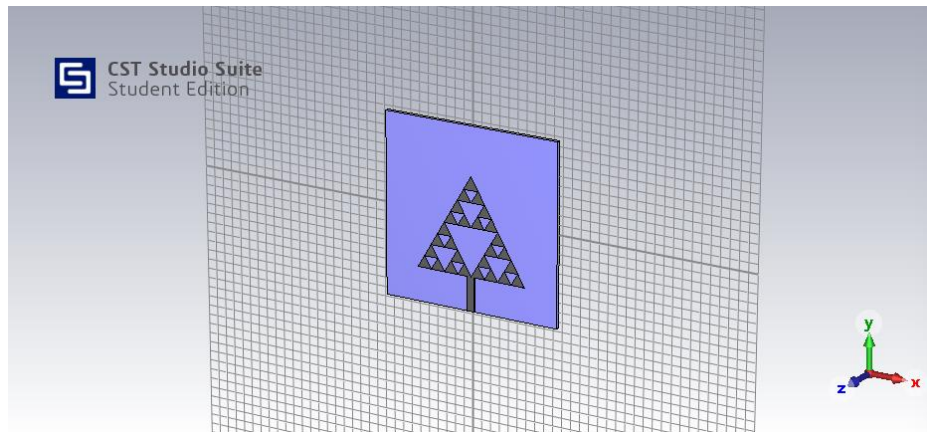


Figura 185. Modelo 14: Componente

### 4.14.2. Coeficiente de Reflexión $S_{11}$

Frecuencias (THz)		$S_{11}$ (dB)
Fr1	74,35	-20,170
Fr2	287,95	-2,702

Tabla 79. Modelo 14: Coeficiente de Reflexión  $S_{11}$

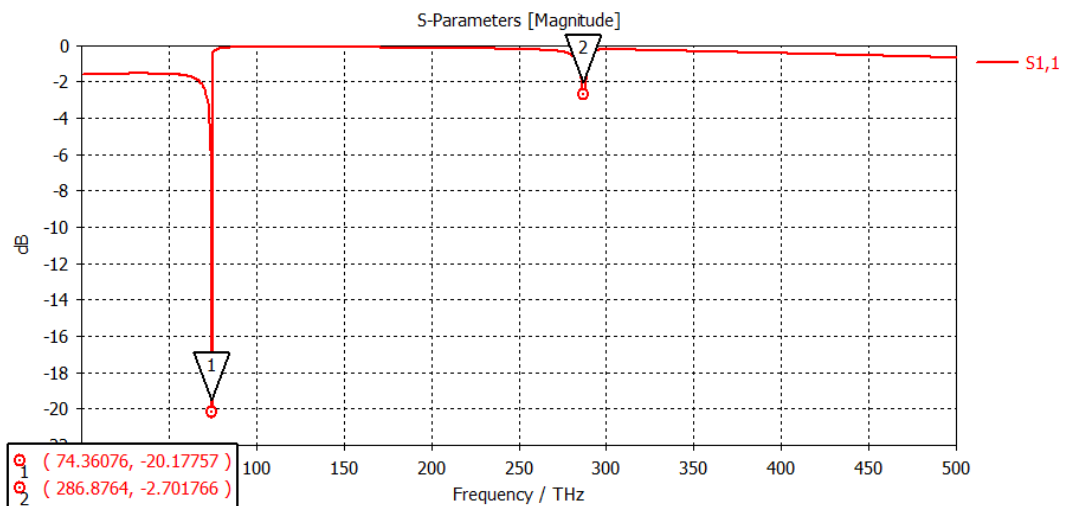


Figura 186. Modelo 14  $S_{11}$

### 4.14.3. Farfields para las frecuencia de resonancia

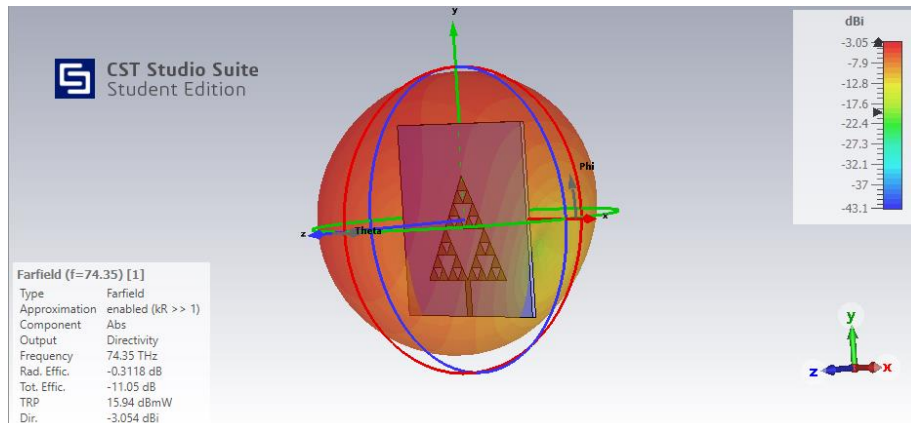


Figura 187. Modelo 14: Diagrama radiación 3D  $f_{r1}$

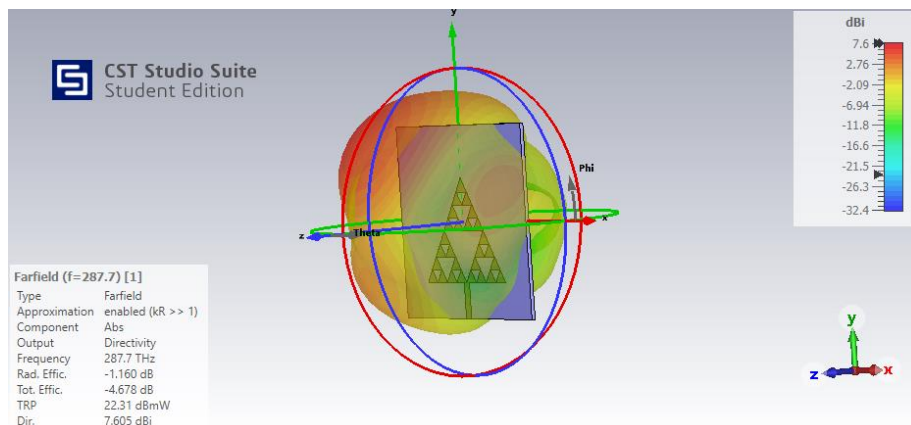


Figura 188. Modelo 14: Diagrama radiación 3D  $f_{r2}$

### 4.14.4. Directividad

Frecuencias (THz)		Directividad (dBi)
Fr1	74,35	-3,054
Fr2	287,95	7,605

Tabla 80. Modelo 14: Directividad



#### 4.14.5. Eficiencias de Radiación

Frecuencias (THz)		Rad.Effic (dB)	Rad.Effic (%)	Tot.Effic (dB)	Tot.Effic (%)
Fr1	74,35	-0,312	93,072	-11,050	7,852
Fr2	287,95	-1,160	76,560	-4,678	34,056

Tabla 81. Modelo 14: Eficiencias de Radiación

#### 4.14.6. Ganancia

Frecuencias (THz)		Gain (dB)
Fr1	74,35	-3,361
Fr2	287,95	6,438

Tabla 82. Modelo 14: Ganancia

#### 4.14.7. Ganancia Realizable

Frecuencias (THz)		Gain Realized (dB)
Fr1	74,35	-14,035
Fr2	287,95	2,916

Tabla 83. Modelo 14: Ganancia Realizable

#### 4.14.8. Potencia Total Radiada (TRP)

Frecuencias (THz)		TPR (3dBm)	TPR (13dBm)
Fr1	74,35	-8,048	1,952
Fr2	287,95	-1,678	8,322

Tabla 84. Modelo 14: Potencia Total Radiada

## 4.15 Resumen comparativo modelos

A continuación se detalla una tabla comparativa de los parámetros obtenidos en los distintos modelos diseñados y simulados:

ORDEN FRACTAL	MATERIALES	Modelo	Frecuencias (THz)	S11 (dB)	Directividad (dBi)	Rad.Effic (dB)	Rad.Effic (%)	Tot.Effic (dB)	Tot.Effic (%)	TPR (3dBm)	TPR (13dBm)	Gain (dB)	Gain Realized (dB)	
1	grafeno: uc = 0,1 eV silicio: e = 11,9 glass_epoxy: e = 4,4	sierpinski_1_grafeno_silicon_pec.cst	Fr1	44,90	-6,110	-7,888	-0,279	93,77	-15,510	2,82	-12,510	-2,510	-8,130	-23,210
			Fr2	196,60	-2,230	4,848	-1,750	66,83	-5,694	26,95	-2,694	7,310	3,059	-0,805
			Fr3	422,20	-10,870	9,645	-3,375	45,97	-3,741	42,26	-0,741	9,260	6,266	5,856
		sierpinski_1_grafeno_glass_epoxy_pec.cst	Fr1	74,35	-21,160	-3,020	-0,294	93,45	-11,270	7,46	-8,266	1,734	-3,314	-14,286
			Fr2	291,50	-2,990	7,870	-0,108	97,54	-4,127	38,66	-1,127	8,873	6,776	3,731
			Fr3	44,90	-6,270	-7,886	-0,489	89,34	-15,540	2,80	-12,540	-2,536	-8,300	-23,280
	grafeno: uc = 0,2 eV silicio: e = 11,9 glass_epoxy: e = 4,4	sierpinski_1_grafeno2_silicon_pec.cst	Fr1	44,90	-2,050	4,600	-2,074	95,34	-6,203	23,97	-3,203	6,797	2,540	-1,591
			Fr2	196,50	-7,640	10,070	-2,806	52,41	-4,232	37,74	-1,232	8,768	7,239	5,799
			Fr3	422,70	-7,640	10,070	-2,806	52,41	-4,232	37,74	-1,232	8,768	7,239	5,799
		sierpinski_1_grafeno2_glass_epoxy_pec.cst	Fr1	74,35	-21,360	-3,029	-0,391	91,40	-11,290	7,43	-8,288	1,712	-3,342	-14,290
			Fr2	292,30	-2,650	7,647	-1,465	71,37	-4,985	31,73	-1,985	8,015	6,195	2,675
			Fr3	44,90	-7,450	-7,920	-0,320	92,89	-15,930	2,55	-12,930	-2,932	-8,236	-23,737
2	grafeno: uc = 0,1 eV silicio: e = 11,9 glass_epoxy: e = 4,4	sierpinski_2_grafeno_silicon_pec.cst	Fr1	191,60	-3,870	4,924	-2,224	59,92	-4,507	35,42	-1,507	8,493	2,702	0,408
			Fr2	250,20	-0,330	7,184	-0,433	90,52	-11,760	6,67	-8,759	1,241	6,714	-4,529
			Fr3	74,35	-20,580	-3,084	-0,291	93,51	-11,070	7,92	-8,073	1,927	-3,370	-14,156
		sierpinski_2_grafeno_glass_epoxy_pec.cst	Fr1	287,95	-3,610	7,873	-1,128	77,13	-3,581	43,84	-0,581	9,419	6,744	4,232
			Fr2	44,90	-7,480	-7,919	-0,539	88,32	-15,930	2,55	-12,930	-2,930	-8,402	-23,811
			Fr3	191,60	-3,850	4,923	-2,179	60,55	-4,441	35,97	-1,441	8,559	2,748	0,475
	grafeno: uc = 0,2 eV silicio: e = 11,9 glass_epoxy: e = 4,4	sierpinski_2_grafeno2_silicon_pec.cst	Fr1	250,20	-0,290	7,195	-0,420	90,78	-11,710	6,75	-8,712	1,288	6,736	-4,469
			Fr2	74,35	-20,70	-3,072	-0,289	93,56	-11,080	7,80	-8,080	3,072	-3,365	-14,151
			Fr3	287,90	-3,68	7,828	-0,996	79,50	-3,423	45,47	-0,581	9,577	6,831	4,336
		sierpinski_2_grafeno4_silicon_pec.cst	Fr1	44,90	-7,780	-7,918	-2,244	58,65	-16,040	2,49	-13,040	-3,041	10,156	-23,927
			Fr2	192,10	-4,390	5,710	-0,207	95,35	-5,482	28,30	-2,482	7,518	5,506	0,213
			Fr3	74,35	-20,440	-3,130	-0,110	97,50	-11,070	7,82	-8,075	1,925	-3,240	-14,205
grafeno: uc = 0,43 eV silicio: e = 11,9 glass_epoxy: e = 4,4	sierpinski_2_grafeno4_glass_epoxy_pec.cst	Fr1	289,80	-2,102	8,398	-0,308	93,15	-4,500	35,48	-12,320	-2,317	8,062	3,899	
		Fr2	44,90	-7,620	-7,921	-0,285	93,64	-15,990	2,52	-12,990	-2,993	-8,205	-23,870	
		Fr3	193,60	-3,520	5,348	-1,704	67,55	-4,283	37,30	-1,283	8,717	3,588	0,947	
	sierpinski_3_grafeno_silicon_pec.cst	Fr1	74,35	-19,825	-3,029	-0,261	94,17	-10,970	8,00	-7,974	2,026	-3,290	-13,991	
		Fr2	287,70	-2,571	7,680	-1,009	79,27	-4,503	35,46	-1,503	8,497	6,641	3,177	
		Fr3	44,90	-7,650	-7,921	-0,467	89,81	-16,000	2,51	-13,000	-3,002	-8,399	-23,923	
grafeno: uc = 0,2 eV silicio: e = 11,9 glass_epoxy: e = 4,4	sierpinski_3_grafeno2_silicon_pec.cst	Fr1	194,10	-3,490	5,503	-1,649	68,41	-4,606	34,63	-1,606	8,394	3,853	0,871	
		Fr2	74,35	-20,170	-3,054	-0,312	93,07	-11,050	7,85	-8,048	1,952	-3,361	-14,035	
		Fr3	287,95	-2,702	7,605	-1,160	76,56	-4,678	34,06	-1,678	8,322	6,438	2,916	

Tabla 85. Resumen comparativo modelos

Como se puede observar no existe un modelo que destaque más en cuanto a prestaciones y tampoco existe demasiada variabilidad entre los distintos modelos. Aún así, se han seleccionado dos modelos que en base al valor del coeficiente de reflexión S<sub>11</sub> y la ganancia, se pueden considerar los dos mejores.

## 4.16 Mejor modelo orden 1

Se considera como mejor modelo de orden 1 de los obtenidos el siguiente del que se detallan las dimensiones los cortes de los diagramas de radiación:

MATERIALES	Modelo	Frecuencias (THz)	S11 (dB)	Directividad (dBi)	Rad.Effic (dB)	Rad.Effic (%)	Tot.Effic (dB)	Tot.Effic (%)	TPR (3dBm)	TPR (13dBm)	Gain (dB)	Gain Realized (dB)	
grafeno: uc= 0,1 eV silicio: e =11,9	sierpinski_1_grafeno_silicon_pec.cst	Fr1	44,90	-6,110	-7,888	-0,279	93,77	-15,510	2,82	-12,510	-2,510	-8,130	-23,210
		Fr2	196,60	-2,230	4,848	-1,750	66,83	-5,694	26,95	-2,694	7,310	3,059	-0,805
		Fr3	422,20	-10,870	9,645	-3,375	45,97	-3,741	42,26	-0,741	9,260	6,266	5,856

Tabla 86. Mejor modelo orden 1: Parámetros

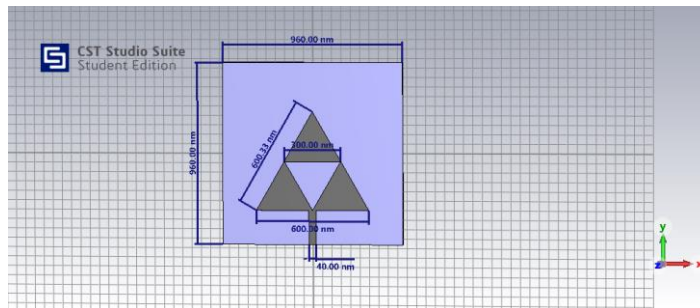


Figura 189. Mejor modelo orden 1: Componente

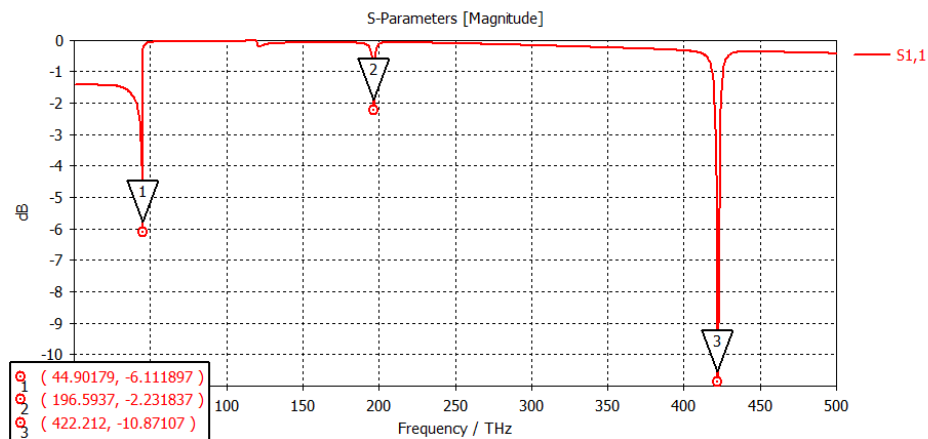


Figura 190. Mejor modelo orden 1: S<sub>11</sub>

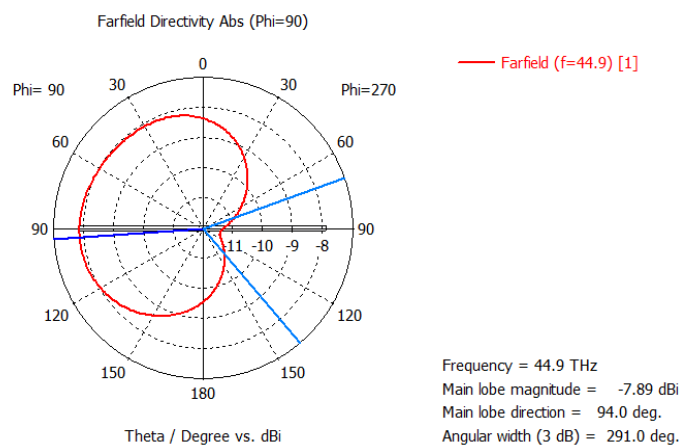


Figura 191. Mejor modelo orden 1: Farfield  $f_{r1}$

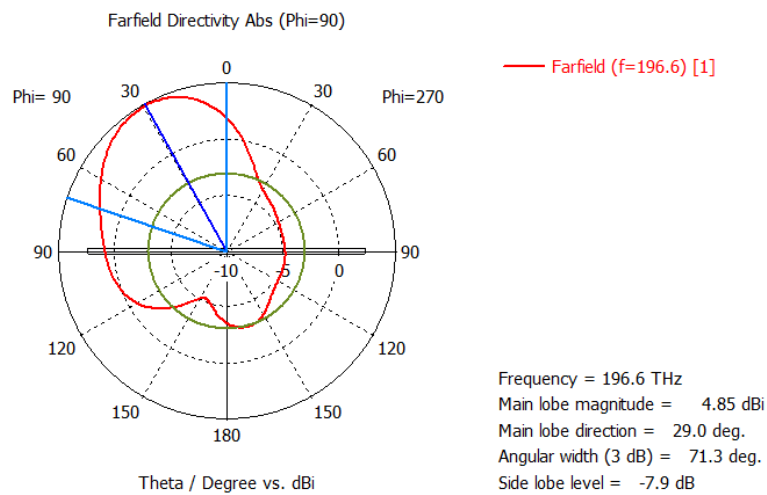


Figura 192. Mejor modelo orden 1: *Farfield*  $f_2$

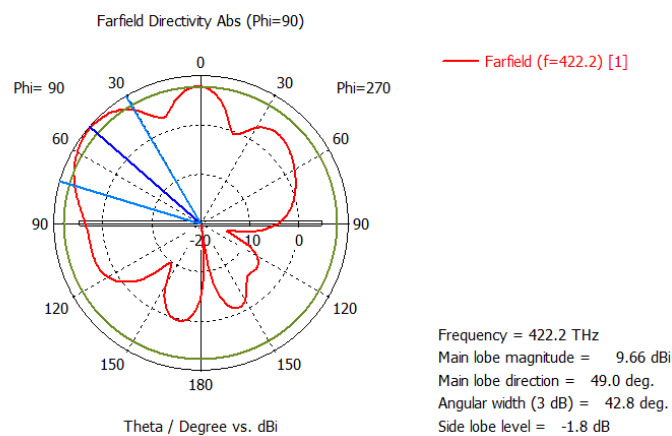


Figura 193. Mejor modelo orden 1: *Farfield*  $f_3$

Con respecto a este modelo es necesario comentar que el valor de  $S_{11}$  no es demasiado bueno y por esta razón se observa que el valor de la Eficiencia total es menor del esperado, un 2,82% para la primera frecuencia de resonancia.

Una posible solución a esta situación es conseguir una mejor adaptación a la primera frecuencia de resonancia  $f_r = 44,9$  THz. Esto se podría conseguir mediante una red de adaptación de impedancias o un transformador.

Así pues, se observa que la tercera frecuencia  $f_3 = 422,2$  THz da mejores valores para la eficiencia, con un valor del coeficiente de reflexión  $S_{11} = -10,87$  dB. Los valores de directividad y ganancia que proporciona también serían buenos:  $D = 9,6$  dB y  $G = 6,3$  dB. Como se puede observar, el valor de la ganancia realizable es de 5,86 dB, hecho que nos indica que a esta frecuencia la antena está mejor adaptada.

## 4.17 Mejor modelo orden 2

Se considera como mejor modelo de orden 2 de los obtenidos el siguiente del que se detalla más información:

MATERIALES	Modelo	Frecuencias (THz)	S11 (dB)	Directividad (dBi)	Rad.Effic (dB)	Rad.Effic (%)	Tot.Effic (dB)	Tot.Effic (%)	TPR (3dBm)	TPR (13dBm)	Gain (dB)	Gain Realized (dB)
grafeno: uc = 0,43 eV silicio: e = 11,9	sierpinski_2_grafeno 4_silicon_pec.cst	Fr1 Fr2	44,90 192,10	-7,780 -7,918	-2,244 5,710	59,649 -0,207	-16,040 -5,482	2,489 28,301	-13,040 -2,482	-3,041 7,518	10,156 5,506	-23,927 0,213

Tabla 87. Mejor modelo orden 2: Parámetros

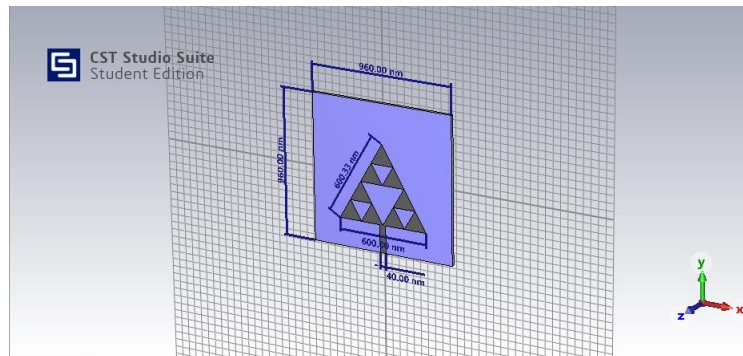


Figura 194. Mejor modelo orden 2: Componente

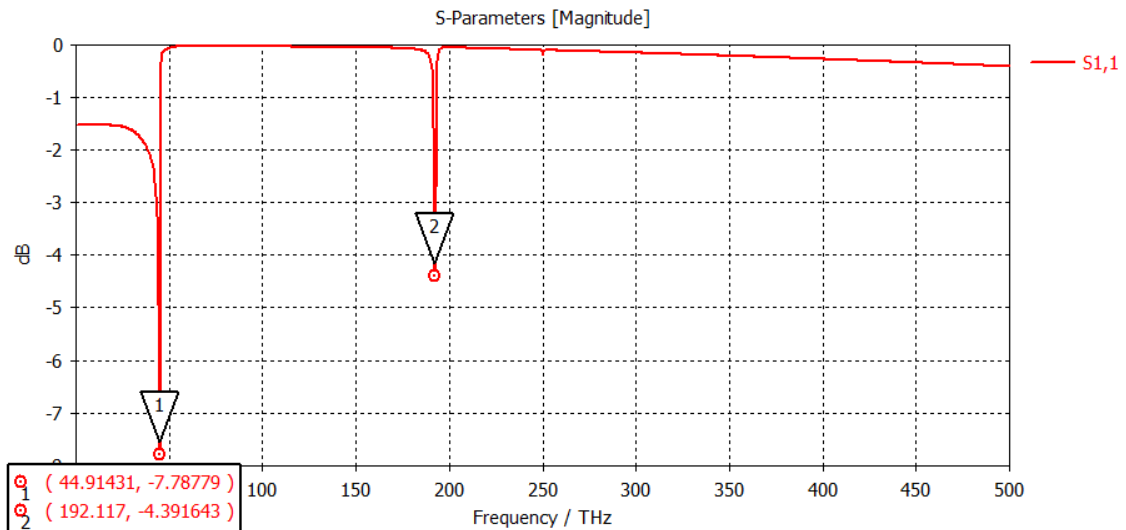


Figura 195. Mejor modelo orden 2: S<sub>11</sub>

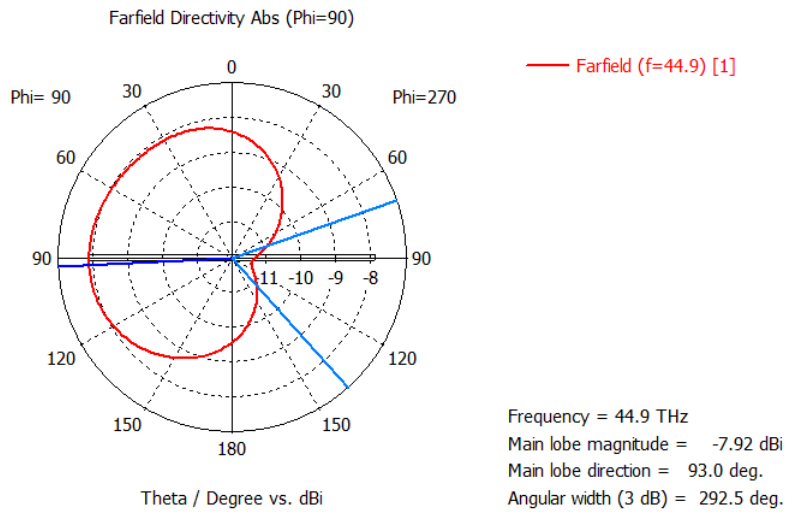


Figura 196. Mejor modelo orden 2: *Farfield*  $f_{r1}$

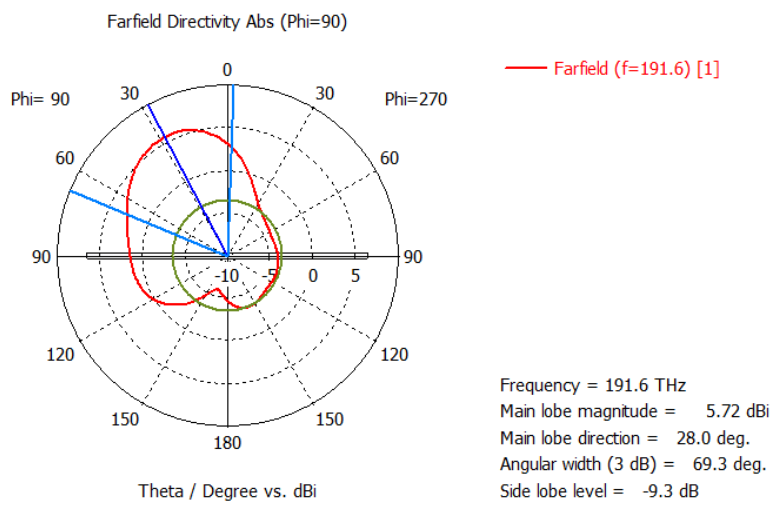


Figura 197. Mejor modelo orden 2: *Farfield*  $f_{r2}$

Al igual que con modelo anterior, es necesario puntualizar que se debería tratar de conseguir una mejor adaptación para la primera frecuencia de resonancia.

## 5. Conclusiones

Como conclusiones del presente trabajo, se han identificado determinados parámetros que al ser variados modifican los parámetros de diseño de la antena como son: frecuencia de resonancia, ganancia realizable, eficiencia de radiación y directividad.

A continuación se detalla cada uno de ellos:

1) Potencial químico del grafeno:

Con las variación del potencial químico del grafeno se observa que aumentan ligeramente las frecuencias de resonancia para un mismo sustrato y un mismo orden de fractal. Esta conclusión va en línea con los trabajos previos investigados en los que se demuestra que el cambio de potencial químico tiene una implicación en la frecuencia de resonancia de la antena.

2) Permitividad eléctrica del sustrato:

Con respecto a la variación de la permitividad eléctrica del sustrato, se observa que cuanto mayor es esta, menores son las frecuencias de resonancia.

3) Orden del fractal:

Conforme aumenta el orden del fractal, para un mismo sustrato y un mismo material de grafeno, las frecuencias de resonancia bajan ligeramente ambos sustratos.

4) Sustrato silicio:

Proporciona mayores ganancias realizables en la antena.

5) Sustrato fibra de vidrio:

Proporciona ligeramente mayores eficiencias de radiación en la antena.

6) Ganancia:

Se observan mejores ganancias para conforme aumenta el orden del fractal.

7) Directividad:

Mejores directividades para las segunda y tercera frecuencias de resonancia que para la primera.

Se hace necesario reflexionar sobre si se ha conseguido lograr los objetivos planteados inicialmente y la respuesta es afirmativa. El presente trabajo pretendía llegar a obtener diseños funcionales de antenas fractales de grafeno y sobre todo proveer de las herramientas para poder modificar de forma rápida y ágil los distintos parámetros de construcción de un determinado modelo.

Puesto que en todo momento se ha utilizado software con versión de estudiantes, se han presentado diversas limitaciones de la licencia de CST a nivel de simulación y diseño de Macros en la propia interfaz del programa, que en parte se han podido subsanar con la automatización desde Matlab. En el caso de las limitaciones de simulación, no se ha podido llegar a simular modelos fractales de orden 5 por ejemplo:

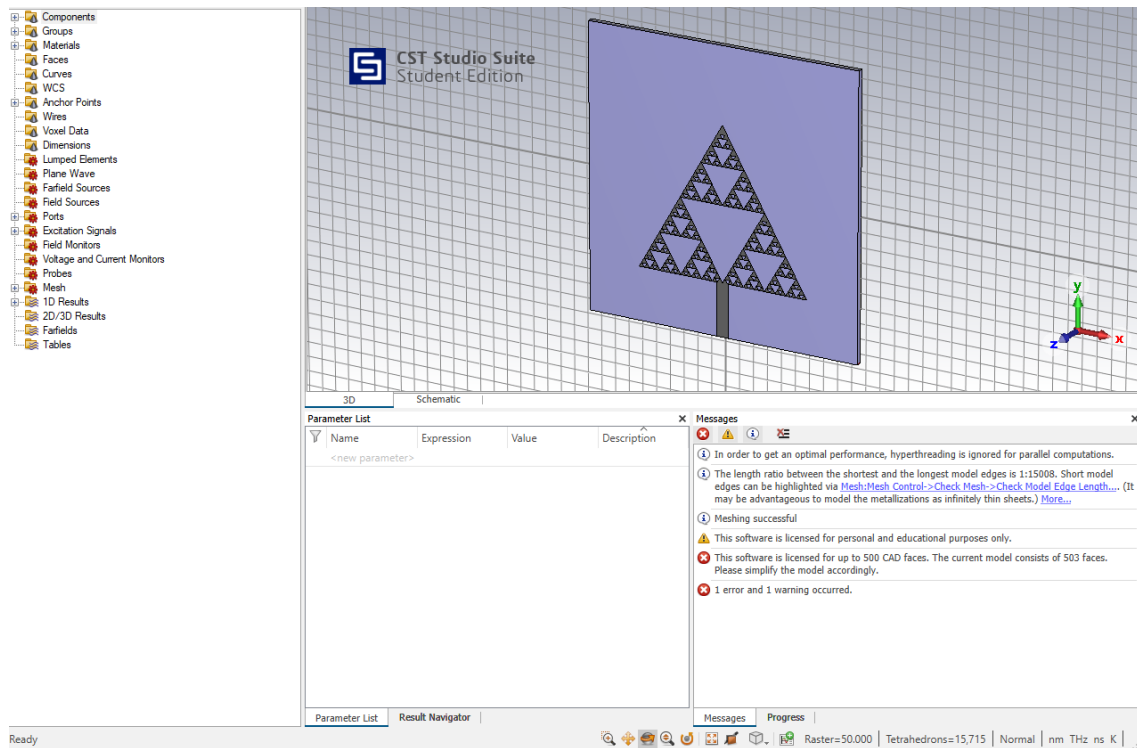


Figura 198. Modelo fractal orden 5



## 6. Líneas futuras

Las líneas de trabajo futuro son numerosas puesto que se pueden explorar distintos diseños de antenas fractales y con diversas combinaciones de materiales de construcción basándose en el presente trabajo como punto de partida.

Sin modificación del trabajo actual se puede partir de los scripts de Matlab creados y sin realizar ningún cambio en los mismos, generar las distintas combinaciones de antena fractal triángulo de Sierpiński para diferentes órdenes, distintos materiales del parche y de sustrato. Están ya configurados y creados los siguientes materiales:

- Material parche: cobre, grafeno ( $\mu_c=0,1$  eV), grafeno2 ( $\mu_c=0,2$  eV), grafeno4 ( $\mu_c=0,43$  eV) y PEC.
- Sustrato: Dióxido de Silicio, Fibra de Vidrio y FR4 estándar.
- Plano de masa: PEC y Oro.

Si se tuviese acceso a una versión de CST Studio Suite comercial se podrían realizar simulaciones de fractales de orden 5 o superior. Con la versión de CST con licencia de estudiante utilizada se presentaban ciertas limitaciones con la simulación y con el diseño de Macros.

Por otro lado, con la modificación en la función `fractal_sierpinski` de algunos parámetros, se podría crear la misma antena para distintos grosores del sustrato y distintas dimensiones del parche. También se pueden crear otras formas fractales, siempre y cuando se programe en Matlab el código que genere el componente en CST. Los entregables otorgan un punto de partida para diseñar antenas con formas fractales diversas.

Se ha elegido un diseño de antena que trabajase en los terahercios, pero con la modificación de las dimensiones de esta y del material del parche, se podría trabajar en las frecuencias de los gigahercios.

Se han creado tres tipos de material grafeno con distintos valores de potencial químico y en el trabajo se indica la forma de crear otros materiales grafeno con los que investigar.

También es interesante mencionar que el proyecto facilita de una forma rápida y eficiente la comprobación del funcionamiento de un determinado diseño.

Por último, hay que mencionar que dentro de los usos comerciales de las antenas de grafeno en la banda de los terahercios, resulta muy interesante el posible uso comercial en aplicaciones de imagen médica gracias a la característica reconfigurable o sintonizable de las antenas de grafeno mediante la variación del potencial químico.

## 7. Glosario

**API (Application Programming Interface):** Interfaz de programación de aplicaciones, es un conjunto de funciones y procedimientos que permite que los programas puedan comunicarse entre ellos.

**Grafeno:** Es un nanomaterial con una conductividad eléctrica muy alta y que aplicado a una antena de dimensiones de nanómetros permite la propagación de las ondas electromagnéticas en las frecuencias de terahercios.

**Polaritón:** Cuasipartícula resultado del fuerte acoplamiento de ondas electromagnéticas con la onda de polarización eléctrica o magnética

**Plasmón:** Cuasipartícula que describe al campo EM producido por las oscilaciones colectivas de los electrones de conducción en un material. Los plasmones de superficie son aquellos que están confinados a las superficies y que forman un polaritón cuando interactúan con la luz.

**PPP (Planar Plasmonic Patch):** Antena de parche plasmónico de grafeno.

**SPP (Polaritón de plasmón de superficie):** Son ondas electromagnéticas que se desplazan entre un metal-dieléctrico o entre metal-aire en la región del espectro electromagnético de los infrarrojos o visible. Son ondas que indican movimiento de carga en el metal (plasmón de superficie) y ondas electromagnéticas en el aire o dieléctricas (polaritón).

## 8. Bibliografía

[1] MANDELBROT, Benoit. *La geometría fractal de la naturaleza*. Barcelona: Tusquets Editores, 1997. ISBN 84-8310-549-7

### **Papers Fractales:**

[2] ANGUERA, Jaume, et al. *Fractal shaped antennas: A review*. Encyclopedia of RF and microwave engineering, 2005. ISBN 0471654485.

[3] ANGUERA, Jaume, et al. Fractal antennas: An historical perspective. *Fractal and Fractional*, 2020, vol. 4, no 1, p. 3.

[4] J. Anguera, E. Martinez-Ortigosa, C. Puente, C. Borja and J. Soler, "Broadband Triple-Frequency Microstrip Patch Radiator Combining a Dual-Band Modified Sierpinski Fractal and a Monoband Antenna," in *IEEE Transactions on Antennas and Propagation*, vol. 54, no. 11, pp. 3367-3373, Nov. 2006, doi: 10.1109/TAP.2006.884209.

[5] Jaume Anguera, E. Martinez, C. Puente, C. Borja and J. Soler, "Broad-band dual-frequency microstrip patch antenna with modified Sierpinski fractal geometry," in *IEEE Transactions on Antennas and Propagation*, vol. 52, no. 1, pp. 66-73, Jan. 2004, doi: 10.1109/TAP.2003.822433.

[6] J. Anguera, C. Puente, C. Borja and J. Soler, "Dual-Frequency Broadband-Stacked Microstrip Antenna Using a Reactive Loading and a Fractal-Shaped Radiating Edge," in *IEEE Antennas and Wireless Propagation Letters*, vol. 6, pp. 309-312, 2007, doi: 10.1109/LAWP.2007.891523.

[7] J. Anguera et al., "Metallized foams for fractal-shaped microstrip antennas," in *IEEE Antennas and Propagation Magazine*, vol. 50, no. 6, pp. 20-37, Dec. 2008, doi: 10.1109/MAP.2008.4772718.

[8] ANGUERA, Jaume, et al. Metallized foams for antenna design: application to fractal-shaped sierpinski-carpet monopole. *Progress In Electromagnetics Research*, 2010, vol. 104, p. 239-251.

[9] ANGUERA, Jaume, et al. The fractal Hilbert monopole: a two-dimensional wire. *Microwave and Optical technology letters*, 2003, vol. 36, no 2, p. 102-104.

[10] ANGUERA, Jaume, et al. Small and high-directivity bow-tie patch antenna based on the Sierpinski fractal. *Microwave and Optical Technology Letters*, 2001, vol. 31, no 3, p. 239-241.

[11] ANGUERA, Jaume, et al. High-directivity microstrip antenna with Mandelbrot fractal boundary. *IET Microwaves, Antennas & Propagation*, 2018, vol. 12, no 4, p. 569-575.

[12] Lan Wang, Jianguo Yu, Tangyao Xie, Kun Bi, "A Novel Multiband Fractal Antenna for Wireless Application", *International Journal of Antennas and Propagation*, vol. 2021, Article ID 9926753, 9 pages, 2021. <https://doi.org/10.1155/2021/9926753>

[13] Xiaoying Ran, Zhen Yu, Tangyao Xie, Yao Li, Xiuxia Wang, Peng Huang, "A Novel Dual-Band Binary Branch Fractal Bionic Antenna for Mobile Terminals", *International Journal of Antennas and Propagation*, vol. 2020, Article D 6109093, 9 pages, 2020. <https://doi.org/10.1155/2020/6109093>

### **Papers Grafeno**

[14] Ju, L., Geng, B., Horng, J. *et al.* Graphene plasmonics for tunable terahertz metamaterials. *Nature Nanotech* 6, 630–634 (2011). <https://doi.org/10.1038/nnano.2011.146>

[15] Li, XS *et al.* Large-Area Synthesis of High-Quality and Uniform Graphene Films on Copper Foils. *Ciencia* 324 , 1312–1314 (2009). DOI: 10.1126/science.1171245

### **Papers Antenas Grafeno**

[16] Llatser, I.; Kremers, C.; Cabellos-Aparicio, A.; Jornet, J.M.; Alarcón, E.; Chigrin, D.N. Graphene-based nano-patch antenna for terahertz radiation. *Photonics and Nanostructures - Fundamentals and Applications* 2012, 10, 353-358.

[17] Mohamed Karim Azizi, Mohamed Amin Ksiksi, Hosni Ajlani, and Ali Gharsallah, "Terahertz Graphene-Based Reconfigurable Patch Antenna," *Progress In Electromagnetics Research Letters*, Vol. 71, 69-76, 2017.

doi:[10.2528/PIERL17081402](https://doi.org/10.2528/PIERL17081402)

<http://www.jpier.org/PIERL/pier.php?paper=17081402>

[18] Aymen Hlali, Zied Houaneb, and Hassen Zairi, "Dual-Band Reconfigurable Graphene-Based Patch Antenna in Terahertz Band: Design, Analysis and Modeling Using WCIP Method," *Progress In Electromagnetics Research C*, Vol.87, 213-226, 2018.

doi:[10.2528/PIERC18080107](https://doi.org/10.2528/PIERC18080107)

<http://www.jpier.org/PIERC/pier.php?paper=18080107>

[19] Hamza Ben Krid, Zied Houaneb, and Hassen Zairi, "Reconfigurable Graphene Annular Ring Antenna for Medical and Imaging Applications," *Progress In Electromagnetics Research M*, Vol. 89, 53-62, 2020.

doi:[10.2528/PIERM19110803](https://doi.org/10.2528/PIERM19110803)

<http://www.jpier.org/PIERM/pier.php?paper=19110803>

[20] Muhammad Irfan Khattak, Muhammad Anab, and Nabeel Muqarrab, "A Duo of Graphene-Copper Based Wideband Planar Plasmonic Antenna Analysis for Lower Region of Terahertz (THz ) Communications," *Progress In Electromagnetics Research C*, Vol. 111, 83-96, 2021.

doi:[10.2528/PIERC21010603](https://doi.org/10.2528/PIERC21010603)

<http://www.jpier.org/PIERC/pier.php?paper=21010603>

[21] J. M. Jornet and I. F. Akyildiz, "Graphene-based Plasmonic Nano-Antenna for Terahertz Band Communication in Nanonetworks," in *IEEE Journal on Selected Areas in Communications*, vol. 31, no. 12, pp. 685-694, December 2013, doi: 10.1109/JSAC.2013.SUP2.1213001.

[22] Zhang, Jinnan, Shijie Tao, Xin Yan, Xia Zhang, Jinxuan Guo, and Zhiqiang Wen. 2021. "Dual-Frequency Polarized Reconfigurable Terahertz Antenna Based on Graphene Metasurface and TOPAS" *Micromachines* 12, no. 9: 1088. <https://doi.org/10.3390/mi12091088>

## **Papers Fractales Grafeno**

[23] Y. Xu, Y. Xu, J. Hu and W. Yin, "Design of a novel reconfigurable Sierpinski fractal graphene antenna operating at THz band," *2013 IEEE Antennas and Propagation Society International Symposium (APSURS)*, 2013, pp. 574-575, doi: 10.1109/APS.2013.6710947.

[24] BORETTI, A., ROSA, L., BLACKLEDGE, J. y CASTELLETTO, S., 2020. A Preliminary Study of a Graphene Fractal Sierpinski Antenna. IOP conference series. Materials Science and Engineering, vol. 840, no. 1, pp. 12003-. ISSN 1757-8981. DOI 10.1088/1757-899X/840/1/012003.

## **Documentación CST y Matlab:**

CST Computer Simulation Technology. *Antenna Simulation. Application Note CST ESTUDIO SUITE*. Kra, Jeb / v1.0 / 02. July 2012. [www.cst.com](http://www.cst.com)

[CST Studio Suite - High Frequency Simulation.pdf \(mit.edu\)](#)

<https://es.mathworks.com/matlabcentral/fileexchange/72228-tcstinterface-cst-studio-suite-to-matlab-interface>

<https://habr.com/ru/post/325012/>

<https://habr.com/ru/post/336396/>

<https://doi.org/10.5281/zenodo.1237969>

## **Documentación de CST OLE Automation server:**

[https://space.mit.edu/RADIO/CST\\_online/mergedProjects/VBA\\_3D/whnjs.htm](https://space.mit.edu/RADIO/CST_online/mergedProjects/VBA_3D/whnjs.htm)

## **Plantillas modificadas:**

[Free PowerPoint Templates Design \(free-powerpoint-templates-design.com\)](#)

<https://www.vertex42.com/ExcelTemplates/simple-gantt-chart.html>

## **Fuente de las imágenes**

*Imagen grafeno*: Copyright: © AlexanderAIUS, CC BY-SA 3.0  
<<https://creativecommons.org/licenses/by-sa/3.0/>>, via Wikimedia Commons

*Imagen portadas*: Copyright: © ChaosEmporium -  
<http://www.redbubble.com/people/ChaosEmporium>

## 9. Anexo I

### Código Matlab

Los scripts adjuntos se corresponden con las funciones implementadas en Matlab para lanzar instrucciones en CST y generar los modelos de forma automática.

### Función principal

```
fractal_sierpinski(it, materialpatch,substrato, groundplane)
```

Como parámetros de entrada tiene: el número de iteraciones, material, substrato y plano de masa.

La función genera la antena de parche fractal de Sierpiński de orden n en CST, con el material, substrato y plano de masa indicados. También genera un nuevo proyecto de CST con la parametrización: unidades, rango de frecuencia, dimensiones del elemento, puerto, monitores y realiza la simulación con el “*FD Solver*”.

```
function fractal_sierpinski(it,materialpatch,substrato,groundplane)

% Script to create a new CST project to simulate a Fractal Sierpinski
% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Material patch: cobre, grafeno, grafeno2, grafeno4, pec
% Substrato: silicon, glass_epoxy, fr4
% Groundplane: pec, gold

% Start a new CST project
cst = actxserver('CSTStudio.application');
mws = cst.invoke('NewMWS');

% Define units
geometry = 'nm';
frecuency = 'THz';
units(mws, geometry, frecuency);

% Define Frequency Range
f1 = 1;
f2 = 500;
frecuency_range(mws, f1, f2);

% Antenna microstrip dimensions
% PATCH: W = Width - L = Lenght - ht = height
W = 600;
L = 600;
hp = 0.04;

%FEEDLINE: W = Width - L = Lenght
Wf = 20;
%GROUNDPLANE: Wg = Width - Lg = Lenght
Lg = 2*L;
Wg = 2*W;
%SUBSTRATE: hs = height
hs=20;

% Fractal dimensions e iterations
```

```

% L is the triangle side too
namecount=0;
side = L;
it=it-1;

% Initialization iteration=0 - We paint the first triangle
sop_x = [0 L/2 L];
sop_y = [0 sqrt(3*(L^2)/4) 0];
triangle(mws,sop_x,sop_y,namecount,hp);
namecount = namecount + 1;
% Iteration>0
if (it>-1)
    side=side/2;
    escalate=3;
    it0_x = [side/2 side 3*side/2];
    it0_y = [sqrt(3*(side^2)/4)-escalate 0 sqrt(3*(side^2)/4)-escalate];
    triangle(mws,it0_x,it0_y,namecount,hp);
    namecount = namecount + 1;
    % Initialization the triangles coordinates
    coor{1,1} = it0_x;
    coor{2,1} = it0_y;
    % Escalate
    escalate = 0;
    if (it == 1)
        escalate = 2;
    elseif (it == 2)
        escalate = 2;
    elseif (it > 2)
        escalate = 3;
    end

    % Loop to iterations
    for i = 1:1:it
        temp = {};
        side = side/2;
        for n = 1:1:size(coor,2)
            h = round(sqrt(3*(side^2)/4))-escalate;
            % Right triangle
            x_r = [coor{1,n}(2)+side/2 coor{1,n}(2)+side coor{1,n}(2)+3*side/2];
            y_r = [coor{2,n}(2)+h coor{2,n}(2) coor{2,n}(2)+h];
            % Upper triangle
            x_u = [coor{1,n}(1)+side/2 coor{1,n}(1)+side coor{1,n}(1)+3*side/2];
            y_u = [coor{2,n}(1)+h coor{2,n}(1) coor{2,n}(1)+h];
            % Left triangle
            x_l = [coor{1,n}(2)-3*side/2 coor{1,n}(2)-side coor{1,n}(2)-side/2];
            y_l = [coor{2,n}(2)+h coor{2,n}(2) coor{2,n}(2)+h];
            % New triangles coordinates storage
            l = size(temp,2);
            temp{1,l+1} = x_r; temp{2,l+1} = y_r;
            temp{1,l+2} = x_u; temp{2,l+2} = y_u;
            temp{1,l+3} = x_l; temp{2,l+3} = y_l;
            % Paint relative triangles
            triangle(mws,x_r,y_r,namecount,hp);
            namecount = namecount + 1;
            triangle(mws,x_u,y_u,namecount,hp);
            namecount = namecount + 1;
            triangle(mws,x_l,y_l,namecount,hp);
            namecount = namecount + 1;
        end
        % Cleaning up temporal cells
        coor = temp;
        clear('temp');
    end
end

% Reset View to Default
invoke(mws,'ResetViewToStructure');

% Create feedline by polygon
% Feedline Wf=20
xr = [L/2-Wf L/2+Wf L/2+Wf L/2-Wf];
yr = [0 0 -L/4-2*Wf+10 -L/4-2*Wf+10];

```

```

% hr May be the same as hp = 0.04;
rectanglefeed(mws,xr,yr,hp)

% Boolean Addition t0 + feedline
component1 = 'component:t0';
component2 = "component:feedline1";
boolean_add(mws,component1,component2);

% Boolean Substraction
for j = 1:1:namecount-1
    component1 = 'component:t0';
    component2 = "component:t"+int2str(j);
    boolean_subs(mws,component1,component2);
end

% Translate triangle
name='component:t0';
type='Translate';
transform(mws,name,type,-L/2,-L/2,hp+hs);

% Rename t0 -> sierpinski
Solid = invoke(mws,'Solid');
invoke(Solid,'Rename','component:t0_1','component:sierpinski');

% Delete t0_1
Solid = invoke(mws,'Solid');
invoke(Solid,'Delete','component:t0');

% Assign material
switch materialpatch
    case 'cobre'
        mat_cooperpurelossy(mws);
        Solid = invoke(mws,'Solid');
        invoke(Solid,'ChangeMaterial','component:sierpinski','Copper
(annealed)');
    case 'grafeno'
        mat_graphene_eps(mws);
        Solid = invoke(mws,'Solid');
        invoke(Solid,'ChangeMaterial','component:sierpinski','Graphene_Eps');
    case 'grafeno2'
        mat_graphene2_eps(mws);
        Solid = invoke(mws,'Solid');
        invoke(Solid,'ChangeMaterial','component:sierpinski','Graphene2_Eps');
    case 'grafeno4'
        mat_graphene4_eps(mws);
        Solid = invoke(mws,'Solid');
        invoke(Solid,'ChangeMaterial','component:sierpinski','Graphene4_Eps');
    otherwise
        disp('PEC')
end

% Create groundplane
Name = 'groundplane';
component = 'component';
material = 'PEC';
Xrange = [-0.4*Wg 0.4*Wg];
Yrange = [-0.4*Lg 0.4*Lg];
Zrange = [-hp/2 hp/2];
brick(mws, Name, component, material, Xrange, Yrange, Zrange);

% Assign material
switch groundplane
    case 'gold'
        ground_gold(mws);
        Solid = invoke(mws,'Solid');
        invoke(Solid,'ChangeMaterial','component:groundplane','Gold');
    otherwise
        %disp('PEC')
end

% Create substrate
subs_FR4lossy(mws);
Name = 'substrate';

```



```

        component = 'component';
        material = 'FR-4 (lossy)';
        Xrange = [-0.4*Wg 0.4*Wg];
        Yrange = [-0.4*Lg 0.4*Lg];
        Zrange = [hp hp+hs];
        brick(mws, Name, component, material, Xrange, Yrange, Zrange);

% Assign material
switch substrato
    case 'glass_epoxy'
        subs_glassepoxy(mws);
        Solid = invoke(mws, 'Solid');
        invoke(Solid, 'ChangeMaterial', 'component:substrate', 'FR-4 (glass
epoxy)');

    case 'silicon'
        subs_siliconlossy(mws);
        Solid = invoke(mws, 'Solid');
        invoke(Solid, 'ChangeMaterial', 'component:substrate', 'Silicon (lossy)');
    otherwise
        %disp('FR4')
end

% Print model properties
fprintf("Fractal Sierpinski\n");
fprintf("Order: "+int2str(it+1)+"\n"+"Patch Material:
"+materialpatch+"\n"+"Substrate: "+substrato+"\n"+"Ground Plane: "+groundplane+"\n");

% We assign a pickface in substrate
name = 'substrate';
number = 3;
pickface(mws,name,number)

% Port to simulate THz
portnumber = 1;
Xrange = [-480 480];
Yrange = [-480 -480];
Zrange = [0 20];
XrangeAdd =[0 0];
YrangeAdd =[0 0];
ZrangeAdd =[0 0];
coordinates = 'Picks';
orientation = 'positive';
waveguideport(mws,portnumber, Xrange, Yrange, Zrange, XrangeAdd, YrangeAdd,
ZrangeAdd,coordinates,orientation)

% Print port properties
fprintf("Wave Guide Port to the Simulation\n");

% Reset View to Default
invoke(mws, 'ResetViewToStructure');

% Resonance Frequencies
switch substrato
    case 'silicon'
        switch it
            case -1 %Fractal orden 0
                %mesh(mws);
                fr1 = 44.9;
                fr2 = 193.1;
                % Define the monitors
                addmonitor(mws, 'Efield', fr1);
                addmonitor(mws, 'Hfield', fr1);
                addmonitor(mws, 'Farfield', fr1);
                addmonitor(mws, 'Efield', fr2);
                addmonitor(mws, 'Hfield', fr2);
            case 0 %Fractal orden 1
                fr1 = 44.9;
                fr2 = 196.6;
                fr3 = 422.2;
                % Define the monitors
                addmonitor(mws, 'Efield', fr1);
                addmonitor(mws, 'Hfield', fr1);
                addmonitor(mws, 'Farfield', fr1);

```

```

        addmonitor(mws, 'Efield', fr2);
        addmonitor(mws, 'Hfield', fr2);
        addmonitor(mws, 'Farfield', fr2);
        addmonitor(mws, 'Efield', fr3);
        addmonitor(mws, 'Hfield', fr3);
        addmonitor(mws, 'Farfield', fr3);
    case 1 %Fractal orden 2
        fr1 = 44.9;
        fr2 = 191.6;
        fr3 = 250.3;
        % Define the monitors
        addmonitor(mws, 'Efield', fr1);
        addmonitor(mws, 'Hfield', fr1);
        addmonitor(mws, 'Farfield', fr1);
        addmonitor(mws, 'Efield', fr2);
        addmonitor(mws, 'Hfield', fr2);
        addmonitor(mws, 'Farfield', fr2);
        addmonitor(mws, 'Efield', fr3);
        addmonitor(mws, 'Hfield', fr3);
        addmonitor(mws, 'Farfield', fr3);
    case 2 %Fractal orden 3
        fr1 = 44.9;
        fr2 = 193.6;
        % Define the monitors
        addmonitor(mws, 'Efield', fr1);
        addmonitor(mws, 'Hfield', fr1);
        addmonitor(mws, 'Farfield', fr1);
        addmonitor(mws, 'Efield', fr2);
        addmonitor(mws, 'Hfield', fr2);
        addmonitor(mws, 'Farfield', fr2);
end
case 'glass_epoxy'
    switch it
    case -1 %Fractal orden 0
        fr1 = 44.9;
        fr2 = 193.1;
        % Define the monitors
        addmonitor(mws, 'Efield', fr1);
        addmonitor(mws, 'Hfield', fr1);
        addmonitor(mws, 'Farfield', fr1);
        addmonitor(mws, 'Efield', fr2);
        addmonitor(mws, 'Hfield', fr2);
        addmonitor(mws, 'Farfield', fr2);
    case 0 %Fractal orden 1
        fr1 = 74.3;
        switch materialpatch
        case 'grafeno' %Material grafeno1
            fr2 = 291.5;
            % Define the monitors
            addmonitor(mws, 'Efield', fr1);
            addmonitor(mws, 'Hfield', fr1);
            addmonitor(mws, 'Farfield', fr1);
            addmonitor(mws, 'Efield', fr2);
            addmonitor(mws, 'Hfield', fr2);
            addmonitor(mws, 'Farfield', fr2);
        case 'grafeno2' %Material grafeno2
            fr2 = 292.3;
            % Define the monitors
            addmonitor(mws, 'Efield', fr1);
            addmonitor(mws, 'Hfield', fr1);
            addmonitor(mws, 'Farfield', fr1);
            addmonitor(mws, 'Efield', fr2);
            addmonitor(mws, 'Hfield', fr2);
            addmonitor(mws, 'Farfield', fr2);
        end
    case 1 %Fractal orden 2
        fr1 = 74.35;
        fr2 = 287.9;
        % Define the monitors
        addmonitor(mws, 'Efield', fr1);
        addmonitor(mws, 'Hfield', fr1);
        addmonitor(mws, 'Farfield', fr1);
        addmonitor(mws, 'Efield', fr2);
        addmonitor(mws, 'Hfield', fr2);

```

```

        addmonitor(mws, 'Farfield', fr2);
    case 2 %Fractal orden 3
        fr1 = 74.35;
        fr2 = 287.7;
        % Define the monitors
        addmonitor(mws, 'Efield', fr1);
        addmonitor(mws, 'Hfield', fr1);
        addmonitor(mws, 'Farfield', fr1);
        addmonitor(mws, 'Efield', fr2);
        addmonitor(mws, 'Hfield', fr2);
        addmonitor(mws, 'Farfield', fr2);
    end
end

%Save the project as a diferent name

invoke(mws, 'SaveAs', "C:\UOC\TFG\CST_Matlab_TFG\sierpinski_"+int2str(it+1)+"_"+materialpatch
+"_"+substrato+"_"+groundplane+".cst", 'true');

% Simulate by Frequency Domain Solver
fdsolver(mws);

invoke(mws, 'Save');

end

```

## Funciones adicionales

`addmonitor.m`: función para definir el e-field, h-field y farfield monitor.

```

function addmonitor(mws, montype, frequency)

% Function to define e-field, h-field and farfield monitors

    switch montype
        case 'efield'
            name = ['e-field', ' (f=', num2str(frequency), ')'];
        case 'hfield'
            name = ['h-field', ' (f=', num2str(frequency), ')'];
        otherwise
            name = [montype, ' (f=', num2str(frequency), ')'];
    end

    mon = sprintf(['With Monitor\n', ...
        '.Reset\n', ...
        '.Name "%s"\n', ...
        '.Dimension "Volume"\n', ...
        '.Domain "Frequency"\n', ...
        '.FieldType "%s"\n', ...
        '.Frequency "%f"\n', ...
        '.UseSubVolume "False"\n', ...
        '.Create\n', ...
        'End With'], ...
        name, montype, frequency);

    invoke(mws, 'Monitor', mon);
    invoke(mws, 'AddToHistory', ['define field monitor: ', name], mon);

end

```

`boolean_add.m`: función para realizar la adición booleana.

```

function boolean_add(mws,component1,component2)

% Function to do boolean addition

Solid = invoke(mws,'Solid');
invoke(Solid,'Add',component1,component2);

end

```

`boolean_subs.m`: función para realizar la substracción booleana.

```

function boolean_subs(mws,component1,component2)

% Function to do boolean substration

Solid = invoke(mws,'Solid');
invoke(Solid,'Subtract',component1,component2);

end

```

`brick.m`: función para crear un bloque.

```

function brick(mws, name, component, material, Xrange, Yrange, Zrange)

% Function to create a brick

brick = invoke(mws,'Brick');
invoke(brick,'Reset');
invoke(brick,'Name',name);
invoke(brick,'component',component);
invoke(brick,'Material',material);
invoke(brick,'Xrange',int2str(Xrange(1)),int2str(Xrange(2)));
invoke(brick,'Yrange',int2str(Yrange(1)),int2str(Yrange(2)));
invoke(brick,'Zrange',int2str(Zrange(1)),int2str(Zrange(2)));
invoke(brick,'Create');
release(brick);

end

```

`fdsolver.m`: función para configurar el tipo de *solver* y lanzar simulación.

```

function fdsolver(mws)

% Function to set the solver type and run the simulation

% Set the Solver
solvertype = invoke(mws,'ChangeSolverType','HF Frequency Domain');
fprintf('Solver Type: HF Frequency Domain\n')

% Run simulation
fprintf('Simulation Running...\n')
FDSolver = invoke(mws,'FDSolver');
type = sprintf(['ChangeSolverType "HF Frequency Domain"']);
invoke(mws,'AddToHistory','ChangeSolverType',type);
invoke(FDSolver,'Start');
fprintf('Simulation Finished\n');

end

```

`frecuency_range.m`: función para seleccionar el rango de frecuencias.

```

function frequency_range(mws,f1,f2)

% Frequency range
range = sprintf(['Solver.FrequencyRange "%f", "%f"'],f1,f2);
solver = invoke(mws,'Solver');
% Add to History
invoke(mws,'AddToHistory',['frequency range: f1 = ',int2str(f1)],range);
invoke(mws,'AddToHistory',['frequency range: f2 = ',int2str(f2)],range);
% Meshsettings
meshsettings = invoke(mws,'MeshSettings');
invoke(meshsettings, 'SetMeshType','Hex');
invoke(meshsettings, 'Set','Version','1%');
% Mesh
mesh = invoke(mws,'Mesh');
invoke(mesh, 'MeshType','PBA');

fprintf("Frequency Range: "+num2str(f1)+" - "+num2str(f2)+" THz\n");

end

```

ground\_gold.m: función para definir el material oro.

```

function ground_gold(mws)

% Define material: Gold

material = invoke(mws,'material');
invoke(material,'Reset');
invoke(material,'Name','Gold');
invoke(material,'FrqType','static');
invoke(material,'Type','Normal');
invoke(material,'SetMaterialUnit','Hz','mm');
invoke(material,'Epsilon','1');
invoke(material,'Mu','1.0');
invoke(material,'Kappa','4.561e+007');
invoke(material,'TanD','0.0');
invoke(material,'TanDFreq','0.0');
invoke(material,'TanDGiven','False');
invoke(material,'TanDModel','ConstTanD');
invoke(material,'KappaM','0');
invoke(material,'TanDM','0.0');
invoke(material,'TanDMFreq','0.0');
invoke(material,'TanDMGiven','False');
invoke(material,'TanDModel','ConstTanD');
invoke(material,'DispModelEps','None');
invoke(material,'DispModelMu','None');
invoke(material,'DispersiveFittingSchemeEps','General 1st');
invoke(material,'DispersiveFittingSchemeMu','General 1st');
invoke(material,'UseGeneralDispersionEps','False');
invoke(material,'UseGeneralDispersionMu','False');
invoke(material,'FrqType','all');
invoke(material,'Type','Lossy metal');
invoke(material,'MaterialUnit','Frequency','GHz');
invoke(material,'MaterialUnit','Geometry','mm');
invoke(material,'MaterialUnit','Time','s');
invoke(material,'MaterialUnit','Temperature','Kelvin');
invoke(material,'Mu','1.0');
invoke(material,'Sigma','4.561e+007');
invoke(material,'Rho','19320.0');
invoke(material,'ThermalType','Normal');
invoke(material,'ThermalConductivity','314.0');
invoke(material,'SpecificHeat','130','J/K/kg');
invoke(material,'MetabolicRate','0');
invoke(material,'BloodFlow','0');
invoke(material,'VoxelConvection','0');
invoke(material,'MechanicsType','Isotropic');
invoke(material,'YoungsModulus','78');
invoke(material,'PoissonsRatio','0.42');
invoke(material,'ThermalExpansionRate','14');
invoke(material,'ReferenceCoordSystem','Global');
invoke(material,'CoordSystemType','Cartesian');
invoke(material,'NLAnisotropy','False');

```

```

invoke(material, 'NLStackingFactor', '1');
invoke(material, 'NLADirectionX', '1');
invoke(material, 'NLADirectionY', '0');
invoke(material, 'NLADirectionZ', '0');
invoke(material, 'Colour', '1', '1', '0');
invoke(material, 'Wireframe', 'False');
invoke(material, 'Reflection', 'False');
invoke(material, 'Allowoutline', 'True');
invoke(material, 'Transparentoutline', 'False');
invoke(material, 'Transparency', '0');
invoke(material, 'Create');
release(material);
end

```

`mat_cooperpurelossy.m`: función para definir el material cobre.

```

function mat_cooperpurelossy(mws)

% Define material: Copper (pure)

material = invoke(mws, 'material');
invoke(material, 'Reset');
invoke(material, 'Name', 'Copper (pure)');
invoke(material, 'FrqType', 'all');
invoke(material, 'Type', 'Lossy metal');
invoke(material, 'MaterialUnit', 'Frequency', 'GHz');
invoke(material, 'MaterialUnit', 'Geometry', 'mm');
invoke(material, 'MaterialUnit', 'Time', 's');
invoke(material, 'MaterialUnit', 'Temperature', 'Kelvin');
invoke(material, 'Mue', '1.0');
invoke(material, 'Sigma', '5.96e+007');
invoke(material, 'Rho', '8930.0');
invoke(material, 'ThermalType', 'Normal');
invoke(material, 'ThermalConductivity', '401.0');
invoke(material, 'HeatCapacity', '0.39');
invoke(material, 'MetabolicRate', '0');
invoke(material, 'BloodFlow', '0');
invoke(material, 'VoxelConvection', '0');
invoke(material, 'MechanicsType', 'Isotropic');
invoke(material, 'YoungsModulus', '120');
invoke(material, 'PoissonsRatio', '0.33');
invoke(material, 'ThermalExpansionRate', '17');
invoke(material, 'ReferenceCoordSystem', 'Global');
invoke(material, 'CoordSystemType', 'Cartesian');
invoke(material, 'NLAnisotropy', 'False');
invoke(material, 'NLStackingFactor', '1');
invoke(material, 'NLADirectionX', '1');
invoke(material, 'NLADirectionY', '0');
invoke(material, 'NLADirectionZ', '0');
invoke(material, 'ParticleProperty', 'SecondaryEmission');
invoke(material, 'SeModel', 'Furman');
invoke(material, 'SeVaughan', '175', '2.25', '0', '1.0', '7.5');
invoke(material, 'SePlot1D', 'True', '0', '100', '0', '1000');
invoke(material, 'SeMaxSecondaries', '10');
invoke(material, 'SeTsParam_T1', '0.66');
invoke(material, 'SeTsParam_T2', '0.80');
invoke(material, 'SeTsParam_T3', '0.70');
invoke(material, 'SeTsParam_T4', '1.00');
invoke(material, 'SeTsParam_SEY', '2.25');
invoke(material, 'SeTsParam_Energy', '175');
invoke(material, 'SeTsParam_S', '1.33');
invoke(material, 'SeTsParam_PN', '0', '1.6');
invoke(material, 'SeTsParam_PN', '1', '2.0');
invoke(material, 'SeTsParam_PN', '2', '1.8');
invoke(material, 'SeTsParam_PN', '3', '4.7');
invoke(material, 'SeTsParam_PN', '4', '1.8');
invoke(material, 'SeTsParam_PN', '5', '2.4');
invoke(material, 'SeTsParam_PN', '6', '1.8');
invoke(material, 'SeTsParam_PN', '7', '1.8');
invoke(material, 'SeTsParam_PN', '8', '2.3');
invoke(material, 'SeTsParam_PN', '9', '1.8');
invoke(material, 'SeTsParam_EpsN', '0', '3.90');
invoke(material, 'SeTsParam_EpsN', '1', '6.20');

```

```

invoke(material, 'SeTsParam_EpsN', '2', '13.00');
invoke(material, 'SeTsParam_EpsN', '3', '8.80');
invoke(material, 'SeTsParam_EpsN', '4', '6.25');
invoke(material, 'SeTsParam_EpsN', '5', '2.25');
invoke(material, 'SeTsParam_EpsN', '6', '9.20');
invoke(material, 'SeTsParam_EpsN', '7', '5.30');
invoke(material, 'SeTsParam_EpsN', '8', '17.80');
invoke(material, 'SeTsParam_EpsN', '9', '10.00');
invoke(material, 'SeRdParam_R', '1.0');
invoke(material, 'SeRdParam_R1', '0.26');
invoke(material, 'SeRdParam_R2', '2.0');
invoke(material, 'SeRdParam_Q', '0.4');
invoke(material, 'SeRdParam_P1Inf', '0.01');
invoke(material, 'SeRdParam_Energy', '40.0');
invoke(material, 'SeBsParam_Sigma', '1.9');
invoke(material, 'SeBsParam_E1', '0.26');
invoke(material, 'SeBsParam_E2', '2.0');
invoke(material, 'SeBsParam_P1Hat', '0.02');
invoke(material, 'SeBsParam_P1Inf', '0.01');
invoke(material, 'SeBsParam_Energy', '0.0');
invoke(material, 'SeBsParam_P', '0.9');
invoke(material, 'SeBsParam_W', '100.0');
invoke(material, 'FrqType', 'static');
invoke(material, 'Type', 'Normal');
invoke(material, 'SetMaterialUnit', 'Hz', 'mm');
invoke(material, 'Epsilon', '1');
invoke(material, 'Mue', '1.0');
invoke(material, 'Kappa', '5.96e+007');
invoke(material, 'TanD', '0');
invoke(material, 'TanDFreq', '0.0');
invoke(material, 'TanDGiven', 'False');
invoke(material, 'TanDModel', 'ConstTanD');
invoke(material, 'KappaM', '0.0');
invoke(material, 'TanDM', '0.0');
invoke(material, 'TanDMFreq', '0.0');
invoke(material, 'TanDMGiven', 'False');
invoke(material, 'TanDMModel', 'ConstTanD');
invoke(material, 'DispModelEps', 'None');
invoke(material, 'DispModelMue', 'None');
invoke(material, 'DispersiveFittingSchemeEps', 'Nth Order');
invoke(material, 'DispersiveFittingSchemeMue', 'Nth Order');
invoke(material, 'UseGeneralDispersionEps', 'False');
invoke(material, 'UseGeneralDispersionMue', 'False');
invoke(material, 'Colour', '1', '1', '0');
invoke(material, 'Wireframe', 'False');
invoke(material, 'Reflection', 'False');
invoke(material, 'Allowoutline', 'True');
invoke(material, 'Transparentoutline', 'False');
invoke(material, 'Transparency', '0');
invoke(material, 'Create');
release(material);

```

end

`mat_grapehe_eps.m`: función para definir el material grafeno  $\mu_c=0.1$  eV

`mat_grapehe2_eps.m`: función para definir el material grafeno  $\mu_c=0.2$  eV

`mat_grapehe4_eps.m`: función para definir el material grafeno  $\mu_c=0.43$  eV

Se incluye sólo una función a modo de ejemplo:

```

function mat_graphene2_eps(mws)

% Define material: Graphene2_Eps

material = invoke(mws, 'material');
invoke(material, 'Reset');
invoke(material, 'Name', 'Graphene2_Eps');
invoke(material, 'Rho', '0.0');
invoke(material, 'ThermalType', 'Normal');
invoke(material, 'ThermalConductivity', '0');
invoke(material, 'SpecificHeat', '0', 'J/K/kg');
invoke(material, 'DynamicViscosity', '0');

```

```

invoke(material, 'Emissivity', "0");
invoke(material, 'MetabolicRate', "0.0");
invoke(material, 'VoxelConvection', "0.0");
invoke(material, 'BloodFlow', "0");
invoke(material, 'MechanicsType', "Unused");
invoke(material, 'FrqType', "all");
invoke(material, 'Type', "Normal");
invoke(material, 'MaterialUnit', "Frequency", "THz");
invoke(material, 'MaterialUnit', "Geometry", "mm");
invoke(material, 'MaterialUnit', "Time", "s");
invoke(material, 'Epsilon', "1");
invoke(material, 'Mu', "1");
invoke(material, 'Sigma', "0");
invoke(material, 'TanD', "0.0");
invoke(material, 'TanDFreq', "0.0");
invoke(material, 'TanDGiven', "False");
invoke(material, 'TanDModel', "ConstSigma");
invoke(material, 'SetConstTanDStrategyEps', "AutomaticOrder");
invoke(material, 'ConstTanDModelOrderEps', "3");
invoke(material, 'DjordjevicSarkarUpperFreqEps', "0");
invoke(material, 'SetElParametricConductivity', "False");
invoke(material, 'ReferenceCoordSystem', "Global");
invoke(material, 'CoordSystemType', "Cartesian");
invoke(material, 'SigmaM', "0");
invoke(material, 'TanDM', "0.0");
invoke(material, 'TanDMFreq', "0.0");
invoke(material, 'TanDMGiven', "False");
invoke(material, 'TanDModel', "ConstSigma");
invoke(material, 'SetConstTanDStrategyMu', "AutomaticOrder");
invoke(material, 'ConstTanDModelOrderMu', "3");
invoke(material, 'DjordjevicSarkarUpperFreqMu', "0");
invoke(material, 'SetMagParametricConductivity', "False");
invoke(material, 'DispModelEps', "None");
invoke(material, 'DispModelMu', "None");
invoke(material, 'DispersiveFittingSchemeEps', "Nth Order");
invoke(material, 'MaximalOrderNthModelFitEps', "10");
invoke(material, 'ErrorLimitNthModelFitEps', "0.001");
invoke(material, 'DispersiveFittingSchemeMu', "Nth Order");
invoke(material, 'MaximalOrderNthModelFitMu', "10");
invoke(material, 'ErrorLimitNthModelFitMu', "0.1");
invoke(material, 'DispersiveFittingFormatEps', "Real_Imag");
invoke(material, 'AddDispersionFittingValueEps', "0.1", "-60475.3238682608", "486221.764314867", "1.0");
invoke(material, 'AddDispersionFittingValueEps', "0.2", "-57825.8428757853", "232355.964446948", "1.0");
invoke(material, 'AddDispersionFittingValueEps', "0.3", "-53830.7466458803", "144251.848648992", "1.0");
invoke(material, 'AddDispersionFittingValueEps', "0.4", "-49064.4481570552", "98689.0944635927", "1.0");
invoke(material, 'AddDispersionFittingValueEps', "0.5", "-44038.7098205628", "70948.5128078921", "1.0");
invoke(material, 'AddDispersionFittingValueEps', "0.6", "-39129.582625602", "52614.647277344", "1.0");
invoke(material, 'AddDispersionFittingValueEps', "0.7", "-34566.0401660329", "39914.9515868025", "1.0");
invoke(material, 'AddDispersionFittingValueEps', "0.8", "-30457.8732324289", "30845.0994178177", "1.0");
invoke(material, 'AddDispersionFittingValueEps', "0.9", "-26834.2685852642", "24220.5367510296", "1.0");
invoke(material, 'AddDispersionFittingValueEps', "1", "-23677.2977360297", "19293.1565064866", "1.0");
invoke(material, 'AddDispersionFittingValueEps', "1.1", "-20945.6122071901", "15570.0139377376", "1.0");
invoke(material, 'AddDispersionFittingValueEps', "1.2", "-18589.1757055959", "12716.757280743", "1.0");
invoke(material, 'AddDispersionFittingValueEps', "1.3", "-16557.5046811435", "10501.6630752001", "1.0");
invoke(material, 'AddDispersionFittingValueEps', "1.4", "-14803.7640208932", "8761.30442299733", "1.0");
invoke(material, 'AddDispersionFittingValueEps', "1.5", "-13286.4364815126", "7378.6697805012", "1.0");
invoke(material, 'AddDispersionFittingValueEps', "1.6", "-11969.6763195574", "6268.81827560385", "1.0");
invoke(material, 'AddDispersionFittingValueEps', "1.7", "-10823.0110043002", "5369.33053416748", "1.0");
invoke(material, 'AddDispersionFittingValueEps', "1.8", "-9827.7678861477", "4633.77351369963", "1.0");
invoke(material, 'AddDispersionFittingValueEps', "1.9", "-8941.40575871746", "4027.22962041835", "1.0");
invoke(material, 'AddDispersionFittingValueEps', "2", "-8166.90753196697", "3523.16261509093", "1.0");
invoke(material, 'AddDispersionFittingValueEps', "2.1", "-7482.18117587023", "3101.20503662901", "1.0");
invoke(material, 'AddDispersionFittingValueEps', "2.2", "-6874.57896594014", "2745.57705046177", "1.0");
invoke(material, 'AddDispersionFittingValueEps', "2.3", "-6333.4769263557", "2443.94435834998", "1.0");
invoke(material, 'AddDispersionFittingValueEps', "2.4", "-5849.92671828778", "2186.58478704066", "1.0");
invoke(material, 'AddDispersionFittingValueEps', "2.5", "-5416.36799517421", "1965.774175628", "1.0");
invoke(material, 'AddDispersionFittingValueEps', "2.6", "-5026.39180599628", "1775.32962940715", "1.0");
invoke(material, 'AddDispersionFittingValueEps', "2.7", "-4674.54639306074", "1610.2667859481", "1.0");
invoke(material, 'AddDispersionFittingValueEps', "2.8", "-4356.17785420901", "1466.54044538429", "1.0");
invoke(material, 'AddDispersionFittingValueEps', "2.9", "-4067.29929737388", "1340.84669438525", "1.0");
invoke(material, 'AddDispersionFittingValueEps', "3", "-3786.89017920842", "1230.47077511401", "1.0");
invoke(material, 'AddDispersionFittingValueEps', "3.1", "-3564.77288606981", "1133.16925969", "1.0");
invoke(material, 'AddDispersionFittingValueEps', "3.2", "-3345.60886936655", "1047.07815106541", "1.0");
invoke(material, 'AddDispersionFittingValueEps', "3.3", "-3144.76846189407", "970.640723397633", "1.0");
invoke(material, 'AddDispersionFittingValueEps', "3.4", "-2963.97127895706", "902.550498287988", "1.0");
invoke(material, 'AddDispersionFittingValueEps', "3.5", "-2790.55624589235", "841.705905911957", "1.0");
invoke(material, 'AddDispersionFittingValueEps', "3.6", "-2635.73070979147", "787.174025667382", "1.0");
invoke(material, 'AddDispersionFittingValueEps', "3.7", "-2489.36648593692", "738.161425901924", "1.0");
invoke(material, 'AddDispersionFittingValueEps', "3.8", "-2355.48097635735", "693.990587424235", "1.0");
invoke(material, 'AddDispersionFittingValueEps', "3.9", "-2231.33485267991", "654.080744085357", "1.0");
invoke(material, 'AddDispersionFittingValueEps', "4", "-2116.02678321935", "617.93223669682", "1.0");
invoke(material, 'AddDispersionFittingValueEps', "4.1", "-2008.75577220172", "585.113675951967", "1.0");
invoke(material, 'AddDispersionFittingValueEps', "4.2", "-1889.98377389595", "555.251363238643", "1.0");
invoke(material, 'AddDispersionFittingValueEps', "4.3", "-1815.54698444758", "528.020533841765", "1.0");
invoke(material, 'AddDispersionFittingValueEps', "4.4", "-1728.4016886956", "503.13807942913", "1.0");
invoke(material, 'AddDispersionFittingValueEps', "4.5", "-1646.86078892071", "480.35647549172", "1.0");
invoke(material, 'AddDispersionFittingValueEps', "4.6", "-1573.12406660447", "459.458695064006", "1.0");
invoke(material, 'AddDispersionFittingValueEps', "4.7", "-1498.79842327479", "440.253932915444", "1.0");
invoke(material, 'AddDispersionFittingValueEps', "4.8", "-1431.48890684272", "422.573998408798", "1.0");
invoke(material, 'AddDispersionFittingValueEps', "4.9", "-1368.19783325389", "406.2706214304", "1.0");
invoke(material, 'AddDispersionFittingValueEps', "5", "-1308.61894251828", "391.211062926983", "1.0");
invoke(material, 'AddDispersionFittingValueEps', "5.1", "-1252.47428164426", "377.279498760126", "1.0");
invoke(material, 'AddDispersionFittingValueEps', "5.2", "-1199.5111512963", "364.371539249182", "1.0");
invoke(material, 'AddDispersionFittingValueEps', "5.3", "-1149.49942642607", "352.394407973769", "1.0");
invoke(material, 'AddDispersionFittingValueEps', "5.4", "-1102.22919991015", "341.265192284971", "1.0");
invoke(material, 'AddDispersionFittingValueEps', "5.5", "-1057.50870583641", "330.909645307098", "1.0");
invoke(material, 'AddDispersionFittingValueEps', "5.6", "-1015.16248546105", "321.261150853712", "1.0");
invoke(material, 'AddDispersionFittingValueEps', "5.7", "-975.029764222137", "312.259826829572", "1.0");
invoke(material, 'AddDispersionFittingValueEps', "5.8", "-936.31003732944", "303.851746680789", "1.0");

```



invoke(material, 'AddDispersionFittingValueEps', "5.9", "-900.826668363321", "295.988261742701", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "6", "-864.77154642376", "288.625410051484", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "6.1", "-833.856082187656", "281.723399437527", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "6.2", "-802.387901569855", "275.246154591812", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "6.3", "-773.232576016164", "269.160919351713", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "6.4", "-744.897925698862", "263.437906780889", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "6.5", "-718.200970385801", "258.049990668318", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "6.6", "-692.031896334295", "252.972433048413", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "6.7", "-668.114139231754", "248.182643075679", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "6.8", "-645.363840010666", "243.65996326598", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "6.9", "-622.415081052787", "239.385479666023", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "7", "-599.3283032308", "235.341852985626", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "7.1", "-580.622349905696", "231.513168129301", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "7.2", "-560.700499153061", "227.884799906624", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "7.3", "-542.318148792177", "224.443292994079", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "7.4", "-525.086465682684", "221.176254472374", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "7.5", "-507.138546016934", "218.072257478982", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "7.6", "-490.600479042648", "215.120754701326", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "7.7", "-474.765394486368", "212.312000596085", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "7.8", "-459.652716511208", "209.636981358391", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "7.9", "-444.919624774311", "207.087351784304", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "8", "-430.521902974888", "204.655378273757", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "8.1", "-417.355652399996", "202.333887311221", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "8.2", "-404.656461323093", "200.116218839795", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "8.3", "-391.856693458739", "197.996184012732", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "8.4", "-379.699328173406", "195.968026866097", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "8.5", "-368.230824419639", "194.026389580402", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "8.6", "-356.817420688973", "192.166280468774", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "8.7", "-346.307654442354", "190.38304588529", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "8.8", "-336.047694271031", "188.672343250315", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "8.9", "-325.935504115336", "187.030117460673", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "9", "-316.671379811696", "185.452578947782", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "9.1", "-306.979004691563", "183.93618368695", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "9.2", "-297.988560516004", "182.477614906327", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "9.3", "-289.317047909957", "181.073766334842", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "9.4", "-281.348456556942", "179.721726845125", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "9.5", "-272.841029263951", "178.418766362208", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "9.6", "-265.055828337358", "177.162322921972", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "9.7", "-257.453337817349", "175.94999077495", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "9.8", "-250.15750369415", "174.779509441565", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "9.9", "-243.066053912913", "173.648753634099", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "10", "-236.148876599036", "172.555723969023", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "10.1", "-229.59982278853", "171.498538400689", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "10.2", "-223.292224085645", "170.47542431401", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "10.3", "-216.982877533803", "169.484711219697", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "10.4", "-210.951190311109", "168.524824000921", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "10.5", "-205.150189239338", "167.59427666504", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "10.6", "-199.296595017805", "166.69166558335", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "10.7", "-194.042725810227", "165.815669005526", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "10.8", "-188.776206385772", "164.965032339331", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "10.9", "-183.60860389043", "164.138573288446", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "11", "-178.735215603475", "163.335172695148", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "11.1", "-173.793520799607", "162.553771536266", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "11.2", "-168.992703911006", "161.793367223258", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "11.3", "-164.55825541653", "161.05301061759", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "11.4", "-160.229490080967", "160.331800543863", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "11.5", "-155.86022863191", "159.62885366137", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "11.6", "-151.641787142414", "158.94345564535", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "11.7", "-147.662115401872", "158.274743823367", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "11.8", "-143.781624056322", "157.620221344896", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "11.9", "-139.92970160827", "156.984596395534", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "12", "-136.306890652479", "156.36181788268", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "12.1", "-132.631579924001", "155.75304298757", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "12.2", "-129.198192873037", "155.15769626109", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "12.3", "-125.738879706651", "154.575206949752", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "12.4", "-122.359299942505", "154.005037847791", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "12.5", "-119.225026649263", "153.446677684936", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "12.6", "-116.166131957827", "152.899639703547", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "12.7", "-113.065528507838", "152.363460324074", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "12.8", "-110.062738728717", "151.837697892727", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "12.9", "-107.237783729011", "151.321931505691", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "13", "-104.629025223632", "150.815759904676", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "13.1", "-101.720910229242", "150.318800438954", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "13.2", "-99.1580429214668", "149.830688089432", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "13.3", "-96.4955919417985", "149.351074559597", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "13.4", "-93.8687659754105", "148.87962736653", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "13.5", "-91.5439410598725", "148.416029117399", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "13.6", "-89.1985375917617", "147.959976653165", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "13.7", "-86.8493741714227", "147.51118037142", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "13.8", "-84.4013277787211", "147.069363536532", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "13.9", "-82.3965007110314", "146.634261637443", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "14", "-80.1909091692452", "146.205621781681", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "14.1", "-78.1710223516671", "145.783202123295", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "14.2", "-76.264108506959", "145.366771322592", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "14.3", "-74.1596421297548", "144.956108035705", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "14.4", "-72.184365604935", "144.551000432148", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "14.5", "-70.3499822443817", "144.151245738634", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "14.6", "-68.5924080989201", "143.756649807579", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "14.7", "-66.7305095991397", "143.367026708771", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "14.8", "-64.9288185023354", "142.982198342835", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "14.9", "-63.2904682659539", "142.60199407519", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "15", "-61.5248194887125", "142.226250389266", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "15.1", "-60.0198181466448", "141.854810557875", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "15.2", "-58.4626169939307", "141.487524331646", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "15.3", "-56.9091791919746", "141.124247643555", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "15.4", "-55.4675108915388", "140.764842328608", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "15.5", "-53.9497806112404", "140.409175857804", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "15.6", "-52.4791397757486", "140.057121085586", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "15.7", "-51.133414569625", "139.70855600998", "1.0");

invoke(material, 'AddDispersionFittingValueEps', '15.8', "-49.8424071443339", "139.363363544749", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '15.9', "-48.4523939266741", "139.02143130285", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '16', "-47.1987960028873", "138.682651390602", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '16.1', "-45.8995136185923", "138.34692021194", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '16.2', "-44.7796977599817", "138.01413828232", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '16.3', "-43.4680153303555", "137.684210051108", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '16.4', "-42.3115476159636", "137.35704373384", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '16.5', "-41.1515551418318", "137.03255115079", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '16.6', "-39.926774415299", "136.710647574501", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '16.7', "-38.9441738657004", "136.391251584033", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '16.8', "-37.8753226240312", "136.074284926148", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '16.9', "-36.8402698247322", "135.759672382984", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '17', "-35.8279513146431", "135.44734164589", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '17.1', "-34.8345738422168", "135.137223195088", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '17.2', "-33.8163582054429", "134.829250184871", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '17.3', "-32.9221262426549", "134.523358334046", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '17.4', "-31.9818131733037", "134.219485821355", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '17.5', "-31.0982556804649", "133.917573185625", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '17.6', "-30.204515167536", "133.6175632304", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '17.7', "-29.3585596327937", "133.319400932843", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '17.8', "-28.411301546595", "133.023033356675", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '17.9', "-27.6988864088713", "132.728409568976", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '18', "-26.9021247318953", "132.435480560629", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '18.1', "-26.1153185428478", "132.144199176254", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '18.2', "-25.3504829750191", "131.854520011442", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '18.3', "-24.6041574500547", "131.566399403142", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '18.4', "-23.8363893172434", "131.279795303036", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '18.5', "-23.1619092381821", "130.994667243774", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '18.6', "-22.4477256033363", "130.710976271921", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '18.7', "-21.7852715752386", "130.42868489484", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '18.8', "-21.1223734091131", "130.147756997913", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '18.9', "-20.4711215254112", "129.868157844446", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '19', "-19.7833587403932", "129.589853970684", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '19.1', "-19.2165042722641", "129.312813163316", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '19.2', "-18.6109997211566", "129.037004406867", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '19.3', "-18.0186226561687", "128.762397838392", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '19.4', "-17.4571766720614", "128.488964704024", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '19.5', "-16.8748274595424", "128.216677317286", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '19.6', "-16.327104277888", "127.9455090191", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '19.7', "-15.7826083795161", "127.675434139397", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '19.8', "-15.2782020428538", "127.406427960275", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '19.9', "-14.7395856330573", "127.138466680628", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '20', "-14.2385424326465", "126.871527382178", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '20.1', "-13.7435021444732", "126.605587996849", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '20.2', "-13.2606589275108", "126.340627275436", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '20.3', "-12.7922162696349", "126.076624757489", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '20.4', "-12.2983602001502", "125.813560742383", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '20.5', "-11.8836950152254", "125.551416261508", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '20.6', "-11.4583616361088", "125.290173051538", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '20.7', "-11.016007714947", "125.029813528722", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '20.8', "-10.6002855958139", "124.770320764174", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '20.9', "-10.1873201278548", "124.511678460096", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '21', "-9.82609157372247", "124.253870926914", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '21.1', "-9.3958892696499", "123.996883061268", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '21.2', "-9.00900629191961", "123.740700324853", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '21.3', "-8.64005654893068", "123.485308724029", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '21.4', "-8.27108361363389", "123.230694790222", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '21.5', "-7.91824637804653", "122.976845561031", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '21.6', "-7.57254305715638", "122.723748562055", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '21.7', "-7.22895824001609", "122.471391789381", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '21.8', "-6.89586234675929", "122.219763692726", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '21.9', "-6.57076418295372", "121.96885315919", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '22', "-6.2730029805769", "121.718649497611", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '22.1', "-5.94230452489834", "121.469142423487", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '22.2', "-5.6732510847502", "121.22032004442", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '22.3', "-5.34228414879076", "120.97217884623", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '22.4', "-5.05094669130914", "120.724703679227", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '22.5', "-4.76946902738322", "120.477887745427", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '22.6', "-4.51614850656804", "120.23172258589", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '22.7', "-4.2226829619329", "119.986200068646", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '22.8', "-3.95162420644365", "119.741312377027", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '22.9', "-3.70080451979259", "119.497051998414", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '23', "-3.3875895510857", "119.253411713383", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '23.1', "-3.2027641571747", "119.010384585232", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '23.2', "-2.98142499389164", "118.767963949878", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '23.3', "-2.72754151441128", "118.52614340611", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '23.4', "-2.53123152326867", "118.284916806182", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '23.5', "-2.274162872006", "118.044278246727", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '23.6', "-2.05039867506372", "117.80422060002", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '23.7', "-1.84169875666485", "117.564742805424", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '23.8', "-1.74182488670607", "117.325835261406", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '23.9', "-1.42926168728842", "117.087494417469", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '24', "-1.22050166162416", "116.849715466632", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '24.1', "-1.03600405167589", "116.612493798062", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '24.2', "-0.76714732174842", "116.375824989972", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '24.3', "-0.661116105347927", "116.139704802764", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '24.4', "-0.476221943061222", "115.904129172413", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '24.5', "-0.303824084556203", "115.669094204063", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '24.6', "-0.248377347825877", "115.434596165853", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '24.7', "3.66115739072251E-02", "115.200631482946", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '24.8', "0.133709322914407", "114.967196731763", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '24.9', "0.360897912932956", "114.734288634409", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '25', "0.603254631128068", "114.501904053294", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '25.1', "0.669711041704015", "114.270039985922", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '25.2', "0.71796570613352", "114.038693559874", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '25.3', "0.963697634591593", "113.807862027942", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '25.4', "1.10256077626137", "113.577542763437", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '25.5', "1.24347634782179", "113.347733255653", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '25.6', "1.45168951776395", "113.118431105476", "1.0");

invoke(material, 'AddDispersionFittingValueEps', "25.7", "1.50963915901762", "112.889634021149", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "25.8", "2.12857283580594", "112.661339814174", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "25.9", "1.76275263436106", "112.43354639535", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "26", "1.79198978491312", "112.206251770944", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "26.1", "2.00335912779788", "111.979454038991", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "26.2", "2.11913540203004", "111.753151385717", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "26.3", "2.23197791640745", "111.527342082078", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "26.4", "2.34194815122273", "111.302024480417", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "26.5", "2.44910627577398", "111.077197011233", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "26.6", "2.55351117893775", "110.852858180055", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "26.7", "2.65522049894106", "110.62900656442", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "26.8", "2.89561847675472", "110.405640810952", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "26.9", "2.85077686229813", "110.182759632542", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "27", "3.2308590951298", "109.960361805612", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "27.1", "3.0362125415149", "109.738446167483", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "27.2", "3.04342003000104", "109.517011613817", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "27.3", "3.21194648043955", "109.296057096155", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "27.4", "3.2719016239427", "109.075581619531", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "27.5", "3.37838021569135", "108.855584240166", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "27.6", "3.49298205568473", "108.636064063244", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "27.7", "3.53589865949045", "108.41702040751", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "27.8", "3.72944787069378", "108.1984519694", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "27.9", "3.6848708431156", "107.980358488616", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "28", "3.7488524266608", "107.76273907859", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "28.1", "3.82655062321779", "107.5455930584", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "28.2", "3.83032829497262", "107.328919784196", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "28.3", "3.95857735308255", "107.11271864744", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "28.4", "4.04253865163243", "106.896989073213", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "28.5", "4.08397652080031", "106.681730518576", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "28.6", "4.12576724071702", "106.466942470983", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "28.7", "4.20216035618211", "106.25264446752", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "28.8", "4.25098691856386", "106.038775989588", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "28.9", "4.31342840813939", "105.825396669152", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "29", "4.38734768073491", "105.612486079685", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "29.1", "4.41806809414896", "105.40004383867", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "29.2", "4.449880444954", "105.188069585553", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "29.3", "4.51635522331968", "104.976562980495", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "29.4", "4.5525369910504", "104.765523703172", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "29.5", "4.6085544948117", "104.554951451622", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "29.6", "4.66843822646106", "104.344845941122", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "29.7", "4.69491997063938", "104.13520690311", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "29.8", "4.69557830013065", "103.926034084147", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "29.9", "4.77569553105484", "103.717327244907", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "30", "4.82061738696651", "103.509086159215", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "30.1", "4.85111530212704", "103.301310613103", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "30.2", "4.86849730122249", "103.094000403915", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "30.3", "4.92140406819736", "102.887155339436", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "30.4", "4.95238067158408", "102.680775237051", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "30.5", "4.98677766332433", "102.474859922942", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "30.6", "5.01545027584425", "102.269409231302", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "30.7", "5.04744334500932", "102.064423003592", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "30.8", "5.07176626332392", "101.859901087815", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "30.9", "5.1036001508103", "101.655843337816", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "31", "5.13243955149142", "101.452249612619", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "31.1", "5.15543923871345", "101.249119775776", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "31.2", "5.18211474743888", "101.046453694749", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "31.3", "5.20314421209333", "100.844251240309", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "31.4", "5.20967297891904", "100.642512285966", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "31.5", "5.24689143003805", "100.441236707413", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "31.6", "5.29361555018767", "100.240424381995", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "31.7", "5.28685030377413", "100.040075188199", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "31.8", "5.31655270289991", "99.840189005164", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "31.9", "5.32318357988623", "99.6407657122088", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "32", "5.34995097847646", "99.441805188313", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "32.1", "5.35604761098531", "99.2433073120238", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "32.2", "5.35308138181868", "99.0452719603575", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "32.3", "5.3859261444486", "98.8476990090829", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "32.4", "5.41107922698582", "98.650588331998", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "32.5", "5.41196291978904", "98.4539398006325", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "32.6", "5.4431039735291", "98.2577532838965", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "32.7", "5.43529720528519", "98.0620286474455", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "32.8", "5.43549791120579", "97.8667657548603", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "32.9", "5.45572872426625", "97.6719644643393", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "33", "5.44999468865948", "97.4776246314068", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "33.1", "5.47338552167017", "97.2837461071334", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "33.2", "5.4909825060614", "97.0903287381689", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "33.3", "5.48839064127432", "96.8973723664895", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "33.4", "5.47609250562279", "96.7048768291547", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "33.5", "5.50086232405855", "96.5128419580783", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "33.6", "5.50411198353424", "96.3212675798087", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "33.7", "5.51091419812067", "96.1301535153221", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "33.8", "5.50252185913646", "95.9394995798249", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "33.9", "5.51865546054223", "95.7493055825668", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "34", "5.51947886815928", "95.5595713266646", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "34.1", "5.52419105157729", "95.3702966089348", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "34.2", "5.51694395269206", "95.1814812197356", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "34.3", "5.52762182152685", "94.993124942819", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "34.4", "5.54109299309674", "94.8052275551902", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "34.5", "5.52904469062909", "94.617788269766", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "34.6", "5.53063065015322", "94.4308085213044", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "34.7", "5.52855280229586", "94.2442863941835", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "34.8", "5.53572638315523", "94.0582221944002", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "34.9", "5.52623566999185", "93.8726156634161", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "35", "5.50502508317942", "93.6874665352766", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "35.1", "5.52217931804998", "93.5027745365237", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "35.2", "5.51372627165224", "93.3185393861174", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "35.3", "5.51646641669691", "93.1347607953627", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "35.4", "5.50287084891312", "92.9514384678433", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "35.5", "5.50917641154758", "92.7685720993609", "1.0");

invoke(material, 'AddDispersionFittingValueEps', "35.6", "5.50667480103524", "92.5861613778809", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "35.7", "5.50038564781824", "92.4042059834835", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "35.8", "5.50674245781649", "92.22270558832", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "35.9", "5.49016748949184", "92.0416598565753", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "36", "5.47387923253402", "91.8610684444341", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "36.1", "5.47859243366186", "91.6809310000534", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "36.2", "5.46365516841451", "91.5012471635388", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "36.3", "5.46572822026294", "91.3220165669259", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "36.4", "5.45869889682208", "91.143236834166", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "36.5", "5.4516399373967", "90.9649135811159", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "36.6", "5.45142185961017", "90.787040415532", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "36.7", "5.43639012243898", "90.6096189370684", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "36.8", "5.42948669534203", "90.4326487372784", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "36.9", "5.4200388591162", "90.2561293996206", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "37", "5.42122679521289", "90.0800604994673", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "37.1", "5.40264387072398", "89.9044416041174", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "37.2", "5.39621882472609", "89.7292722728119", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "37.3", "5.38426060965271", "89.5545520567531", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "37.4", "5.37353628322692", "89.3802804991263", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "37.5", "5.36494234338025", "89.2064571351247", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "37.6", "5.35876706666163", "89.0330814919771", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "37.7", "5.34474023708085", "88.8601530889786", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "37.8", "5.31553582512223", "88.687614375227", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "37.9", "5.32370343299637", "88.5156360411379", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "38", "5.31618114564532", "88.3440463955246", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "38.1", "5.30187912670265", "88.1720919885957", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "38.2", "5.3078086830906", "88.0022023005189", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "38.3", "5.27931264040618", "87.8319468037607", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "38.4", "5.26650526875694", "87.6621349631334", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "38.5", "5.25604749339194", "87.4927662358432", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "38.6", "5.25927498971984", "87.32384007154", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "38.7", "5.2321254697438", "87.15535591237", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "38.8", "5.22316178509603", "86.9873131930289", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "38.9", "5.20758668344849", "86.8197113408169", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "39", "5.20124739251456", "86.6525497756962", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "39.1", "5.18246964099368", "86.4858279103483", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "39.2", "5.1726438624143", "86.3195451502339", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "39.3", "5.15681130156178", "86.153700893654", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "39.4", "5.14827935964624", "85.9882945318117", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "39.5", "5.13064713491822", "85.8233254488756", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "39.6", "5.12067768661557", "85.6587930220439", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "39.7", "5.10401117708999", "85.4946966216106", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "39.8", "5.09578532455441", "85.3310356110309", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "39.9", "5.07693608392138", "85.1678093468988", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "40", "5.06529331145328", "85.005017194692", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "40.1", "5.04945318259744", "84.842658451818", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "40.2", "5.02815778407106", "84.6807325008211", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "40.3", "5.02159252121159", "84.5192386567706", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "40.4", "5.01283890735543", "84.3581762433571", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "40.5", "4.99338291646157", "84.1975445786415", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "40.6", "4.98306512887892", "84.0373429733272", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "40.7", "4.96485199954555", "83.8775707326338", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "40.8", "4.95284667897714", "83.7182271554699", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "40.9", "4.9360262603298", "83.559311534688", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "41", "4.92707392221787", "83.4008231571581", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "41.1", "4.90693108986046", "83.2427613038429", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "41.2", "4.89462526231731", "83.0851252498732", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "41.3", "4.87759082128036", "82.9279142646229", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "41.4", "4.8570268254428", "82.7711276117847", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "41.5", "4.84802876921706", "82.6147645494468", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "41.6", "4.83214320637867", "82.4588243301682", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "41.7", "4.81826726770138", "82.3033062010554", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "41.8", "4.8079792331401", "82.1482094038389", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "41.9", "4.78832770667432", "81.9935331749498", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "42", "4.77240929017601", "81.8392767455964", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "42.1", "4.75823056713608", "81.685439341841", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "42.2", "4.75359889880641", "81.5320201846765", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "42.3", "4.7279954549942", "81.3790184901031", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "42.4", "4.71588865836974", "81.2264334692057", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "42.5", "4.69764113365507", "81.0742643282293", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "42.6", "4.6878655955709", "80.9225102686569", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "42.7", "4.66718555541341", "80.7711704872851", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "42.8", "4.65374671078516", "80.6202441763011", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "42.9", "4.63664589168335", "80.469730523584", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "43", "4.62813838446001", "80.3196287116534", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "43.1", "4.60603856211599", "80.1699379200009", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "43.2", "4.59160301019727", "80.0206573229103", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "43.3", "4.57537926264567", "79.8717860906603", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "43.4", "4.56683835218486", "79.7233233893752", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "43.5", "4.5466829925077", "79.5752683810991", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "43.6", "4.53509692390573", "79.4276202238712", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "43.7", "4.51396408026473", "79.2803780717997", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "43.8", "4.49757180427591", "79.1335410751365", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "43.9", "4.48323620888109", "78.9871083803512", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "44", "4.46765990088261", "78.841079130204", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "44.1", "4.45251243988094", "78.6954524638192", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "44.2", "4.42690120646078", "78.5502275167586", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "44.3", "4.42180523662279", "78.4054034210931", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "44.4", "4.40162334029236", "78.2609793054756", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "44.5", "4.39112648672764", "78.1169542952123", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "44.6", "4.37208695489569", "77.9732375123344", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "44.7", "4.36048752369005", "77.8300980756686", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "44.8", "4.34405639155261", "77.6872651009082", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "44.9", "4.32989914770126", "77.5448277006828", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "45", "4.31492555137483", "77.4027849846281", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "45.1", "4.29937164571914", "77.2611360594553", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "45.2", "4.28573182705169", "77.1198800290198", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "45.3", "4.26891481080662", "76.9790159943895", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "45.4", "4.25402367640162", "76.8385430539129", "1.0");

invoke(material, 'AddDispersionFittingValueEps', "45.5", "4.23853796077151", "76.698460303286", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "45.6", "4.22134083392119", "76.5587668356201", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "45.7", "4.20824995612951", "76.4194617415073", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "45.8", "4.19787352047356", "76.2805441090875", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "45.9", "4.17805921741872", "76.1420130241128", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "46", "4.16561432285858", "76.0038675700135", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "46.1", "4.14797374188673", "75.8661068279619", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "46.2", "4.13339915171485", "75.7287298769365", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "46.3", "4.11800111957608", "75.5917357937854", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "46.4", "4.10681927181962", "75.4551236532898", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "46.5", "4.08814854882837", "75.3188925282255", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "46.6", "4.08125147606346", "75.1830414894256", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "46.7", "4.05842285122975", "75.0475696058414", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "46.8", "4.04512327402801", "74.9124759446036", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "46.9", "4.02883048601641", "74.7777597510826", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "47", "4.02205628165901", "74.643419548948", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "47.1", "3.99937756396276", "74.5094549402282", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "47.2", "3.98665829020348", "74.3758648053691", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "47.3", "3.97006986076799", "74.2426482032923", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "47.4", "3.94280888389793", "74.1098041914524", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "47.5", "3.94091282996195", "73.9773318258947", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "47.6", "3.92074171238068", "73.8452301613115", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "47.7", "3.91191161534581", "73.7134982510979", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "47.8", "3.88775903105434", "73.5821351474078", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "47.9", "3.88307106298617", "73.4511399012085", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "48", "3.8746793165593", "73.3205115623351", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "48.1", "3.85439573277692", "73.1902491795447", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "48.2", "3.84182576366196", "73.0603518005693", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "48.3", "3.82588990958685", "72.9308184721687", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "48.4", "3.80743542886385", "72.8016482401831", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "48.5", "3.79755761400542", "72.6728401495843", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "48.6", "3.77530230092747", "72.544393244527", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "48.7", "3.76940261270302", "72.4163065683997", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "48.8", "3.75272554817302", "72.2885791638742", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "48.9", "3.74142842841912", "72.1612100729558", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "49", "3.75195383267379", "72.0341983370316", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "49.1", "3.71363834959133", "71.9075429969196", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "49.2", "3.69703800394824", "71.7812430929162", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "49.3", "3.68603543963863", "71.6552976648437", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "49.4", "3.7013145203865", "71.5297057520972", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "49.5", "3.65862254591068", "71.4044663936907", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "49.6", "3.63648752385905", "71.2795786283034", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "49.7", "3.63140230831665", "71.1550414943243", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "49.8", "3.64368726144744", "71.0308540298974", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "49.9", "3.60437716764223", "70.9070152729658", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "50", "3.59976434201793", "70.7835242613152", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "50.1", "3.5775493735692", "70.6603800326175", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "50.2", "3.54819151544362", "70.5375816244729", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "50.3", "3.55092099240579", "70.4151280744524", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "50.4", "3.52983014464927", "70.2930184201395", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "50.5", "3.5244939145389", "70.1712516991711", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "50.6", "3.52024012496485", "70.0498269492779", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "50.7", "3.49826986161847", "69.928743208325", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "50.8", "3.47768760728348", "69.8079995143515", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "50.9", "3.4722503934831", "69.6875949056088", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "51", "3.46879969770266", "69.5675284206002", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "51.1", "3.44643691483593", "69.447799081184", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "51.2", "3.43849382980699", "69.3284059772835", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "51.3", "3.42083068167987", "69.2093480975796", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "51.4", "3.39388248219674", "69.0906244988919", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "51.5", "3.39543280752077", "68.9722342215428", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "51.6", "3.39283574420227", "68.8541763063274", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "51.7", "3.37024426934696", "68.736449794549", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "51.8", "3.34405740633425", "68.6190537280532", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "51.9", "3.34526591339184", "68.5019871492632", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "52", "3.32615393431617", "68.3852491012124", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "52.1", "3.32049840668943", "68.2688386275787", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "52.2", "3.29505264386252", "68.1527547727168", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "52.3", "3.29594251242717", "68.0369965816909", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "52.4", "3.31979104121389", "67.9215631003065", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "52.5", "3.2715985510658", "67.806453375142", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "52.6", "3.2595061949488", "67.6916664535795", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "52.7", "3.24746696551493", "67.5772013838361", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "52.8", "3.2354808975466", "67.4630572149929", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "52.9", "3.22354801551799", "67.3492329970261", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "53", "3.2116683377591", "67.2357277808352", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "53.1", "3.19984187667747", "67.1225406182725", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "53.2", "3.18211559322812", "67.0096705621713", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "53.3", "3.1763486247006", "66.897166663739", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "53.4", "3.16468182979275", "66.7848779857593", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "53.5", "3.15306824372778", "66.6729535762703", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "53.6", "3.13578499011145", "66.5613424949404", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "53.7", "3.13000063046347", "66.4500437999199", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "53.8", "3.10732417558727", "66.339056550502", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "53.9", "3.10714559815608", "66.2283798071486", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "54", "3.09029597958395", "66.1180126315149", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "54.1", "3.08450288043217", "66.0079540864745", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "54.2", "3.09646344168401", "65.8982032361439", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "54.3", "3.06207213498992", "65.788759145906", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "54.4", "3.04564680486792", "65.679620882434", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "54.5", "3.03985294715718", "65.5707875137147", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "54.6", "3.0524413470725", "65.462258109071", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "54.7", "3.01784483331808", "65.3540317391844", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "54.8", "3.01245679957217", "65.2461074761175", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "54.9", "2.99604724421349", "65.1384843933351", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "55", "2.99649929984132", "65.0311615657257", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "55.1", "2.97445956811969", "64.924138069623", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "55.2", "2.9588550363962", "64.8174129828256", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "55.3", "2.95308113390913", "64.7109853846181", "1.0");

invoke(material, 'AddDispersionFittingValueEps', '55.4', "2.96378320110699", "64.6048543557905", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '55.5', "2.93191121399752", "64.4990189786579", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '55.6', "2.91670335280772", "64.3934783370798", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '55.7', "2.91094902718159", "64.2882315164789", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '55.8', "2.91747382436104", "64.1832776038596", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '55.9', "2.89019374137064", "64.0786156878262", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '56', "2.87537352268728", "63.9742448586009", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '56.1', "2.86964447621645", "63.8701642080411", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '56.2', "2.86365622813153", "63.766372829657", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '56.3', "2.849380830564415", "63.6628698186279", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '56.4', "2.84016995216782", "63.5596542718194", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '56.5', "2.82916026028877", "63.4567252877993", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '56.6', "2.82126543931268", "63.3540819668537", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '56.7', "2.80922332983884", "63.2517234110024", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '56.8', "2.79515173805568", "63.1496487240146", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '56.9', "2.78948846529267", "63.0478570114236", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '57', "2.77150065053017", "62.9463473805417", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '57.1', "2.76995458112822", "62.8451189404746", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '57.2', "2.76096522386566", "62.7441708021356", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '57.3', "2.75062055739104", "62.6435020782595", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '57.4', "2.75998226742912", "62.5431118834158", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '57.5', "2.7314852417024", "62.4429993340226", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '57.6', "2.72585281759377", "62.3431635483589", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '57.7', "2.7125474511904", "62.2436036465781", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '57.8', "2.70451858894816", "62.1443187507197", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '57.9', "2.69380597434727", "62.0453079847221", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '58', "2.68465510963499", "61.946570474434", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '58.1', "2.675259572815", "61.8481053476267", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '58.2', "2.65877094809751", "61.7499117340047", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '58.3', "2.6569069831018", "61.6519887652174", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '58.4', "2.65131090296416", "61.5543355748698", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '58.5', "2.63874691823228", "61.4569512985331", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '58.6', "2.62272971210552", "61.3598350737551", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '58.7', "2.62077806933257", "61.2629860400704", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '58.8', "2.61905049626134", "61.1664033390102", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '58.9', "2.6029991071544", "61.0700861141117", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '59', "2.59423804455651", "60.9740335109281", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '59.1', "2.58540868353876", "60.8782446770371", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '59.2', "2.57652588270692", "60.7827187620504", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '59.3', "2.56800543282145", "60.6874549176219", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '59.4', "2.55837326952782", "60.5924522974567", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '59.5', "2.55078797318374", "60.4977100573188", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '59.6', "2.53906966423548", "60.4032273550394", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '59.7', "2.53375490794847", "60.3090033505249", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '59.8', "2.53003707962002", "60.2150372057641", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '59.9', "2.51690482682466", "60.121328084836", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '60', "2.50549061994549", "60.0278751539163", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '60.1', "2.50023630710183", "59.9346775812853", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '60.2', "2.4859729072826", "59.8417345373337", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '60.3', "2.48374791479659", "59.7490451945698", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '60.4', "2.4726302098485", "59.6566087276257", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '60.5', "2.46743820575128", "59.5644243132632", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '60.6', "2.46302407144787", "59.47249113038", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '60.7', "2.45130572668931", "59.3808083600152", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '60.8', "2.44047710464509", "59.2893751853554", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '60.9', "2.43534901622616", "59.1981907917395", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '61', "2.42660927508461", "59.107254366664", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '61.1', "2.4195660583817", "59.0165650997882", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '61.2', "2.41138027199197", "58.9261221829392", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '61.3', "2.40395702079228", "58.8359248101158", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '61.4', "2.39088018156437", "58.7459721774938", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '61.5', "2.3885187810355", "58.6562634834299", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '61.6', "2.37824657218902", "58.5667979284656", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '61.7', "2.37325040204765", "58.4775747153318", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '61.8', "2.36459289752076", "58.3885930489518", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '61.9', "2.35815039565748", "58.2998521364456", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '62', "2.35007715124736", "58.2113511871329", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '62.1', "2.34321727082407", "58.1230894125366", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '62.2', "2.34246301823379", "58.035066026386", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '62.3', "2.32844953438436", "57.9472802446193", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '62.4', "2.32055311018609", "57.8597312853873", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '62.5', "2.31384569176846", "57.7724183690551", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '62.6', "2.30185480885078", "57.6853407182053", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '62.7', "2.29940424768334", "57.59849755764", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '62.8', "2.29184588500673", "57.5118881143833", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '62.9', "2.28512370676642", "57.4255116176831", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '63', "2.28086330691422", "57.3393672990132", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '63.1', "2.27100257421013", "57.2534543920752", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '63.2', "2.26176042315848", "57.1677721327998", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '63.3', "2.25703935635845", "57.0823197593486", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '63.4', "2.24572062513395", "56.9970965121157", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '63.5', "2.24323256127599", "56.9121016337281", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '63.6', "2.23933024040598", "56.8273343690481", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '63.7', "2.22958069929139", "56.7427939651729", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '63.8', "2.22538163437953", "56.6584796714367", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '63.9', "2.21608228351563", "56.5743907394106", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '64', "2.2090156957682", "56.4905264229035", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '64.1', "2.20273583033551", "56.4068859779628", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '64.2', "2.19519782919051", "56.323468628744", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '64.3', "2.18953985988429", "56.2402737381632", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '64.4', "2.18572347068332", "56.1573004665932", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '64.5', "2.17649289649004", "56.0745481131674", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '64.6', "2.1661103893403", "55.9920159451279", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '64.7', "2.16359346910151", "55.9097032319556", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '64.8', "2.15967426021055", "55.82760924537", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '64.9', "2.15084011169373", "55.7457332593283", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '65', "2.14358417300096", "55.6640745500259", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '65.1', "2.13823136365336", "55.5826323958945", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '65.2', "2.13013365235255", "55.5014060776024", "1.0");

invoke(material, 'AddDispersionFittingValueEps', "65.3", "2.12576577014447", "55.4203948780531", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "65.4", "2.11889283794639", "55.3395980823847", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "65.5", "2.11344188245578", "55.2590149779685", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "65.6", "2.10697170474851", "55.1786448544883", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "65.7", "2.10125825832995", "55.0984870035388", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "65.8", "2.09447322787305", "55.0185407194244", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "65.9", "2.08921346227533", "54.9388052983581", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "66", "2.08282019256251", "54.8592800388594", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "66.1", "2.07730606586145", "54.7799642416734", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "66.2", "2.07554670501892", "54.7008752097684", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "66.3", "2.06553464799781", "54.6219582483346", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "66.4", "2.05805405990837", "54.5432666647825", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "66.5", "2.05389779519784", "54.4647817687401", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "66.6", "2.04747636599149", "54.3865028720518", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "66.7", "2.04239410182767", "54.3084292887756", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "66.8", "2.03875037587779", "54.2305603351815", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "66.9", "2.03102217034055", "54.1528953297488", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "67", "2.0247068433872", "54.075433593164", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "67.1", "2.01978061149772", "53.9981744483184", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "67.2", "2.0138649757597", "53.9211172203054", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "67.3", "2.00866804457559", "53.8442612364183", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "67.4", "2.00253108117127", "53.7676058261472", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "67.5", "1.99768309756066", "53.6911503211769", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "67.6", "1.99077193003328", "53.614890553835", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "67.7", "1.98682440733197", "53.538836364832", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "67.8", "1.9808622268745", "53.4629765877733", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "67.9", "1.9760906198316", "53.3873140646411", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "68", "1.97044872510493", "53.3118481380492", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "68.1", "1.96548039022397", "53.2365781527878", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "68.2", "1.9614161880574", "53.1615034558209", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "68.3", "1.95499238300434", "53.0866233962828", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "68.4", "1.9517044341299", "53.0119373254752", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "68.5", "1.94462527233654", "52.9374445968631", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "68.6", "1.9368227052295", "52.8631445566072", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "68.7", "1.9343777417809", "52.7890365908848", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "68.8", "1.93006437963723", "52.715120031237", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "68.9", "1.924248848481246", "52.6413942492144", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "69", "1.91873671182814", "52.5678586090489", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "69.1", "1.91423620472964", "52.4945124771149", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "69.2", "1.90801598863474", "52.421352219257", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "69.3", "1.90433961479421", "52.3483862141296", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "69.4", "1.89696693919992", "52.2756048265059", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "69.5", "1.89455743832271", "52.2030104339616", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "69.6", "1.89047481372445", "52.1306024135268", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "69.7", "1.88488840876945", "52.0583801443512", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "69.8", "1.879652576607", "51.9863430076998", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "69.9", "1.87533126980179", "51.914490386949", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "70", "1.86942943225398", "51.8428216675826", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "70.1", "1.86588475736758", "51.7713362371873", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "70.2", "1.86089333743524", "51.7000334854488", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "70.3", "1.85654768975575", "51.6289128041476", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "70.4", "1.85272999345414", "51.5579735871544", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "70.5", "1.84731878764948", "51.4872152304261", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "70.6", "1.84054473024783", "51.4166371320014", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "70.7", "1.83819685417298", "51.3462386919964", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "70.8", "1.8325965443615", "51.2760193126", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "70.9", "1.82918068493162", "51.2059783980696", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "71", "1.82259456387578", "51.1361153547266", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "71.1", "1.82026908604606", "51.06642950952", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "71.2", "1.81481349237925", "50.9969205171813", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "71.3", "1.81146087418075", "50.9275875459007", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "71.4", "1.80682755604329", "50.8584300916418", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "71.5", "1.80275487656598", "50.7894475709771", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "71.6", "1.79744010726681", "50.7206394025157", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "71.7", "1.79414993101556", "50.652005006898", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "71.8", "1.7895895424356", "50.5835438067911", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "71.9", "1.78564488593861", "50.5152522688846", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "72", "1.78046715136141", "50.4471386938847", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "72.1", "1.7772386003466", "50.3791936365105", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "72.2", "1.77618257370506", "50.3114194854883", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "72.3", "1.7689299438582", "50.2438156735469", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "72.4", "1.76468723662861", "50.1763816354133", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "72.5", "1.76071779668511", "50.1091168078066", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "72.6", "1.75482988567539", "50.0420206294342", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "72.7", "1.75260104964938", "49.9750925409861", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "72.8", "1.74848809507666", "49.90833198513", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "72.9", "1.74457860414909", "49.8417384065966", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "73", "1.7388526484209", "49.7753112517239", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "73.1", "1.73664937215556", "49.7090499693529", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "73.2", "1.73266267328417", "49.642954009922", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "73.3", "1.7288122761928", "49.577022825912", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "73.4", "1.72478859640353", "49.5112558717508", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "73.5", "1.72106624931546", "49.4456526038087", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "73.6", "1.71789156787203", "49.3802124803928", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "73.7", "1.71341023508367", "49.314934961742", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "73.8", "1.70799411162721", "49.2498195100216", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "73.9", "1.70584318753431", "49.1848655893184", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "74", "1.70204262088178", "49.1200726656352", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "74.1", "1.69836407114963", "49.0554402068858", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "74.2", "1.69458695226557", "48.9909676828893", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "74.3", "1.69097186082227", "48.9266545653652", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "74.4", "1.6877990030717", "48.8625003279282", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "74.5", "1.68366554181813", "48.7985044460823", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "74.6", "1.67995771019616", "48.7346663972161", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "74.7", "1.67644410973619", "48.6709856605972", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "74.8", "1.67212742160088", "48.6074617173668", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "74.9", "1.66930657046593", "48.5440940505346", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "75", "1.66573837135792", "48.4808821449731", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "75.1", "1.66225194014218", "48.4178254874123", "1.0");

invoke(material, 'AddDispersionFittingValueEps', "75.2", "1.65941618254884", "48.3549235664346", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "75.3", "1.65527924509811", "48.2921758724691", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "75.4", "1.65043030158938", "48.2295818978862", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "75.5", "1.64838752181546", "48.1671411364925", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "75.6", "1.64495219165304", "48.1048530845249", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "75.7", "1.64157581687337", "48.0427172396456", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "75.8", "1.63951405857152", "47.9807331014364", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "75.9", "1.63484318689477", "47.9189001712934", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "76", "1.63151211865824", "47.8572179524214", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "76.1", "1.62818869849114", "47.7956859498286", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "76.2", "1.62740414887497", "47.7343036703211", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "76.3", "1.62161142820574", "47.6730706224973", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "76.4", "1.61835208305375", "47.6119863167427", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "76.5", "1.61511046245479", "47.551050265224", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "76.6", "1.61064497853593", "47.4902619818841", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "76.7", "1.60868489746762", "47.4296209824364", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "76.8", "1.60553387462509", "47.3691267843591", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "76.9", "1.60233383922499", "47.3087789068902", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "77", "1.59925715062647", "47.2485768710213", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "77.1", "1.59605640339635", "47.188520199493", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "77.2", "1.59358922759885", "47.1286084167886", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "77.3", "1.5898517152755", "47.068841049129", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "77.4", "1.58682053738828", "47.0092176244672", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "77.5", "1.58371890971559", "46.9497376724827", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "77.6", "1.58071206601647", "46.8904007245759", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "77.7", "1.57765713106248", "46.8312063138628", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "77.8", "1.57354265048136", "46.7721539751695", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "77.9", "1.5716653308757", "46.7132432450265", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "78", "1.56933144996797", "46.6544736616633", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "78.1", "1.5674327891912", "46.5958447650029", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "78.2", "1.56293596270626", "46.5373560966564", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "78.3", "1.55988954097335", "46.4790071999173", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "78.4", "1.556463698523", "46.4207976197563", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "78.5", "1.55410350088421", "46.3627269028152", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "78.6", "1.55244267491863", "46.3047945974022", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "78.7", "1.54838434943262", "46.247000253486", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "78.8", "1.54504461988968", "46.1893434226901", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "78.9", "1.54273128647508", "46.1318236582877", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "79", "1.53893819026475", "46.0744405151961", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "79.1", "1.53714352087143", "46.0171935499712", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "79.2", "1.53443939445708", "45.9600823208018", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "79.3", "1.53162027041239", "45.9031063875047", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "79.4", "1.52904631062212", "45.8462653115185", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "79.5", "1.52616076174624", "45.7895586558988", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "79.6", "1.52298637008008", "45.7329859853124", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "79.7", "1.52076423030527", "45.6765468660317", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "79.8", "1.51717022354373", "45.6202408659297", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "79.9", "1.51542992023176", "45.5640675544742", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "80", "1.51342434489834", "45.5080265027224", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "80.1", "1.51015708430346", "45.4521172833158", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "80.2", "1.50770745796962", "45.3963394704742", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "80.3", "1.50494498385922", "45.3406926399908", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "80.4", "1.50346291478211", "45.2851763692265", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "80.5", "1.49979288872365", "45.2297902371045", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "80.6", "1.49743578834756", "45.1745338241052", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "80.7", "1.49470007713216", "45.1194067122605", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "80.8", "1.49228195885936", "45.0644084851483", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "80.9", "1.48966583565561", "45.0095387278875", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "81", "1.48736113583824", "44.9547970271324", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "81.1", "1.48468945912467", "44.9001829710675", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "81.2", "1.48335566691453", "44.8456961494017", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "81.3", "1.47977025055431", "44.7913361533636", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "81.4", "1.47752931572138", "44.7371025756958", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "81.5", "1.47490752106809", "44.6829950106494", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "81.6", "1.47210914625141", "44.629013053979", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "81.7", "1.47010058982243", "44.5751563029374", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "81.8", "1.46796797328303", "44.5214243562698", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "81.9", "1.46534878393092", "44.4678168142093", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "82", "1.46261967358875", "44.4143332784709", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "82.1", "1.46065143838874", "44.3609733522465", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "82.2", "1.4585520441067", "44.3077366401996", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "82.3", "1.45600789599677", "44.2546227484603", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "82.4", "1.45430100958095", "44.2016312846196", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "82.5", "1.45141750728621", "44.148761857243", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "82.6", "1.44843562441455", "44.0960140782721", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "82.7", "1.44687963044434", "44.0433875582059", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "82.8", "1.44428353999781", "43.9908819109089", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "82.9", "1.44239363123406", "43.9384967511993", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "83", "1.4422605131594", "43.8862316953251", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "83.1", "1.43795888292853", "43.8340863609588", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "83.2", "1.43589356284192", "43.7820603671926", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "83.3", "1.43357476622691", "43.7301533345329", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "83.4", "1.43074613364631", "43.6783648848952", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "83.5", "1.42924066918404", "43.6266946415991", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "83.6", "1.4267705875769", "43.5751422293631", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "83.7", "1.42495598713535", "43.5237072742993", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "83.8", "1.42310475018012", "43.4723894039087", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "83.9", "1.42072012262248", "43.4211882470759", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "84", "1.41921456146078", "43.3701034340636", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "84.1", "1.41653248531944", "43.3191345965084", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "84.2", "1.41558276270252", "43.2682813674151", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "84.3", "1.41239249195896", "43.2175433811516", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "84.4", "1.41004179045348", "43.1669202734445", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "84.5", "1.40829956625916", "43.1164116813733", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "84.6", "1.4056848581382", "43.066017243366", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "84.7", "1.40425313885059", "43.0157365991936", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "84.8", "1.40238044026746", "42.9655693899656", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "84.9", "1.40025264720354", "42.9155152581247", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', "85", "1.3977052446194", "42.8655738474419", "1.0");



invoke(material, 'AddDispersionFittingValueEps', '85.1', "1.3962975355586", "42.8157448030116", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '85.2', "1.39447650757888", "42.7660277712466", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '85.3', "1.39238725484092", "42.7164223998732", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '85.4', "1.39245460567858", "42.6669283379265", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '85.5', "1.38852126261628", "42.6175452357449", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '85.6', "1.3863798816551", "42.5682727449661", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '85.7', "1.38469902299232", "42.5191105185213", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '85.8', "1.38228058136729", "42.4700582106309", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '85.9', "1.38092000656177", "42.4211154767996", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '86', "1.37923220166561", "42.3722819738116", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '86.1', "1.3771836903295", "42.3235573597252", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '86.2', "1.37482701581134", "42.274941293869", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '86.3', "1.37348957564224", "42.2264334368361", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '86.4', "1.37141063115848", "42.1780334504801", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '86.5', "1.3698370981197", "42.1297409979097", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '86.6', "1.36846413100987", "42.0815557434845", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '86.7', "1.36622580758538", "42.0334773528098", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '86.8', "1.36419699366301", "41.985505492732", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '86.9', "1.36265518799805", "41.9376398313342", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '87', "1.36041688748524", "41.889880037931", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '87.1', "1.35912474738398", "41.8422257830641", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '87.2', "1.35856102797597", "41.7946767384976", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '87.3', "1.35663399976914", "41.7472325772132", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '87.4', "1.35628501920227", "41.6998929734057", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '87.5', "1.35218246511227", "41.6526576024784", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '87.6', "1.35074400197393", "41.6055261410381", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '87.7', "1.34876966923848", "41.558498266891", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '87.8', "1.34664319906856", "41.5115736590378", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '87.9', "1.34539514377298", "41.4647519976692", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '88', "1.34350887087239", "41.4180329641612", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '88.1', "1.34205842607567", "41.3714162410709", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '88.2', "1.34205818009985", "41.3249015121314", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '88.3', "1.3387590591761", "41.2784884622479", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '88.4', "1.33795414097826", "41.232176774926", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '88.5', "1.33549659170888", "41.1859661451007", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '88.6', "1.33347579085662", "41.1398562534653", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '88.7', "1.33227057784977", "41.0938467921338", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '88.8', "1.33153541798169", "41.0479374518027", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '88.9', "1.32980857725222", "41.0021279243133", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '89', "1.32711041391424", "40.9564179026475", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '89.1', "1.32592615498426", "40.9108070809232", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '89.2', "1.32417146568803", "40.86529515439", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '89.3', "1.32280688146628", "40.8198818194246", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '89.4', "1.32088594851954", "40.7745667735269", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '89.5', "1.31972232340908", "40.729349715315", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '89.6', "1.31918947891613", "40.6842303445212", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '89.7', "1.3166720887524", "40.6392083619878", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '89.8', "1.3147990206558", "40.5942834696625", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '89.9', "1.31365573660431", "40.5494553705941", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '90', "1.31198312170202", "40.5047237689285", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '90.1', "1.31067286718075", "40.4600883699042", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '90.2', "1.30884633870537", "40.415448879848", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '90.3', "1.30772307674595", "40.3711050061708", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '90.4', "1.30608989736395", "40.3267564573635", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '90.5', "1.30480596655302", "40.2825029429925", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '90.6', "1.30302469155133", "40.2383441736957", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '90.7', "1.30192114278557", "40.1942798611783", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '90.8', "1.30032637033939", "40.1503097182087", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '90.9', "1.29906821649943", "40.1064334586139", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '91', "1.29733094671245", "40.0626507972758", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '91.1', "1.2962468035651", "40.0189614501271", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '91.2', "1.29468943783911", "39.9753651341467", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '91.3', "1.29345652461069", "39.9318615673561", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '91.4', "1.29176204851173", "39.8884504688148", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '91.5', "1.29069700496558", "39.845131558617", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '91.6', "1.28917607331537", "39.8019045578865", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '91.7', "1.28796787460432", "39.7587691887735", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '91.8', "1.28631501627859", "39.7157251744502", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '91.9', "1.28526876809137", "39.6727722391067", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '92', "1.28378332466796", "39.6299101079472", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '92.1', "1.28259932452613", "39.5871385071858", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '92.2', "1.28098694258475", "39.5444571640429", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '92.3', "1.2799591874887", "39.5018658067405", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '92.4', "1.27850831248401", "39.4593641644991", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '92.5', "1.27734800498612", "39.4169519675331", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '92.6', "1.27577499151379", "39.3746289470472", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '92.7', "1.27476542939907", "39.3323948352325", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '92.8', "1.27334822831224", "39.2902493652622", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '92.9', "1.2722111742918", "39.2481922712884", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '93', "1.27231059578419", "39.2062232884377", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '93.1', "1.26968473004696", "39.1643421528073", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '93.2', "1.26830033297107", "39.1225486014615", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '93.3', "1.26718593244004", "39.0808423724278", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '93.4', "1.26568846098592", "39.0392232046928", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '93.5', "1.26471439396218", "38.9976908381986", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '93.6', "1.26336195489071", "38.9562450138392", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '93.7', "1.26226978808266", "38.9148854734561", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '93.8', "1.26257893420543", "38.8736119598355", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '93.9', "1.25985179233616", "38.8324242167036", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '94', "1.25853048848869", "38.7913219887233", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '94.1', "1.25746008827343", "38.7503050214906", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '94.2', "1.25603410395337", "38.7093730615307", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '94.3', "1.25509436141214", "38.6685258562941", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '94.4', "1.25380339257897", "38.6277631541535", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '94.5', "1.25275430118848", "38.5870847043996", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '94.6', "1.2513626132507", "38.5464902572377", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '94.7', "1.25043960090914", "38.5059795637838", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '94.8', "1.24917818881415", "38.465523760615", "1.0");  
invoke(material, 'AddDispersionFittingValueEps', '94.9', "1.24814995770395", "38.4252084469978", "1.0");

```

invoke(material, 'AddDispersionFittingValueEps', "95", "1.24679163872675", "38.3849475304198", "1.0");
invoke(material, 'AddDispersionFittingValueEps', "95.1", "1.24588507247894", "38.3447693810512", "1.0");
invoke(material, 'AddDispersionFittingValueEps', "95.2", "1.24465246016026", "38.3046737545083", "1.0");
invoke(material, 'AddDispersionFittingValueEps', "95.3", "1.24364464986984", "38.264660407297", "1.0");
invoke(material, 'AddDispersionFittingValueEps', "95.4", "1.24231879939511", "38.2247290968089", "1.0");
invoke(material, 'AddDispersionFittingValueEps', "95.5", "1.24142839819629", "38.1848795813178", "1.0");
invoke(material, 'AddDispersionFittingValueEps', "95.6", "1.24022384940448", "38.1451116199763", "1.0");
invoke(material, 'AddDispersionFittingValueEps', "95.7", "1.23923602941633", "38.1054249728121", "1.0");
invoke(material, 'AddDispersionFittingValueEps', "95.8", "1.24059734621356", "38.0658194007247", "1.0");
invoke(material, 'AddDispersionFittingValueEps', "95.9", "1.23706725908155", "38.0262946654819", "1.0");
invoke(material, 'AddDispersionFittingValueEps', "96", "1.23589005769451", "37.9868505297163", "1.0");
invoke(material, 'AddDispersionFittingValueEps', "96.1", "1.23492180629262", "37.9474867569216", "1.0");
invoke(material, 'AddDispersionFittingValueEps', "96.2", "1.23365829523575", "37.9082031114495", "1.0");
invoke(material, 'AddDispersionFittingValueEps', "96.3", "1.23279939365544", "37.8689993585064", "1.0");
invoke(material, 'AddDispersionFittingValueEps', "96.4", "1.23164884310998", "37.8298752641492", "1.0");
invoke(material, 'AddDispersionFittingValueEps', "96.5", "1.23069974723768", "37.790830595283", "1.0");
invoke(material, 'AddDispersionFittingValueEps', "96.6", "1.2294661570317", "37.7518651196567", "1.0");
invoke(material, 'AddDispersionFittingValueEps', "96.7", "1.22862259652577", "37.7129786058604", "1.0");
invoke(material, 'AddDispersionFittingValueEps', "96.8", "1.23136067102149", "37.6741708233215", "1.0");
invoke(material, 'AddDispersionFittingValueEps', "96.9", "1.22656767438264", "37.6354415423017", "1.0");
invoke(material, 'AddDispersionFittingValueEps', "97", "1.22536320448877", "37.5967905338932", "1.0");
invoke(material, 'AddDispersionFittingValueEps', "97.1", "1.22453471700544", "37.5582175700162", "1.0");
invoke(material, 'AddDispersionFittingValueEps', "97.2", "1.22343545394225", "37.5197224234146", "1.0");
invoke(material, 'AddDispersionFittingValueEps', "97.3", "1.22252346388435", "37.4813048676535", "1.0");
invoke(material, 'AddDispersionFittingValueEps', "97.4", "1.2213473368809", "37.4429646771155", "1.0");
invoke(material, 'AddDispersionFittingValueEps', "97.5", "1.22053365776129", "37.4047016269973", "1.0");
invoke(material, 'AddDispersionFittingValueEps', "97.6", "1.21945906775614", "37.3665154933071", "1.0");
invoke(material, 'AddDispersionFittingValueEps', "97.7", "1.21856504458949", "37.3284060528605", "1.0");
invoke(material, 'AddDispersionFittingValueEps', "97.8", "1.21741650547259", "37.2903730832779", "1.0");
invoke(material, 'AddDispersionFittingValueEps', "97.9", "1.21661737349342", "37.2524163629811", "1.0");
invoke(material, 'AddDispersionFittingValueEps', "98", "1.21556683284778", "37.2145356711898", "1.0");
invoke(material, 'AddDispersionFittingValueEps', "98.1", "1.21469039672903", "37.1767307879187", "1.0");
invoke(material, 'AddDispersionFittingValueEps', "98.2", "1.213568712235", "37.1390014939746", "1.0");
invoke(material, 'AddDispersionFittingValueEps', "98.3", "1.2127838696447", "37.1013475709524", "1.0");
invoke(material, 'AddDispersionFittingValueEps', "98.4", "1.21175677160843", "37.0637688012328", "1.0");
invoke(material, 'AddDispersionFittingValueEps', "98.5", "1.21089755064242", "37.0262649679785", "1.0");
invoke(material, 'AddDispersionFittingValueEps', "98.6", "1.20980200859114", "36.9888358551315", "1.0");
invoke(material, 'AddDispersionFittingValueEps', "98.7", "1.20903120113954", "36.9514812474098", "1.0");
invoke(material, 'AddDispersionFittingValueEps', "98.8", "1.20802695543235", "36.9142009303043", "1.0");
invoke(material, 'AddDispersionFittingValueEps', "98.9", "1.20718458553099", "36.8769946900757", "1.0");
invoke(material, 'AddDispersionFittingValueEps', "99", "1.20611449418972", "36.8398623137514", "1.0");
invoke(material, 'AddDispersionFittingValueEps', "99.1", "1.20535747115185", "36.8028035891224", "1.0");
invoke(material, 'AddDispersionFittingValueEps', "99.2", "1.20437550349841", "36.7658183047404", "1.0");
invoke(material, 'AddDispersionFittingValueEps', "99.3", "1.20354962824041", "36.7289062499145", "1.0");
invoke(material, 'AddDispersionFittingValueEps', "99.4", "1.20265286256939", "36.6920672147086", "1.0");
invoke(material, 'AddDispersionFittingValueEps', "99.5", "1.20176082990162", "36.6553009899375", "1.0");
invoke(material, 'AddDispersionFittingValueEps', "99.6", "1.20087350231725", "36.6186073671651", "1.0");
invoke(material, 'AddDispersionFittingValueEps', "99.7", "1.19999085207109", "36.5819861387002", "1.0");
invoke(material, 'AddDispersionFittingValueEps', "99.8", "1.19911285159151", "36.5454370975945", "1.0");
invoke(material, 'AddDispersionFittingValueEps', "99.9", "1.19823947347936", "36.5089600376389", "1.0");
invoke(material, 'AddDispersionFittingValueEps', "100", "1.19737069050692", "36.4725547533608", "1.0");
invoke(material, 'UseGeneralDispersionEps', "True");
invoke(material, 'UseGeneralDispersionMu', "False");
invoke(material, 'NLAnisotropy', "False");
invoke(material, 'NLAStackingFactor', "1");
invoke(material, 'NLADirectionX', "1");
invoke(material, 'NLADirectionY', "0");
invoke(material, 'NLADirectionZ', "0");
invoke(material, 'Colour', "0.25", "0.25", "0.25");
invoke(material, 'Wireframe', "False");
invoke(material, 'Reflection', "False");
invoke(material, 'Allowoutline', "True");
invoke(material, 'Transparentoutline', "False");
invoke(material, 'Transparency', "0");
invoke(material, 'Create');
release(material);
end

```

[pickface.m](#): función para seleccionar una cara del objeto.

```

function pickface(mws,name,number)

%Function to select a pickface

Pick = invoke(mws,'Pick');
invoke(Pick,'PickFaceFromId',strcat('component:',name),num2str(number));

end

```

[rectanglefeed.m](#): función para dibujar un rectángulo.

```

function rectanglefeed(mws,x,y,h)

% Function to draw a rectanglefeed

    rectanf = invoke(mws,'Extrude');
    invoke(rectanf,'Reset');
    invoke(rectanf,'Name',"feedline1");
    invoke(rectanf,'Component',"component");
    invoke(rectanf,'Material',"PEC");
    invoke(rectanf,'Mode',"Pointlist");
    invoke(rectanf,'Height',h);
    invoke(rectanf,'Point',int2str(x(1)), int2str(y(1)));
    invoke(rectanf,'LineTo',int2str(x(2)), int2str(y(2)));
    invoke(rectanf,'LineTo',int2str(x(3)), int2str(y(3)));
    invoke(rectanf,'LineTo',int2str(x(4)), int2str(y(4)));
    invoke(rectanf,'LineTo',int2str(x(1)), int2str(y(1)));
    invoke(rectanf,'Create');
    release(rectanf);

end

subs_FR4lossy.m: función para definir el material FR4.

```

```

function subs_FR4lossy(mws)

% Define material: FR-4 (lossy)

material = invoke(mws,'material');
invoke(material,'Reset');
invoke(material,'Name',"FR-4 (lossy)");
invoke(material,'FrqType','all');
invoke(material,'Type','Normal');
invoke(material,'SetMaterialUnit','THz','um');
invoke(material,'Epsilon','4.3');
invoke(material,'Mue','1.0');
invoke(material,'Kappa','0.0');
invoke(material,'TanD','0.025');
invoke(material,'TanDFreq','10.0');
invoke(material,'TanDGiven','True');
invoke(material,'TanDModel','ConstTanD');
invoke(material,'KappaM','0.0');
invoke(material,'TanDM','0.0');
invoke(material,'TanDMFreq','0.0');
invoke(material,'TanDMGiven','False');
invoke(material,'TanDModel','ConstKappa');
invoke(material,'DispModelEps','None');
invoke(material,'DispModelMue','None');
invoke(material,'DispersiveFittingSchemeEps','General 1st');
invoke(material,'DispersiveFittingSchemeMue','General 1st');
invoke(material,'UseGeneralDispersionEps','False');
invoke(material,'UseGeneralDispersionMue','False');
invoke(material,'Rho','0.0');
invoke(material,'ThermalType','Normal');
invoke(material,'ThermalConductivity','0.3');
invoke(material,'SetActiveMaterial','all');
invoke(material,'Colour','0.94','0.82','0.76');
invoke(material,'Wireframe','False');
invoke(material,'Transparency','0');
invoke(material,'Create');
release(material);

end

```

subs\_glassepoxy.m: función para definir el material fibra de vidrio.

```

function subs_glassepoxy(mws)

    % Define material: FR-4 (glass epoxy)

    material = invoke(mws,'material');
    invoke(material,'Reset');
    invoke(material,'Name','FR-4 (glass epoxy)');
    invoke(material,'Rho','19000');
    invoke(material,'ThermalType','Normal');
    invoke(material,'ThermalConductivity','0');
    invoke(material,'SpecificHeat','0','J/K/kg');
    invoke(material,'DynamicViscosity','0');
    invoke(material,'Emissivity','0');
    invoke(material,'MetabolicRate','0.0');
    invoke(material,'VoxelConvection','0.0');
    invoke(material,'BloodFlow','0');
    invoke(material,'MechanicsType','Unused');
    invoke(material,'MaterialUnit','Frequency','THz');
    invoke(material,'MaterialUnit','Geometry','nm');
    invoke(material,'MaterialUnit','Time','ns');
    invoke(material,'MaterialUnit','Temperature','Kelvin');
    invoke(material,'Epsilon','4.4');
    invoke(material,'Mu','1');
    invoke(material,'Sigma','0');
    invoke(material,'TanD','0.02');
    invoke(material,'TanDFreq','0.0');
    invoke(material,'TanDGiven','False');
    invoke(material,'TanDModel','ConstTanD');
    invoke(material,'SetConstTanDStrategyEps','AutomaticOrder');
    invoke(material,'ConstTanDModelOrderEps','3');
    invoke(material,'DjordjevicSarkarUpperFreqEps','0');
    invoke(material,'SetElParametricConductivity','False');
    invoke(material,'ReferenceCoordSystem','Global');
    invoke(material,'CoordSystemType','Cartesian');
    invoke(material,'SigmaM','0');
    invoke(material,'TanDM','0.0');
    invoke(material,'TanDMFreq','0.0');
    invoke(material,'TanDMGiven','False');
    invoke(material,'TanDMModel','ConstTanD');
    invoke(material,'SetConstTanDStrategyMu','AutomaticOrder');
    invoke(material,'ConstTanDModelOrderMu','3');
    invoke(material,'DjordjevicSarkarUpperFreqMu','0');
    invoke(material,'SetMagParametricConductivity','False');
    invoke(material,'DispModelEps','None');
    invoke(material,'DispModelMu','None');
    invoke(material,'DispersiveFittingSchemeEps','Nth Order');
    invoke(material,'MaximalOrderNthModelFitEps','10');
    invoke(material,'ErrorLimitNthModelFitEps','0.1');
    invoke(material,'DispersiveFittingSchemeMu','Nth Order');
    invoke(material,'MaximalOrderNthModelFitMu','10');
    invoke(material,'ErrorLimitNthModelFitMu','0.1');
    invoke(material,'UseGeneralDispersionEps','False');
    invoke(material,'UseGeneralDispersionMu','False');
    invoke(material,'NLAnisotropy','False');
    invoke(material,'NLAStackingFactor','1');
    invoke(material,'NLADirectionX','1');
    invoke(material,'NLADirectionY','0');
    invoke(material,'NLADirectionZ','0');
    invoke(material,'Colour','0.501961','0.501961','1');
    invoke(material,'Wireframe','False');
    invoke(material,'Reflection','False');
    invoke(material,'Allowoutline','True');
    invoke(material,'Transparentoutline','False');
    invoke(material,'Transparency','0');
    invoke(material,'Create');
    release(material);
end

```

`subs_siliconlossy.m`: función para definir el material dióxido de silicio.

```

function subs_siliconlossy(mws)

    % Define material: Silicon (lossy)

    material = invoke(mws,'material');
    invoke(material,'Reset');
    invoke(material,'Name','Silicon (lossy)');
    invoke(material,'Rho',"2330.0");
    invoke(material,'ThermalType',"Normal");
    invoke(material,'ThermalConductivity',"148");
    invoke(material,'SpecificHeat',"700","J/K/kg");
    invoke(material,'DynamicViscosity',"0");
    invoke(material,'Emissivity',"0");
    invoke(material,'MetabolicRate',"0.0");
    invoke(material,'VoxelConvection',"0.0");
    invoke(material,'BloodFlow',"0");
    invoke(material,'MechanicsType',"Isotropic");
    invoke(material,'YoungsModulus',"112");
    invoke(material,'PoissonsRatio',"0.28");
    invoke(material,'ThermalExpansionRate',"5.1");
    invoke(material,'FrqType',"all");
    invoke(material,'Type',"Normal");
    invoke(material,'MaterialUnit',"Frequency","GHz");
    invoke(material,'MaterialUnit',"Geometry","mm");
    invoke(material,'MaterialUnit',"Time","s");
    invoke(material,'Epsilon',"11.9");
    invoke(material,'Mu',"1.0");
    invoke(material,'Sigma',"2.5e-004");
    invoke(material,'TanD',"0.0");
    invoke(material,'TanDFreq',"0.0");
    invoke(material,'TanDGiven',"False");
    invoke(material,'TanDModel',"ConstTanD");
    invoke(material,'SetConstTanDStrategyEps',"AutomaticOrder");
    invoke(material,'ConstTanDModelOrderEps',"3");
    invoke(material,'DjordjevicSarkarUpperFreqEps',"0");
    invoke(material,'SetElParametricConductivity',"False");
    invoke(material,'ReferenceCoordSystem',"Global");
    invoke(material,'CoordSystemType',"Cartesian");
    invoke(material,'SigmaM',"0.0");
    invoke(material,'TanDM',"0.0");
    invoke(material,'TanDMFreq',"0.0");
    invoke(material,'TanDMGiven',"False");
    invoke(material,'TanDMModel',"ConstTanD");
    invoke(material,'SetConstTanDStrategyMu',"AutomaticOrder");
    invoke(material,'ConstTanDModelOrderMu',"3");
    invoke(material,'DjordjevicSarkarUpperFreqMu',"0");
    invoke(material,'SetMagParametricConductivity',"False");
    invoke(material,'DispModelEps',"None");
    invoke(material,'DispModelMu',"None");
    invoke(material,'DispersiveFittingSchemeEps',"1st Order");
    invoke(material,'DispersiveFittingSchemeMu',"1st Order");
    invoke(material,'UseGeneralDispersionEps',"False");
    invoke(material,'UseGeneralDispersionMu',"False");
    invoke(material,'NLAnisotropy',"False");
    invoke(material,'NLAStackingFactor',"1");
    invoke(material,'NLADirectionX',"1");
    invoke(material,'NLADirectionY',"0");
    invoke(material,'NLADirectionZ',"0");
    invoke(material,'Colour',"0.501961","0.501961","0.752941");
    invoke(material,'Wireframe',"False");
    invoke(material,'Reflection',"False");
    invoke(material,'Allowoutline',"True");
    invoke(material,'Transparentoutline',"False");
    invoke(material,'Transparency',"0");
    invoke(material,'Create');
    release(material);

end

```

`transform.m`: función para realizar la transformación de un componente.

```

function transform(mws,name,type,x,y,z)

% Function to realize a transform component

    transform = invoke(mws,'Transform');
    invoke(transform,'Reset');
    invoke(transform,'Name',name);
    invoke(transform,'Vector',int2str(x), int2str(y),int2str(z));
    invoke(transform,'UsePickedPoints',"False");
    invoke(transform,'InvertPickedPoints',"False");
    invoke(transform,'MultipleObjects',int2str(1));
    invoke(transform,'GroupObjects',"False");
    invoke(transform,'Repetitions',"1");
    invoke(transform,'MultipleSelection',"False");
    invoke(transform,'Destination',"");
    invoke(transform,'Material',"PEC");
    invoke(transform,'Transform',"Shape",type);
    release(transform);

end

```

`triangle.m`: función para dibujar un rectángulo.

```

function triangle(mws,x,y,name,h)

% Function to draw a triangle

    extrude = invoke(mws,'Extrude');
    invoke(extrude,'Reset');
    invoke(extrude,'Name',"t" + int2str(name));
    invoke(extrude,'Component',"component");
    invoke(extrude,'Material',"PEC");
    invoke(extrude,'Mode',"Pointlist");
    invoke(extrude,'Height',h);
    invoke(extrude,'Point',int2str(x(1)), int2str(y(1)));
    invoke(extrude,'LineTo',int2str(x(2)), int2str(y(2)));
    invoke(extrude,'LineTo',int2str(x(3)), int2str(y(3)));
    invoke(extrude,'LineTo',int2str(x(1)), int2str(y(1)));
    invoke(extrude,'Create');
    release(extrude);

end

```

`units.m`: función para definir las unidades del proyecto.

```

function units(mws, geometry, frequency, varargin)

%Select de units nm and THz

    %geometry = 'nm';
    %frequency = 'THz';
    geom = geometry;
    fre = frequency;
    time = 'ns';
    temp = 'Kelvin';
    voltage = 'V';
    current = 'A';
    resistance = 'Ohm';
    conductance = 'S';
    capacitance = 'PikoF';
    inductance = 'NanoH';

    p = inputParser;

    p.addParameter('Geometry',geom);
    p.addParameter('Frequency',fre);
    p.addParameter('Time',time);
    p.addParameter('Temperature',temp);
    p.addParameter('Votlage',voltage);
    p.addParameter('Current',current);
    p.addParameter('Resistance',resistance);

```

```

p.addParameter('Conductance',conductance);
p.addParameter('Capacitance',capacitance);
p.addParameter('Inductance',inductance);

p.parse(varargin{:});
[geom,freq,time,temp] =
deal(p.Results.Geometry,p.Results.Frequency,p.Results.Time,p.Results.Temperature);

units = sprintf(['With Units\n',...
'.Geometry "%s"\n',...
'.Frequency "%s"\n',...
'.Time "%s"\n',...
'.TemperatureUnit "%s"\n',...
'.Voltage "V"\n',...
'.Current "A"\n',...
'.Resistance "Ohm"\n',...
'.Conductance "Siemens"\n',...
'.Capacitance "PikoF"\n',...
'.Inductance "NanoH"\n',...
'End With' ],geom,freq,time,temp);

invoke(mws,'Units', units);
invoke(mws,'AddToHistory',[ 'define units: ',geom],units);
invoke(mws,'AddToHistory',[ 'define units: ',freq],units);
fprintf("Units: geometry = "+geom+"\n");
fprintf("      frequency = "+freq+"\n");
end

```

`waveguideport.m`: función para un puerto tipo guía de onda.

```

function waveguideport(mws,number, Xrange, Yrange, Zrange, XrangeAdd, YrangeAdd,
ZrangeAdd,coordinates,orientation)

```

```

%Function to create a waveguide port

```

```

portw = sprintf(['With Port\n',...
'.Reset\n',...
'.PortNumber "%d"\n',...
'.Normal "Y"\n',...
'.Label ""\n',...
'.Folder ""\n',...
'.NumberOfModes "1"\n',...
'.AdjustPolarization "False"\n',...
'.PolarizationAngle "0.0"\n',...
'.ReferencePlaneDistance "0"\n',...
'.TextSize "50"\n',...
'.TextMaxLimit "1"\n',...
'.Coordinates "%s"\n',...
'.Orientation "%s"\n',...
'.PortOnBound "False"\n',...
'.ClipPickedPortToBound "False"\n',...
'.Xrange "%s", "%s"\n',...
'.Yrange "%s", "%s"\n',...
'.Zrange "%s", "%s"\n',...
'.XrangeAdd "0.0", "0.0"\n',...
'.YrangeAdd "0.0", "0.0"\n',...
'.ZrangeAdd "0.0", "0.0"\n',...
'.SingleEnded "False"\n',...
'.WaveguideMonitor "False"\n',...
'.Create\n',...
'End With' ],...
number,coordinates,orientation,int2str(Xrange(1)),int2str(Xrange(2)),int2str(Yrange(1)),
int2str(Yrange(2)),int2str(Zrange(1)),int2str(Zrange(2)));

invoke(mws,'Port',portw);
invoke(mws,'AddToHistory',[ 'define waveguide port: ',int2str(number)],portw);
end

```