

UNIVERSIDAD AUTÓNOMA DEL ESTADO DE
MORELOS

INSTITUTO DE INVESTIGACIÓN EN CIENCIAS BÁSICAS Y
APLICADAS

CENTRO DE INVESTIGACIÓN EN INGENIERÍA Y CIENCIAS
APLICADAS

**APLICACIÓN DEL MÉTODO FDTD PARA EL
ESTUDIO DE FENÓMENOS
ELECTROMAGNÉTICOS EN MATERIALES CON
NANOESTRUCTURA**

TESIS PROFESIONAL PARA OBTENER EL GRADO DE:
DOCTOR EN INGENIERÍA Y CIENCIAS APLICADAS CON
OPCIÓN TERMINAL EN: TECNOLOGÍA ELÉCTRICA

PRESENTA: MICA. GUSTAVO MEDINA ÁNGEL

DIRECTOR DE TESIS: DR. GENNADIY BURLAK

SINODALES: DRA. MARGARITA TECPOYOTL TORRES, DR.
J JESÚS ESCOBEDO ALATORRE, DR. JOSÉ ALBERTO
HERNÁNDEZ AGUILAR, DR. YURY KARLOVICH, DRA.
CECILIA CUEVAS ARTEAGA Y DR. JOSÉ ALFREDO
HERNÁNDEZ PÉREZ.

CUERNAVACA MOR.

AGOSTO 2020



INSTITUTO DE INVESTIGACIÓN EN CIENCIAS BÁSICAS Y APLICADAS
 Jefatura de Posgrado en Ingeniería y Ciencias Aplicadas



Cuernavaca, Morelos, a 07 de julio de 2020.

DR. ROSENBERG JAVIER ROMERO DOMÍNGUEZ
COORDINADOR DEL POSGRADO EN
INGENIERÍA Y CIENCIAS APLICADAS
P R E S E N T E

Atendiendo a la solicitud para emitir DICTAMEN sobre la revisión de la TESIS titulada: Aplicación del método FDTD para el estudio de fenómenos electromagnéticos en materiales con nanoestructura, que presenta el alumno **Gustavo Medina Angel**, para obtener el título de **Doctorado en Ingeniería y Ciencias Aplicadas**.

Nos permitimos informarle que nuestro voto es:

NOMBRE	DICTAMEN	FIRMA
DR. J JESÚS ESCOBEDO ALATORRE		
DR. JOSÉ ALFREDO HERNÁNDEZ PÉREZ		
DRA. CECILIA CUEVAS ARTEAGA		
DRA. MARGARITA TECPOYOTL TORRES		
DR. YURI KARLOVICH		
DR. JOSÉ ALBERTO HERNÁNDEZ AGUILAR		
DR. GENNADIY BURLAK		

PLAZO PARA LA REVISIÓN 20 DÍAS HÁBILES (A PARTIR DE LA FECHA DE RECEPCIÓN DEL DOCUMENTO)

NOTA. POR CUESTION DE REGLAMENTACIÓN LE SOLICITAMOS NO EXCEDER EL PLAZO SEÑALADO, DE LO CONTRARIO LE AGRADECEMOS SU ATENCIÓN Y NUESTRA INVITACIÓN SERÁ CANCELADA.



UNIVERSIDAD AUTÓNOMA DEL
ESTADO DE MORELOS

Se expide el presente documento firmado electrónicamente de conformidad con el ACUERDO GENERAL PARA LA CONTINUIDAD DEL FUNCIONAMIENTO DE LA UNIVERSIDAD AUTÓNOMA DEL ESTADO DE MORELOS DURANTE LA EMERGENCIA SANITARIA PROVOCADA POR EL VIRUS SARS-COV2 (COVID-19) emitido el 27 de abril del 2020.

El presente documento cuenta con la firma electrónica UAEM del funcionario universitario competente, amparada por un certificado vigente a la fecha de su elaboración y es válido de conformidad con los LINEAMIENTOS EN MATERIA DE FIRMA ELECTRÓNICA PARA LA UNIVERSIDAD AUTÓNOMA DE ESTADO DE MORELOS emitidos el 13 de noviembre del 2019 mediante circular No. 32.

Sello electrónico

JOSE ALBERTO HERNANDEZ AGUILAR | Fecha:2020-07-08 17:28:37 | Firmante
fXoqU9XDmpKLhZUBdhOY1klEmNi7SEX9ZwO2K3HDJgwKfO21T6YaWkkEaxNk2aMrfYkiw7HcplryqjK09sq7ZcJxlr7z9L9dK9Dh/azp7Lg9/dewV9dJ+rL5/zTAYZXCrTK2XHWf1fiazUyKAutLzK9OXQTYIglMqUwSQLzhAMEZkueq55B19+ItRE0ZVKsSXls8N1KRX+3GiWHe2DrAYSyZmXKJq8d9yPpeaDdFthf6x9s6WN/Hgc2joZPVlqmG3rZ7vMU3RYN HGR5tklQJouh7AUCM7tMMpJwNxdcpfMyK4GkeH3RNBo92uvvtq2M8FcTslNkUcXw==

MARGARITA TECPOYOTL TORRES | Fecha:2020-07-08 17:39:04 | Firmante
P/J8Jh8/ANHJTEA7yWgHqYHnykVypkb8Ue8N07yTIWnrAGod+AmlGvTo695m4JeCGz2t668pbXSURDCutVaXyoXT4XO/QgJm4JMTMMsDyLDJL7SrRXadJw/TqgkijtraB+/wahXSIXvFhmIHg9MKRpZ2eezmXIN9KcbWsy/9jnrTMsCDwg3AdiHdhdu28JJVzQuJ/0EbaEwNIUorSMbgrim6gBIHCMaMuS6gDLZpKSCx0xUwecabCeQczuZDZ4Isnp7bHBsD2pBqpDeEJ8WTF+BzmT0qN7qsZuScpvhouq2XbK4Odz8JqX/ULpYBeTOauek6QwtOdtur7w==

GENNADIY BURLAK | Fecha:2020-07-08 17:42:20 | Firmante
hwXMc9Zxozs/mKiceu4qfzxdZqQqkk8FXgvFpt6gvEjfu9VYHWXz6sDxSy/yEinUzpoi+vjgnB5mwpuHP2SEaPqx4OS1HskMU7T3twEXkqKfZ6HQsr5ppac2TUK6xOosVmr6Msf11KMIOxQOAL04d9VHQdPpWskxW7Jbflh48PlyrPFoWC+r0GjzPonpm38uoBouKJKxoyeDyxV2E/TbPI06VGdNE/iuRjBAbqYFU+Imq5fiEjHmDJAMgO7KXyFHAapUaO1vtm3TT1nR4ON2O95S/e9w10Wqv12pIT93t3MEKMY+fp+ddkdaadwjGY94RjXGjuwQ5DelpiA==

YURIY KARLOVYCH | Fecha:2020-07-08 17:43:23 | Firmante
mZUC5gnT94blv9vyVIN2u/apo6AsKTV2stN2zslITEYhYOV2dDqYQUJE5zDI/FrzvRAORAlxbY1BbDNoYUNdVw7WZ16dDdu3duXojVwSRgZnCDU5uzrnsDPAJzeEiWTKQJ03i+v30iCkjkNaB0GomuBGrwuyoy+2THXjsHRcJKAOOp+IfELa8MC5ayrh6rrf11HgIy/+39C4W2Rnd+filU5gJmQfkytNAdfNK4GYV11jSunN6Jav6a85c8CHTzqkcv7rjIwBx0/MXaGzGUC/W8T285QLE6rEWRBTBqtEuQQfUN2mCn2ZlaBkobUVV6i29cx1d2koloYkFwWuQ==

J JESUS ESCOBEDO ALATORRE | Fecha:2020-07-09 09:51:45 | Firmante
tQqyYX8Zwt4qoivxw7z/ws1ZWPcQWtXmzqb8HYYSBlqkmOj2APSSuc7PgsSngleXGM/nwlyCM/AY8mjhyvlyeLAq1KdBcRFbXuZcw625HQqGgRXE0Ny7/atm8+8rWF5Df6A7qyTBW10Ni35Llo41Su8SIWEfFxa1HN2jTbvQilGxyG+1A0sA8OAAyZv20+lcjh1QJrR2jNuifuUEyg7xKIVDU8Z3f2B9KH9Kiz1NRLIwNdVsbolkRQmFaNfIKoK4RLQWAA792DWBSTXgb42xJKUxC+wmGvLqAEK13v7lmzq4wZmlNAKWzqDOA/Gy1i6G1nxV1Cn3OnrCA==

JOSE ALFREDO HERNANDEZ PEREZ | Fecha:2020-07-09 21:43:00 | Firmante
JULw6yPdyDvFg0k9ppXLx2NtrR/PINK1UTvdkNPW/BMW585ilqHAXsTPgza3K9g9Gygf+uu8QgLnGAGpdWMqKC4I/IQTzBHKOp7IDH5s8DH5T/cabi6vB7hUN5+Uk0Re5+s90VoC6qY+hVXWzeCETQzINI/70oU1D6e3u4ZmPjFdqUxm87MGa0Th6bd+khUc27zolvv4occeb0+osMyYdZY12W41VIZ6EPVozA50sYv7dzKvq1+LA75FhrErBhrCke+8bxz/8ZT3hTEYV1G8ml2gPttRyAzpid6ue3UT8CgG9mOYPDSaAZMfagtklXbJX/QL9dpxFFt00UQ==

CECILIA CUEVAS ARTEAGA | Fecha:2020-07-10 13:06:18 | Firmante
gHRGzJNgRgh1e9KF+nwJVRn4PjVqG63RNjyPvk4690tzOeTckux9Gcin/Du48LqzJxbw9Gx7SDtJbuA9hRjgRveDgneUBWw0hsAR4P/oh1HsVsTR5VbhEG7LwcozE6/kjNFogkN2Ggp2uB8NoqHHMVuFahhKdlLlMUYXwl7KOxLUz8JKVixelDq/pDH7E19Ac9uGbhvwo4CwINTBz8ICUVqyPnhh9gJ4/J2AC8GaW1oWYwJlvPbvWuEwgIPlsRfWBlFkd5XZJsSaaluM3Zaj/kL6Nv+70A7xrfb4+zs3ERcUbk7wFsYgatj5xEbKaRP3fP25Swn5ISA==

Puede verificar la autenticidad del documento en la siguiente dirección electrónica o escaneando el código QR ingresando la siguiente clave:



PRKIAQ

<https://efirma.uaem.mx/noRepudio/DeNogophCZD4Q79BCilgDk0UYUnbXIZQ>



Una universidad de excelencia

RECTORÍA
2017-2023

**APLICACIÓN DEL MÉTODO FDTD PARA EL
ESTUDIO DE FENÓMENOS
ELECTROMAGNÉTICOS EN MATERIALES CON
NANOESTRUCTURA**

MICA. GUSTAVO MEDINA ÁNGEL

Resumen

En el presente trabajo estudiamos los fenómenos electromagnéticos que se generan en elementos con nanoestructura mediante técnicas de simulación computacionales, utilizando métodos matemáticos y métodos numéricos. Utilizamos el método FDTD (Diferencias Finitas en el Dominio del tiempo), técnica en la que podemos modelar estructuras y cuerpos con morfología sólida en tres-dimensional. Estudiamos el fenómeno de percolación que se produce al en un sistema 3D compuesto por nanotubos de carbono CNT, materiales semiconductores aplicados en diferentes ramas de la tecnología como conductores de corriente eléctrica. Mediante la técnica de optimización global en paralelo estimamos la probabilidad crítica de conducción en nuestro sistema conformado por CNT. En nuestro estudio encontramos la probabilidad crítica con parámetros establecidos para las longitudes y radios en los nanotubos aplicando una distribución normal o distribución Gaussiana, y una desviación estándar en la morfología de cada nanoestructura. Este enfoque nos permite estudiar los detalles de la conductividad eléctrica en nanocompuestos eléctricos, incluso al nivel de las fluctuaciones de filtración en materiales. Los CNTs además de tener propiedades de conducción eléctrica, se pueden utilizar en la rama de generación de plasma, y dicho fenómeno fue estudiado también en este trabajo. Utilizando el método FDTD incorporamos un arreglo periódico de nanotubos de carbono de pared simple SWCNT en una constante dieléctrica de dimensión 3D. Aplicamos una frecuencia de plasma ω_p a la nanoestructura de CNTs haciéndolas resonar a una alta frecuencia. Por encima de la nanoestructura pasa una partícula (Radiación de Cherenkov) que interactúa con el sistema CNT y como resultado de esta interacción se genera un campo electromagnético dentro de la estructura de nanotubos mientras que a la vez se generan plasmones-polaritones (Señales con frecuencias en THz) de superficie sobre la ruta principal donde pasa la carga. Estudiamos un rango considerable de ω_p , parámetro importante y dependiente en la generación de plasmones de superficie, en donde encontramos una frecuencia optima en el cual se genera una mayor proporción de plasmones de superficie y por lo tanto altos niveles de energía. En nuestro estudio consideramos nanotubos CNTs con frecuencias altas y nanotubos a base de TiO_2 (Dióxido de carbono) que oscilan a frecuencias menores a los CNTs, los resultados mostraron

una mayor ganancia de energía en las nanoestructuras CNTs sobre los nanotubos de TiO_2 cuando la partícula interactúa con ambas nanoestructuras resonantes. Consideramos también una nano estructura en forma de malla de CNTs, morfología muy similar a las nanoestructuras en forma de red creadas en la industria de la nanotecnología. Para este último caso de estudio obtuvimos un decaimiento de energía para un grupo de velocidades con la que viaja la carga mientras la frecuencia de plasma incrementa. Agregamos un trabajo adicional relacionado a nuestro tema de investigación, utilizando una red neuronal artificial el cual detecta la transición de fase en sistemas percolables con filtración en materiales con radios que tienen defectos en su uniformidad, en dicho sistema, medimos el nivel error que soporta nuestra red al identificar la percolación en sistemas 2D y 3D.

Abstract

In the present work we study the electromagnetic phenomena that are generated in elements with nano structure by means of computational simulation techniques, using mathematical methods and numerical methods. We use the FDTD method (Finite Differences in the Time Domain), a technique in which we can model structures and bodies with solid three-dimensional morphology. We study the percolation phenomenon that occurs in a 3D system composed of CNT carbon nanotubes semiconductor materials applied in different branches of technology as conductors of electric current. Using the global optimization technique in parallel, we estimate the critical probability of conductivity in our CNT system. In our study we found the critical probability with established parameters for the lengths and radii in the nanotubes, applying a normal distribution or Gaussian distribution, and a standard deviation in the morphology of each nanostructure. This approach allows us to study the details of electrical conductivity in nano electrical compounds, even at the level of material fluctuations. In addition to having electrical conduction properties, CNTs can be used in the plasma generation branch and this phenomenon was also studied in this work. Using the FDTD method we incorporate a periodic array of single-walled carbon nanotubes SWCNT into a dielectric constant of 3D dimension. We apply a ω_p plasma frequency to the nanostructure of CNTs by resonating them at a high frequency. A particle (Cherenkov radiation) that interacts with the CNT system passes over the nanostructure and as a result of this interaction an electromagnetic field its generate within the nanotube structure while at the same time plasmon-polaritons its generate (Signals with frequencies in THz) of surface on the main route where the load passes. We study a considerable range of ω_p , an important and dependent parameter in the generation of surface plasmons, where we find an optimal frequency in which a greater proportion of surface plasmons is generated and therefore high energy levels. In our study we consider CNTs nanotubes with high frequencies and nanotubes based on TiO_2 Carbon dioxide that oscillate at frequencies lower than CNT's, the results showed a greater in nanostructures CNTs energy gain over the TiO_2 nanotubes when the particle interacts with both nanostructures resonants. We also consider a mesh-like nano structure of CNTs, morphology very similar to

the network-shaped nano-structures created in the nanotechnology industry. For this last study we obtained an energy decay for a group of speeds with which the charge travels while the plasma frequencies increase. We added additional work related to our research topic, using an artificial neural network which detects the phase transition in percolable systems with filtration in materials with radii that have defects in their uniformity, in this system, we measure the error level that supports our network by identifying percolation in 2D and 3D systems.

Agradecimientos

Agradezco a mi director de tesis, el Dr. Gennadiy Burlak. Sus conocimientos, su manera de trabajar, paciencia, perseverancia y motivación han sido fundamentales para mi formación. Gracias por todo el apoyo recibido durante toda mi formación Doctoral.

Gracias al comité revisor de este trabajo: Dr. Gennadiy Burlak, Dra. Margarita Tecpoyotl Torres, Dr. J Jesús Escobedo Alatorre, Dr. José Alberto Hernández Aguilar, Dr. Yury Karlovich, Dra. Cecilia Cuevas Arteaga y al Dr. José Alfredo Hernández Pérez; por sus valiosos consejos y por dedicar el tiempo para mejorar este trabajo.

Agradezco al Centro de Investigación en Ingeniería y Ciencias Aplicadas de la Universidad Autónoma del Estado de Morelos, por haberme dedicado un espacio de formación y estudio y por el apoyo económico que obtuve a través de la beca del Doctorado, otorgada por el Consejo Nacional de Ciencia y Tecnología (CONACyT).

*Dedicado a
mi familia*

Organización de la tesis

Esta tesis se divide en 6 capítulos, 5 apéndices y 1 anexo, en el capítulo 1, se aborda la introducción general de la tesis, el planteamiento del problema, la justificación, las hipótesis y los objetivos generales y específicos. En el capítulo dos se aborda el estudio de la percolación mediante CNTs en nanoestructuras de carbono (CNT) para alcanzar la conducción eléctrica, está a su vez dividida en, en la introducción del capítulo, la generación de arreglos con nanoestructuras de radio y longitud fija, para posteriormente abordar nanoestructuras con longitud y radio aleatorio aplicando una distribución normal (Gauss), para esto se analizan las propiedades del objeto cilindro y sus intercepciones en el sistema para la localización del clúster y el clúster infinito que energiza el sistema de conducción por CNTs, se analiza la probabilidad crítica del sistema que se necesitó para alcanzar percolación en una rejilla y en múltiples rejillas con diferente probabilidad aprovechando una programación en paralelo, de los resultados obtenidos variando la longitud, el grosor, la desviación aplicada a los radios y la desviación aplicada a los grosores se analizan y se aplica el método Fit en Matlab, para el ajuste de los resultados. En el capítulo 3, abordamos la precisión finita de la función finita en el dominio del tiempo, las ecuaciones del método FDTD para una dimensión, segunda dimensión y tercera dimensión, así como el algoritmo de Ye para sus respectivas mallas dimensionales, también se exponen los cuerpos que son posibles modelar con el método FDTD, su respectivo código y el diagrama de clases básico de nuestro proyecto. En el capítulo 4 abordamos la radiación de Cherenkov sobre un sistema de nanotubos periódicos. En este capítulo se abordan el concepto de la radiación de Cherenkov de la partícula que viaja sobre un medio así como la superficie de plasmon-polariton sobre las nanoestructuras CNT conductoras para el caso de nanoestructuras infinitas teorizadas. A este fenómeno aplicamos la función del dominio del tiempo FDTD para el caso cuando la carga se mueve sobre el sistema periódico de nanotubos. Estudiamos un rango de frecuencias para localizar el punto que los nanotubos producen mayor energía en interacción con la carga, en este capítulo también se abordó el mismo fenómeno pero con nanotubos de morfología aleatoria y se procedió a buscar el punto con

mayor energía, mientras que otra variación para el mismo estudio con nanotubos de morfología aleatoria se comparó también con nanotubos de aleatorios de Dióxido de Titanio TiO_2 con oscilaciones de frecuencia bajas. Existen múltiples configuraciones que se pueden aplicar al estudio de las nanoestructuras y una de ellas es la configuración en malla aboradada también en el capítulo 4, en donde la carga pasa sobre la malla resonante conformado de nanotubos periódicos de manera horizontal formando intercepciones y nodos entre los nanotubos de carbono. En el capítulo 5 agregamos abordamos el estudio de filtración sobre materiales porosos, para identificar que sistemas son parcelables utilizando y entrenando una red neuronal supervisada que predice buenos resultados para mallas pequeñas. En el capítulo 6 se encuentran las conclusiones de nuestro estudio, los apéndices A y B se encuentra el código de las clases que se utilizaron en los capítulos 3 y 4, para el estudio de conductividad eléctrica y la radiación óptica de Cherenkov sobre CNTs. En el apéndice C se describen algunas generalidades del capítulo 4 de la radiación de Cherenkov, el apéndice D, describe el procedimiento experimental para la preparación de y síntesis de las películas y de los nanotubos de TiO_2 , en el apéndice E se describen algunas ecuaciones de gradiente correspondientes a la red neuronal supervisada mediante el método BackPropagation (Retropropagación), con sus respectivas funciones de error y funciones de pérdida. En el anexo A, se muestran algunos artículos publicados relacionados a nuestro proyecto de investigación, y por último se muestra la bibliografía pertinente a esta investigación.

Índice general

1. Antecedentes.	1
1.1. Introducción	1
1.2. Planteamiento del problema.	2
1.3. Justificación	2
1.4. Hipótesis	3
1.5. Objetivo General	4
1.6. Objetivos Específicos	4
2. Estudio de la conductividad con percolación en CNTs	5
2.1. Introducción a la conductividad eléctrica median CNT's	5
2.1.1. Array de Nanotubos	6
2.2. Arreglo de nanotubos con radio, altura y orientación fijos.	6
2.3. Nanotubos con longitud, radio y orientación aleatoria.	10
2.4. Propiedades del Objeto Cilindro.	10
2.5. Intercepciones entre los nanotubos.	12
2.6. Clúster de Percolación (Clúster infinito).	12
2.7. Distribución Normal-Gauss aplicado a tamaños y radios en CNTs	14
2.8. Probabilidad Crítica P_{cr}	15
2.9. Estructura de programación en paralelo	17
2.9.1. Parámetros de entrada	18
2.10. Resultados	19
2.11. Ajuste de los resultados	21
2.12. Resultados variando la longitud de los nanotubos	22
2.13. Resultados para varios desplazamientos σ_h	24
2.14. Resultados variando el radio de los nanotubos con desplazamiento σ_r	26
2.15. Resultados para varios grosores en los nanotubos μ_R	27
2.16. Conclusiones parciales	27

3. El método FDTD	29
3.1. Introducción	29
3.1.1. Precisión finita	29
3.2. FDTD en 1D	30
3.2.1. Algoritmo de Yee	30
3.2.2. Actualización de ecuaciones en 1D	31
3.2.3. Implementación de del algoritmo FDTD en una dimension.	35
3.3. Simulaciones del método FDTD en 2D (Simulaciones bidimensionales)	38
3.3.1. Arreglos bidimensionales.	38
3.4. Polarización TM^z en dos dimensiones -2D-	39
3.4.1. Ejemplo de TM^z	42
3.5. El límite TFSF para la polarización TM^z	42
3.5.1. Ejemplo de límite de TM^z TFSF	45
3.6. FDTD tridimensional -3D-	47
3.6.1. Arreglos tridimensionales	47
3.7. Ecuaciones gobernantes en una malla 3D.	48
3.7.1. Ejemplo de simulación de un dipolo 3D de una antena	51
3.8. Diagrama de la celda de Yee -3D	53
3.9. Cuerpos que se pueden simular con el método FDTD	54
3.10. Distribución espacial de formas 3D	55
3.11. Código para formas geométricas	57
3.12. Diagrama de clases	58
4. Radiación óptica de Cherenkov en CNTs	59
4.1. Introducción	59
4.2. Radiación de Cherenkov	60
4.3. La superficie de plasmón-polaritones en nanotubos conductores infinitos (caso analítico)	62
4.4. La excitación de campo en nanotubos periodicos reales por una carga móvil mediante el método FDTD	65
4.5. Acoplamiento óptico entre carga y la nanoestructura CNT-3D	68
4.6. Estructura 3D del campo óptico de la nanoestructura CNT	69
4.7. Radiacion de Cherenkov sobre una nanoestructura de nanotubos con morfología aleatoria	72
4.8. Aprovechamiento numérico - FDTD	74
4.9. Estudio de frecuencias de plasma ω_p y acoplamiento óptico sobre nanotubos con estructura aleatoria	75
4.10. Comparación de CNTs frente a Nanoestructuras de TiO_2	79
4.10.1. Orientaciones de las nanoestructuras	83
4.11. Radiación de Cherenkov sobre una malla de nanotubos resonantes	84
4.12. Rango de frecuencias ω_p para velocidades finitas	88

5. RNA detecta transición de fase mediante BackPropagation	92
5.1. Introducción	92
5.2. Ecuaciones básicas	93
5.3. Red neuronal supervisada	94
5.4. Expectativa vs Predicción	97
5.5. Conclusiones parciales	103
6. Conclusiones.	104
Apéndices	105
A. Código para la percolación con CNTs	106
A.1. Código One-Cylinder	106
A.2. Código WorkWithCylinders	107
A.3. Código de la clase WorkWithCylindersParal	117
B. Código de Radiación de Cherenkov en CNTs	121
B.0.1. Clase FdtdVars	121
B.1. Código para la Radiación de Cherenkov	126
C. Generalidades: Radiación de Cherenkov sobre CNTs	132
C.1. Introducción	132
C.2. Modelación de cuerpos Block+Rod	132
C.3. Fenómeno de la radiación de Cherenkov	133
C.4. Plasmón de superficie	138
C.5. Cálculo en paralelo.	139
C.5.1. Gráfica Energía v_{min} - v_{max} vs ω_p	141
D. Procedimiento Experimental para nanotubos de TiO_2	144
D.1. Síntesis de las películas nano-tubulares de TiO_2	144
D.2. Caracterización Eléctrica	145
D.3. Caracterización Física	145
D.4. Resultados	146
E. BackPropagation	149
E.1. Introducción	149
E.2. Derivación	150
E.3. Búsqueda de la derivación de error.	151
E.4. Función de pérdida	153
E.5. Suposiciones	153
E.6. Ejemplo de función de pérdida	153
Anexos	153
A. Artículos Publicados	154

Índice de figuras

2.1. Morfología típica de nanoestructura de Carbono CNT de pared simple a base de grafeno.	5
2.2. Llenado de un Arreglo 3D con Nanotubos CNT, donde el nivel de probabilidad P no puede ser mayor a 1 ni menor a 0.	6
2.3. Orientaciones del CNT a partir del centro de una determinada posición del Arreglo 3D.	7
2.4. La imagen (a) con una probabilidad $P=0.1$ en un tamaño de rejilla $L=30$, podemos observar las direcciones de los vectores hacia donde crecen los nanotubos, mientras que en la imagen (b) con $P= 0.3$ podemos observar un mayor número de nanotubos creados.	7
2.5. Clústeres de cilindros aglomerados identificados por cada color. La probabilidad de este sistema es $P = 0,5$, en un tamaño de rejilla $L = 50$	9
2.6. 3D que muestra un clúster infinito de percolación y conductividad eléctrica en donde con una Probabilidad $P = 0,5$ y una tamaño de rejilla de $L = 50$	9
2.7. Modelo 3D de nano-compuestos CNT's desordenados.	10
2.8. Esquema espacial del nanotubo 3D con sus parametros y variables correspondientes a las ecuaciones 2.7-2.9.	11
2.9. CNT's Con longitud aleatoria de 1 hasta 15 <i>nm</i> de longitud en una malla tridimensional de $L = 30$, y probabilidad $P = 0,1$	11
2.10. Esquema que muestra la intercepción entre dos cilindros con contacto perpendicular correspondientes a las ecuaciones 2.10-2.12	12
2.11. Clústeres de cilindros aglomerados identificados por cada color. La probabilidad de este sistema es $P = 0,5$, en un tamaño de rejilla $L = 50$	12

2.12. Rejilla 3D de tamaño $L=100$, donde fueron distribuidos Nanotubos (representados por cilindros) de tamaños variables. Podemos observar como se forma el clúster infinito de principio a fin en la rejilla simulada.	13
2.13. 11 Esquema de la distribución de Gauss utilizada en nuestro caso de estudio para generar medidas aleatorias en los nanotubos. . . .	14
2.14. Histogramas: (a) Distribución normal correspondiente a la altura $\mu_H=20$ y desviación de $\sigma_h=2$, con $hmin = 13$ y $hmax = 27,5$, (b) tenemos valores de aplicados a los radios de las nanoestructuras de $\mu_R=0.01$ y desviación de $\sigma_r=0.025$, con $rmax = 0,1$ y $rmin = 0,01$.	15
2.15. Distribución de probabilidades con diferentes saturaciones de nanoestructuras con valores $P = 0,01$ para el caso (a), para el caso (b) tenemos una $P = 0,07$, y el caso se observamos una probabilidad de $P = 0,65$	16
2.16. (a) Clúster con probabilidad $P = 0,02$ y (b) Clúster con $P = 0,25$ con longitud $\mu_H=20$ y radio de $\mu_R = 0,01$ para ambos casos. Con orientación de abajo hacia arriba.	16
2.17. Esquema de programación distribuida: (a), muestra el modelo de nanotubo presente en el sistema, (b), muestra un sistema 3D compuesto de nanotubos, (c) esquema de programación en serie para 10 probabilidades distribuidas, (d) imagen que muestra una programación en paralelo donde se distribuye un rango de probabilidades a la vez y cada tarea es independiente a las otras.	18
2.18. Distribución de un intervalo de probabilidades P , con un intervalo de $P_i = 0,1$ hasta $P_f = 0,5$, para diferentes $Task(Threads)$, (b) se muestran los resultados de ejecución del número de clústeres encontrados en el sistema 3D.	19
2.19. Gráfico que muestra el comportamiento de $P(p)$ volumen de los cilindros en la rejilla para 10 calculaciones, entre las probabilidades: 0.01 hasta 0.2	21
2.20. Gráfica con datos ajustado de la figura 2.19, en un rango de probabilidad $P=0.1$ a 0.2, con valores de: $L = 50$, $\mu_H=15$, $\sigma_h=1.4$, $\mu_R=0.1$ y $\sigma_r=0.025$. aplicando la ecuación: $P(p) = b(p - pc)\beta$ equivalente a la curva <i>Fit</i> : $p(p) = 1,82(p - 0,07)0,32$	21
2.21. Distribución de longitudes en las nanoestructuras; (a) $\Rightarrow 15\text{ nm}$, (b) $\Rightarrow 16\text{ nm}$, (c) $\Rightarrow 17\text{ nm}$ (i) $\Rightarrow 25\text{ nm}$, con incrementos de 1 nm ascendentemente para cada caso y con una desviación $\sigma_H = 1,4$	22
2.22. Ajuste de los datos aplicando $P(p) = b(p - pc)\beta$ para los casos de la figura 2.21(a - i)	23
2.23. Se muestra el comportamiento de P_{cr} , para varias pruebas en donde se varia la longitud del cilindro μ_H de 15 a 25 nanómetros, para una desviación $\sigma_H=1.4$	24

2.24. Se muestran los resultados graficados para los rangos de $\mu_H = 12$ a 25 nanómetros pero con 6 desviaciones diferentes: $\sigma_h = 0.01, 0.05, 0.4, 0.5, 0.6, 1, 1.2$ y 1.4 , con grosores de radio $\mu_R = 0.1$ y $\sigma_r = 0.025$ en una rejilla de dimensión $L = 50 \text{ nm}$ por cada lado del sistema cúbico 3D para las 8 curvas. 25

2.25. Comportamiento de los valores de P_{cr} , para un intervalo de desviaciones de $\sigma_r = 0.001$ hasta $\sigma_r = 0.015$ cuando: $\mu_H = 15, \sigma_h = 1.8$ y $\mu_R = 0.2$ 26

2.26. Comportamiento de los valores de P_{cr} , para un intervalo de desviaciones de $\sigma_r = 0.001$ hasta $\sigma_r = 0.015$ cuando: $\mu_H = 15, \sigma_h = 1.8$ y $\mu_R = 0.2$ 27

3.1. Disposición de los nodos de campo eléctrico y magnético en el espacio y el tiempo. Los nodos del campo eléctrico se muestran como círculos y los nodos del campo magnético como triángulos. El punto indicado es donde se expande la ecuación de diferencia para obtener una ecuación de actualización para H_y 32

3.2. Espacio-tiempo después de actualizar el campo magnético. La línea divisoria entre valores futuros y pasados ha avanzado un paso medio temporal. El punto indicado es donde se escribe la ecuación de diferencia para obtener una ecuación de actualización para Ez 33

3.3. FDTD unidimensional que muestra el desplazamiento espacial entre los campos magnético y eléctrico.. . . . 35

3.4. Espacio FDTD unidimensional que muestra la disposición espacial supuesta de los nodos de campo eléctrico y magnético en las matrices ez e hy . Existe un nodo de campo eléctrico a la izquierda del nodo de campo magnético con el mismo índice. 36

3.5. Gráfico que muestra el resultado del programa para 1D: Bare-bones, utilizando el método FDTD 37

3.6. Salida generada por el programa anterior pero con maxTime establecido en 1000. 37

3.7. Disposición espacial de nodos de campo eléctrico y magnético para polarización TM^z . Los nodos de campo eléctrico se muestran como círculos y los nodos de campo magnético como cuadrados con una línea que indica la orientación del componente de campo. 41

3.8. Disposición espacial de nodos de campo eléctrico y magnético para polarización TM^z . Los nodos de campo eléctrico se muestran como círculos y los nodos de campo magnético como cuadrados con una línea que indica la orientación del componente de campo. 43

3.9. Representación de un límite de campo total / campo disperso en una cuadrícula TMz. El tamaño de la región TF está definido por los índices del primer y último nodo de campo eléctrico que están dentro de la región. En el lado derecho del límite, los nodos H_y con el mismo índice x como el "último" nodo estará en la región SF. De manera similar, en la parte superior de la cuadrícula, los nodos H_x con el mismo índice y que el último nodo estarán en la región SF. Por lo tanto, se debe prestar atención al componente de campo así como a los índices para determinar si un nodo está en la región SF o TF. 44

3.10. Se utiliza una cuadrícula auxiliar unidimensional para calcular el campo incidente que se supone que se propaga en la dirección $+x$. Las flechas verticales indican los nodos cuyos valores son necesarios para implementar el límite TFSF. El campo H_x incidente es cero y, por lo tanto, no se necesita corrección en asociación con los nodos E_z que tienen un nodo H_x vecino en el otro lado del límite. . . . 45

3.11. Visualización del campo Ez en un dominio computacional que emplea un límite TFSF. Las instantáneas se toman en los pasos de tiempo (a) 30, (b) 100 y (c) 170. La fuente de onda plana pulsada corresponde a una ondícula Ricker con 30 puntos por longitud de onda en su frecuencia más energética. 46

3.12. Visualización del campo Ez en un dominio computacional que emplea un límite TFSF. Hay una placa vertical PEC que se realiza al poner a cero el campo Ez sobre una línea que tiene 41 celdas de alto y 20 celdas desde el borde izquierdo del dominio computacional. Las instantáneas se toman en los pasos de tiempo (a) 30, (b) 100 y (c) 170. Se utiliza un ABC de segundo orden para terminar la cuadrícula. 46

3.13. Representación de elementos de una matriz con dimensiones $3 \times 4 \times 3$ en las direcciones x , y y z , respectivamente. Los índices m , n y p se utilizan para especificar las preocupaciones x , y y z , respectivamente. El elemento en el origen tiene índices $(0, 0, 0)$ y se muestra en la esquina superior izquierda del plano inferior. 47

3.14. Los nodos de campo eléctrico se desplazan medio paso en la dirección en la que apuntan, mientras que los nodos de campo magnético se desplazan medio paso en las dos direcciones que no apuntan. También se entiende implícitamente que los nodos de campo eléctrico y magnético se compensan entre sí medio paso en el tiempo. 49

3.15. Los nodos en una cuadrícula FDTD 3D a menudo se dibujan en forma de un cubo o celda Yee . En esta representación, no todos los nodos tienen los mismos índices. El cubo consistiría en cuatro nodos E_x , cuatro nodos E_y y cuatro nodos E_z , es decir, los campos eléctricos están a lo largo de los bordes del cubo. Los campos magnéticos están en las caras del cubo y , por lo tanto, habría dos nodos H_x , dos nodos $H - y$ y dos nodos H_z	49
3.16. Caras de un dominio computacional que es $5 \times 9 \times 7$ en las direcciones x, y y z , respectivamente. En la cara constante x , los campos tangenciales son E_y y E_z , en la cara constante y son E_x y E_z , y en la cara constante z son E_x y E_y . También hay nodos de campo magnético que existen en estas caras, pero su orientación es normal a la cara.	50
3.17. Muestra la propagación de la E_z desde el dipolo en el nivel del plano XY con el espacio en el dipolo. Por supuesto, hay radiación en la dirección Z a medida que avanzamos. Esto ilustra un problema importante en las simulaciones tridimensionales: a menos que uno tenga gráficos inusualmente buenos, visualizar tres dimensiones será difícil [34].	51
3.18. Diferentes momentos de onda omnidireccional de una antena radiante de una antena transmisora. La amplitud de la onda está dada en los colores de la barra <i>colorBar</i> . Del campo E_z de fuerte a débil (rojo-azul) [35].	52
3.19. Superficie 3D de la propagación de onda electromagnética E_z emitida por el dipolo de una antena transmisora. Los círculos indican la antena emisora donde se encuentra el receptor de interés la intensidad roja indica mayor potencia y donde se irradia la mayor propagación de onda direccional. [35].	52
3.20. En esta figura se muestran los cuadrantes en donde podemos implementar los cuerpos, su comportamiento y movimiento en el espacio y tiempo.	54
3.21. Esquema que muestra todas las figuras que se pueden simular con el método FDTD.	55
3.22. Representación espacial de bloques (1)-(5) formando una estructura compuesta.	56
3.23. Diagrama de clases principales que indican la relación cuando se realiza el llamado a una subrutina e invocación de otras clases así como la relación de objetos y métodos.	58
4.1. Onda cónica que produce la radiación de Cherenkov.	61
4.2. Esquema parcial del cilindro infinito desmagnetizado que puede producir plasma.	62

4.3. Dispersión para la de solución a la ecuación (4.7), donde $x = \omega a/c$, $y = ka$ con parámetros diferentes en $q = \omega a/c$, $k = \omega/c$ para (a) $p = 0.2$ y (b) $p=0.9$ 63

4.4. Estructura radial del campo E_z (para la ecuación (4.3) para $q = \omega/a$ $c = 0,1$, $kr = 9$, y diferentes valores $p < a \leq b = 0.2; 0.8; 0.9$, y 0.95 64

4.5. Arreglo de nanotubos CNT, resonando a una frecuencia ω_p y una carga que produce el efecto de la radiación de Cherenkov. 66

4.6. Radiación de Cherenkov en la superficie, con frecuencia de plasma (a) $\omega_p = 10^{12}s^{-1}$ y(b) $\omega_p = 2,1 \times 10^{15}s^{-1}$. Observamos que las amplitudes de los campos excitados en cada SWCNT dependen significativamente del valor ω_p . Para $\omega_p = 10^{12}s^{-1}$ bajo, el campo se excita principalmente dentro del área de la ruta de carga, panel (a), pero para $\omega_p = 2,1 \times 10^{15}s^{-1}$ mayor, (b) los SWCNT excitados se encuentran en el frente de onda total de Cherenkov. Otros parámetros son: el radio de los nanotubos es $R = 175nm$, el radio interno es $R_i = 140nm$, la longitud de los nanotubos es $h = 3,10^3nm$ y el tamaño del sistema es $5 \times 104nm$ 67

4.7. La energía radiada de la carga (unidades arbitrarias), integrada a lo largo de la ruta de la dirección de la carga en función de ω_p para diferente velocidad de carga v . En el área $\omega_p = 1 \times 10^{15}, 4 \times 10^{15}]s^{-1}$ observamos un patrón de energía de doble pozo con un máximo (indicado por el símbolo B) en $\omega_p = 2,1 \times 10^{15}s^{-1}$ y dos mínimos (indicados por los símbolos A, C) en $\omega_p = 1,2 \times 10^{15}s^{-1}$ y $\omega_p = 3,5 \times 10^{15}s^{-1}$ respectivamente para todas las velocidades. La posición de dicho extremo depende débilmente del valor de v_x . El pico corresponde a un fuerte régimen de acoplamiento entre el campo de carga radiante y el campo excitado en la estructura CNT. 68

4.8. La estructura del componente de campo generado E_x (unidades arbitrarias) en un sistema 3D que contiene carga radiante (se mueve con velocidad $v = 0,98$) y la matriz de CNT. La estructura del campo $E_x(x, y, z_i)$ se muestra en varias capas ($i - th$ cortes) del eje Z (de (a) a (i)) del sistema 3D para $\omega_p = 2,1 \times 10^{15}s^{-1}$ en la posición del pico del campo, ver figura 4.7, flecha B . El panel (a) muestra las amplitudes de campo en el área superior $i = 2$ de la carga móvil (ver figura 4.3), el panel (i) muestra el campo en el área inferior $i = 100$. La carga se mueve de izquierda a derecha. En los paneles (d) - (g) observamos la estructura del frente de onda de Cherenkov irradiado por la carga. Los puntos de color correspondientes al campo de los nanotubos conductores se ven en los paneles(c) - (h). Los valores de los campos se indican por las barras de colores. En los paneles (d) - (h) observamos el campo de interconexión de gran amplitud entre la CNT. 70

4.9. Se muestran los mismos cortes que en la figura 6, pero para (a) $\omega_p = 1,2 \times 10^{15} s^{-1}$ en la posición del campo izquierdo mínimo, y para (b) $\omega_p = 3,5 \times 10^{15} s^{-1}$ en la posición de mínimo del campo derecho, ver las flechas A y C de la figura 4.7 de los valores mínimos de los posos respectivamente. 70

4.10. La imagen (a) muestra la distribución de campo 3D en E_x con la carga móvil (lado derecho) y los nanotubos (lado izquierdo) para la frecuencia de plasma $\omega_p = 2,1 \times 10^{15} s^{-1}$ que corresponde al pico de la figura 4.7 pico con flecha "B". La flecha azul muestra la dirección de entrada de la carga. La flecha P y Q inician el área del campo Cherenkov y el campo CNT respectivamente. En el centro (flecha Q) observamos la estructura periódica del campo 3D excitada en el área de los nanotubos. Se puede observar que en $\omega_p = 2,1 \times 10^{15} s^{-1}$ el campo de la estructura CNT tiene una forma periódica bien definida. La imagen (b), muestra la estructura de los nanotubos resonantes en el interior de la nanoestructura y bajo la superficie de donde pasa la carga. 71

4.11. Patrón de CNTs paralelos con tamaños aleatorios (longitud y radio) en la parte inferior del sistema. La carga rápida se mueve paralela al plano XY con velocidad uniforme $v \parallel X$ sobre el nanotubo como se indica con la flecha Q; (b) el recuadro muestra a detalle la morfología de un SWCNT (Nanotubo de pared blando). [69]. . . 72

4.12. Muestra la solución a la ecuación 4.9 de dispersion con los parámetros de dependencia. La flecha $A_{1,6}$ y $B_{1,6}$ muestra la solución cuando $l=0$ y $l=1$. La flecha C muestra la frecuencia de la superficie del plasmon la flecha D corresponde a la onda de la luz $k = \omega/c$. La línea punteada con la flecha E se aproxima al espectro de densidad de energía radiada por la carga. 73

4.13. El campo Cherenkov E_x (unidades arbitrarias) en el plano central generado por una carga que se mueve con velocidad uniforme $v = 0,866$ en el caso de SWCNT con frecuencia de plasma (a) $\omega_p = 6 \times 10^{14} s^{-1}$ y (b) $\omega_p = 4,1 \times 10^{14} s^{-1}$. Observamos que las amplitudes de los campos excitados en cada SWCNT dependen significativamente del valor ω_p . Para $\omega_p = 6 \times 10^{14} s^{-1}$ el campo se excita principalmente dentro de la ruta de carga, pero para $\omega_p = 4,1 \times 10^{14} s^{-1}$ (b), hay pequeños huecos cóncavos y picos de grupos de campos SWCNT excitados en la onda Cherenkov. . . . 75

<p>4.14. La energía radiada integrada (unidades arbitrarias) en toda ruta de la dirección que sigue la carga (path) en función de ω_p a diferentes velocidades de carga v. En el área $\omega_p = 1 \times 10^{14}, 5 \times 10^{16}]s^{-1}$ observamos un patrón de energía con el máximo nivel de energía (indicado por el símbolo B) en $\omega_p = 4,1 \times 10^{15}s^{-1}$ y un mínimo de energía inicial A en $\omega_p = 6 \times 10^{14}s^{-1}$ respectivamente para todas las velocidades. La posición de dicho extremo depende débilmente del valor de v_x. El pico corresponde a un fuerte régimen de acoplamiento entre el campo de carga radiante y el campo excitado en la estructura CNT.</p> <p>4.15. Estructura del componente del campo generado E_x (unidades arbitrarias) en un sistema 3D que contiene carga radiante (moviéndose con velocidad $v = 0,98$) y la matriz de CNT. La estructura del campo $E_x(x, y, z_i)$ se muestra en varias capas (i-th cortes) del eje Z (de (a) a (i)) para $\omega_p = 6 \times 10^{14}s^{-1}$ en la posición del campo mínimo, ver figura 4.14, flecha A. El panel (a) muestra las amplitudes de campo en el área superior $i = 2$ de la carga móvil (ver figura 4.14), el panel (i) muestra el campo en el área inferior $i = 100$. La carga se mueve de izquierda a derecha. En los paneles (d) - (g) observamos la estructura del frente de onda de Cherenkov irradiada por la carga. Los puntos de color correspondientes al campo de los nanotubos conductores se visualizan en todos los planos (c) - (h). Los valores de los campos están indicados por las barras de colores. En los paneles (d)-(h) observamos el campo de interconexión de gran amplitud entre la CNT.</p> <p>4.16. Distribución de campo 3D E_x(a) sistema compuesto con la carga móvil y nanotubos para la frecuencia de plasma $\omega_p = 6 \times 10^{14}s^{-1}$ que corresponde al valor mínimo de la figura 4.14 (flecha A). La flecha azul muestra la dirección de entrada de la carga. Las flechas P y Q indican el área del campo Cherenkov y el campo CNT respectivamente. En el centro (flecha Q) observamos la estructura periódica del campo 3D excitada en el área de los nanotubos. Se puede ver que en tal $\omega_p = 6 \times 10^{14}s^{-1}$ el campo de la estructura CNT tiene una forma periódica bien definida. Par el caso (b) observamos las mismas caracterizas pero correspondientes al punto B al valor máximo de la figura 4.14 (flecha B), punto donde se emite mayor energía de plasmones-polaritones dentro de la nanoestructura con frecuencia $\omega_p = 4,1 \times 10^{15}s^{-1}$.</p> <p>4.17. Imagen real tomada con Micrografías SEM (Scanning electron microscope) que muestran superficies de fractura de compuestos de CNT / epoxy alineados después de la prueba de tracción [91] . . .</p> <p>4.18. Nanoestructuras de dióxido de Titanio (TiO_2) imagen de microscopio de barrido. Muestra el costado de los nanotubos de TiO_2, con forma de bambu bien definida.</p>	<p>76</p> <p>77</p> <p>78</p> <p>79</p> <p>79</p>
--	---

4.19. Energía que irradiada la carga con dependencia de la velocidad de la carga para diferentes frecuencias de plasma; $\omega_p = [10^{13}, 10^{14}, 10^{15}, 2,1 \times 10^{15}, 3,5 \times 10^{15} \text{ y } 8,5 \times 10^{15}]$ para los nanotubos de CNT's (a) y $\omega_p = [10^3, 10^5, 10^8 \text{ y } 10^{10}]s^{-1}$ para el caso (b) TiO_2 , en un intervalo de velocidades $v=0.5$ hasta $v=0.98$. Dependencia de velocidad v para ambos nanoestructuras.	80
4.20. Estructura interna de los nanotubos desde la superficie hasta la base etiquetados desde $i = [2, 21, 31, 41, 51, 61, 71, 81, 100]$, en las que podemos observar el campo óptico que generan la radiación de la partícula para cada sección transversal.	81
4.21. Estructura Óptica 3D con $\omega_p = 10^{10}s^{-1}$, cuando la carga que produce la radiación de Cherenkov va a una velocidad $v=0.85$. Punto máximo indicado por la flecha B de la figura 4.19(b).	81
4.22. Podemos observar en primera instancia que para las frecuencias ω_p para los nanotubos de TiO_2 , existe una alta concentración de energía constante para todo el rango de familias de $\omega_p = 10^{10}s^{-1}$ cuando la velocidad es igual a $v = 0,81$ (Línea de color verde). Caso contrario para los CNTs donde existe un cambio considerablemente importante de energía cuando la carga va a una velocidad $v=0.98$ (Línea de color azul) en un pico máximo de $\omega_p = 1,2 \times 10^{16}s^{-1}$ indicada por la flecha A	82
4.23. Tomografías en forma de paneles (a-i) que muestran los cortes de altura del campo E_x en altura y paralelo al eje Z para el pico A de la figura 4.22, podemos ver que la tomografía, con más intensidad es el panel (i=61 correspondiente al panel (f)).	82
4.24. Cono de Cherenkov producido por la carga sobre los nanotubos con velocidad $v=0.866$ y frecuencia $\omega_p = 1,2 \times 10^{16}s^{-1}$. (Pico máximo de radiación de energía(indicada por la flecha A de la figura 4.22)).	83
4.25. Amplitud del campo E_x en el interior de la estructura CNT con nanotubos aleatorios, con frecuencia: $\omega_p = 1,2 \times 10^{16}s^{-1}$ (pico máximo de radiación de energía(indicada por la flecha A de la figura 4.22)).	83
4.26. Microscopía electrónica que visualiza las fibras de CNT con diámetros de 100-200 nm. [84]	83
4.27. (a) Sistema de simulación 3D de nanotubos + carga. Donde los nanotubos muestran la morfología con forma de red (plano XY) y la carga rápida (flecha Q) que se mueve por encima de la red de CNT. (b) muestra la morfología del SWCNT.	84
4.28. Podemos observar el campo óptico (Mediante el método FDTD [98]) que produce la carga en la superficie de la nanoestructura (Red de CNTs). Las flechas indican la entrada y salida de la carga. Para el caso (a), tenemos una frecuencia de $\omega_p = 1 \times 10^{13}$ y para el caso $\omega_p = 5 \times 10^{15}$ con una velocidad de carga para ambos casos de $v = 0,75$	85

4.29. Tomografías de dos paneles en profundidad del campo E_x 3D en pequeños cortes que van desde $i = 2$ (capa cerca de la superficie de la nanoestructura) hasta $i = 100$ (capa más profunda del sistema 3D). Para el caso (a) las capas de tomografía corresponden a una frecuencia de $\omega_p = 1 \times 10^{13}$, para el caso (b) las capas corresponden a una frecuencia $\omega_p = 5 \times 10^{15}$, ambos casos con una velocidad $v = 0,75$ 86

4.30. Campo óptico E_x que produce la malla de CNT dentro del sistema 3D. Para el caso (a), las flechas amarillas corresponden a la entrada y salida de la carga con velocidad $v = 0,75$ en la nanoestructura que resuena a una frecuencia de $\omega_p = 1 \times 10^{13} s^{-1}$. Las flechas P y Q indican el área del campo Cherenkov y del campo CNT, respectivamente. El caso (b) corresponde a un corte en el medio del sistema (a) donde observamos en detalle la forma del campo de los nanotubos. Lo mismo se aplica en los casos (c) y (d) pero con una frecuencia de $\omega_p = 5 \times 10^{15} s^{-1}$. Observamos que el campo visible interno que producen los nanotubos dentro del sistema 3D queda descubierto en las imágenes (b) y (d) y el campo resonante de los CNTs están orientadas en dirección XY formando una malla (paralelas a la dirección de la carga) extendida a lo largo del sistema. En la barra de colores E_x observamos la intensidad del campo eléctrico alcanzando como consecuencia entre la interacción de la carga y el nano-sistema. Las flechas amarillas indican la dirección de donde entra y sale la carga. Observamos claramente una fuerza de acoplamiento e intensidad del campo diferente entre las frecuencias $\omega_p = 1 \times 10^{13} s^{-1}$ y $\omega_p = 5 \times 10^{15} s^{-1}$ 87

4.31. Investigamos un rango de frecuencia de $\omega_p = [1 \times 10^{13}, 1 \times 10^{16}]$ para un amplio rango de velocidades de carga ($v = 0.06, 0.12, 0.17, 0.23, 0.29, 0.35, 0.40, 0.46, 0.52, 0.58, 0.64, 0.69, 0.75, 0.81, 0.87, 0.92, 0.98$). Podemos observar que para la velocidad $v = 0.75$ hay un aumento significativo en la energía W_c que disminuye mientras aumenta la frecuencia ω_p en los nanotubos. 88

4.32. Encontramos una familia de velocidades $v = [0,706, 0,791]$ que generan diferentes picos de energía W_c a ciertas frecuencias, que comienzan en el punto $v = 0,706$ y $\omega_p = 3,4 \times 10^{15}$ y termina en el punto $v = 0,791$ con $\omega_p = 4 \times 10^{14}$, siendo este, el último pico de energía encontrado en nuestra investigación. También encontramos un pico máximo de energía indicado con la flecha M con una velocidad $v = 0,779$ y una frecuencia $\omega_p = 1 \times 10^{13}$ siendo el pico más alto de toda la familia investigada. 89

4.33. La imagen (a) muestra el campo que produce la carga en la superficie de la nanoestructura de red de nanotubos, (b) tenemos un panel de tomografías en profundidad del campo E_x 3D en cortes pequeños, la capa con valor $i = 2$, es la capa más cercana a la superficie, mientras que la capa con el valor $i = 100$ es la capa más profunda del sistema. (c) Observamos el campo óptico 3D interno que produce la carga en el sistema, la imagen (d) muestra un corte a la mitad del mismo sistema. Las flechas amarillas corresponden a la entrada y salida de la carga. Las flechas P y Q indican el área del campo Cherenkov y el campo CNT. Las imágenes (a-d) corresponden al pico de energía máxima (flecha M) de la figura 4.32 con velocidad $v = 0,779$ y frecuencia $\omega_p = 1 \times 10^{13}$ 90

5.1. El parámetro de orden $P(p)$ (línea verde) en función de la probabilidad del número de poros p para el sistema de filtración $2D$ $L \times L$, $L = 30$ designada para la filtración extendida con poros aleatorios. (La línea azul muestra $P(p)$ para la filtración estándar con poros fijos). Los recuadros muestran la distribución espacial de los poros con radio aleatorio r en varios valores p : (a) y (b) por debajo de la probabilidad crítica $p < p_c$ (en este caso, el grupo infinito no existe), (c) $p \simeq p_c$ (solo aparece un grupo incipiente de filtración), y (d) $p > p_c$ (se encuentran los grupos masivos). Debido a la aleatoriedad del radio de poros r , el valor crítico p_c difiere del conocido valor estándar $p_c = 0,59$. Los clusteres predominantes de percolación acoplados se muestran en diferentes colores predominantes sobre los recuadros (a-d) de la figura. 94

5.2. La estructura espacial de los poros aleatorios en $p \simeq p_c$. En nuestro estudio utilizamos dos distribuciones diferentes de poros de tamaño aleatorio con parámetros $\langle r \rangle = 0,706$, $\sigma_r = 0,12$: (a) distribuciones uniformes, y (b) distribución normal (Gauss). Los paneles (c) y (d) muestran los histogramas para poros aleatorios obtenidos en nuestras respectivas simulaciones. 95

5.3. La matriz $2D$ $L \times L$ (con estructura espacial de los clústeres) se reforma en el vector $1D$ con una longitud L^2 que representa una fila del conjunto de datos (clase de entidad) (Ver figura. 5.4). Dichos vectores se utilizan como entrada para el entrenamiento de la red neuronal supervisada. Los últimos elementos de cada vector $1D$ (etiqueta de la última columna) tienen valores 0 o 1 dependiendo de la existencia de la transición de fase de filtración en el sistema para la probabilidad dada (Ver figura 5.1). 96

5.4.	El conjunto de datos inicial consta de un vector $1D$ con filas de datos convertidos a partir de matrices $2D$ (estructura espacial del material de filtración), la etiqueta (última columna 0 o 1) calculada por cada p indicado en la derecha lado indica un valor booleano para indicar si existe o no percolación en el sistema de filtración. Dicho conjunto de datos se utiliza para alimentar y capacitar a la red supervisada para detectar la transición de filtración en el sistema extendido $2D$	97
5.5.	Pruebas de expectativa vs predicción, en la imagen (a), tenemos un ruido de 0.1 aplicado a los radios de los poros, observando que la red neuronal soporta este nivel de ruido, en el caso de la imagen (b) tenemos un nivel de ruido de 0.5, observamos que la red neuronal comienza a fallar en exactitud en sus predicciones. Donde n es el número de prueba o iteración.	98
5.6.	Intervalo de amplitud de ruido para probar la eficiencia de la red neuronal al reconocer una transición de fase. Observamos que esta red soporta ruidos menores o iguales a 0.3, después de la eficiencia decrece si el ruido aumenta. La línea azul indica el límite de la eficiencia de la red. La etiqueta <i>accuracy</i> indica el nivel de precisión de la red y $L=30$ es el tamaño de la matriz $2D$ de percolación en la cual se aplicaron las pruebas.	99
5.7.	Intervalo de amplitud de ruido para probar la eficiencia de la red neuronal al reconocer una transición de fase con tamaños de rejilla $L=30, 32, 34, 36, 38$ y 40 . La etiqueta <i>accuracy</i> indica el nivel de precisión de la red para una distribución Uniforme.	100
5.8.	Intervalo de amplitud de ruido para probar la eficiencia de la red neuronal al reconocer una transición de fase con tamaños de rejilla $L=30, 32, 34, 36, 38$ y 40 . La etiqueta <i>accuracy</i> indica el nivel de precisión de la red para una distribución Gauss.	100
5.9.	Precisión de la predicción de la red neuronal de la transición de fase de filtración (sistema $2D L \times L$ con diferentes $L = 30 \div 40$) en función de la amplitud de ruido a . En los casos con radio de poros distribuido uniformemente (a) y normal (b). Observamos que hasta una modulación de ruido moderada a $a < 0,3$ la precisión de la predicción es de aproximadamente 100%. Pero para ruidos mayores $a > 0,3$ la precisión disminuye.	102
5.10.	Distribución espacial 3D con poros de radio aleatorio r en el sistema de filtración 3D ($L \times L \times L$; $L = 11$) ,la probabilidad crítica se encuentra cerca de $p \simeq p_c = 0,32$. Para este caso 3D se visualiza el clúster infinito filtración (color verde). Los grupos no acoplados se muestran con diferentes colores.	102

5.11. Observamos lo mismo que en la figura 5.9 (a) (con distribución uniforme) pero para el caso del sistema 3D con $L \times L \times L$ para $L = 10, 11$. Observamos nuevamente que a una amplitud de ruido $a < 0,3$ la precisión de predicción es de aproximadamente 100 % como en el sistema 2D, ver figura 5.9 (a). Pero para una amplitud mayor $a > 0,3$ la precisión de la predicción disminuye.	103
C.1. Cuerpos (a) Blocks que sirven para la base esquemática de nuestra simulación, (b) cilindros (rods) para representar CNTs y (c), la combinación de a+b.	133
C.2. Partícula que atraviesa un arreglo de nanotubos y se muestran los campos eléctricos (E) y magnéticos (H) estáticos (En un momento en el tiempo) con el paso de la partícula	133
C.3. Partícula que atraviesa un arreglo de nanotubos. El campo eléctrico o picos de la imagen, son el resultado del efecto que causa la partícula que deforma los campos eléctricos de los nanotubos (caso periódico), que se encuentran por debajo de la superficie por donde pasa la partícula este efecto es llamado Radiación de Cherenkov.	135
C.4. Se muestran los campos eléctricos para la propagación vectorial (a) E_x , (b) E_y , (c) E_z , sobre la superficie nano-tubular.	135
C.5. Se muestran los campos magnéticos para la propagación vectorial (a) H_x , (b) H_y , (c) H_z , sobre la superficie nano-tubular.	135
C.6. Ruta del campo eléctrico generado por una carga radiante, observamos picos de campos generados que corresponden a los nanotubos que hay debajo de la superficie. La orientación de la carga es de izquierda a derecha.	136
C.7. Geometría 2D de la ruta de la carga, de la figura C.6. Al sumar los picos máximos que produce la carga durante toda su trayectoria obtendremos un punto total de energía máxima para una determinada frecuencia.	136
C.8. Transformada de Fourier para el comportamiento de la ruta de energía que se muestra en la figura C.7, en donde se muestra la amplitud amp por frecuencia f de onda registrada.. . . .	137
C.9. Se muestra el vector de longitud <i>Wavevector</i> k_x/K_p vs la frecuencia, respecto a la frecuencia de plasma <i>Frequency</i> ω/ω_p que corresponden a la amplitud de la figura C.8, donde la línea naranja corresponde a la frecuencia guiada por la luz.	137
C.10. Gráfica de una partícula sobre un metal, entre más velocidad alcance la partícula, mas plasmon se generara.	138
C.11. Para el valor del plasmon siempre será menor en el metal, que en el aire.	139
C.12. Figura que muestra, la simulación, la calculación y resultados que se pretenden realizar, se muestran "n" partículas sobre una superficie metálica S	139

C.13. Se muestran los procesos ejecutándose de manera paralela para 20 tareas simultáneamente y el porcentaje de avance para cada proceso	140
C.14. Gráfica que muestra el valor de la calculación para la energía En .	140
C.15. Gráfica que muestra el valor de la calculación para el componente P_x	140
C.16. Gráfica que muestra el valor de la calculación para el componente del vector de flujo P_y .	141
C.17. Gráfica que muestra el valor de la calculación para el componente de vector de flujo P_z	141
C.18. Gráfico que muestra los resultados para las frecuencias $\omega_p = 1e12$, $\omega_p = 1e13$, $\omega_p = 1e14$, $\omega_p = 1e15$ y $\omega_p = 1e16$, en un rango de velocidades que van desde $v_{min} = 1,3$ hasta $v_{max} = 1,7$ ($v_{min} = 0,81$ a $v_{max} = 0,98$), para el cálculo de la energía (Energy).	142
C.19. Gráfico que muestra los resultados para las frecuencias $\omega_p = 1e12$, $\omega_p = 1e13$, $\omega_p = 1e14$, $\omega_p = 1e15$ y $\omega_p = 1e16$, en un rango de velocidades que van desde $v_{min} = 1,3$ hasta $v_{max} = 1,7$ ($v_{min} = 0,81$ a $v_{max} = 0,98$), para el cálculo de la energía (Energy) en vista 3D.	142
C.20. Gráfico del vector de flujo Poynting en su propagación P_x , para los mismo valores que la gráfica C.18	143
C.21. Gráfico del vector de flujo Poynting en su propagación P_x , para los mismo valores que la gráfica C.19 en su vista 3D.	143
C.22. Gráfico de la energía para los mismo valores que la gráfica C.18 ω_p pero para nanoestructuras con morfología aleatoria.	143
C.23. Gráfico de la energía para los mismo valores que la gráfica C.19 ω_p pero para nanoestructuras con morfología aleatoria con vista 3D.	143
C.24. Gráfico del flujo Poynting para los mismo valores que la gráfica C.18 ω_p pero para nanoestructuras con morfología aleatoria.	143
C.25. Gráfico del flujo Poynting para los mismo valores que la gráfica C.19 ω_p pero para nanoestructuras con morfología aleatoria con vista 3D.	143
D.1. (a) Dispositivo de fuente de voltaje, (b) estructura metálica que sostiene la película.	145
D.2. Vista superior: nanotubos abiertos (a) y su vista en sección transversal (b) de una matriz nano-tubular clásica de TiO_2 auto ordenada y sintetizada a través de la exposición de láminas Ti en (50 : 50Vol. %) H_2O : glicerol + 0,27MNH4F a un voltaje constante de 20V por 3,5h.	146
D.3. Resultados del voltaje de ruptura (a) y la corriente de fuga (b) de las películas nano-tubulares amorfas y cristalizadas TiO_2 obtenidas durante las pruebas de rigidez dieléctrica	147
D.4. Huella del área de contacto entre la película TiO_2 nano-tubular y el alambre de cobre.	148
E.1. Diagrama de una red neuronal artificial.	151

Índice de tablas

2.1.	Se muestran los resultados hechos para un rango de probabilidad de 0.01 hasta 0.2 con 10 probabilidades distribuidas en el rango. .	20
2.2.	Se muestra el comportamiento de P_{cr} , para varias pruebas en donde se varia la longitud del cilindro $\rightarrow h$ de 15 a 25 nanómetros, para una desviación $\mu_H = 1.4$	24
2.3.	Se muestran los resultados hechos para un rango de probabilidad de 0.01 hasta 0.2 con 10 probabilidades distribuidas en el rango. .	26
3.1.	Se muestra las posiciones de las tablas a las que se harán referencia al arreglo bidimensional.	38
3.2.	Se muestran los valores enteros en el arreglo.	38
5.1.	Parámetros de entrenamiento de la red neuronal para matrices 2D con distribución Uniforme.	101
5.2.	Parámetros de entrenamiento de la red neuronal para matrices 2D con distribución Gauss.	101
D.1.	Rigidez dieléctrica amorfa y cristalina. TiO_2 nanotubular film . .	147
D.2.	Resistencia y resistividad de las muestras amorfas y cristalinas determinadas a partir de las mediciones experimentales de rigidez dieléctrica.	148

ANTECEDENTES.

1.1. Introducción

La industria de la nanotecnología ha tomado gran fuerza en la construcción de nano dispositivos que impulsan a investigar nuevas posibilidades en la construcción, creación y modelado de nanoestructuras que faciliten su aplicación en las tecnologías emergentes aplicadas a la ciencia e innovación de las mismas.

Actualmente existen nanoestructuras con propiedades extraordinarias que son utilizadas en diferentes aplicaciones industriales. Actualmente y por sus grandes propiedades mecánicas, conductoras, resonantes, electromecánicas, y ópticas; los Nanotubos de Carbono (CNT) son uno de las nanoestructuras más comúnmente utilizadas en la ingeniería y en la construcción de infinidad de dispositivos. Algunas de las aplicaciones más frecuentes de los CNT son en la rama de la medicina para encapsular los virus por medio de películas a bases de CNT, aviación y espacio comúnmente aplicadas en los materiales aeronáuticos por su eficaz resistencia, en comidas como recubrimientos conservadores que retrasan su descomposición, en la electrónica por sus excelentes propiedades conductoras y en la generación y almacenamiento de energía [1]. Estas dos últimas tecnologías son abordadas en este trabajo de tesis, en donde nos enfocamos en el fenómeno de percolación en nanotubos de carbono en la conducción de corriente eléctrica que simulamos dentro de un sistema 3D compuesto por nanoestructuras con diferentes longitudes y grosores. Investigamos la probabilidad crítica (Número de nanotubos de carbono necesarios en el sistema para generar conducción) en el fenómeno de percolación, dato crítico importante y muy valorado por la industria de la nano-ingeniería en la construcción de los materiales.

El segundo fenómeno estudiado en este trabajo es la generación de energía mediante la interacción de una partícula y una estructura periódica de nano tubos de carbono resonante, dicha interacción produce la generación de plasmones de superficie, las cuales son frecuencias de onda superiores a la velocidad de la luz en un medio dieléctrico con oscilaciones que ascienden a niveles súper rápidos de Terahertz que son conveniente aprovechadas en la industria de las comunicaciones ópticas.

En esta investigación se presentan resultados que pueden servir como referencia a futuras investigaciones, y pueden servir de base en la creación de nuevos dispositivos electrónicos y en la aplicación de micro señales.

1.2. Planteamiento del problema.

Las necesidades tecnológicas en cuanto a rapidez y robustez de transmisión de información exigen cada día la mayor eficacia para cumplir las exigencias que satisfagan una amplia gama de aplicaciones tecnológicas. En la actualidad el estudio de polímeros como lo son las nanoestructuras de nanotubos de carbono tienen infinidad de aplicaciones, sobre todo en la aplicación de los ordenadores, ya que los CNT's tiene propiedades conductoras que pueden remplazar los hilos de cobre y agilizar la transmisión de datos. Las necesidades aeroespaciales también exigen un mayor rendimiento tanto en materiales como en transmisión de datos (Comunicaciones satelitales), nuestra investigación basada en energía y plasma que se genera con la interacción de una nanoestructura y una partícula radiante brinda una posibilidad de abastecer estas necesidades aprovechando las bajas permitividades en el espacio, que permiten acelerar la partícula a velocidades más cercanas a la de la luz con mayor facilidad.

1.3. Justificación

La primera parte de esta investigación se centra en encontrar la probabilidad crítica en un sistema compuesto por nanotubos de CNT para conducir la corriente eléctrica, este dato es muy apreciado en la industria de los materiales ya que se pueden ahorrar grandes cantidades en la construcción de cables conductores. Es decir en lugar de llenar todo el molde de cable conductor, solo se ocuparía la cantidad necesaria de CNT para alcanzar la conductividad pero sin alterar el resultado final. Este tiene gran significancia en el ahorro de materiales y por lo tanto en los costos en los materiales.

En la segunda etapa de investigación nosotros proponemos una forma de generar energía en forma de plasmones-polaritones en metales Drude considerando una carga con velocidades cercanas a la velocidad de La luz, que interactúa con un grupo nanoestructuras resonantes a base de CNT, en la actualidad es muy difícil generar este fenómeno microscópico es por ello que esta investigación se centra en generar una simulación numérica que brinda resultados que pueden ser aprovechados en tecnologías futuras, dado que actualmente generar un alto grado de resonancia en nanoestructuras que alcancen altos niveles de oscilaciones es demasiado caro. Nuestros resultados y simulaciones numéricas pueden servir como datos que brinden una aproximación cercana a la real que significaría ahorros de tiempo y dinero. Debido a que existe un amplio rango de frecuencias resonantes que pueden ser aplicadas a las nanoestructuras, resulta complejo realizar pruebas

para todo el espectro de frecuencias ω_p hasta encontrar una frecuencia optima que satisfaga las necesidades de la ingeniería. Nuestros resultados muestran un mejor acoplamiento entre la carga y el sistema de nanotubos (para la generación de energía) que pueden ser aprovechados en la transmisión de microseñales.

1.4. Hipótesis

1. La probabilidad crítica para un sistema 3D puede verse alterada cuando consideramos una desviación estándar σ sobre la longitud de los nanotubos de carbono al considerarlas dentro del sistema 3D.
2. Se puede aplicar la teoría de dispersión de cilindros para una consideración analítica para la creación de plasma sobre cilindros infinitos, mientras que con el método FDTD, se pueden considerar sistemas de cilindros finitos independientes.
3. Se puede considerar la formación de un campo electromagnético al interactuar una nanoestructuras periódica de CNT + una carga radiante (radiación de Cherenkov).
4. La frecuencia de plasma ω_p puede alterar los niveles en la creación de plasmones polaritones en metales Drude.
5. A mayor frecuencia de plasma, mayor cantidad de energía W_c se genera.
6. Los niveles de energía obtenidos para nanoestructuras periódicas son diferentes a los niveles de energía con nanoestructuras random en la morfología de los nanotubos.
7. Los niveles de energía en sistemas periódicos son generalmente mayores a los nanoestructuras con morfología random en su radio y altura.
8. Los niveles de energía generados entre la interacción (nanotubos + carga) CNT son diferentes a los nanoestructuras de TiO_2 .
9. Las velocidades de carga con la que viaja la partícula afecta considerablemente los niveles de generación de plasmones-polaritones de superficie.
10. Se puede optimizar un sistema en sus cálculos numéricos utilizando programación en paralelo, aplicados al estudio de fenómenos físicos. Reduciendo considerablemente los tiempos de ejecución aprovechando al tope de más capacidades del software de la computadora.

1.5. Objetivo General

Aplicar el método de diferencias finitas en el dominio del tiempo en nanoestructuras con propiedades electromagnéticas

1.6. Objetivos Específicos

- Estudiar el comportamiento en las nanoestructuras con morfología CNTs para encontrar sus propiedades críticas de conducción eléctrica utilizando una distribución uniforme aplicado a los radios y longitudes de cada nanotubo.
- Aplicar del método FDTD (Función de Diferencias Finitas en el Dominio del Tiempo) para el modelado de nanoestructuras 3D a fenómenos electromagnéticos.
- Estudiar la radiación de Cherenkov sobre metales Drude.
- Modelar una malla 3D con nanoestructuras SWCNT (Nanotubos de carbono de pared Blanda) con nanoestructuras periódicas con espacios vacantes y aleatorios.
- Realizar pruebas para diferente puntos de ω_p (frecuencia de plasma aplicada a la nanoestructura CNTs) que genera una partícula con un velocidad v determina que pasa sobre ellas.
- Encontrar el pico Máximo de radiación de energía para diferentes frecuencias en las nanoestructuras en un intervalo ω_p .
- Investigar el mejor acoplamiento entre carga y Nanotubos para los casos CNTs vs TiO_2 con radio y longitud variable en sus morfologías
- Investigar la variación de cambios de energía cuando la configuración de las nanoestructuras cambia de orientación para los casos de nanoestructuras periódicas. Caso: Malla de CNTs.

ESTUDIO DE LA CONDUCTIVIDAD MEDIANTE PERCOLACIÓN EN NANOTUBOS DE CARBONO.

2.1. Introducción a la conductividad eléctrica median CNT's

En este capítulo abordamos el estudio de la conductividad eléctrica usando nanoestructuras; para nuestro caso de estudio utilizamos nanotubos de carbono (CNT), que son nanoestructuras de pared simple a base de grafeno que actúan como conductores de la corriente eléctrica con poca resistividad al paso de los electrones. Podemos observar en la Figura 2.1 la estructura de un CNT típico con enlaces hexagonales de carbono de manera enrolladas para formar un CNT. La

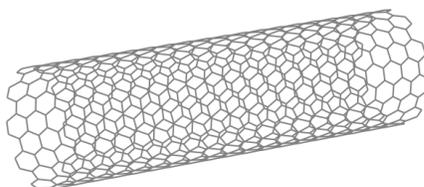


Figura 2.1: Morfología típica de nanoestructura de Carbono CNT de pared simple a base de grafeno.

idea es simular varios CNT contenidos en un arreglo tridimensional donde las intercepciones o contactos entre estos CNT podrán ayudar al electrón a viajar entre las capas de grafeno cilíndricas y mediante estas intercepciones formar una percolación de CNT's. El primer paso es generar el arreglo 3D con alguna dimensión deseada considerada en nanómetros para escalarla a las medidas de los nanotubos que van de 1 nm a 15 nm los cuales son las medidas típicas usadas en la ingeniería para la creación de CNT's [2–17]. Para cada posición en el arreglo 3D existe una nanoestructura, pero también podemos elegir el número de nanoestructuras que pueden existir en el arreglo que los contiene.

La idea es simular varios CNT contenidos en un arreglo tridimensional donde las intercepciones o contactos entre estos CNT podrán ayudar al electrón a viajar entre las capas de grafeno cilíndricas y mediante estas intercepciones formar una percolación de CNT's. El primer paso es generar el arreglo 3D con alguna dimensión deseada considerada en nanómetros para escalarla a las medidas de los nanotubos que van de 1 nm a 15 nm los cuales son las medidas típicas usadas en la ingeniería para la creación de CNT's. Para cada posición en el arreglo 3D existe una nanoestructura, pero también podemos elegir el número de nanoestructuras que pueden existir en el arreglo que los contiene. El objetivo es economizar el mayor número de CNT's o material de grafeno utilizable en la conducción de energía eléctrica y esto se puede lograr probando varios niveles de probabilidad hasta encontrar la probabilidad crítica que es el valor que minimizará el número de nanotubos necesarios o suficientes que se necesitan para conducir la corriente eléctrica en el sistema 3D de nanotubos, quizás para un ejemplo puede ser poco ahorro pero a nivel escalar como lo son en las industrias estos ahorros pueden representar grandes cantidades de ahorro de dinero en material de grafeno, en tiempo y ahorro tecnológico en la producción de las nanoestructuras.

2.1.1. Array de Nanotubos

El primer paso es generar CNT's en el arreglo 3D con un cierto nivel de Probabilidad, donde la probabilidad P es el porcentaje que puede variar de 0 a 1, donde 0 significa que no se han generado nanotubos y 1 es cuando todo el arreglo está lleno de CNT's. Ver figura 2.2.

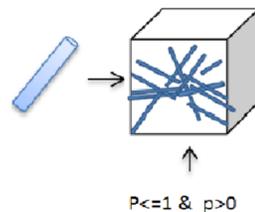


Figura 2.2: Llenado de un Arreglo 3D con Nanotubos CNT, donde el nivel de probabilidad P no puede ser mayor a 1 ni menor a 0.

2.2. Arreglo de nanotubos con radio, altura y orientación fijos.

Generamos un sistema computacional en donde modelamos cilindros que representan los nanotubos de carbono contenidos en un arreglo, en primera instancia generamos nanotubos con una longitud fija $h=2\text{ nm}$ y 0.5 nm de grosor o radio r y con 6 posibles orientaciones a partir del centro del cilindro, es decir que este cilindro comenzara a formarse a partir del punto central del arreglo como podemos

visualizar en la Figura 2.3.

Podemos observar de la figura 2.3 las posibles orientaciones que pueda tomar el nanotubo simulado en una determinada posición de la rejilla tridimensional, las posiciones crecen 2 nm hacia arriba, abajo, adelante, atrás, hacia la izquierda y hacia la derecha del centro de su posición en el arreglo 3D. De esta manera generamos diferentes cilindros en todo el arreglo variando la probabilidad P , que es el número de nanotubos o

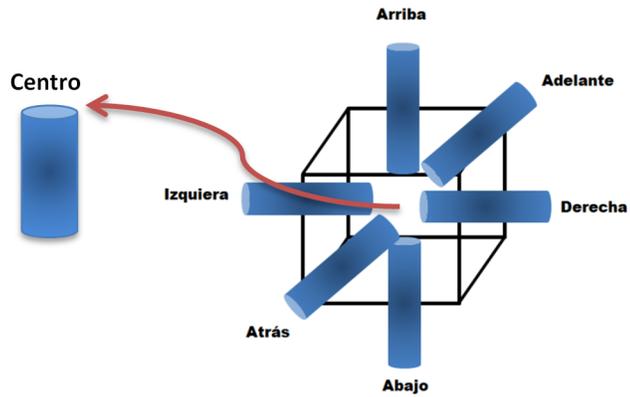


Figura 2.3: Orientaciones del CNT a partir del centro de una determinada posición del Arreglo 3D.

porcentaje de nanotubos que se deben crear en el sistema tridimensional. Utilizamos el lenguaje *CShar* para simular y poder representar nuestro sistema visualmente, en donde por medio de vectores representamos la dirección que toman los nanotubos como podemos observar en la Figura 2.4, con diferentes probabilidades P . Para las imágenes 2.4(a) con una Probabilidad $P=0.1$ podemos observar las direcciones de los vectores hacia donde crecen los nanotubos, mientras que en la figura 2.4(b) con $P=0.4$ podemos observar un mayor número de nanotubos creados.

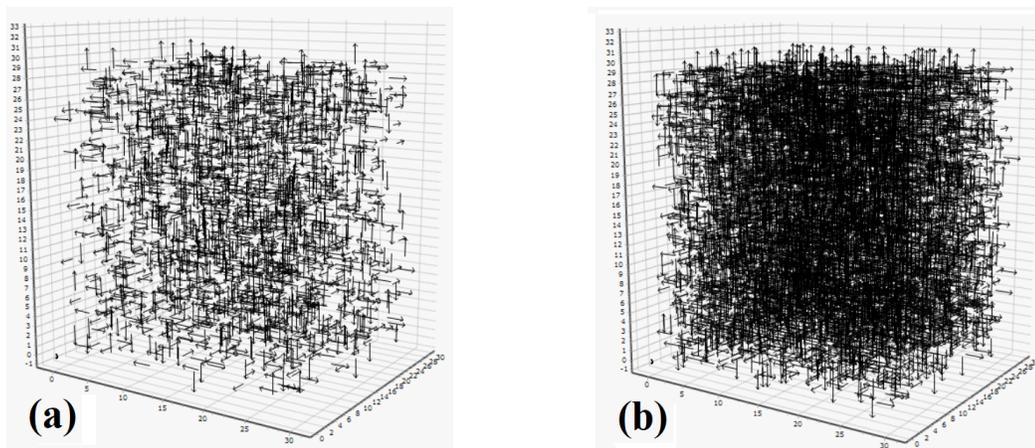


Figura 2.4: La imagen (a) con una probabilidad $P=0.1$ en un tamaño de rejilla $L=30$, podemos observar las direcciones de los vectores hacia donde crecen los nanotubos, mientras que en la imagen (b) con $P=0.3$ podemos observar un mayor número de nanotubos creados.

El siguiente paso es identificar a los clústeres, es decir identificar que cilindros tiene

contactos entre si y que forman un grupo aglomerado de cilindros. Las ecuaciones 2.1-2.6 permiten interceptar los contactos para cada cilindro de 2 nm (nanómetros) de largo entre las celdas para cada elemento del sistema 3D.

$$C - Abajo = \sum_{i=1}^{L-2} \sum_{j=1}^{L-2} \sum_{k=1}^{L-2} G(i, j, k) + G(i, j - 2, k) \quad (2.1)$$

$$C - Arriba = \sum_{i=1}^{L-2} \sum_{j=1}^{L-2} \sum_{k=1}^{L-2} G(i, j, k) + G(i, j + 2, k) \quad (2.2)$$

$$C - Izquierda = \sum_{i=1}^{L-2} \sum_{j=1}^{L-2} \sum_{k=1}^{L-2} G(i, j, k) + G(i - 2, j, k) \quad (2.3)$$

$$C - Derecha = \sum_{i=1}^{L-2} \sum_{j=1}^{L-2} \sum_{k=1}^{L-2} G(i, j, k) + G(i + 2, j, k) \quad (2.4)$$

$$C - Adelante = \sum_{i=1}^{L-2} \sum_{j=1}^{L-2} \sum_{k=1}^{L-2} G(i, j, k) + G(i, j, k - 2) \quad (2.5)$$

$$C - Atras = \sum_{i=1}^{L-2} \sum_{j=1}^{L-2} \sum_{k=1}^{L-2} G(i, j, k) + G(i, j, k + 2) \quad (2.6)$$

En las ecuaciones 2.1 a 2.6 se analizan los 6 vecinos más próximos del nanotubo en donde puede existir algún contacto, para el caso de la Ecuación 2.1 (Prefijo C=Contacto), donde existe 3 sumatorias anidadas; la primer sumatoria representa las capas del sistema 3D, es decir en un sistema cubico esta sumatoria representa el valor Z , la segunda sumatoria representa las columnas de una matriz y la tercer sumatoria representa las filas de una matriz. La suma: $A(i, j, k) + A(i, j, k + 2)$, suma los valores entre los cilindros vecinos. Si la suma es 2, significa que hay un cilindro vecino, pero si la suma es 1, significa que no existe ningún contacto vecino y el valor de 1, es el valor del cilindro que está en el centro: $A(i, j, k)$. Los mismo sucede para las ecuaciones 2.2 a 2.6, pero analizando sus vecinos correspondientes, arriba $\Rightarrow A(i, j + 2, k)$, izquierda $\Rightarrow A(i - 2, j, k)$, derecha $\Rightarrow A(i + 2, j, k)$, adelante $\Rightarrow A(i, j, k + 2)$ y atrás $\Rightarrow A(i, j, k - 2)$ respectivamente. Si alguno de los 6 vecinos arroja una suma de 2, el cilindro evaluado que está en el centro será etiquetado para formar un clúster de lo contrario no se etiquetara y será borrado del sistema 3D. Podemos observar la aplicación de estas ecuaciones en la figura 2.5.

Utilizamos el algoritmo Hoshen-Kopelman para el etiquetado de los colores en los vectores de la matriz. Podemos observar de la Figura 2.5 agrupaciones multicolores, los cuales son grupos de contactos entre los cilindros formando un pequeño grupo con uniones entre sí, identificados por un color diferente para cada clúster. Es evidente que si producimos una corriente eléctrica desde la base, no todos los

clústeres que se encuentran en el sistema conducirán la corriente eléctrica, ya podemos observar en la Figura 2.5 que pueden existir muchas interconexiones entre los cilindros y formar numerosos clústeres pero no todos tiene contacto con la base del sistema hacia la superficie del sistema (de abajo hacia arriba). Solo el clúster

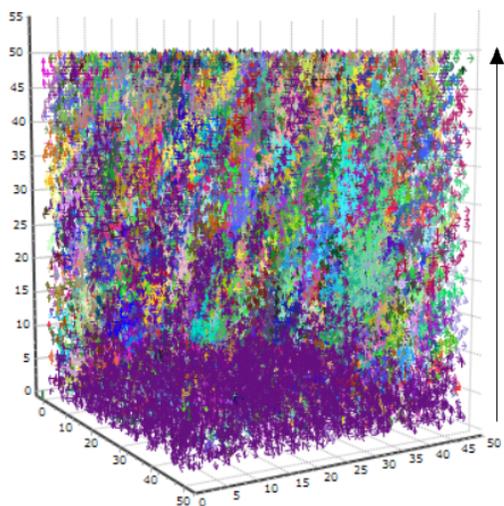


Figura 2.5: Clústeres de cilindros aglomerados identificados por cada color. La probabilidad de este sistema es $P = 0,5$, en un tamaño de rejilla $L = 50$.

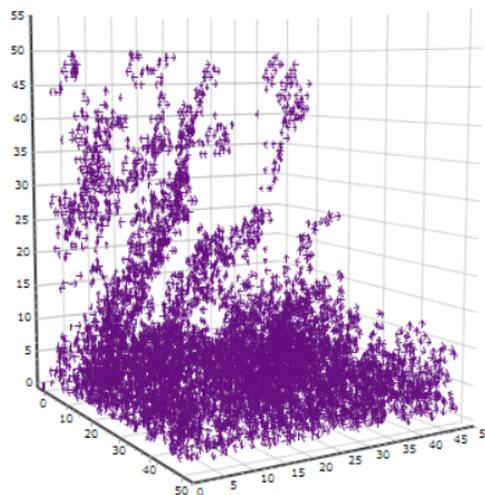


Figura 2.6: 3D que muestra un clúster infinito de percolación y conductividad eléctrica en donde con una Probabilidad $P = 0,5$ y una tamaño de rejilla de $L = 50$.

más grande y que crece desde la base hacia la superficie indicada con una flecha en la Figura 2.5, puede conducir la corriente eléctrica. En la figura 2.6 podemos observar cómo se han eliminado todos aquellos clústeres que no tienen contacto con la base de nuestro sistema y que solo aquel clúster predominante o clúster infinito atraviesa todo el sistema desde la base hasta la superficie conduciendo la corriente eléctrica, ver figura 2.6. Podemos observar un sistema definido de nanoestructuras representados por vectores que forman un clúster infinito de conductividad eléctrica en el cual existe percolación, más sin embargo este sistema está delimitado por el largo de las nanoestructuras el cual solo pueden alcanzar longitudes de 2 nm y delimitada también solo para 6 orientaciones definidas. Este modelo aun que resulta ser bueno, tiene sus limitantes ya que en la percolación real de nanoestructuras, las longitudes pueden tener una longitud mayor a 2 nm , la longitud promedio de los CNT's oscila entre los 15 a 25 nm de longitud con variaciones de radio desde 0.1 nm hasta 2 nm de grosor para casos prácticos, además de que la orientaciones no están limitadas en direcciones rectas a 6 puntos de orientación (arriba, abajo, derecha izquierda, adelante y atrás) si no que los ángulos pueden tener una inclinación deliberada de 360 grados en cualquier dirección del nanotubo, es por ello que en la siguiente sección enfocamos nuestro estudio a desarrollar este modelo, resolviendo las limitantes anteriormente mencionadas. Sin embargo este

caso de estudio puede servir para modelar fenómenos con propiedades similares en la naturaleza.

2.3. Nanotubos con longitud, radio y orientación aleatoria.

En la práctica real pueden crearse nanotubos con diferentes longitudes y diferentes radios, los CNT's típicos utilizados en la industria con medidas que van desde 1 *nm* hasta los 15 a 25 nano metros de longitud h y con variaciones de grosores hasta 2 *nm* de radio r .

Es por ello que realizamos un segundo sistema donde las nano estructuras pueden ser lo suficiente mente largos y con un grosor variable para adaptar las condiciones físicas de la creación de nanotubos de carbono lo más parecido a como se generan industrialmente. Estos sistemas son utilizados en la industria para crear sistemas piezoeléctricos, membranas conductoras y micro-antenas. Este sistema consiste en generar nanotubos aleatorios en el sistema 3D con diferente orientación es decir; los nanotubos se generan a partir del centro de la posición determinada del arreglo hacia cualquier ángulo de la nano estructura.

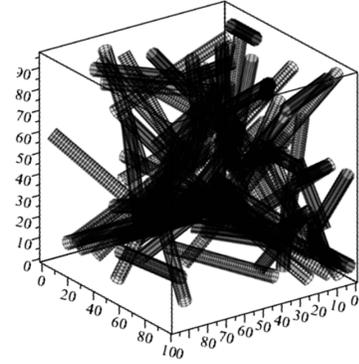


Figura 2.7: Modelo 3D de nanocompuestos CNT's desordenados.

2.4. Propiedades del Objeto Cilindro.

Utilizamos las siguientes ecuaciones para poder generar las nanoestructuras para determinar su forma espacial en el arreglo 3D.

$$z_2 = z_1 + h \cdot \cos(\theta) \quad (2.7)$$

$$y_2 = y_1 + h \cdot \sin(\theta) \cdot \sin(\phi) \quad (2.8)$$

$$x_2 = x_1 + h \cdot \sin(\theta) \cdot \cos(\phi) \quad (2.9)$$

Dónde: x_1, y_1, z_1 : Son los valores iniciales de donde el nanotubo iniciará. h es la altura o longitud del nanotubo con distribución normal o Gauss. r es el radio o grosor del nanotubo con distribución normal. De los parámetros mencionados anteriormente, se obtendrán los valores finales y con ello se determinará la ubicación y forma del nanotubo en la rejilla 3D. x_2, y_2 y z_2 son las coordenadas finales del cilindro θ y ϕ son ángulos aleatorios que determinan la orientación, h es la altura o longitud del cilindro y r es el radio del cilindro o grosor de nanotubo.

En la figura 2.7 podemos observar el objeto cilindro en su forma espacial 3D, con

un punto de inicio; x_1, y_1, z_1 , y un punto final x_2, y_2 y z_2 con un radio r y una altura $h = d$. Donde x_1, y_1, z_1 son valores aleatorios que determinaran los puntos finales x_2, y_2 y z_2 , a partir de las ecuaciones 2.7-2.9

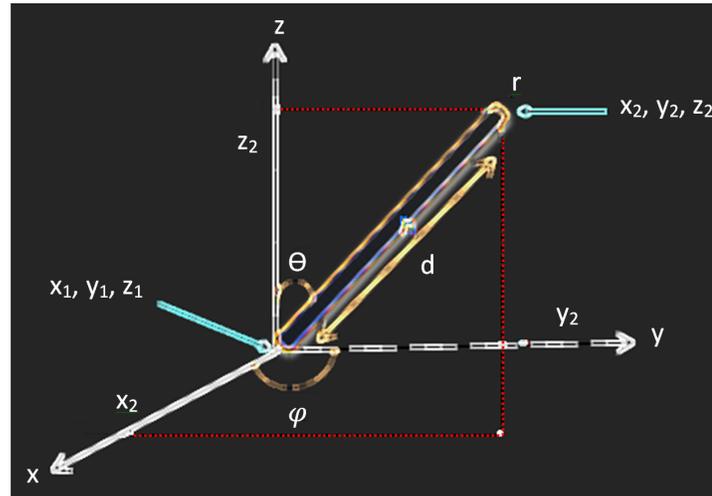


Figura 2.8: Esquema espacial del nanotubo 3D con sus parametros y variables correspondientes a las ecuaciones 2.7-2.9.

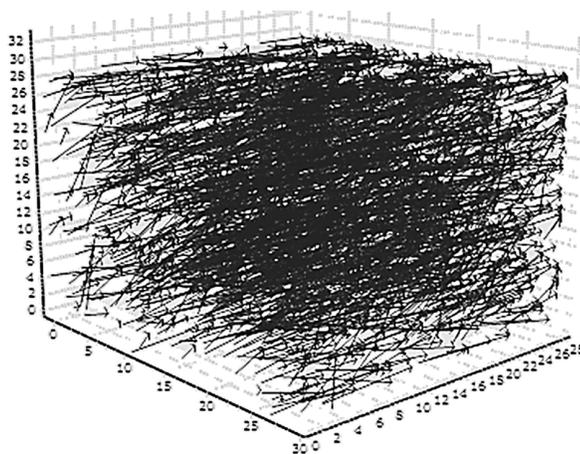


Figura 2.9: CNT's Con longitud aleatoria de 1 hasta 15 nm de longitud en una malla tridimensional de $L = 30$, y probabilidad $P = 0,1$.

Podemos observar diferentes vectores que representan los nanotubos con diferentes direcciones, orientaciones y longitudes para cada objeto dentro del arreglo 3D.

En la figura 2.9 podemos observar los vectores utilizados como en el primer ejemplo pero con diferente orientación y longitud. Utilizamos la librería Stemma-Techart para hacer usos de los gráficos y mostrar nuestras simulaciones gráficamente, sin embargo esta librería se ve limitada a utilizar vectores visualizando las longitudes de cada vector mas no así sus grosores. En la figura 2.9 observamos varios cilindros en una malla numérica simulada de $L = 30$, con una probabilidad

2.5. Intercepciones entre los nanotubos.

Para identificar las intercepciones en el sistema y poder visualizar los clústeres en la rejilla tridimensional es necesario encontrar los contactos entre los nanotubos, para ellos utilizamos las ecuaciones:

$$d^2 = r_1^2 + 2r_1r_2 \cdot \cos(\theta) + r_2^2 \quad (2.10)$$

$$r_1^2 + r_2^2 - d^2 = -d \cdot r_1r_2 \cdot \cos(\theta) < 2r_1r_2 \quad (2.11)$$

$$r_1^2 + r_2^2 - d^2 < 2r_1r_2 \quad (2.12)$$

Dónde: $\frac{\pi}{2} \leq \theta \leq \pi$, r_1 , es el radio del nanotubo horizontal, r_2 es el radio del nanotubo vertical y d es la distancia entre los centros de sus radios.

En donde validamos el radio de los nanotubos para encontrar un contacto entre ellos, si la condición se cumple entonces habrá un contacto seguro entre los nanotubos, podemos observar el esquema de las ecuaciones en la figura ??, donde observamos un traslape entre dos nanotubos con diferentes orientaciones, un nanotubo vertical y un nanotubo horizontal con diferentes radios r , entre ambos.

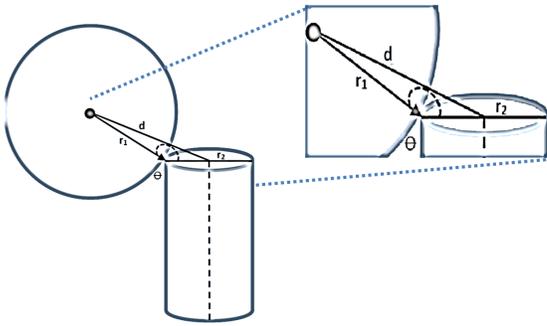


Figura 2.10: Esquema que muestra la intercepción entre dos cilindros con contacto perpendicular correspondientes a las ecuaciones 2.10-2.12

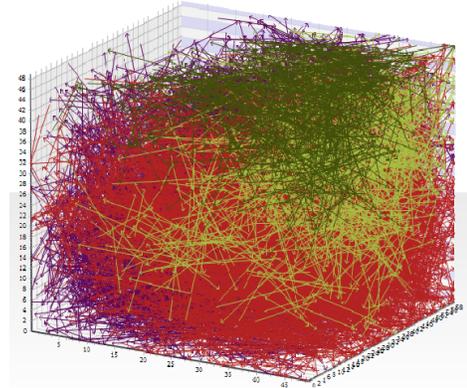


Figura 2.11: Clústeres de cilindros aglomerados identificados por cada color. La probabilidad de este sistema es $P = 0,5$, en un tamaño de rejilla $L = 50$.

En la figura 2.11, podemos observar una rejilla que contiene los diferentes clústeres en el arreglo con un color indicado para cada uno. Los clústeres son las formaciones de grupos aglomerados que forman una estructura con contacto entre ellos.

2.6. Clúster de Percolación (Clúster infinito).

Para nuestro caso de estudio es importante conocer los clústeres dentro del sistema y que tienen contacto entre sí, más sin embargo la finalidad de nuestro

estudio es conocer cuál de esos clústeres conducirá la corriente eléctrica, El clúster infinito es denominado comúnmente clúster de percolación, y es precisamente este clúster el clúster más largo, en donde los contactos comienzan desde un extremo del sistema hasta alcanzar el otro extremo es decir cilindros que tienen contacto desde el principio hasta el final. Es precisamente por este clúster de percolación que el sistema puede conducir una corriente eléctrica desde un punto a otro. En la figura 2.12 podemos ver la estructura típica de un clúster de percolación conformado por nanotubos que atraviesa la rejilla simulada desde un extremo al otro. La flecha roja indica la dirección creciente del clúster de percolación, la altura que forma la estructura del conjunto de nanotubos es importante ya que en nuestras simulaciones numéricas validamos la altura del clúster, si la altura del clúster es igual o mayor al sistema habrá percolación, de lo contrario no existirá percolación en el sistema.

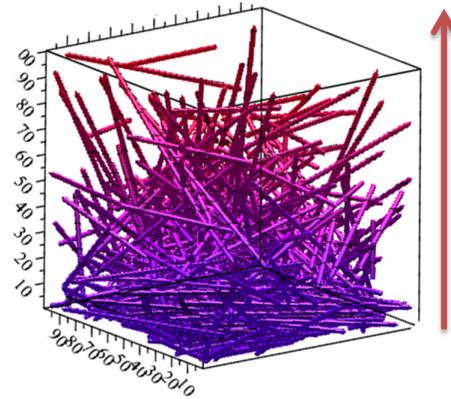


Figura 2.12: Rejilla 3D de tamaño $L=100$, donde fueron distribuidos Nanotubos (representados por cilindros) de tamaños variables. Podemos observar como se forma el clúster infinito de principio a fin en la rejilla simulada.

El sistema simulado en la figura 2.12, para fines visuales consta de un tamaño $L=100$, donde fueron distribuidos nanotubos (representados por cilindros) de tamaños variables. Podemos observar como como prevalece el clúster infinito de principio a fin en la rejilla simulada.

2.7. Distribución Normal-Gauss aplicado a tamaños y radios en CNTs

Es bien sabido que en la industria de los nanotubos es difícil crear nanotubos periódicos y siempre existen errores en su altura o longitud, así como en el grosor de cada uno de los nanotubos creados artificialmente. En nuestro estudio es considerada esta propiedad para apegarse lo más posible a los resultados reales en la fabricación de tales nanoestructuras. Por otra parte sabemos que en todo aspecto de la vida siempre ese considerado un promedio para un intervalo de características estudiadas, por ejemplo si estudiamos la altura de las mujeres de una determinada población, habrá siempre una altura promedio que prevalece, es decir habrá algunas cuantas personas con altura bajita y otras con altura muy alta, pero las otras restantes estarán dentro de un promedio más uniforme. Otra analogía puede ser el del peso de las personas, puede haber personas más gorditas o más flaquitas pero siempre prevalecerá un promedio, según las condiciones de la región estudiada.

Esta características es también considerada en nuestro estudio es por ello que aplicamos la ecuación de Gauss que puede atender estos 2 parámetros importantes, la desviación de error y el promedio de alturas y radios de los CNTs.

La distribución de Gauss sigue una forma de campana, para nuestro caso de estudio es considerado la altura y radio de los nanotubos basándonos en el ejemplo de la altura, el principio de la campana se interpreta como: la existencia nanotubos con alturas pequeñas en el sistema, la punta de la campana significa que hay en promedio muchos más nanotubos con esa altura, mientras que el final de la campana significa que hay algunos cuantos nanotubos con altura más grande al promedio. Lo mismo se aplica al radio para generar el grosor de los nanotubos. En la figura 2.13 podemos considerar a σ o desviación que puede tener una determinada media (altura y radio de los nanotubos), la variable a es el punto promedio donde existen más nanotubos con alturas y grosores similares, el valor negativo (-) está cerca de las medidas más chicas de los nanotubos y el valor (+), es considerado que se acercan a medias que están sobre el promedio.

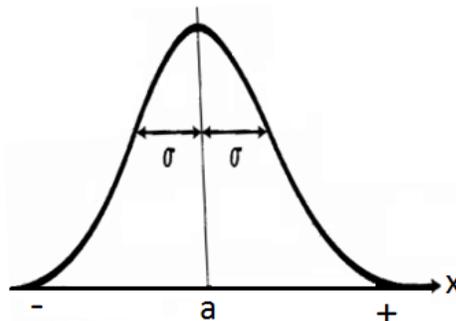


Figura 2.13: 11 Esquema de la distribución de Gauss utilizada en nuestro caso de estudio para generar medidas aleatorias en los nanotubos.

La ecuación de la distribución normal está dada por:

$$\frac{1}{\sqrt{2\pi}\sigma_i} e^{-\left(\frac{x - \mu_i}{2\sigma_i}\right)^2}, \quad i = r, h. \quad (2.13)$$

Aplicamos a esta ecuación dos parámetros importantes; el radio para los nanotubos y un determinada altura para los mismos, así como una desviación para obtener un promedio en las medidas de los nanotubos, en las imágenes de las figuras 2.14(a) y 2.14(b) podemos observar los histogramas de distribución para los radios y las alturas.

La figura 2.14(a), muestra una distribución normal con medidas para la altura $\mu_H=20$ de los nanotubos y una desviación de $\sigma_h=2$, teniendo como resultado una distribución de alturas que van desde alturas pequeñas $hmin = 13$ hasta $hmax = 27,5$. En el caso e la figura 2.14(b) observamos una distribución para los radios de los nanotubos con una desviación de $\sigma_r=0.025$ para nanotubos delgados con un promedio de grosor $\mu_R=0.01$, teniendo como resultado nanotubos con radio de grosor entre $rmax = 0,1$ y $rmin = 0,01$.

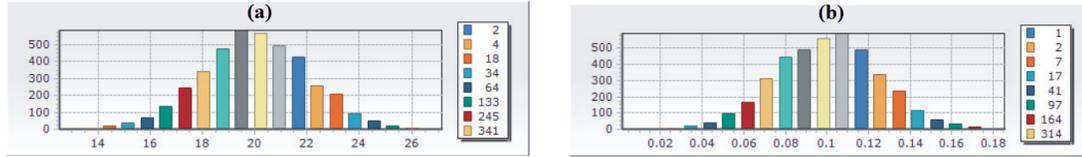


Figura 2.14: Histogramas: (a) Distribución normal correspondiente a la altura $\mu_H=20$ y desviación de $\sigma_h=2$, con $hmin = 13$ y $hmax = 27,5$, (b) tenemos valores de aplicados a los radios de las nanoestructuras de $\mu_R=0.01$ y desviación de $\sigma_r=0.025$, con $rmax = 0,1$ y $rmin = 0,01$.

2.8. Probabilidad Crítica P_{cr} .

La probabilidad crítica está ligada directamente con el valor P (Valor de probabilidad). El valor P es un parámetro importante que puede variar en un intervalo de 0 a 1, si $P = 0$, dentro de nuestro sistema o rejilla simulada no se generaran nano estructuras, ya que P lleva el control de cuantas nano estructuras hay que generar en el sistema, es decir será el porcentaje de nanotubos generados en la rejilla que representa un valor entero $\rightarrow 1$, Si el valor de $P = 1$, esto representa una rejilla simulada llena de nanoestructuras, ese decir toda la rejilla estará ocupada de nanotubos. Mientras que si asignamos $P = 0,5$, la mitad del sistema estará ocupada por nanotubos. Podemos observar esta relación en las imágenes de las figuras 2.15(a), con una probabilidad $P = 0,01$ con pocos nanotubos, 2.15(b) con $P = 0,07$, mientras en la imagen 2.15(c), tenemos una producción de nanotubos con $P = 0,65$. Para poder estimar y encontrar la probabilidad crítica de cualquier sistema es necesario variar el valor de P , ya que la probabilidad crítica P_{cr} es una probabilidad que al cambiar su valor en un estado crítico este puede generar los

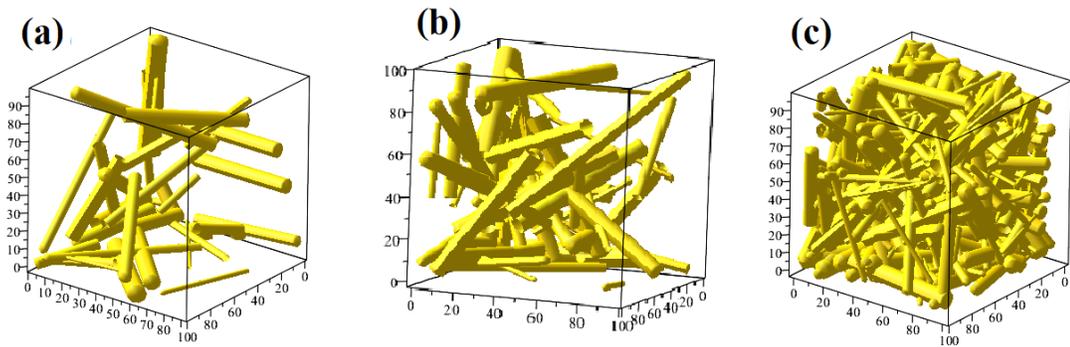


Figura 2.15: Distribución de probabilidades con diferentes saturaciones de nano estructuras con valores $P = 0,01$ para el caso (a), para el caso (b) tenemos una $P = 0,07$, y el caso se observamos una probabilidad de $P = 0,65$.

suficientes nanotubos que formen un clúster de nanotubos suficiente mente grande que atraviese el sistema de un extremo a otro.

Para casos prácticos es necesario visualizar solo el clúster predominante en el sistema y eliminar los clúster sobrantes, para poder observar la altura del clúster más grande con una probabilidad asignada en las imágenes de la figura 2.16 podemos observar dos casos en donde visualizamos los clústeres más grandes en el sistema con tamaño de rejilla $L = 100$ para ambos casos, para el caso 2.16(a) con $P = 0,02$ y el caso 2.16(b) con $P = 0,25$, observamos una diferencia en la altura del clúster entre los dos casos, ya que en el primer caso la altura del clúster no alcanza a tocar el otro extremo del sistema, pero para el caso (b), existe un clúster infinito y por lo tanto existe percolación en la rejilla 3D. Para ambos casos simulamos nanotubos largos y delgados con longitud $\mu_H=20$ y radio de $\mu_R=0.01$.

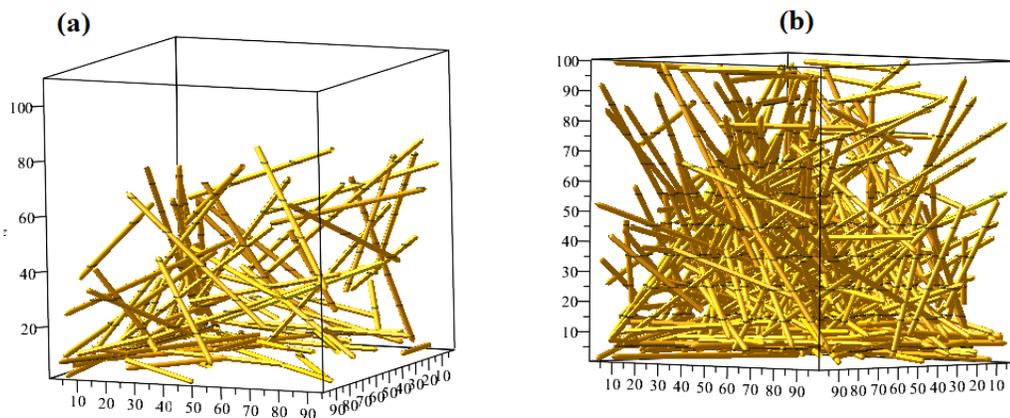


Figura 2.16: (a) Clúster con probabilidad $P = 0,02$ y (b) Clúster con $P = 0,25$ con longitud $\mu_H=20$ y radio de $\mu_R = 0,01$ para ambos casos. Con orientación de abajo hacia arriba.

Como podemos observar de las imágenes de la figura 2.16(a, b), la Probabilidad P afecta la percolación en los sistemas tridimensionales. En el ejemplo mencionado anteriormente observamos dos casos, en donde el primer ejemplo alcanza claramente la percolación y el otro no. Sin embargo las probabilidades P para la figura 2.16(a) y 2.16(b), distan mucho entre ambas. Podemos observar que la figura 2.16(a), tiene una altura aproximada de 80, mientras que la altura de la figura 2.16(b) tiene una altura tope de 100. Para poder encontrar la probabilidad P , necesitamos obtener un rango de probabilidades e incrementar P con pasos pequeños. De esta manera podremos registrar el punto P donde el clúster alcanza la altura necesaria para generar conductividad, a esta probabilidad de transición es llamada probabilidad crítica P_{cr} .

La probabilidad crítica para cada sistema pueden variar según los parámetros de entrada, es decir una probabilidad crítica no será la misma cuando asignamos un mayor o menor radio en la distribución o una corta o una larga longitud en las medidas del nanotubo. En nuestra investigación consideramos ambos factores. En primera instancia fijamos el valor de μ_H y variamos el valor de μ_R , centrándonos en el estudio de la probabilidad crítica cuando los radios varían y en la longitud de las nanotubos fijando μ_H y variando μ_R .

2.9. Estructura de programación en paralelo

Para la programación y generación de los resultados utilizamos como lenguaje de programación, Microsoft Visual *CSharp*, en su versión 2017. Se crearon 4 clases principales; la primera clase *Cilindro*, donde se construye el objeto cilindro que simulará cada uno de los CNT, la segunda clase *ArrayCilindros*, la cual contiene varios objetos en la rejilla, en esta clase se define si existe o no percolación para una determinada probabilidad P y también se obtiene el volumen de clúster infinito sobre total de los cilindros en el sistema $P(p)$. La tercera clase, *ParallCilindros* se encarga de realizar varias calculaciones para una distribución de P en paralelo usando la librería *TPL(TaskParallLabel)* de *CSharp*, ya que para cada probabilidad puede existir o no clústeres infinitos, es por ello que se necesitan calcular varias pruebas variando P , en donde se tienen que definir una probabilidad inicial, una probabilidad final y el número pruebas que se quieren distribuir en la probabilidad, de esta manera se distribuyen N probabilidades en un rango de probabilidad a estudiar.

Para encontrar la probabilidad crítica P_{cr} distribuimos 10 calculaciones en un intervalo de probabilidad inicial P_i hasta una probabilidad final P_f , (Donde; $i = inicial$, $f = final$). En las figuras 2.17(c) y 2.17(d) podemos apreciar un ejemplo de la distribución de las probabilidades $P_i = 0,1$ y $P_f = 1$, donde el valor 1 significa que todo el sistema estará lleno de nanotubos. La diferencia entre ambos casos es el orden en que se ejecutan las acciones, mientras que en el caso de la figura 2.17(c), observamos la ejecución de las acciones en serie, es decir que si necesitábamos 10 probabilidades en un rango $P_i - P_f$, necesitamos hacer y

calcular una probabilidad a la vez, lo que hacía que la calculación de los resultados demoran más, ya que una calculación no podía comenzar hasta terminar la anterior, es por ello que empleamos *Thread's* (hilos), herramienta que permiten el multiproceso de tareas mientras que ejecutan otras a la vez como se muestra en la figura 2.17(d), esto permite optimizar y distribuir tareas, asignando un rango de probabilidades a la vez para ejecutarse, donde cada tarea es independiente de la otra.

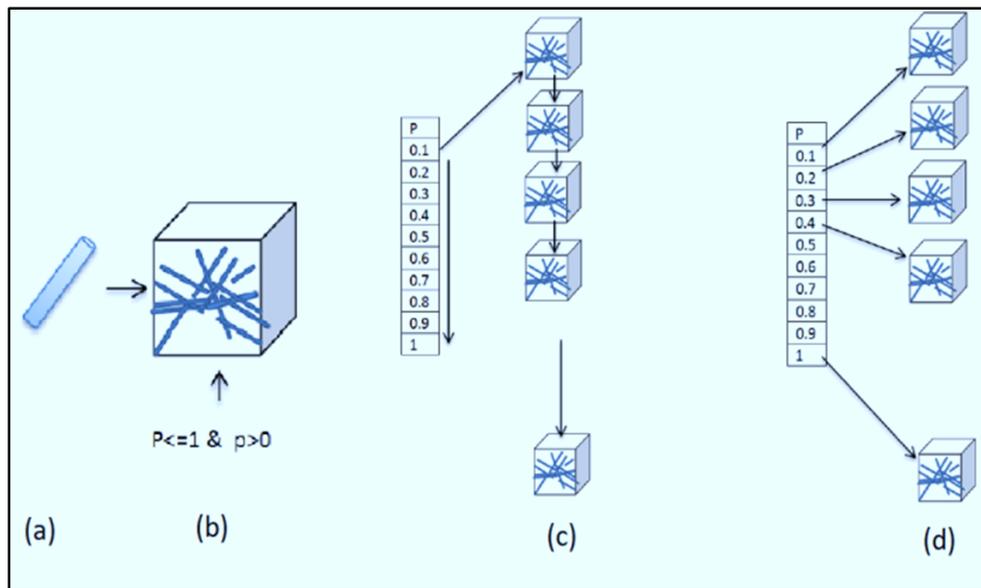


Figura 2.17: Esquema de programación distribuida: (a), muestra el modelo de nanotubo presente en el sistema, (b), muestra un sistema 3D compuesto de nanotubos, (c) esquema de programación en serie para 10 probabilidades distribuidas, (d) imagen que muestra una programación en paralelo donde se distribuye un rango de probabilidades a la vez y cada tarea es independiente a las otras.

2.9.1. Parámetros de entrada

Los parámetros de entrada son: $L=50,50,60, hRn=T(15,1E-05), rRn=T(0.1,0.025)$, $Gau, sd=12, av=1, P=0.1$. Donde L es el tamaño del sistema 3D 50×50 de base y 60 de altura. Las nanoestructuras o CNT's se construyen independientemente con una altura de $\mu_H=15 \text{ nm}$ (nanómetros), y una desviación estándar $\sigma_r = 1 \times 10^{-5} \text{ nm}$, para el caso de los radios CNT's tenemos $\mu_R=0.1 \text{ nm}$ y con una desviación $\sigma_r = 0,025 \text{ nm}$. El parámetro Gau , representa una distribución normal aplicada $P(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-a)^2}{2\sigma^2}}$, $sd(seed)$ es el parámetro semilla para los Random (aleatorios) con valor 12, av (average) es el promedio de repeticiones para un mismo punto de probabilidad $P = 0,1$.

El tiempo para una calculación fue aproximadamente de 10 a 15 minutos para una sola calculación, pero cuando se programó el sistema de forma paralela se obtuvo

un tiempo aproximado de 50 a 60 minutos para 10 calculaciones al mismo tiempo. Estos resultados se ejecutaron en una computadora Dell Intel (R) Core (TM) i7-6700HQ de 64 bits, a 2.60 GHZ con 16 GB de RAM de almacenamiento. El software para la calculación para obtener los resultados de la probabilidad crítica fueron programados en un lenguaje Visual *CSharp* en su versión 2017, mientras que los resultados fueron graficados con Matlab en su versión 2017, a partir de los resultados de *CSharp* almacenados en un archivo *.dat.

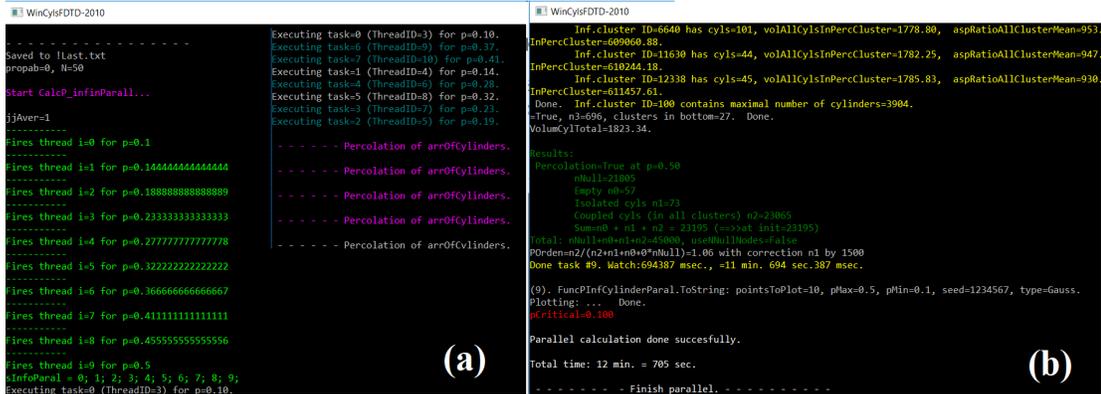


Figura 2.18: Distribución de un intervalo de probabilidades P , con un intervalo de $P_i = 0,1$ hasta $P_f = 0,5$, para diferentes $Task(Threads)$, (b) se muestran los resultados de ejecución del número de clústeres encontrados en el sistema 3D.

En las imágenes de la figura 2.18(a) se muestran la distribución de las tareas ($Task$) en un rango de probabilidad de $P_i = 0,1$ hasta $P_f = 0,5$ con iteraciones i que van desde 0 hasta 9. En la figura de la imagen 2.18(b) podemos observar los resultados de la calculación. Donde se indica si existe o no percolaciones el rango de probabilidad asignada ($Percolation = True$ at $p = 0,50$), clúster independientes que no tienen contacto con el clúster infinito ($nNull = 21805$), cilindros independientes ($Isolatedcylsn1 = 73$), numero de cilindros en el cluster infinito ($Coupled cyls(in all clusters)n2 = 23065$), total de cilindros en el sistema ($Total : nNull + n0 + n1 + n2 = 45000, useNNullNodes = False$), el tiempo de calculación estimado para la tarea #9 ($Donetask \#9.Watch : 694387msec., = 11min,694sec,387msec.$), el punto de probabilidad critica donde se encontró percolación ($pCritical = 0,100$) y el tiempo final de calculación ($Totaltime : 12min. = 705sec.$).

2.10. Resultados

Realizamos pruebas variando el grosor de los radios de los nanotubos en el orden de nanómetros, recordando que un nanómetro $1 nm = 1 \times 10^{-9}m$. Se consideró un valor fijo para μ_R y μ_H con diferentes variaciones para σ_r y σ_h , en donde: μ_R y μ_H son el grosor ($r =$ radio) y longitud del nanotubo ($h =$ altura/longitud) respectivamente y σ_r y σ_h son las desviaciones que hay correspondientes a la dis-

tribución Gaussiana, al no poder crear nanotubos uniformes del mismo grosor y longitud.

Es importante conocer qué volumen ocupa el clúster infinito entre todos los nanotubos generados en la rejilla, ya que este volumen está directamente asociado al umbral de percolación o probabilidad crítica. Para esto, es necesario dividir el volumen total de nanotubos que existen en todo el sistema o rejilla, entre el volumen de los nanotubos que forman el clúster infinito mediante la siguiente ecuación:

$$P(p) = \frac{\sum_1^L(V)}{\sum_1^C(V)} \quad (2.14)$$

Dónde: L =Número de cilindros en el **sistema**. C =Número de cilindros en el **cluster inf**. V =Volumen del **cilindro**. Generamos varias calculaciones, distribuyendo los valores un rango de probabilidad P de 0.02 a 0.2 simulada en una rejilla tridimensional de tamaño L igual a $50 \times 50 \times 50$ nm. En la tabla 2.1, podemos observar que existe percolación a partir del valor de probabilidad de 0.052222222, que se encuentra sombreada de azul en la tabla, para los valores posteriores a este, existen clústeres infinitos. Los valores utilizados para las dimensiones del cilindro fueron:

N	Probabilidad	$P(p)$
1	0.01	0.00422535211267606
2	0.031111111	0.0726643598615917
3	0.052222222	0.334449884050502
4	0.073333333	0.686885549990725
5	0.094444444	0.844773960216998
6	0.115555556	0.952211338676165
7	0.136666667	1.00441944556047
8	0.157777778	1.00013037809648
9	0.178888889	1.00600046157397
10	0.2	1.01206854936037

Tabla 2.1: Se muestran los resultados hechos para un rango de probabilidad de 0.01 hasta 0.2 con 10 probabilidades distribuidas en el rango.

longitud del cilindro: $\mu_H=20$ nm, grosor del cilindro: $\mu_R=0.1$ nm, y con desviaciones que pueden sufrir las longitudes es de $\sigma_h=1.4$ en su longitud y $\sigma_r=0.025$ en sus grosores, donde nm es la unidad básica para 1 nanómetro [15]. En la figura 2.19, también se puede apreciar el volumen del clúster para cada prueba que está asociada a la conductividad eléctrica S [16]. Podemos observar que conforme aumenta la probabilidad, el volumen en el clúster aumenta. En la figura 2.19 podemos observar que, a partir del valor 0.05 de probabilidad, comienzan existir clústeres infinitos, es decir, a partir de ese valor existe percolación: todos los materiales a los que se les asigne esa probabilidad serán conductores, y los menores a ellos no conducirán la energía eléctrica [23]. Para tener una noción más precisa de los resultados, aplicamos a estos datos un ajuste para calcular la aproximación.

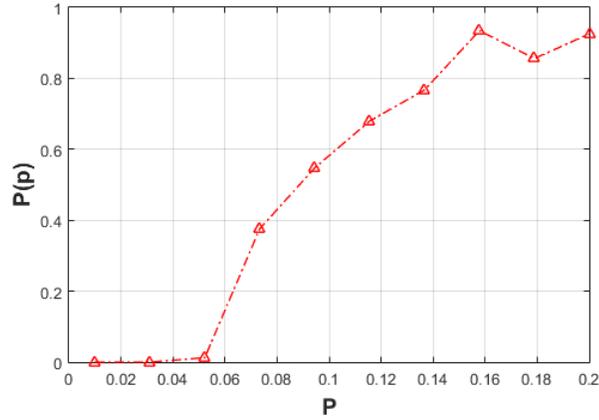


Figura 2.19: Gráfico que muestra el comportamiento de $P(p)$ volumen de los cilindros en la rejilla para 10 calculaciones, entre las probabilidades: 0.01 hasta 0.2

2.11. Ajuste de los resultados

Cuando se requiere el análisis de datos experimentales, el ajuste al modelo físico es de gran importancia. Frecuentemente, se usa el ajuste lineal, pero esto solo es factible si los fenómenos en estudio aceptan una solución lineal. Es por ello que se realizó un ajuste numérico de los datos utilizando la función *Fit* en Matlab a los resultados anteriores determinados por la ecuación:

$$P(p) = b(p - pc)\beta \tag{2.15}$$

Dónde: b , pc y β , son constantes. En la figura 2.20 podemos apreciar la aplicación de la función *Fit* a los valores de la gráfica de la figura 2.19.

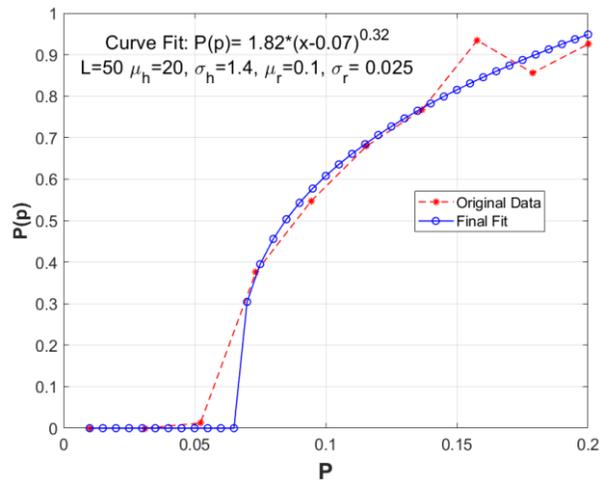


Figura 2.20: Gráfica con datos ajustado de la figura 2.19, en un rango de probabilidad $P=0.1$ a 0.2, con valores de: $L = 50$, $\mu_H=15$, $\sigma_h=1.4$, $\mu_R=0.1$ y $\sigma_r=0.025$. aplicando la ecuación: $P(p) = b(p - pc)\beta$ equivalente a la curva *Fit*: $p(p) = 1,82(p - 0,07)0,32$.

2.12. Resultados variando la longitud de los nanotubos

En nuestro estudio consideramos diferentes medidas de longitudes para las nanoestructuras, con una pequeña desviación para cada caso, en la figura 19 podemos observar una agrupación de imágenes (a – i) con valores: $L = 50$, con longitudes $\mu_H=15-23$, $\mu_R=0.1$ y $\sigma_r=0.025$ para los radios de los nanotubos, En donde el valor μ_H va desde 15 hasta 23 nanómetros aplicando una desviación estándar para todos los casos de $\sigma_h=1.4$ en sus longitudes. En la figura 2.21, se muestra un

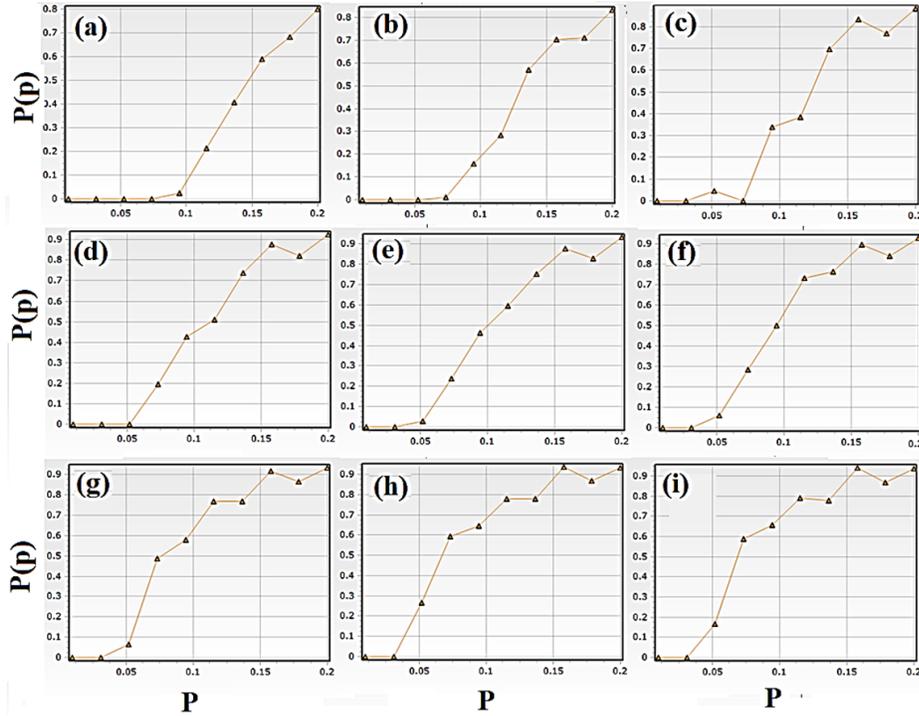


Figura 2.21: Distribución de longitudes en las nanoestructuras; (a) $\Rightarrow 15 \text{ nm}$, (b) $\Rightarrow 16 \text{ nm}$, (c) $\Rightarrow 17 \text{ nm}$ (i) $\Rightarrow 25 \text{ nm}$, con incrementos de 1 nm ascendentemente para cada caso y con una desviación $\sigma_H = 1,4$

aumento en μ_H de 15 a 23 en donde vemos que $RB(\text{volumen del clúster infinito})$, va disminuyendo en su probabilidad crítica mientras mayor longitud tienen los nanotubos. En la figura 2.22 se muestran los ajustes de los datos. La relación de las gráficas anteriores la podemos observar en la tabla 2.2, podemos observar que para cada valor de μ_h , obtenemos un valor de probabilidad crítica P_{cr} , que disminuye conforme aumentamos la longitud del nanotubo, esto es lógico, pues mientras más largo y grueso sea el nanotubo, hay más posibilidad de generar intercepción entre los nanotubos y por lo tanto clústeres infinitos, recordemos que la probabilidad crítica, entre más pequeña sea, será más el ahorro que podemos tener en cuanto a materiales, es por ello que es bastante importante encontrar una probabilidad crítica satisfactoria para μ_h y μ_r , para este trabajo enfocamos nuestro estudio hacia la longitud del cilindro μ_h y para algunas desviaciones σ_h .

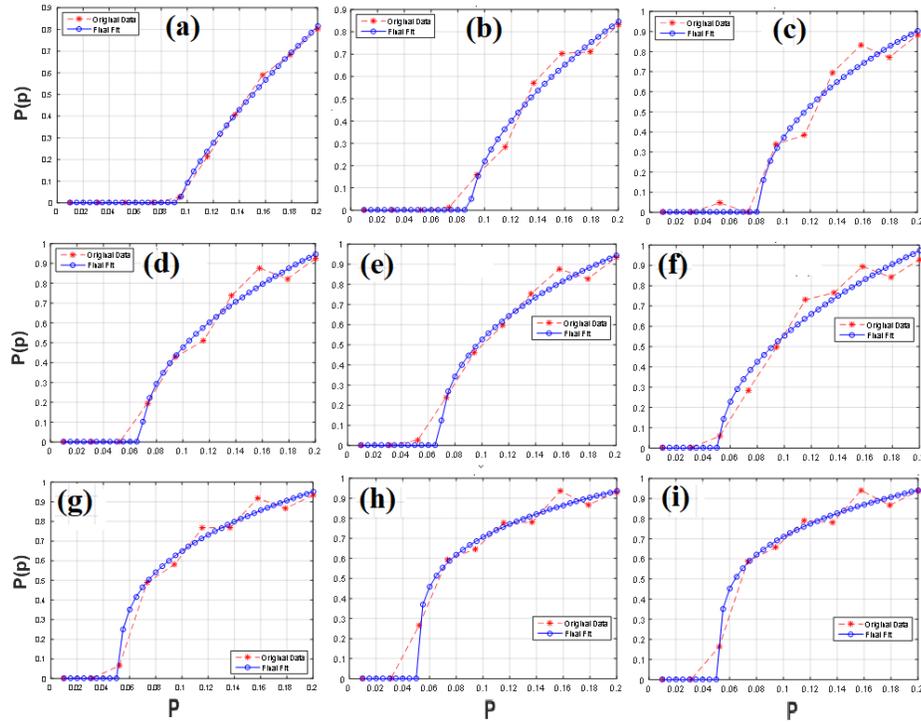


Figura 2.22: Ajuste de los datos aplicando $P(p) = b(p-p_c)\beta$ para los casos de la figura 2.21(a-i)

De la tabla 2.2 podemos observar que en los primeros valores existen fluctuaciones en la curva que pueden subir y bajar su valor a partir del valor $\mu_H = 20$ nm se puede ver una estabilidad en los valores con una probabilidad crítica establenente descendente. Esta relación y comportamiento de los valores de la tabla 2.2 los podemos observar en la imagen de la figura 2.23, donde se plasman los valores de la tabla 2.2 para un desplazamiento estándar de $\sigma_H = 1,4$. En la figura 2.23, tenemos una dependencia de las diferentes longitudes de nanotubos μ_H vs P_{cr} , podemos observar que mientras más largos sean las longitudes de los nanotubos, la probabilidad crítica tiende a caer.

μ_H	L	σ_r	μ_R	σ_r	P_{cr}
15 nm	50	1.4	0.1	0.25	0.084
16 nm	50	1.4	0.1	0.25	0.09
17 nm	50	1.4	0.1	0.25	0.091
18 nm	50	1.4	0.1	0.25	0.072
19 nm	50	1.4	0.1	0.25	0.073
20 nm	50	1.4	0.1	0.25	0.066
21 nm	50	1.4	0.1	0.25	0.052
22 nm	50	1.4	0.1	0.25	0.052
23 nm	50	1.4	0.1	0.25	0.052
24 nm	50	1.4	0.1	0.25	0.052
25 nm	50	1.4	0.1	0.25	0.052

Tabla 2.2: Se muestra el comportamiento de P_{cr} , para varias pruebas en donde se varia la longitud del cilindro $\rightarrow h$ de 15 a 25 nanómetros, para una desviación $\mu_H = 1.4$.

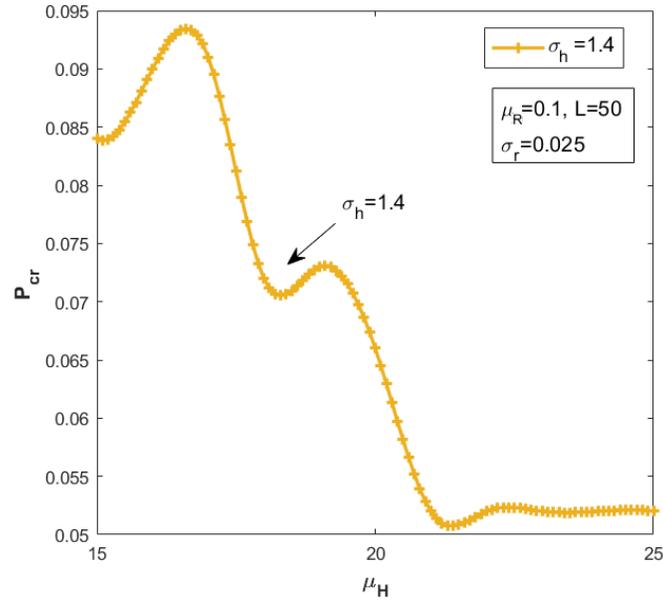


Figura 2.23: Se muestra el comportamiento de P_{cr} , para varias pruebas en donde se varia la longitud del cilindro μ_H de 15 a 25 nanómetros, para una desviación $\sigma_H = 1.4$

2.13. Resultados para varios desplazamientos σ_h

Realizamos algunas pruebas variando las desviaciones para el rango de μ_H de 15 a 25 nanómetros y con otras desviaciones de: 0.01, 0.05, 0.4, 1, 1.2 y 1.4 nanómetros, en el gráfico de la figura 2.24 podemos observar el comportamiento de estas desviaciones para rangos $\mu_H = 15$ a 25 nanómetros. Podemos apreciar que para los valores de 0.001, 0.05, 0.4 los valores convergen aproximadamente en un rango de probabilidad crítica P_{cr} de 0.09 a 0.1 cuando los nanotubos miden 15 nanómetros a diferencia de las desviaciones mayores a 1, donde los valores de P_{cr} , caen por

debajo de 0.09 o por encima de 0.1. A partir del valor $\mu_H = 22$ y valores mayores

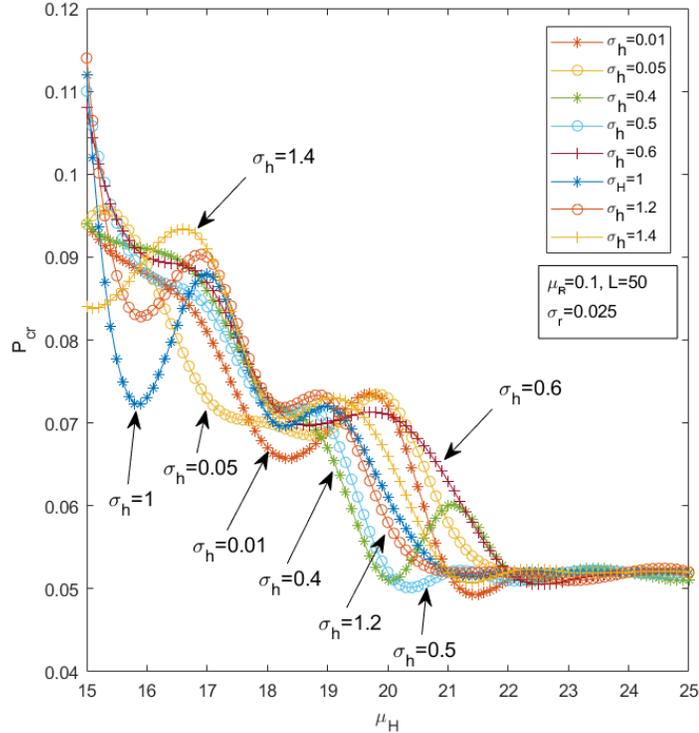


Figura 2.24: Se muestran los resultados graficados para los rangos de $\mu_H = 12$ a 25 nanómetros pero con 6 desviaciones diferentes: $\sigma_h = 0.01, 0.05, 0.4, 0.5, 0.6, 1, 1.2$ y 1.4, con grosores de radio $\mu_R = 0.1$ y $\sigma_r = 0.025$ en una rejilla de dimensión $L = 50 \text{ nm}$ por cada lado del sistema cúbico 3D para las 8 curvas.

a este podemos, observar que la probabilidad crítica se estabiliza sin importar la desviación que se les aplique. También observamos que las fluctuaciones en las curvas son muy similares en picos para los puntos en el eje μ_H . A partir de los resultados obtenidos del último gráfico podemos decir que existe cierta tendencia para alcanzar una probabilidad crítica pequeña cuando los nanotubos son largos, aproximadamente mayores o iguales a $\mu_H = 22$ nanómetros y que las desviaciones no influyen cuando los nanotubos tienen este valor, pero si influyen cuando son menores a éste.

El P_{cr} más estable para todas las desviaciones en nuestras pruebas es aproximadamente de 0.05 para nanotubos mayores o iguales a 22 nanómetros y como consecuencia en este rango no habrá fallas en la conductividad eléctrica S , ya que el número de nanotubos y su volumen es bastante grande como para poder formar clústeres infinitos que conduzcan la electricidad de un punto a otro del medio. Mientras que para longitudes menores a 22 la probabilidad crítica se alcanza en un valor más alto. Esto significa que es preferible ocupar nanotubos largos para las desviaciones consideradas en la figura 2.24.

2.14. Resultados variando el radio de los nanotubos con desplazamiento σ_r .

Es importa también analizar las desviaciones que pueden llegar a tener los nanotubos en sus grosores, ya que al igual que en el caso de las longitudes este puede llegar a tener significancia en los contactos para formar un clúster infinito, es por ello que consideramos diferentes desviaciones σ_r . En la tabla 2.3, se muestran los resultados de su probabilidad crítica (P_{cr}), con desviaciones (σ_r) que van desde 0.001 hasta 0.15, con tamaño de la rejilla de $L = 50x50x50$ nanómetros, longitud del cilindro $\mu_H=15$ y su desviación $\sigma_h=1.8$ y con un radio $\mu_R=0.2$. En el grafico 2.25, podemos observar que hay un cambio en la probabilidad crítica cuando el error o desviación σ_r es más grande cuando se pretende crear nanotubos de 0.2 nanómetros de grosor. Por su parte, la probabilidad crítica es estable cuando existe una desviación pequeña hasta el valor de 0.009 cuando el error o desviación comienza a ser mayor.

σ_r	P_{cr}
0.001	0.051
0.002	0.051
0.003	0.051
0.004	0.051
0.005	0.051
0.006	0.051
0.007	0.051
0.008	0.051
0.009	0.052
0.01	0.052
0.011	0.052
0.012	0.052
0.013	0.052
0.014	0.052
0.014	0.052

Tabla 2.3: Se muestran los resultados hechos para un rango de probabilidad de 0.01 hasta 0.2 con 10 probabilidades distribuidas en el rango.

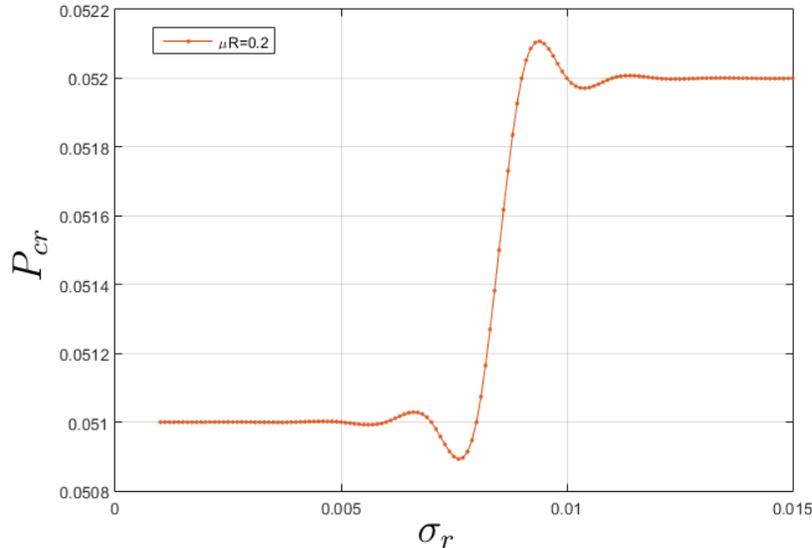


Figura 2.25: Comportamiento de los valores de P_{cr} , para un intervalo de desviaciones de $\sigma_r=$ 0.001 hasta $\sigma_r= 0.015$ cuando: $\mu_H = 15$, $\sigma_h = 1.8$ y $\mu_R = 0.2$.

2.15. Resultados para varios grosores en los nanotubos μ_R

Por último realizamos varias pruebas en este mismo intervalo de variación ($\sigma_r = 0.001$ hasta $\sigma_r = 0.015$) con valores fijos para $\mu_H = 15$ y $\sigma_h = 1.8$ pero considerando varios $\mu_R = 0.1, 0.2, 0.3, 0.5$ y 1 . El comportamiento para estas variaciones se muestra en la gráfica de la figura 2.26.

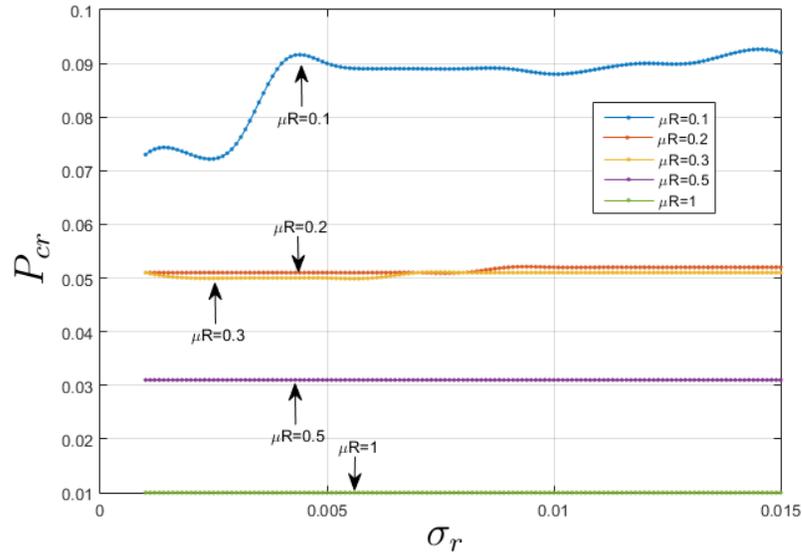


Figura 2.26: Comportamiento de los valores de P_{cr} , para un intervalo de desviaciones de $\sigma_r = 0.001$ hasta $\sigma_r = 0.015$ cuando: $\mu_H = 15$, $\sigma_h = 1.8$ y $\mu_R = 0.2$.

Podemos observar, de la figura 2.26, que, para diferentes valores de μ_R , la probabilidad crítica P_{cr} se mantiene más estable cuando el valor de los grosores de los cilindros es mayor, sin importar la desviación o error que se le asigne al generar los nanotubos en la simulación, esto a partir de grosores de radio aproximadamente iguales o mayores a 0.5 nanómetros.

2.16. Conclusiones parciales

A partir de los resultados obtenidos del último gráfico podemos decir que existe cierta tendencia para alcanzar una probabilidad crítica pequeña cuando los nanotubos son largos, aproximadamente mayores o iguales a $\mu_H = 22$ nanómetros y que las desviaciones no influyen cuando los nanotubos tienen este valor, pero si influyen cuando son menores a este. El P_{cr} más estable para todas las desviaciones en nuestras pruebas es aproximadamente de 0.05 para nanotubos mayores o iguales a 22 nanómetros y como consecuencia en este rango no habrá fallas en la conductividad eléctrica S , ya que el número de nanotubos y su volumen es bastante grande como para poder formar clústeres infinitos que conduzcan la electricidad

de un punto a otro del medio.

utilizando la distribución normal y con base en los resultados, pudimos corroborar en nuestras pruebas que la desviación σ_r aplicada para efectos de crear nanotubos con diferentes desviaciones en sus grosores afecta a la percolación de las estructuras y por lo tanto en la conducción eléctrica en un sistema de nanotubos desordenados de nanotubos de carbono.

EL MÉTODO DE DIFERENCIAS FINITAS EN EL DOMINIO DEL TIEMPO (FDTD).

3.1. Introducción

El método FDTD, es un método computacional que resuelve de forma numérica problemas que involucran fenómenos electromagnéticos con una solución muy aproximada a la solución analítica. La discretización de soluciones analíticas a digitales utilizando los recursos de la computadora generalmente es muy efectiva, más sin embargo también trae consigo soluciones de precisión finita.

3.1.1. Precisión finita

El caso más simple es el de sumar 3 veces $1/3 \rightarrow 1/3+1/3+1/3=1$. Esto es correcto para la lógica humana más sin embargo un computador debe discretizar primero los valores de $1/3$, es decir $1/3=0.33333\dots$, entonces el resultado de sumar los valores discretos será: $0.999999999\dots$ con una pérdida de precisión finita. Mas sin embargo este error de precisión se puede minimizar mientras más decimales añadimos al valor numérico, en términos computacionales se necesitaría más espacio de memoria RAM para almacenar los decimales necesarios para llegar a un resultado aproximado al analítico y dado que el aprovechamiento de las capacidades tecnológicas aumentan considerablemente, el método FDTD se ha vuelto convenientemente apropiado para nuestro caso de estudio.

Algunos de los fenómenos electromagnéticos en las que puede emplearse el método FDTD son en la aplicación de la Ley de Coulomb y campo eléctrico que estudia la fuerza eléctrica en partículas cargadas, la densidad de flujo eléctrico en materiales con partículas cargadas eléctricamente, campos eléctricos estáticos; cuando se considera una distribución de cargas como una colección de cargas puntuales, para calcular el gradiente y al divergencia para describir los campos eléctricos

y magnéticos. El método FDTD provee algoritmos para calcular las condiciones de la frontera del campo electromagnético [18]. Debido a que la densidad de flujo eléctrico solo comienza o termina con la carga, el componente normal D solo puede cambiar en la interfaz si hay una carga en la malla numérica de simulación, es decir, la carga superficial está presente. Las condiciones de contorno en el componente tangencial del campo eléctrico se pueden determinar integrando el campo eléctrico sobre un circuito cerrado que es esencialmente un rectángulo que encierra la malla 1D, 2D o 3D de simulación numérica [19–33]. Otro fenómeno físico que se puede comprender muy bien usando el método FDTD es el análisis y simulación de las ecuaciones de onda para posteriormente desacoplarlas mediante las ecuaciones de rizo.

3.2. FDTD en 1D

El método FDTD es uno de los métodos más comunes para calcular las ecuaciones de onda para fenómenos electromagnéticos. El método FDTD puede solucionar problemas complicados y de precisión muy finita pero generalmente es computacionalmente costoso, ya que pueden requerir una gran cantidad de memoria, procesamiento en RAM y tiempo de calculación. EL método FDTD encaja libremente en la categoría de técnicas de resonancia y frecuencias en algún orden de longitud de onda aunque estas longitudes sean más pequeñas que el objeto que lo produce (nano-partículas).

3.2.1. Algoritmo de Yee

El algoritmo FDTD propuesto por primera vez por Kane Yee en 1966 emplea diferencias centrales de segundo orden. El algoritmo se puede resumir de la siguiente manera:

1. Reemplaza todas las derivadas en las leyes de Ampere y Faraday con diferencias finitas. Discretiza el espacio y el tiempo para que los campos eléctricos y magnéticos se escalonen tanto en el espacio como en el tiempo.
2. Resuelve las ecuaciones de diferencia resultantes para obtener *ecuaciones de actualización* que expresen los campos futuros (desconocidos) en términos de campos pasados (conocidos).
3. Evalúa los campos magnéticos un paso de tiempo hacia el futuro para que se conozcan ahora (efectivamente se convierten en campos pasados).
4. Repite los dos pasos anteriores hasta que los campos se hayan obtenido durante la duración deseada.

Sin embargo, al desarrollar el conjunto completo de ecuaciones tridimensionales sería excesivo y, por lo tanto, el algoritmo se presentará primero en una dimensión.

3.2.2. Actualización de ecuaciones en 1D

Consideramos un espacio unidimensional donde solo hay variaciones en la dirección x . Suponga que el campo eléctrico solo tiene un componente z . En este caso, la ley de Faraday se puede escribir:

$$-\mu \frac{\partial \mathbf{E}}{\partial t} = \nabla \times \mathbf{E} = \begin{vmatrix} \hat{a}_x & \hat{a}_y & \hat{a}_z \\ \frac{\partial}{\partial t} & 0 & 0 \\ \frac{\partial}{\partial x} & E_y & 0 \end{vmatrix} = -\hat{a}_y \frac{\partial E_z}{\partial x} \quad (3.1)$$

Por lo tanto, H_y debe ser el único componente no nulo del campo magnético que varía en el tiempo. Dado que el lado derecho de esta ecuación tiene solo un componente y , el campo magnético puede tener componentes distintos de cero en las direcciones x y z pero deben ser estáticos. No nos ocuparemos de los campos estáticos aquí. La ley de Ampere se puede escribir

$$\epsilon \frac{\partial \mathbf{H}}{\partial t} = \nabla \times \mathbf{E} = \begin{vmatrix} \hat{a}_x & \hat{a}_y & \hat{a}_z \\ \frac{\partial}{\partial t} & 0 & 0 \\ \frac{\partial}{\partial x} & 0 & E_z \end{vmatrix} = -\hat{a}_z \frac{\partial H_y}{\partial x} \quad (3.2)$$

Las ecuaciones escalares son:

$$\mu \frac{\partial H_y}{\partial t} = \frac{\partial E_z}{\partial x}, \quad (3.3)$$

$$\epsilon \frac{\partial E_z}{\partial t} = \frac{\partial H_y}{\partial x}, \quad (3.4)$$

La primera ecuación da la derivada temporal del campo magnético en términos de la derivada espacial del campo eléctrico de la ecuación (3.3). Por el contrario, la segunda ecuación da la derivada temporal del campo eléctrico en términos de la derivada espacial del campo magnético de la ecuación (3.4). Como se mostrará, la primera ecuación se usará para avanzar el campo magnético a tiempo, mientras que la segunda se usará para avanzar el campo eléctrico. Un método en el que se avanza un campo y luego el otro, y luego se repite el proceso, se conoce como método de salto de rana.

El siguiente paso es reemplazar las derivadas en 3.3 y 3.4 con diferencias finitas. Para hacer esto, el espacio y el tiempo deben ser discretizados. La siguiente notación se utilizará para indicar la ubicación donde se muestrean los campos en el espacio y el tiempo.

Donde Δ_x es el desplazamiento espacial entre los puntos de muestra y Δ_t es el desplazamiento temporal. El índice m corresponde al paso espacial, efectivamente la ubicación espacial, mientras que el índice q corresponde al paso temporal. Cuando se escribe como un superíndice q todavía representa el paso temporal. Al implementar algoritmos FDTD, veremos que los índices espaciales se usan como índices de matriz, mientras que el índice temporal, que es esencialmente un parámetro

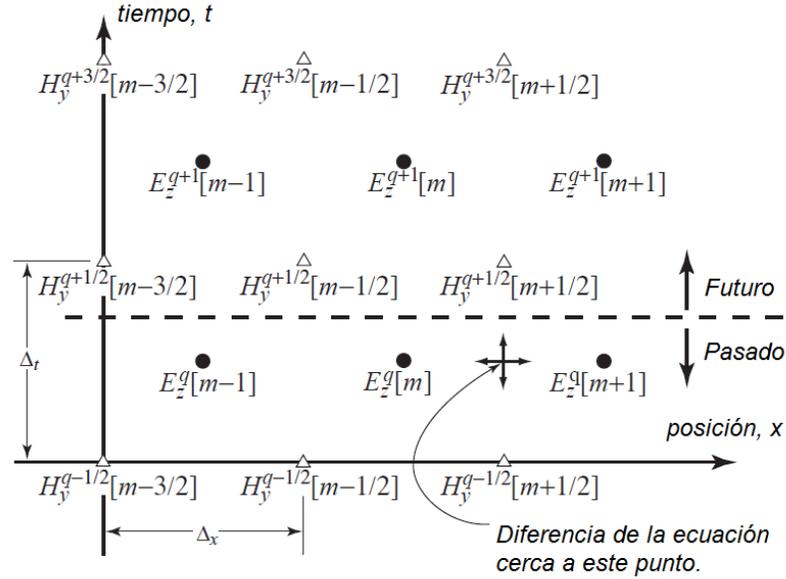


Figura 3.1: Disposición de los nodos de campo eléctrico y magnético en el espacio y el tiempo. Los nodos del campo eléctrico se muestran como círculos y los nodos del campo magnético como triángulos. El punto indicado es donde se expande la ecuación de diferencia para obtener una ecuación de actualización para H_y .

global, no se especifica explícitamente para cada ubicación de campo. Aunque solo tenemos una dimensión espacial, el tiempo puede considerarse como otra dimensión. Por lo tanto, esta es efectivamente una forma de problema bidimensional. Pero ¿Cómo deben organizarse los puntos de muestreo de campo eléctrico y magnético, también conocidos como nodos, en el espacio y el tiempo? La respuesta se muestra en la figura 3.1. Los nodos de campo eléctrico se muestran como círculos y los nodos de campo magnético como triángulos. Suponga que todos los campos debajo de la línea punteada son conocidos, se consideran del pasado, mientras que los campos arriba de la línea punteada son campos futuros y, por lo tanto, desconocidos. El algoritmo FDTD proporciona una forma de obtener los campos futuros de los campos pasados. Como se indica en la figura 3.1, considere la ley de Faraday en el punto espacio-tiempo $((m + 1/2)\Delta x, q\Delta t)$.

$$\mu \frac{\partial H_z}{\partial t} \Big|_{(m+1/2)\Delta x, q\Delta t} = \frac{\partial E_z}{\partial x} \Big|_{(m+1/2)\Delta x, q\Delta t} \quad (3.5)$$

La derivada temporal se reemplaza por una diferencia finita que involucra a $H_y^{q+\frac{1}{2}}[m+\frac{1}{2}]$ y $H_y^{q-\frac{1}{2}}[m+\frac{1}{2}]$ mientras que la derivada espacial se reemplaza por una diferencia finita que implica $E_z^q[m+1]$ and $E_z^q[m]$. Esto produce:

$$\mu \frac{H_y^{q+\frac{1}{2}}[m+\frac{1}{2}] - H_y^{q-\frac{1}{2}}[m+\frac{1}{2}]}{\Delta t} = \frac{E_z^q[m+1] - E_z^q[m]}{\Delta x}. \quad (3.6)$$

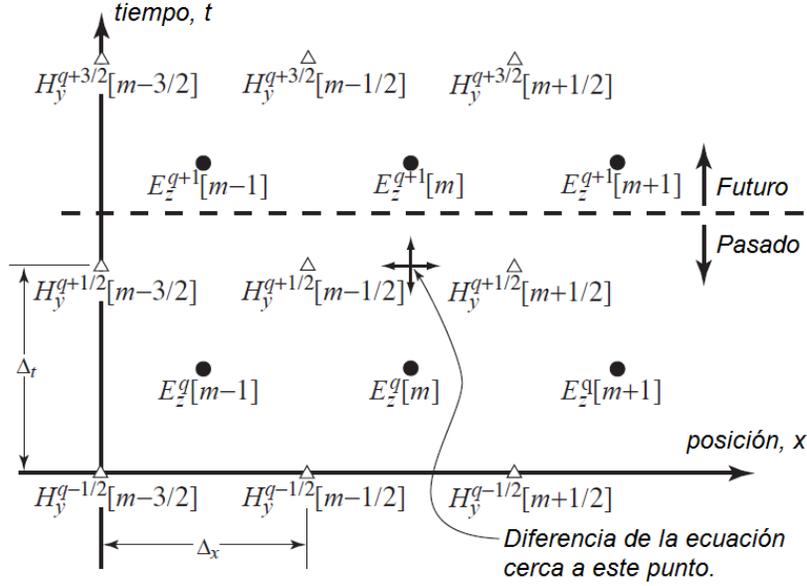


Figura 3.2: Espacio-tiempo después de actualizar el campo magnético. La línea divisoria entre valores futuros y pasados ha avanzado un paso medio temporal. El punto indicado es donde se escribe la ecuación de diferencia para obtener una ecuación de actualización para E_z .

Resolviendo esto para: $H_y^{q+\frac{1}{2}}[m+\frac{1}{2}]$ rendimientos

$$H_y^{q+\frac{1}{2}}[m+\frac{1}{2}] = H_y^{q+\frac{1}{2}}[m+\frac{1}{2}] + \frac{\Delta t}{\mu \Delta x} (E_z^q[m+1] - E_z^q[m]). \quad (3.7)$$

Esto se conoce como una ecuación de actualización, específicamente la ecuación de actualización para el campo H_y . Es una ecuación genérica que se puede aplicar a cualquier nodo de campo magnético. Muestra que el valor futuro de H_y depende solo de su valor permeable y de los campos eléctricos vecinos. Después de aplicar la ecuación (3.7) a todos los nodos de campo magnético, la línea divisoria entre valores futuros y pasados ha avanzado medio paso. La cuadrícula espacio-tiempo aparece así como se muestra en la Figura 3.2, que es idéntica a la Figura 3.1, excepto por el avance de la línea divisoria pasado/futuro.

Ahora consideramos la ley de Ampere aplicado al espacio tiempo $m\Delta x, (q+\frac{1}{2}\Delta t)$, como se indica en la figura 3.2.

$$\epsilon \frac{\partial E_z}{\partial t} \Big|_{m\Delta x, (q+1/2)q\Delta t} = \frac{\partial H_y}{\partial x} \Big|_{m\Delta x, (q+1/2)q\Delta t} \quad (3.8)$$

Se reemplaza la derivada temporal a la izquierda con una diferencia finita que involucra $E_z^{q+1}[m]$ and $E_z^q[m]$ y reemplazando la derivada espacial a la derecha con una diferencia finita que involucra $H_y^{q+\frac{1}{2}}[m+\frac{1}{2}]$ y $H_y^{q+\frac{1}{2}}[m-\frac{1}{2}]$ rendimientos

$$\epsilon \frac{E_z^{q+1}[m] - E_z^q[m]}{\Delta t} = \frac{H_y^{q+\frac{1}{2}}[m+\frac{1}{2}] - H_y^{q+\frac{1}{2}}[m-\frac{1}{2}]}{\Delta x} \quad (3.9)$$

resolviendo para: $E_z^{q+1}[m] - E_z^{q+1}[m]$

$$E_z^{q+1}[m] - E_z^{q+1}[m] = E_z^{q+1}[m] + \frac{\epsilon \Delta t}{\Delta x} \left(H_y^{q+\frac{1}{2}} \left[m + \frac{1}{2} \right] - H_y^{q+\frac{1}{2}} \left[m - \frac{1}{2} \right] \right) \quad (3.10)$$

Un espacio FDTD unidimensional que muestra el desplazamiento espacial entre los campos magnético y eléctrico. La ecuación 3.10 es la ecuación de actualización para el campo E_z . Los índices en esta ecuación son genéricos que la misma ecuación se cumple para cualquier nodo E_z . Después de aplicar la ecuación 3.10 a cada nodo de campo eléctrico en la red, la línea divisoria es conocida y lo desconocido avanza otro paso medio-temporal. Volviendo a la situación representada en la figura 3.2, los campos futuros más cercanos a la línea divisoria entre el futuro y el pasado son los campos magnéticos. Se actualizarían nuevamente, luego los campos eléctricos son actualizados y así sucesivamente. A menudo es conveniente representar los coeficientes de actualización $\Delta t/\epsilon \Delta x$ y $\Delta t/\mu \Delta x$ en términos de relación de cuánto puede propagarse la energía en un solo paso temporal al paso espacial. La velocidad máxima a la que viaja la energía electromagnética es la velocidad de la luz en el espacio libre $c = 1/\sqrt{\epsilon_0 \mu_0}$ y por lo tanto la distancia máxima que puede viajar en un paso de tiempo es $c \Delta t$. La relación $c \Delta t/\Delta x$ a menudo se llama el número de Courant que etiquetamos como S_c . Juega un papel importante en la determinación de la estabilidad de una simulación (de ahí el uso de S) y se considerará más adelante. $\mu = \mu_r \mu_0$ y $\epsilon_r \epsilon_0$, son los coeficientes de las ecuaciones 3.10 y 3.7 se pueden escribir como:

$$\frac{1}{\epsilon} \frac{\Delta t}{\Delta x} = \frac{1}{\epsilon_r \epsilon_0} \frac{\sqrt{\epsilon_0 \mu_0}}{\sqrt{\epsilon_0 \mu_0}} \frac{\Delta t}{\Delta x} = \frac{\sqrt{\epsilon_0 \mu_0}}{\sqrt{\epsilon_0 \mu_0}} \frac{c \Delta t}{\Delta x} = \frac{1}{\epsilon_r} \sqrt{\frac{\mu_0}{\epsilon_0}} \frac{c \Delta t}{\Delta x} = \frac{\eta_0}{\epsilon_r} \frac{c \Delta t}{\Delta x} = \frac{\eta_0}{\epsilon_r} S_c \quad (3.11)$$

$$\frac{1}{\mu} \frac{\Delta t}{\Delta x} = \frac{1}{\mu_r \mu_0} \frac{\sqrt{\epsilon_0 \mu_0}}{\sqrt{\epsilon_0 \mu_0}} \frac{\Delta t}{\Delta x} = \frac{\sqrt{\epsilon_0 \mu_0}}{\sqrt{\epsilon_0 \mu_0}} \frac{c \Delta t}{\Delta x} = \frac{1}{\mu_r} \sqrt{\frac{\epsilon_0}{\mu_0}} \frac{c \Delta t}{\Delta x} = \frac{1}{\mu_r \eta_0} \frac{c \Delta t}{\Delta x} = \frac{1}{\mu_r \eta_0} S_c \quad (3.12)$$

Donde $\eta_0 = \sqrt{\mu_0/\epsilon_0}$ es la característica de impedancia del espacio libre.

En las simulaciones FDTD hay restricciones sobre cuán grande puede ser un paso temporal. Si es muy grande., el algoritmo produce resultados inestables (es decir, los números obtenidos no tienen sentido y generalmente tienden rápidamente al infinito). En esta etapa no consideraremos un análisis riguroso de la estabilidad. Al pensar en la forma en que los campos se propagan en una cuadrícula FDTD, parece lógico que la energía no debería poder propagarse más allá de un paso espacial para cada paso temporal, es decir, $c \Delta t \leq \Delta x$. Esto se debe a que en el algoritmo FDTD cada nodo solo afecta a sus vecinos más cercanos. En un ciclo completo de actualización de los campos, la mayor posible propagación de una perturbación es de un solo paso espacial. Resulta que la relación óptima para el número de Courant (en términos de minimizar el número errores) es también la

proporción máxima.
Inicialmente, usaremos:

$$S_c = \frac{c\Delta_t}{\Delta_x} = 1. \quad (3.13)$$

Al obtener por primera vez las ecuaciones de actualización para el algoritmo FDTD, es útil pensar en términos de espacio-tiempo. Sin embargo, tratar el tiempo como una dimensión adicional puede ser incómodo. Por lo tanto, en la mayoría de las situaciones es más conveniente pensar en términos de una sola dimensión espacial donde los campos eléctricos y magnéticos se compensan a medio paso espacial entre sí. Esto se representa en la figura 3.3. El desplazamiento temporal entre el campo eléctrico y el campo magnético siempre se entiende si se muestra explícitamente o no.

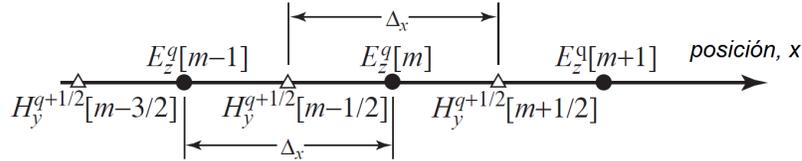


Figura 3.3: FDTD unidimensional que muestra el desplazamiento espacial entre los campos magnético y eléctrico..

3.2.3. Implementación de del algoritmo FDTD en una dimension.

Nuestro objetivo ahora es traducir las ecuaciones de actualización 3.7 a 3.10 en un programa de computadora utilizable. El primer paso es descartar, al menos en cierta medida, los superíndices: el tiempo es un parámetro global y se registrará en una sola variable entera. El tiempo no es algo de lo que deba preocuparse cada nodo. A continuación se muestra un código sencillo donde se utilizan los pasos importantes en la actualización de los campos eléctricos y magnéticos.

Al escribir un programa de computadora para implementar el algoritmo FDTD, uno no se molesta en tratar de construir un programa que explícitamente use compensaciones de la mitad. Los nodos se almacenan en matrices y, como es práctica habitual, los elementos de matriz individuales se especifican con índices enteros. Por lo tanto, el programa de computadora incorpora implícitamente el hecho de que los campos eléctricos y magnéticos se compensan al usar solo índices enteros para especificar la ubicación. Por ejemplo, suponga que se declaran dos matrices, ez e hy , que contendrán los campos E_z e H_y en 200 nodos:

```
double ex[200], hy[200], imp0=377.0;
```

La variable $imp0$ es la impedancia característica del espacio libre y se utilizará en la siguiente discusión (se inicializa a un valor de 377.0 en esta declaración). Las matrices ez e hy están compensadas entre sí por medio paso espacial accediendo a los valores del arreglo utilizando un índice entero. Un elemento de matriz ez

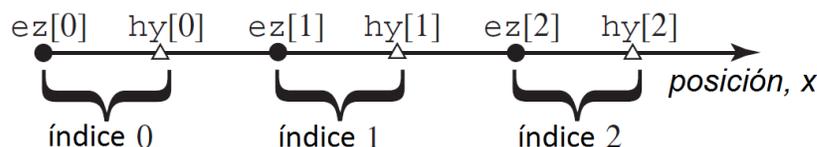


Figura 3.4: Espacio FDTD unidimensional que muestra la disposición espacial supuesta de los nodos de campo eléctrico y magnético en las matrices ez e hy . Existe un nodo de campo eléctrico a la izquierda del nodo de campo magnético con el mismo índice.

existe a la derecha o la izquierda de un elemento hy con el mismo índice (suponemos que la izquierda corresponde a valores decrecientes de x mientras que la derecha corresponde a valores crecientes), asumiremos que los nodos ez están a la izquierda de los nodos hy con el mismo índice. Esto se ilustra en la figura 3.4 donde $ez[0]$ está a la izquierda de $hy[0]$, $ez[1]$ está a la izquierda de $hy[1]$, y así sucesivamente. En general, cuando se usa una fuente Courier, por ejemplo, $hy[m]$, estamos considerando una matriz y cualquier desplazamiento de la mitad asociado con esa matriz se entiende implícitamente. Cuando se usa $H_y^{q+\frac{1}{2}}[m+\frac{1}{2}]$. Asumimos el campo en sí y las compensaciones se darán explícitamente.

Consideremos una simulación de una onda que se propaga en el espacio libre donde hay 200 nodos de campo eléctrico y magnético.

```

1  /* Bare-bones 1D FDTD simulation with a hard source. */
2  #include <stdio.h>
3  #include <math.h>
4  #define SIZE 200
5  int main ()
6  {
7      double ez[SIZE] = {0.}, hy[SIZE] = {0.}, imp0 = 377.0;
8      int qTime, maxTime = 250, mm;
9
10     /* do time stepping */
11     for (qTime = 0; qTime < maxTime; qTime++) {
12
13         /* update magnetic field */
14         for (mm = 0; mm < SIZE - 1; mm++)
15             hy[mm] = hy[mm] + (ez[mm + 1] - ez[mm]) / imp0;
16         /* update electric field */
17         for (mm = 1; mm < SIZE; mm++)
18             [mm] = ez[mm] + (hy[mm] - hy[mm - 1]) * imp0;
19         /* hardware a source node */
20         ez[0] = exp(-(qTime - 30.) * (qTime - 30.) / 100.);
21         printf("g\n", ez[50]);
22     } /* end of time-stepping */
23     return 0;
24 }

```

En las figuras de las imágenes 3.5 y 3.5, observamos los resultados del código anterior.

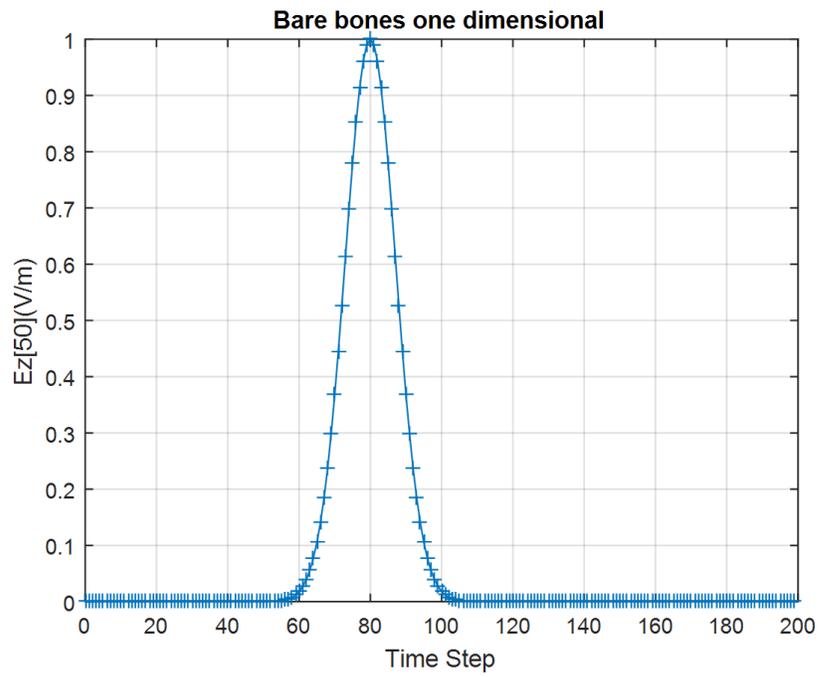


Figura 3.5: Gráfico que muestra el resultado del programa para 1D: Bare-bones, utilizando el método FDTD

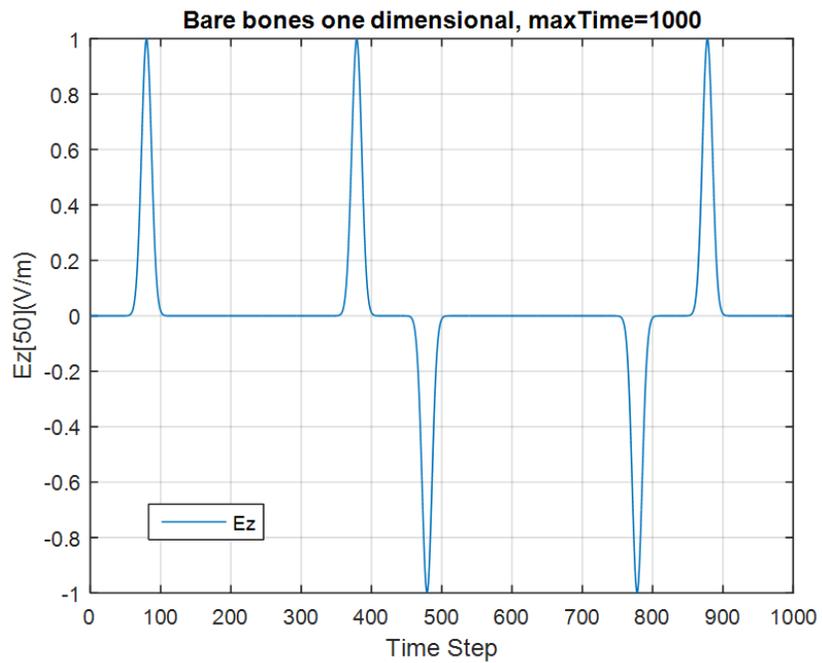


Figura 3.6: Salida generada por el programa anterior pero con `maxTime` establecido en 1000.

	n=0	n=1	n=2	n=3
m=0	ez(0,0)	ez(0,1)	ez(0,2)	ez(0,3)
m=1	ez(1,0)	ez(1,1)	ez(1,2)	ez(1,3)
m=2	ez(2,0)	ez(2,1)	ez(2,2)	ez(2,3)

Tabla 3.1: Se muestra las posiciones de las tablas a las que se harán referencia al arreglo bidimensional.

	n=0	n=1	n=2	n=3
m=0	ez[0]	ez[1]	ez[2]	ez[3]
m=1	ez[4]	ez[5]	ez[6]	ez[7]
m=2	ez[8]	ez[9]	ez[10]	ez[11]

Tabla 3.2: Se muestran los valores enteros en el arreglo.

3.3. Simulaciones del método FDTD en 2D (Simulaciones bidimensionales)

Una de las mayores ventajas del método FDTD es que no se limita a calcular solamente una dimensión, si no que ese posible escalar a dos y tres dimensiones, utilizando convenientemente los recursos de la computadora. La complejidad de otras técnicas numéricas a menudo aumenta sustancialmente a medida que aumenta el número de dimensiones. Con el método FDTD basta con comprender el algoritmo una dimensión para comprender los algoritmos en dos o tres dimensiones. Con el método FDTD se pueden implementar simulaciones bidimensionales con el campo magnético o el campo eléctrico ortogonal al plano normal de propagación. Las simulaciones multidimensionales requieren que pensemos en términos de una cuadrícula multidimensional y, por lo tanto, requerimos matrices multidimensionales para almacenar los campos.

3.3.1. Arreglos bidimensionales.

Para las matrices de dimensiones superiores a 1D, debemos especificar dos o más índices para dictar el elemento de interés. La traducción de múltiples índices a una sola dirección puede dejarse al compilador o esa traducción es algo que podemos hacer nosotros mismos. Ilustraremos las matrices en forma de tablas bidimensionales en forma de filas y columnas, en donde cada posición de la tabla está indicado por dos índices n=número de columna y m= número de fila. En la tabla 3.2, se muestran los índices de los arreglos m, n en las posiciones físicas de la memoria. Cada fila contiene elementos de columnas numéricas. La primera posición en el arreglo será ez(0,0)... hasta ez(2,3) en la memoria de almacenamiento temporal RAM. Nuevamente, en este ejemplo, cuando la fila se incrementa en uno, el índice de la matriz se incrementa en 4. (que es el número de columnas). Esto se debe al hecho de que estamos almacenando los elementos por filas. Se almacena una fila completa de valores, luego la siguiente fila, y así sucesivamente.

3.4. Polarización TM^z en dos dimensiones -2D-

Los problemas unidimensionales considerados hasta ahora suponían un componente z distinto de cero para campo eléctrico y una variación solo en la dirección x . Esto requería la existencia de un componente distinto de cero del campo magnético, se supone que el campo varía en las direcciones x e y , pero no en la dirección z . Desde el principio incluiremos la posibilidad de una conductividad magnética σ_m , mediante estos supuestos, la ley de Faraday se convierte en:

$$-\sigma_m \mathbf{H} - \mu \frac{\partial \mathbf{H}}{\partial t} = \nabla \times E = \begin{vmatrix} \hat{a}_x & \hat{a}_y & \hat{a}_z \\ \frac{\partial}{\partial x} & \frac{\partial}{\partial y} & 0 \\ 0 & 0 & E_z \end{vmatrix} = \hat{a}_x \frac{\partial E_z}{\partial y} - \hat{y} \frac{\partial E_z}{\partial x} \quad (3.14)$$

Dado que el lado derecho solo tiene componentes distintos de cero en las direcciones x e y , los componentes del campo magnético que varían en el tiempo solo pueden tener componentes x e y distintos de cero. El campo magnético es transversal a la dirección zy , por lo tanto, este se designa como el caso TM^z . La ley de Ampere se convierte en:

$$\sigma \mathbf{E} = \mathbf{H} + \epsilon \frac{\partial \mathbf{E}}{\partial t} = \nabla \times H = \begin{vmatrix} \hat{a}_x & \hat{a}_y & \hat{a}_z \\ \frac{\partial}{\partial x} & \frac{\partial}{\partial y} & 0 \\ H_x & H_y & 0 \end{vmatrix} = \hat{a}_z \left(\frac{\partial H_y}{\partial y} - \frac{\partial H_x}{\partial x} \right). \quad (3.15)$$

Las ecuaciones escalares obtenidas de las ecuaciones 3.14 y 3.15 son:

$$-\sigma_m H_x - \mu \frac{\partial H_x}{\partial t} = \frac{\partial E_z}{\partial y} \quad (3.16)$$

$$\sigma_m H_y + \mu \frac{\partial H_y}{\partial t} = \frac{\partial E_z}{\partial x} \quad (3.17)$$

$$\sigma E_z + \epsilon \frac{\partial E_z}{\partial t} = \frac{\partial H_y}{\partial x} - \frac{\partial H_x}{\partial y} \quad (3.18)$$

Ignorando los términos de conducción por un momento, la derivada temporal del campo magnético está relacionada con la derivada espacial del campo eléctrico y viceversa. La única diferencia del caso unidimensional es el componente de campo adicional H_x y las derivadas en la dirección y .

El espacio-tiempo ahora está discretizado para que las ecuaciones (3.16) - (3.18) se pueda expresar en términos de diferencias finitas. A partir de estas ecuaciones de diferencia, los campos futuros se pueden expresar en términos de campos pasados. Siguiendo la siguiente notación,

$$H_x(x, y, t) = H_x(m\Delta_x, n\Delta_y, q\Delta_t) = H_x^q[m, n], \quad (3.19)$$

$$H_y(x, y, t) = H_y(m\Delta_x, n\Delta_y, q\Delta_t) = H_y^q[m, n], \quad (3.20)$$

$$H_z(x, y, t) = H_z(m\Delta_x, n\Delta_y, q\Delta_t) = H_z^q[m, n]. \quad (3.21)$$

El índice m corresponde al paso espacial en la dirección x , mientras que el índice q corresponde al paso temporal. Además, el índice n representa el paso espacial en la dirección y .

El índice m corresponde al paso espacial en la dirección x , mientras que el índice q corresponde al paso temporal. Además, el índice n representa el paso espacial en la dirección y . Los tamaños de paso espacial en las direcciones x e y son Δx e Δy , respectivamente.

La matriz H_x sería $M \times (N - 1)$ mientras que la matriz H_y sería $(M - 1) \times N$. Los nodos del campo eléctrico caen en pasos espaciales enteros y los nodos del campo magnético se compensan medio paso espacial en la dirección x o y . Al igual que con una dimensión, se supone que el campo eléctrico existe en múltiplos enteros del paso temporal, mientras que ambos componentes de los campos magnéticos se compensan a medio paso de los campos eléctricos. Con esta disposición en mente, la aproximación de diferencia finita de la figura (8.3) expandida sobre el punto espacio-tiempo $(m\Delta_x, (n + 1/2)\Delta_y, q\Delta_t)$ es:

$$\begin{aligned} & -\sigma_m \frac{H_x^{q+\frac{1}{2}}[m, n + \frac{1}{2}] + H_x^{q-\frac{1}{2}}[m, n + \frac{1}{2}]}{2} \\ & -\mu \frac{H_x^{q+\frac{1}{2}}[m, n + \frac{1}{2}] + H_x^{q-\frac{1}{2}}[m, n + \frac{1}{2}]}{\frac{\Delta_t}{\Delta_y} (E_z^q[m, n + 1] - E_z^q[m, n])}. \end{aligned} \quad (3.22)$$

Esto puede resolverse para el valor futuro $H_x^{q+\frac{1}{2}}[m, n + \frac{1}{2}]$ en términos de los valores "pasados". La resultante la ecuación de actualización es:

$$\begin{aligned} H_x^{q+\frac{1}{2}}[m, n + \frac{1}{2}] &= \frac{1 - \frac{\sigma_m \Delta_t}{2\mu}}{1 + \frac{\sigma_m \Delta_t}{2\mu}} H_x^{q+\frac{1}{2}}[m, n + \frac{1}{2}] \\ & - \frac{1}{1 + \frac{\sigma_m \Delta_t}{2\mu}} \frac{\Delta_t}{\mu \Delta_x} (E_z^q[m, n + 1] - E_z^q[m, n]). \end{aligned} \quad (3.23)$$

Como fue el caso en una dimensión, los parámetros de material μ y σ_m son los que pertenecen al punto de evaluación dado. La ecuación de actualización para el componente y del campo magnético se obtiene mediante la aproximación de diferencia finita expandida sobre el punto espacio-tiempo $((m+1/2)\Delta_x, n\Delta_y, q\Delta_t)$.

la ecuación es:

$$H_x^{q+\frac{1}{2}} [m + \frac{1}{2}, n] = \frac{1 - \frac{\sigma_m \Delta t}{2\mu}}{1 + \frac{\sigma_m \Delta t}{2\mu}} H_x^{q+\frac{1}{2}} [m + \frac{1}{2}, n] - \frac{1}{1 + \frac{\sigma_m \Delta t}{2\mu}} \frac{\Delta t}{\mu \Delta x} (E_z^q [m + 1, n] - E_z^q [m, n]). \quad (3.24)$$

Los parámetros de material μ y σ_m son los que pertenecen al punto de evaluación dado. Tenga en cuenta que los nodos H_y están desplazados en el espacio desde los nodos H_x . Por lo tanto, los μ y σ_m que aparecen en las ecuaciones (3.23) y (3.24) no son necesariamente los mismos, incluso cuando m y n son iguales. Las líneas

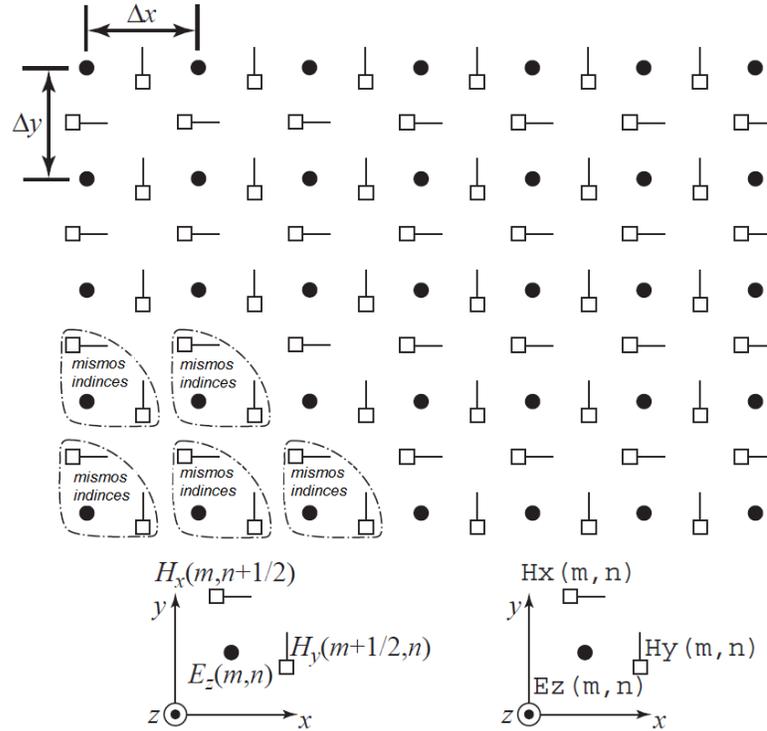


Figura 3.7: Disposición espacial de nodos de campo eléctrico y magnético para polarización TM^z . Los nodos de campo eléctrico se muestran como círculos y los nodos de campo magnético como cuadrados con una línea que indica la orientación del componente de campo.

discontinuas de la figura 3.7 en forma triangular indican agrupaciones de nodos que tienen los mismos índices de matriz. Por ejemplo, en la esquina inferior izquierda de la cuadrícula, todos los nodos tendrían índices en un programa de computadora de $(m = 0, n = 0)$. En este caso, el desplazamiento espacial de los campos se entiende implícitamente. Esta agrupación se repite en toda la cuadrícula. Sin embargo, los grupos en la parte superior de la cuadrícula carecen de un nodo H_x y los grupos

en el borde derecho carecen de un nodo H_y . El diagrama en la parte inferior izquierda de la figura indica nodos con sus desplazamientos dados explícitamente en los argumentos espaciales, mientras que el diagrama en la parte inferior derecha indica cómo se especificarían los mismos nodos en un programa de computadora donde los desplazamientos se entienden implícitamente.

3.4.1. Ejemplo de TM^z

Implementación FDTD de una cuadrícula TM^z con una fuente wavelet Ricker en el centro de la cuadrícula. No se ha implementado ABC, por lo que la simulación es efectivamente de un resonador.

```

1  /* TMz simulation with Ricker source at center of grid. */
2  #include "fdd-alloc.h"
3  #include "fdd-macro-tmz.h"
4  #include "fdd-protol.h"
5  #include "ezinc.h"
6  int main ()
7  {
8  Grid *g;
9  ALLOC_ID(g, 1, Grid); // allocate memory for Grid
10 gridInit(g); // initialize the grid
11 ezIncInit(g);
12 snapshotInit2d(g); // initialize snapshots
13 /* do time stepping */
14 for (Time = 0; Time < MaxTime; Time++) {
15 updateH2d(g); // update magnetic field
16 updateE2d(g); // update electric field
17 Ez(SizeX / 2, SizeY / 2) = ezInc(Time, 0.0); // add a source
18 snapshot2d(g); // take a snapshot (if appropriate)
19 } // end of time-stepping
20 return 0;
21 }

```

3.5. El límite TFSS para la polarización TM^z

Para una fuente distante que ilumina a un dispersador, no es factible discretizar el espacio que rodea la fuente, discretizar el espacio entre la fuente y el dispersador, y discretizar el espacio que rodea al dispersador.

Incluso si se pudiera obtener una computadora lo suficientemente grande que fuera capaz de almacenar todo ese espacio discretizado, uno simplemente no querría usar la cuadrícula FDTD para propagar el campo desde la fuente al dispersador. Tal esfuerzo sería lento, increíblemente ineficiente y sufriría artefactos numéricos innecesarios. En cambio, se debe discretizar el espacio que rodea el dispersador e introduzca el campo incidente a través de un límite de campo total/campo disperso. Cuando la fuente está distante del dispersor, el campo incidente es casi plano y, por lo tanto, restringiremos la consideración a las ondas planas incidentes.

En dos dimensiones, la cuadrícula se divide nuevamente en una región TF y una región SF. En este caso, el límite entre las dos regiones ya no es un punto. La Figura 3.9 muestra una cuadrícula TM^z con un límite rectangular TFSS. (El límite no tiene que ser rectangular, pero los detalles de implementación son más simples cuando el límite tiene lados rectos y, por lo tanto, nos limitaremos a los límites de TFSS que son rectangulares). En esta figura, la región TF está encerrada dentro del TFSS límite que se dibuja con una línea discontinua. La región SF es cualquier parte de la cuadrícula que está fuera de este límite. Los nodos que tienen un vecino en el otro lado del límite están encerrados en un

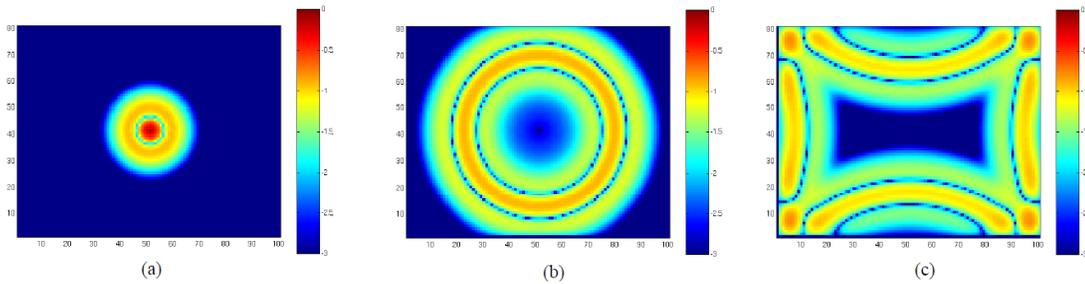


Figura 3.8: Disposición espacial de nodos de campo eléctrico y magnético para polarización TM^z . Los nodos de campo eléctrico se muestran como círculos y los nodos de campo magnético como cuadrados con una línea que indica la orientación del componente de campo.

rectángulo sólido con esquinas redondeadas. Tenga en cuenta que estos nodos encerrados son tangenciales al límite de TFSF (consideramos que el campo E_z , que apunta fuera de la página, es angular al límite si imaginamos que el límite se extiende a la tercera dimensión). Los campos que son normales al límite, como los nodos H_y a lo largo de la parte superior e inferior del límite TFSF, no tienen vecinos que estén al otro lado del límite (aunque el campo podría considerarse adyacente al límite).

En la implementación utilizada aquí, la región TF se define por los índices de los nodos de campo eléctrico primero y último que se encuentran en la región TF. Estos nodos se muestran en la Figura 3.9 donde el primer nodo es el que está en la esquina inferior izquierda y el último está en la esquina superior derecha. Tenga en cuenta que los campos eléctricos y los campos magnéticos con los mismos índices no están necesariamente en el mismo lado del límite. Por ejemplo, los nodos E_z en el lado derecho de la región TF tienen uno de sus nodos H_y vecinos en la región SF. Esto es cierto a pesar del hecho de que estos nodos H_y comparten el mismo índice x que los nodos E_z . En construcción particular de un límite TFSF, los campos eléctricos tangenciales al límite TFSF siempre están en la región TF. Estos nodos tendrán al menos un nodo de campo magnético vecino que se encuentra en la región SF. La corrección necesaria para obtener una actualización consistente de estos nodos de campo eléctrico implicaría agregar al campo incidente los campos magnéticos vecinos en el otro lado del límite de TFSF.

Por el contrario, los nodos de campo magnético que son tangenciales al límite de TFSF siempre están en la región SF. Estos nodos tendrán un nodo de campo eléctrico vecino que se encuentra en la región TF. Para obtener una actualización consistente de estos nodos de campo magnético implicaría restar el campo incidente del nodo de campo eléctrico en el otro lado del límite de TFSF.

Como en el caso unidimensional, para implementar el método TFSF, se debe conocer el campo incidente en cada nodo que tiene un vecino en el otro lado del límite TFSF. El campo del incidente debe conocerse en todos estos puntos y para cada paso de tiempo.

Sin embargo, el campo incidente no se propaga de la misma manera en la cua-

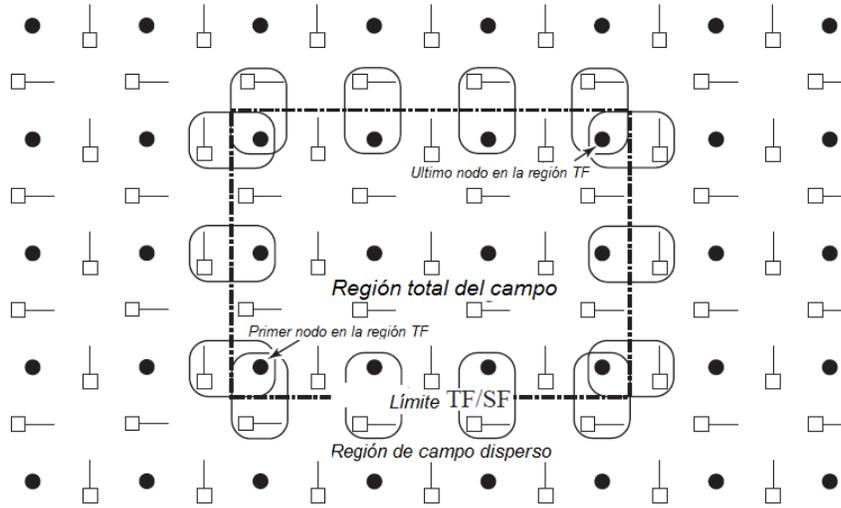


Figura 3.9: Representación de un límite de campo total / campo disperso en una cuadrícula TMz. El tamaño de la región TF está definido por los índices del primer y último nodo de campo eléctrico que están dentro de la región. En el lado derecho del límite, los nodos H_y con el mismo índice x como el "último" nodo estará en la región SF. De manera similar, en la parte superior de la cuadrícula, los nodos H_x con el mismo índice y que el último nodo estarán en la región SF. Por lo tanto, se debe prestar atención al componente de campo así como a los índices para determinar si un nodo está en la región SF o TF.

drícula FDTD de manera continua.

El cálculo de estas funciones es algo computacionalmente costoso, al menos en comparación con algunos cálculos algebraicos simples. Si las funciones trascendentales tienen que calcularse en numerosos puntos para cada paso de tiempo, esto puede imponer un costo computacional potencialmente significativo. Afortunadamente, siempre que la dirección de la propagación del campo incidente coincida con uno de los ejes de la cuadrícula, hay una manera de garantizar que el campo incidente coincida exactamente con la forma en que el campo incidente se propaga en la cuadrícula bidimensional FDTD. Además, el cálculo del campo del incidente se puede realizar de manera eficiente. La Figura 3.10 muestra la cuadrícula auxiliar 1D junto con la cuadrícula 2D. La base de las flechas verticales que apuntan desde la cuadrícula 1D a la cuadrícula 2D indica los nodos en la cuadrícula 1D de los cuales los nodos en la cuadrícula 2D obtienen el campo incidente (solo los nodos en la cuadrícula 2D adyacente al límite de TFSF requieren conocimiento del campo incidente). Como el campo incidente se propaga en la dirección $+x$, no hay campo H_x incidente. Por lo tanto, los nodos que dependen de un nodo H_x en el otro lado del límite TFSF no necesitan corregirse ya que $H_x^{inc} = 0$.

En la figura 3.10 La cuadrícula auxiliar 1D es impulsada por una fuente rígida en el lado izquierdo. La cuadrícula 1D debe terminarse adecuadamente en el lado derecho para modelar un dominio infinito. El tamaño de la cuadrícula 1D es algo independiente del tamaño de la cuadrícula 2D: debe ser lo suficientemente grande

como para proporcionar un campo incidente asociado con cada una de las flechas verticales que se muestran arriba, pero de lo contrario puede ser mayor o menor que el ancho total de la cuadrícula 2D.

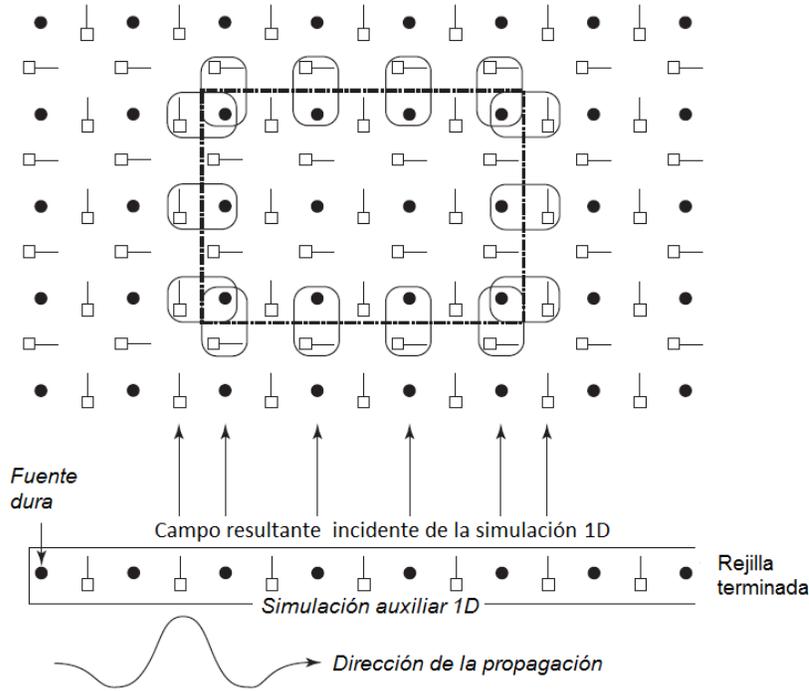


Figura 3.10: Se utiliza una cuadrícula auxiliar unidimensional para calcular el campo incidente que se supone que se propaga en la dirección $+x$. Las flechas verticales indican los nodos cuyos valores son necesarios para implementar el límite TFSF. El campo H_x incidente es cero y, por lo tanto, no se necesita corrección en asociación con los nodos E_z que tienen un nodo H_x vecino en el otro lado del límite.

3.5.1. Ejemplo de límite de TM^z TFSF

La Figura 3.11 muestra tres instantáneas de un dominio computacional que incorpora un límite TFSF. El tamaño de la cuadrícula es de 101 nodos de ancho y 81 nodos de alto. El campo incidente es una wavelet Ricker con 30 puntos por longitud de onda en su frecuencia más energética. Los índices para el primer nodo de campo eléctrico en la región TF son (5, 5) y los índices del último nodo en la región TF son (95, 75). No hay un dispersor presente y, por lo tanto, no hay campos visibles en la región SF. En la figura 3.11 (a) se ve que el campo incidente ha entrado en el lado izquierdo de la región TF. Hay una discontinuidad abrupta en el campo cuando uno cruza el límite de TFSF. Esta discontinuidad es visible a la izquierda de la región TF, así como a lo largo de una parte de la parte superior e inferior de la región. En la figura 3.11 (b) el pulso está casi completamente dentro de la región TF. En la figura 3.11 (b), el pulso incidente ha encontrado el

lado derecho de la región TF. En este punto, el campo del incidente aparentemente desaparece. Las correcciones a los campos en el lado derecho del límite son cuantos que el campo incidente no escapa de la región TF.

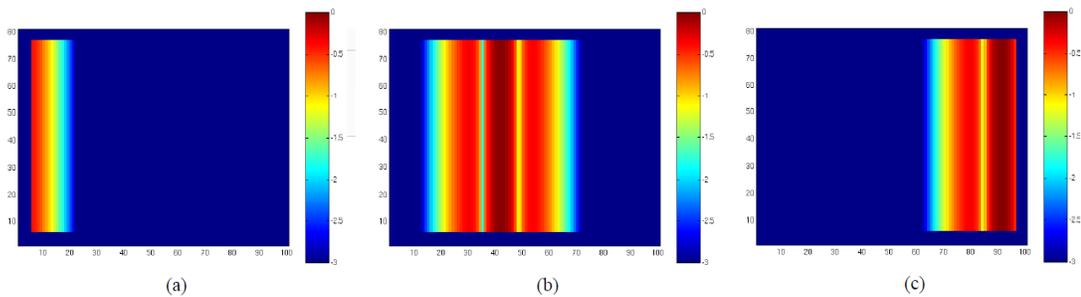


Figura 3.11: Visualización del campo E_z en un dominio computacional que emplea un límite TFSS. Las instantáneas se toman en los pasos de tiempo (a) 30, (b) 100 y (c) 170. La fuente de onda plana pulsada corresponde a una ondícula Ricker con 30 puntos por longitud de onda en su frecuencia más energética.

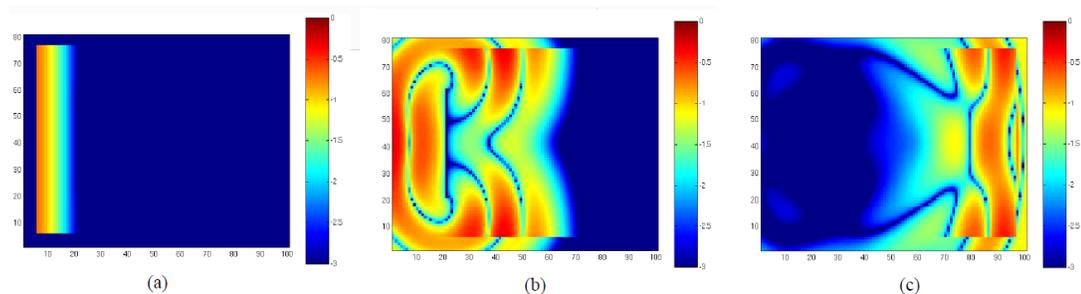


Figura 3.12: Visualización del campo E_z en un dominio computacional que emplea un límite TFSS. Hay una placa vertical PEC que se realiza al poner a cero el campo E_z sobre una línea que tiene 41 celdas de alto y 20 celdas desde el borde izquierdo del dominio computacional. Las instantáneas se toman en los pasos de tiempo (a) 30, (b) 100 y (c) 170. Se utiliza un ABC de segundo orden para terminar la cuadrícula.

La figura 3.12 muestra tres instantáneas de un dominio computacional que es similar al que se muestra en la figura 8.6. La única diferencia es que una placa PEC se ha colocado en la cuadrícula. La placa se realiza poniendo a cero los nodos E_z a lo largo de una línea vertical. Esta línea de nodos está desplazada 20 celdas desde el lado izquierdo del dominio computacional y corre verticalmente desde 20 celdas desde la parte inferior del dominio hasta 20 celdas desde la parte superior. En la Fig. 3.12 (a) el campo incidente apenas ha alcanzado la placa. Todavía no hay dispersión evidente y, por lo tanto, no hay campos dispersos visibles en la región SF. En la figura 3.12 (b) la interacción del campo con la placa es obvia. Uno puede ver cómo los campos se han difractado alrededor de los bordes de la placa. Como se puede ver, el campo dispersado desde la placa ha tenido tiempo de propagarse en la región SF. La Figura 3.12 (c) también muestra el campo distinto

de cero en la región SF.

3.6. FDTD tridimensional -3D-

Una cuadrícula 3D se puede ver como capas apiladas de cuadrículas Te^z y TM^z que se compensan medio paso espacial en la dirección z . Las ecuaciones de actualización para los nodos H_z y E_z son casi idénticas a las que ya se han proporcionado; la única diferencia es un índice adicional para especificar la ubicación z . Las ecuaciones de actualización para los otros componentes del campo requieren ligeros cambios para tener en cuenta las variaciones en la dirección z . Se detallan la disposición de los nodos en 3D y las ecuaciones de actualización asociadas para arreglos de dimensiones 3D en C .

3.6.1. Arreglos tridimensionales

Para los campos en un espacio 3D, es natural especificar la ubicación de un nodo utilizando tres índices que representan el desplazamiento en las direcciones x , y y z .

Para las matrices 3D, el incremento del tercer índice en uno cambia la variable que se especifica a la siguiente variable consecutiva en la memoria. Pensando en el tercer índice como correspondiente a la dirección z , esto implica que los nodos que son adyacentes entre sí en la dirección z también son adyacentes entre sí en la memoria. Por otro lado, cuando el primer o segundo índice se incrementa en uno, eso no se corresponderá con la siguiente variable en la memoria si no a la misma dimensión del índice Z (tercer índice). Una representación general del espacio 3D modelando sistemas tridimensionales en memoria con sus respectivos índices x , y y z , la podemos observar en la figura 3.13.

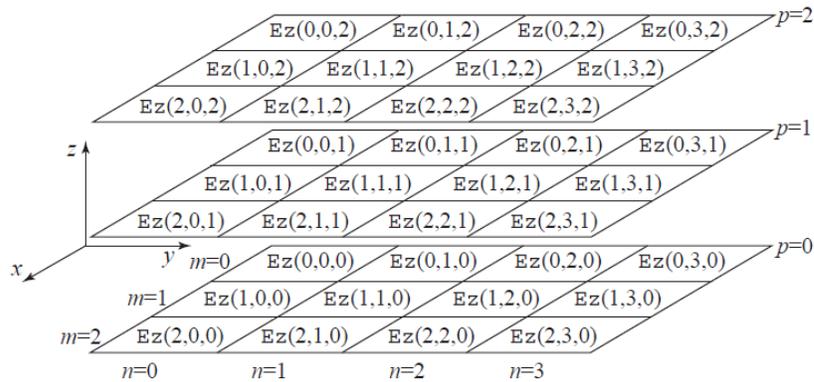


Figura 3.13: Representación de elementos de una matriz con dimensiones $3 \times 4 \times 3$ en las direcciones x , y y z , respectivamente. Los índices m , n y p se utilizan para especificar las preocupaciones x , y y z , respectivamente. El elemento en el origen tiene índices $(0,0,0)$ y se muestra en la esquina superior izquierda del plano inferior.

3.7. Ecuaciones gobernantes en una malla 3D.

Como ha sido el caso anteriormente, las leyes de Ampere y Faraday son las ecuaciones de gobierno relevantes en la construcción del algoritmo FDTD. Estas ecuaciones son:

$$-\sigma_m \mathbf{H} - \mu \frac{\partial \mathbf{H}}{\partial t} = \nabla \times E = \begin{vmatrix} \hat{a}_x & \hat{a}_y & \hat{a}_z \\ \frac{\partial}{\partial x} & \frac{\partial}{\partial y} & \frac{\partial}{\partial z} \\ E_x & E_y & E_z \end{vmatrix} \quad (3.25)$$

$$-\sigma_m \mathbf{E} - \mu \frac{\partial \mathbf{E}}{\partial t} = \nabla \times H = \begin{vmatrix} \hat{a}_x & \hat{a}_y & \hat{a}_z \\ \frac{\partial}{\partial x} & \frac{\partial}{\partial y} & \frac{\partial}{\partial z} \\ H_x & H_y & H_z \end{vmatrix} \quad (3.26)$$

Los componentes de estas ecuaciones, cuando se aproximan por diferencias finitas en los puntos apropiados en el espacio-tiempo, producen las ecuaciones de actualización discretas. Esta agrupación de seis nodos puede considerarse el elemento fundamental de una cuadrícula 3D. Se utiliza la siguiente notación:

$$H_x(x, y, z, t) = H_x(m\Delta_x, n\Delta_y, p\Delta_z, q\Delta_t) = H_x^q[n, m, p], \quad (3.27)$$

$$H_y(x, y, z, t) = H_y(m\Delta_x, n\Delta_y, p\Delta_z, q\Delta_t) = H_y^q[n, m, p], \quad (3.28)$$

$$H_z(x, y, z, t) = H_z(m\Delta_x, n\Delta_y, p\Delta_z, q\Delta_t) = H_z^q[n, m, p], \quad (3.29)$$

$$E_x(x, y, z, t) = E_x(m\Delta_x, n\Delta_y, p\Delta_z, q\Delta_t) = E_x^q[n, m, p], \quad (3.30)$$

$$E_y(x, y, z, t) = E_y(m\Delta_x, n\Delta_y, p\Delta_z, q\Delta_t) = E_y^q[n, m, p], \quad (3.31)$$

$$E_z(x, y, z, t) = E_z(m\Delta_x, n\Delta_y, p\Delta_z, q\Delta_t) = E_z^q[n, m, p], \quad (3.32)$$

En la figura 3.14 Los nodos de campo eléctrico existen en múltiplos enteros del paso de tiempo y que los nodos de campo magnético existen a la mitad de un paso temporal lejos de los nodos de campo eléctrico.

En la figura 3.14 Los nodos de campo eléctrico existen en múltiplos enteros del paso de tiempo y que los nodos de campo magnético existen a la mitad de un paso temporal lejos de los nodos de campo eléctrico. Los nodos de campo eléctrico se desplazan medio paso en la dirección en que apuntan, mientras que los nodos de campo magnético se desplazan medio paso en las dos direcciones que no apuntan.

En la figura 3.15 se muestra otra vista de una parte de la cuadrícula 3D. Este tipo de representación generalmente se llama cubo Yee o celda Yee. Este cubo consta de nodos de campo eléctrico en los bordes del cubo (por lo tanto, cuatro nodos de cada componente de campo eléctrico) y nodos de campo magnético en las caras (dos nodos de cada componente de campo magnético). En una cuadrícula 3D,

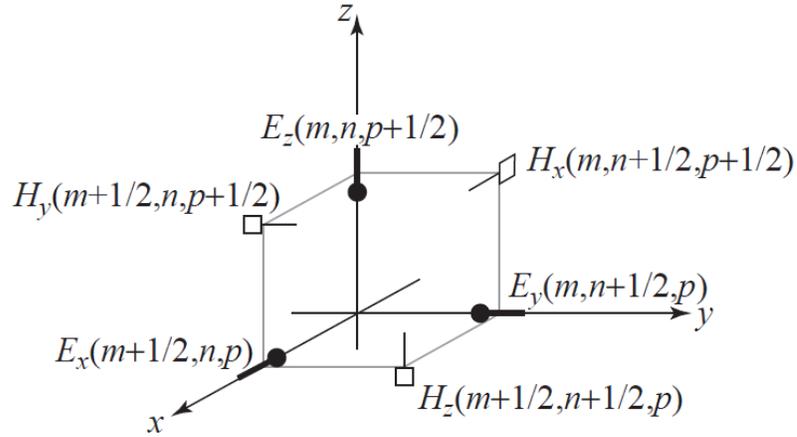


Figura 3.14: Los nodos de campo eléctrico se desplazan medio paso en la dirección en la que apuntan, mientras que los nodos de campo magnético se desplazan medio paso en las dos direcciones que no apuntan. También se entiende implícitamente que los nodos de campo eléctrico y magnético se compensan entre sí medio paso en el tiempo.

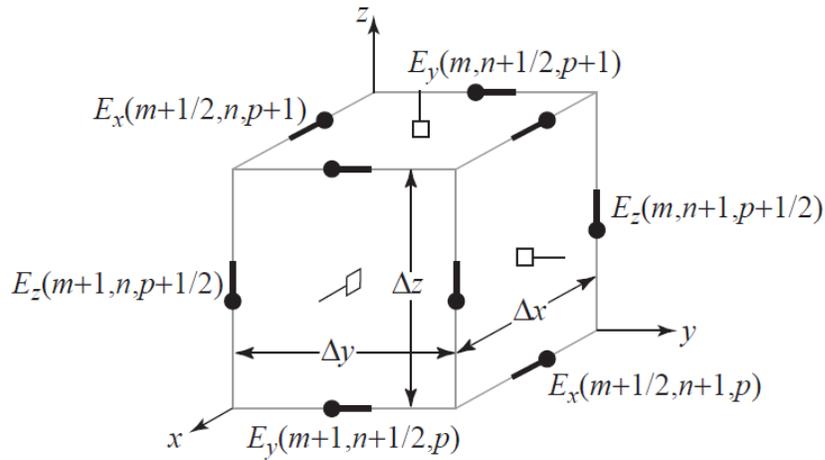


Figura 3.15: Los nodos en una cuadrícula FDTD 3D a menudo se dibujan en forma de un cubo o celda Yee. En esta representación, no todos los nodos tienen los mismos índices. El cubo consistiría en cuatro nodos E_x , cuatro nodos E_y y cuatro nodos E_z , es decir, los campos eléctricos están a lo largo de los bordes del cubo. Los campos magnéticos están en las caras del cubo y , por lo tanto, habría dos nodos H_x , dos nodos H_y y dos nodos H_z .

se puede cambiar el origen de este cubo para que los nodos de campo magnético estén a lo largo de los bordes y los nodos de campo eléctrico estén en las caras.

Con la disposición de los nodos mostrados en las Figs. 3.14 y 3.15, los componentes de 3.25 y 3.26 expresados en los puntos de evaluación apropiados son:

$$-\sigma H_x - \mu \frac{\partial H_x}{\partial t} = \frac{\partial E_z}{\partial E_y} - \frac{\partial E_y}{\partial E_z} \Big|_{x=m\Delta_x, y=(m+1/2)\Delta_y, z=(p+1/2)\Delta_z, t=q\Delta_t}, \quad (3.33)$$

$$-\sigma H_y - \mu \frac{\partial H_y}{\partial t} = \frac{\partial E_x}{\partial E_z} - \frac{\partial E_z}{\partial E_x} \Big|_{x=(m+1/2)\Delta_x, y=n\Delta_y, z=(p+1/2)\Delta_z, t=q\Delta_t}, \quad (3.34)$$

$$-\sigma H_z - \mu \frac{\partial H_z}{\partial t} = \frac{\partial E_y}{\partial E_x} - \frac{\partial E_x}{\partial E_y} \Big|_{x=(m+1/2)\Delta_x, y=(n+1/2)\Delta_y, z=p\Delta_z, t=q\Delta_t}, \quad (3.35)$$

$$\sigma E_x + \epsilon \frac{\partial E_x}{\partial t} = \frac{\partial H_z}{\partial H_y} - \frac{\partial E_y}{\partial E_z} \Big|_{x=(m+1/2)\Delta_x, y=n\Delta_y, z=p\Delta_z, t=(q+1/2)\Delta_t}, \quad (3.36)$$

$$\sigma E_y + \epsilon \frac{\partial E_y}{\partial t} = \frac{\partial H_x}{\partial H_z} - \frac{\partial E_z}{\partial E_x} \Big|_{x=m\Delta_x, y=(n+1/2)\Delta_y, z=p\Delta_z, t=(q+1/2)\Delta_t}, \quad (3.37)$$

$$\sigma E_z + \epsilon \frac{\partial E_z}{\partial t} = \frac{\partial H_y}{\partial H_x} - \frac{\partial E_x}{\partial E_y} \Big|_{x=m\Delta_x, y=n\Delta_y, z=(p+1/2)\Delta_z, t=(q+1/2)\Delta_t}, \quad (3.38)$$

En las ecuaciones 3.33-3.38, la derivada temporal de cada componente de campo siempre viene dada por la derivada espacial de dos componentes del otro campo. Además, los componentes de un campo están relacionados con los dos componentes ortogonales del otro campo. Como se ha hecho anteriormente, el término de pérdida puede ser aproximado por el promedio del campo en dos pasos.

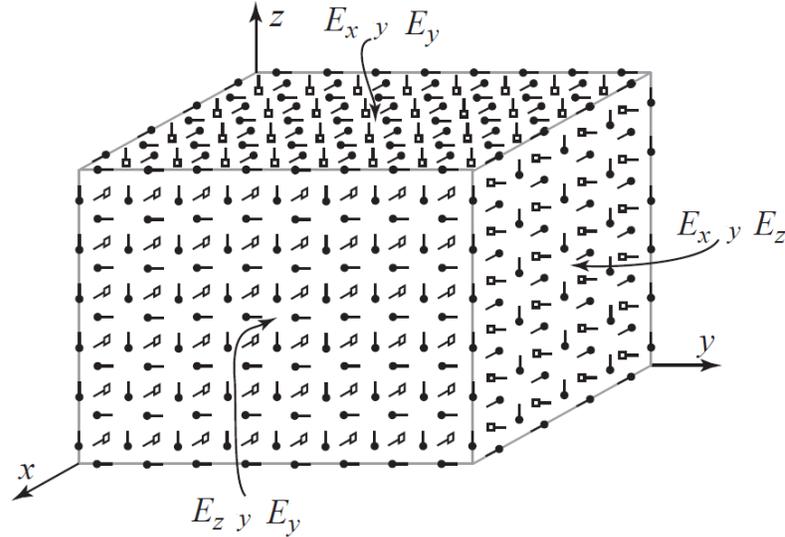


Figura 3.16: Caras de un dominio computacional que es $5 \times 9 \times 7$ en las direcciones x, y y z , respectivamente. En la cara constante x , los campos tangenciales son E_y y E_z , en la cara constante y son E_x y E_z , y en la cara constante z son E_x y E_y . También hay nodos de campo magnético que existen en estas caras, pero su orientación es normal a la cara.

En nuestra construcción de cuadrículas 3D, las caras de la cuadrícula siempre

estarán terminadas de modo que haya dos componentes de campo eléctrico tangenciales a la cara y un campo magnético normal a ella como se muestra en la figura 3.16. El dominio computacional que se muestra en esta figura es uno que describimos como que tiene dimensiones de $5 \times 9 \times 7$ en las direcciones x, y y z , respectivamente.

Aunque llamamos a esto una cuadrícula de $5 \times 9 \times 7$, ninguna de las matrices asociadas con este dominio computacional tiene estas ¡dimensiones! Los campos de un dominio computacional que es $M \times N \times P$ tendrían dimensiones de $E_x : (M - 1) \times N \times P$, $E_y : M \times (N - 1) \times P$, $E_z : M \times N \times (P - 1)$, $H_x : M \times (N - 1) \times (P - 1)$, $H_y : (M - 1) \times N \times (P - 1)$ y $H_z : (M - 1) \times (N - 1) \times P$. Podemos visualizar que la propagación de las ondas electromagnéticas ocurren dentro del cubo de Yee, en la cual se representada en la figura 3.16, en donde existirán los momentos de diferencias finitas en un tiempo determinado en el dominio computacional.

3.7.1. Ejemplo de simulación de un dipolo 3D de una antena

Ejemplo de radiación del campo E_z usando diferencias finitas en dominio del tiempo, para la simulación de un dipolo 3D de una antena. Observamos los momentos para los tiempos: $T=30, 40, 50$ y 60 respectivamente de las imágenes de la figura 3.17(a-d). Podemos observar el mismo caso de onda en la figura 3.18, con

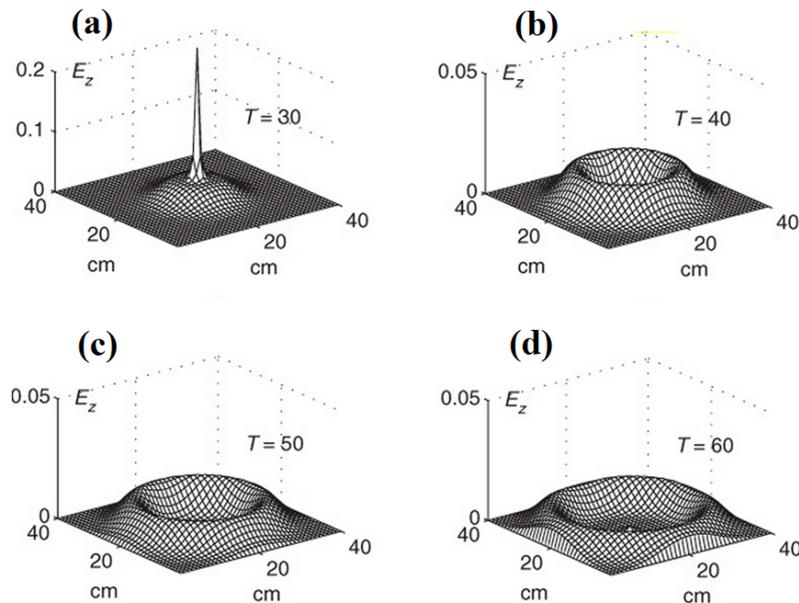


Figura 3.17: Muestra la propagación de la E_z desde el dipolo en el nivel del plano XY con el espacio en el dipolo. Por supuesto, hay radiación en la dirección Z a medida que avanzamos. Esto ilustra un problema importante en las simulaciones tridimensionales: a menos que uno tenga gráficos inusualmente buenos, visualizar tres dimensiones será difícil [34].

amplitudes diferentes de propagación de una onda en sentidos omnidireccionales del campo E_z en las imágenes de la figura 3.18(a-c), mientras que en la imagen de la figura 3.18(d) observamos una vista XY de la superficie de la onda. Podemos observar una imagen 3D completa de la propagación de onda en la figura 3.19.

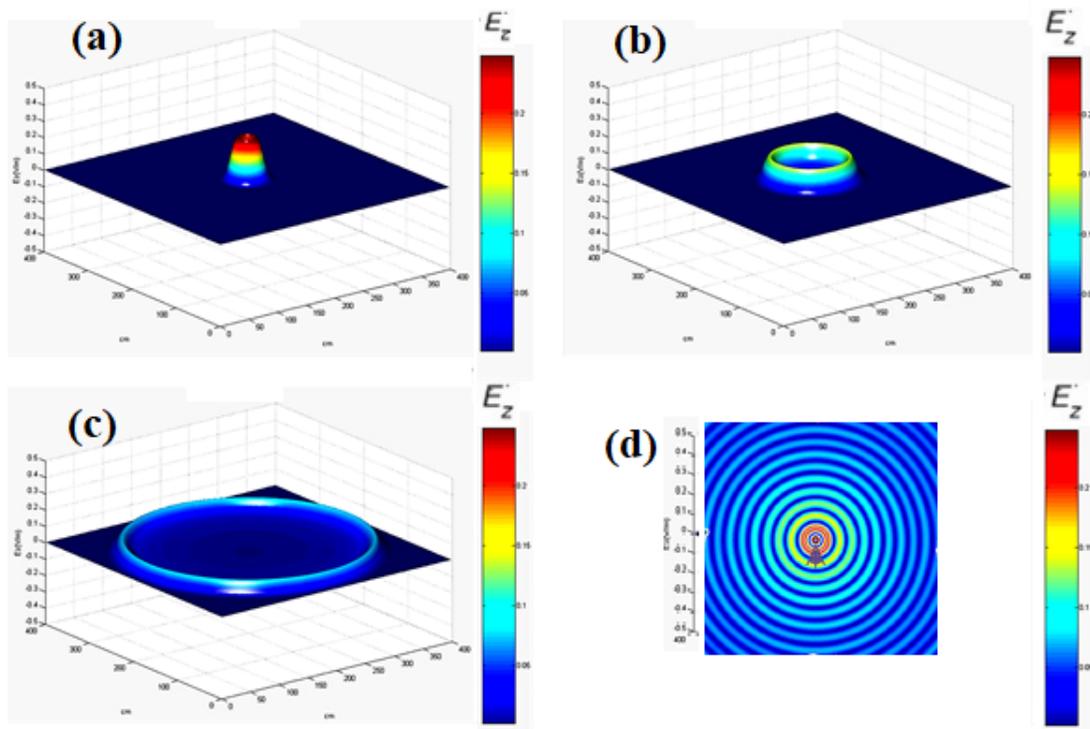


Figura 3.18: Diferentes momentos de onda omnidireccional de una antena radiante de una antena transmisora. La amplitud de la onda está dada en los colores de la barra *colorBar*. Del campo E_z de fuerte a débil (rojo-azul) [35].

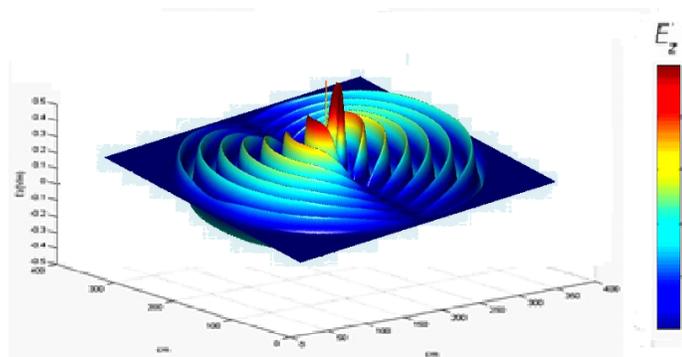


Figura 3.19: Superficie 3D de la propagación de onda electromagnética E_z emitida por el dipolo de una antena transmisora. Los círculos indican la antena emisora donde se encuentra el receptor de interés la intensidad roja indica mayor potencia y donde se irradia la mayor propagación de onda direccional. [35].

El método de diferencias finitas en el tiempo (FDTD), es también una poderosa herramienta de simulación, creada para resolver las ecuaciones de Maxwell, las ecuaciones a resolver deben ser remplazadas por un sistema de ecuaciones expresadas en diferencias finitas donde se debe elegir las componentes a evaluar de manera conveniente. Como es bien sabido, una onda electromagnética que se propaga por el vacío no puede ir más rápido que la velocidad de la luz, por lo tanto para lograr su propagación en dirección paralela a los ejes, se requiere un tiempo mínimo:

$$\Delta t = \frac{\Delta x}{c}, \quad (3.39)$$

Δt =Incremento en el tiempo

Δx =Incremento en el eje x

c =Velocidad de la luz de 300 000 km/s

Las ecuaciones de Maxwell (3.40-3.43) formuladas para el caso general en el vacío y en un medio material respectivamente.

En el vacío

$$\vec{\nabla} \cdot \vec{E} = \frac{\rho}{\epsilon_0} \quad (3.40)$$

$$\vec{\nabla} \cdot \vec{B} = 0 \quad (3.41)$$

En el medio:

$$\vec{\nabla} \cdot \vec{D} = \rho \quad (3.42)$$

$$\vec{\nabla} \cdot \vec{B} = 0 \quad (3.43)$$

Dónde:

$$\vec{B} = \mu \vec{H} = \mu_0 \mu_r \vec{H},$$

$$\vec{D} = \epsilon \vec{E} = \epsilon_0 \epsilon_r \vec{E}$$

ρ =Densidad de carga en el medio del interior de la superficie cerrada (S)

ϵ_0 =Permitividad del vacío = $8,854 \times 10^{-12} \frac{F}{m}$

μ_0 =Permeabilidad magnética = $4\pi \times 10^{-7} \frac{H}{m}$

Nota: Donde el material es un medio dieléctrico.

Las derivadas parciales de las ecuaciones de Maxwell, se remplazan por un sistema de ecuaciones de diferencias finitas en el tiempo para la resolución de derivadas parciales [18]. Las ecuaciones de Maxwell, también describen la evolución del tiempo y espacio de los campos eléctricos \mathbf{E} y los campos magnéticos \mathbf{H} . Las ecuaciones de Maxwell son tan generales que resumen las principales leyes electromagnéticas de la física.

3.8. Diagrama de la celda de Yee -3D

Fue propuesta originalmente por Kane Yee en 1966. En la Figura 3.20, se muestra la configuración de los 6 componentes de campo en la misma celda (i, j, k). Todos los componentes de campo tienen la misma posición de memoria (i, j, k) que $E_x[i][j][k]$. Sin embargo, si representamos la posición del campo precisamente

hasta $1/2$ cuadrícula, entonces, por ejemplo, $E_x[i][j][k]$ se puede representar como $E_x | i + 1/2, j, k$. De la misma manera, se puede ver que $H_x[i][j][k] \longleftrightarrow H_x | i, j + 1/2, k + 1/2$. Tenga en cuenta que la celda **Yee** de la unidad que se muestra en la Figura 3.20 se rellena todo el espacio 3 - D sin huecos. También hemos asumido una celda unitaria paralelepípeda rectangular, en la que las longitudes laterales son diferentes ($4xi, 4yj, 4zk$) para diferentes posiciones (i, j, k). Ahora representamos las primeras derivadas en las ecuaciones de Maxwell en términos de las diferencias finitas.

$$\frac{\partial E}{\partial t} = \frac{1}{\epsilon} \nabla \times H \quad (3.44)$$

$$\frac{\partial H}{\partial t} = \frac{1}{\mu} \nabla \times E \quad (3.45)$$

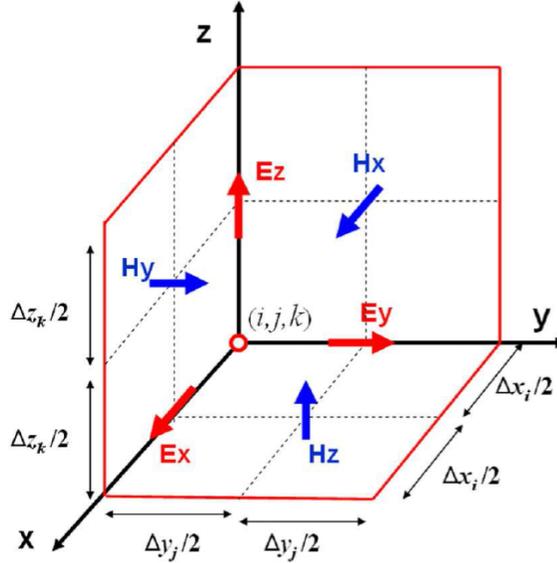


Figura 3.20: En esta figura se muestran los cuadrantes en donde podemos implementar los cuerpos, su comportamiento y movimiento en el espacio y tiempo.

3.9. Cuerpos que se pueden simular con el método FDTD

Los cuerpos que se pueden simular mediante el método de diferencias finitas en el dominio del tiempo son: Blocks (Bloques) es el cuerpo comúnmente mas usado en simulaciones FDTD, rod (cilindros, en orientación horizontal paralelo al eje x, Horizontal paralelo al eje y, en orientación vertical paralelo al eje z), Donut (Donas y Toroides), Elipses con altura, Elipsoidales 3D, Cones (Conos) con radios R_1 - R_2 , para la base y punta respectivamente. Spheere (Esferas), shell (Esferas huecas),

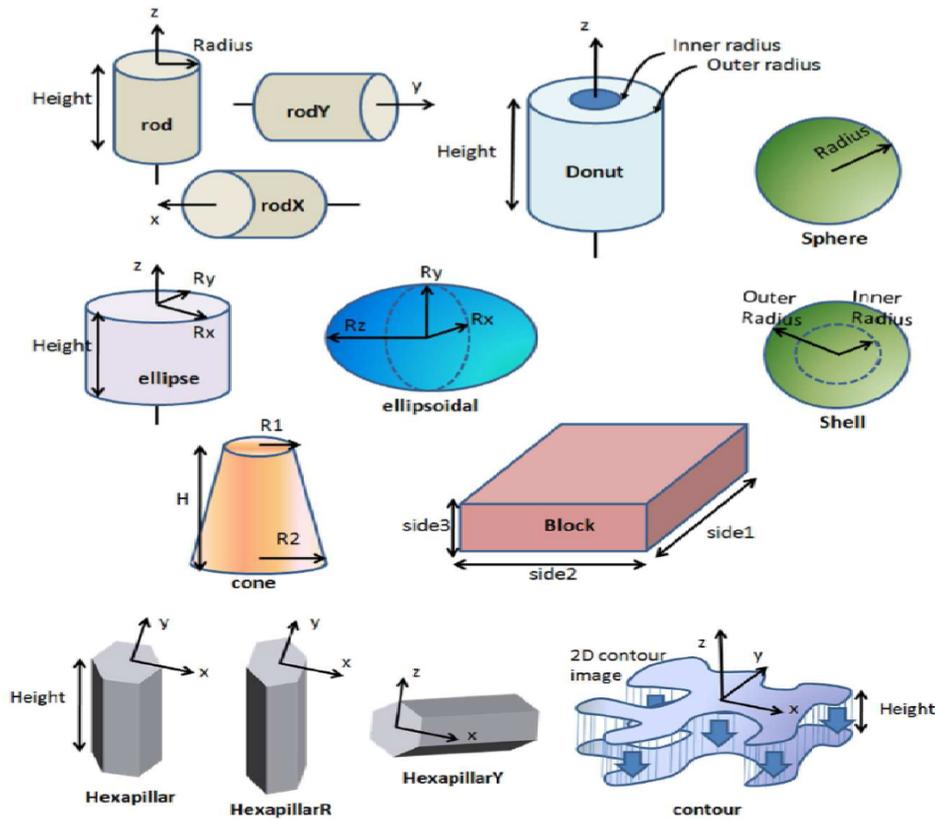


Figura 3.21: Esquema que muestra todas las figuras que se pueden simular con el método FDTD.

Hexapillars (Pilares Hexagonales) con orientación vertical y horizontal, y contornos que forman un cuerpo 3D área de su con base igual a su superficie. Ver figura 3.21.

3.10. Distribución espacial de formas 3D

Podemos combinar de diferentes maneras y a conveniencia las diferentes figuras 3D proporcionadas por el método FDTD. En la figura 3.22 podemos observar la combinación de bloques, enumeradas por secciones de (1) a (5) que forman una estructura combinada, en donde podemos observar una distribución espacial de los elementos combinados, con puntos medios que indican el punto medio de cada bloque. El primer bloque (1), representa un bloque de espacio, que generalmente es utilizado para observar el comportamiento de las propagaciones de onda del fenómeno producido de las otras estructuras, por lo general este bloque se le confina el valor de vacío o aire, este primer bloque tiene una dimensión o tamaño de S_1 . En el bloque número 2 suponemos algunas estructuras cilíndricas que ocupa un espacio de tamaño S_2 . Los bloques (1) a (3) representan boques solidos

que pueden ser considerados como metales, conductores o dieléctricos convenientes para la propagación de ondas electromagnéticas. Podemos observar la distribución espacial entre cada bloque que conforman una estructura combinada de dimensión Z_0 , con incrementos de tamaño escalonado que se incorpora un bloque el tamaño espacial aumenta y compone un tamaño total de la estructura Z_0 , esta relación las podemos observar en las ecuaciones 3.47-3.51, donde P , es una Parte del tamaño total Z_0 .

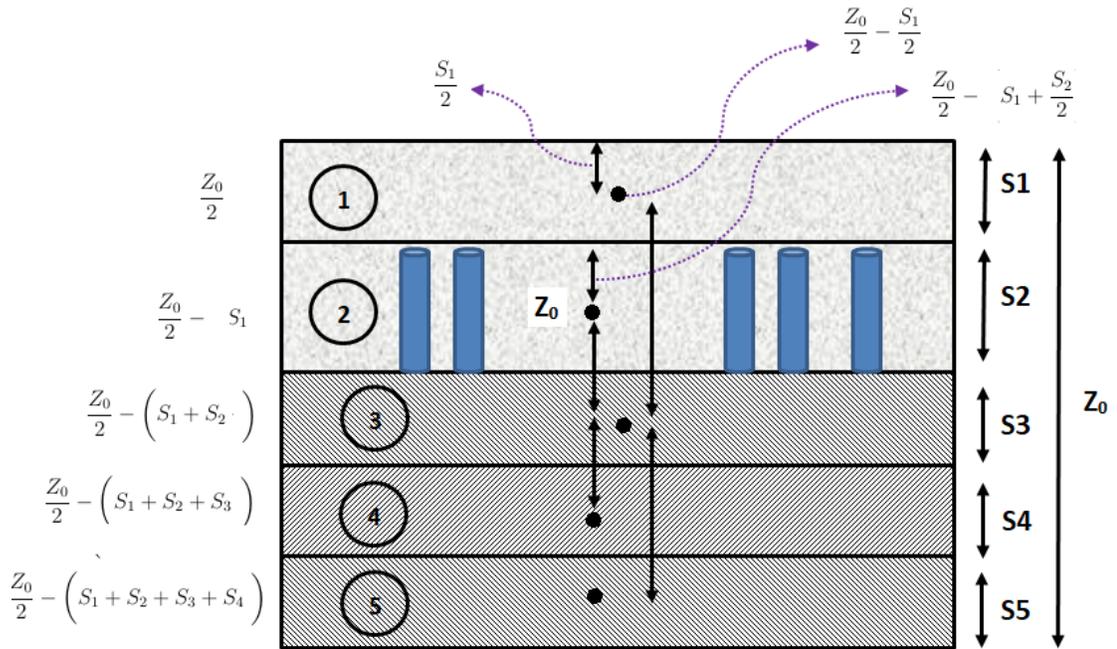


Figura 3.22: Representación espacial de bloques (1)-(5) formando una estructura compuesta.

La ecuación: 3.46, muestra el total del espacio que ocupan los bloques representado por el valor Z_0 .

$$Z_0 = \frac{1}{2}(S_1 + S_2 + S_3 + S_4 + S_5) \quad (3.46)$$

$$P_1 = \frac{Z_0}{2} - \frac{S_1}{2}; \quad (3.47)$$

$$P_2 = \frac{Z_0}{2} - \left(S_1 + \frac{S_2}{2}\right); \quad (3.48)$$

$$P_3 = \frac{Z_0}{2} - \left(S_1 + S_2 + \frac{S_3}{2}\right); \quad (3.49)$$

$$P_4 = \frac{Z_0}{2} - \left(S_1 + S_2 + S_3 + \frac{S_4}{2}\right); \quad (3.50)$$

$$P_5 = \frac{Z_0}{2} - \left(S_1 + S_2 + S_3 + S_4 + \frac{S_5}{2}\right); \quad (3.51)$$

3.11. Código para formas geométricas

Se muestra una un segmento del código de la clase: *MetalObjetscs*, para la creación de figuras 3D implementando el método FDTD en el lenguaje de programación C#.

```

1  /* Creación de figuras 3D implementando el método FDTD en el lenguaje de programación C#
2  if (shape == "rod") // Se crea un cilindro
3  {
4  object0.centeri = (centerx + _xCenter) * _lattice_x;
5  object0.centerj = (centery + _yCenter) * _lattice_y;
6  object0.centerk = (centerz);
7  object0.size1 = size1 * _lattice_x; // radio
8  object0.size2 = (size2);
9  }
10 else if (shape == "rodY") // Cilindro con orientation vertical
11 {
12 object0.centeri = (centerx + _xCenter) * _lattice_x;
13 object0.centerj = (centery + _yCenter) * _lattice_y;
14 object0.centerk = (centerz + _zCenter) * _lattice_z;
15 object0.size1 = size1 * _lattice_x; // radio
16 object0.size2 = size2 * _lattice_z; // y longitud
17 }
18 else if (shape == "donut") // Se crea una objeto con forma de dona
19 {
20 object0.centeri = (centerx + _xCenter) * _lattice_x;
21 object0.centerj = (centery + _yCenter) * _lattice_y;
22 object0.centerk = (centerz);          object0.size1 = size1 * _lattice_x; // radio interno
23 object0.size2 = size2 * _lattice_x;
24 }
25 else if (shape == "sphere") // Se crea una esfera
26 {
27 object0.centeri = (centerx + _xCenter) * _lattice_x;
28 object0.centerj = (centery + _yCenter) * _lattice_y;
29 object0.centerk = (centerz + _zCenter) * _lattice_z;
30 object0.size1 = size1 * _lattice_x; // radius
31 }
32 else if (shape == "ellipse") // Se crea una ellipse
33 {
34 object0.centeri = (centerx + _xCenter) * _lattice_x;
35 object0.centerj = (centery + _yCenter) * _lattice_y;
36 object0.centerk = (centerz);
37 object0.size1 = size1 * _lattice_x; // radio
38 object0.size2 = (size2); // amplitud
39 object0.size3 = size3; // radio=ry/rx
40 }

```

En el ejemplo del código anterior, podemos ver la invocación para la construcción de objetos: Rod? cilindro, Donut? toroide, Sphere? Esfera y una Ellipse? Elipse, que compara con la variable shape? Forma, los parámetros que determinan la ubicación en la nanoestructura los controlan las variables: `object0.centeri`, `object0.centerj` y `object0.centerz`.

En el siguiente capítulo abordaremos el acoplamiento que produce una carga que emite la radiación de Cherenkov sobre una grupo de nanotubos de carbono, que compone una nanoestructura, en el cual fue necesario utilizar el método de diferencias finitas en el dominio en el tiempo para calcular la radiación de energía emitida de plasmones-polaritones dentro y sobre la superficie.

Utilizamos el paradigma de programación orientada a objetos, construyendo varias clases que interactúan entre sí. En la figura 3.23, se muestra el esquema de clases principales que se utilizó en este proyecto.

3.12. Diagrama de clases

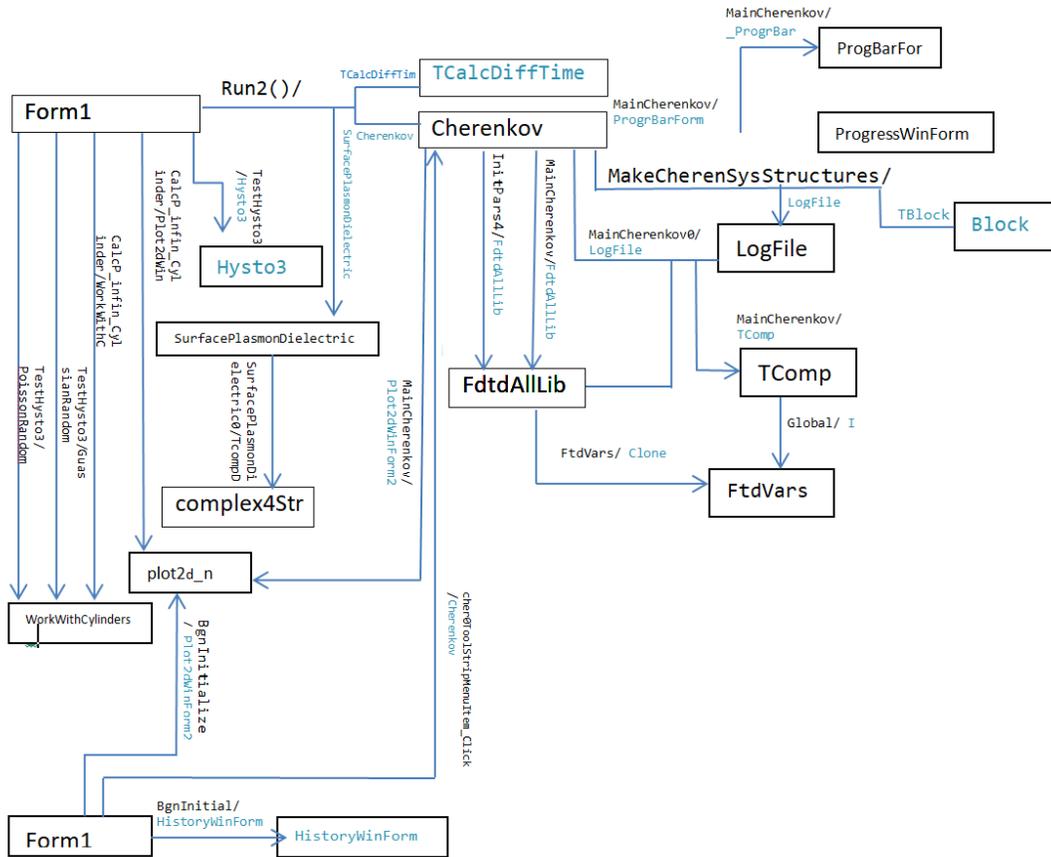


Figura 3.23: Diagrama de clases principales que indican la relación cuando se realiza el llamado a una subrutina e invocación de otras clases así como la relación de objetos y métodos.

En la figura 3.23, podemos observar las clases principales utilizadas en este proyecto, donde la clase Form1, es la clase Forma la cual implementamos los métodos de acción para llamar a las de más clases. Podemos observar cada clase encerrada por un rectángulo. Las flechas indican la dirección de la cual una clase hace un llamado a otra clase o subclase. Las clases encerradas en un rectángulo con texto negro: Cherenkov, SurfacePLasamonDielectric, complx4Str, $plot2s_n$, WorkWithCylinders, Form, FtdtAllLib, LogFil, ProgBarForm, ProgBarForm, TComp y FtdVars, son clases independientes, mientras que las clases marcadas con azul: Hysto3, HistoryWinForm, TCalcDiffTime y Block, son subclases que dependen de otras. La etiqueta que acompaña las flechas (Metodo/Variable) indican primero el método en el cual se llama la clase, construyendo una instancia con la variable del objeto con color de texto negro y azul respectivamente. Por ejemplo en el primer caso la clase Form, contiene un método Run en el cual se invoca la clase TalcDiffime instanciando la clase en la variable TCalcDiffTim.

ESTRUCTURA DE LA RADIACIÓN ÓPTICA DE CHERENKOV EN UN MEDIO COMPUESTO TRIDIMENSIONAL CON NANOTUBOS DE CARBONO CON MORFOLOGÍA PERIÓDICA Y ALEATORIA.

4.1. Introducción

La plasmónica representa una nueva área para la aplicación de plasmones de superficie a nano-escala en la rama de la fotonica y la electrónica [36]. Los plasmones-polaritones de superficie pueden servir como base para la creación de circuitos ópticos a nano-escala que podrán transportar las señales en varios rangos de frecuencia [37–44].

Las propiedades conductoras de los nanotubos de carbono (CNT) son de gran interés. Para modelar tales objetos, se utiliza la teoría de dispersión de cilindros [45–47]. Las propiedades de tales sistemas en diferentes configuraciones se han estudiado en detalle en documentos que tratan tanto de la comunicación óptica como de la física del plasma que ya están bien establecidas. Las propiedades importantes de tales sistemas son la señal de banda estrecha de la fuente y su considerable extensión espacial, que permite la formación de modos espaciales estables independientes bien definidos. Sin embargo, aunque tales estructuras cilíndricas son relativamente simples, la clasificación de los modos y la comprensión de los efectos observados aquí están lejos de ser triviales. La situación se vuelve más complicada en el caso de las CNT [48–51]. Primero, el factor aleatorio es importante aquí, porque en vista de las peculiaridades tecnológicas de la preparación de CNT, la longitud de los nanotubos, así como la frecuencia de plasma de sus

paredes, varía dentro de límites bastante amplios como se mencionó anteriormente en el capítulo II. Esto último lleva al hecho de que las frecuencias de la superficie los plasmones-polaritones resultan estar ubicados en un rango bastante amplio. En segundo lugar, a la pequeña distancia entre los nanotubos en el sistema CNT y el acoplamiento cruzado interno entre los CNT puede surgir en la localización moderada de plasmones (cuando la longitud de onda es comparable a la distancia promedio entre nanotubos), lo que conduce a nuevos efectos asociados con las oscilaciones de campo colectivo en dicho sistema CNT.

4.2. Radiación de Cherenkov

La radiación de Cherenkov es un tipo de onda que produce el brillo azulado característico de los reactores nucleares. Este es un fenómeno similar al de la generación de onda de choque cuando se supera la velocidad del sonido. En este caso los frentes de onda esféricos se superponen y forman una sola con forma cónica, debido a que la luz también es una onda electromagnética, puede producir los mismos efectos si su velocidad es superada, y esto solo puede ocurrir cuando las partículas que viajan en un medio distinto del vacío, lo hacen a velocidades superiores a la de los fotones en dicho medio [52, 53].

La radiación Cherenkov sólo se produce si la partícula que atraviesa el medio esta cargada eléctricamente, como por ejemplo, un protón. Para que se produzca radiación Cherenkov el medio debe ser un dieléctrico. Es decir; debe estar formado por átomos o moléculas capaces de verse afectados por un campo eléctrico. Por tanto, un protón viajando a través de un medio hecho de neutrones, por ejemplo, no emitiría radiación Cherenkov.

Los rayos cósmicos, compuestos principalmente por partículas cargadas, al incidir (interactuar) sobre los átomos y moléculas de la atmósfera terrestre (el medio), producen otras partículas, las cuales producen más partículas, y éstas producen más, creándose una verdadera cascada de partículas (muchas de ellas cargadas eléctricamente). Cada una de estas partículas polariza asimétricamente las moléculas de nitrógeno y oxígeno (componentes principales de la atmósfera terrestre) con las que se encuentra a su paso, las cuales, al despolarizarse espontáneamente, emiten radiación Cherenkov (detectada con telescopios Cherenkov). Es decir; son las moléculas de la atmósfera (el dieléctrico) las que emiten la radiación, no la partícula incidente. La polarización es asimétrica porque las moléculas que hay delante de la partícula no se han polarizado cuando las de detrás ya lo han hecho. Las de delante no se han polarizado porque, la partícula viaja más rápido que su propio campo eléctrico. Cuando la polarización es simétrica (cuando la partícula viaja a menor velocidad que la de la luz en el medio) no se produce radiación Cherenkov.

El efecto Cherenkov es de gran utilidad en los detectores de partículas donde la susodicha radiación es usada como trazador. Particularmente en los detectores de neutrinos en agua pesada como el Super-Kamiokande. También en el tipo de

Figura 4.1: Onda cónica que produce la radiación de Cherenkov.

telescopio conocido como telescopio Cherenkov como el telescopio MAGIC y el observatorio de rayos gamma HAWC, que detectan la luz Cherenkov producida en la atmósfera terrestre generada por la llegada de rayos gamma de muy alta energía procedentes del espacio. Los rayos gamma no tienen carga eléctrica, pero al incidir con los átomos de la atmósfera terrestre se produce una cascada de partículas que sí tienen carga eléctrica.

Recientemente, la radiación de Cherenkov se investiga ampliamente en el caso de diversos materiales, meta materiales y diferentes configuraciones, se estudió la excitación de campo en nanotubos de carbono por una carga móvil con el uso de la técnica numérica FDTD.

La frecuencia del espectro de la radiación de Cherenkov de una partícula en un medio dieléctrico está dada por la fórmula de Frank-Tamm:

$$\frac{d^2E}{dx d\omega} = \frac{q^2}{4\pi} \omega \left(1 - \frac{c^2}{v^2 n^2(\omega)} \right), \quad (4.1)$$

Dónde: $n(\omega)$ es el índice de refracción del material donde la carga de la partícula se mueve con velocidad v , ω es la frecuencia emitida por la radiación de la partícula, q es la carga que atraviesa el medio y c es la velocidad de la luz. La radiación de Cherenkov es una frecuencia continua y alrededor del espectro visible; La intensidad relativa por unidad de frecuencia es aproximadamente proporcional a la frecuencia ω . Es decir, las frecuencias más altas son más intensas en la radiación de Cherenkov. En un material transparente, esta radiación se emite en un ángulo θ en un intervalo de $\sin \theta d\theta = c/vn^2(\omega)(dn(\omega)/d\omega)$. Recientemente, la radiación de Cherenkov ha sido ampliamente investigada en el caso de varios materiales, como son los dieléctricos a base de grafeno o CNTs [54–60].

4.3. La superficie de plasmón-polaritones en nanotubos conductores infinitos (caso analítico)

Para analizar la energía que emite un objeto utilizamos la teoría de conductividad dispersiva en cilindros infinitos. Consideramos una columna de plasma magnetizada de radio a , en un nanotubo conductor de radio b dentro de una sustancia con constante dieléctrica. Ver figura 4.2. En una aproximación analítica la

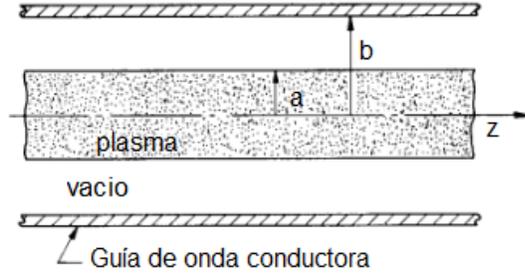


Figura 4.2: Esquema parcial del cilindro infinito desmagnetizado que puede producir plasma.

conductividad en los nanotubos es representada por la conductividad en el cilindro (nanotubo) teniendo una permitividad dieléctrica de Drude como se muestra en la siguiente ecuación.

$$\varepsilon(\omega) = \varepsilon_h - \frac{\omega_p^2}{\omega^2 + i\gamma_e\omega}, \quad (4.2)$$

Donde ω_p es la frecuencia del plasma ($\omega_p^2 = e^2 N_e / \varepsilon_0 m_e$), $\varepsilon_h = 1$, para casos $\varepsilon_h > 1$, lo abordaremos posteriormente con el método FDTD (Función en Dominio del Tiempo de Diferencias Finitas). ε_0 , es la permitividad en el vacío, γ_e es la pérdida de energía del electrón, N_e es la densidad del electrón m_e , es la masa de la carga del electrón.

Suponemos que las CNT están rodeadas por vecinos similares CNT colocados por una distancia (radio b) que se puede interpretar como medio conductor circundante. Para modos con pérdida las ecuaciones de Maxwell con condiciones de la frontera para los modos de la propagación electromagnética están dados por:

$$\begin{cases} \left\{ \nabla_T^2 + \left[\frac{\omega^2}{c^2} \left(1 - \frac{\omega_p^2}{\omega^2} \right) - k^2 \right] \right\} \begin{bmatrix} E_{1z} \\ B_{1z} \end{bmatrix} = 0, 0 < r < a \\ \left[\nabla_T^2 + \left(\frac{\omega^2}{c^2} - k^2 \right) \right] \begin{bmatrix} E_{1z} \\ B_{1z} \end{bmatrix} = 0, a < r < b \end{cases} \quad (4.3)$$

Donde ∇_T^2 es la transversa laplaciana

$$\nabla_T^2 = \frac{1}{r} \frac{\partial}{\partial r} r \frac{\partial}{\partial r} + \frac{1}{r^2} \frac{\partial^2}{\partial \theta^2}.$$

Las ondas asociadas con la solución de la ecuación. (4.3) para E_z se denominan modos E , mientras que los asociados con B_z se denominan modos B . La solución a la ecuación (4.3) para modos E simétricos azimutalmente que se propagan a menos de la velocidad de la luz ($k^2 > \omega^2/c^2$) puede escribirse como:

$$E_{1z} = A \frac{I_0(\tau r)}{I_0(\tau a)} e^{i(kz - \omega t)} + c.c., 0 < r < a \quad (4.4)$$

$$A \frac{I_0(\tau_0 r) K_0(\tau_0 b) - I_0(\tau_0 b) K_0(\tau_0 r)}{I_0(\tau_0 a) K_0(\tau_0 b) - I_0(\tau_0 b) K_0(\tau_0 a)} e^{i(kz - \omega t)} + c.c., a < r < b \quad (4.5)$$

Donde:

$$\tau_0^2 = k^2 - \frac{\omega^2}{c^2} > 0, \tau^2 = k^2 - \frac{\omega^2}{c^2} + \frac{\omega_p^2}{c^2} > 0. \quad (4.6)$$

Las funciones I_0 y K_0 son funciones modificadas de Bessel del primer y segundo orden respectivamente. Imponemos las condiciones de los límites de la frontera para los campos: las soluciones son finitas en el centro ($r = 0$), continuas en el límite CNT ($r = a$), y son cero en el límite externo ($r = b$) Esto último lleva a la siguiente relación de dispersión [61]:

$$\left(1 - \frac{\omega_p^2}{\omega^2}\right) \frac{1}{\tau a} \frac{I_0'(\tau a)}{I_0(\tau a)} = \frac{1}{\tau_0 a} \frac{I_0'(\tau_0 a) K_0(\tau_0 b) - I_0(\tau_0 b) K_0'(\tau_0 a)}{I_0(\tau_0 a) K_0(\tau_0 b) - I_0(\tau_0 b) K_0(\tau_0 a)}, \quad (4.7)$$

Donde la prima denota un derivado con respecto al argumento. Para estudiar la ecuación de dispersión (4.7) numéricamente usamos las siguientes variables adimensionales $x = \omega/c$, $y = ka$, $q = \omega a/c$, y $p = a/b$. La figura 4.3, muestra la solución numérica de la ecuación de dispersión ecuación (4.7) para varios parámetros q y p . La estructura de radiación correspondiente del campo E_z se muestra en la figura 4.3.

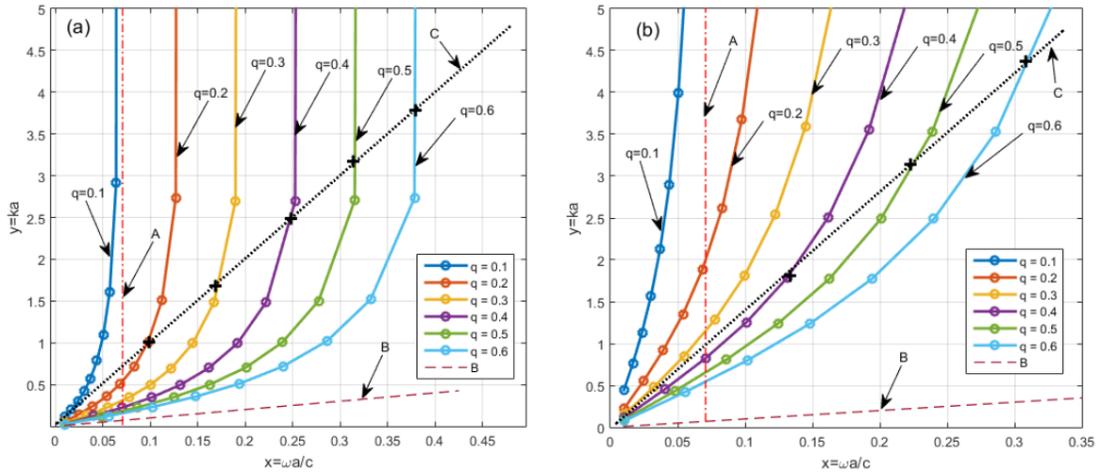


Figura 4.3: Dispersión para la de solución a la ecuación (4.7), donde $x = \omega a/c$, $y = ka$ con parámetros diferentes en $q = \omega a/c$, $k = \omega/c$ para (a) $p = 0.2$ y (b) $p = 0.9$.

La figura 4.3 muestra las dependencias de dispersión (solución a la ecuación (4.7) $y = y(x)$), donde $x = \omega a/c$ y $y = ka$ para parámetro diferente $q = \omega_p a/c = 0,1..,0,6$ y $p = a/b = 0,2,0,9$. La estructura radial correspondiente del campo E_z se muestra en la figura 4.4. Observamos de la figura 4.3 que para grandes $y > 1$ los valores x alcanzan los valores $q/\sqrt{2}$ respectivamente correspondientes a la frecuencia de plasmón superficial en la CNT. Esto significa que para ω_p fijo hay oscilaciones electromagnéticas de baja frecuencia a $0 < \omega < \omega_p/\sqrt{2}$ que son plasmones de superficie, indicados en la figura 4.3 para $q = 0,1$ por la flecha *A*. En $\omega_p/\sqrt{2} < \omega < \omega_p$ hay una brecha de frecuencia que separa el espectro del plasmón superficial de la rama de alta frecuencia indicada por la flecha *B* en la figura 4.3. La línea roja del punto de guiñón (*A*) muestra una frecuencia de polaritón de plasmón de superficie de este tipo por $q = 0,1$. La línea del guiñón (*B*) corresponde a la rama ligera $k = \omega/c$. La línea de punto negro (*C*) se aproxima a la parte lineal (en ω) de la densidad de energía espectral (unidades arbitrarias) de la carga radiante (ecuación (4.1)). Observamos que cuanto mayor es la energía irradiada por la carga en movimiento, más cercana es la frecuencia del campo excitado en CNT a la frecuencia de la superficie del plasmón-polaritón $\omega_p/\sqrt{2}$.

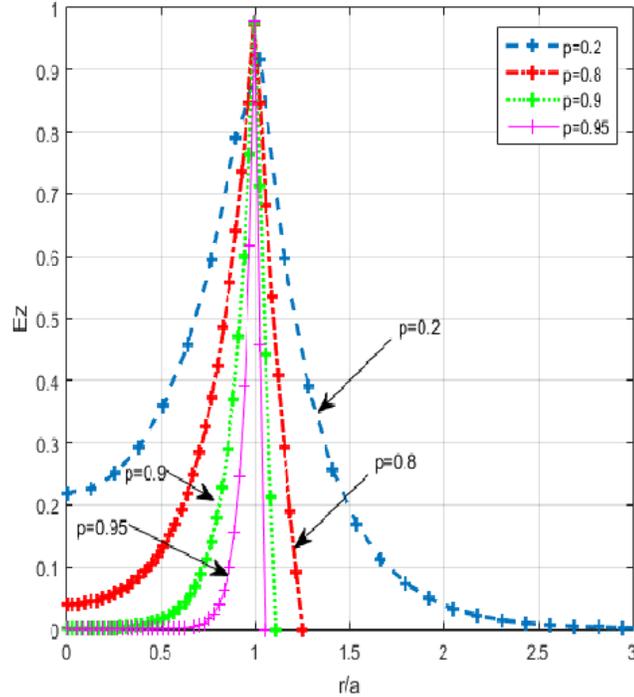


Figura 4.4: Estructura radial del campo E_z (para la ecuación (4.3) para $q = \omega/a c = 0,1$, $kr = 9$, y diferentes valores $p < a \leq b = 0.2; 0.8; 0.9$, y 0.95).

Se puede concluir que el rango de frecuencia $\omega \simeq \omega_p/\sqrt{2}$ aporta una contribución principal a las excitaciones de campo en CNT. La figura 4.4 muestra la estructura radial del campo E_z de CNT (ver la ecuación (4.3)) para los paráme-

tros $q = \omega_p a/c = 0,1$ y $p = a/b = 0,2, 0,8, 0,9, 0,95$, (ver Figura 4.3). Observamos de la figura 4.4 que E_z tiene un valor máximo bien definido en el límite CNT $r = a$; el campo E_z se desintegra rápidamente lejos del límite en el área $a < r \leq b$ que corresponde al plasmón de superficie localizado en CNT.

El valor del campo es bastante pequeño en el centro de CNT $r = 0$. De tal consideración se desprenden dos conclusiones: (i) el espectro de campo en CNT está determinado por la frecuencia de corte de los plasmones de superficie y (ii) cuanto mayor es la energía irradiada por la carga en movimiento, más cercana es la frecuencia del campo excitado en CNT a la superficie plasmón-polariton frecuencia. Sin embargo, en un caso real de nanotubos limitados, este enfoque simple se vuelve inválido. Para describir correctamente la estructura y la dinámica del campo óptico en un nanosistema compuesto, es importante explorar la nanoestructura tridimensional (3D) ligada a pérdida compuesta que se muestra en la figura 4.5.

4.4. La excitación de campo en nanotubos periodicos reales por una carga móvil mediante el método FDTD

La dispersión para las ecuaciones 4.3-4.7, solo sirven para casos con nanotubos infinitos más sim embargo en la vida real esto no es así. El caso de la ecuación 4.2 solo sirve para dar una aproximación en los campos reales de arreglos 3D conformados por nanotubos. Es por esta razón que utilizamos la solución numérica FDTD (Diferencias finitas en el dominio del tiempo) [62], para estudiar a detalle la estructura del campo óptico in nanotubos reales. Simulamos una carga se mueve a una velocidad uniforme v por encima del arreglo de nanotubos de tamaño $L \times L \times L$, donde $L = 100$ y con una densidad Gaussiana de carga:

$$f(r, t) = W^{-3} \exp\{ -[(x - v_0 t)^2 + y^2 + z^2]/W^2 \} \quad (4.8)$$

Donde W es la amplitud $W \rightarrow 0$ con distribución isotrópica. Realizamos un estudio numérico sistemático de dicho sistema con el uso de la técnica 3D FDTD. Modelamos un fragmento espacial con distribuciones de nanotubos de carbono habiendo algunos huecos de manera aleatoria debajo de donde la partícula se mueve, en la figura 4.5, se visualiza este ejemplo en un plano paralelo XY de la estructura de los CNT, con altura de los nanotubos Z como se muestra en la figura 4.5.

En la parte inferior (ver Figura 4.5) consideramos una nanoestructura periódica 10×6 CNTs paralelos al eje Z y \perp a la dirección de carga. Para simular un factor aleatorio asumimos que en algunas posiciones aleatorias de nanoestructura hay espacios vacios (con probabilidad $p = 0,2$) en lugar de una CNT. El radio externo del CNT de pared simple (SW) es $R = 0,35l = 175nm$, el radio interno es $R_i = 0,8R = 140nm$, la longitud es $h = 6l = 3,10^3nm$, y el tamaño del sistema es $L \cdot l = 5,10^4nm$, donde $l = 500nm$ es una escala espacial típica para dicha nanoestructura [63].

Dichos valores de los parámetros de dimensión son más útiles para el usuario en nuestras simulaciones numéricas, La CNT de pared simple (SWCNT) tiene una

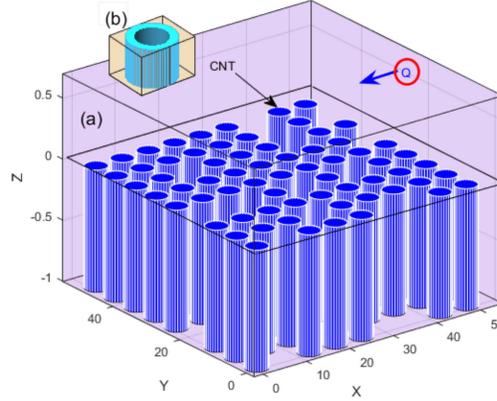


Figura 4.5: Arreglo de nanotubos CNT, resonando a una frecuencia ω_p y una carga que produce el efecto de la radiación de Cherenkov.

permitividad dieléctrica dispersiva de carbono $\varepsilon(\omega) = \varepsilon_h - \omega_p^2/(\omega^2 + i\gamma\omega)$, donde la permitividad de la matriz del host es $\varepsilon_h = 2$, ω_p es la frecuencia de plasma, $\gamma \ll \omega$ es el coeficiente de absorción de campo. (En este caso, la frecuencia del plasmón-polaritón de superficie es $\omega_p/(\varepsilon_h + 1)^{0,5}$).

La parte central (núcleo) de SWCNT se cumple con el mismo dieléctrico (Color violeta) con $\varepsilon = 1,5$ como parte superior del sistema. En nuestra simulación, normalizamos la velocidad de carga a la velocidad de la luz en el vacío c . Debido a un espectro de radiación de banda ancha de la carga en movimiento (ver Ecuación 4.3) estudiamos la propiedad de dicho sistema 3D compuesto (carga + nanotubos) por técnica **FDTD** en el dominio del tiempo.

Podemos observar la superficie del campo eléctrico E_x , que produce la radiación óptica de Cherenkov sobre la superficie de la nanoestructura conformada por los CNTs, una vez que la carga ha pasado por ese sistema. La carga que se mueve con velocidad uniforme $v = 0.866$ en el caso de SWCNT con frecuencia de plasma (a) $\omega_p = 10^{12} s^{-1}$ y (b) $\omega_p = 2,1 \times 10^{15} s^{-1}$. Ver figuras 4.6(a,b). Consideramos con detalles la geometría cuando la carga comienza a moverse en el tiempo $t = 0$ (desde el lado derecho, ver figura. 4.6) y estudiamos la dinámica del campo hasta el momento en que la carga abandona la muestra (el lado izquierdo) para diferentes valores ω_p de SWCNTs. La figura 4.6 muestra el campo generado en la intersección central del plano de muestra cuando la carga se mueve con velocidad $v = 0,866$ sobre la estructura CNT que tiene la figura 4.6 (a) $\omega_p = 10^{12} s^{-1}$ y la figura 4.6 (b) $\omega_p = 2,1 \times 10^{12} s^{-1}$ en $\gamma = 0,1\omega_p$. Se puede ver en la figura 4.6 (a) que por $\omega_p = 10^{12} s^{-1}$ bajo, el campo de CNT se excita principalmente cerca del área de la ruta de carga, pero para grandes $\omega_p = 2,1 \times 10^{15} s^{-1}$. En la figura 4.6 (b) los SWCNT excitados se encuentran en el frente de onda total de Cherenkov con $\cos(\theta) < v/nc$, donde $n = 1,22$ es el índice de refracción del medio transparente. Las amplitudes de campo que se muestran en la Fig. 4.6 (a), (b) son muy diferentes en diferentes ω_p . Por lo tanto, es de gran interés calcular la cantidad de energía radiada de la carga integrada a lo largo de la ruta de carga en función

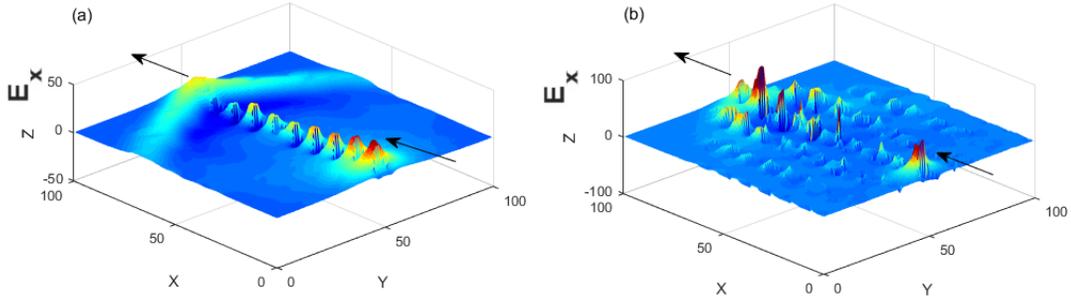


Figura 4.6: Radiación de Cherenkov en la superficie, con frecuencia de plasma (a) $\omega_p = 10^{12} s^{-1}$ y (b) $\omega_p = 2,1 \times 10^{15} s^{-1}$. Observamos que las amplitudes de los campos excitados en cada SWCNT dependen significativamente del valor ω_p . Para $\omega_p = 10^{12} s^{-1}$ bajo, el campo se excita principalmente dentro del área de la ruta de carga, panel (a), pero para $\omega_p = 2,1 \times 10^{15} s^{-1}$ mayor, (b) los SWCNT excitados se encuentran en el frente de onda total de Cherenkov. Otros parámetros son: el radio de los nanotubos es $R = 175 nm$, el radio interno es $R_i = 140 nm$, la longitud de los nanotubos es $h = 3,10^3 nm$ y el tamaño del sistema es $5 \times 10^4 nm$.

de ω_p para varias velocidades de carga diferentes v . (Para el caso sin dispersión y sin pérdidas, tal dependencia viene dada por la ecuación. 4.3). La figura 4.7 muestra una energía radiada en función de ω_p para varias velocidades de carga v . Tal dependencia se muestra en la figura. 4.7. Podemos distinguir en la figura 4.6(a,b) la distribución colectiva de plasmones-polaritones radiada sobre el campo eléctrico E_x . Observamos la altura que alcanza la energía radiada sobre el eje z , y observamos que para el caso 4.6(a) con $\omega_p = 10^{12} s^{-1}$ alcanza una altura de 50 mientras que para el caso 4.6(b) con $\omega_p = 2,1 \times 10^{15} s^{-1}$ alcanza una altura de 100, podemos darnos cuenta que existe una mayor colectividad de plasmones de superficie cuando la frecuencia es mayor.

4.5. Acoplamiento óptico entre carga y la nanoestructura CNT-3D

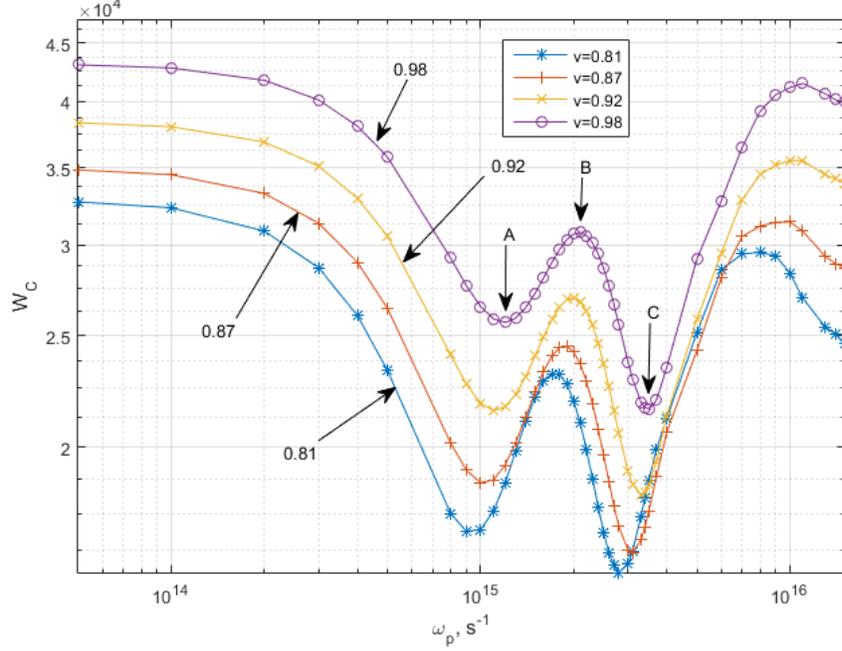


Figura 4.7: La energía radiada de la carga (unidades arbitrarias), integrada a lo largo de la ruta de la dirección de la carga en función de ω_p para diferente velocidad de carga v . En el área $\omega_p = 1 \times 10^{15}, 4 \times 10^{15}]s^{-1}$ observamos un patrón de energía de doble pozo con un máximo (indicado por el símbolo B) en $\omega_p = 2,1 \times 10^{15}s^{-1}$ y dos mínimos (indicados por los símbolos A, C) en $\omega_p = 1,2 \times 10^{15}s^{-1}$ y $\omega_p = 3,5 \times 10^{15}s^{-1}$ respectivamente para todas las velocidades. La posición de dicho extremo depende débilmente del valor de v_x . El pico corresponde a un fuerte régimen de acoplamiento entre el campo de carga radiante y el campo excitado en la estructura CNT.

Podemos ver en la figura 4.7 que la estructura CNT causa una diferencia drástica de la dependencia dada por la ecuación. 4.3. Observamos de la figura 4.7 que en la zona $1 \times 10^{15}s^{-1} < \omega_p < 4 \times 10^{15}s^{-1}$ un perfil de energía de forma de doble pozo emerge en tal dependencia. Existe un máximo (indicado por el símbolo B) en $\omega_p \simeq 2,1 \times 10^{15}s^{-1}$ y dos mínimos (indicados por los símbolos A, C) en $\omega_p \simeq 1,2 \times 10^{15}s^{-1}$ y $\omega_p \simeq 3,5 \times 10^{15}s^{-1}$ respectivamente que existen en un rango de carga con velocidades $v = 0,81, 0,87, 0,92, 0,98$.

Se puede ver que las posiciones de tales extremos dependen débilmente del valor de la velocidad v . Para analizar el significado del pico B en la figura 4.7 evaluamos el factor adimensional $G = (\Lambda_{SPP}/l)\kappa$ donde Λ_{SPP} es el plasmón-polariton wavelength, $\kappa = R/l$ es una relación entre los radios de un CNT y le valor de l es la escala espacial típica. Para $\omega_p \simeq 2,1 \times 10^{15}s^{-1}$, tenemos $\Lambda_{SPP} = 1,55\mu m$, eso para $R = 0,35l$ y $l = 500nm$ trae el valor $G \simeq 1$. Esto significa que para

$\omega_p \simeq 2,1 \times 10^{15} s^{-1}$ en los nanotubos acoplados se produce una resonancia: las excitaciones de plasmón-polaritón de superficie se generan de manera resonante por la carga en movimiento a la frecuencia resonante de la nanoestructura CNT $\omega_p/(\varepsilon_h + 1)^{0,5}$. El pico en $\omega_p \simeq 2,1 \times 10^{15} s^{-1}$ corresponde a un fuerte régimen de acoplamiento entre el campo de carga radiante y las excitaciones ópticas colectivas en una estructura CNT. Tal observación nos motiva a estudiar una estructura de campo excitado en 3D en una zona tan resonante con detalles.

Ahora estudiamos la distribución de campo 3D (carga + nanotubos) en el momento $t = t_e$ (donde t_e es el momento en que la carga toca la estructura de nanotubos) para diferentes valores ω_p de SWCNT. Como ya se mencionó, la frecuencia de los plasmones de superficie $\omega_p/\sqrt{\varepsilon_h + 1}$ define un límite en el espectro de frecuencia de los nanotubos conductores. Entonces, el rango del espectro de Cherenkov alrededor de $\omega_p/\sqrt{\varepsilon_h + 1}$ contribuye principalmente al espectro de campo excitado en nanotubos. Investigamos los patrones de campo 3D en la zona resonante $1 \times 10^{15} s^{-1} < \omega_p < 4 \times 10^{15} s^{-1}$ donde emerge un perfil de forma de doble pozo, ver figura 4.7.

4.6. Estructura 3D del campo óptico de la nanoestructura CNT

En la figura 4.8 muestra los patrones (corte de tomografía) del campo excitado en un sistema 3D que contiene una carga radiante (que se mueve con velocidad $v = 0,98$) y la nanoestructura CNT (ver figura 4.3) en varias intersecciones (de (a) a (i)) para el eje Z del sistema 3D en $\omega_p = 2,1 \times 10^{15} s^{-1}$, ver figura 4.8, flecha B . En todos los paneles de la figura 4.8 la carga se mueve de izquierda a derecha. Los paneles (a), (b) (parte superior del área de carga) e (i) (parte inferior del sistema CNT) exhiben el único frente de onda Cherenkov (que tiene grandes amplitudes) y no muestran el campo CNT debido a amplitudes muy pequeñas de este último en su periferia. Sin embargo, en los paneles centrales (d) - (g) observamos tanto el frente de onda de Cherenkov bien definido que está acoplado al campo de la CNT excitada en varios cortes (manchas de color).

Observamos que en la figura 4.8(f) la amplitud del campo CNT es prácticamente igual a la amplitud del frente de onda de Cherenkov y en dos veces mayor que las amplitudes CNT en los paneles vecinos (i) y (g). Tal aumento de campo corresponde al acoplamiento resonante del campo de la carga móvil al campo de la estructura CNT. Tal aumento de la amplitud del campo en la frecuencia máxima $2,1 \times 10^{15} s^{-1}$ también se confirma en la figura 4.6 (b), donde el aumento del campo resonante de la estructura CNT en se muestra una comparación con el caso de una frecuencia más baja en la figura 4.6(a). En la figura 4.9 se muestran las capas seccionadas para el valor en cortes de la altura Z , donde se encuentran orientado los nanotubos verticales, podemos apreciar que producen los valores (a) para $\omega_p = 1,2 \times 10^{15} s^{-1}$ y (b) para $\omega_p = 3,5 \times 10^{15} s^{-1}$, los valores mínimos del campo de energía de la figura 4.7.

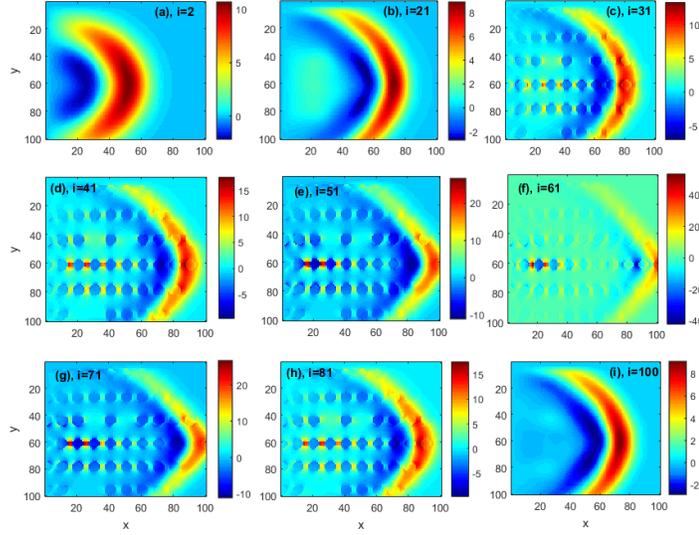


Figura 4.8: La estructura del componente de campo generado E_x (unidades arbitrarias) en un sistema 3D que contiene carga radiante (se mueve con velocidad $v = 0,98$) y la matriz de CNT. La estructura del campo $E_x(x, y, z_i)$ se muestra en varias capas (i -th cortes) del eje Z (de (a) a (i)) del sistema 3D para $\omega_p = 2,1 \times 10^{15} s^{-1}$ en la posición del pico del campo, ver figura 4.7, flecha B . El panel (a) muestra las amplitudes de campo en el área superior $i = 2$ de la carga móvil (ver figura 4.3), el panel (i) muestra el campo en el área inferior $i = 100$. La carga se mueve de izquierda a derecha. En los paneles (d) - (g) observamos la estructura del frente de onda de Cherenkov irradiado por la carga. Los puntos de color correspondientes al campo de los nanotubos conductores se ven en los paneles(c) - (h). Los valores de los campos se indican por las barras de colores. En los paneles (d) - (h) observamos el campo de interconexión de gran amplitud entre la CNT.

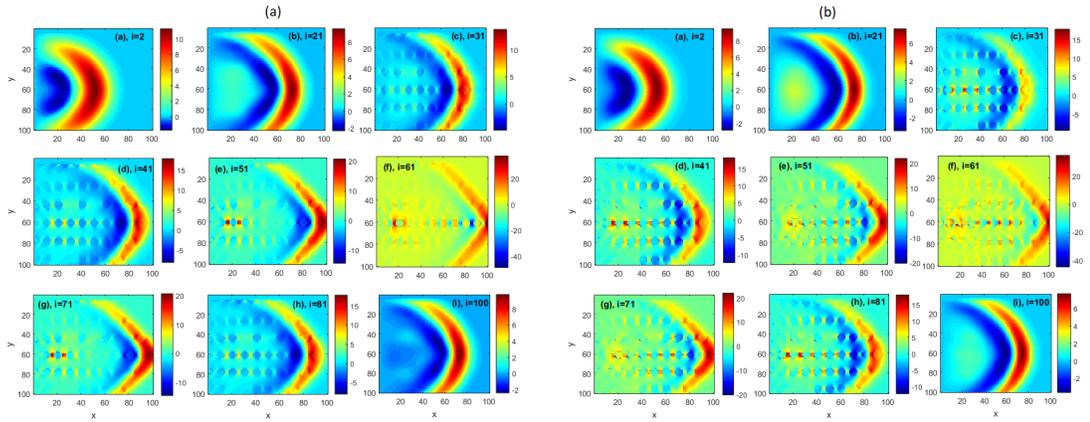


Figura 4.9: Se muestran los mismos cortes que en la figura 6, pero para (a) $\omega_p = 1,2 \times 10^{15} s^{-1}$ en la posición del campo izquierdo mínimo, y para (b) $\omega_p = 3,5 \times 10^{15} s^{-1}$ en la posición de mínimo del campo derecho, ver las flechas A y C de la figura 4.7 de los valores mínimos de los posos respectivamente.

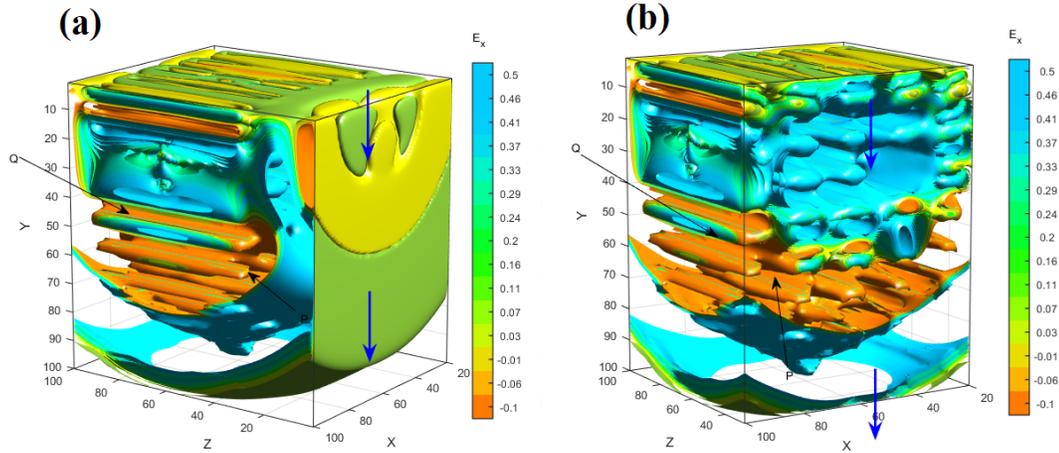


Figura 4.10: La imagen (a) muestra la distribución de campo 3D en E_x con la carga móvil (lado derecho) y los nanotubos (lado izquierdo) para la frecuencia de plasma $\omega_p = 2,1 \times 10^{15} s^{-1}$ que corresponde al pico de la figura 4.7 pico con flecha "B". La flecha azul muestra la dirección de entrada de la carga. La flecha P y Q inician el área del campo Cherenkov y el campo CNT respectivamente. En el centro (flecha Q) observamos la estructura periódica del campo 3D excitada en el área de los nanotubos. Se puede observar que en $\omega_p = 2,1 \times 10^{15} s^{-1}$ el campo de la estructura CNT tiene una forma periódica bien definida. La imagen (b), muestra la estructura de los nanotubos resonantes en el interior de la nanoestructura y bajo la superficie de donde pasa la carga.

La figura 4.9(a) muestra lo mismo que en la figura 4.8 pero para $\omega_p = 1,2 \times 10^{15} s^{-1}$ en la posición de la frecuencia baja, ver figura 4.7, flecha A con frecuencia menor. De la figura 4.9, observamos que a una frecuencia de plasma dada ω_p , la amplitud del campo en el plano central es mucho menor que en el caso que se muestra en la figura 4.8. La figura 4.9(b) muestra lo mismo que en la figura 4.8, con una frecuencia de plasma mucho mayor de $\omega_p = 3,5 \times 10^{15} s^{-1}$, que corresponde a un segundo mínimo (símbolo C) en la figura 4.7. Observamos también que la frecuencia de plasma dada ω_p , la amplitud del campo en el plano central es mucho menor que en el caso que se muestra en la figura 4.8. Esto confirma nuevamente la observación de que una transferencia de resonancia de energía de radiación desde una carga en movimiento a una nanoestructura periódica ocurre a la frecuencia $\omega_p = 2,1 \times 10^{15} s^{-1}$. A menudo es más informativo explorar la estructura completa del campo óptico en un medio 3D total. Las figuras 4.10(a,b) muestran el campo 3D de la distribución de campo E_x en dicho sistema compuesto (CNT + con carga móvil) para la frecuencia de plasma $\omega_p = 2,1 \times 10^{15} s^{-1}$ que corresponde al pico que se muestra en la figura 4.7 (símbolo B). Las flechas azules muestran las direcciones de entrada y salida de la carga (de arriba a abajo). En la parte inferior observamos el frente de onda de Cherenkov que se representa como una superficie de fase idéntica que toca la salida del sistema 3D. La flecha P y Q indican el área del campo Cherenkov y el campo CNT respectivamente. En el centro (flecha Q) observamos el patrón periódico del campo 3D excitado en el área de

los nanotubos a $\omega_p = 2,1 \times 10^{15} s^{-1}$ como se muestra en la figura 4.10(a), mientras que en la figura 4.10(b), se muestra un corte cerca de la superficie que visualiza los nanotubos resonantes que componen la nanoestructura.

Se puede concluir que sobre el espectro de energía de campo en E_x como una función de una frecuencia de plasma exhibe una forma clara de doble pozo no monótona que se puede observar en la figura 4.7. Se observa un pico de energía en una zona de excitaciones de plasmón-polaritón de superficie que representa un acoplamiento del campo de carga irradiada de Cherenkov a la estructura de nanotubos de la CNT, el punto de mayor acoplamiento y radiación para una frecuencia de nanotubos de carbono es en el punto de frecuencia $\omega_p = 2,1 \times 10^{15} s^{-1}$ con una velocidad de carga $v = 0,98$ para este estudio.

4.7. Radiación de Cherenkov sobre una nanoestructura de nanotubos con morfología aleatoria

Dado que hoy en día el nivel de ingeniería no permite crear estructuras CNT periódicas, consideramos una estructura de nanotubos con un factor aleatorio en las longitudes y radios para que nuestro estudio sea adecuado a la morfología real de CNT y su escala espacial espectroscópica y nano-métrica en un rango ω_p [66].

Como se mencionó anteriormente que es muy complicado construir nanotubos de estructuras uniformes es por eso que nuestro estudio trata de apearse lo más cercano posible a la morfología de los nanotubos reales que hoy en día se utilizan, para ello utilizamos el método Mote Carlo [49, 68, 70]. La idea es usar el valor random para el factor aleatorio en un rango de $\gamma \in [0, 1]$ para nuestras variables aleatorias $[a, b]$ donde $b > a$ y $h = a$ y $r = b$. Aplicamos el método Random para a la longitud de los nanotubos como $h = h_{min} + (h_{max} - h_{min})\gamma$ y el radio externo del nanotubo con $R = r_{min} + (r_{max} - r_{min})\gamma$ el radio interno del nanotubo es $R_i = 0,8R$. En nuestras simulaciones estandarizamos las medidas como se muestran a continuación: $l = 500nm$, $h_{max} = 6l = 3x103nm$, $h_{min} = 0$ y $r_{max} = 1,5R_0$, $r_{min} = 0,5R_0$, $R_0 = 0,35l = 175nm$.

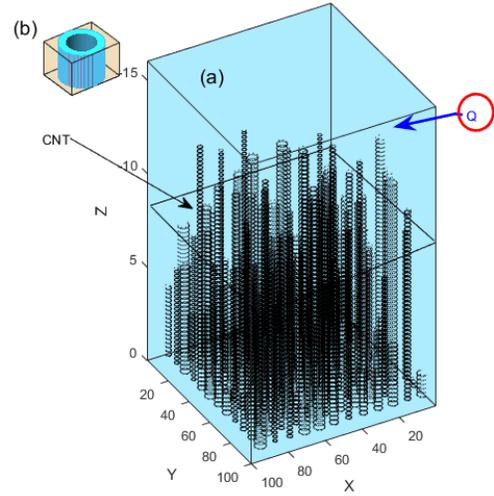


Figura 4.11: Patrón de CNTs paralelos con tamaños aleatorios (longitud y radio) en la parte inferior del sistema. La carga rápida se mueve paralela al plano XY con velocidad uniforme $v \parallel X$ sobre el nanotubo como se indica con la flecha Q ; (b) el recuadro muestra a detalle la morfología de un SWCNT (Nanotubo de pared blanda). [69].

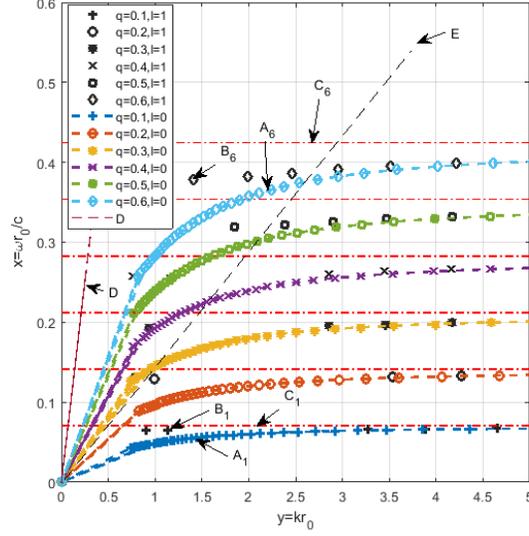


Figura 4.12: Muestra la solución a la ecuación 4.9 de dispersión con los parámetros de dependencia. La flecha $A_{1,6}$ y $B_{1,6}$ muestra la solución cuando $l=0$ y $l=1$. La flecha C muestra la frecuencia de la superficie del plasmon la flecha D corresponde a la onda de la luz $k = \omega/c$. La línea punteada con la flecha E se aproxima al espectro de densidad de energía radiada por la carga.

Otra manera de expresar la dispersión de plasma sobre cilindros de longitud y radio aleatorio para modos con pérdida en las ecuaciones de Maxwell con condiciones de la frontera con modos de la propagación electromagnética, se expresa en la siguiente ecuación [67].

$$\left[\frac{1}{\kappa_2} I_l(\kappa_1 r_0) K_l'(\kappa_2 r_0) - \frac{1}{\kappa_1} I_l'(\kappa_1 r_0) K_l(\kappa_2 r_0) \right] \times$$

$$\left[\frac{1}{\kappa_2} I_l(\kappa_1 r_0) K_l'(\kappa_2 r_0) - \frac{8}{\kappa_1} I_l'(\kappa_1 r_0) K_l(\kappa_1 r_0) \right] = \quad (4.9)$$

$$\frac{c^2 k_z^2 l^2}{\omega^2 r_0^2} I_l^2(\kappa_1 r_0) K_l^2(\kappa_2 r_0) \left(\frac{1}{\kappa_1} - \frac{1}{\kappa_2} \right)^2$$

Donde I_l y K_l son funciones modificadas de Bessel de primer y segundo orden, la prima denota una derivada con respecto al argumento $l = 0, 1, 2, \dots$, r_0 es el radio y:

$$\kappa_1^2 = k_z^2 - \frac{\omega^2}{c^2} \varepsilon(\omega), \quad \kappa_2^2 = k_z^2 - \frac{\omega^2}{c^2} \geq 0. \quad (4.10)$$

La figura 4.12, muestra la dependencia de dispersión de la ecuación 4.9. Dónde: $y = y(x)$, $x = \omega r_0/c$ e $y = kr_0$ con diferente valor de parámetro $q = \omega_p r_0/c$ y $l = [0, 1]$. En la figura 4.12, observamos que para grandes $y > 1$; los valores x alcanzan los valores $q/\sqrt{2}$ respectivamente, que corresponde a la frecuencia de

plasmón superficial en la superficie de CNT. Podemos ver que cuanto mayor es la energía irradiada por la carga en movimiento, más cercana está la frecuencia del nanotubo a la frecuencia del plasmón superficial.

De la figura 4.12, cuanto mayor es la energía irradiada por la carga en movimiento, más cercana está la frecuencia del nanotubo a la frecuencia del plasmón superficial. Sin embargo, la descripción correcta de la estructura real y la dinámica del campo óptico en un nano-sistema compuesto requiere la exploración de una nanoestructura compuesta completa con distribuciones tridimensionales y pérdidas ($\gamma_e \neq 0$) CNT, que se muestran en la figura 4.11. En las siguientes secciones, aplicamos la técnica numérica FDTD para estudiar dicho sistema 3D.

4.8. Aprovechamiento numérico - FDTD

Vale la pena señalar que el estudio FDTD del espectro de plasmón-polaritones de superficie en un sistema desordenado tridimensional es una tarea laboriosa, incluso para cálculos paralelos. Por lo tanto, en este estudio nos limitamos al régimen antes mencionado. La comparación de resultados para diferentes niveles de trastorno requiere expansiones significativas y el uso de diferentes distribuciones estadísticas. Dicha investigación se llevará a cabo en el futuro. En tales nanotubos desordenados con parámetros aleatorios (ver figura 4.11), se produce la generación de oscilaciones locales (no homogéneas) de los campos 3D de superficie de plasmón-polaritón. Dichos campos tienen una forma complicada y una simetría axial que difiere significativamente con respecto al caso de los nanotubos periódicos estudiado en nanotubos con estructura periódica. Consideramos SWCNT con permitividad dieléctrica dispersiva similar a la de los metales Drude $\varepsilon(\omega) = \varepsilon_h - \omega_p^2/(\omega^2 + i\gamma_e\omega)$, donde $\varepsilon_h = 2$ es la permitividad dieléctrica de la matriz del anfitrión, ω_p es la frecuencia de plasma y $\gamma_e \ll \omega_p$. Para la superficie de plasmón-polaritón la frecuencia es $\omega_p/\sqrt{\varepsilon_h + 1}$. La parte central (núcleo) de SWCNT se cumple con el mismo dieléctrico con $\varepsilon = 1,5$ como la parte superior del sistema.

En la figura 4.13 podemos observar el campo colectivo de plasmones-polaritones de superficie que produce la radiación de Cherenkov del campo excitado de los nanotubos, podemos ver que las amplitudes varían dependiendo la frecuencia de plasma que se le aplique a cada nanoestructura en el caso de la imagen (a) con frecuencia de plasma; $\omega_p = 6 \times 10^{14} s^{-1}$, observamos una amplitud visible en forma de cono con altura máxima de 60 sobre el campo E_x , mientras que para el caso (b) con frecuencia de plasma; $\omega_p = 4,1 \times 10^{14} s^{-1}$, se alcanza un mayor número colectivo de plasmones de superficie con una altura de 100 sobre el campo eléctrico E_x [71, 72]. En la figura 4.13, podemos observar el cono característico que produce la onda de la radiación de Cherenkov sobre la superficie de las nanoestructuras para el caso de la figura 4.13(a), muy diferente a la forma de onda que se plasma en la figura 4.13(b), que es distorsionada debido a que existe una mayor frecuencia de plasma mucho mayor en las nanoestructuras para este último caso. Dicho com-

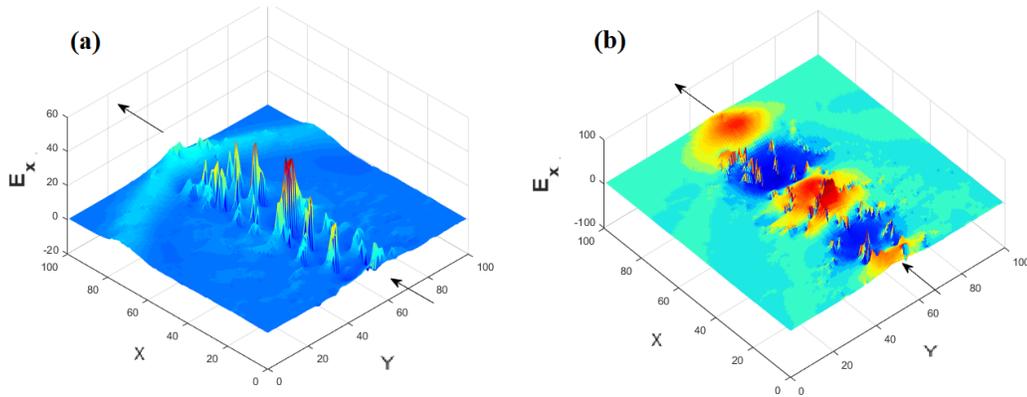


Figura 4.13: El campo Cherenkov E_x (unidades arbitrarias) en el plano central generado por una carga que se mueve con velocidad uniforme $v = 0,866$ en el caso de SWCNT con frecuencia de plasma (a) $\omega_p = 6 \times 10^{14} s^{-1}$ y (b) $\omega_p = 4,1 \times 10^{14} s^{-1}$. Observamos que las amplitudes de los campos excitados en cada SWCNT dependen significativamente del valor ω_p . Para $\omega_p = 6 \times 10^{14} s^{-1}$ el campo se excita principalmente dentro de la ruta de carga, pero para $\omega_p = 4,1 \times 10^{15} s^{-1}$ (b), hay pequeños huecos cóncavos y picos de grupos de campos SWCNT excitados en la onda Cherenkov.

portamiento de plasma nos motiva a estudiar un rango de diferentes frecuencias aplicada a la nanoestructura.

4.9. Estudio de frecuencias de plasma ω_p y acoplamiento óptico sobre nanotubos con estructura aleatoria

Nuestro siguiente objetivo es saber cómo se comporta la emisión de plasmones de superficie cuando variamos el valor de la frecuencia ω_p (frecuencia que se aplica a los nanotubos) para un determinado rango de frecuencias, para esto estudiamos un espectro amplio de frecuencias y así encontrar el punto de frecuencia en donde se emite más energía. Estudiamos un rango de frecuencias, tomando como dato principal que la magnitud del campo cambia en relación con la frecuencia de los CNTs, como se observa en los dos ejemplos de la figura 4.13(a, b), realizamos varias pruebas para calcular el campo eléctrico en un rango de frecuencia para ω_p de $\omega_p = [1 \times 10^{14} \text{ hasta } 5 \times 10^{16}] s^{-1}$ para 4 velocidades diferentes de la carga ($v_1 = 0,81$, $v_2 = 0,87$, $v_3 = 0,92$ y $v_4 = 0,98$) para la radiación de Cherenkov. En estas pruebas, el objetivo es encontrar la frecuencia ω_p , donde el campo es mayor y, por lo tanto, más acoplado con la radiación de Cherenkov. Encontramos la frecuencia donde se genera un mayor acoplamiento entre las CNT y la carga está en $\omega_p = 6 \times 10^{14} s^{-1}$ que se indica con la flecha B , mientras que en el rango estudiado el acoplamiento más pequeño se da en el punto mínimo para $\omega_p = 4,1 \times 10^{14} s^{-1}$, en la figura. 4.14 muestra los resultados de esta investigación. Investigamos un amplio rango de frecuencia donde se ubicaron mayores cambios en la radiación de plasmones, el rango de frecuencia con mayores cambios se pue-

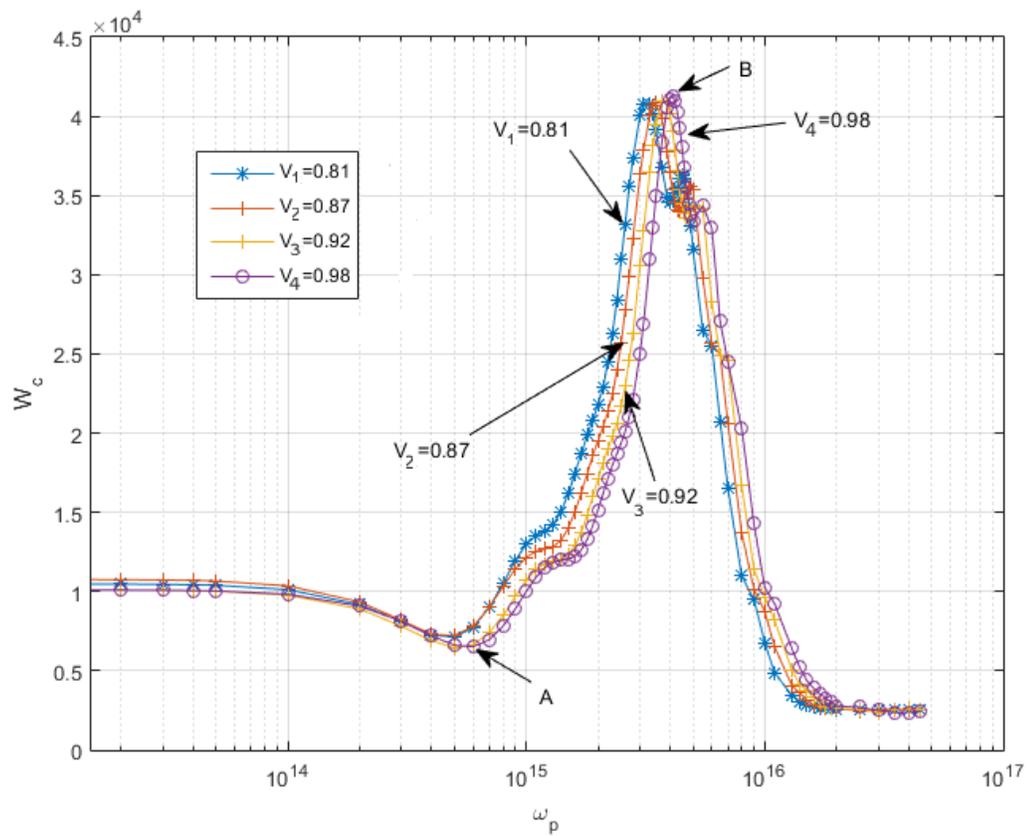


Figura 4.14: La energía radiada integrada (unidades arbitrarias) en toda ruta de la dirección que sigue la carga (path) en función de ω_p a diferentes velocidades de carga v . En el área $\omega_p = 1 \times 10^{14}, 5 \times 10^{16}] s^{-1}$ observamos un patrón de energía con el máximo nivel de energía (indicado por el símbolo B) en $\omega_p = 4,1 \times 10^{15} s^{-1}$ y un mínimo de energía inicial A en $\omega_p = 6 \times 10^{14} s^{-1}$ respectivamente para todas las velocidades. La posición de dicho extremo depende débilmente del valor de v_x . El pico corresponde a un fuerte régimen de acoplamiento entre el campo de carga radiante y el campo excitado en la estructura CNT.

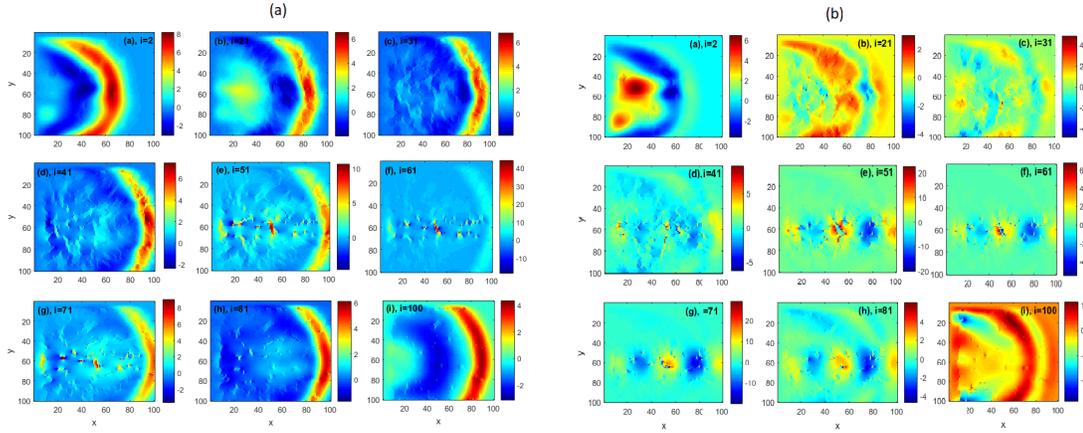


Figura 4.15: Estructura del componente del campo generado E_x (unidades arbitrarias) en un sistema 3D que contiene carga radiante (moviéndose con velocidad $v = 0,98$) y la matriz de CNT. La estructura del campo $E_x(x, y, z_i)$ se muestra en varias capas (i -th cortes) del eje Z (de (a) a (i)) para $\omega_p = 6 \times 10^{14} s^{-1}$ en la posición del campo mínimo, ver figura 4.14, flecha A. El panel (a) muestra las amplitudes de campo en el área superior $i = 2$ de la carga móvil (ver figura 4.14), el panel (i) muestra el campo en el área inferior $i = 100$. La carga se mueve de izquierda a derecha. En los paneles (d) - (g) observamos la estructura del frente de onda de Cherenkov irradiada por la carga. Los puntos de color correspondientes al campo de los nanotubos conductores se visualizan en todos los planos (c) - (h). Los valores de los campos están indicados por las barras de colores. En los paneles (d)-(h) observamos el campo de interconexión de gran amplitud entre la CNT.

de observar en la figura 4.14, que va desde $\omega_p = \omega_p = 6 \times 10^{14} s^{-1}$ donde hay un ligero sedimento bajo (Punto inicial mínimo en la que la energía asciende), alcanza un máximo nivel de energía en donde la generación de plasmones es mayor en el rango $\omega_p = 4,1 \times 10^{15} s^{-1}$ y vuelve a decaer en el valor de la frecuencia del valor $\omega_p = \omega_p = 1 \times 10^{16} s^{-1}$. Para valores de frecuencia posteriores se sigue una relación sin cambios de energía.

Observamos que la resonancia del campo de la estructura CNT aumenta en comparación con frecuencias más bajas y frecuencias más altas que el valor de $\omega_p = 4,1 \times 10^{15} s^{-1}$, valor donde hay una mayor dispersión de energía y un mayor acoplamiento entre la relación (carga + CNT). Consideramos importante saber cuál es el comportamiento dentro de los puntos **A** y **B**, punto mínimo ascendente de energía y punto máximo de energía y de acoplamiento óptico entre la carga y el sistema de nanotubos es. Es por ello que en la figura 4.15(a,b) observamos las tomografías para el caso de los puntos **A** y **B**.

En los paneles (a-i) de la figura 4.15(a, b), se muestran las secciones transversales del campo E_x del punto mínimo de frecuencia $\omega_p = 6 \times 10^{14} s^{-1}$, se muestran los campos de la mano-estructura de CNT irradiados en profundidad Z (orientación de los nanotubos), donde (a = 2) está cerca de la superficie, (e = 51) muestra un corte en el medio de la estructura de los nanotubos y (i = 100) está cerca de la base donde se encuentran las CNT. Podemos ver que hay una mayor distorsión

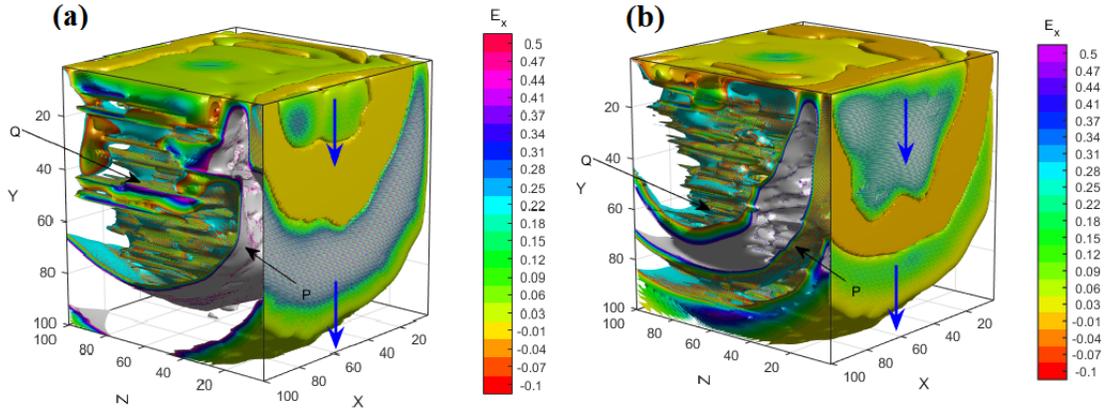


Figura 4.16: Distribución de campo 3D E_x (a) sistema compuesto con la carga móvil y nanotubos para la frecuencia de plasma $\omega_p = 6 \times 10^{14} s^{-1}$ que corresponde al valor mínimo de la figura 4.14 (flecha A). La flecha azul muestra la dirección de entrada de la carga. Las flechas P y Q indican el área del campo Cherenkov y el campo CNT respectivamente. En el centro (flecha Q) observamos la estructura periódica del campo 3D excitada en el área de los nanotubos. Se puede ver que en tal $\omega_p = 6 \times 10^{14} s^{-1}$ el campo de la estructura CNT tiene una forma periódica bien definida. Par el caso (b) observamos las mismas caracterizas pero correspondientes al punto B al valor máximo de la figura 4.14 (flecha B), punto donde se emite mayor energía de plasmones-polaritones dentro de la nanoestructura con frecuencia $\omega_p = 4,1 \times 10^{15} s^{-1}$.

del campo entre los nanotubos para los valores de las secciones $c - g$ ($i = 31$ a 71) para ambos casos de la figura (a, b) de los niveles bajo y máximo de energía de las letras A y B de la figura 4.14. Podemos ver en la barra de color de cada figura el rango de intensidad alcanzado en cada uno de los cortes para cada valor (i), donde el valor para la sección (f) tiene un mayor intensidad (intensidad de color = 40) en comparación con las otras secciones para el caso de la figura 4.15(a), mientras que para el caso de la figura 4.15(b), el mismo corte con frecuencia alcanza un valor de intensidad máximo de 100.

Para comprender mejor la distribución del campo, unimos todas las secciones o capas de los valores $i = 1$ hasta $i = 100$ para generar una imagen 3D que representa el campo completo en profundidad de los nanotubos Z , y la expansión del campo E_x que se produce en el interior de los nanotubos. La imagen 3D de la figura 4.16(a), muestra el valor mínimo (flecha A) que se muestra en el gráfico de la figura 4.14, donde podemos observar una baja dispersión de plasmón-polaritón para esta frecuencia $\omega_p = 6 \times 10^{14} s^{-1}$, donde Q representa la dirección de la carga que cruza la estructura de los nanotubos XY y paralela al eje Z , podemos observar formas de campo de los CNTs verticales y de diferente espesor, debajo La ruta de carga. Observamos lo mismo para el caso 4.16(b) con el punto máximo de mayor acoplamiento de la figura 4.14 (flecha B), con frecuencia mayor de $\omega_p = 4,1 \times 10^{15} s^{-1}$. En la figura 4.16(b), la imagen en 3D se muestra para el punto máximo de plasmones-polaritones y la dispersión del campo E_x correspondiente al punto máximo $\omega_p = 4,1 \times 10^{15} s^{-1}$ del gráfico de la figura 4.14, donde podemos

observar el campo de nanotubos para los CNTs con de diferente radio y longitud. Este valor representa el mayor acoplamiento y una mejor dispersión de energía del campo E_x entre la radiación que emite la carga de Cherenkov y la frecuencia ω_p emitida por los CNTs.

4.10. Comparación de CNTs frente a Nanoestructuras de TiO_2

Como se mencionó en el capítulo 2, dadas las condiciones tecnológicas es muy complicado crear nanoestructuras periódicas. Los CNT que se producen comúnmente en laboratorio que tienen radios y longitudes diferentes (Ver figura 4.17). Dentro del mercado industrial existen otra variedad de nanoestructuras: nanotubos de TiO_2 (Dióxido de Titanio) con importantes capacidades conductoras, es por ello que decidimos incluir estas nanoestructuras a nuestra simulación. Ver figura 4.18. Las nanoestructuras a base de grafeno (CNT), pueden tener oscilaciones

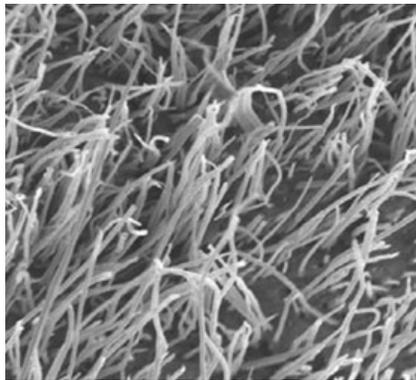


Figura 4.17: Imagen real tomada con Micrografías SEM (Scanning electron microscope) que muestran superficies de fractura de compuestos de CNT / epoxy alineados después de la prueba de tracción [91]

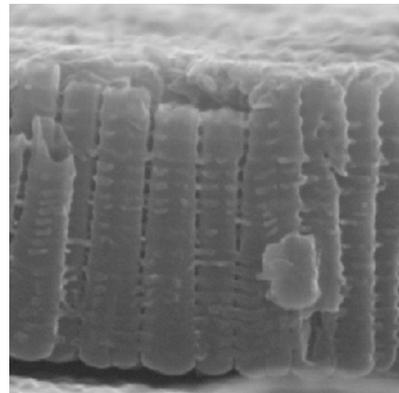


Figura 4.18: Nanoestructuras de dióxido de Titanio (TiO_2) imagen de microscopio de barrido. Muestra el costado de los nanotubos de TiO_2 , con forma de bambu bien definida.

superiores a los de 1×10^{13} , mientras que las nanoestructuras de TiO_2 , generalmente solo operan en frecuencias relativamente bajas, iguales o menores a 1×10^{10} , en nuestro interés por saber qué tipo de nanoestructura tubular resulta ser más eficiente, realizamos varias pruebas en los rangos de frecuencia para las nanoestructuras, 1×10^3 (a partir de kilohertz-KHz) hasta 1×10^{10} para las estructuras de dióxido de titanio, mientras que para el caso del CNT ocupamos frecuencias de plasma mayores o iguales a 1×10^{14} hasta $1,2 \times 10^{16}$ para encontrar que tipo de nanoestructura permite mejores transmisión de señales en modo de plasmones-polaritones. Estudiamos dos dependencias principales; la primera muestra una dependencia de frecuencias distribuidas en un intervalo de velocidades (ver figura

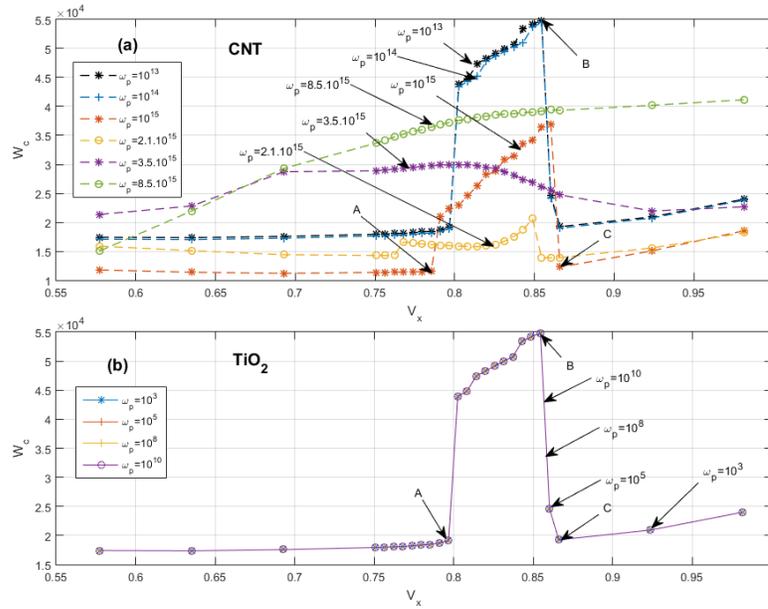


Figura 4.19: Energía que irradia la carga con dependencia de la velocidad de la carga para diferentes frecuencias de plasma; $\omega_p = [10^{13}, 10^{14}, 10^{15}, 2,1 \times 10^{15}, 3,5 \times 10^{15}, 8,5 \times 10^{15}]$ para los nanotubos de CNT's (a) y $\omega_p = [10^3, 10^5, 10^8, 10^{10}]s^{-1}$ para el caso (b) TiO_2 , en un intervalo de velocidades $v=0.5$ hasta $v=0.98$. Dependencia de velocidad v para ambos nanoestructuras.

4.19), mientras que la segunda dependencia estudiamos algunas velocidades distribuidas en un rango de frecuencias (ver figura 4.19). El rango de velocidades de la carga que utilizamos fueron desde una velocidad inicial $v=0.5$ hasta $v=0.98$ para frecuencias bajas $\omega_p = [10^3, 10^5, 10^8, 10^{10}]s^{-1}$ para el caso de nanotubos de TiO_2 y para frecuencias altas $\omega_p = [10^{13}, 10^{14}, 10^{15}, 2,1 \times 10^{15}, 3,5 \times 10^{15}, 8,5 \times 10^{15}]s^{-1}$ para el caso CNT's, ver figura 4.19(a). Podemos observar que para frecuencias pequeñas $\omega_p = [10^3, 10^5, 10^8, 10^{10}]s^{-1}$, caso TiO_2 , (Ver figura 4.19(b)), no existe variación en la radiación de energía, caso contrario para frecuencias grandes $\omega_p = [10^{13}, 10^{14}, 10^{15}, 2,1 \times 10^{15}, 3,5 \times 10^{15}, 8,5 \times 10^{15}]$. Donde existe una familia más variada con respecto a la familia de frecuencias ($\omega_p > 10^3$) que tomamos para estas pruebas como se observa en la figura 4.19(a). Observamos que ambos casos de la figura 4.19(a) y (b), coinciden con la misma relación de energía radiada para ambos casos de nanoestructuras. En la figura 4.20 y 4.21, observamos las tomografías en profundidad y la estructura óptica 3D del valor máximo irradiado para las nanoestructuras de Dióxido de Titanio TiO_2 con frecuencia $\omega_p = 10^{10}s^{-1}$ con una velocidad de $v = 0,85$, punto máximo indicado por la flecha B de la figura 4.19(b). La figura 4.20 muestra los paneles en vistas de tomografías de las nanoestructuras en profundidad desde $i=2$ hasta $i=100$, mientras que la figura 4.21 muestra la resonancia 3D de los nanotubos.

En ocasiones es preferible estudiar una dependencia de velocidades sobre un rango de frecuencias, ya que brinda una mejor precisión de un punto mayor de energía radiada, esta relación la apodemos observar en la figura 4.22, en donde se

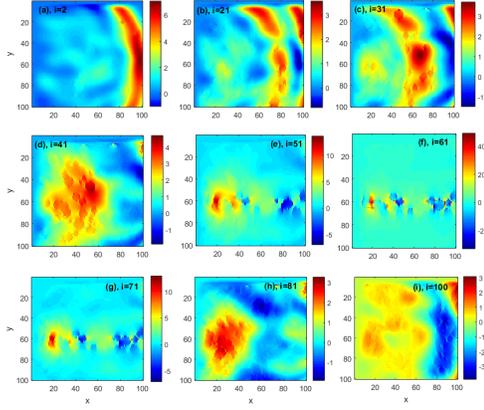


Figura 4.20: Estructura interna de los nanotubos desde la superficie hasta la base etiquetados desde $i = [2, 21, 31, 41, 51, 61, 71, 81, 100]$, en las que podemos observar el campo óptico que generan la radiación de la partícula para cada sección transversal.

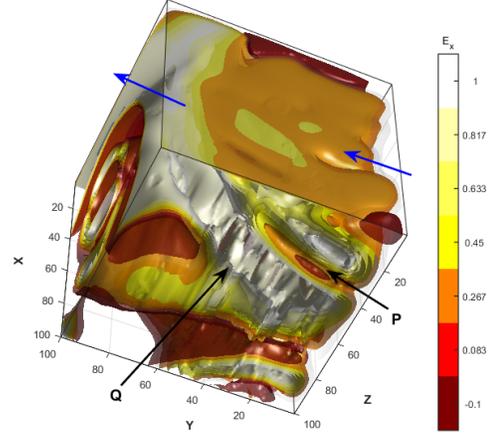


Figura 4.21: Estructura Óptica 3D con $\omega_p = 10^{10} s^{-1}$, cuando la carga que produce la radiación de Cherenkov va a una velocidad $v=0.85$. Punto máximo indicado por la flecha **B** de la figura 4.19(b).

traza una línea puntada divisoria separando el ancho de banda de las frecuencias, en lado izquierdo encontramos un intervalo de frecuencias para nanoestructuras de TiO_2 y lado derecho, frecuencias para los $CNTs$. Claramente podemos observar una mayor generación de energía W_c sobre la superficie mayor en los nanotubos de carbono cuando se varía la frecuencia de los nanotubos ω_p . En nuestro aprovechamiento numérico utilizamos una saturación de 200 nanotubos sobre nuestra malla numérica de $CNTs$, cantidad aproximada al arreglo de los nanotubos de TiO_2 creadas en laboratorio para nuestros fines.

En la figura 4.23 podemos observar los cortes en profundidad del campo E_x paralelo al eje Z , correspondientes al punto de mayor acoplamiento de los nanotubos CNT con la carga radiante al punto máximo de energía de la gráfica de la figura 4.22, indicada por la flecha *A* con frecuencia $\omega_p = 1,2 \times 10^{16} s^{-1}$.

En la figura 4.23 observamos que el panel (f) aproximadamente que a la mitad de la altura de los paneles correspondientes a la mitad de la nanoestructura alcanza un campo de mayor intensidad reflejada en la barra de colores, con un valor igual a 40. En la figura 4.24, observamos la radiación de superficie que emite la carga sobre la nanoestructura del pico máximo de radiación de energía (indicada por la flecha *A* de la figura 4.22). Podemos observar también la amplitud del campo E_x , que deja el campo óptico producido por la carga que pasa sobre los nanotubos (picos que sobresalen de la superficie de los nanotubos) y que viaja a una velocidad $v=0.866$ con un índice de refracción $n = 1,22$ para los cuatro casos pero con una frecuencia de $\omega_p = 1,2 \times 10^{16} s^{-1}$. En la figura 4.24, podemos observar el campo indicado también por la barra de colores E_x , que indica un valor aproximado de 140, valor que alcanza la altura máxima irradiada de plasmones de superficie

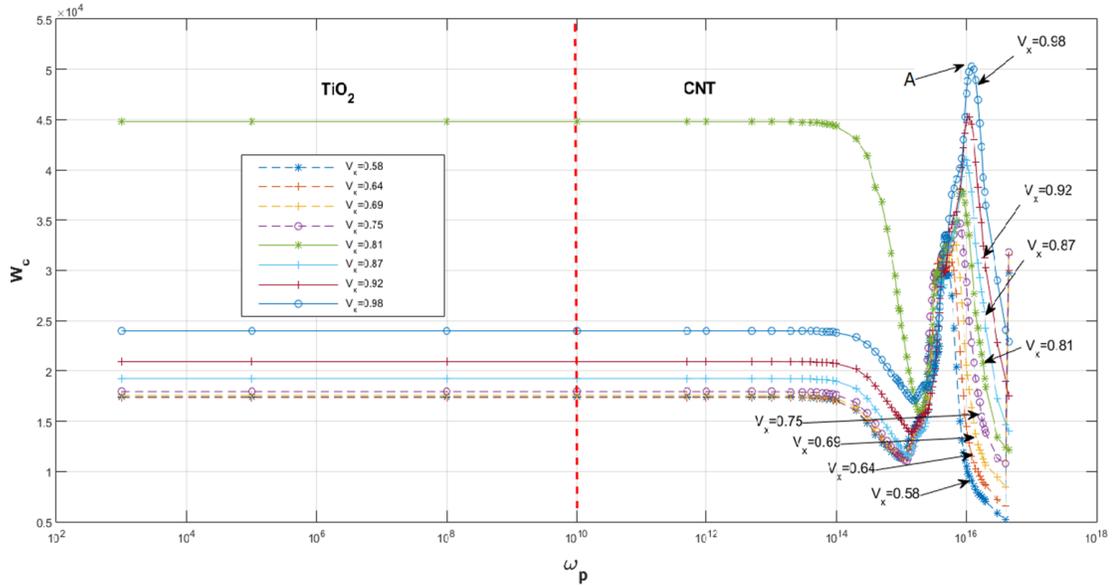


Figura 4.22: Podemos observar en primera instancia que para las frecuencias ω_p para los nanotubos de TiO_2 , existe una alta concentración de energía constante para todo el rango de familias de $\omega_p = 10^{10} s^{-1}$ cuando la velocidad es igual a $v = 0,81$ (Línea de color verde). Caso contrario para los CNTs donde existe un cambio considerablemente importante de energía cuando la carga va a una velocidad $v=0.98$ (Línea de color azul) en un pico máximo de $\omega_p = 1,2 \times 10^{16} s^{-1}$ indicada por la flecha **A**.

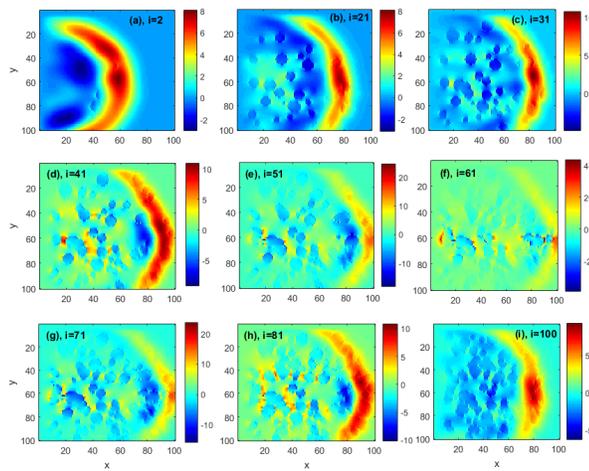


Figura 4.23: Tomografías en forma de paneles (a-i) que muestran los cortes de altura del campo E_x en altura y paralelo al eje Z para el pico **A** de la figura 4.22, podemos ver que la tomografía, con más intensidad es el panel ($i=61$ correspondiente al panel (f)).

sobre el eje de la altura \mathbf{Z} . Mientras que en la figura 4.25, podemos ver el comportamiento interno de la estructura de los nanotubos CNT del punto máximo de energía de la figura 4.22 (indicada por la flecha A de la figura) con frecuencia $\omega_p = 1,2 \times 10^{16} s^{-1}$, podemos observar un plasma con pequeñas formaciones de plasma en forma de nanotubos paralelas al eje \mathbf{Z} de la amplitud del campo E_x , donde las flecha azul indica la dirección de la carga radiante.

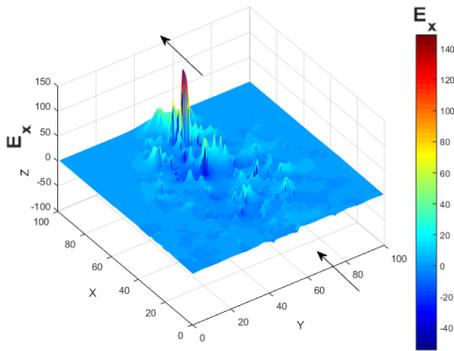


Figura 4.24: Cono de Cherenkov producido por la carga sobre los nanotubos con velocidad $\mathbf{v}=\mathbf{0.866}$ y frecuencia $\omega_p = 1,2 \times 10^{16} s^{-1}$. (Pico máximo de radiación de energía (indicada por la flecha A de la figura 4.22).

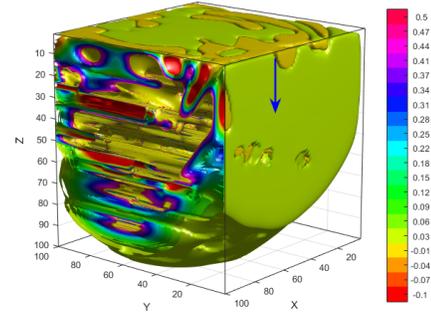


Figura 4.25: Amplitud del campo E_x en el interior de la estructura CNT con nanotubos aleatorios, con frecuencia: $\omega_p = 1,2 \times 10^{16} s^{-1}$ (pico máximo de radiación de energía (indicada por la flecha A de la figura 4.22).

4.10.1. Orientaciones de las nanoestructuras

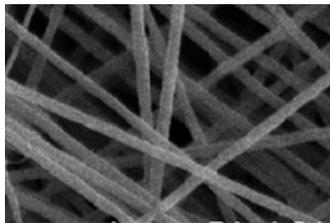


Figura 4.26: Microscopía electrónica que visualiza las fibras de CNT con diámetros de 100-200 nm. [84]

por ello que nuestro siguiente objetivo es abordar este tipo de morfología cruzada (en red), para conocer el comportamiento entre la nanoestructura resonante en red y la carga que produce la radiación de Cherenkov [73–83].

Las estructuras estudiadas hasta ahora (Morfologías periódicas y aleatorias) tienen una dirección vertical y una orientación paralela al eje \mathbf{Z} , sin embargo, existen otras variedades de arreglos en las nanoestructuras, una morfología muy comúnmente conocida es la morfología de red (Similares a la imagen de la figura 4.26), en la que los nanotubos forman una red horizontal a lo largo del arreglo (\mathbf{XY}) con intercepciones y contactos entre ellos. Es

4.11. Radiación de Cherenkov sobre una malla de nanotubos resonantes

La malla compuesta por CNT con morfología de red se encuentra muy comúnmente en la creación de nanoestructuras [84–92], por lo que nuestra investigación proporciona información que puede ser de gran interés científico y tecnológico en pruebas reales considerando este tipo de morfología en la CNT's. Analizando esta consideración, podemos observar que los nano-compuestos forman una malla con contacto mecánico entre ellos que genera una conducción inminente de energía a través de la malla CNT [94] que no había sido estudiada en investigaciones previas [95,96].

Consideramos una nanoestructura periódica 10×6 CNTs paralelos al eje XY y \perp a la dirección de carga. El radio externo de los CNT de pared simple (SW) es $R = 0,35l = 175nm$, el radio interno es $R_i = 0,8R = 140nm$, la longitud es $h = 6l = 3,10^3nm$, y el tamaño del sistema es $L \cdot l = 5,10^4nm$, donde $l = 500nm$ es una escala espacial típica para dicha nanoestructura.

Dichos valores de los parámetros de dimensión son más útiles para el usuario en nuestras simulaciones numéricas. La CNT de pared simple (SWCNT) tiene una permitividad dieléctrica dispersiva de carbono $\varepsilon(\omega) = \varepsilon_h - \omega_p^2/(\omega^2 + i\gamma\omega)$, donde la permitividad de la matriz del host es $\varepsilon_h = 2$, ω_p es la frecuencia de plasma, $\gamma \ll \omega$ es la coeficiente de absorción de campo. (En este caso, la frecuencia del plasmón-polaritón de superficie es $\omega_p/(\varepsilon_h + 1)^{0,5}$). Para este caso se estudió nanotubos periódicos en longitud y grosor, es decir, nanotubos uniformes pero con diferente orientación. La parte central (núcleo) de SWCNT se cumple con el mismo dieléctrico (Color rosa) con $\varepsilon = 1,5$ como parte superior del sistema. En nuestra simulación, normalizamos la velocidad de carga a la velocidad de la luz en el vacío c [97]. Debido a un espectro de radiación de banda ancha de la carga en movimiento (ver Ecuación 4.3). Podemos observar en la figura 4.28, también la emisión de energía radiada sobre la superficie de la nanoestructura de CNTs y encontramos un distribución muy diferente de energía a la de los casos anteriores, donde los nanotubos tenían una orientación hacia el eje Z(nanotubos verticales). Podemos ver en la figura 4.28 los picos de energía del campo E_x generados en

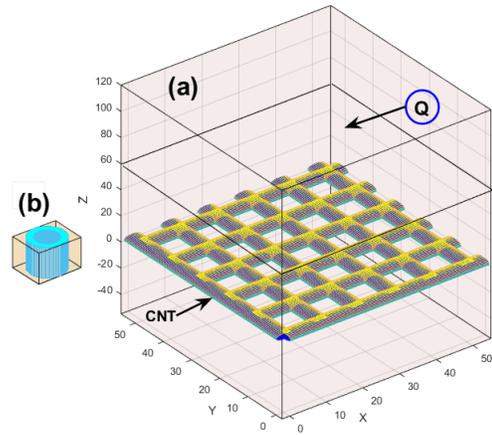


Figura 4.27: (a) Sistema de simulación 3D de nanotubos + carga. Donde los nanotubos muestran la morfología con forma de red (plano XY) y la carga rápida (flecha Q) que se mueve por encima de la red de CNT. (b) muestra la morfología del SWCNT.

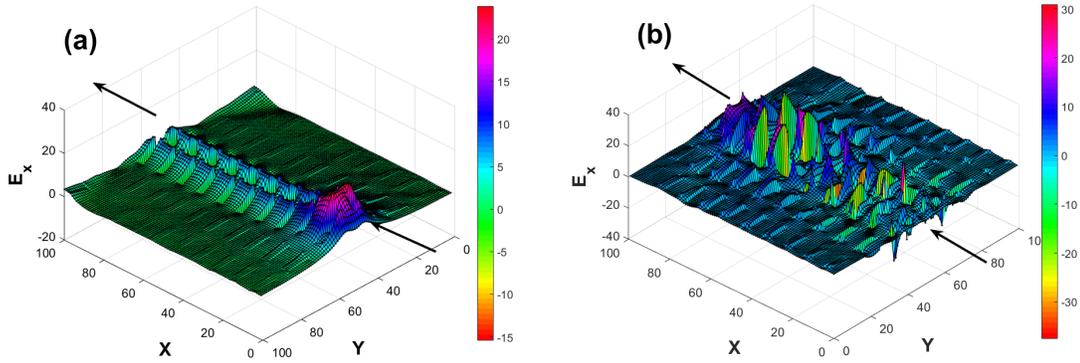


Figura 4.28: Podemos observar el campo óptico (Mediante el método **FDTD** [98]) que produce la carga en la superficie de la nanoestructura (Red de CNTs). Las flechas indican la entrada y salida de la carga. Para el caso (a), tenemos una frecuencia de $\omega_p = 1 \times 10^{13}$ y para el caso (b) $\omega_p = 5 \times 10^{15}$ con una velocidad de carga para ambos casos de $v = 0,75$

la superficie después de que la carga ha pasado sobre la superficie de la nanoestructura y ha alcanzado los límites de nuestra simulación. Las flechas indican la entrada y salida de la carga moviéndose a una velocidad uniforme hasta que sale del sistema simulado. Para el caso de la Figura 4.28 (a) la malla CNT se excita a una frecuencia de $\omega_p = 1 \times 10^{13}$, observamos una ruta bien definida de picos de plasmones-polaritones que deja la radiación Cherenkov en la estructura CNT en el centro de la superficie del sistema. Para el caso de la Figura 4.28 (b) la malla CNT se excita a una frecuencia de $\omega_p = 5 \times 10^{15}$ donde observamos picos bien definidos en el centro del sistema con pequeños picos de energía a los lados que acompañan a la ruta principal donde pasa la carga, observamos que los SWCNT excitados se encuentran en el frente de onda de Cherenkov donde el índice de refracción en el medio transparente es $n = 1,22$. Podemos ver una diferencia notable entre los casos (a) y (b) debido a la diferencia en sus frecuencias. Estos cambios significativos de energía para el mismo sistema con velocidad de carga igual a $v = 0,75$ pero con diferente frecuencia nos motiva a estudiar la energía radiada para diferentes valores de ω_p y diferentes velocidades de carga.

Podemos observar una uniformidad en la radiación de energía para todas las velocidades de carga, excepto cuando la carga tiene una velocidad de $v = 0.75$, donde podemos observar claramente un aumento considerable de energía en el punto de frecuencia $\omega = 1 \times 10^{13}$ que disminuye mientras aumenta la frecuencia. Este significativo pico de energía nos lleva a estudiar en detalle las velocidades asociadas a él, ya que este fenómeno no se consideró anteriormente. Las figuras 4.29 (a) y 4.29 (b) muestran algunos cortes en forma de tomografías para el campo E_x dentro del sistema 3D correspondiente a las Figuras 4.29(a) y 4.29(b) con frecuencias de $\omega_p = 1 \times 10^{13}$ y (b) $\omega_p = 5 \times 10^{15}$ respectivamente. Para ambos casos tenemos cortes del efecto que produce la radiación Cherenkov en el sistema donde se encuentra ubicada la malla de nanotubos, tenemos cortes que van desde $i = 2$, corte más cerca de la superficie hasta $i = 100$, corte más profundo del sistema.

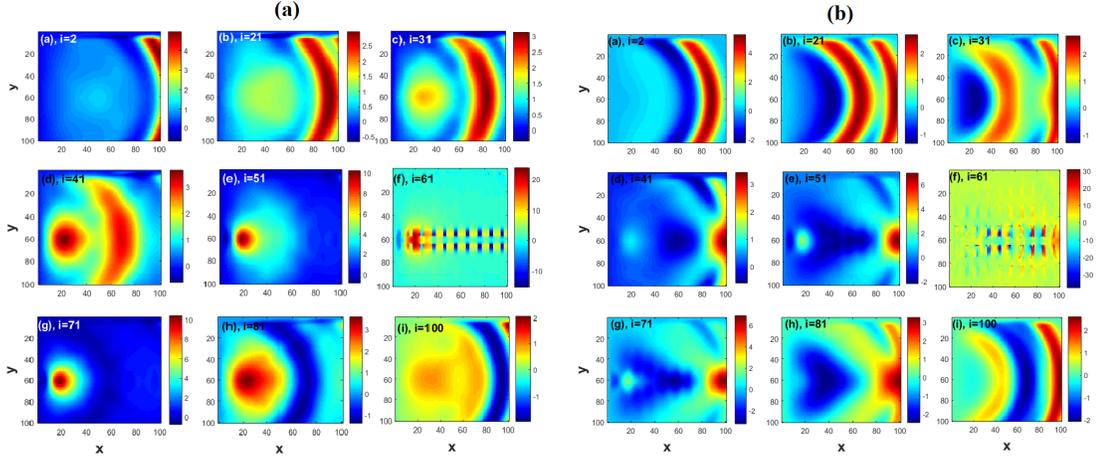


Figura 4.29: Tomografías de dos paneles en profundidad del campo E_x 3D en pequeños cortes que van desde $i = 2$ (capa cerca de la superficie de la nanoestructura) hasta $i = 100$ (capa más profunda del sistema 3D). Para el caso (a) las capas de tomografía corresponden a una frecuencia de $\omega_p = 1 \times 10^{13}$, para el caso (b) las capas corresponden a una frecuencia $\omega_p = 5 \times 10^{15}$, ambos casos con una velocidad $v = 0,75$

Observamos una mayor excitación de los nanotubos en los cortes con $i = 61$, cortes muy cercanos en donde está ubicada de la malla en el sistema, podemos observar el campo óptico bien definido y con mayor fuerza, para el caso 4.29 (a) observamos una excitación de campo en la ruta principal donde pasa la carga, para el caso 4.29 (b) con una frecuencia mayor observamos un campo óptico más distribuido en todo el campo de nanotubos y con mayor irradiación en el panel (f). Los cortes de las figuras 4.29 (a) y 4.29 (b), nos dan una noción general del comportamiento que ocurre en el campo óptico interno E_x del sistema, para tener una idea más precisa del campo óptico 3D de la malla que podemos observar en las Figuras 4.30 (a-d) el campo 3D en detalle, las figuras 4.30 (a) y 4.30 (b), pertenecen al campo óptico interno de la figura 4.28 (a) con $\omega_p = 1 \times 10^{13}$ mientras que las figuras 4.30 (c) y 4.30 (d) pertenecen al campo óptico interno de La figura 4.28 (b) con $\omega_p = 5 \times 10^{15}$.

Las figuras 4.30 (a) y 4.30 (c) muestran todo el sistema de campo óptico que deja la carga en el sistema, las flechas P y Q indican el área del campo de carga en los nanotubos excitados. Las figuras 4.30 (b) y 4.30 Fig5 (d) son secciones transversales a la altura donde se encuentra la nanoestructura de con morfología de red dentro del medio dieléctrico; Los campos ópticos producidos por las nanoestructuras se descubren visiblemente.

En la Figura 4.30 (d) podemos ver el campo de la malla de CNTs bien definido en el área XY donde está colocada la malla CNTs. Las flechas amarillas muestran la entrada y salida de la carga en el sistema.

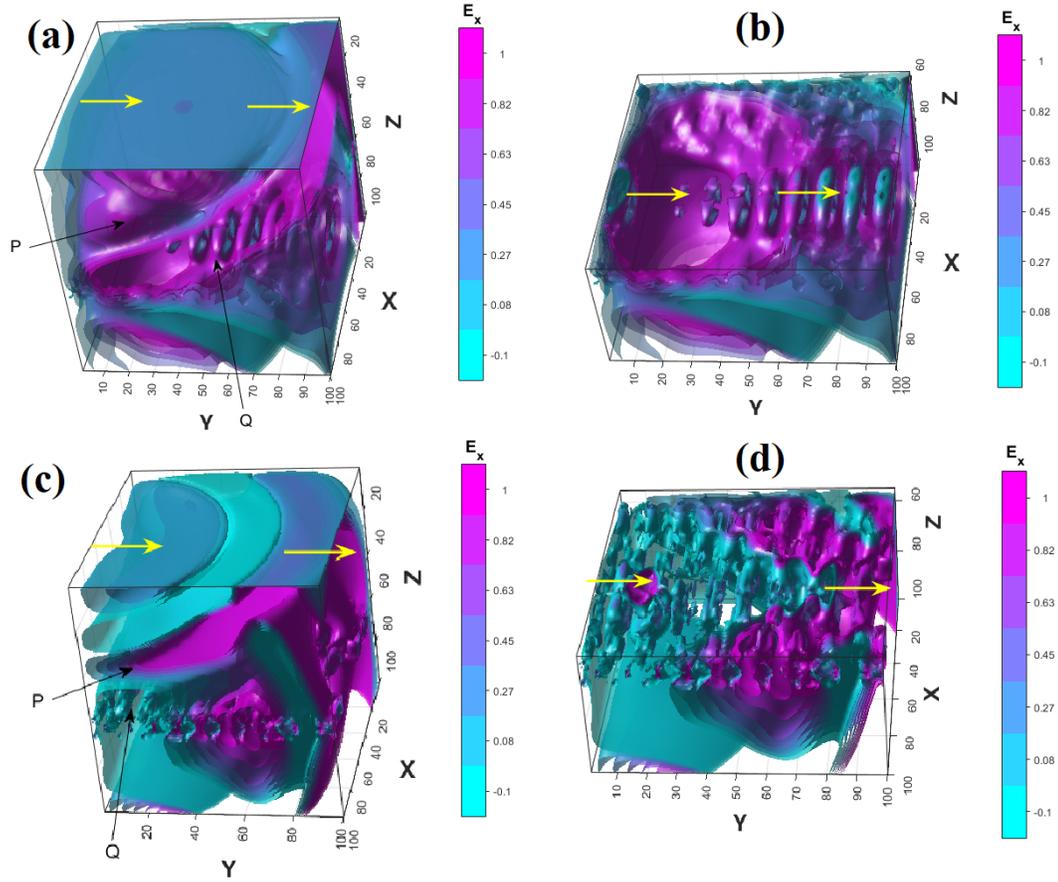


Figura 4.30: Campo óptico E_x que produce la malla de CNT dentro del sistema 3D. Para el caso (a), las flechas amarillas corresponden a la entrada y salida de la carga con velocidad $v = 0,75$ en la nanoestructura que resuena a una frecuencia de $\omega_p = 1 \times 10^{13} s^{-1}$. Las flechas P y Q indican el área del campo Cherenkov y del campo CNT, respectivamente. El caso (b) corresponde a un corte en el medio del sistema (a) donde observamos en detalle la forma del campo de los nanotubos. Lo mismo se aplica en los casos (c) y (d) pero con una frecuencia de $\omega_p = 5 \times 10^{15} s^{-1}$. Observamos que el campo visible interno que producen los nanotubos dentro del sistema 3D queda descubierto en las imágenes (b) y (d) y el campo resonante de los CNTs están orientadas en dirección XY formando una malla (paralelas a la dirección de la carga) extendida a lo largo del sistema. En la barra de colores E_x observamos la intensidad del campo eléctrico alcanzando como consecuencia entre la interacción de la carga y el nano-sistema. Las flechas amarillas indican la dirección de donde entra y sale la carga. Observamos claramente una fuerza de acoplamiento e intensidad del campo diferente entre las frecuencias $\omega_p = 1 \times 10^{13} s^{-1}$ y $\omega_p = 5 \times 10^{15} s^{-1}$

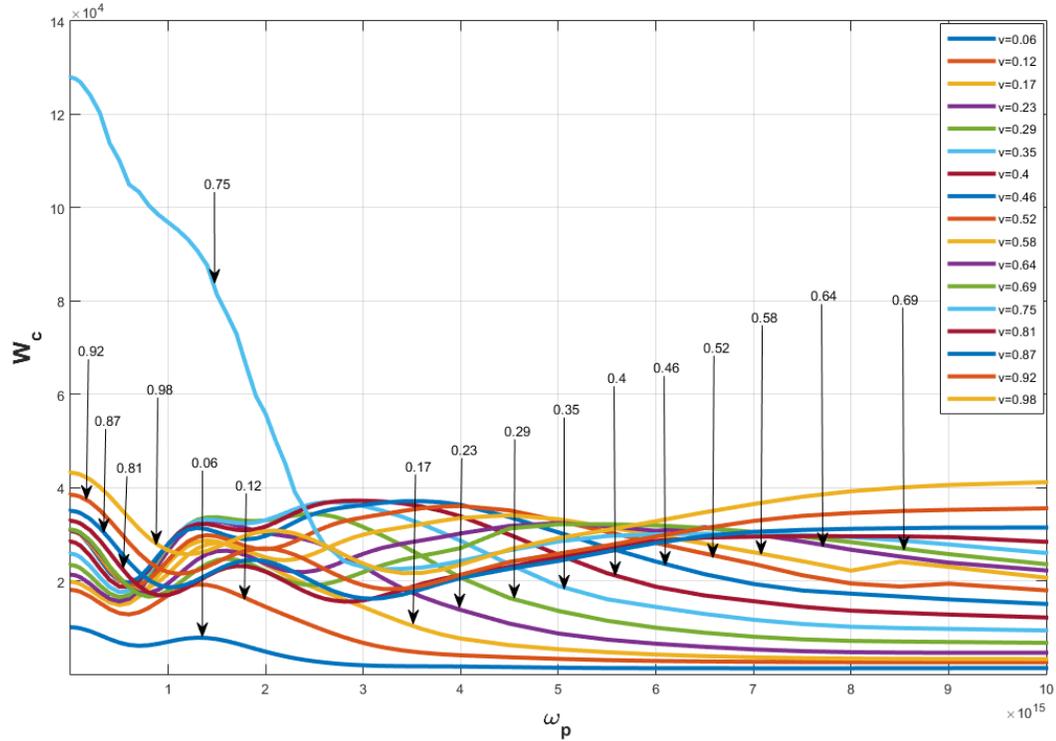


Figura 4.31: Investigamos un rango de frecuencia de $\omega_p = [1 \times 10^{13}, 1 \times 10^{16}]$ para un amplio rango de velocidades de carga ($v = 0.06, 0.12, 0.17, 0.23, 0.29, 0.35, 0.40, 0.46, 0.52, 0.58, 0.64, 0.69, 0.75, 0.81, 0.87, 0.92, 0.98$). Podemos observar que para la velocidad $v = 0.75$ hay un aumento significativo en la energía W_c que disminuye mientras aumenta la frecuencia ω_p en los nanotubos.

4.12. Rango de frecuencias ω_p para velocidades finitas

Podemos observar que en las figuras anteriores hay un cambio importante cuando el parámetro ω_p se modifica y por lo tanto en la energía que irradia el sistema cuando se altera este valor. Es por ello que estudiamos un rango de frecuencia para diferentes velocidades de la carga a través del sistema. Estudiamos una familia de velocidades que van desde $v = 0,6$ hasta $v = 0,98$. La figura 4.31 muestra una relación con la dependencia de frecuencia ω_p vs W_c para velocidades $v = 0.06, 0.12, 0.17, 0.23, 0.29, 0.35, 0.40, 0.46, 0.52, 0.58, 0.64, 0.69, 0.75, 0.81, 0.87, 0.92, 0.98$, incluimos velocidades de carga bajas y altas. La energía radiada total W_c para cada velocidad respectiva se obtiene sumando los valores que irradian los plasmones de superficie en el sistema en cada valor de frecuencia ω_p . Estudiamos un rango de frecuencia de $\omega = 1 \times 10^{13}$ hasta $\omega_p = 1 \times 10^{16}$, despreciamos las frecuencias $\omega_p < 1 \times 10^{13}$, ya que los valores de energía permanecieron constantes en este punto. De la figura 4.31, podemos observar un pico sobresaliente de energía que comienza en una frecuencia 1×10^{13} y decrece mientras aumenta la frecuencia cuando la carga va a una velocidad de $v = 0,75$ (Línea azul de la figura 4.31).

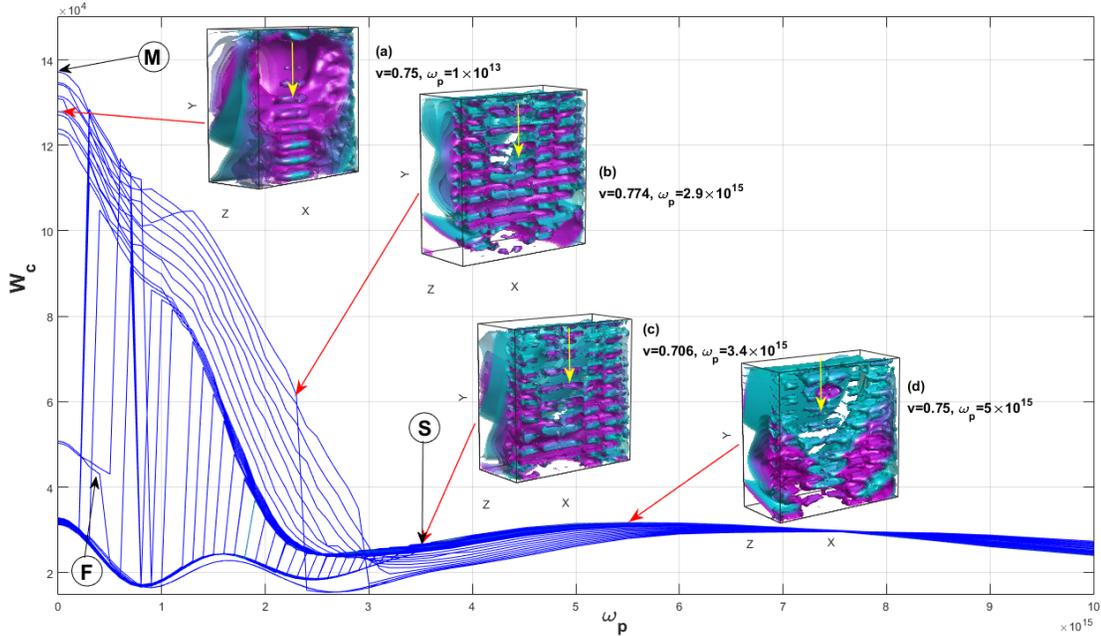


Figura 4.32: Encontramos una familia de velocidades $v = [0,706, 0,791]$ que generan diferentes picos de energía W_c a ciertas frecuencias, que comienzan en el punto $v = 0,706$ y $\omega_p = 3,4 \times 10^{15}$ y termina en el punto $v = 0,791$ con $\omega_p = 4 \times 10^{14}$, siendo este, el último pico de energía encontrado en nuestra investigación. También encontramos un pico máximo de energía indicado con la flecha M con una velocidad $v = 0,779$ y una frecuencia $\omega_p = 1 \times 10^{13}$ siendo el pico más alto de toda la familia investigada.

Para comprender más detalladamente el pico de energía en la Figura 4.31 con velocidades de carga $v = 0.75$, estudiamos las velocidades de carga cercanas a esta velocidad y encontramos una familia de picos de energía muy similar a la generada por la carga con $v = 0.75$. En la figura 4.32 observamos una familia de velocidades con un rango de $v = [0,706 - 0,791]$, en todo este rango estudiado existen picos de energía registrados en diferentes puntos ω_p . El primer pico de energía registrado en nuestros resultados numéricos corresponde al valor de velocidad $v = 0.706$ con una frecuencia $\omega_p = 3,4 \times 10^{15}$, mientras que el último pico de energía se registró en $v = 0,791$ con $\omega_p = 4 \times 10^{14}$. En la Figura 4.32, observamos 4 casos del campo óptico interno con medio corte donde podemos ver el campo óptico de los nanotubos. En las Figuras 4.32 (a-d), observamos cuatro casos del campo óptico interno a medio corte, donde podemos ver el campo óptico de los nanotubos, observamos que para el caso (a) $v = 0,75$ existe una frecuencia $\omega_p = 1 \times 10^{13}$, para el caso (b), $v = 0.774$ con $\omega_p = 2,9 \times 10^{15}$, para el caso (c) $v = 0,706$ con $\omega_p = 3,4 \times 10^{15}$, y para el caso (d) hay un $v = 0,75$ y $\omega_p = 5 \times 10^{15}$. En los casos de la figura 4.32 (b), (c) y (d) con altas frecuencias, podemos ver un campo bien definido de nanotubos en forma de malla CNT. Las flechas amarillas indican la dirección en la que se mueve la carga. Las fluctuaciones de energía comienzan a ocurrir en un punto con velocidad $v = 0,706$ $\omega_p = 3,4 \times 10^{15}$, indicado por la flecha

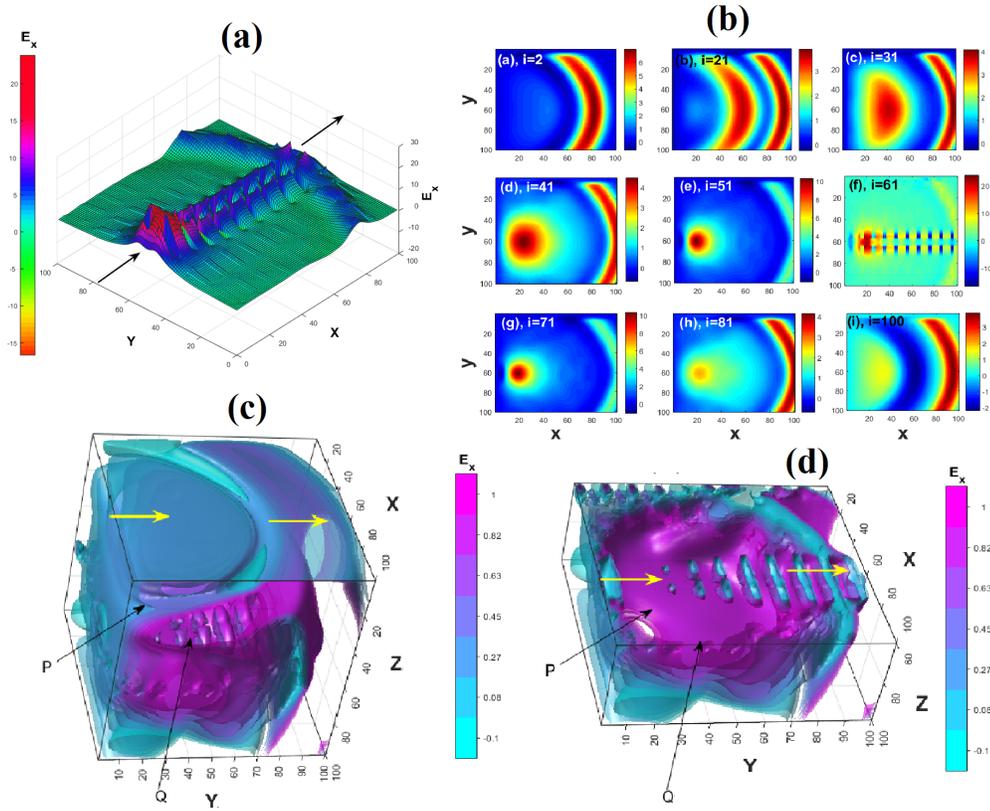


Figura 4.33: La imagen (a) muestra el campo que produce la carga en la superficie de la nanoestructura de red de nanotubos, (b) tenemos un panel de tomografías en profundidad del campo E_x 3D en cortes pequeños, la capa con valor $i = 2$, es la capa más cercana a la superficie, mientras que la capa con el valor $i = 100$ es la capa más profunda del sistema. (c) Observamos el campo óptico 3D interno que produce la carga en el sistema, la imagen (d) muestra un corte a la mitad del mismo sistema. Las flechas amarillas corresponden a la entrada y salida de la carga. Las flechas P y Q indican el área del campo Cherenkov y el campo CNT. Las imágenes (a-d) corresponden al pico de energía máxima (flecha M) de la figura 4.32 con velocidad $v = 0,779$ y frecuencia $\omega_p = 1 \times 10^{13}$

S (Inicio) y aumentan con cada incremento de velocidad hasta alcanzar un punto máximo indicado con la flecha M (máximo) con valores de velocidad y frecuencia $v = 0,779$ y $\omega_p = 1 \times 10^{13}$ respectivamente. Posteriormente, las fluctuaciones de energía disminuyen hasta que tienen una fluctuación final de energía en el punto de velocidad $v = 0,791$ y frecuencia $\omega_p = 4 \times 10^{14}$, indicado por la flecha F (Final). El pico de radiación de energía máxima indicado por la flecha M de la Figura 4.32 representa un acoplamiento máximo de energía entre la carga y el sistema de nanotubos resonantes. En la Figura 4.33 (a-d), apreciamos el comportamiento en la superficie y dentro del sistema dieléctrico donde la malla de CNT correspondiente a este acoplamiento se muestra en detalle (Ver flecha M de la figura 4.32). En la Figura 4.33 (a) observamos la radiación cherenkov en forma de cono (frente de onda excitado) y un campo E_x excitado en la ruta principal por donde pasa la

carga. La figura 4.32 (b) corresponde a cortes del sistema, observamos una resonancia más fuerte en el corte $i = 61$, corte muy cerca de donde está la malla CNT. En la Figura 4.32 (c) podemos ver la distribución del campo E_x que produce la carga dentro del sistema, las flechas amarillas indican la entrada y salida de la carga, la figura 4.32 (d) muestra el mismo sistema con un corte transversal por la mitad muy cerca de donde se encuentra la malla CNT. Las flechas Q y P indican el área del campo excitada por la radiación de Cherenkov.

Podemos concluir que en la última sección de esta investigación; investigamos la radiación de Cherenkov producida por una partícula en un nano sistema en forma de malla compuesta de nanotubos de carbono donde aplicamos una frecuencia de plasma. Encontramos una familia de velocidades de carga que producen fluctuaciones de energía en diferentes puntos de ω_p , de los cuales se destaca un pico de energía máxima con un mejor acoplamiento entre la carga y la malla de nanotubos.

RED NEURONAL QUE DETECTA UNA TRANSICIÓN DE FASE PARA UN SISTEMA DE PERCOLACIÓN MEDIANTE BACKPROPAGATION

5.1. Introducción

Las investigaciones de estructuras conglomeradas espaciales en materiales de filtración en diversas aplicaciones físicas aplicadas en particular para la radiación de nano-emisores desordenados tridimensionales en materiales sólidos con percolación son áreas de investigación activa [99–108]. En una pequeña concentración de defectos en tales sistemas, el número de grupos es insignificante. Sin embargo, si la concentración de grupos excede un cierto valor umbral $p > p_c$, entonces el grupo de expansión se forma en el sistema, extendiéndose por todo el volumen. Este grupo cambia cualitativamente las propiedades de un medio y produce una conductividad generalizada en el sistema que originalmente no posee esa propiedad [109–113]. Se dice que dos poros o huecos en el material (vinculados a los nodos) se comunican si la distancia entre ellos es menor que su radio. Sin embargo, tal modelo con tamaños de poros fijos es demasiado simple para ser adecuado para casos reales de nano-cristales. Aquí consideramos una situación más avanzada cuando el radio de cada poro, donde r una función de valor aleatorio. Esto lleva a que la media $R = \langle r \rangle$ y una variación de $\sigma_r = (\langle r^2 \rangle - \langle r \rangle^2)^{1/2}$ emergen como nuevas escalas espaciales que dan como resultado que la probabilidad crítica de percolación p_c se convierta en la función complicada de R y σ_r . En dicha filtración extendida, el grupo de expansión incipiente (y la transición de fase de filtración) puede surgir en una p_c diferente del valor estándar 0.59 (para **2D**) y 0.318 (para **3D**).

Comprender las transiciones de fase de filtración estructural es una parte importante de la física moderna. Recientemente, las investigaciones en el campo de las

transiciones de fase en sistemas correlacionados de muchos cuerpos se convierten en un área activa de desarrollos con el uso de máquinas inteligentes en el contexto de investigaciones científicas [114–122]. La existencia de una transición de fase de filtración en p_c de probabilidad crítica entre las fases de filtración y no filtración nos permite clasificar los dos tipos diferentes configuraciones (2D y 3D) directamente y sin el uso de implementaciones del método Monte Carlo. En este trabajo, construimos una red neuronal de retroalimentación para realizar un aprendizaje supervisado con estas configuraciones de clústeres de filtración espacial. Descubrimos que dicha red neuronal después de entrenar en una amplia gama de datos con probabilidad por encima y por debajo de la p_c , es capaz de clasificar correctamente la transición de filtración en los conjuntos de prueba para los sistemas 2D y 3D. Vale la pena señalar que debido a las formas espaciales fractales de los grupos de filtración (con dimensión fractal $L = 31$), la predicción de las transiciones de fase de filtración para 2D y específicamente para casos 3D se convierten en tareas involucradas.

5.2. Ecuaciones básicas

Dos poros (vacíos) producen un grupo si tienen al menos un vecino en común. En dicho enfoque, la filtración ocurre cuando el tamaño del clúster más grande (que abarca) alcanza el tamaño del sistema. Vale la pena señalar que para el sistema con poros aleatorios, no se conoce con exactitud ninguna expresión analítica para el umbral de percolación solo datos numéricos [123]. Para simular un clúster local, introducimos la ecuación:

$$\Psi_{ij} = d_{ij} - (r_i + r_j) \quad (5.1)$$

Donde d_{ij} es la distancia entre i y j poros, r_i es el radio del poro i y los índices i, j son las dimensiones x, y . Por consideraciones geométricas, está claro que $\Psi_{ij} < 0$ si i y j son objetos acoplados, de lo contrario, si $\Psi_{ij} \geq 0$ los poros están desacoplados. En nuestro estudio, normalizamos todas las longitudes del sistema a una constante de red de referencia a_0 . Para los poros más cercanos en geometría más simple tenemos $d_{ij} = 1$ y $r_i = r$ que producen $\Psi_{ij} = 1 - 2r$. Esto significa que todos los poros están desacoplados si $r \leq r_c = 0,5$. En el caso general, caracterizaremos el radio r como una función de valor aleatorio distribuida uniformemente en el intervalo $[b, a]$, con $b = (1 + \alpha_0)R$, $\alpha = (1 - \alpha_0)R$, donde $0 < \alpha_0 < 1$ y $R > 0$ son parámetros libres. Esto permite simular r_i como $r_i = (1 - \alpha_0 + 2\alpha_0 \xi_i)R$, donde ξ son números aleatorios. El cuadrado de la desviación estándar (dispersión) para r es $\sigma_r^2 = \langle r^2 \rangle - \langle r \rangle^2 = R^2\alpha_0/3$, entonces $\alpha_0 = \sqrt{3}\sigma_r/R$, por lo tanto Ψ_{ij} se lee:

$$\Psi_{ij} = d_{ij} - 2[1 - \alpha + \alpha(\xi_i + \xi_j)]R. \quad (5.2)$$

Las condiciones acopladas en un sistema tan extendido es $\Psi_{ij} < 0$ que permite reconocer los parámetros α_0, R como nuevas escalas espaciales. Solo para el caso de poros fijos más cercano a ($\alpha = 0, R = 0,5$) de la ecuación (5.2) se puede simplificar a $\Psi_{ij} = 1 - 2r$. Pero en general, para los cristales reales se ocupa $\sigma_r \neq 0$, por lo que

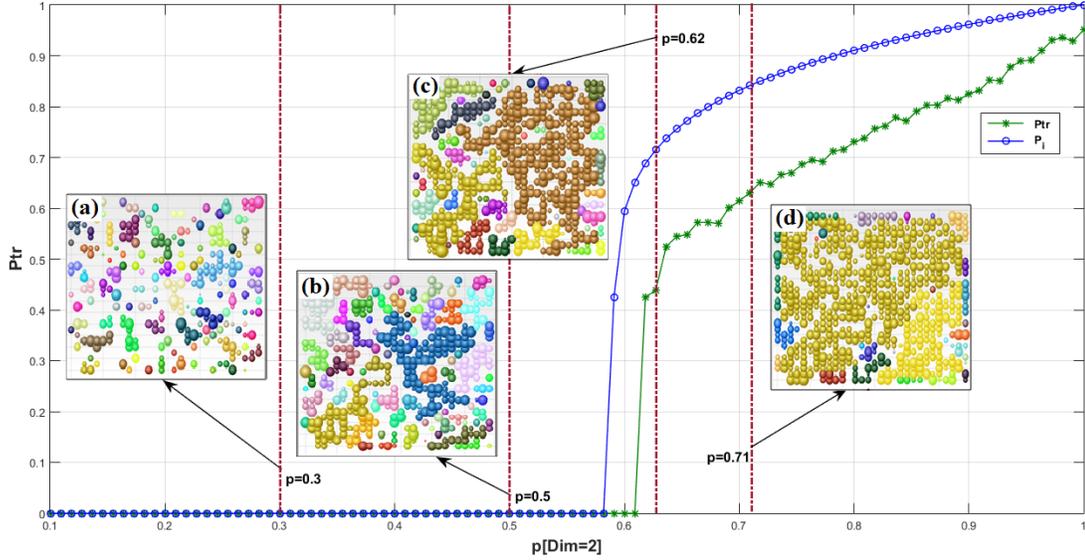


Figura 5.1: El parámetro de orden $P(p)$ (línea verde) en función de la probabilidad del número de poros p para el sistema de filtración $2D L \times L$, $L = 30$ designada para la filtración extendida con poros aleatorios. (La línea azul muestra $P(p)$ para la filtración estándar con poros fijos). Los recuadros muestran la distribución espacial de los poros con radio aleatorio r en varios valores p : (a) y (b) por debajo de la probabilidad crítica $p < p_c$ (en este caso, el grupo infinito no existe), (c) $p \simeq p_c$ (solo aparece un grupo incipiente de filtración), y (d) $p > p_c$ (se encuentran los grupos masivos). Debido a la aleatoriedad del radio de poros r , el valor crítico p_c difiere del conocido valor estándar $p_c = 0,59$. Los clusters predominantes de percolación acoplados se muestran en diferentes colores predominantes sobre los recuadros (a-d) de la figura.

dicho sistema adquiere nuevas propiedades con respecto al modelo de percolación estándar. Para evitar las intersecciones múltiples de poros, restringimos nuestra atención para el caso $R = \langle r \rangle < \sqrt{2}/2 \simeq 0,707$ (el valor medio de R es menos de la mitad de la diagonal de celda más cercana). Sin embargo, el estudio de un sistema de este tipo, incluso para el caso de $2D$, no se puede hacer analíticamente y requiere cálculos numéricos intensivos que se ocupan de identificar la existencia del clúster infinito. El parámetro de orden de percolación $P(p)$ se define como la relación entre el número de poros que pertenecen al grupo de expansión y el número general de poros. En general, el comportamiento de dicho sistema se convierte en una función complicada, ya que otros parámetros como σ_r , R (Desviación de error en una distribución) están involucrados en esta investigación.

5.3. Red neuronal supervisada

Primero consideramos el caso $2D$ donde el radio r de cada poro i ($i = x, y$) es una cantidad valorada aleatoriamente con el valor medio $R = \langle r \rangle$ y $\sigma_r^2 = \langle r^2 \rangle - \langle r \rangle^2$. La figura 5.1 muestra el parámetro de orden de filtración P (línea verde) en función de la probabilidad de población $0 < p < 1$ para $2D$ sistema

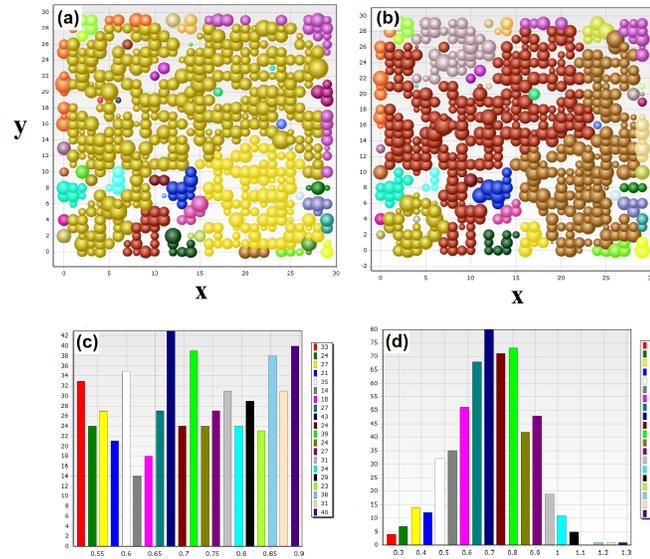


Figura 5.2: La estructura espacial de los poros aleatorios en $p \simeq p_c$. En nuestro estudio utilizamos dos distribuciones diferentes de poros de tamaño aleatorio con parámetros $\langle r \rangle = 0,706, \sigma_r = 0,12$: (a) distribuciones uniformes, y (b) distribución normal (Gauss). Los paneles (c) y (d) muestran los histogramas para poros aleatorios obtenidos en nuestras respectivas simulaciones.

de filtración $L \times L$, donde $L = 30$ para $R = 0,707 < r_c$ y $\sigma_r = 0,16$, la curva azul muestra el parámetro de orden de filtración para el modelo de filtración 2D estándar, donde la filtración surge en la probabilidad $p = p_c \simeq 0,59$ [109]. En la figura 5.1 los recuadros muestran la distribución espacial 2D de poros con radio aleatorio r a varios valores de probabilidad p . En la figura 5.1 los recuadros (a), (b) muestran la distribución por debajo de la probabilidad crítica $p < p_c$ (en este caso, el grupo infinito aún no existe), (c) en $p \simeq p_c$ (solo se forma un grupo incipiente de filtración), y (d) en $p > p_c$ se encuentra un grupo de filtración masiva. Debido a la aleatoriedad del radio de poros r , el valor p_c en la figura 5.1 difiere ligeramente del valor estándar $p_c = 0,59$ [112], [123]. Los grupos de percolación desacoplados se muestran en la figura 5.1 por diferentes colores. Para completar nuestro estudio, consideramos dos tipos de distribuciones para los poros aleatorios: la distribución uniforme y la distribución normal (Gauss). La figura 5.2 (a), (b) muestra las estructuras espaciales de los poros con casos uniformes y normales, respectivamente, cerca de p_c , mientras que el paneles 5.2(c) y (d) muestran los histogramas para radio de poros uniforme y normal calculado en nuestras simulaciones. Cada imagen en la figura 5.2 (a-d) representa una matriz numérica bidimensional que contiene los números $r = 0$ (sin poros) o r correspondiente al radio de un poro dado. Cuando la concentración de poros cercana al valor crítico $p \simeq p_c$, los pequeños grupos de percolación separados (que se muestran en la 5.1 (a, b)) se fusionan y forman un grupo infinito o clúster infinito (de expansión) (Ver figura 5.1(c)), que establece una transición crítica. Es de gran interés detectar tal

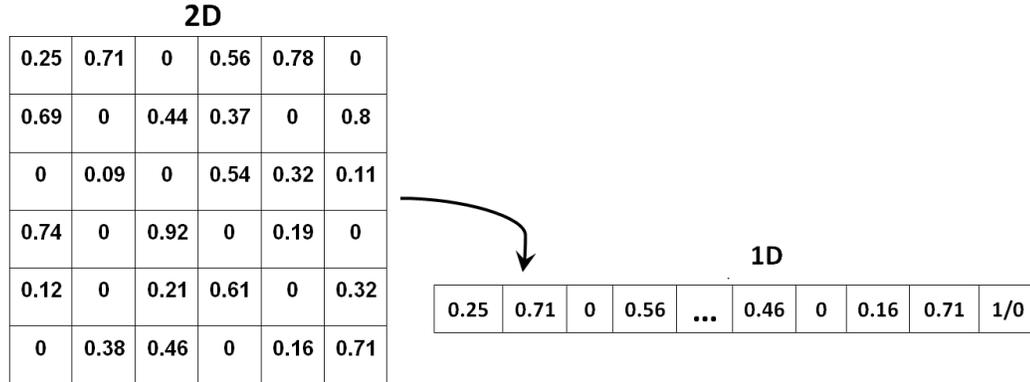


Figura 5.3: La matriz $2D$ $L \times L$ (con estructura espacial de los clústeres) se reformata en el vector $1D$ con una longitud L^2 que representa una fila del conjunto de datos (clase de entidad) (Ver figura. 5.4). Dichos vectores se utilizan como entrada para el entrenamiento de la red neuronal supervisada. Los últimos elementos de cada vector $1D$ (etiqueta de la última columna) tienen valores 0 o 1 dependiendo de la existencia de la transición de fase de filtración en el sistema para la probabilidad dada (Ver figura 5.1).

transición para materiales aleatorios en general que es una tarea importante para las nanoestructuras modernas.

Para explorar dicho problema mediante el enfoque de aprendizaje automático, creamos una red neuronal (SNN) supervisada. En nuestro SNN, las matrices aleatorias que se filtran (reformadas en vectores unidimensionales ($1D$), ver figura 5.3) se utilizan como entrada (feed) para el entrenamiento de SNN, mientras que la última columna etiquetada con valores 1 o 0 (última entrada) indica la presencia de transición de filtración, ver figura 5.4. Vale la pena señalar que este enfoque se puede generalizar fácilmente para sistemas de dimensiones superiores $D > 2$ de manera directa, ver figura 5.6 para el caso $3D$ [124–132]. En la figura 5.4 podemos observar un rango amplio de probabilidades P , en donde para cada vector indica un sistema de percolación $2D$, para cada vector tenemos asignado un valor al final de 0 o 1 que indica si existe o no percolación, podemos observar que conforme la probabilidad P es pequeña en los primeros vectores que forman de la base de datos \mathbf{R} (Base de datos de entrada para entrenar la red neuronal), no existe percolación, mientras que en los últimos vectores con P cercanos a 1, si existe percolación, este valor de verdadero o falso se ve reflejado en cada una de las últimas columnas sombreadas del vector. La figura 5.4 muestra el conjunto de datos \mathbf{R} (en nuestra base de datos para entrenar la red) que creamos para el intervalo de probabilidad $p \in [p_{\text{mín}}, p_{\text{máx}}]$ (ver figura. 5.1) para entrenar a la red supervisada. Para probar la precisión de la predicción de transición, hemos utilizado el siguiente enfoque: (i) Se extrajo un vector \mathbf{P} (clase de entidad) (usando el muestreo aleatorio) de la base de datos inicial, ver figura 5.4. (ii) El vector asociado (que mencionamos como datos "nuevos") se genera como $\mathbf{Q} = f\mathbf{P}$ al modular todos los componentes \mathbf{P} por un valor aleatorio $f = 1 - a + 2a\gamma$, donde $1 > a \geq 0$, y γ es un ruido

$$R = \left\{ \begin{array}{cccccccccc} 0.25 & 0 & 0 & 0 & \dots & 0.46 & 0 & 0 & 0 & 0 & 0 & p=0.01 \\ 0 & 0.24 & 0 & 0.31 & \dots & 0.12 & 0 & 0 & 0.64 & 0 & 0 & p=0.02 \\ 0.61 & 0 & 0 & 0.65 & \dots & 0 & 0.47 & 0.51 & 0.85 & 0 & 0 & p=0.03 \\ \vdots & & & & & & & & & & & \\ 0.33 & 0.15 & 0.71 & 0 & \dots & 0.63 & 0.41 & 0 & 0.16 & 1 & 1 & p=0.61 \\ 0.21 & 0.41 & 0 & 0.25 & \dots & 0.49 & 0.55 & 0.14 & 0.17 & 1 & 1 & p=0.98 \\ 0.44 & 0.18 & 0.64 & 0.37 & \dots & 0.51 & 0.38 & 0 & 0.14 & 1 & 1 & p=0.99 \\ 0.44 & 0.18 & 0.64 & 0.37 & \dots & 0.51 & 0.38 & 0.22 & 0.14 & 1 & 1 & p=1 \end{array} \right\}$$

Figura 5.4: El conjunto de datos inicial consta de un vector $1D$ con filas de datos convertidos a partir de matrices $2D$ (estructura espacial del material de filtración), la etiqueta (última columna 0 o 1) calculada por cada p indicado en la derecha lado indica un valor booleano para indicar si existe o no percolación en el sistema de filtración. Dicho conjunto de datos se utiliza para alimentar y capacitar a la red supervisada para detectar la transición de filtración en el sistema extendido $2D$.

aleatorio uniforme que se recalcula en cada paso de computación.

Si $a = 0$ ($f = 1$) los vectores \mathbf{P} y \mathbf{Q} son idénticos; pero si $a > 0$ los vectores \mathbf{P} y \mathbf{Q} se vuelven diferentes (pero manteniendo cierta similitud) debido a una modulación aleatoria con factor $f \in [1 - a, 1 + a]$. El vector \mathbf{Q} se utiliza para verificar la precisión de la tarea de clasificación que es una predicción del valor de la etiqueta \bar{P}_k por lo tanto \bar{Q}_k . La precisión media de la predicción de transición en N experimentos se define por la ecuación:

$$A = N^{-1} \sum_{k=1}^N \delta(\bar{P}_k, \bar{Q}_k), \quad (5.3)$$

Donde \bar{P}_k y \bar{Q}_k son valores de las etiquetas correspondientes, $\delta(i, j) = 1$ en $i = j$ y 0 de lo contrario (el símbolo de Kronecker), es $N = 100$.

5.4. Expectativa vs Predicción

Una vez entrenada la red neuronal realizamos algunas pruebas probando los datos de los resultados reales con el resultado de la predicción de la red neuronal, si la red neuronal está bien entrenada los datos predichos deberían de ser iguales a los datos de la prueba maestra, cuando no existe ruido (conocida como amplitud de noise) que alteren los resultados. Cuando hablamos de ruido nos referimos a una pequeña desviación en los datos generados para las pruebas aleatorias (En nuestro caso el ruido lo aplicamos a los radios de los poros). En la figura 5.5(a) observamos esta característica donde el nivel de ruido es 0.1 y todas las predicciones de la red neuronal coinciden con la expectativa (Donde n es el número de

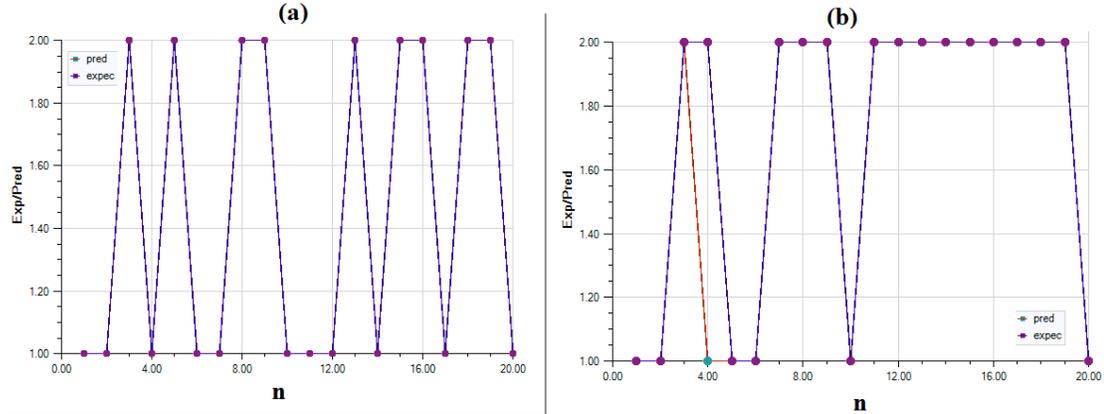


Figura 5.5: Pruebas de expectativa vs predicción, en la imagen (a), tenemos un ruido de 0.1 aplicado a los radios de los poros, observando que la red neuronal soporta este nivel de ruido, en el caso de la imagen (b) tenemos un nivel de ruido de 0.5, observamos que la red neuronal comienza a fallar en exactitud en sus predicciones. Donde n es el número de prueba o iteración.

iteración). (**expec**). Mientras que en la figura 5.5(b) tenemos un nivel de ruido de 0.5 en los radios de los poros. Observamos una pequeña discrepancia (línea roja que sobresale) en la tercera prueba de veinte. Los parámetros de entrenamiento para las pruebas realizadas son; un tamaño de rejilla $L \times L$ donde $L=30$ en una rejilla cuadra, con un rango de probabilidad de 0.1 ha 1, una distribución uniforme con $\sigma_r = 0,05$, que consideramos como parámetros más relevantes. Podemos observar de la aprueba anterior que la red neuronal soporta un cierto rango de ruido aplicado y es capaz de reconocer y predecir si existe o no percolación en el sistema siempre y cuando el ruido no sea grande, por tal motivo es importante conocer la brecha o punto crítico de nuestra red neuronal en la que ese eficiente con respecto a los niveles de ruido aplicado. En la figura 5.6 observamos esta característica añadiendo un intervalo de ruido (Amp-Noise) de 0.1 hasta 20, donde observamos que después de un nivel de ruido igual a 0.3, la gráfica o eficiencia de la predicción decrece (línea punteada de círculos rojos). Como podemos darnos cuenta de la gráfica 5.6 existe un decaimiento de eficiencia en función de la amplitud de noise o ruido aplicado a la red neuronal en su eficiencia para predecir los resultados cuando la matriz de estudio tiene una dimensión de $L \times L$ donde $L=30$, esto nos lleva a estudiar el mismo caso pero para diferentes valores de L . En este estudio consideramos dimensiones de las matrices para entrenar nuestra red neuronal con valores de $L = 30$ hasta $L = 40$, ya que ese difícil encontrar los parámetros precisos para que la red neuronal arroje buenos resultados en el entrenamiento, las tablas 5.1 y 5.2 muestran los parámetros de entrenamiento utilizados en la red para el caso Uniforme y el caso Gauss respectivamente, que posteriormente se utilizaron para las pruebas y las predicciones. Podemos observar de las tablas 5.1 y 5.2 un factor importante; el parámetro \mathbf{N} , que indica el número de puntos distribuidos sobre la probabilidad $P = 0,1$ hasta $P = 1$ (debido que es necesario estudiar todo el espectro de probabilidades para una predicción más integra), como lo puntos

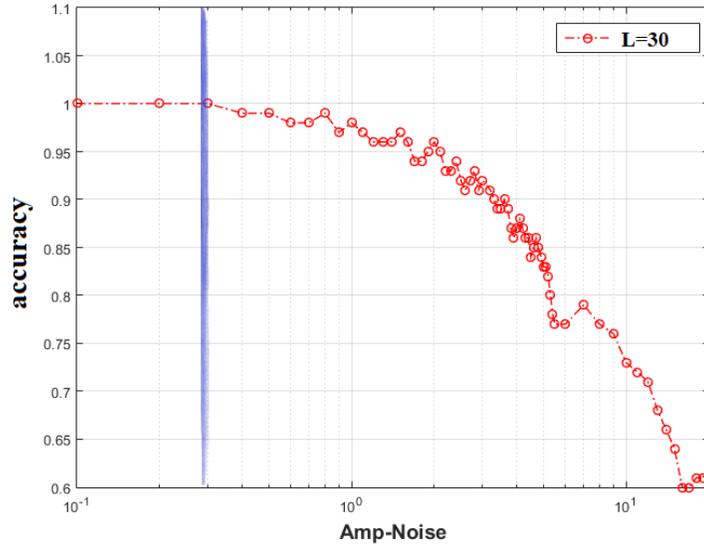


Figura 5.6: Intervalo de amplitud de ruido para probar la eficiencia de la red neuronal al reconocer una transición de fase. Observamos que esta red soporta ruidos menores o iguales a 0.3, después de la eficiencia decrece si el ruido aumenta. La línea azul indica el límite de la eficiencia de la red. La etiqueta *accuracy* indica el nivel de precisión de la red y $L=30$ es el tamaño de la matriz 2D de percolación en la cual se aplicaron las pruebas.

distribuidos en la figura 5.1, en donde vemos que cada punto es una probabilidad estudiada. Otro parámetro importante es la desviación de los radios de los poros σ_r que varía un poco sobre todo para el caso de la distribución Uniforme. Cabe señalar que esta familia de parámetros encontrados en nuestra investigación arroja un 100 por ciento de precisión en las pruebas de nuestra predicción, siempre y cuando el ruido no sea grande (Otros tamaños del sistema no fueron estudiados en este trabajo, ya que encontrar los parámetros precisos se vuelve una tarea compleja, debido a que no se sigue un cierto orden o patrón para encontrar los parámetros correctos para el entrenamiento de la red neuronal). El entrenamiento consistió en menos de 1000 épocas de entrenamiento y usó el algoritmo de propagación hacia atrás para los cálculos de gradiente para encontrar una red local óptima. En la figura 5.6, observamos el decaimiento que sufre la eficiencia de la red neuronal cuando se incrementa el ruido sobre las pruebas estudiadas en un rango de ruido en una matriz de tamaño $L=30$, este valor revela un apreciable dato del punto en que comienzan a fallar las predicciones, sin embargo en matrices de diferentes tamaños este valor puede variar un poco, ese por ello que realizamos más pruebas de expectativa vs predicción para matrices con tamaños $L=30$ hasta $L=40$. El comportamiento de estas pruebas las podemos observar en las gráficas de las figuras 5.7 y 5.8. Como podemos observar de las figuras 5.7 y 5.8, el comportamiento de ambas distribuciones es muy similar al estudiar la eficiencia de la red neuronal, aunque en el caso de la figura 5.7 para el caso uniforme las eficiencias por cada dimensión del sistema de percolación se mantiene más estables. En la

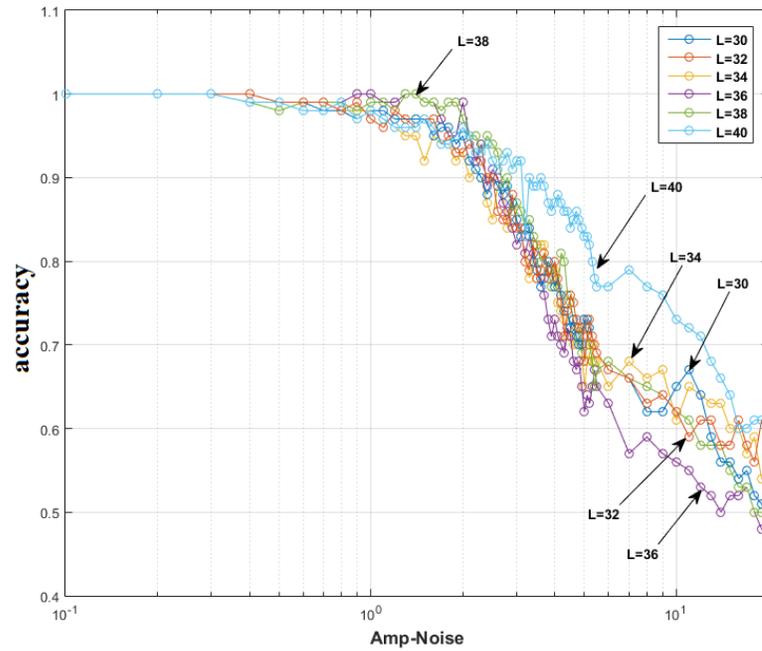


Figura 5.7: Intervalo de amplitud de ruido para probar la eficiencia de la red neuronal al reconocer una transición de fase con tamaños de rejilla $L=30, 32, 34, 36, 38$ y 40 . La etiqueta *accuracy* indica el nivel de precisión de la red para una distribución Uniforme.

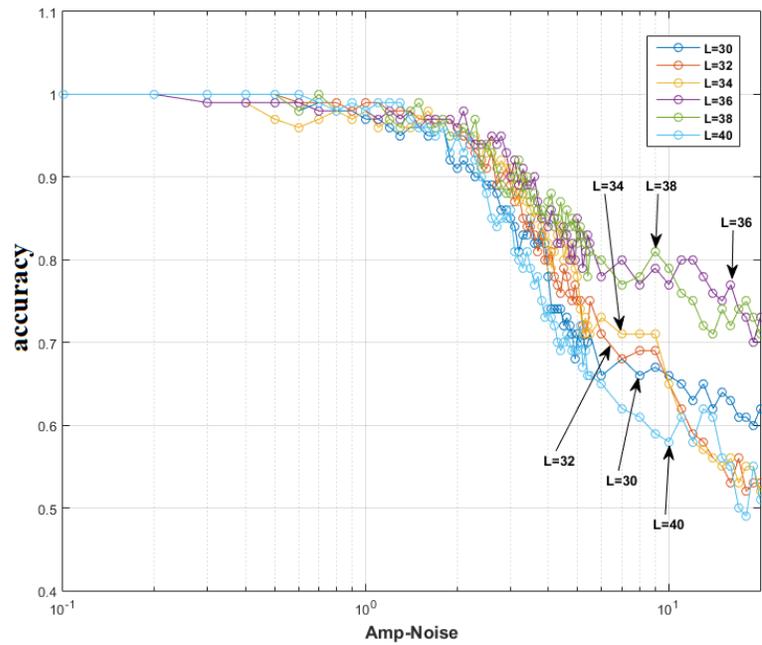


Figura 5.8: Intervalo de amplitud de ruido para probar la eficiencia de la red neuronal al reconocer una transición de fase con tamaños de rejilla $L=30, 32, 34, 36, 38$ y 40 . La etiqueta *accuracy* indica el nivel de precisión de la red para una distribución Gauss.

L	P_{min}	P_{max}	N	σ_r
30	0.1	1	50	0.16
32	0.1	1	65	0.06
34	0.1	1	65	0.05
36	0.1	1	65	0.16
38	0.1	1	65	0.06
40	0.1	1	81	0.05

Tabla 5.1: Parámetros de entrenamiento de la red neuronal para matrices 2D con distribución Uniforme.

L	P_{min}	P_{max}	N	σ_r
30	0.1	1	50	0.05
32	0.1	1	60	0.05
34	0.1	1	80	0.05
36	0.1	1	90	0.05
38	0.1	1	100	0.05
40	0.1	1	105	0.05

Tabla 5.2: Parámetros de entrenamiento de la red neuronal para matrices 2D con distribución Gauss.

figura 5.9, podemos apreciar más a detalle esta transición de eficiencia de la red sobre los puntos de interés. En términos de dimensiones para sistemas 2D, nuestro aprovechamiento para calcular la percolación con una red neuronal supervisada arroja muy buenos resultados, esto tiene gran significancia en las ramas de los materiales ya que realizar esta predicción con una red neuronal se ocupan reconocer matrices de pixeles generalmente extendiéndose a valores de megas bytes y se ocupa mucho espacio de almacenamiento ya que al guardar imágenes de pixeles (y no valores enteros como en nuestro caso de estudio), los aumentos en espacio de almacenamiento incrementan, de igual manera pasa con el tiempo de procesamiento, para entrenar la red neuronal, ya que los tiempos para generar una base de datos buena que sea aprovechada para un futuro *training*, se basan en técnicas como la convolución entre matrices de pixeles, donde se ocupa una pequeña matriz laplaciana para realizar un producto cruz a lo largo de toda la matriz de pixeles, es de imaginar que esto requiere grandes requerimientos computacionales tanto en RAM como en espacio de disco duro, nuestra red neuronal convierte matrices de enteros a vectores y posteriormente los guarda en archivos para crear nuestra base de datos, así que no requiere mucho espacio de almacenamiento y tiempo de procesamiento a comparación de las técnicas convencionales como la convolución entre matrices. Por otra parte, se vuelve altamente difícil calcular percolaciones en 3D, ya en el caso de la convolución solo puede aplicarse a matrices bidimensionales y no es posible aplicar esta misma técnica para matrices tridimensionales. Sin embargo en nuestro caso estudiado es posible transformar una matriz 3D a una tabla 2D y posteriormente a un vector 1D, por lo que se vuelve más sencillo abordar la percolación en estructuras de 3 dimensiones. De igual manera como lo hicimos en las figuras 5.2(a, b), donde observamos un clúster infinito predominante con color para cada clúster, donde el clúster infinito predomina y atraviesa toda la rejilla 2D, realizamos lo mismo para el caso 3D, etiquetamos los clústeres para cada grupo de poros a conglomerado y le asignamos un color, podemos observar esta relación equivalente a la figura 5.2(a, b) pero para el caso 3D con distribución uniforme con desviación $\sigma_r = 0,01$ con poca variación en sus radios en la figura 5.10. El tamaño de la rejilla para este caso 3D corresponde a un tamaño de $L \times L \times L$; $L = 11$, una distribución de probabilidad $P_{min} = 0,1$ hasta $P_{max} = 1$

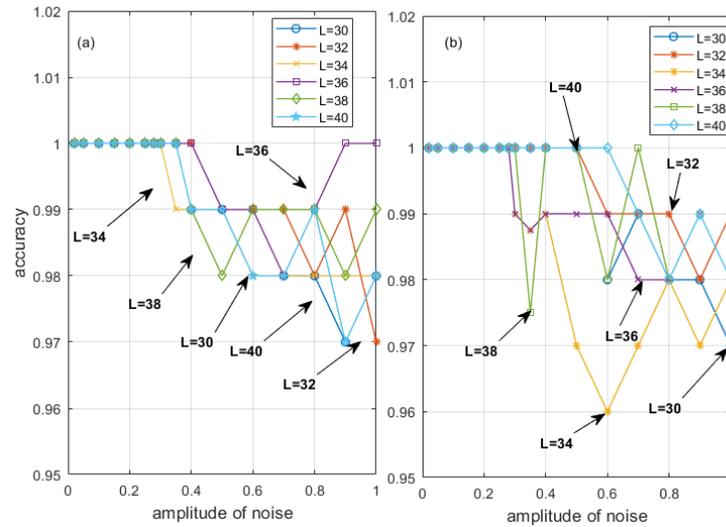


Figura 5.9: Precisión de la predicción de la red neuronal de la transición de fase de filtración (sistema $2D L \times L$ con diferentes $L = 30 \div 40$) en función de la amplitud de ruido a . En los casos con radio de poros distribuido uniformemente (a) y normal (b). Observamos que hasta una modulación de ruido moderada a $a < 0,3$ la precisión de la predicción es de aproximadamente 100%. Pero para ruidos mayores $a > 0,3$ la precisión disminuye.

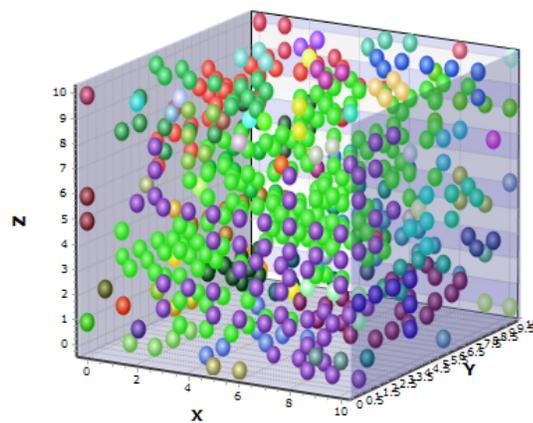


Figura 5.10: Distribución espacial 3D con poros de radio aleatorio r en el sistema de filtración 3D ($L \times L \times L$; $L = 11$), la probabilidad crítica se encuentra cerca de $p \simeq p_c = 0,32$. Para este caso 3D se visualiza el clúster infinito filtración (color verde). Los grupos no acoplados se muestran con diferentes colores.

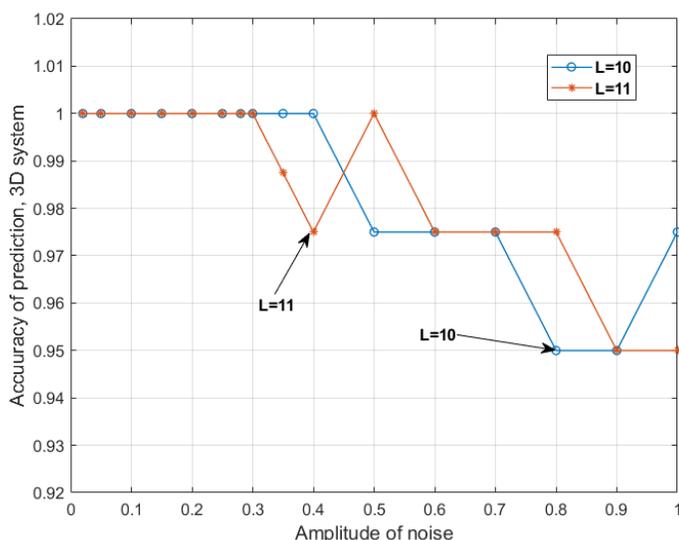


Figura 5.11: Observamos lo mismo que en la figura 5.9 (a) (con distribución uniforme) pero para el caso del sistema 3D con $L \times L \times L$ para $L = 10, 11$. Observamos nuevamente que a una amplitud de ruido $a < 0,3$ la precisión de predicción es de aproximadamente 100% como en el sistema 2D, ver figura 5.9 (a). Pero para una amplitud mayor $a > 0,3$ la precisión de la predicción disminuye.

con 40 puntos N distribuidos en ese rango con distribución de radio uniforme en sus poros y desviación $\sigma_r = 0,01$. Esto lleva al hecho de que predecir la transición de fase para el caso tridimensional es una tarea más difícil en relación con el caso bidimensional.

Para el caso 3D aplicamos las mismas pruebas de ruido como en el caso 2D (figuras 5.7, 5.8 y 5.9, el último caso con puntos cerca de donde la precisión de la predicción cambia) para encontrar el punto de eficiencia de nuestra red entrenada para el caso tridimensional. En la figura 5.11 se muestra la precisión de la transición de predicción para el sistema 3D $L \times L \times L$; $L = 11$ para la distribución aleatoria uniforme del radio de poro. Observamos de la figura 5.11 que hasta el valor de amplitud de ruido $a < 0,3$ (donde a es la amplitud de ruido o ruido aplicado en los radios de los poros del sistema para las pruebas de predicción de la red neuronal) la precisión de la predicción es de aproximadamente el 100% como en el caso del sistema 2D. Pero para un $a > 0,3$, la precisión de la predicción disminuye.

5.5. Conclusiones parciales

Podemos concluir que nuestra red neuronal probó ser eficiente en los problemas para transición de fase 2D y 3D para dimensiones parciales aplicando métodos de conversión de matrices 3D a 2D y de 2D a 1D. La eficiencia de nuestra red neuronal supervisada soporta niveles ruidos $a > 0,3$.

CONCLUSIONES.

De acuerdo al trabajo de tesis de nuestra investigación podemos concluir que:

1. Estudiamos un sistema 3D de nanotubos desordenados y encontramos resultados importantes en las probabilidades críticas de conducción para materiales con nanoestructura a base de CNT.
2. Simulamos varias estructuras desordenadas de diferentes tamaños variando las longitudes de los nanotubos ($\mu_H = 15$ a $\mu_H = 25 \text{ nm}$) y encontramos que para los nanotubos mayores o iguales a $\mu_H = 22$ la probabilidad crítica disminuye. Ver figura 2.24.
3. Simulamos varias estructuras desordenadas de diferentes tamaños variando los grosores de los nanotubos, aplicando una distribución normal (Gauss-Ver ecuación 2.13) para un intervalo de desviaciones, y encontramos que con grosores en los nanotubos mayores o iguales a $\mu_R = 0,5 \text{ nm}$ la probabilidad crítica disminuye. Ver figura 2.26.
4. Se estudió también el efecto que produce la radiación de Cherenkov sobre nanoestructuras CNT con morfologías periódicas y morfologías con estructura aleatoria aplicado en sus radios y longitudes resonando a una frecuencia de plasma ω_p , mientras pasa una carga que emite la radiación de Cherenkov viajando una velocidad v .
5. Se encontró un acoplamiento óptico en donde las nanoestructuras periódicas (en un arreglo 3D, ver figura 4.5) con radio y longitud uniforme que se encuentran resonando a una frecuencia de $\omega_p = 2,1 \times 10^{15} \text{ s}^{-1}$ produce más energía y liberación de plasmones-polaritones sobre la superficie de las nanoestructuras.
6. Se estudió también la interacción entre carga y el sistema de CNT cuando las nanoestructuras son de morfología aleatoria, con radios y longitudes diferentes en las medidas de los nanotubos en una arreglo 3D, ver figura 4.11 y se encontró que para este caso de energía un pico máximo de energía y

mejor acoplamiento entre la carga y la estructura resonante a una frecuencia de plasma es $\omega_p = 4,1 \times 10^{15} s^{-1}$. Ver figura 4.14.

7. En nuestra investigación consideramos también nanoestructuras de TiO_2 (Dióxido de titanio) que oscilan a bajas frecuencias y realizamos una comparación con los nanotubos de carbono, el resultado de esta comparación arrojó que los nanotubos de carbono CNT producen mayor energía respecto a la liberación de plasmones-polaritones de superficie. (Punto máximo acoplamiento de frecuencia para estructuras de TiO_2 fue $\omega_p = 1 \times 10^{10} s^{-1}$ y para los CNTs $\omega_p = 1,2 \times 10^{16} s^{-1}$, ver figura 4.25).
8. Investigamos la radiación de Cherenkov producida por una partícula en un nano sistema en forma de malla compuesta de nanotubos de carbono de estructura periódica, donde aplicamos una frecuencia de plasma. Encontramos una familia de velocidades de carga que producen fluctuaciones de energía en diferentes puntos de ω_p , de los cuales se destaca un pico de energía máxima con un mejor acoplamiento entre la carga y la malla de nanotubos con frecuencia $\omega_p = 1 \times 10^{13} s^{-1}$ y velocidad $v = 0,75$. Ver figura 4.32.
9. Como elemento adicional a esta investigación también se construyó una red neuronal artificial supervisada utilizando un entrenamiento de retro propagación en el cual estudiamos el fenómeno de transición de fase en materiales percolables, nuestra red neuronal probó ser eficiente para detectar la transición de fase para sistemas 2D y 3D (Para rejillas pequeñas) incluso al agregarse ruido en los datos de prueba (amplitud de noise).
10. Se realizaron varias pruebas para medir la eficiencia de la red neuronal supervisada que detecta una transición de fase (Punto porosidad en donde los materiales comienzan a tener percolación) agregando una amplitud de noise(ruido) y se registró que para los casos 2D y 3D, soportaron un ruido hasta el punto $a \leq 0,3$. Ver figuras 5.9 y 5.11.

CÓDIGO PARA LA PERCOLACIÓN CON CNTs

A.1. Código One-Cylinder

Se utilizó el código de la clase OneCylinder para crear un objeto cilindro que representa nuestro CNT, ubicado en un nodo de la rejilla tridimensional 3D con propiedades aleatorias de altura h , radio del cilindro r y orientación con ángulos θ y ϕ . Primera parte de la imagen (a) del esquema de calculación de la figura 2.17.

```

1  using System.Text;
2  namespace WinCylsFDTD
3  {
4      public class OneCylinder
5      {
6          #region user
7          double _xc, _yc, _zc;
8          double _h, _r;
9          double _phi, _theta;
10         char _chType = '0'; // initial char
11         double _x1, _x2, _y1, _y2, _z1, _z2, _volumCyl = 0;
12         double _cosTheta, _sinTheta;
13         int _cyIID = -1, _clusterID = -1/*ID del clúster que contiene el cilindro */;
14         int _L = 0, _M = 0, _N = 0;
15         int _cylsInCluster = 0; // Número total de cilindros que contiene el clúster
16         //public List<int> _listIDsForThisCyl = new List<int>();
17         int[] _iFather = new int[] { -1, -1 }; // initial cell i,j,k=0 in the toppest layer // iniciar las
18         capas superiores con valor de 0.
19     #endregion
20     #region set/get
21     public bool isEqualFathers(OneCylinder c1, OneCylinder c2)
22     {
23         bool b0 = c1.get_iFather()[0] == c2.get_iFather()[0];
24         bool b1 = c1.get_iFather()[1] == c2.get_iFather()[1];
25         return b0 && b1;
26     }
27     public char get_chType() { return _chType; }
28     public void set_chType(char c) { _chType = c; }
29     public void set_iFather(int i, int j) { _iFather[0] = i; _iFather[1] = j; }
30     public void set_iFather(int[] iFather) { _iFather[0] = iFather[0]; _iFather[1] = iFather[1]; }
31     public void get_iFather(out int i, out int j) { i = _iFather[0]; j = _iFather[1]; }
32     public int[] get_iFather() { return (int[])_iFather.Clone(); }
33     public double get_volumCyl() { return _volumCyl; }
34     public int get_cyIID() { return _cyIID; }
35     public void set_cyIID(int c) { _cyIID = c; }
36     public int get_clusterID() { return _clusterID; }
37     public void set_clusterID(int c) { _clusterID = c; }
38     public int get_cylsInCluster() { return _cylsInCluster; }
39     public void set_cylsInCluster(int c) { _cylsInCluster = c; }
40     public double get_x1() { return _x1; } public double get_y1() { return _y1; } public double get_z1()
41     { return _z1; }
42     public double get_x2() { return _x2; } public double get_y2() { return _y2; } public double get_z2()
43     { return _z2; }
44     public double get_h() { return _h; } public double get_r() { return _r; }
45     #endregion
46     #region constructors
47     public OneCylinder() { }
48     public OneCylinder(OneCylinder cc1)
49     {
50         _xc = cc1._xc; _yc = cc1._yc; _zc = cc1._zc;
51         _h = cc1._h; _r = cc1._r; _phi = cc1._phi; _theta = cc1._theta;
52         _cyIID = cc1._cyIID;
53         _N = cc1._N; _chType = cc1._chType;
54         _x1 = cc1._x1; _y1 = cc1._y1; _z1 = cc1._z1;
55         _x2 = cc1._x2; _y2 = cc1._y2; _z2 = cc1._z2;
56     }
57     }
58 }

```

```

50  _cylsInCluster = ccl._cylsInCluster;
51  _iFather = (int[]) ccl._iFather.Clone();
52  _clusterID = ccl._clusterID;
53  _volumCyl = ccl._volumCyl;
54  } // OneCylinder
55  #endregion
56  public void InitOneCylinderEnds(double xc, double yc, double zc, double h, double r, double phi,
    double cosTheta, double sinTheta, int L, int M, int N)
57  {
58  _xc = xc; _yc = yc; _zc = zc; _h = h; _r = r;
59  _phi = phi; _theta = theta; _L = L; _M = M; _N = N;
60  _cosTheta = cosTheta; _sinTheta = sinTheta;
61  _volumCyl = Math.PI * r * r * Math.Abs(h);
62  } // InitOneCylinderEnds
63  // ----- CalcIntervalCyl -----
64  public void CalcEndsIntervalOfCylinder()
65  {
66  _x1 = _xc; _y1 = _yc; _z1 = _zc;
67  _x2 = _xc + _h * Math.Cos(_phi) * _sinTheta;
68  _y2 = _yc + _h * Math.Sin(_phi) * _sinTheta;
69  _z2 = _zc + _h * _cosTheta;
70  _x2 = _x2 < _L - 1 ? _x2 : _L - 1;
71  _y2 = _y2 < _M - 1 ? _y2 : _M - 1;
72  _z2 = _z2 < _N - 1 ? _z2 : _N - 1;
73  _x2 = _x2 < 0 ? 0 : _x2;
74  _y2 = _y2 < 0 ? 0 : _y2;
75  _z2 = _z2 < 0 ? 0 : _z2;
76  } // CalcIntervalCyl
77  // ----- ToStringPlot -----
78  public string ToStringPlot()
79  {
80  string sOut = "r=" + _r + ", h=" + _h + ", phi=" + _phi + ", theta=" + _theta + ", cylID=" + _cylID +
81  " , chType=" + _chType;
82  return sOut;
83  } // ToStringPlot
84  // ----- ToString -----
85  public override string ToString()
86  {
87  string sOut = "\nx1=" + _x1 + ", y1=" + _y1 + ", z1=" + _z1 + ",\n x2=" + _x2 + ", y2=" + _y2 + ", z2
88  = " + _z2
89  + "\n r=" + _r + ", h=" + _h + ",\n phi=" + _phi + ", theta=" + _theta + ", cylID=" + _cylID
90  + " , chType=" + _chType + " , clusterID=" + _clusterID + " , cylsInCluster=" + _cylsInCluster + " , volum
91  = " + _volumCyl;
92  return sOut;
93  } // ToString
94  } // class OneCylinder

```

A.2. Código WorkWithCylinders

El código de la clase: *WorkWithCylinders* contiene los métodos necesarios para calcular si existe la percolación en una rejilla, se encarga de crear nuevos objetos a partir de la clase *OneCylinder* para llenar un array (con valores de 0 en la parte superior del arreglo) de objetos de este tipo, posteriormente se signa un objeto cilindro en cada uno de los nodos de la rejilla 3D para todos aquellos valores que tienen un valor 1 (Existe un cilindro) según el factor P (Probabilidad) asignado, y se verifica si existe algún contacto con algún otro objeto cilindro en la rejilla 3D, de modo que si existe los cilindros con contacto se etiquetan con un ID para identificar el grupo (o clúster) de interacción de estos cilindros. Se obtiene el volumen de cada clúster, así como el número existente de clústeres en el arreglo, los cilindros que no pertenecen a un grupo (cilindros que están sueltos en el sistema) y el clúster que cruza la rejilla 3D, en caso de haberlo (Dependiendo la probabilidad asignada en la prueba numérica).

```

1  #define isCycleKLast
2  using System;
3  using System.Collections.Generic;
4  using System.Linq;
5  using System.Text;
6  using progressBar;
7
8  namespace WinCylsFDTD
9  {
10 // ----- class WorkWithCylinders -----
11 class WorkWithCylinders
12 {
13 #region user
14 int _L = -1, _M = -1, _N = -1;

```

```

15 int _iterCylContacts = 1;
16 int _seed = 0, _maxCylsInCluster = 0, _maxCylsInInfinCluster = 0, _currentThread = -1, _idum;
17 int _cylsReachedBott = 0, _totalCylInArr = 0;
18 double _hMax = 0, _rMax = 0;
19 double _hMin = 0, _rMin = 0;
20 double _pCritLoc = 2;
21 double _currProbability = -1;
22 double _POrdenCyl, _volumCylTotalFrac = -1, _volumePercolCylsFrac = -1;
23 double _volAllCylsInPercCluster = 0, _aspRatioAllClusterMean = 0, _lengthAllCylsInPercCluster=0;
24 bool _isPercolation = false, _isRandomPos = false, _calcClusters=false;
25 bool _StopNow = false, _useNNullNodes = false;
26 bool _isRRandom, _isHRandom, _isPhiRandom, _isThetaRandom, _populAllInTop = true;
27 bool _useAllAngles = false;
28 //bool _isGaussianCyls = false;
29 string _typeOfRandCyls = "";
30 string _sInfo = "";
31 OneCylinder[, ] _arrOfCylinders = null;
32 Random _rnd7;
33 TRan3 _rn3 = new TRan3();
34 #endregion
35 #region set/get
36 public void set_calcClusters(bool c) { _calcClusters = c; }
37 public void set_populAllInTop(bool c) { _populAllInTop = c; }
38 public bool get_populAllInTop() { return _populAllInTop; }
39 public bool get_isRRandom() { return _isRRandom; }
40 public bool get_isHRandom() { return _isHRandom; }
41 public bool get_isPhiRandom() { return _isPhiRandom; }
42 public bool get_isThetaRandom() { return _isThetaRandom; }
43 public bool get_useAllAngles() { return _useAllAngles; }
44 public void set_isRandomPos(bool v) { _isRandomPos = v; }
45 public void set_iterCylContacts(int v) { _iterCylContacts = v; }
46 public OneCylinder[, ] get_arrOfCylinders() { return _arrOfCylinders; }
47 public OneCylinder get_arrOfCylinders(int i, int j, int k) { return _arrOfCylinders[i, j, k]; }
48 public double get_volumCylTotalFrac() { return _volumCylTotalFrac; }
49 public double get_volumePercolCylsFrac() { return _volumePercolCylsFrac; }
50 public double get_volAllCylsInPercCluster() { return _volAllCylsInPercCluster; }
51 public double get_aspRatioAllClusterMean() { return _aspRatioAllClusterMean; }
52 public double get_lengthAllCylsInPercCluster() { return _lengthAllCylsInPercCluster; }
53 public double get_pCritLoc() { return _pCritLoc; }
54 public int get_cylsReachedBott() { return _cylsReachedBott; }
55 public int get_maxCylsInCluster() { return _maxCylsInCluster; }
56 public string get_sInfo() { return _sInfo; }
57 public void set_sInfo(string c) { _sInfo = c; }
58 public double get_POrdenCyl() { return _POrdenCyl; }
59 private void set_POrdenCyl(double p) { _POrdenCyl = p; }
60 public bool get_isPercolation() { return _isPercolation; }
61 public bool get_useNNullNodes() { return _useNNullNodes; }
62 public void set_useNNullNodes(bool b) { _useNNullNodes = b; }
63 public string get_typeOfRandCyls() { return _typeOfRandCyls; }
64 public void set_typeOfRandCyls(string b) { _typeOfRandCyls = b; }
65 public void set_useAllAngles(bool b) { _useAllAngles = b; }
66 public void set_seed(int seed) { _seed = seed; }
67 public int get_seed() { return _seed; }
68 public void set_currentThread(int N) { _currentThread = N; }
69 public int get_L() { return _L; }
70 public int get_M() { return _M; }
71 public int get_N() { return _N; }
72 public double get_currProbability() { return _currProbability; }
73 public void set_currProbability(double b) { _currProbability = b; }
74 public void set_StopNow(bool b) { _StopNow = b; }
75 public double get_hMax() { return _hMax; }
76 public double get_rMax() { return _rMax; }
77 public double get_hMin() { return _hMin; }
78 public double get_rMin() { return _rMin; }
79 // - - - - - SetGridSize - - - - -
80 public void SetGridSize(int L, int M, int N)
81 { _L = L; _M = M; _N = N; } //SetGridSize
82 #endregion
83 #region constructors
84 public WorkWithCylinders() { }
85 public WorkWithCylinders(WorkWithCylinders cc1)
86 {
87     _hMax = cc1._hMax; _rMax = cc1._rMax;
88     _hMin = cc1._hMin; _rMin = cc1._rMin;
89     _sInfo = cc1._sInfo;
90     _StopNow = cc1._StopNow;
91     _currProbability = cc1._currProbability; _L = cc1._L; _M = cc1._M; _N = cc1._N;
92     _POrdenCyl = cc1._POrdenCyl; _seed = cc1._seed;
93     _isPercolation = cc1._isPercolation;
94     _isRRandom = cc1._isRRandom; _isHRandom = cc1._isHRandom; _isPhiRandom = cc1._isPhiRandom;
95     _isThetaRandom = cc1._isThetaRandom;
96     _useAllAngles = cc1._useAllAngles;
97     _typeOfRandCyls = cc1._typeOfRandCyls;
98     _populAllInTop = cc1._populAllInTop;
99     _useNNullNodes = cc1._useNNullNodes;
100    _totalCylInArr = cc1._totalCylInArr;
101    _maxCylsInCluster = cc1._maxCylsInCluster;
102    _volumCylTotalFrac = cc1._volumCylTotalFrac;
103    _isRandomPos = cc1._isRandomPos;
104    _rnd7 = cc1._rnd7;
105    _iterCylContacts = cc1._iterCylContacts;
106    _currentThread = cc1._currentThread;
107    _calcClusters = cc1._calcClusters;
108    _rn3 = cc1._rn3;
109    _idum = cc1._idum;
110    _maxCylsInInfinCluster = cc1._maxCylsInInfinCluster;
111    _volAllCylsInPercCluster = cc1._volAllCylsInPercCluster;
112    _aspRatioAllClusterMean = cc1._aspRatioAllClusterMean;
113    _lengthAllCylsInPercCluster = cc1._lengthAllCylsInPercCluster;
114 } //WorkWithCylinders
115 #endregion

```

```

116 // - - - - - CalcPercolCyls - - - - -
117 public void InitHRMax(double hMax, double rMax, double hMin, double rMin)
118 { _hMax = hMax; _rMax = rMax; _hMin = hMin; _rMin = rMin; }
119 // - - - - - CalcPercolCyls - - - - -
120 public void InitRandoms(bool isRRandom, bool isHRandom, bool isPhiRandom, bool isThetaRandom, int
    seed)
121 {
122     _isRRandom = isRRandom; _isHRandom = isHRandom;
123     _isPhiRandom = isPhiRandom; _isThetaRandom = isThetaRandom;
124     if (seed == 0)
125     {
126         _rnd7 = new Random(Guid.NewGuid().GetHashCode());
127         _idum = -1;
128         _rn3.ran3(ref _idum);
129     }
130     else
131     {
132         _rnd7 = new Random(_seed);
133         _idum = -seed;
134         _rn3.ran3(ref _idum);
135     }
136 }
137 //InitRandoms
138 // - - - - - InitRandomGen - - - - -
139 public void InitRandomGen(int seed)
140 {
141     if (seed == 0)
142         _rnd7 = new Random(Guid.NewGuid().GetHashCode());
143     else
144         _rnd7 = new Random(_seed);
145 } //InitRandomGen
146 // - - - - - CalcPercolCyls PopulateCylinders - - - - -
147 public void PopulateCylinders(ref OneCylinder[, ] arrOfCylinders, int nSeed, bool isRRandom, bool
    isHRandom, bool isPhiRandom, bool isThetaRandom
    , double probab)
148 {
149     Random rnd;
150     GaussianRandom guassianRandom = new GaussianRandom(nSeed);
151     PoissonRandom poissonRandom = new PoissonRandom(nSeed);
152     if (nSeed > 0)
153         rnd = new Random(nSeed);
154     else
155         rnd = new Random(Guid.NewGuid().GetHashCode());
156     Console.ForegroundColor = ConsoleColor.Magenta;
157     Console.WriteLine("\n - - - - - Percolation of arrOfCylinders. - - - - -");
158     Console.ResetColor();
159     OneCylinder c = null;
160     Console.WriteLine("probab={0:0.00}, L={1},M={2},N={3}, nSeed={4}.", probab, _L, _M, _N, nSeed);
161     int nCylInArr = 0;
162     double xc, yc, zc;
163     double h, r, phi, theta;
164     double hMax = get_hMax(), rMax = get_rMax(), hMin = get_hMin(), rMin = get_rMin();
165     Console.WriteLine("hMax={0:0.00}, rMax={1:0.00}.", hMax, rMax);
166     Console.WriteLine("h is Random={0}, r is Random={1}", isHRandom, isRRandom);
167     Console.WriteLine("phi is Random={0}, theta is Random={1}", isPhiRandom, isThetaRandom);
168     Console.WriteLine("Generation random arrOfCylinders...");
169     double muH = 0, sigmaH = 0, muR = 0, sigmaR = 0, lambdaH = 0, lambdaR = 0;
170     bool isGaussianCyls = false, isPoissonCyls = false, isContinueInitPoints = false;
171     //!!! if (_StopNow == true) return;
172     switch (get_typeofRandCyls())
173     {
174     case "uniform": isGaussianCyls = false; break;
175     case "Gauss":
176         isGaussianCyls = true;
177         muH = hMax; sigmaH = hMin;
178         muR = rMax; sigmaR = rMin;
179         break;
180     case "Poisson":
181         isPoissonCyls = true;
182         lambdaH = hMax; //sigmaH = _hMin;
183         lambdaR = rMax; //sigmaR = _rMin;
184         break;
185     case "continue points":
186         isContinueInitPoints = true; break;
187     default:
188         Console.ForegroundColor = ConsoleColor.Red;
189         Console.WriteLine("!! WorkWithCylinders --> Error. Unknown typeOfRandCyls");
190         Console.ResetColor();
191         break;
192     }
193 }
194 // - - - - - populate arrOfCylinders - - - - -
195 #if isCycleKLast
196 #else
197 for (int k = 0; k < arrOfCylinders.GetLength(2); k++) // prepare arrOfCylinders and add to array
198 #endif
199 for (int i = 0; i < arrOfCylinders.GetLength(0); i++)
200 for (int j = 0; j < arrOfCylinders.GetLength(1); j++)
201 #if isCycleKLast
202 for (int k = 0; k < arrOfCylinders.GetLength(2); k++) // prepare arrOfCylinders and add to array
203 #endif
204 {
205     //Preparar un cilindro para la rejilla
206     if ((get_populAllInTop() == true && k == 0) // Todos los cilindros en la capa superior son 0
207         || rnd.NextDouble() < probab
208         ) //población de los nodos en la capa superior x,y,z
209         { //>>>I population
210             c = new OneCylinder(); // construir cilindro
211             // - - xc,yc,zc - - continue points -----
212             if (isContinueInitPoints == true)
213             {
214                 xc = rnd.NextDouble() * get_L(); yc = rnd.NextDouble() * get_M(); // continuar con coordenadas [0..N

```

```

215 //zc = rnd.NextDouble() * N;
216 }
217 else
218 {
219 xc = i; yc = j; // fijar coordenadas en la malla [0,..,N-1]
220 }
221 zc = k;
222
223 // -- h -- (altura/longitud del cilindro)
224 if (isGaussianCyls == true) //gauss
225 h = isHRandom == true ? gaussianRandom.NextGaussian(muH, sigmaH) : muH; // generar h
226 else if (isPoissonCyls == true) //Poisson
227 h = isHRandom == true ? poissonRandom.NextPoisson(lambdaH) : lambdaH; // generar h
228 else //uniform
229 h = isHRandom == true ? NextUnifRandom(get_hMax(), get_hMin(), rnd) : hMax; // generate h
230 //h = Math.Abs(h); // para valores negativos
231 // -- r --
232 if (isGaussianCyls == true) //gauss
233 r = isRRandom == true ? gaussianRandom.NextGaussian(muR, sigmaR) : muR; // generate h
234 else if (isPoissonCyls == true) //Poisson
235 r = isRRandom == true ? poissonRandom.NextPoisson(lambdaR) : lambdaR; // generate h
236 else //uniform
237 r = isRRandom == true ? NextUnifRandom(get_rMax(), get_rMin(), rnd) : rMax; // generar r
238 // -- theta, phi --
239 double cosTheta = 0, sinTheta = 1;
240 if (get_useAllAngles() == true)
241 {
242 phi = 2 * Math.PI * (isPhiRandom == true ? rnd.NextDouble() : 1); // generar 0 < phi <= 2*PI
243 theta = Math.PI / 2 * (isThetaRandom == true ? rnd.NextDouble() : 1); // generar 0 < theta <= PI/2
244 cosTheta = 1 - 2 * rnd.NextDouble();
245 double gg = rnd.NextDouble();
246 sinTheta = 2 * Math.Sqrt((1 - gg) * gg);
247 }
248 else
249 {
250 phi = Math.PI / 2 * (isPhiRandom == true ? rnd.NextDouble() : 1); // generar 0 < phi <= PI/2
251 theta = Math.PI / 2 * (isThetaRandom == true ? rnd.NextDouble() : 1); // generar 0 < theta <= PI/2
252 }
253 c.InitOneCylinderEnds(xc, yc, zc, h, r, phi, theta, cosTheta, sinTheta, get_L(), get_M(), get_N());
254 // inicializar un cilindro
255 c.CalcEndsIntervalOfCylinder();
256 //nCylInArr++;
257
258 if (k == 0) //todos los cilindros i,j en la capa superior k=0
259 {
260 c.set_chType('1');
261 c.set_iFather(i, j);
262 c.set_clusterID(nCylInArr); //Establecer el mismo ID
263 a los clusters de la capa superior
264 }
265 c.set_cylID(nCylInArr); //Colocar un ID a cada cilindro
266 arrOfCylinders[i, j, k] = new OneCylinder(c); //Insertar el cilindro en el sistema (array)
267 nCylInArr++;
268 } //<<<i, població, if (k == 0 || rnd.NextDouble() < probab)
269 // k,j,i
270 Console.WriteLine("Cylinders in array={0}", nCylInArr);
271 _totalCylInArr = nCylInArr; // total de cilindros en el sistema (array)
272 //return arrOfCylinders;
273 } //PopulateCylinders
274 // ----- CalcPercolCyls -----
275 public void CalcPercolCyls()
276 {
277 OneCylinder[, ,] aCyls = new OneCylinder[get_L(), get_M(), get_N()];
278 PopulateCylinders(ref aCyls, get_seed(), get_isRRandom(), get_IsHRandom(), get_isPhiRandom(),
279 get_isThetaRandom());
280 , get_currProbability());
281 CalcPercolCyls(aCyls, _currProbability, out _POrdenCyl, out _isPercolation);
282 } //CalcPercolCyls
283 // ----- CalcPercolCyls -----
284 public void CalcPercolCyls(OneCylinder[, ,] arrOfCylinders, double probab, out double POrden, out
285 bool isPercolation)
286 {
287 Console.WriteLine("\nCheck if all the cylinders are initialized in top layer i,j, and k=0...");
288 bool isNotInit = false;
289 int nTop = 0;
290 for (int i = 0; i < arrOfCylinders.GetLength(0); i++) // checar si los
291 cilindros esta inicializados en os índices i,j, de la capa k=0
292 for (int j = 0; j < arrOfCylinders.GetLength(1); j++)
293 if (arrOfCylinders[i, j, 0] == null || arrOfCylinders[i, j, 0].get_chType() != '1')
294 {
295 isNotInit = true;
296 if (_populAllInTop == true) {
297 Console.ForegroundColor = ConsoleColor.Red;
298 Console.WriteLine("Error!!! Is not initialized cylinder at i={0}, j={1} and k=0", i, j);
299 Console.ResetColor();
300 }
301 }
302 else nTop++;
303
304 if (_populAllInTop == true) {
305 if (isNotInit == true)
306 Console.WriteLine(" Not all cylinders it top layer nTop={0} from {1} are initialized.", nTop, _L *
307 _M);
308 else
309 Console.WriteLine(" The cylinders it top layer nTop={0} from {1} are initialized. Done.", nTop, _L *
310 _M);
311 }
312 else
313 Console.WriteLine(" The cylinders it top layer k=0 are initialized nTop={0} cyls from {1}. Done.",
314 nTop, _L * _M);
315
316 int n00 = 0, n10 = 0, nNull0 = 0;

```

```

307 #if isCycleKLast
308 #else
309 for (int k2 = 0; k2 < arrOfCylinders.GetLength(2); k2++)
310 #endif
311 for (int i2 = 0; i2 < arrOfCylinders.GetLength(0); i2++) for (int j2 = 0; j2 <
    arrOfCylinders.GetLength(1); j2++)
312 #if isCycleKLast
313 for (int k2 = 0; k2 < arrOfCylinders.GetLength(2); k2++)
314 #endif
315 {
316 OneCylinder cyl = arrOfCylinders[i2, j2, k2];
317 if (cyl == null)
318 {
319 nNull0++;
320 continue;
321 }
322 if (cyl.get_chType() == '0') n00++;
323 if (cyl.get_chType() == '1') n10++;
324 }
325 int totalCylInArr = _totalCylInArr;
326 Console.WriteLine("\nStatistics (before calc.contacts):\n\tnNull={0},n0={1}, n1={2}, sum={3} (at init
    ={4})\n"
    , nNull0, n00, n10, (n10 + n00), totalCylInArr);
327
328 Console.WriteLine(" Done.");
329 // -----
330 Console.WriteLine(" Iteration contacts={0}...", _iterCylContacts);
331 int iterCylContacts = _iterCylContacts;
332 int contcsOld = 0, foundContacts=0, iterCount;
333 bool isConvergContact = false;
334 for (iterCount = 1; iterCount <= iterCylContacts; iterCount++)
335 { // Checar el cambio de número de contactos Recalcular: CheckAllMutualContacts
336 isConvergContact = false;
337 foundContacts = CheckAllMutualContacts(ref arrOfCylinders, probab);
338 if (foundContacts == contcsOld)
339 {
340 isConvergContact = true;
341 break;
342 }
343 else
344 contcsOld = foundContacts;
345 }
346 Console.WriteLine("\t@Contacts = {0} for iteration iterCount={1}, conver={2}. Done.", foundContacts,
    iterCount-1, isConvergContact);
347
348 // ----- Finalizar el estudio del contacto y comience el estudio de la estructura de los grupos.
349 // -----
350 int n0 = 0, n1 = 0, n2 = 0, nNull = 0, nFather = 0, n3 = 0;
351 if (_calcClusters == true) //Se estudia la estructura de los clúster?s
352 StudyStructOfClusters(ref arrOfCylinders, totalCylInArr, probab); // --> _maxCylsInCluster
353 // ----- Finalizar la calculación de la estructura del clúster-----
354 Console.WriteLine("\nChecking percolation...\n");
355 double volumCylTotal2 = 0, volumCylTotal = 0, volumeAllPercolatingCyls = 0;
356 isPercolation = CheckPercolation(ref arrOfCylinders, probab
    , out volumCylTotal2 /* Volumen del clúster conformado por los cilindros*/
    , out volumCylTotal /* Volumen de todos los cilindros que no se encuentran unidos a aun clústers*/
    , out nFather
    , out nNull, out n0, out n1, out n2, out n3, out volumeAllPercolatingCyls);
357
358 int maxCylsInInfinCluster = -1;
359 StudyStructOfInfinCluster(ref arrOfCylinders, totalCylInArr, probab, out maxCylsInInfinCluster);
360 _maxCylsInInfinCluster = maxCylsInInfinCluster;
361 // Resultados:
362 _cylsReachedBott = n3; //Número de cilindros que tienen contacto
363 volumCylTotalFrac = volumCylTotal2 / volumCylTotal; // Volumen normalizado de los cilindros
364 _volumePercolCylsFrac = volumeAllPercolatingCyls / volumCylTotal;
365 // _volumCylTotalFrac = volumCylTotal2 / (N * N * N); //normalizado por todo el volumen
366 Console.WriteLine("={0}, n3={1}, clusters in bottom={2}. Done.", isPercolation, n3, nFather);
367 Console.WriteLine("VolumCylTotal={0:0.00}.", volumCylTotal);
368 string sFlag = n1 + n0 == totalCylInArr ? " " : "=====";
369 if (isPercolation == true)
370 {
371 Console.ForegroundColor = ConsoleColor.DarkGreen;
372 }
373 Console.WriteLine("\nResults:\n Percolation={0} at p={1:0.00}", isPercolation, probab);
374 Console.WriteLine("\tnNull={0}", nNull);
375 Console.WriteLine("\tEmpty n0={0}", n0);
376 Console.WriteLine("\tIsolated cyls n1={0}", n1);
377 Console.WriteLine("\tCoupled cyls (in all clusters) n2={0}", n2);
378 Console.WriteLine("\tSum=n0 + n1 + n2 = {0} (" + sFlag + "at init={1})", n0 + n1 + n2, totalCylInArr)
379 ;
380 Console.WriteLine("Total: nNull+n0+n1+n2={0}, useNNullNodes={1}", nNull + n0 + n1 + n2,
    _useNNullNodes);
381 Console.ResetColor();
382 n1 = nTop; //Corrección para los cilindros en la capa superior.
383 Console.WriteLine("POrden=n2/(n2+n1+n0+0*nNull)={0:0.00} with correction n1 by {1}", POrden, nTop);
384 //, pCrit);
385 set_POrdenCyl(POrden);
386 set_currProbability(probab);
387 //CalcPercolCyls
388 // ----- CheckAllMutualContacts -----
389 int CheckAllMutualContacts(ref OneCylinder[, ] arrOfCylinders, double probab)
390 {
391 Console.WriteLine("\nChecking the mutual contacts of cylinders in arrOfCylinders. Please wait...");
392 int ii = 0;
393
394 ProgrBarForm progrBar = new ProgrBarForm();
395 int maxCount = arrOfCylinders.GetLength(2); // N; //N * N * N; // list Cyl. Count;
396 progrBar.Text = "Check contacts. Please wait" + " " + maxCount + " + maxCount;
397 progrBar.InitProgrBar("Check contacts. Please wait...", maxCount); //!!!*/
398 //https://stackoverflow.com/questions/4299138/generate-n-random-and-unique-numbers-within-a-range
399 int[] intUniqRandArray1 = null;
400

```



```

488 #if !isCycleKLast
489 for (int k2 = 0; k2 < arrOfCylinders.GetLength(2); k2++)
490 {
491 #endif
492 for (int i2 = 0; i2 < arrOfCylinders.GetLength(0); i2++) // preparar arrOfCylinders y agregar a la
493     lista
494     {
495     for (int j2 = 0; j2 < arrOfCylinders.GetLength(1); j2++)
496     {
497     #if isCycleKLast
498     for (int k2 = 0; k2 < arrOfCylinders.GetLength(2); k2++)
499     {
500     #endif
501     OneCylinder cyl = arrOfCylinders[i2, j2, k2];
502     if (cyl == null) continue;
503     //if (cyl.get_cylID() == iiClust)
504     if (cyl.get_clusterID() == iiClustID)
505     {
506     idCyls[iiClustID]++;
507     cyl.set_cylsInCluster(idCyls[iiClustID]);
508     arrOfCylinders[i2, j2, k2] = cyl;
509     }
510     } //j2
511     //if (allowBreak && isPercolation == true) break;
512     } //i2
513     //if (allowBreak && isPercolation == true) break;
514     } //iiClustID
515     }
516     int nClusters = 0, idMax = 0, maxCylsInCluster = 0;
517     //for (int iiClust = 1; iiClust <= totalCylInArr; iiClust++) // preparar arrOfCylinders y agregar a
518     la lista
519     for (int iiClust = 0; iiClust <= totalCylInArr; iiClust++) // preparar arrOfCylinders y agregar a la
520     lista
521     {
522     if (idCyls[iiClust] > 1)
523     {
524     nClusters++;
525     if (idCyls[iiClust] > maxCylsInCluster)
526     {
527     maxCylsInCluster = idCyls[iiClust];
528     idMax = iiClust;
529     }
530     }
531     Console.WriteLine(" Done.");
532     _maxCylsInCluster = maxCylsInCluster;
533     Console.ForegroundColor = ConsoleColor.Cyan;
534     Console.WriteLine("\tAt p={0:0.00}: unconnected clusters in system={1}.", probab, nClusters);
535     Console.WriteLine("\tCluster ID={0} contains maximal number of cylinders={1} .", idMax,
536     maxCylsInCluster);
537     Console.ResetColor();
538     } //Study Struct Of Clusters
539     //----- CheckPercolation -----
540     bool CheckPercolation(ref OneCylinder[, ] arrOfCylinders, double probab, out double volumCylTotal2
541     , out double volumCylTotal, out int nFather, out int nNull, out int n0, out int n1, out int n2
542     , out int n3, out double volumeAllPercolatingCyls)
543     {
544     bool isPercolation = false;
545     bool allowBreak = false;
546     nNull = 0; n0 = 0; n1 = 0; n2 = 0; n3 = 0;
547     List<string> listFathers = new List<string>();
548     bool isPercOld = false;
549     volumCylTotal = 0; volumCylTotal2 = 0; volumeAllPercolatingCyls = 0;
550     double volumCylTotal0 = 0, volumCylTotal1 = 0;
551     #if isCycleKLast
552     int k2;
553     #else
554     for (int k2 = 0; k2 < arrOfCylinders.GetLength(2); k2++)
555     {
556     #endif
557     for (int i2 = 0; i2 < arrOfCylinders.GetLength(0); i2++) // preparar arrOfCylinders y agregar a la
558     lista
559     {
560     for (int j2 = 0; j2 < arrOfCylinders.GetLength(1); j2++)
561     {
562     #if isCycleKLast
563     for (int k2c = 0; k2c < arrOfCylinders.GetLength(2); k2c++)
564     {
565     k2 = k2c;
566     #endif
567     OneCylinder cyl = arrOfCylinders[i2, j2, k2];
568     if (cyl == null)
569     {
570     nNull++;
571     continue;
572     }
573     if (cyl.get_chType() == '0') { volumCylTotal0 += cyl.get_volumCyl(); n0++; } //isolated
574     if (cyl.get_chType() == '1' && cyl.get_cylsInCluster() == 1) { volumCylTotal1 += cyl.get_volumCyl();
575     n1++; } //single
576     if (cyl.get_chType() == '1' && cyl.get_cylsInCluster() > 1) // cilindros que son parte de cualquier
577     cluster
578     {
579     volumCylTotal2 += cyl.get_volumCyl();
580     n2++;
581     }
582     double x1 = cyl.get_x1(), y1 = cyl.get_y1(), z1 = cyl.get_z1();
583     if (cyl.get_chType() == '1' && cyl.get_cylsInCluster() > 1 && z1 >= _N - 1) // se acercó a una capa
584     inferior, percolación?
585     {
586     cyl.set_chType('2'); //Entonces hay percolación!

```

```

581 isPercolation = true;
582 if (isPercOld == false && isPercolation == true)
583 {
584   this._pCritLoc = probab; // Encontrando P_cr, Probabilidad crítica
585   isPercOld = isPercolation;
586 }
587 string sFathers = cyl.get_iFather()[0] + "," + cyl.get_iFather()[1];
588 if (listFathers.Contains(sFathers) == false)
589 {
590   listFathers.Add(sFathers);
591 }
592 volumeAllPercolatingCyls += cyl.get_volumCyl();
593 arrOfCylinders[i2, j2, k2] = cyl;
594 if (allowBreak && isPercolation == true) break;
595 } //if
596
597 if (allowBreak && isPercolation == true) break;
598 } //j2
599 if (allowBreak && isPercolation == true) break;
600 } //i2
601 } //k2
602 nFather = listFathers.Count;
603 // _cylsReachedBott = n3;
604 volumCylTotal = 0 * volumCylTotal0 + volumCylTotal1 + volumCylTotal2; // no use el volumen de
        clústeres (no agrupados)
605 return isPercolation;
606 } //CheckPercolation
607 // - - - - - StudyStructOfInfinCluster - - - - -
608 void StudyStructOfInfinCluster(ref OneCylinder[, ,] arrOfCylinders, int totalCylInArr, double probab,
        out int maxCylsInInfinCluster) //-->_maxCylsInCluster
609 {
610   Console.WriteLine("StudyStructOfInfinCluster... ");
611   int[] idCyls = new int[totalCylInArr + 1];
612   List<OneCylinder>[] arrListOfPercolClusters = new List<OneCylinder>[totalCylInArr + 1];
613   List<OneCylinder> listPercCyls = null;
614   int nNewPercCyls = -1;
615   // preparar arrOfCylinders y agregar a la lista
616   for (int iiClustID = 0; iiClustID < totalCylInArr + 1; iiClustID++)
617     // preparar arrOfCylinders y agregar a la lista
618     {
619       #if !isCycleKLast
620       for (int k2 = 0; k2 < arrOfCylinders.GetLength(2); k2++)
621       {
622         #endif
623         for (int i2 = 0; i2 < arrOfCylinders.GetLength(0); i2++)
624           // preparar arrOfCylinders y agregar a la lista
625           for (int j2 = 0; j2 < arrOfCylinders.GetLength(1); j2++)
626           {
627             #if isCycleKLast
628             for (int k2 = 0; k2 < arrOfCylinders.GetLength(2); k2++)
629             {
630               #endif
631               OneCylinder cyl = arrOfCylinders[i2, j2, k2];
632               if (cyl == null) continue;
633               if (cyl.get_chType() == '0' || cyl.get_chType() == '1') continue;
634
635               if (cyl.get_clusterID() == iiClustID)
636                 {
637                 listPercCyls = ChangeCharOfPercolCluster(ref arrOfCylinders, '2', iiClustID, out nNewPercCyls
638                 idCyls[iiClustID] = nNewPercCyls; // asigna el número de perc.
639                 // asigna el número de perc.
640                 arrListOfPercolClusters[iiClustID] = new
641                 List<OneCylinder>(listPercCyls); // agrega una nueva lista de perc. cluster
642                 //idCyls[iiClustID]++;
643                 }
644             } //if
645             } //k2
646           } //j2
647         } //i2
648       } //iiClustID
649
650   int nClusters = 0, idMax = 0;
651   maxCylsInInfinCluster = 0;
652   // preparar arrOfCylinders y agregar a la lista
653   for (int iiClust = 0; iiClust <= totalCylInArr; iiClust++)
654     preparar arrOfCylinders y agregar a la lista
655     if (idCyls[iiClust] > 0)
656     {
657       nClusters++;
658       Console.WriteLine("Infin. cluster with ID={0} has cylinders={1}", iiClust, idCyls[iiClust]);
659       if (idCyls[iiClust] > maxCylsInInfinCluster)
660       {
661         maxCylsInInfinCluster = idCyls[iiClust];
662         idMax = iiClust;
663       }
664     } //for
665   // Acumula todos los volúmenes de celdas en clúster perc. parcial
666   double volAllCylsInPercCluster = 0, lengthAllCylsInPercCluster = 0, aspRatioAllClusterMean = 0; //
667   arrListOfPercolClusters[iiClust](get)
668   for (int iiClust = 0; iiClust <= totalCylInArr; iiClust++) // preparar arrOfCylinders y agregar a la
669     lista
670     {
671       if (arrListOfPercolClusters[iiClust] != null)
672       {
673         foreach (var cyl in arrListOfPercolClusters[iiClust])
674         {
675           volAllCylsInPercCluster += cyl.get_volumCyl(); // Sumar los volúmenes de todos los ciclos de este
676           grupo de perc.
677           lengthAllCylsInPercCluster += cyl.get_h();
678           aspRatioAllClusterMean += cyl.get_h() / cyl.get_r(); // suma de todas las asp.ratio para perc. Cyl.
679           en este grupo
680         }
681       }
682       aspRatioAllClusterMean /= arrListOfPercolClusters[iiClust].Count;
683       Console.ForegroundColor = ConsoleColor.Yellow;

```

```

671 Console.WriteLine("\tInf.cluster ID={0} has cyls={1}, volAllCylsInPercCluster={2:0.00},
        aspRatioAllClusterMean={3:0.00}, lengthAllCylsInPercCluster={4:0.00}."
672 , iiClust , arrListOfPercolClusters[iiClust].Count, volAllCylsInPercCluster , aspRatioAllClusterMean ,
        lengthAllCylsInPercCluster);
673 Console.ResetColor();
674 }
675 }//for (int iiClust
676 _volAllCylsInPercCluster = volAllCylsInPercCluster; _aspRatioAllClusterMean = aspRatioAllClusterMean;
677 _lengthAllCylsInPercCluster = lengthAllCylsInPercCluster;
678 #region comment
679 #endregion
680 Console.WriteLine(" Done.");
681 Console.ForegroundColor = ConsoleColor.Yellow;
682 Console.WriteLine("\tInf.cluster ID={0} contains maximal number of cylinders={1}.", idMax,
        maxCylsInInfCluster);
683 #region comment
684 #endregion
685 Console.ResetColor();
686 }//StudyStructOfClusters
687 // - - - - - ChangeCharOfPercolCluster - - - - -
688 List<OneCylinder> ChangeCharOfPercolCluster(ref OneCylinder[, ,] arrOfCylinders, char percChr, int
        iiClustID, out int nNewPercCyls) //-->_maxCylsInCluster)
689 {
690 List<OneCylinder> listCyls = new List<OneCylinder>();
691 nNewPercCyls = 0;
692 for (int i2 = 0; i2 < arrOfCylinders.GetLength(0); i2++) // preparar arrOfCylinders y agregar a la
        lista
        {
693     for (int j2 = 0; j2 < arrOfCylinders.GetLength(1); j2++)
694     {
695         #if isCycleKLast
696         for (int k2 = 0; k2 < arrOfCylinders.GetLength(2); k2++)
697         {
698             #endif
699             OneCylinder percolCyl = arrOfCylinders[i2, j2, k2];
700             if (percolCyl == null) continue;
701             if (percolCyl.get_clusterID() == iiClustID)
702             {
703                 percolCyl.set_chType(percChr);
704                 nNewPercCyls++;
705                 arrOfCylinders[i2, j2, k2] = percolCyl;
706                 listCyls.Add(percolCyl);
707             }
708         }
709     }
710 }//i2
711 return listCyls;
712 }//ChangeCharOfPercolCluster
713
714 #region Aux
715 // - - - - - buildHist - - - - -
716 double[] buildHist(int bins, double[] data, out double max, out double min)
717 { //https://stackoverflow.com/questions/1413644/frequency-histogram-in-c
718     double[] hist = new double[bins];
719     int n = data.Length;
720     min = 1e12; max = -1e12;
721     for (int i = 0; i < n; ++i)
722     {
723         if (max < data[i]) max = data[i];
724         if (min > data[i]) min = data[i];
725     }
726
727     for (int i = 0; i < n; ++i)
728     {
729         int bin = (int)((data[i] - min) / ((max - min) / (bins)));
730         if (bin < bins)
731             if ((bin >= 0) && (bin < n)) hist[bin]++;
732     }
733     return hist;
734 }//buildHist
735 // - - - - - NextUnifRandom - - - - -
736 double NextUnifRandom(double b, double a, Random rnd)
737 {
738
739     double meanH = (b + a) / 2;
740     double x = a + (b - a) * rnd.NextDouble();
741     return x;
742 }//NextUnifRandom
743 // - - - - - PercOrdTheor - - - - -
744 public double PercOrdTheor(double p, double L, double pTh3d)
745 {
746     double nu = 0.9, beta = 0.45;
747     double normCoeff = F_scaling((p - pTh3d) * Math.Pow(L, 1 / nu), beta, 1)
748     * Math.Pow(L, -beta / nu); return normCoeff;
749 }//PercOrdTheor
750 // - - - - - F_scaling - - - - -
751 private double F_scaling(double x, double par1, double par2)
752 {
753     double y = 0;
754     if (x <= 0)
755         y = 0;
756     else
757         y = Math.Pow(x, par1) / par2; // normaliz.
758     return y < 0 ? 0 : y;
759 }//F_scaling
760 // - - - - - CorrLength - - - - -
761 private double CorrLength(double x, double par1, double beta)
762 {
763     double y = 0;
764     if (x <= 0)
765         y = 0;
766     else

```

```

767 y = parl * Math.Pow(Math.Abs(x), -beta);
768 return y < 0 ? 0 : y;
769 } //CorrLength
770 #endregion
771
772 // - - - - - ToStringShort - - - - -
773 public string ToStringShort()
774 {
775     string sOut = "Cylinder###" + ".sd=" + _seed;
776     return sOut;
777 } //ToStringShort
778 } // class WorkWithCylinders
779 // - - - - - class GuassianRandom - - - - -
780 public sealed class GuassianRandom
781 { //http://es.softuses.com/24775
782     private bool _hasDeviate;
783     private double _storedDeviate;
784     int _idum = -1;
785     private readonly Random _randomGauss = new Random(Guid.NewGuid().GetHashCode());
786     private readonly TRan3 _randomGauss3 = new TRan3();
787
788     /// <summary>
789     /// Obtiene números aleatorios distribuidos normalmente (gaussianos), usando el Box-Muller
790     /// transformación. Esta transformación toma dos desviaciones distribuidas uniformemente
791     /// dentro del círculo unitario, y los transforma en dos independientemente
792     /// distribuido desviaciones normales.
793     /// </summary>
794     /// <param name = "mu"> La media de la distribución. El valor predeterminado es cero. </param>
795     /// <param name = "sigma"> La desviación estándar de la distribución. El valor predeterminado es uno.
796     /// </param>
797     /// <returns></returns>
798     public GuassianRandom() { }
799     public GuassianRandom(int nSeed)
800     {
801         if (nSeed == 0)
802         {
803             _randomGauss = new Random(Guid.NewGuid().GetHashCode());
804             _idum = -1;
805             _randomGauss3.ran3(ref _idum);
806         }
807         else
808         {
809             _randomGauss = new Random(nSeed);
810             _idum = -nSeed;
811             _randomGauss3.ran3(ref _idum);
812         }
813     }
814     public double NextGuassian(double mu, double sigma)
815     {
816         if (sigma <= 0)
817             throw new ArgumentException("sigma", "Must be greater than zero.");
818
819         if (_hasDeviate == true)
820         {
821             _hasDeviate = false;
822             return _storedDeviate * sigma + mu;
823         }
824
825         double v1, v2, rSquared;
826         do
827         {
828             // dos valores aleatorios entre -1.0 y 1.0
829             v1 = 2 * _randomGauss3.ran3(ref _idum) - 1;
830             v2 = 2 * _randomGauss3.ran3(ref _idum) - 1;
831             rSquared = v1 * v1 + v2 * v2;
832             // asegurar dentro del círculo de la unidad
833         } while (rSquared >= 1 || rSquared == 0);
834         // calcular la transformación polar para cada desviación
835         var polar = Math.Sqrt(-2 * Math.Log(rSquared)) / rSquared;
836         // almacenar primera desviación
837         _storedDeviate = v2 * polar;
838         _hasDeviate = true;
839         // retornar segunda desviación
840         return v1 * polar * sigma + mu;
841     } //NextGuassian
842 } //GuassianRandom
843 // - - - - - class PoissonRandom - - - - -
844 public sealed class PoissonRandom
845 { //http://www.johndcook.com/csharp_poisson.html
846     Random _rndPoisson = new Random(Guid.NewGuid().GetHashCode());
847     public PoissonRandom() { }
848     public PoissonRandom(int nSeed)
849     {
850         if (nSeed > 0)
851             _rndPoisson = new Random(nSeed);
852         else
853             _rndPoisson = new Random(Guid.NewGuid().GetHashCode());
854     }
855     public int NextPoisson(double lambda)
856     {
857         return (lambda < 30.0) ? PoissonSmall(lambda) : PoissonLarge(lambda);
858     }
859     private int PoissonSmall(double lambda)
860     {
861         // Algoritmo de Donald Knuth, 1969.
862         double p = 1.0, L = Math.Exp(-lambda);
863         int k = 0;
864         do
865         {
866             k++;

```

```

867 }
868 while (p > L);
869 return k - 1;
870 }
871 private int PoissonLarge(double lambda)
872 {
873 // "Método de rechazo PA" de "The Computer Generation of
874 // Variables aleatorias de Poisson "por A. C. Atkinson,
875 // Revista de la Real Sociedad Estadística Serie C
876 // (Estadísticas aplicadas) vol. 28, núm. 1. (1979)
877 // El artículo está en las páginas 29-35.
878 // El algoritmo dado aquí está en la página 32.
879 double c = 0.767 - 3.36 / lambda;
880 double beta = Math.PI / Math.Sqrt(3.0 * lambda);
881 double alpha = beta * lambda;
882 double k = Math.Log(c) - lambda - Math.Log(beta);
883 for (; )
884 {
885 double u = _rndPoisson.NextDouble(); //Get Uniform();
886 double x = (alpha - Math.Log((1.0 - u) / u)) / beta;
887 int n = (int)Math.Floor(x + 0.5);
888 if (n < 0)
889 continue;
890 double v = _rndPoisson.NextDouble(); //Get Uniform();
891 double y = alpha - beta * x;
892 double temp = 1.0 + Math.Exp(y);
893 double lhs = y + Math.Log(v / (temp * temp));
894 double rhs = k + n * Math.Log(lambda) - gammln(n + 1); //!!!??
895 if (lhs <= rhs)
896 return n;
897 }
898 }
899 double gammln(double xx)
900 {
901 int j;
902 double x = 0;
903 double y = 0;
904 double tmp = 0;
905 double ser = 0;
906 double[] cof = { 76.18009172947146, -86.50532032941677, 24.01409824083091, -1.231739572450155,
907 0.1208650973866179e-2, -0.5395239384953e-5 };
908 y = x = xx;
909 tmp = x + 5.5;
910 tmp -= (x + 0.5) * Math.Log(tmp);
911 ser = 1.000000000190015;
912 for (j = 0; j < 6; j++)
913 ser += cof[j] / ++y;
914 return -tmp + Math.Log(2.5066282746310005 * ser / x);
915 }
916 } //class PoissonRandom

```

A.3. Código de la clase `WorkWithCylindersParal`

Esta clase controla las ejecuciones en paralelo, crea una lista de tareas en el cual se invoca a la clase: `WorkWithCylinders` y crea una lista de subrutinas con ayuda de la biblioteca: `System.Threading.Tasks`; que controla y distribuye las porciones de memoria y recursos en el sistema, y monitorea el estado de la tarea y sus estados, como cada tarea es independiente, una tarea no afecta a la otra ni viceversa a menos de que se cancele todo el proceso. Una vez que la tarea es completada, los resultados de calculación se registran en un gráfico y estos a su vez se almacenan en un archivo de resultados.

```

1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.ComponentModel;
6 using System.Windows.Forms;
7 using Plots_n;
8 using System.Threading;
9 using System.Threading.Tasks;
10 using System.Diagnostics;
11 using System.Collections;
12
13 namespace WinCylsFDTD
14 {
15 class WorkWithCylindersParal
16 {
17 #region user
18 bool _StopNowParallel = false;
19 ArrayList _winList_plot2d_1 = new ArrayList();
20 Form1 _mainWinObj = null;
21 bool _showRealTimePlot = true;

```

```

22 int nx = 50, _nAver = 1;
23 double _pCritical = 2, _pCriticalOld = 0, _pCritical0 = 0;
24 double[,] _averData = null;
25 private Plot2dWinForm2 _realTimePlotParal = new Plot2dWinForm2();
26 public void set_StopNowParallel(bool b) { _StopNowParallel = b; }
27 public Form1 get_mainWinObj() { return _mainWinObj; }
28 public void set_mainWinObj(Form1 s) { _mainWinObj = s; }
29 public void set_averCylCalc(int s) { _nAver = s; }
30 public bool get_showRealTimePlot() { return _showRealTimePlot; }
31 public double[,] get_averData() { return _averData; }
32
33 public void set_pCritical(double pc) { _pCritical = pc; }
34 public double get_pCritical() { return _pCritical; }
35 #endregion
36 // - - - - - CalcPInfCylinderForAllP_Paral - - - - -
37 public void CalcPInfCylinderForAllP_Paral(FuncPInfCylinderParal funcPInfCylinderParal, int nAver)
38 {
39     _mainWinObj.StartMyTimer();
40     #region init vars
41     int pointsToPlot = funcPInfCylinderParal.get_pointsToPlot();
42     FuncPInfCylinderParal[] pInfArray = new FuncPInfCylinderParal[pointsToPlot * nAver];
43
44     if (get_showRealTimePlot() == true && _realTimePlotParal != null)
45     {
46         _realTimePlotParal = new Plot2dWinForm2(100, 100);
47         _realTimePlotParal.Show();
48     }
49     var watch0 = Stopwatch.StartNew();
50     // - - - - - preparar calculation en paralelo
51     var tokenSource = new CancellationTokensource();
52     var token = tokenSource.Token;
53     List<Task> tasks = new List<Task>();
54     var ui = TaskScheduler.FromCurrentSynchronizationContext();
55
56     int nTasks = pointsToPlot;
57     double[,] dataPlot = initDblArr2(nTasks, nTasks, 4);
58     double hp = funcPInfCylinderParal.get_hp();
59     #endregion
60
61     int n5 = 5;
62     _averData = new double[pointsToPlot, n5];
63     double pMin = funcPInfCylinderParal.get_pMin();
64     double pMax = funcPInfCylinderParal.get_pMax();
65     _nAver = nAver;
66     string sInfoParal = "";
67
68     for (int jjAver = 1; jjAver <= nAver; jjAver++)
69     {
70         Console.WriteLine("\njjAver=" + jjAver);
71         for (int ii = 0; ii < pointsToPlot; ii++)
72         {
73             if (_StopNowParallel == true) break;
74             double currP = pMin + ii * hp;
75             Console.ForegroundColor = ConsoleColor.Green;
76             Console.WriteLine("-----\nFires thread i={0} for p={1}\n", ii, currP);
77             int jj = ii;
78             int jj0 = CalcInd(ii, jjAver, pointsToPlot, nAver);
79             sInfoParal += jj0 + " ";
80
81             funcPInfCylinderParal.set_ind(jj0);
82             pInfArray[jj0] = new FuncPInfCylinderParal(funcPInfCylinderParal);
83
84             var compute = Task.Factory.StartNew(() => //Resultados de computo
85             {
86                 SubroutPInfCylinderParallel(token, jj0, pInfArray[jj0], ref dataPlot, currP, pointsToPlot);
87             }, tokenSource.Token);
88
89             tasks.Add(compute);
90             if (get_showRealTimePlot() == true)
91             { var display = compute.ContinueWith(resultTask => //Se muestra la información del hilo en la
92                 pantalla
93                 ShowRealPlot(pInfArray[jj0], jj0, pInfArray[jj0].get_workWithCylinders().get_sInfo())
94                 , ui);
95             }
96
97     #region _StopNowParallel
98     if (_StopNowParallel == true)
99     { //http://msdn.microsoft.com/en-us/library/dd537607.aspx
100         // Solicitud de cancelación del hilo desde la GUI
101         tokenSource.Cancel();
102         Console.WriteLine("\nTask cancellation requested.");
103         // Opcional: observe el cambio en la propiedad Estado en la tarea.
104         // No es necesario esperar en las tareas que se han cancelado. Sin embargo,
105         // si espera, debe encerrar la llamada en un bloque try-catch para
106         // capturar las OperationCanceledExceptions que se lanzan. Si lo haces
107         // no espere, no se lanza OCE si el token que se pasó al
108         // El método StartNew es el mismo token que solicitó la cancelación.
109         Console.ForegroundColor = ConsoleColor.White;
110         if (_StopNowParallel == true)
111             Console.WriteLine("\nParallel calculation was interrupted.");
112         else
113             Console.WriteLine("\nParallel calculation done successfully.");
114     }
115     #region Optional_WaitOnTasksToComplete
116     try
117     {
118         Task.WaitAll(tasks.ToArray());
119     }
120     catch (AggregateException e)
121     {
122         // Para fines de demostración, mostrar el mensaje OCE.

```

```

122 foreach (var v in e.InnerExceptions)
123 Console.WriteLine("msg: " + v.Message);
124 }
125 // Comprobar que todas las tareas estén en un estado cancelado
126 for (int i7 = 0; i7 < tasks.Count; i7++)
127 Console.WriteLine("task[{0}] status is now {1}", i7, tasks[i7].Status);
128 #endregion
129 } // if
130 #endregion
131 } // for ii
132 } // jjAver
133
134 Console.WriteLine("sInfoParal = " + sInfoParal);
135
136 #region post, when completed
137 Task.Factory.ContinueWhenAll(tasks.ToArray(),
138 result =>
139 {
140 int nn = tasks.Count / nAver;
141 double[,] dtPlot = new double[nn, 2 + 2 + 2 + 2 + 1];
142 for (int jjAver = 1; jjAver <= nAver; jjAver++)
143 for (int i = 0; i < nn; i++)
144 {
145 int jj0 = CalcInd(i, jjAver, pointsToPlot, nAver);
146 if (pInfArray[jj0] != null)
147 {
148 double p = pInfArray[jj0].get_workWithCylinders().get_currProbability();
149 double L = funcPInfCylinderParal.get_workWithCylinders().get_L();
150 int reachBott = +funcPInfCylinderParal.get_workWithCylinders().get_cylsReachedBott() / nAver;
151 dtPlot[i, 0] = p;
152 dtPlot[i, 1] += pInfArray[jj0].get_workWithCylinders().get_POordenCyl() / nAver;
153
154 #if useFracVolum
155 dtPlot[i, 2] += pInfArray[jj0].get_workWithCylinders().get_volumePercolCylsFrac() * L / nAver;
156 #else
157 dtPlot[i, 2] += pInfArray[jj0].get_workWithCylinders().get_reachBott() * 1.0 / (L * L) / nAver;
158 #endif
159 //
160 dtPlot[i, 3] += funcPInfCylinderParal.get_workWithCylinders().PercOrdTheor(p, L, get_pCritical() -
161 funcPInfCylinderParal.get_hp()) / nAver;
162 dtPlot[i, 4] += pInfArray[jj0].get_workWithCylinders().get_maxCylsInCluster() / nAver;
163 dtPlot[i, 5] += pInfArray[jj0].get_workWithCylinders().get_volumCylTotalFrac() / nAver;
164 dtPlot[i, 6] += pInfArray[jj0].get_workWithCylinders().get_volAllCylsInPercCluster() / nAver;
165 dtPlot[i, 7] += pInfArray[jj0].get_workWithCylinders().get_aspRatioAllClusterMean() / nAver;
166 dtPlot[i, 8] += pInfArray[jj0].get_workWithCylinders().get_lengthAllCylsInPercCluster() / nAver;
167 }
168 }
169
170 string sInfo = funcPInfCylinderParal.get_workWithCylinders().get_sInfo();
171 Console.WriteLine("Plotting: ...");
172 Plot2dWinForm2 plot2d = new Plot2dWinForm2(DateTime.Now.ToString() + " - PInfParal");
173 plot2d.Line_DrawAll(dtPlot, nn, 0 + 3 + 2 + 2 + 1, sInfo
174 , "Order parameter.", sInfo
175 , "p", "", "Pi, RB, Pth, mx, vol, volCl, asp, lgth", 7, "sPicsFFT");
176 plot2d.Show();
177 Console.WriteLine("\tDone.");
178 _realTimePlotParal.Close();
179 _mainWinObj._winList_plot2d.Add(plot2d);
180 int cnt = dataPlot.GetLength(0);
181
182 var time = watch0.ElapsedMilliseconds;
183 string s = Math.Round(TimeSpan.FromMilliseconds(time).TotalMinutes, 0) + " min. = "
184 + Math.Round(TimeSpan.FromMilliseconds(time).TotalSeconds, 0) + " sec.";
185
186 string sMin = Math.Round(TimeSpan.FromMilliseconds(time).TotalMinutes, 1) + " min. ";
187 string sSec = Math.Round(TimeSpan.FromMilliseconds(time).TotalSeconds, 3) + " sec.";
188
189 if (get_pCritical() < 1)
190 Console.ForegroundColor = ConsoleColor.Red;
191 #else
192 Console.ForegroundColor = ConsoleColor.Magenta;
193 Console.WriteLine("pCritical={0:0.000}", get_pCritical());
194 Console.ResetColor();
195
196 Console.ForegroundColor = ConsoleColor.White;
197 if (_StopNowParallel == true)
198 Console.WriteLine("\nParallel calculation was interrupted.");
199 #else
200 Console.WriteLine("\nParallel calculation done succesfully.");
201 Console.WriteLine("\nTotal time: " + s);
202 Console.WriteLine("\n - - - - - Finish parallel. - - - - - \n");
203 Console.ResetColor();
204
205 Console.WriteLine(funcPInfCylinderParal.ToStringLong());
206 _mainWinObj.StoptMyTimer();
207 _mainWinObj.WriteTextToInfo("\nTotal time: " + s);
208 MessageBox.Show("Completed CalcPInfCylinderForAllP_Paral");
209 }, CancellationToken.None, TaskContinuationOptions.None, ui);
210 #endregion
211 } // CalcPInfCylinderForAllP_Paral
212 // - - - - - SubroutPInfCylinderParallel - - - - -
213 void SubroutPInfCylinderParallel(CancellationToken ct, int i, FuncPInfCylinderParal pInfCylParal, ref
214 double[,] dataPlot2D
215 , double currP, int pointsToPlot)
216 {
217 if (_StopNowParallel == true)
218 {
219 Console.ForegroundColor = ConsoleColor.DarkGreen;
220 Console.WriteLine("Task {0} was cancelled before it got started.", i);
221 Console.ResetColor();
222 ct.ThrowIfCancellationRequested();

```

```

221 }
222
223 if (_StopNowParallel == false)
224 {
225 int ID = Thread.CurrentThread.ManagedThreadId;
226 Console.ResetColor();
227 Console.ForegroundColor = ConsoleColor.DarkCyan;
228 Console.WriteLine("Executing task={0} (ThreadID={1}) for p={2:0.00}.", i, ID, currP);
229 Console.ResetColor();
230
231 var watch = new Stopwatch();
232 watch.Start();
233 // - - - - -
234
235 pInfCylParal.get_workWithCylinders().set_currentThread(i);
236 pInfCylParal.get_workWithCylinders().set_currProbability(currP);
237 pInfCylParal.get_workWithCylinders().CalcPercolCyls();
238
239 double pp = pInfCylParal.get_workWithCylinders().get_pCritLoc();
240 if (get_pCritical() > pp)
241 {
242 set_pCritical(pp);
243 _pCritical0 = _pCriticalOld;
244 }
245 _pCriticalOld = currP;
246 Console.ForegroundColor = ConsoleColor.Yellow;
247 Console.WriteLine("Done task #" + i + ". Watch: " + watch.ElapsedMilliseconds + " msec. = "
248 + (watch.ElapsedMilliseconds / 60000) + " min. " + (watch.ElapsedMilliseconds / 1000) + " sec."
249 + (watch.ElapsedMilliseconds * 1000 + " msec.\n");
250 Console.ResetColor();
251 }
252 } // SubroutPInfParallel
253 // - - - - - ShowRealPlot - - - - -
254 void ShowRealPlot(FuncPInfCylinderParal funcPInfCylinderParal, int j, string sInfo)
255 {
256 Console.WriteLine("(" + j + "). " + funcPInfCylinderParal.ToStringLong());
257 double p = funcPInfCylinderParal.get_workWithCylinders().get_currProbability();
258 double L = funcPInfCylinderParal.get_workWithCylinders().get_L();
259
260 #if useFracVolum
261 double volumePercolCylsFrac = funcPInfCylinderParal.get_workWithCylinders().get_volumePercolCylsFrac
262 () * L;
263 #else
264 int reachBott = funcPInfCylinderParal.get_workWithCylinders().get_reachBott();
265 #endif
266
267 int maxClust = 0;
268 double volumCylTotal = funcPInfCylinderParal.get_workWithCylinders().get_volumCylTotalFrac();
269
270 double pTheor = 0;
271 double parOrd = funcPInfCylinderParal.get_workWithCylinders().get_POrdenCyl();
272
273 #if useFracVolum
274 double[] yy = { p, parOrd, volumePercolCylsFrac, maxClust, volumCylTotal };
275 #else
276 double[] yy = { p, parOrd, reachBott*1.0 / (L * L), maxClust, volumCylTotal };
277 #endif
278 _realTimePlotParal.bgnRealAdd(true, p, yy, "Pi,rB,mx,vol,", "p");
279 _realTimePlotParal.set_Header(sInfo);
280 } // ShowRealPlot
281
282 #region Aux
283 int CalcInd(int i, int j, int iFin, int jFin) { return i + iFin * (j - 1); }
284 // - - - - - initDb1Arr - - - - -
285 void initDb1Arr(out double[][] Ar, int nDim1, int nDim2)
286 {
287 Ar = new double[nDim1][];
288 for (int i = 0; i < nDim1 + 0; i++)
289 Ar[i] = new double[nDim2];
290 } // initDb1Arr
291 // - - - - - initDb1Arr2 - - - - -
292 void initDb1Arr2(out double[][] Ar, int nDim1, int nDim2)
293 {
294 Ar = new double[nDim1][,];
295 for (int i = 0; i < nDim1 + 0; i++)
296 Ar[i] = new double[nDim2, nDim2];
297 } // initDb1Arr2
298 // - - - - - initDb1Arr2 - - - - -
299 double[][] initDb1Arr2(int nDim1, int nDim2, int nDim3)
300 {
301 double[][] Ar = new double[nDim1][,];
302 for (int i = 0; i < nDim1 + 0; i++)
303 Ar[i] = new double[nDim2, nDim3];
304 return Ar;
305 } // initDb1Arr2
306
307 #endregion
308 } // class
309 }

```

CÓDIGO UTILIZADO EN LA RADIACIÓN DE CHERENKOV SOBRE CNTs

Para nuestra simulación numérica aplicada a la radiación de Cherenkov sobre un arreglo de nanoestructuras de CNTs, se programaron las clases: *Arrays.cs*, *AuxClasses.cs*, *BGNFunc.cs*, *BgnReadFiles.cs*, *CalcDiffTime.cs*, *CherenBase.cs*, *Cherenkov.cs*, *CherenParall.cs*, *complex.cs*, *complex4Str.cs*, *ComplexFFT.cs*, *FarField.cs*, *FdtdAllLib.cs*, *FdtdVars.cs*, *FftVizNum.cs*, *Field3d.cs*, *FiveLayers.cs*, *FiveLayersParall.cs*, *FuncPInfCylinderParal.cs*, *Gasdev.cs*, *Input.cs*, *LogFile.cs*, *Memory.cs*, *MetalObj.cs*, *Output.cs*, *Parameters.cs*, *plot2d-n.Designer.cs*, *PrintMatrix.cs*, *PrintMatrixFormat.cs*, *Ran3.cs*, *RandBgn.cs*, *Result-distance.cs*, *RWAarray3D.cs*, *sFun.cs*, *SurfacePlasmonCyls.cs*, *SurfacePlasmonDielectric.cs*, *TConstants.cs*, *TimeUpdate.cs*, *Transmap.cs* y *WinBars2.Designer.cs*, *Energy.cs*. En este apéndice abordaremos algunos fragmentos de códigos más importantes utilizados en la programa. Presentamos los códigos principales para nuestra programación.

B.0.1. Clase FdtdVars

En esta clase se definen valores y variables para el calculo del modelo de Drude, la permitividad dieléctrica ϵ (epsilon) y la frecuencia omega ω , y se definen valores para los cálculos espaciales de los campos: *PosFromSpaceX*, *PosFromSpaceY* y *PosFromSpaceZ* respectivamente.

```

1 namespace WinCalifFDTD_withClassesLib
2 {
3     #region Definitions of structures
4     internal struct obj
5     {
6         internal string shape;
7         internal double centeri;
8         internal double centerj;
9         internal double centerk;
10        internal double size1;
11        internal double size2;
12        internal double size3;
13        internal double epsilon;
14        internal double[,] matrix; // contour matrix data
15    }
  
```

```

16 internal int col; // matrix col num
17 internal int row; // matrix row num
18 }; //obj
19 //
20 public struct mobj
21 {
22 internal string shape;
23 internal double centeri;
24 internal double centerj;
25 internal double centerk;
26 internal double size1; //////////////// Drude model //////////////////////
27 internal double size2; ////////////////
28 internal double size3; //////////////// (omega_p)^2 //
29 internal double epsilon_b; // eps(w) = eps_b - ----- //
30 internal double omega_p; // w^2 + j w gamma_0 //
31 internal double gamma_0; ////////////////
32 internal double[,] matrix; // contour matrix data
33 internal int col; // matrix col num
34 internal int row; // matrix row num
35 }; //mobj
36 //
37 internal struct ngrid_info // for non-uniform grid function
38 {
39 internal int lattice_nz;
40 internal int nz_start;
41 internal int nz_end;
42 }; //ngrid_info
43 //
44 //////////////// Drude model //////////////////////
45 //////////////// (omega_p)^2 //
46 // eps(w) = eps_b - ----- // Note: The former mobj is now generalized to Lobj.
47 // w^2 + j w gamma_0 //
48 ////////////////
49 ////////////////
50 internal struct Lobj
51 {
52 internal string shape;
53 internal double centeri;
54 internal double centerj;
55 internal double centerk;
56 internal double size1;
57 //////////////// Lorentz model //////////////////////
58 internal double size2; //////////////// (omega_p)^2 //
59 internal double size3; ////////////////
60 internal double epsilon_b; // eps(w) = eps_b + ----- //
61 internal double omega_p; // omega_0^2 - w^2 - j w gamma_0 //
62 internal double gamma_0; ////////////////
63 internal double omega_0; ////////////////
64 internal double[,] matrix; // contour matrix data
65 internal int col; // matrix col num
66 internal int row; // matrix row num
67 };
68 //
69 internal struct DLobj
70 {
71 internal string shape;
72 internal double centeri;
73 internal double centerj;
74 internal double centerk;
75 internal double size1; //////////////// Double Lorentz Model
76 internal double size2; ////////////////
77 internal double size3; //////////////// (omega_p_1)^2 (-1)^S* //
78 internal double epsilon_b; // eps(w) = eps_b + ----- +
79 // omega_p_2^2 // omega_0_1^2 - w^2 - j w gamma_0_1 omega_0_2^2 - w^2
80 // - j w gamma_0_2 //
81 ////////////////
82 internal double omega_0_1;
83 internal double omega_p_2;
84 internal double gamma_0_2;
85 internal double omega_0_2;
86 internal double[,] matrix; // contour matrix data
87 internal int col; // matrix col num
88 internal int row; // matrix row num
89 };
90 // - - - - struct k_vector - - - - -
91 internal struct k_vector
92 {
93 internal double x;
94 internal double y;
95 };
96 //
97 #endregion
98 //
99 class Vect3D
100 {
101 /*internal double _x { get; set; }
102 internal double _y { get; set; }
103 internal double _z { get; set; }*/
104 internal double _x=0, _y=0, _z=0;
105 //
106 internal Vect3D() { _x = 0; _y = 0; _z = 0; }
107 internal Vect3D(double x, double y, double z) { _x = x; _y = y; _z = z; }
108 internal void SetVect3D(double x, double y) { _x = x; _y = y; _z = 0; }
109 internal void SetVect3D(double x, double y, double z) { _x = x; _y = y; _z = z; }
110 }

```

```

111
112 class FdtdVars
113 {
114 #region User
115
116 internal Form1 _mainWinObj = null;
117 internal readonly TcompD I = new TcompD(0, 1);
118 //internal const int RAND_MAX = 0x7fff; //!!!??
119 internal const int CUTFACTOR = 1;
120 internal const string _EMP = null;
121 internal const int _CUTFACTOR = 1;
122 internal const double _KFDTDver = 8.803;
123 // -- int --
124 internal int _metal_type = -1;
125 internal int _currentThread = -1;
126 internal int _maxThread = -1;
127 internal int _deltaT = -1;
128 internal int _tMid = -1;
129 internal int _pointsFft = -1;
130 internal int _tMax = -1;
131 // -- double --
132 internal double _freq = -1;
133 internal double _phase = -1;
134 internal double _height = -1;
135 internal double _R = -1;
136 internal double _rangeRandom = -1;
137 internal double _heightMax = -1;
138 internal double _detect = -1;
139 internal double _thresPercol = 0, _vxCurr=0, _omPlasmaCurr=0;
140 internal double _omegaSoc=0;
141 internal double _wave_vector_x = 0;
142 internal double _wave_vector_y = 0;
143 double _fiedBound0 = 0, _fiedBound1 = 0, _randProbab;
144 double _a_nm = 0;
145 // -- string --
146 internal string _component = null;
147 internal string _sthradID0 = "";
148 string _lrsStruct = null;
149 // -- bool --
150 internal bool _allowWriteToFile = true;
151 internal bool _allowPercol = true;
152 internal bool _isSymmert = true;
153 //internal bool _isMovingSoc = true;
154
155 // -- array --
156 internal double[,] _epsPlaneX = null;
157 internal double[,] _epsPlaneY = null;
158 internal double[,] _epsPlaneZ = null;
159
160 /* input dielectric structure */
161 internal char[,] _position;
162 // -- double array --
163 internal double[,] _epsilonX, _epsilonY, _epsilonZ;
164 internal double[,] _mEpsilon, _mOmega, _mGamma;
165
166 internal double[] _Omega = null;
167
168 /* field components */
169 internal double[,] fieldOut = null;
170 internal double[,] _Ex, _Ey, _Ez; // all fields will init in double[,] float_3d_memory(int imax,
171 // and clear in void ClearFields(int i, int j, int k, double d)
172 //internal double[,] _Ex0, _Ey0, _Ez0;
173 //internal double[,] _Hx0, _Hy0, _Hz0;
174 internal double[,] _Jx, _Jy, _Jz;
175 internal double[,] _Jmx, _Jmy, _Jmz;
176 internal double[,] _Jx0, _Jy0, _Jz0;
177 internal double[,] _Jmx0, _Jmy0, _Jmz0;
178 internal double[,] _Hx, _Hy, _Hz;
179 internal double[,] _Dx, _Dy, _Dz;
180 internal double[,] _Bx, _By, _Bz;
181
182 internal double[,] _Ex1 = null, _Ey1 = null, _Ez1 = null; //components of el. field
183 internal double[,] _Hx1 = null, _Hy1 = null, _Hz1 = null; //components of magn. field
184 internal double[,] _Jx1 = null, _Jy1 = null, _Jz1 = null; //components of magn. field
185 internal double[,] _Jmx1 = null, _Jmy1 = null, _Jmz1 = null; //components of magn. field
186
187 /* imaginary field (periodic FDTD) */
188 internal double[,] _iEx, _iEy, _iEz;
189 internal double[,] _iJx, _iJy, _iJz;
190 internal double[,] _iJmx, _iJmy, _iJmz;
191 internal double[,] _iHx, _iHy, _iHz;
192 internal double[,] _iDx, _iDy, _iDz;
193 internal double[,] _iBx, _iBy, _iBz;
194
195 /* FDTD update coefficients */
196 internal double[,] _ddx, _ddy, _ddz;
197
198 internal double[,] _aaX, _aaY, _aaZ;
199 internal double[,] _bbX, _bbY, _bbZ;
200 internal double[,] cex, ccy, ccz;
201
202 internal double[,] eex, eey, eez;
203 internal double[,] ffx, ffy, ffz;
204 internal double[,] ggx, ggy, ggz;
205 internal double[,] hhx, hhy, hhz;
206 internal double[,] iix, iiy, iiz;
207 internal double[,] jjx, jjy, jjz;
208 internal double[,] kkx, kky, kkz;
209 internal double[,] llx, lly, llz;
210

```



```

308 public FtdtVars() { }
309
310
311 #region virtual
312 internal virtual double[] SourceDisper(double x, double y, double z, double t) // source is in class
    TFourLevel
313 { throw new Exception("Attempt to call virtual double[] SourceDisper"); }
314 internal virtual void SourceDisper(double x, double y, double z, double t, ref double[] soc)
315 { throw new Exception("Attempt to call virtual void SourceDisper"); } // source is in class TFourLevel
316 #endregion
317
318 // -----
319 internal void ClearFields(int i, int j, int k, double d)
320 {
321     _Ex[i, j, k] = _Ey[i, j, k] = _Ez[i, j, k] = _Hx[i, j, k] = _Hy[i, j, k] = _Hz[i, j, k]
322     = _Jx[i, j, k] = _Jy[i, j, k] = _Jz[i, j, k] = _Jmx[i, j, k] = _Jmy[i, j, k] = _Jmz[i, j, k] = d;
323 }
324
325 internal void ClearFields()
326 {
327     for (int i = 0; i < _Ex.GetLength(0); i++)
328     for (int j = 0; j < _Ex.GetLength(1); j++)
329     for (int k = 0; k < _Ex.GetLength(2); k++)
330         _Ex[i, j, k] = _Ey[i, j, k] = _Ez[i, j, k] = _Hx[i, j, k] = _Hy[i, j, k] = _Hz[i, j, k]
331         = _Jx[i, j, k] = _Jy[i, j, k] = _Jz[i, j, k]
332         = _Dx[i, j, k] = _Dy[i, j, k] = _Dz[i, j, k] = _Bx[i, j, k] = _By[i, j, k] = _Bz[i, j, k]
333         = 0;
334
335     for (int i = 0; i < _Ex_sin.GetLength(0); i++)
336     for (int j = 0; j < _Ex_sin.GetLength(1); j++)
337     for (int k = 0; k < _Ex_sin.GetLength(2); k++)
338         _Ex_cos[i, j, k] = _Ex_sin[i, j, k] = _Ey_cos[i, j, k] = _Ey_sin[i, j, k]
339         = _Hx_cos[i, j, k] = _Hx_sin[i, j, k] = _Hy_cos[i, j, k] = _Hy_sin[i, j, k]
340         = 0;
341 } // ClearFields
342
343 // ----- PosFromSpaceX -----
344 internal int PosFromSpaceX(double x, out double xml)
345 {
346     int i = (int)Math.Floor(0.5 + ((x + _xCenter) * _lattice_x));
347     xml = (i - 0.5) / _lattice_x - _xCenter;
348     int im1 = (int)Math.Floor(0.5 + ((xml + _xCenter) * _lattice_x));
349     if (im1 != i - 1)
350     {
351         int rr = 0;
352     }
353     return i;
354 } // PosFromSpaceX
355 // ----- PosFromSpaceY -----
356 internal int PosFromSpaceY(double y, out double yml)
357 {
358     int i = (int)Math.Floor(0.5 + ((y + _yCenter) * _lattice_y)); // i
359     yml = (i - 0.5) / _lattice_y - _yCenter; // y[i-1]
360     int im1 = (int)Math.Floor(0.5 + ((yml + _yCenter) * _lattice_y)); // i-1
361     if (im1 != i - 1)
362     {
363         int rr = 0;
364     }
365     return i;
366 } // PosFromSpaceY
367 // ----- PosFromSpaceZ -----
368 internal int PosFromSpaceZ(double z, out double zml)
369 {
370     int i = (int)Math.Floor(0.5 + ((z + _zCenter) * _lattice_z));
371     zml = (i - 0.5) / _lattice_z - _zCenter;
372     int im1 = (int)Math.Floor(0.5 + ((zml + _zCenter) * _lattice_z));
373     if (im1 != i - 1)
374     {
375         int rr = 0;
376     }
377     return i;
378 } // PosFromSpaceZ
379 #endregion

```

B.1. Código para la Radiación de Cherenkov

La clase Cherenkov es la clase que es invocada para la interacción con nuestro sistema CNT, en esta clase se definen los métodos que invocan subrutinas para la creación del campos electromagnéticos, así como el punto de sobre la malla donde se establece la carga inicial y las dimensiones de los bloques numéricos involucrados.

```

1 namespace WinCylsFDTD
2 {
3   class Cherenkov : CherenBase
4   {
5
6     // - - - - - CreateCherenStruct - - - - -
7     void CreateCherenStruct(ref FdtdAllLib fdtdAllLibLoc)
8     {
9       fdtdAllLibLoc.set_gen_numberRandObj(_nRandObj);
10      fdtdAllLibLoc.set_seedRand(_SeedRandValueComboBox);
11
12      double epsBlock = _fdtdAllLib.get_slab_epsilon();
13      logFile.WriteLog("_fdtdAllLib: Fill rods");
14      get_mainWinObj().WriteTextToInfo("Fill: rods\n"); Application.DoEvents();
15      if (_center313 == true)
16      {
17        fdtdAllLibLoc.input_object("rod", EMP, 0, 0, 0 + fdtdAllLibLoc.get_shift(), (R + 0.01), heightZ, 0,
18          epsBlock); //center
19        fdtdAllLibLoc.input_object("rod", EMP, -1, 0, 0 + fdtdAllLibLoc.get_shift(), (R + 0.01), heightZ, 0,
20          epsBlock); //1
21        fdtdAllLibLoc.input_object("rod", EMP, 1, 0, 0 + fdtdAllLibLoc.get_shift(), (R + 0.01), heightZ, 0,
22          epsBlock); //2
23        fdtdAllLibLoc.input_object("rod", EMP, 0.5, (Math.Sqrt(3) * 0.5), 0 + fdtdAllLibLoc.get_shift(), (R +
24          0.01), heightZ, 0, epsBlock); //3
25        fdtdAllLibLoc.input_object("rod", EMP, -0.5, (Math.Sqrt(3) * 0.5), 0 + fdtdAllLibLoc.get_shift(), (R
26          + 0.01), heightZ, 0, epsBlock); //4
27        fdtdAllLibLoc.input_object("rod", EMP, 0.5, -(Math.Sqrt(3) * 0.5), 0 + fdtdAllLibLoc.get_shift(), (R
28          + 0.01), heightZ, 0, epsBlock); //5
29        fdtdAllLibLoc.input_object("rod", EMP, -0.5, -(Math.Sqrt(3) * 0.5), 0 + fdtdAllLibLoc.get_shift(), (R
30          + 0.01), heightZ, 0, epsBlock); //6
31      }
32      //object List Separ Fill - - - - -
33    }
34    #endregion
35    // Dig:
36    #region Dig
37    if (_center33 == true)
38    {
39      fdtdAllLibLoc.input_object("rod", EMP, (1 + R - Rm) * -1, 0, 0 + fdtdAllLibLoc.get_shift(), Rm,
40        heightZ, 0, epsExternal); //1
41      fdtdAllLibLoc.input_object("rod", EMP, (1 + R - Rm) * 1, 0, 0 + fdtdAllLibLoc.get_shift(), Rm,
42        heightZ, 0, epsExternal); //2
43      fdtdAllLibLoc.input_object("rod", EMP, ((1 + R - Rm) * 0.5), (1.1 * Math.Sqrt(3) * 0.5), 0 +
44        fdtdAllLibLoc.get_shift(), Rm, heightZ, 0, epsExternal); //3
45      fdtdAllLibLoc.input_object("rod", EMP, ((1 + R - Rm) * -0.5), (1.1 * Math.Sqrt(3) * 0.5), 0 +
46        fdtdAllLibLoc.get_shift(), Rm, heightZ, 0, epsExternal); //4
47      fdtdAllLibLoc.input_object("rod", EMP, ((1 + R - Rm) * 0.5), (1.1 * -Math.Sqrt(3) * 0.5), 0 +
48        fdtdAllLibLoc.get_shift(), Rm, heightZ, 0, epsExternal); //5
49      fdtdAllLibLoc.input_object("rod", EMP, ((1 + R - Rm) * -0.5), (1.1 * -Math.Sqrt(3) * 0.5), 0 +
50        fdtdAllLibLoc.get_shift(), Rm, heightZ, 0, epsExternal); //6
51    }
52    #endregion
53    #region Side blocks
54    if (_side == true)
55    {
56      logFile.WriteLog("_fdtdAllLib: Fill blocks");
57      get_mainWinObj().WriteTextToInfo("Fill: blocks\n"); Application.DoEvents();
58      _fdtdAllLib.input_object("block", EMP, (lengthX / 2), 0, 0 + _fdtdAllLib.get_shift(), 1.2, lengthY +
59        1, heightZ, epsBlock);
60      _fdtdAllLib.input_object("block", EMP, -(lengthX / 2), 0, 0 + _fdtdAllLib.get_shift(), 1.2, lengthY +
61        1, heightZ, epsBlock);
62      _fdtdAllLib.input_object("block", EMP, 0, (lengthY / 2), 0 + _fdtdAllLib.get_shift(), (lengthX + 1),
63        1.2, heightZ, epsBlock);
64      _fdtdAllLib.input_object("block", EMP, 0, -(lengthY / 2), 0 + _fdtdAllLib.get_shift(), (lengthX + 1),
65        1.2, heightZ, epsBlock);
66    }
67    #endregion
68    #region out_epsilon
69    if (nRods > 0)
70    {
71      fdtdAllLibLoc.out_epsilon("x", 0, "epsilon.x");
72      //void out_epsilon(char *plane, float value, char *name):
73      /* Print out a 2-D text matrix containing one of cross sections of the whole FDTD structure.
74       * plane can take one of the followings, %x?, %y?, and %z?. value determines the location of the
75       * plane on which a cross sectional image of the FDTD epsilon structure is to be obtained.
76       * For example, a structure image on the z = 0 plane will be obtained with parameter
77       * settings of *plane=%z? and value=0. The unit of value is the FDTD unit length, a. *name
78       * requires the output file name including the file extension.
79       */
80      fdtdAllLibLoc.out_epsilon("y", 0, "epsilon.y");
81      fdtdAllLibLoc.out_epsilon("z", 0 + fdtdAllLibLoc.get_shift(), "epsilon.z");
82      if (_ShowEpsZCheckBox == true)
83      get_mainWinObj().CreateNew3dPlot3(fdtdAllLibLoc._epsPlaneZ, "t=" + "Z"
84        , "epsZ(x,y,z=c)"
85        , _fdtdAllLib.ToStringCher3(), "legend", 30, 120, 400, 500, get_mainWinObj()._winList_surfPlotEps,

```

```

        true, false, 0);
69
70
71 if (_ShowEpsXYCheckBox == true)
72 {
73   get_mainWinObj().CreateNew3dPlot3(fdtAllLibLoc._epsPlaneX, "t=" + "X"
74   , "epsX(x,y,z=c)"
75   , _fdtdAllLib.ToStringCher3(), "legend", 30, 120, 400, 500, get_mainWinObj()._winList_surfPlotEps,
76   true, false, 0);
77
78   get_mainWinObj().CreateNew3dPlot3(fdtAllLibLoc._epsPlaneY, "t=" + "Y"
79   , "epsY(x,y,z=c)"
80   , _fdtdAllLib.ToStringCher3(), "legend", 30, 120, 400, 500, get_mainWinObj()._winList_surfPlotEps,
81   true, false, 0);
82 }
83
84
85 // - - - - - GenerateCylStructure - - - - -
86 void GenerateCylStructure(FdtdAllLib fdtAllLib, bool isRodZ = true, bool isRodX = false, bool isRodY
87 = false)
88 {
89   Console.WriteLine("Enter to GenerateCylStructure: isRodZ={0}, isRodX={1}, isRodY={2}.", isRodZ,
90   isRodX, isRodY);
91   double spatScale = 1, centerU = 0 + _fdtdAllLib.get_shift(), centerV = 1;           double heightZ =
92   _height, R = _rRods;
93   double lengthP = 1, lengthQ = 1;
94   double lengthX = fdtAllLib.get_xsize(), lengthY = fdtAllLib.get_y_size(), lengthZ = fdtAllLib.
95   get_zsize();
96   double lattice_n = 100e-9;
97   bool allowPercol = fdtAllLib._allowPercol;
98   double thresh = fdtAllLib._thresPercol;
99   var EMP = FdtdAllLib._EMP;
100  double hRad=1;
101  double Rm = _rmRods;//0.25;
102  Random rnd = new Random();
103  double p = 0; //percolation
104  for (double pp = -(lengthX / 2); pp <= (lengthX / 2); pp++)
105  for (double qq = -(lengthY / 2); qq <= (lengthY / 2); qq++)
106  {
107    if (allowPercol == true)
108    p = (rnd.NextDouble() < thresh) ? 0 : 1;
109    else
110    p = 1;
111    double centerP = pp * spatScale;
112    double centerQ = Math.Sqrt(3) * qq * spatScale;
113    //heightZ = centerV,
114    double heightX = heightZ, heightY = heightZ;
115    double centerx = 0, centery = 0, centerz = 0;
116    centerx = centerP; centery = centerQ; centerz = centerU; //P, Q, U
117    double mltZ = get_cherMultCylsXY();
118    if (isRodZ == true) //Z
119    {
120      if (_isMetalCyls == true) // generate metall cylinder
121      {
122        //w_p = angular frequency in unit of _Hz
123        fdtAllLib.input_Drude_medium("rod", centerx, centery, centerz, R, heightZ * p * hRad, 0,
124        _epsCylFiveLrs, _omega_p, _gamma_0, lattice_n);
125        if (_isEmptyCylsInMetalCyls == true) //put empty cylinder in central part of metall cylinder
126        fdtAllLib.input_object("rod", EMP, centerx, centery, centerz, R * 0.8, heightZ * p * hRad, 0,
127        _EpsInCylCenter); // <-- empty cylinder in center of metall cylinder
128      }
129      else // generate dielectric cylinder
130      fdtAllLib.input_object("rod", EMP, centerx, centery, centerz, R, heightZ * p * hRad, 0,
131        _epsCylFiveLrs);
132    }
133    if (isRodX == true) //X
134    {
135      double sz2 = (heightZ * p * hRad) * mltZ;
136      if (_isMetalCyls == true) // generate metall cylinder
137      {
138        //w_p = angular frequency in unit of _Hz
139        fdtAllLib.input_Drude_medium("rodX", centerx, centery, centerz, R, sz2, 0, _epsCylFiveLrs, _omega_p,
140        _gamma_0, lattice_n);
141        if (_isEmptyCylsInMetalCyls == true) //put empty cylinder in central part of metall cylinder
142        fdtAllLib.input_object("rodX", EMP, centerx, centery, centerz, R * 0.8, sz2, 0, _EpsInCylCenter); //
143        <-- empty cylinder in center of metall cylinder
144      }
145      else // generate dielectric cylinder
146      fdtAllLib.input_object("rodX", EMP, centerx, centery, centerz, R, sz2, 0, _epsCylFiveLrs);
147    }
148    if (isRodY == true) //Y
149    {
150      double sz2 = (heightZ * p * hRad) * mltZ;
151      if (_isMetalCyls == true) // generate metall cylinder
152      {
153        //w_p = angular frequency in unit of _Hz
154        fdtAllLib.input_Drude_medium("rodY", centerx, centery, centerz, R, sz2, 0, _epsCylFiveLrs, _omega_p,
155        _gamma_0, lattice_n);
156        if (_isEmptyCylsInMetalCyls == true) //put empty cylinder in central part of metall cylinder
157        fdtAllLib.input_object("rodY", EMP, centerx, centery, centerz, R * 0.8, sz2, 0, _EpsInCylCenter); //
158        <-- empty cylinder in center of metall cylinder
159      }
160      else // generate dielectric cylinder
161      fdtAllLib.input_object("rodY", EMP, centerx, centery, centerz, R, sz2, 0, _epsCylFiveLrs);
162    }
163  }
164  } // for x,y
165  } //GenerateCylStructure
166  // - - - - - MainCherenkov - - - - -
167  public void MainCherenkov()

```

```

156 {
157     Console.ForegroundColor = ConsoleColor.Yellow;
158     Console.WriteLine("\t\n- - - - Entering to MainCherenkov - - - ");
159     Console.ForegroundColor = ConsoleColor.Gray;
160     get_mainWinObj().set_CherInfoRichTextBox("");
161
162     #region init
163     int tMax = _tMax;
164     get_mainWinObj().WriteTextToInfo(String.Format("\ntMax={0}\n", tMax));
165     _fdtdAllLib = new FdtdAllLib();
166     if (_fdtdAllLib == null || _structWasChanged == true)
167     {
168         if (_structWasChanged == true)
169             MessageBox.Show("The structure of system was changed and will be recalculated.");
170         _structWasChanged = false;
171         try
172         {
173             CreateCherenStruct(ref _fdtdAllLib);
174         }
175         catch (Exception ee)
176         {
177             if (_progrBar != null) _progrBar.Close();
178             MessageBox.Show(ee.Message + ". Exit.");
179             return;
180         }
181         Console.ForegroundColor = ConsoleColor.Yellow;
182         Console.WriteLine("\t\n- - - - Continue with Cherenkov - - - ");
183         Console.ForegroundColor = ConsoleColor.Gray;
184     }
185
186     _fdtdAllLib.propagate(); //FDTD time updates
187     #region real plot and touch at boundary
188     int indField = sFieldToPlot == "Ex" ? 0 : sFieldToPlot == "Ey" ? 1 : sFieldToPlot == "Ez" ? 2
189     : sFieldToPlot == "Hx" ? 3 : sFieldToPlot == "Hy" ? 4 : sFieldToPlot == "Hz" ? 5 : ErrorMessage("
190         Wrong indField"); // Error
191     if (isVxVelocity == true) //realtime plot
192     {
193         try
194         {
195             fieldsInBoundsXFin = _fdtdAllLib.GetFieldVectorInGrid(xFin, 0, 0); //X
196         }
197         catch (Exception e)
198         {
199             MessageBox.Show("1" + e.Message);
200         }
201         try
202         {
203             fieldsInBoundsXStart = _fdtdAllLib.GetFieldVectorInGrid(xStart0, 0, 0);
204         }
205         catch (Exception e)
206         {
207             MessageBox.Show("2" + e.Message);
208         }
209
210         fieldsInBoundsXCenter = _fdtdAllLib.GetFieldVectorInGrid(0, 0, 0);
211         boundValue3 = _fdtdAllLib.GetFieldVectorInGrid(xStart0 / 2, 0, 0);
212         boundValue5 = _fdtdAllLib.GetFieldVectorInGrid(3 * xStart0 / 4, 0, 0);
213         boundValue4 = _fdtdAllLib.GetFieldVectorInGrid(xFin / 2, 0, 0); //X
214         yyPlot = new double[] { t, fieldsInBoundsXStart[indField], boundValue5[indField], boundValue3[
215             indField],
216             fieldsInBoundsXCenter[indField], boundValue4[indField], fieldsInBoundsXFin[indField] };
217         lst7.Add(yyPlot);
218     }
219     { //>>>
220         for (int iField = 0; iField < 6; iField++) //plot 3D all components
221         {
222             yyPlot = new double[] { t, fieldsInBoundsXStart[iField], boundValue5[iField], boundValue3[iField],
223                 fieldsInBoundsXCenter[iField], boundValue4[iField], fieldsInBoundsXFin[iField] };
224             lst7All[iField].Add(yyPlot);
225         }
226
227         double drudeEner = _fdtdAllLib.Drude_energy_loss_in_block(_blockWithCyls._centerx, _blockWithCyls.
228             _centery, _blockWithCyls._centerz
229             , _blockWithCyls._size1, _blockWithCyls._size2, _blockWithCyls._size3);
230         double[] poyntVect = _fdtdAllLib.Poynting_totalVect();
231         yyPlotEnergy = new double[] { t, _multEnergy * _fdtdAllLib.total_E_energy() / (t == 0 ? 1 : 1) //with
232             Drude??
233             , poyntVect[0], poyntVect[1], poyntVect[2], 1e-0 * drudeEner };
234         dataPlotEnergy[it, 0] = t;
235         dataPlotEnergy[it, 1] = yyPlotEnergy[1];
236         dataPlotEnergy[it, 2] = yyPlotEnergy[2]; //poyntVect[0]
237         dataPlotEnergy[it, 3] = yyPlotEnergy[3]; //poyntVect[1]
238         dataPlotEnergy[it, 4] = yyPlotEnergy[4]; //poyntVect[2]
239         dataPlotEnergy[it, 5] = yyPlotEnergy[5]; //DrudeEnergy
240     } //<<<
241     else
242     {
243         ErrorMessage("Wrong isVxVelocity");
244     }
245     _realTimePlot.bgnRealAdd(true, t, yyPlot, "start,-3/4,-1/4,center,1/4,fin,", "t");
246     _realTimePlot.set_Header("Ex: " + _fdtdAllLib.ToStringCher());
247
248     _realTimeEnergy.bgnRealAdd(true, t, yyPlotEnergy, "Et,Pt0,Pt1,Pt2,Dru,", "t");
249     _realTimeEnergy.set_Header("Ex: " + _fdtdAllLib.ToStringCher());
250     //////////////////////////////////////
251
252     bool isTouch = this._stopAtTouch == true;

```

```

253 if (isTouch == true && Math.Abs(fieldsInBoundsXFin[0]) > minBoundVal)
254 {
255 Console.WriteLine("\n###Break calc.: isTouch={0}, FieldsInBounds[0]={1:0:0.000}", isTouch,
        fieldsInBoundsXFin[0]);
256 Console.WriteLine("FieldsInBounds[1]={0:0:0.000}, FieldsInBounds[2]={1:0:0.000}", fieldsInBoundsXFin
        [1], fieldsInBoundsXFin[2]);
257 break;
258 }
259 #endregion
260 #region show 3D
261 if (it dtStepShow == 0 // show plot 3D
262 {
263 int i4 = 0, j4 = 0, k4 = 0;
264 if (isVxVelocity == true)
265 {
266 zz = zCentrPlaneStart; // center (0,0,0)
267 for (xx = -_fddtdAllLib.get_xsize() / 2 + tmpPml * _fddtdAllLib._pmlil; xx < _fddtdAllLib.get_xsize() /
        2 - tmpPml * _fddtdAllLib._pmlir; xx += hx)
268 for (double yy = -_fddtdAllLib.get_yysize() / 2 + tmpPml * _fddtdAllLib._pmljl; yy < _fddtdAllLib.
        get_yysize() / 2 - tmpPml * _fddtdAllLib._pmljr; yy += hy)
269 {
270 try
271 {
272 Efield = _fddtdAllLib.GetFieldVectorInGrid(xx, yy, zz, out i4, out j4, out k4); //X
273 }
274 catch (Exception e)
275 {
276 if (_progrBar != null) _progrBar.Close();
277 MessageBox.Show("3" + e.Message);
278 }
279 }
280 }
281 if (i4 >= 0 && i4 < dataPlot3D.GetLength(0))
282 {
283 if (k4 >= 0 && k4 < dataPlot3D.GetLength(0))
284 {
285 if (j4 >= 0 && j4 < dataPlot3D.GetLength(1))
286 {
287 try
288 {
289 dataPlot3D[i4, j4] = epsLoc * Efield[indField];
290 }
291 catch (Exception e)
292 {
293 string s = "i4=" + i4 + ", j4=" + j4 + ", indField=" + indField + ", Efield.length" + Efield.Length;
294 MessageBox.Show(s + "\n" + e.Message);
295 }
296 }
297 }
298 if (SubtractInitCond == true)
299 if (it == 0)
300 dataPlot3D_0[i4, j4] = dataPlot3D[i4, j4]; // save init state
301 else
302 dataPlot3D[i4, j4] -= dataPlot3D_0[i4, j4]; // subtract init state
303 }
304 }
305 }
306 }
307 }
308 else //isVxVelocity == false
309 {
310 xx = xCentrPlaneStart; // center (0,0,0)
311 for (double yy = -_fddtdAllLib.get_yysize() / 2; yy < _fddtdAllLib.get_yysize() / 2; yy += hy)
312 for (zz = -_fddtdAllLib.get_zsize() / 2; zz < _fddtdAllLib.get_zsize() / 2; zz += hz)
313 {
314 double Efield5 = _fddtdAllLib.GetFieldInGrid(this._FieldToPlot, xx, yy, zz, out i4, out j4, out k4);
315 if (i4 >= 0 && i4 < dataPlot3D.GetLength(0))
316 {
317 if (k4 >= 0 && k4 < dataPlot3D.GetLength(1))
318 {
319 if (j4 >= 0 && j4 < dataPlot3D.GetLength(0))
320 dataPlot3D[j4, k4] = Efield5; // Ez
321 }
322 }
323 }
324 }
325 }
326 if (this._isStopNow == true)
327 {
328 if (_progrBar != null) _progrBar.Close();
329 Console.WriteLine("\n\tLoop-t finishes at it={0}, t={1:0:00}.\n", it, t);
330 //goto Fin;
331 return;
332 }
333 }
334 double tmpPml1 = 0;
335 if (hasShowAll3D == true)
336 { // >>> 3
337 int n10 = 10 * 0;
338 for (xx = -_fddtdAllLib.get_xsize() / 2 + n10; xx < _fddtdAllLib.get_xsize() / 2 - n10; xx += hx)
339 for (double yy = -_fddtdAllLib.get_yysize() / 2 + n10; yy < _fddtdAllLib.get_yysize() / 2 - n10; yy += hy)
340 for (zz = -_fddtdAllLib.get_zsize() / 2 + n10; zz < _fddtdAllLib.get_zsize() / 2 - n10; zz += hz)
341 {
342 try
343 {
344 Efield = _fddtdAllLib.GetFieldVectorInGrid(xx, yy, zz, out i4, out j4, out k4); //X
345 }
346 catch (Exception e)
347 {
348 if (_progrBar != null) _progrBar.Close();

```

```

349 MessageBox.Show("3" + e.Message);
350 }
351
352 if (i4 >= 0 && i4 < dataPlotAll3D.GetLength(0))
353 {
354     if (j4 >= 0 && j4 < dataPlotAll3D.GetLength(1))
355     {
356         if (k4 >= 0 && k4 < dataPlotAll3D.GetLength(2))
357         {
358             if (hasShowAll3D == true)
359                 dataPlotAll3D[i4, j4, k4] = 1 * Efield[0];
360         }
361     }
362 }
363 }
364 }<<<3
365
366 if (_deletePred3dPlotCheckBox == true)
367     get_mainWinObj().CloseExtra3DPlots();
368
369 get_mainWinObj().CreateNew3dPlot3(dataPlot3D, "t=" + t
370 , _fddtdAllLib.ToStringCher() + ".xC=" + xCentrPlaneStart + ".zC=" + zCentrPlaneStart + ".v0x=" + v0X
371 + ".v0y=" + v0Y + ".v0z=" + v0Z + ".plot=" + this._FieldToPlot
372 , _fddtdAllLib.ToStringCher3(), "legend", 30, 120, 400, 500, get_mainWinObj()._winList_surfPlot, true,
    false, shft);
373 Application.DoEvents();
374
375 }//if (it 5 == 0
376 #endregion
377 if (_isStopNow == true)
378 {
379     Console.WriteLine("\n\tLoop-t finishes at it={0}, t={1:0.00}.\n", it, t);
380     return;
381 }
382 cnt++;
383 }//for it
384 }//for (it = 0;
385 Console.WriteLine("\n\tLoop-t finishes at it={0}, t={1:0.00}.\n", it, t);
386 #endregion
387 }// CircleTime()
388
389 // - - - - - calcAllField - - - - -
390 void calcAllField(int indField, ref FddtdAllLib _fddtdAllLib, double hx, double hy, double hz, int
    sizePml, string sFieldToPlot, ref double[,] dtPlot3D)
391 {
392     double xx, yy, zz;
393     int i4 = -1, j4 = -1, k4 = -1;
394     double[] Efield = null;
395
396     for (xx = -_fddtdAllLib.get_xsize() / 2 + sizePml; xx < _fddtdAllLib.get_xsize() / 2 - sizePml; xx +=
        hx)
397     for (yy = -_fddtdAllLib.get_ysize() / 2 + sizePml; yy < _fddtdAllLib.get_ysize() / 2 - sizePml; yy +=
        hy)
398     for (zz = -_fddtdAllLib.get_zsize() / 2; zz < _fddtdAllLib.get_zsize() / 2; zz += hz)
399     {
400         try
401         {
402             Efield = _fddtdAllLib.GetFieldVectorInGrid(xx, yy, zz, out i4, out j4, out k4);//X
403         }
404         catch (Exception e)
405         {
406             MessageBox.Show("3-calcAllField: " + e.Message);
407         }
408     }
409
410     if (i4 >= 0 && i4 < dtPlot3D.GetLength(0))
411     {
412         if (j4 >= 0 && j4 < dtPlot3D.GetLength(1))
413         {
414             if (k4 >= 0 && k4 < dtPlot3D.GetLength(2))
415             {
416                 dtPlot3D[i4, j4, k4] = Efield[indField];
417             }
418         }
419     }
420 }
421 #region aux
422 // - - - - - WhichObjects - - - - -
423 void WhichObjects(out int nRods, out int nBlocks)
424 {
425     nRods = 0; nBlocks = 0;
426     for (int i = 0; i < _fddtdAllLib._objectList.Length; ++i)
427     {
428         obj s = _fddtdAllLib._objectList[i];
429         if (s.shape != null)
430         {
431             if (s.shape.Contains("rod")) ++nRods;
432             if (s.shape.Contains("block")) ++nBlocks;
433         }
434     }
435 }
436 // - - - - - WhichMetalObjects - - - - -
437 void WhichMetalObjects(out int nRods, out int nBlocks)
438 {
439     nRods = 0; nBlocks = 0;
440     for (int i = 0; i < _fddtdAllLib._mObjectList.Length; ++i)
441     {
442         mobj s = _fddtdAllLib._mObjectList[i];
443         if (s.shape != null)
444         {
445             if (s.shape.Contains("rod")) ++nRods;

```

```

446 if (s.shape.Contains("block")) ++nBlocks;
447 }
448 }
449 }
450 // - - - - - BoolToStr - - - - -
451 string BoolToStr(bool b)
452 {
453     string ss = b == true ? "T" : "F";
454     return ss;
455 }
456 // - - - - - TransposeRowsAndColumns - - - - -
457 double[][] TransposeRowsAndColumns(double[][] arr)
458 {
459     int rowCount = arr.Length;
460     int columnCount = arr[0].Length;
461     double[][] transposed = new double[columnCount][];
462     if (rowCount == columnCount)
463     {
464         transposed = (double[][]) arr.Clone();
465         for (int i = 1; i < rowCount; i++)
466         {
467             for (int j = 0; j < i; j++)
468             {
469                 double temp = transposed[i][j];
470                 transposed[i][j] = transposed[j][i];
471                 transposed[j][i] = temp;
472             }
473         }
474     }
475     else
476     {
477         for (int column = 0; column < columnCount; column++)
478         {
479             transposed[column] = new double[rowCount];
480             for (int row = 0; row < rowCount; row++)
481             {
482                 transposed[column][row] = arr[row][column];
483             }
484         }
485     }
486     return transposed;
487 }
488 // - - - - - RunTimeMessageBox - - - - -
489 string RunTimeMessageBox(string myText, string caption, int sec)
490 {
491     MessageBoxEx msgBox = MessageBoxExManager.CreateMessageBox(null);
492     msgBox.Caption = caption;
493     msgBox.Text = myText;
494     msgBox.AddButtons(MessageBoxButtons.YesNo);
495     msgBox.Icon = MessageBoxExIcon.Question;
496     //Wait for 10 seconds for the user to respond
497     msgBox.Timeout = sec * 1000;
498     msgBox.TimeoutResult = TimeoutResult.Timeout;
499     string result = msgBox.Show();
500     return result;
501 } //RunTimeMessageBox
502 // - - - - - To2D - - - - -
503 T[,] To2D<T>(T[][] source)
504 {
505     try
506     {
507         int FirstDim = source.Length;
508         int SecondDim = source.GroupBy(row => row.Length).Single().Key; //
509         var result = new T[FirstDim, SecondDim];
510         for (int i = 0; i < FirstDim; ++i)
511             for (int j = 0; j < SecondDim; ++j)
512                 result[i, j] = source[i][j];
513     }
514     return result;
515 }
516 catch (InvalidOperationException)
517 {
518     throw new InvalidOperationException("The given jagged array is not rectangular.");
519 }
520 } //To2D
521
522 #endregion
523
524 }}

```

GENERALIDADES: RADIACIÓN DE CHERENKOV Y UN ARREGLO DE ESTRUCTURAS CNT RESONANTES.

C.1. Introducción

En este apéndice abordaremos generalidades acerca del capítulo 4, en el cual se muestran algunos resultados que se producen entre la interacción de una carga (Radiación de Cherenkov) y un arreglo de estructuras CNT resonantes utilizando el método FDTD, que provee los recursos para la modelación de estructuras geométricas y calculos de resultados en el dominio del tiempo (Ver Capitulo 3) para nuestro aprovechamiento numérico.

C.2. Modelación de cuerpos Block+Rod

En las imágenes de la figura C.1, observamos las principales estructuras utilizadas en nuestra simulación numérica. En la imagen (a) de la figura C.1, se muestra el bloque (block), que delimita la base donde estarán dispersos los CNTs, también nos indica los límites de la frontera sobre la cual se va a simular nuestro fenómeno y donde la carga se moverá desde un tiempo inicial hasta un tiempo final definido por el usuario (Generalmente la simulación se detiene hasta que la carga atraviesa toda el bloque o malla numérica o hasta tocar el límite de la rejilla). En la imagen (b) de la figura C.1, observamos un grupo de nanoestructuras periódicas cilíndricas (rod) y alargadas (el cual haremos resonar al aplicarle una frecuencia de plasma). En la imagen (c) de la figura C.1 observamos la combinación de los cuerpos de las imágenes (a)+(b), que aplicaremos en nuestra simulación (Cuerpo final que se expondrá a la radiación de Cherenkov). La idea es simular un sistema en donde se efectuó la radiación de Cherenkov, acelerando una partícula que pase a través del sistema de los nanotubos metálicos generando un campo electromagnético que produce energía en forma de plasmones de superficie. Ver figura C.2.

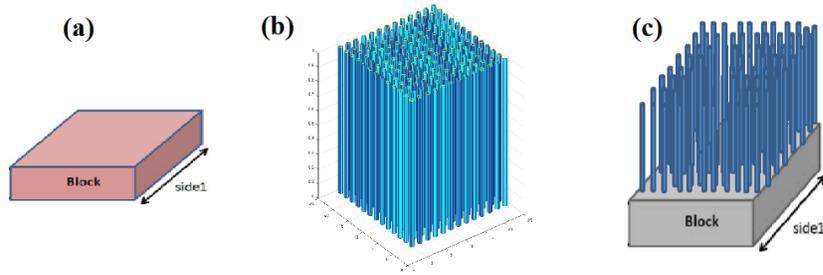


Figura C.1: Cuerpos (a) Blocks que sirven para la base esquemática de nuestra simulación, (b) cilindros (rods) para representar CNTs y (c), la combinación de a+b.

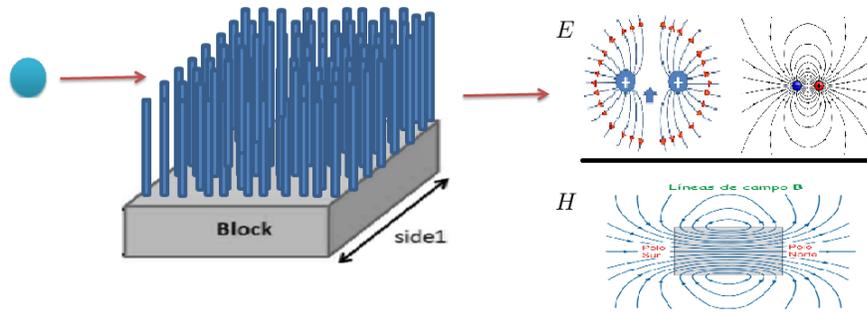


Figura C.2: Partícula que atraviesa un arreglo de nanotubos y se muestran los campos eléctricos (E) y magnéticos (H) estáticos (En un momento en el tiempo) con el paso de la partícula

C.3. Fenómeno de la radiación de Cherenkov

El físico ruso Pável Cherenkov fue el primero en caracterizarla rigurosamente y explicar su producción. Pavel Cherenkov recibió el Premio Nobel de Física en 1958. La radiación de Cherenkov (también escrito Cerenkov) es una radiación de tipo electromagnético producida por el paso de partículas cargadas eléctricamente en un determinado medio a velocidades superiores a las de la luz en ese medio. La velocidad de la luz depende del medio, y alcanza su valor máximo en el vacío. El valor de la velocidad de la luz en el vacío no puede superarse, pero sí en un medio en el que está es forzosamente inferior. Donde la velocidad no puede ser mayor a la velocidad de la luz: $V_o > \frac{c}{n}$, Donde: c =Velocidad de la luz de 300 000 km/seg. y,

$$n = \sqrt{\epsilon_r \mu_r}, \quad n > 1 \quad (C.1)$$

Las nanoestructuras pueden ser de metales (TiO_2) o dieléctricos(CNT), consideramos un arreglo de nanotubos conformados por 1000 objetos para el caso periódico y aleatorio. Y una rango de frecuencia de prueba de $\omega_p = 1 \times 10^1 \dots\dots\dots 1 \times 10^{19}$ con una pérdida de $\gamma = 1 \times 10^1 \dots\dots\dots 1 \times 10^{19}$ Con una permitividad en el medio

para metales Drude (ϵ):

$$\epsilon = \epsilon_b - \frac{\omega_p^2}{(\omega - i\gamma)\omega} \quad (\text{C.2})$$

Y donde: ω_p es la frecuencia del plasma, ω es la frecuencia del campo electromagnético en sec^{-1} , γ es la pérdida de energía en sec^{-1} , donde:

$$\omega_p^2 = \frac{e^2 N}{\epsilon_0 m_e} \quad (\text{C.3})$$

Con valores: N =Densidad de los electrones en un metal. m_e =Masa de los electrones.

Nuestro objetivo es estudiar una malla finita, pero en la realidad la propagación de los campos excede este límite y en algunos casos pueden ser infinitos, para ello utilizamos la actuación de PML (Perfectly Matched Layers) que evita reflexiones de las propagaciones del campo eléctrico y magnético.

-PML(Evita reflexiones en la malla) en la malla con dimensiones; -Size X,Y,Z, es conocido también como (Boundary conditions). Posicionamos la carga en una dimensión inicial, Start hasta una posición final, Fin. Donde x,y,z es un valor asignado libremente por el usuario para las pruebas numéricas, Posición inicial de la partícula en el plano de Yee.

-xStart -xFin
-yStart -yFin
-zStart -zFin

v_x =Velocidad de la partícula que atraviesa la superficie de los nanotubos

En la figura C.3, observamos la simulación que se realizó en donde se muestra la alteración del campo eléctrico sobre la superficie de los nanotubos, podemos ver la dirección de donde se mueve la carga en una posición $x_{\text{Start}} = y_{\text{Start}} = z_{\text{Start}} = 0$ hasta llegar a una posición final, donde la partícula toca los límites $x_{\text{Fin}} = y_{\text{Fin}} = z_{\text{Fin}} = \text{PML}(\text{final } PML_y)$. Para este ejemplo estudiamos el campo eléctrico que se produce en E_x con picos orientados a la dirección de la superficie (Altura Z) hacia donde apuntan los nanotubos y con una trayectoria de carga que entra de izquierda a derecha. El resultado de esta ejecución numérica se muestra en la imagen C.3, pero también se pueden registrar los campos generados en E_y , E_z , H_x , H_y y H_z (donde E es el campo eléctrico y H el campo magnético), como se muestra en la figura C.4(a-c) y C.5(a-c). En la figura C.4 se muestran los campos eléctricos para las propagaciones E_x (a), E_y (b) y E_z (c) mientras que en las imágenes de la figura C.5 se muestran las propagaciones del campo magnético en las direcciones de propagación H_x (a), H_y (b) y H_z (c).

En primera instancia tenemos el caso de estudio periódico para poder comprender mejor el comportamiento del fenómeno de estudio del campo eléctrico E_x , ya

Figura C.3: Partícula que atraviesa un arreglo de nanotubos. El campo eléctrico o picos de la imagen, son el resultado del efecto que causa la partícula que deforma los campos eléctricos de los nanotubos (caso periódico), que se encuentran por debajo de la superficie por donde pasa la partícula este efecto es llamado Radiación de Cherenkov.

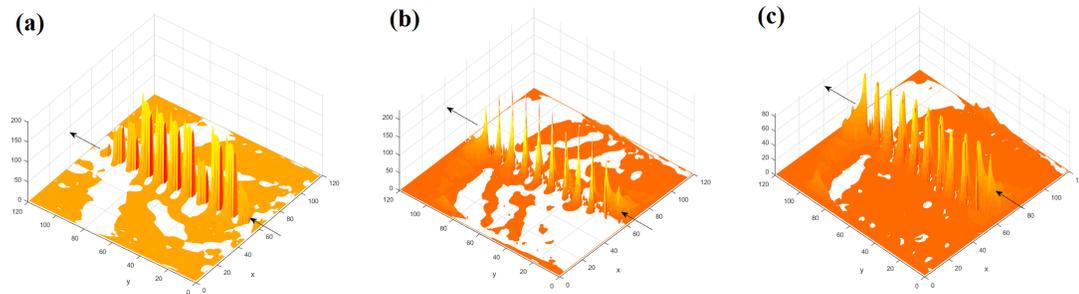


Figura C.4: Se muestran los campos eléctricos para la propagación vectorial (a) E_x , (b) E_y , (c) E_z , sobre la superficie nano-tubular.

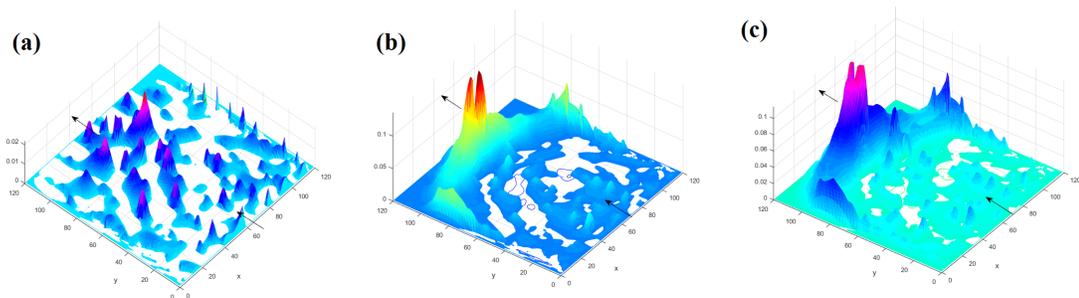


Figura C.5: Se muestran los campos magnéticos para la propagación vectorial (a) H_x , (b) H_y , (c) H_z , sobre la superficie nano-tubular.

que la partícula se mueve por encima de la nanoestructura tubular. Se puede observar en la figura C.6 la ruta que sigue la partícula y el campo eléctrico de los nanotubos que se ve alterado mediante su paso. En la figura C.7, observamos la simetría del campo 2D que se produce los nanotubos por arriba y por debajo de la superficie nano-tubular, observamos picos máximo y mínimos los cuales nos sirven de referencia para obtener el total de energía radiada que produce la carga entre esta interacción con los nanotubos. Al sumar los picos máximos que produce la carga durante toda su trayectoria obtendremos un punto total de energía máxima para una determinada frecuencia ω_p . Se observa en el gráfico que para cada pico

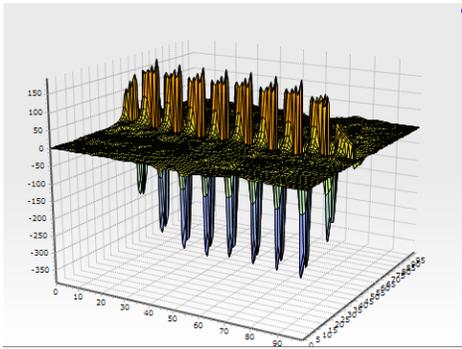


Figura C.6: Ruta del campo eléctrico generado por una carga radiante, observamos picos de campos generados que corresponden a los nanotubos que hay debajo de la superficie. La orientación de la carga es de izquierda a derecha.

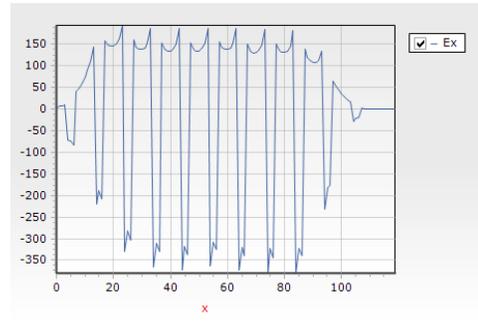


Figura C.7: Geometría 2D de la ruta de la carga, de la figura C.6. Al sumar los picos máximos que produce la carga durante toda su trayectoria obtendremos un punto total de energía máxima para una determinada frecuencia.

calculado por las transformadas de Fourier de la figura C.8 con relación a la frecuencia (f), los puntos que se grafican en la gráfica de la figura C.9 de un rejilla fina (Fine Grid), son más cercanos, es por ello que en las transformadas de Fourier aparecen más cerca una de la otra, mientras que en la figura C.9, dichas cercanías están dadas por la altura de los puntos en el par ordenado, ya que la distancia entre cada punto es más corta del vector de onda. Existe una onda espacial, una onda llamada plasmón, que se genera a partir de la radiación de Cherenkov.

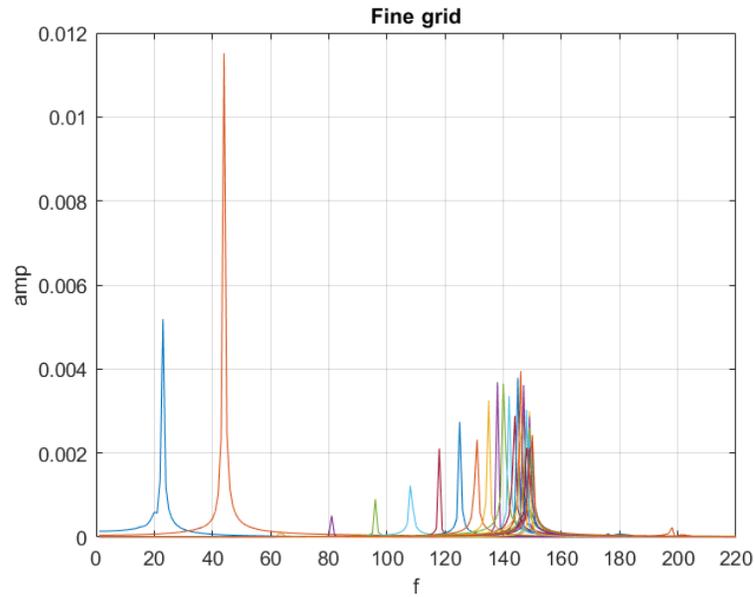


Figura C.8: Transformada de Fourier para el comportamiento de la ruta de energía que se muestra en la figura C.7, en donde se muestra la amplitud amp por frecuencia f de onda registrada..

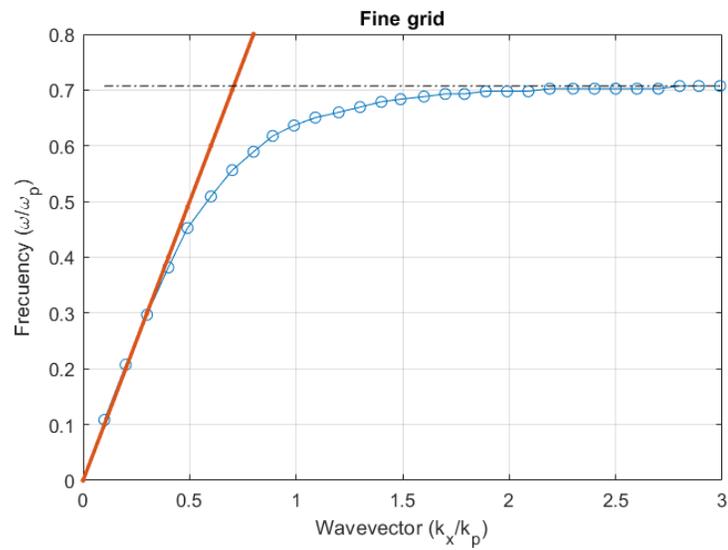


Figura C.9: Se muestra el vector de longitud $Wavevector\ k_x/K_p$ vs la frecuencia, respecto a la frecuencia de plasma $Frecuencia\ \omega/\omega_p$ que corresponden a la amplitud de la figura C.8, donde la línea naranja corresponde a la frecuencia guiada por la luz.

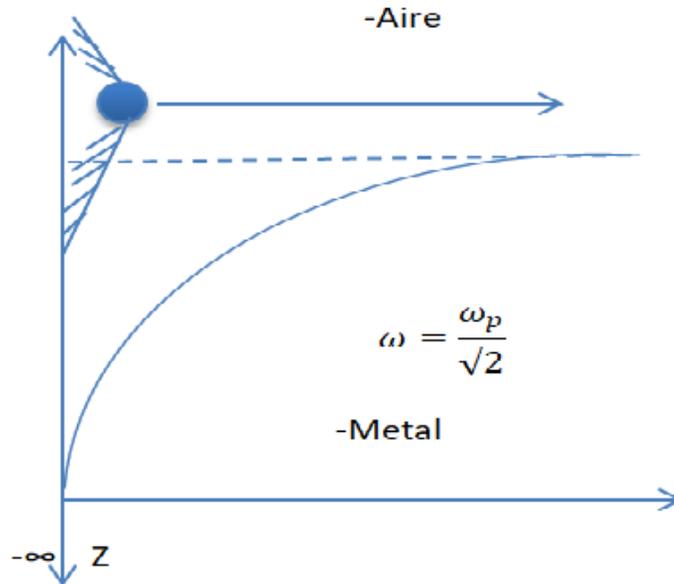


Figura C.10: Gráfica de una partícula sobre un metal, entre más velocidad alcance la partícula, mas plasmon se generara.

C.4. Plasmón de superficie

Se forma cuando el metal es tratado como un cristal tridimensional de iones positivos, junto a un gas de electrones deslocalizado que se mueve en esta red de iones que forman un potencial periódico.

Los plasmones de superficie son aquellos plasmones que están confinados a las superficies y que forman un polaritón cuando interactúan con la luz. Ocurren en la interfaz entre un dieléctrico y un metal. Permiten explicar las anomalías en la difracción de una red de difracción metálica.

Los plasmones son considerados como medios de transmisión de información en microprocesadores y chips de computadoras ya que pueden alcanzar altas frecuencias (de hasta $100 THz$, mientras que los cables convencionales alcanzan las decenas de GHz).

Los plasmones involucran movimientos rápidos de los electrones a través del sólido, donde la pérdida óhmica desaparece. Para que la electrónica basada en plasmones sea útil debe inventarse el plasmonster, el análogo del transistor.

Considerando metales Drude; metales que producen un campo electromagnético, podemos observar en la figura C.10, una partícula sobre un metal de altura Z , oscilando a una frecuencia de $\omega = \frac{\omega_p}{\sqrt{2}}$ que puede aumentar. Observamos a la partícula (con forma cónica, típica de la radiación de Cherenkov) con trayectoria paralela a la superficie del metal, que viaja por el aire. Podemos observar que la radiación que emite la partícula puede afectar en el interior del metal.

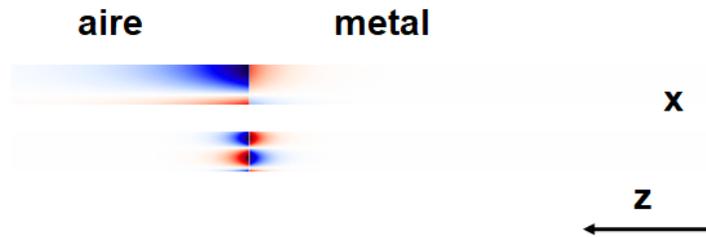


Figura C.11: Para el valor del plasmon siempre será menor en el metal, que en el aire.



Figura C.12: Figura que muestra, la simulación, la calculación y resultados que se pretenden realizar, se muestran "n"partículas sobre una superficie metálica S .

En la figura C.11 observamos el nivel de energía radiada de plasmones de superficie en el metal y sobre el aire. Observamos una mayor dispersión de plasmones que se suspenden en el aire que los que están confinados dentro del metal. La idea es generar varias simulaciones de partículas que actúan a la vez, ya que el plasmon que genera cada partícula independiente es muy pequeña, se necesitan por lo menos 100 partículas radiando para tener un valor significativo, es por ello que se implementó una programación en paralelo para probar el efecto que producen las partículas sobre los nanotubos o nano-compuestos cilíndricos, en la figura C.12, se observa esta implementación que se pretende simular. Esta simulación puede tener aplicaciones en la nano-tecnología como son en las aplicaciones de Smartphon, donde los transistores son bastante pequeños.

C.5. Cálculo en paralelo.

Para la programación en paralelo se realizaron "n"hilos que son definidos desde la interfaz por el usuario, y se gráfica el valor de la energía para la radiación de Cherenkov sobre los nano-compuestos cilíndricos, para nano compuestos periódicos y aleatorios con un rango de velocidad. Los parámetros de entrada para la calculación fueron los siguientes: $Metal = true, Cylinderinplane = Periodic, nObject = 1000, \omega_p = 1e16, \gamma = 1E15, Parall = true, Symetric = true, percolation\ in\ examples = false, vxMin = 1,3, vMax = 1,75, Threads = 20$. El tiempo de la calculación fue aproximadamente de 27-30 minutos en una laptop

Dell intel (R) core(TM) i7-6700HQ, a 2.60 GHZ.

En la figura C.13 observamos las barras de proceso en donde se invocaron 20 hilos y se asignó una tarea (task) para cada una. Creamos la clase ParallChrenkov que nos permitió crear una lista de objetos a partir de la clase Cherenkov que es utilizada para la radiación de una partícula emitida a una determinada frecuencia (Ver figura C.7) para obtener la energía total que emite la carga, una vez terminado este proceso los valores se van agregando a las siguientes graficas de resultados.

De los resultados de la simulación anterior de varias partículas, se generó una bajada de energía como se muestra en la figura C.14. Variando desde una velocidad mínima de $v_{min} = 0,81$ a $v_{max} = 0,98$ (velocidades normalizadas a la velocidad de la luz c) con una frecuencia $\omega_p = 1 \times 10^{16}$ y con una perdida gamma $\gamma = 1 \times 10^{15}$ para nanotubos periódicos en la rejilla. Teniendo como principal ley

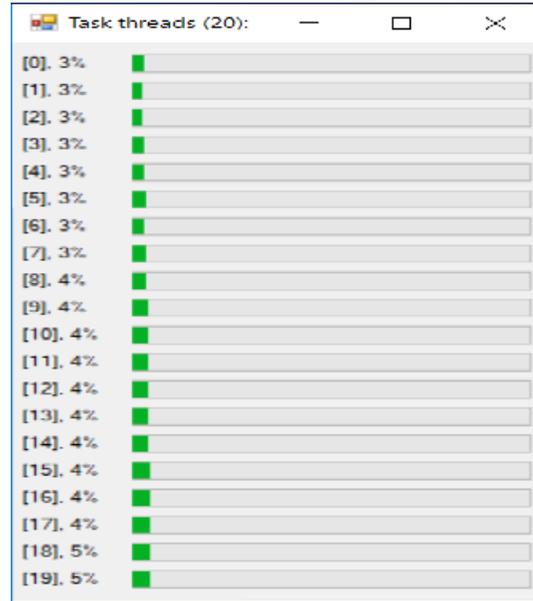


Figura C.13: Se muestran los procesos ejecutándose de manera paralela para 20 tareas simultáneamente y el porcentaje de avance para cada proceso

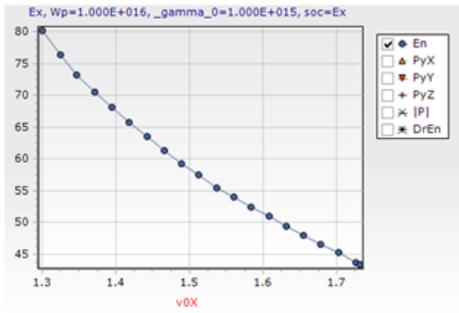


Figura C.14: Gráfica que muestra el valor de la calculación para la energía En .

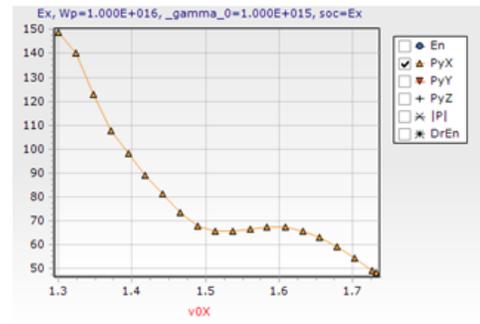


Figura C.15: Gráfica que muestra el valor de la calculación para el componente P_x

de la conservación de la energía en las ondas electromagnéticas, utilizamos el teorema de Poynting; Que establece que la disminución de energía electromagnética en una región se debe a la disipación de potencia en forma de calor (por efecto Joule) y al flujo hacia el exterior del vector de Poynting \vec{P} . Para el caso del flujo de poynting, en su componente: P_x , decae como se muestra en la figura C.15,

teniendo que:

$$\vec{P} = \sqrt{P_x^2 + P_y^2 + P_z^2} \quad (\text{C.4})$$

Para los vectores \vec{P} de flujo poynting en todas sus direcciones. Para el caso de P_y y P_z , los valores calculados en sus diferentes velocidades, el flujo tiende a subir, es importante mencionar que las frecuencias para $\omega_p = 1e15$ en unidades sec^{-1} , para todas las velocidades ($v_{min} = 0,81$ a $v_{max} = 0,98$).

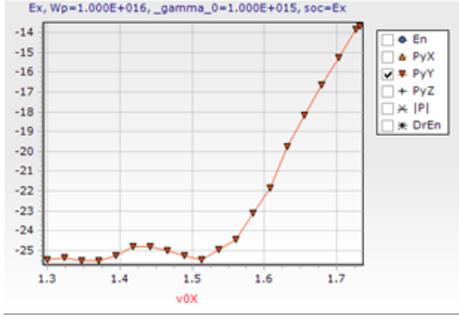


Figura C.16: Gráfica que muestra el valor de la calculación para el componente del vector de flujo P_y .

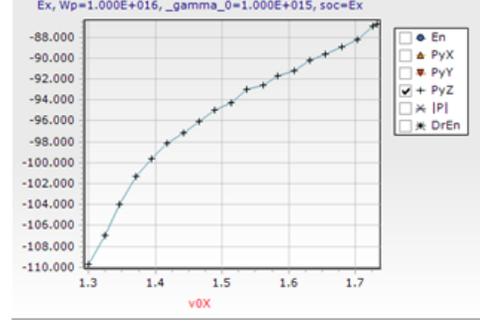


Figura C.17: Gráfica que muestra el valor de la calculación para el componente de vector de flujo P_z .

C.5.1. Gráfica Energía $v_{min}-v_{max}$ vs ω_p

Se realizaron varias calculaciones para determinadas frecuencias de ω_p , para realizar una comparación de los resultados obtenidos para cada valor de ω_p . A continuación, se muestran en las figuras C.18, C.19, C.20 y C.21 los resultados calculados y graficados en Matlab, en un rango de velocidades 1.3 a 1.75 (correspondientes a $v_{min} = 0,81$ a $v_{max} = 0,98$), y para frecuencias de: $\omega_p = 1e12$, $\omega_p = 1e13$, $\omega_p = 1e14$, $\omega_p = 1e15$, $\omega_p = 1e16$.

En las gráficas C.18-C.19 apreciamos la relación de energía para las frecuencias antes mencionadas, observamos un incremento considerable para las dos últimas frecuencias: $\omega_p = 1e15$ y $\omega_p = 1e16$. De la misma manera graficamos el vector del flujo Poynting en su componente P_x para las misma velocidades de las figuras C.18-C.19 y el mismo rango de velocidad para **nanoestructuras periódicas**. Ver figura C.20 y grafica C.21 con vista del comportamiento 3D. Observamos un decaimiento de flujo de energía para todas las frecuencias consideradas.

Los resultados de las gráficas de las figuras C.18, C.19, C.20 y C.21 muestran los resultados en el caso de nanoestructuras periódicas, pero en nuestro aprovechamiento numérico también hemos considerado nanoestructuras con morfología aleatoria.

Los resultados de dicha interacción entre nanoestructuras aleatorias y la carga para la propagación de energía y flujo de Poynting se muestran en las gráficas de

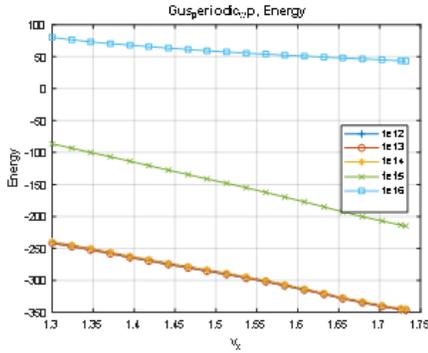


Figura C.18: Gráfico que muestra los resultados para las frecuencias $\omega_p = 1e12$, $\omega_p = 1e13$, $\omega_p = 1e14$, $\omega_p = 1e15$ y $\omega_p = 1e16$, en un rango de velocidades que van desde $v_{min} = 1,3$ hasta $v_{max} = 1,7$ ($v_{min} = 0,81$ a $v_{max} = 0,98$), para el cálculo de la energía (Energy).

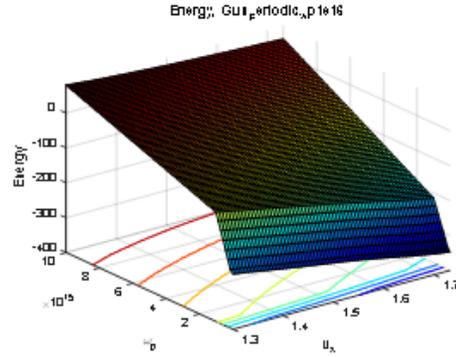


Figura C.19: Gráfico que muestra los resultados para las frecuencias $\omega_p = 1e12$, $\omega_p = 1e13$, $\omega_p = 1e14$, $\omega_p = 1e15$ y $\omega_p = 1e16$, en un rango de velocidades que van desde $v_{min} = 1,3$ hasta $v_{max} = 1,7$ ($v_{min} = 0,81$ a $v_{max} = 0,98$), para el cálculo de la energía (Energy) en vista 3D.

las figuras C.22, C.23, C.24 y C.25 respectivamente.

Con respecto a las pruebas realizadas se observó que para nanotubos con morfología periódica y aleatoria, para un rango de velocidades que van desde $v_{min} = 1,3$ hasta $v_{max} = 1,7$ ($v_{min} = 0,81$ a $v_{max} = 0,98$), teniendo como velocidad permitida máxima $3 \times 10^{08} \text{ km/seg}$ para las frecuencias: $\omega_p = 1e12$, $\omega_p = 1e13$, $\omega_p = 1e14$, $\omega_p = 1e15$, $\omega_p = 1e16$, $\omega_p = 1e17$ y $\omega_p = 1e18$, hubo un incremento muy notorio de propagación de energía a partir de la frecuencia: $\omega_p = 1e15$, como se aprecian en las figuras C.18 y C.22, mientras que para frecuencias bajas ($3 \times 10^{08} \text{ km/seg}$ para las frecuencias: $\omega_p = 1e12$, $\omega_p = 1e13$, $\omega_p = 1e14$) los niveles de energía permanecieron casi constantes para ambos tipos de nanoestructuras (Periódico y aleatorio).

Para el flujo de Poynting se registró un decremento esperado al teórico para ambos casos de morfología en las nanoestructuras (Periódico y aleatorio). Estos resultados nos motivan a estudiar un rango de frecuencias para varias velocidades de cargas que se abordan en el capítulo 4.

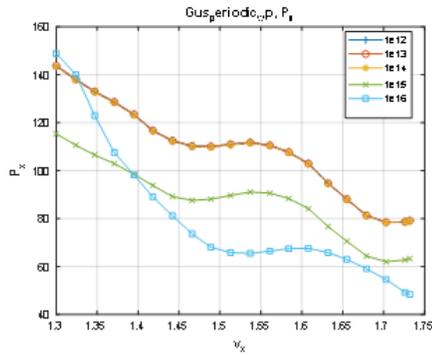


Figura C.20: Gráfico del vector de flujo Poynting en su propagación P_x , para los mismo valores que la gráfica C.18

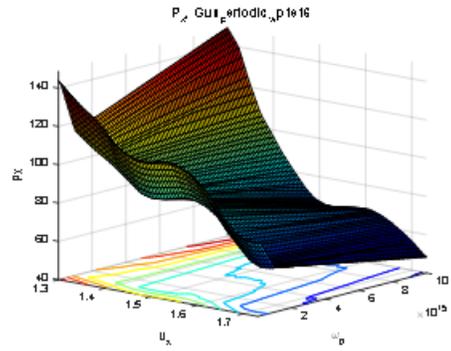


Figura C.21: Gráfico del vector de flujo Poynting en su propagación P_x , para los mismo valores que la gráfica C.19 en su vista 3D.

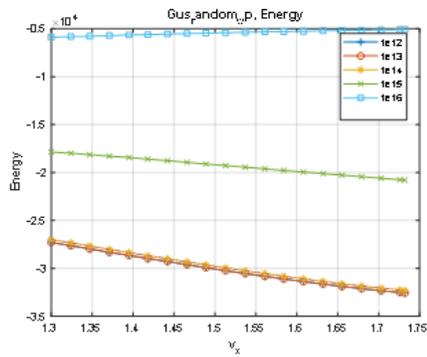


Figura C.22: Gráfico de la energía para los mismo valores que la gráfica C.18 ω_p pero para nanoestructuras con morfología aleatoria.

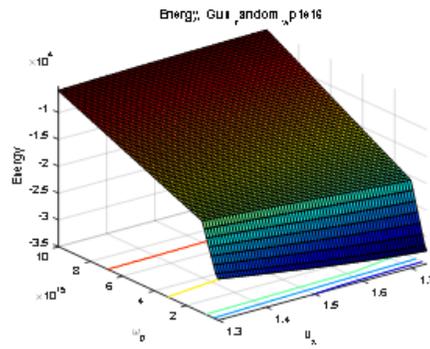


Figura C.23: Gráfico de la energía para los mismo valores que la gráfica C.19 ω_p pero para nanoestructuras con morfología aleatoria con vista 3D.

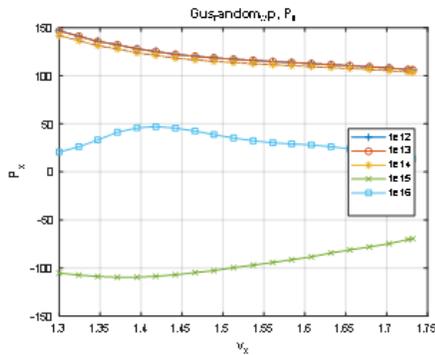


Figura C.24: Gráfico del flujo Poynting para los mismo valores que la gráfica C.18 ω_p pero para nanoestructuras con morfología aleatoria.

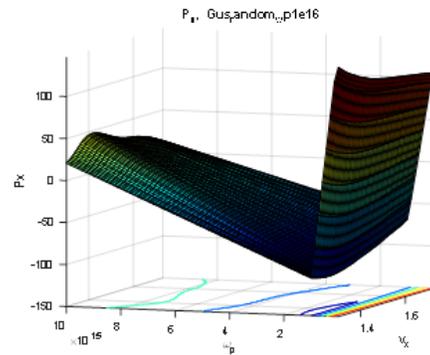


Figura C.25: Gráfico del flujo Poynting para los mismo valores que la gráfica C.19 ω_p pero para nanoestructuras con morfología aleatoria con vista 3D.

PROCEDIMIENTO EXPERIMENTAL PARA NANOTUBOS DE DIÓXIDO DE TITANIO TiO_2

D.1. Síntesis de las películas nano-tubulares de TiO_2

Para la síntesis de las películas nano-tubulares de TiO_2 , se usó una celda clásica de politetrafluoroetileno (Teflón) cilíndrica con dimensiones: $40,5mm$ de diámetro interior, $80,5mm$ diámetro exterior, $49mm$ de profundidad interior y $69,7mm$ de altura externa. La celda electroquímica estaba constituida por dos electrodos de trabajo, el electrodo (ánodo) formado por una lámina de $2,0cm^2$ de Ti (Titanio) de alta pureza (99,7 %, 0,127 espesores Sigma-Aldrich) que se usó como sustrato mientras que el electrodo (cátodo) estaba constituido por una malla de platino. El sustrato fue pulido en espejo. Después del pulido, la lámina Ti se desengrasó en un baño ultrasónico en isopropanol y acetona durante 15 minutos, seguido de un enjuague con agua des-ionizada y secada con una corriente de nitrógeno.

Todos los experimentos de anodización se llevaron a cabo a temperatura ambiente ($25^\circ C$) en condiciones aireadas sin agitación. Para hacer crecer películas nano-tubulares auto organizadas, se expusieron láminas de Ti en $4ml$ de (50 : 50Vol. %) H_2O : glicerol + $0,27MNH_4F$ a un voltaje constante de $20V$ por $3,5h$ cite Sintillo: 2017. Una vez finalizada la anodización, las muestras se enjuagaron con agua desionizada y se secaron con gas nitrógeno. Luego se realizó la caracterización física a partir de imágenes FE-SEM determinando la morfología. Los nanotubos abiertos de la vista superior y su vista en sección transversal de una matriz clásica de nano-tubular TiO_2 se obtuvieron de un **FE-SEM Tescan, Mod. Mira3-LMU** microscopio.



Figura D.1: (a) Dispositivo de fuente de voltaje, (b) estructura metálica que sostiene la película.

D.2. Caracterización Eléctrica

Para determinar la rigidez dieléctrica (DR) de TiO_2 películas nano-tubulares, se usó una fuente de voltaje de potencia programable **Tectronix PWS 4721** (Ver figura D.1 (a)). Se realizó una comparación de la rigidez dieléctrica con una muestra amorfa y una cristalina de las películas nano-tubulares TiO_2 . La muestra cristalina se obtuvo cuando se expuso a $450^\circ C$ durante $2h$. La película nano-tubular TiO_2 , cuyo sustrato es la lámina Ti , se sostuvo por medio de un alfiler, que estaba en contacto con la estructura metálica de un servomecanismo de reproductor de discos compactos utilizado como herramienta para cerrar las muestras y también como un contacto eléctrico para aplicar voltaje (ver figura D.1 (b)). La estructura metálica fue conectada a la terminal negativa de la fuente de voltaje, mientras que la terminal positiva de la fuente fue conectada a un cable de cobre muy delgado y flexible (para obtener un contacto eléctrico). El cable de cobre se fijó a un carrete aislado, teniendo cuidado de no haber algún contacto con la estructura metálica para evitar un cortocircuito, tal carrete aislado podría ser desplazado micrométricamente para que el alambre de cobre pueda acercarse o alejarse de la nano-película. El desplazamiento del alambre de cobre fue monitoreado a través de una lupa. Una vez que el cable de cobre alcanza la nano-película TiO_2 , se le aplica un voltaje gradualmente desde cero al voltaje de ruptura para cada muestra registrando el comportamiento de la corriente de fuga con el voltaje aplicado.

D.3. Caracterización Física

La figura D.2 muestra algunas imágenes FE-SEM de la parte superior y la sección transversal de películas auto-organizadas TiO_2 nano-tubular. Desde las imágenes superiores (ver figura D.2 (a)), es posible observar que la forma y el diámetro de los nanotubos (que muestran una zona central con un grupo de nanotubos ligeramente más pequeños en diámetro y longitud con respecto al grupo presente a su alrededor) indican algunas diferencias en la velocidad de crecimiento

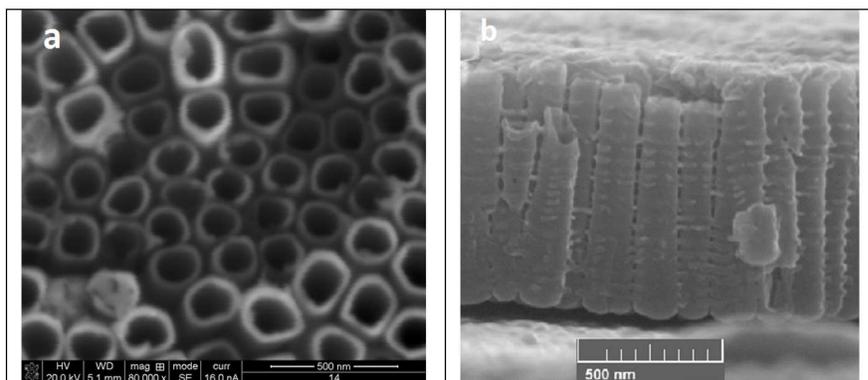


Figura D.2: Vista superior: nanotubos abiertos (a) y su vista en sección transversal (b) de una matriz nano-tubular clásica de TiO_2 auto ordenada y sintetizada a través de la exposición de láminas Ti en (50 : 50Vol.%) H_2O : glicerol + 0,27MNH $4F$ a un voltaje constante de 20V por 3,5h.

de los nanotubos. En general, las paredes de los nanotubos están bien definidas, son relativamente gruesas e independientes entre sí, y muestran un espacio libre entre las paredes externas del tubo, muy similares a las nanoestructuras simuladas de CNT del capítulo 4 de este trabajo. La imagen de la sección transversal de los nanotubos en la figura D.2 (b) muestra una dimensión (ID) de una matriz nano-tubular que presenta ondulaciones periódicamente espaciadas a su alrededor, que se ha denominado nano-arquitectura de tipo bambú con una parte superior abierta y un fondo cerrado [133].

Se realizó un cálculo promedio de la geometría de esta película para determinar el siguiente tamaño: el diámetro interno de 98nm; el grosor de la pared de 24nm y la longitud de los nanotubos de 1100nm. Los nanotubos de tipo bambú muestran un área de superficie más alta en comparación con los nanotubos lisos TiO_2 con una mayor integridad mecánica [134], lo que resulta porque las ondas están conectadas a través de las paredes externas de los nanotubos para mantenerlos unidos. La característica física única de estas matrices se debe a los nanotubos que se formaron en un electrolito de alto contenido de agua, cuya ventaja es su alta adhesión del óxido al sustrato de titanio, que atribuye buenas propiedades de transporte de electrones para aplicaciones donde la naturaleza semiconductor del titanio es importante para procesos específicos, como la fotocatalisis y las células solares sensibilizadas por colorantes [135].

D.4. Resultados

Los resultados de las mediciones de rigidez dieléctrica (DR) obtenidas para las películas nano-tubulares $TiOL$ amorfas y cristalinas se pueden ver en la figura D.2. La figura D.3 (a) muestra las curvas generadas al aplicar voltaje a las muestras amorfas y cristalinas. La figura D.3 (b) corresponde a la corriente de fuga

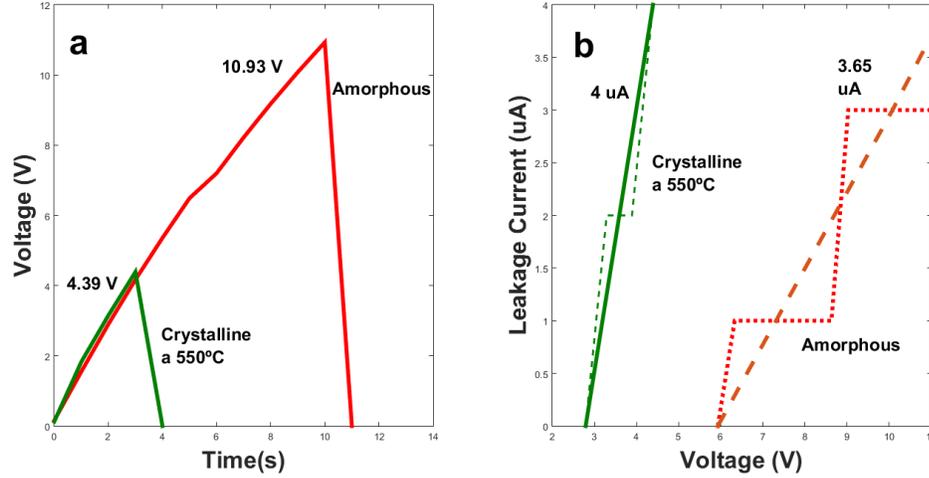


Figura D.3: Resultados del voltaje de ruptura (a) y la corriente de fuga (b) de las películas nanotubulares amorfas y cristalizadas TiO_2 obtenidas durante las pruebas de rigidez dieléctrica

generada por el voltaje aplicado. Podemos ver que se ha alcanzado el voltaje de ruptura cuando el voltaje cae a cero debido a un contacto eléctrico que se ha establecido entre el titanio y el cable de cobre causado por la ruptura de la película nanotubular TiO_2 . Esta ruptura ocurre principalmente debido a dos hechos: el intenso campo eléctrico ha superado el DR de la película nanotubular TiO_2 y su resistencia ha cambiado abruptamente de un valor alto a cero. Según la ley de Ohm ($V = IR$), cualquiera que sea la corriente de fuga en esa condición (generalmente alta), al multiplicarla por la resistencia nula, el voltaje resultante es cero. Una vez que los voltajes a los cuales las películas nano-tubulares TiO_2 han sufrido una ruptura ($10,93V$ para la muestra amorfa y $4,39V$ para la muestra cristalina) y sabiendo que el grosor de las películas es $1100nm$, es posible determinar el DR (Rigidez dieléctrica) a través de la siguiente ecuación:

$$E = V/e, \quad (D.1)$$

Donde E es la rigidez dieléctrica en *voltios/metro*, V es el voltaje de ruptura en *voltios* y e es el espesor de la película en metros. La tabla D.1 muestra los valores de rigidez dieléctrica para cada muestra. Como podemos ver en la figura D.3 (b),

TiO_2 película nano-tubular	rigidez dieléctrica (MV/m)
Amorfa	9.93
Cristalina	3.99

Tabla D.1: Rigidez dieléctrica amorfa y cristalina. TiO_2 nanotubular film

la pendiente de la corriente de fuga de la película amorfa es menor que la pendiente de la película cristalina al tener en cuenta la ley de Ohm, es posible afirmar que la resistencia de corriente continua de la muestra amorfa durante la prueba es mayor que la muestra cristalina. Dado que la corriente de fuga en ambos casos muestra

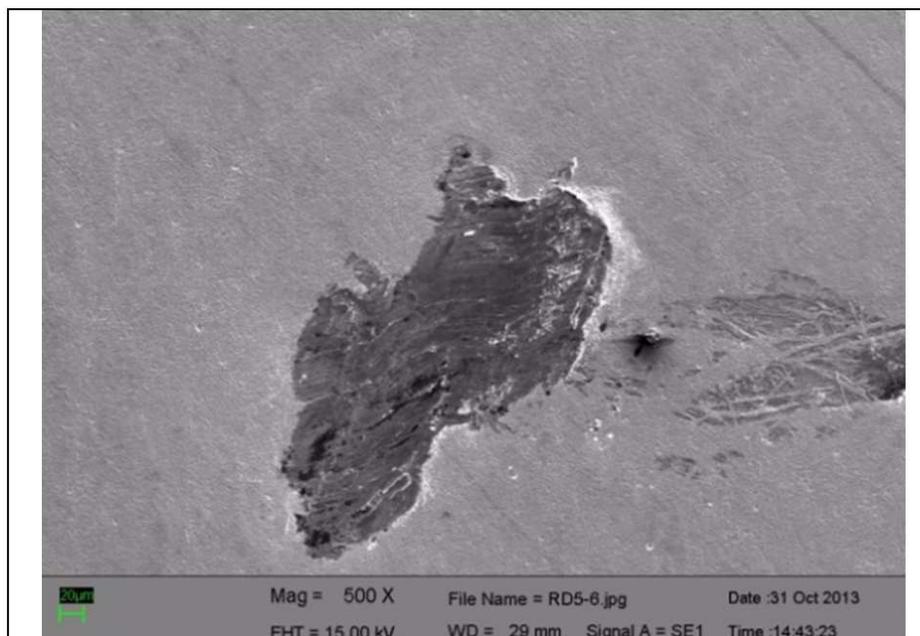


Figura D.4: Huella del área de contacto entre la película TiO_2 nano-tubular y el alambre de cobre.

<i>Película</i> nanotubular de TiO_2	Resistencia _(CD) ($M\Omega$)	Resistividad ρ ($M\Omega.m$)
Amorfa	2.994	137.0
Cristalina	1.097	50.5

Tabla D.2: Resistencia y resistividad de las muestras amorfas y cristalinas determinadas a partir de las mediciones experimentales de rigidez dieléctrica.

un comportamiento proporcional al voltaje aplicado, es posible tomar cualquier valor de corriente de fuga con su valor de voltaje correspondiente y aplicar la Ley de Ohm para determinar los valores de la resistencia que se han presentan en la Tabla D.1. Cabe señalar que estos valores no definen la resistividad de la nano película, que puede determinarse a partir de la ecuación de Ohm, así como la resistividad del material en la resistencia eléctrica en el área de contacto entre la película y el cable de cobre en metros cuadrados así como también y el grosor de los nanotubos en metros, $\rho = RS/l$. La huella del área de contacto entre la película y el cable de cobre podría estimarse en $50,7 \times 10^{-9}m^2$ a partir de la imagen FE-SEM presentada en la figura D.4. La resistividad de un material como propiedad eléctrica es estudiada más comúnmente en metales para determinar la resistencia al paso de la corriente eléctrica, sin embargo, con fines comparativos, se realizó una estimación de la resistividad de la película nano-tubular TiO_2 y se presentó en la Tabla D.2.

ECUACIONES PARA EL GRADIENTE EN REDES NEURONALES: BACKPROPAGATION

E.1. Introducción

La propagación hacia atrás o retropropagación es un método que utiliza algoritmos de aprendizaje para entrenar a redes neuronales (training) para resolver problemas generales. Este método emplea una adaptación de dos fases. En primera instancia se necesita una entrada de variables o parámetros para poder entrenar una red que se propagan a las siguientes capas (capas ocultas) de la red hasta generar una salida o resultado. El resultado se compara con el con resultado deseado y se calcula un error de resultado.

Este error de resultado se propaga de las salidas a las capas ocultas, es decir de adelante hacia atrás generando una época en cada propagación. Cada neurona de la capa oculta recibe información parcial del error para volver realizar el entrenamiento y reducir al mínimo el error. A medida que se entrena la red las neuronas se organizan de tal manera que aprender a reconocer los parámetros de entrada, de tal modo que cuando se ingresen valores semejantes a los parámetros o con un cierto nivel de ruido, la red neuronal, será capaz de producir (predecir) un resultado correcto o muy cercano al teórico con base a los parámetros entrenados. La retropropagación también puede referirse a la forma en que el resultado de una jugada se propaga por el árbol de búsqueda en la búsqueda de árbol de Monte Carlo.

En el aprendizaje automático, específicamente el aprendizaje profundo, la propagación hacia atrás (backpropagation, [136] BP) es un algoritmo ampliamente utilizado en el entrenamiento de redes neuronales de alimentación directa para el aprendizaje supervisado. ¿Existen generalizaciones de retro propagación para otras redes neuronales artificiales (ANN) y para funciones en general? Una clase de algoritmos se conoce genéricamente como retro propagación". [En el aprendizaje profundo, la propagación hacia atrás calcula el gradiente de la función de pérdida

con respecto a los pesos de la red para un solo ejemplo de entrada-salida, y lo hace de manera eficiente, a diferencia de un cálculo directo ingenuo del gradiente con respecto a cada peso individualmente. Esta eficiencia hace posible el uso de métodos de gradiente para entrenar redes multicapa, actualizar pesos para minimizar pérdidas; Descenso de gradiente, o variantes como el descenso de gradiente estocástico, se usan comúnmente. El algoritmo de retro propagación funciona calculando el gradiente de la función de pérdida con respecto a cada peso según la regla de la cadena, calculando el gradiente una capa a la vez, iterando hacia atrás desde la última capa para evitar cálculos redundantes de términos intermedios en la regla de la cadena; este es un ejemplo de programación dinámica

E.2. Derivación

El método de descenso de gradiente implica calcular la derivada de la función de pérdida con respecto a los pesos de la red. Esto normalmente se realiza mediante retro propagación. Suponiendo una neurona de salida, $[h]$ la función de error al cuadrado es:

$$E = L(t, y) \quad (\text{E.1})$$

Dónde

E es la pérdida para la salida y y el valor objetivo t ,
 t es el resultado objetivo para una muestra de entrenamiento y
 y es la salida real de la neurona de salida.

Para cada neurona j , su salida o_j se define como:

$$O_j = \varphi(\mathbf{net}_j) = \varphi\left(\sum_{k=1}^n w_{kj} O_k\right) \quad (\text{E.2})$$

Donde la función de activación φ es no lineal y diferenciable). Una función de activación utilizada históricamente es la función logística:

$$\varphi(z) = \frac{1}{e^{-z}} \quad (\text{E.3})$$

Que tiene una derivada conveniente de:

$$\frac{d\varphi(z)}{dz} = \varphi(z)(1 - \varphi(z)) \quad (\text{E.4})$$

La entrada \mathbf{net}_j a una neurona es la suma ponderada de las salidas o_k de las neuronas anteriores. Si la neurona está en la primera capa después de la capa de entrada, los o_k de la capa de entrada son simplemente las entradas x_k a la red. El número de unidades de entrada a la neurona es n . La variable w_{kj} denota el peso entre la neurona k de la capa anterior y la neurona j de la capa actual. Ver figura [E.1](#).

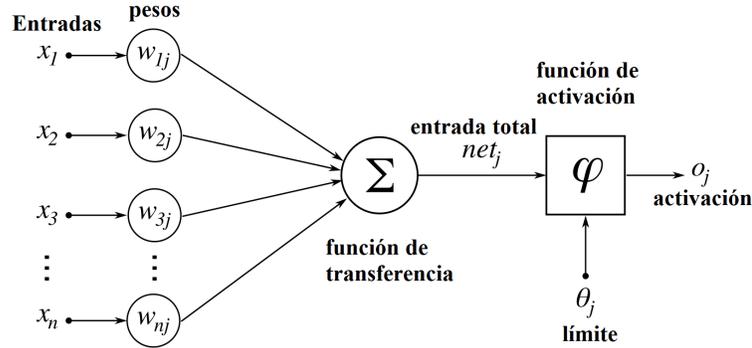


Figura E.1: Diagrama de una red neuronal artificial.

E.3. Búsqueda de la derivación de error.

El cálculo de la derivada parcial del error con respecto a un peso w_{ij} se realiza utilizando la regla de la cadena dos veces:

$$\frac{\partial E}{\partial w_{ij}} = \frac{\partial E}{\partial o_j} \frac{\partial o_j}{\partial w_{ij}} = \frac{\partial E}{\partial o_j} \frac{\partial o_j}{\partial net_j} \frac{\partial net_j}{\partial w_{ij}} \quad (\text{E.5})$$

En el último factor del lado derecho, solo un término en la suma, net_j , depende de w_{ij} , de modo que:

$$\frac{\partial net_j}{\partial w_{ij}} = \frac{\partial}{\partial w_{ij}} \left(\sum_{k=1}^n w_{kj} o_k \right) = \frac{\partial}{\partial w_{ij}} w_{ij} o_i = o_i. \quad (\text{E.6})$$

Si la neurona está en la primera capa después de la capa de entrada, o_i es solo x_i . La derivada de la salida de la neurona j con respecto a su entrada es simplemente la derivada parcial de la función de activación:

$$\frac{\partial o_j}{\partial net_j} = \frac{\partial \varphi(net_j)}{\partial net_j} \quad (\text{E.7})$$

que para el caso de la función de activación es:

$$\frac{\partial o_j}{\partial net_j} = \frac{\partial}{\partial net_j} \varphi(net_j) = \varphi(net_j)(1 - \varphi(net_j)) = o_j(1 - o_j) \quad (\text{E.8})$$

Esta es la razón por la cual la retro propagación requiere que la función de activación sea diferenciable. El primer factor es sencillo de evaluar si la neurona está en la capa de salida, porque entonces $o_j = y$ y

$$\frac{\partial E}{\partial o_j} = \frac{\partial E}{\partial y} \quad (\text{E.9})$$

Si la función logística se usa como activación y el error cuadrado como función de pérdida, podemos reescribirla como

$$\frac{\partial E}{\partial o_j} = \frac{\partial E}{\partial y} = \frac{\partial}{\partial y} \frac{1}{2} (t - y)^2 = y - t \quad (\text{E.10})$$

Sin embargo, si j está en una capa interna arbitraria de la red, encontrar la derivada E con respecto a o_j es menos obvio. Considerando E como una función con las entradas siendo todas las neuronas $L = \{u, v, \dots, w\}$ recibiendo información de la neurona j .

$$\frac{\partial E(o_j)}{\partial o_j} = \frac{\partial E(\text{net}_u, \text{net}_v, \dots, \text{net}_w)}{\partial o_j} \quad (\text{E.11})$$

y tomando la derivada total con respecto a o_j , se obtiene una expresión recursiva para la derivada:

$$\frac{\partial E}{\partial o_j} = \left(\sum_{\ell \in L} \frac{\partial E}{\partial \text{net}_\ell} \frac{\partial \text{net}_\ell}{\partial o_j} \right) = \sum_{\ell \in L} \left(\frac{\partial E}{\partial o_\ell} \frac{\partial o_\ell}{\partial \text{net}_\ell} \frac{\partial \text{net}_\ell}{\partial o_j} \right) = \sum_{\ell \in L} \left(\frac{\partial E}{\partial o_\ell} \frac{\partial o_\ell}{\partial \text{net}_\ell} w_{j\ell} \right) \quad (\text{E.12})$$

Por lo tanto, la derivada con respecto a o_j se puede calcular si se conocen todas las derivadas con respecto a las salidas o_ℓ de la siguiente capa, las más cercanas a la neurona de salida. Tenga en cuenta que si alguna de las neuronas en el conjunto L no estuviera conectada a la neurona j , serían independientes de w_{ij} y la derivada parcial correspondiente bajo la suma desaparecería a 0.

Sustituyendo las ecuaciones E.6, E.7, E.8 y E.9 en la ecuación E.5 obtenemos:

$$\frac{\partial E}{\partial w_{ij}} = \frac{\partial E}{\partial o_j} \frac{\partial o_j}{\partial \text{net}_j} \frac{\partial \text{net}_j}{\partial w_{ij}} = \frac{\partial E}{\partial o_j} \frac{\partial o_j}{\partial \text{net}_j} o_i \quad (\text{E.13})$$

$$\frac{\partial E}{\partial w_{ij}} = o_i \delta_j \quad (\text{E.14})$$

con,

$$\delta_j = \frac{\partial E}{\partial o_j} \frac{\partial o_j}{\partial \text{net}_j} = \begin{cases} \frac{\partial L(o_j, t)}{\partial o_j} \frac{d\varphi(\text{net}_j)}{d\text{net}_j} \text{ si } j \text{ es una neurona de salida,} \\ \left(\sum_{\ell \in L} w_{j\ell} \delta_\ell \right) \frac{d\varphi(\text{net}_j)}{d\text{net}_j} \text{ si } j \text{ es una neurona de union,} \end{cases} \quad (\text{E.15})$$

si φ es la función logística, y el error es el error cuadrado:

$$\delta_j = \frac{\partial E}{\partial o_j} \frac{\partial o_j}{\partial \text{net}_j} = \begin{cases} (o_j - t_j) o_j (1 - o_j) \text{ si } j \text{ es una neurona de salida,} \\ \left(\sum_{\ell \in L} w_{j\ell} \delta_\ell \right) o_j (1 - o_j) \text{ si } j \text{ es una neurona de union,} \end{cases} \quad (\text{E.16})$$

Para actualizar el peso w_{ij} usando el descenso en gradiente, se debe elegir una tasa de aprendizaje, $\eta > 0$. El cambio en el peso debe reflejar el impacto en E de un aumento o disminución en w_{ij} . Si $\frac{\partial E}{\partial w_{ij}} > 0$, un aumento en w_{ij} aumenta E , por el contrario, si $\frac{\partial E}{\partial w_{ij}} < 0$, un aumento en w_{ij} disminuye E . El nuevo Δw_{ij} se agrega al peso anterior, y el producto de la tasa de aprendizaje y el gradiente, multiplicado por -1 garantiza que w_{ij} cambia de una manera que siempre disminuye E . En otras palabras, en la ecuación inmediatamente debajo de $-\eta \frac{\partial E}{\partial w_{ij}}$ siempre cambia

w_{ij} de tal manera que E se reduce a:

$$\Delta w_{ij} = -\eta \frac{\partial E}{\partial w_{ij}} = -\eta o_i \delta_j \quad (\text{E.17})$$

E.4. Función de pérdida

La función de pérdida es una función que asigna valores de una o más variables a un número real que representa intuitivamente algún costo asociado con esos valores. Para la retropropagación, la función de pérdida calcula la diferencia entre la salida de la red y su salida esperada, después de que un ejemplo de entrenamiento se haya propagado a través de la red.

E.5. Suposiciones

La expresión matemática de la función de pérdida debe cumplir dos condiciones para que pueda ser utilizada en la propagación hacia atrás [137]. La primera es que se puede escribir como un promedio $E = \frac{1}{n} \sum_x E_x$ sobre las funciones de error E_x , para n ejemplos de entrenamiento individual x . La razón de esta suposición es que El algoritmo de retropropagación calcula el gradiente de la función de error para un solo ejemplo de entrenamiento que debe generalizarse a la función de error general. La segunda suposición es que se puede escribir en función de las salidas de la red neuronal.

E.6. Ejemplo de función de pérdida

Supongamos que y, y' sean vectores en R^n . Seleccione una función de error $E(y, y')$ que mide la diferencia entre dos salidas. La opción estándar es el cuadrado de la distancia euclidiana entre los vectores y y y' :

$$E(y, y') = \frac{1}{2} \| y - y' \|^2 \quad (\text{E.18})$$

La función de error sobre n ejemplos de entrenamiento se puede escribir como un promedio de pérdidas sobre ejemplos individuales:

$$E = \frac{1}{2n} \sum_x \| (y(x) - y'(x)) \|^2 \quad (\text{E.19})$$

ARTICULOS PUBLICADOS

1. Burlak, G., and Medina-Angel, G. (2018). The Optimization at Studying of Electrical Conductivity in the Dielectric Nanocomposites with Disordered Nanotubes. *Progress In Electromagnetics Research*, 74, 77-82.
2. Medina-Ángel, G., and Burlak, G. (2018). Estudio de la conductividad eléctrica en nano compuestos eléctricos con nano tubos desordenados, *Revista de Ingeniería Eléctrica*, 2(5), 18-26.
3. Burlak, G., and Medina-Ángel, G. (2019). Structure of optical Cherenkov radiation in a three-dimensional compound medium with carbon nanotubes. *JOSA B*, 36(2), 187-193.
4. Burlak, G., Cuevas-Arteaga, C., Medina-Ángel, G., Martínez-Sánchez, E., and Calderón-Segura, Y. Y. (2019). Plasmon-polariton oscillations in three-dimensional disordered nanotubes excited by a moving charge. *Journal of Applied Physics*, 126(1), 013101.
5. Medina-Ángel, G., and Burlak, G. (2019). Software para calcular la probabilidad crítica de percolación en un sistema de nanotubos desordenados aplicado al estudio de sus radios en materiales conductores. *Programación Matemática y Software*, 11(2), 24-34.
6. Medina, G., Calderón, Y. Y., Burlak, G., and Hernández, J. A. (2020). Study of the critical probability of percolation in a 3D system with pores of random radius for variable grids. *Revista Mexicana de Física*, 66(3 May-Jun), 315-321 [[138](#)].

Progress In Electromagnetics Research Letters, Vol. 74, 77–82, 2018

The Optimization at Studying of Electrical Conductivity in the Dielectric Nanocomposites with Disordered Nanotubes

Gennadiy Burlak* and Gustavo Medina-Angel

Abstract—We study the electrical conductivity of a three-dimensional (3D) nanocomposite with incorporated random carbon nanotubes (CNT). A large length of the remote nanotubes generates a lot of intersections that induce a rather small percolating threshold of the global conductivity in this medium. We simulate such a system by random cylinders placed in a percolating parallelepiped with the use of Monte Carlo method. The conductivity of such a structure is associated with the critical phenomena, where the main transition parameter is defined by the value of the percolation threshold. We calculate the minimal percolating threshold and determine the functional form of the conductivity by the global optimization technique. Such an approach allows studying the details of the electrical conductivity in nanocomposites even at significant level of the percolating fluctuations.

1. INTRODUCTION

The incorporation of disordered carbon nanotubes (CNT) in a dielectric nanocomposite leads to formation of spatial percolating channels. Groups of such nanotubes are jointed in clusters, located in a chaotic fashion in a nanocomposite. The integration of separated clusters happen when the concentration of nanotubes approaches a certain critical value. As a result in the system spreading (infinite) cluster is formed, which leads to appearance of the electric contacts between external electrodes in such a compound [1–16]. Experimental studies [1] have shown that the conductivity in nanotube polymeric and ceramic matrix materials follows a percolation-like behavior. Specifically, at a relatively low concentration of nanotubes the conductivity follows the scaling law of percolation theory [17]. The percolation in a dielectric nanocomposite with carbon nanotube is of high interest due to the potential to create electrically conductive systems with an extremely low mass of carries.

It is important to note that in the system with long cylinders the nearest neighbors approximation is not applicable. A large length of the nanotubes (e.g., in [10] it is reported that ultralong CNTs over 18.5 cm long were grown on Si substrates) leads to the fact that the global conductivity occurs in nanocomposites due to contacts of remote objects. The upper boundary of the contacts numbers can be estimated in the following simple way. Two straight (no parallel) lines (nanotubes with high aspect ratio) have only one point of intersection. Three straight lines can have three contacts. It is easy to see that n (not parallel) straight lines have in general $n(n - 1)/2$ intersections which at $n \gg 1$ produce $n^2/2$ contacts. In the case of $n = 100$, there can be 5000 contacts. Such a large number of contacts increases conductivity and reduces the influence of details of internal contacts for nanotubes. Besides, as shown in our paper, the independent randomness in height and radius of the cylinders becomes an important factor in vicinity of the percolating threshold. In what follows, we refer to the system with configuration similar to experiment [1] where the samples were connected to incorporated nanotubes by external metallic electrodes. The electric conductivity in such a compound

Received 4 December 2017, Accepted 3 March 2018, Scheduled 23 March 2018

* Corresponding author: Gennadiy Burlak (gburlak@uaem.mx).

The authors are with the CIICAP, Universidad Autónoma del Estado de Morelos, Av. Universidad 1001, Cuernavaca, Morelos 62210, México.

Estudio de la conductividad eléctrica en nano compuestos eléctricos con nano tubos desordenados**Study of the electrical conductivity in electrical nanocomposites with disordered nanotubes**

MEDINA-ÁNGEL, Gustavo†* & BURLAK, Gennadiy

*CIICAp, Universidad Autónoma del Estado de Morelos, Avenida Universidad 1001, Colonia Chamilpa, Cuernavaca, Morelos, México, C.P. 62209.*ID 1^{er} Autor: *Gustavo, Medina-Ángel* / ORC ID: 0000-0002-0279-3492, Researcher ID Thomson: G-7236-2018, CVU CONACYT ID: 624958ID 1^{er} Coautor: *Gennadiy, Burlak* / ORC ID: 0000-0003-4829-8435, Researcher ID Thomson: T-7967-2018, CVU CONACYT ID: 20332

Recibido Mayo 27, 2018; Aceptado Junio 30, 2018

Resumen

Estudiamos la conductividad eléctrica de nano compuestos en una rejilla de tres dimensiones (3D) incorporando nano tubos de carbono (CNT) de forma aleatoria dentro del sistema. Nosotros variamos la longitud y radio de los nanotubos utilizando una distribución normal (Gauss) y generamos una posición aleatoria del nano tubo dentro del sistema usando el método Monte Carlo. La conductividad del sistema es asociada al fenómeno crítico de percolación en los CNT, en donde la intercepción o contacto entre los nano tubos crean un clúster infinito que es capaz de transmitir la conductividad eléctrica. Nosotros calculamos la probabilidad mínima necesaria para la transmisión de conductividad, mediante la técnica de optimización global en paralelo que permite estudiar el comportamiento de la conductividad en nano compuestos. Este enfoque nos permite estudiar los detalles de la conductividad eléctrica en nano compuestos eléctricos, incluso al nivel de las fluctuaciones de filtración en materiales a nano escala en el campo de la ingeniería y conductividad en los materiales.

Nano-Compuestos, Conductividad, Percolación**Abstract**

We study the electrical conductivity of a three-dimensional (3D) nanocomposite with incorporated random carbon nanotubes (CNT) within the system. We varied the length and radius of the nanotubes using a normal distribution (Gauss) and generated a random position of the nano tube within the system using the Monte Carlo method. The conductivity of the system is associated with the critical phenomenon of percolation in the CNT, where the interception or contact in the nano tubes creates an infinite group that is capable of transmitting the electrical conductivity. We calculate the minimum probability necessary for the transmission of conductivity, by means of the technique of global optimization in parallel that allows to study the behavior of the conductivity in nanocomposites. This approach allows us to study the details of the electrical conductivity in electrical nanocomposites, even at the level of the fluctuations of filtering in nano-scale materials in the field of engineering and conductivity in materials.

Nanocomposites, Conductivity, Percolation

Citación: MEDINA-ÁNGEL, Gustavo & BURLAK, Gennadiy. Estudio de la conductividad eléctrica en nano compuestos eléctricos con nano tubos desordenados. *Revista de Ingeniería Eléctrica*. 2018. 2-5: 18-26.

† Investigador contribuyendo como primer autor.

*Correspondencia al Autor Correo Electrónico: (gustavo.isc@hotmail.com)



Structure of optical Cherenkov radiation in a three-dimensional compound medium with carbon nanotubes

GENNADIY BURLAK^{1,*} AND GUSTAVO MEDINA-ÁNGEL^{1,2}

¹CIICAp, Universidad Autónoma del Estado de Morelos, Av. Universidad 1001, Cuernavaca, Morelos 62210, Mexico

²e-mail: gustavo.isc@hotmail.com

*Corresponding author: gburlak@uaem.mx

Received 17 September 2018; revised 22 October 2018; accepted 22 November 2018; posted 26 November 2018 (Doc. ID 345599); published 7 January 2019

We study the optical Cherenkov radiation in a compound three-dimensional (3D) system with a charge moving over a periodic system of parallel carbon nanotubes. To consider a case of bounded single-walled nanotubes, we apply the numerical finite-difference time-domain technique. We found that, because of the broadband radiation of a charge, the spectrum of the optical field in such a nanosystem is determined by the surface plasmon-polariton excitations. Our study of the field energy spectrum as a function of the plasma frequency detects a nonmonotonic double-well dependency within the area of a surface plasmon-polariton resonance. A well-defined peak that indicates a generation of collective excitations in nanotubes by Cherenkov field is registered. ©2019 Optical Society of America

<https://doi.org/10.1364/JOSAB.36.000187>

1. INTRODUCTION

Plasmonics represents a new area for the application of surface and interface plasmons in a nanoscale sphere of photonics and electronics [1]. The surface plasmon-polaritons can serve as a basis for creation of nanoscale optical circuits that will be able to carry the signals in various frequency ranges [2–9]. The peculiarities of interaction of conducting carbon nanotubes (CNTs) of a cylindrical shape with optical fields are of great interest. To model such objects, the theory of conducting dispersion cylinders [10–12] is used. The properties of such systems in different configurations have been studied in detail in papers dealing with both optical communication and plasma physics are already well established. Important properties of such systems are the narrowband signal of the source and their considerable spatial extent, which allows the formation of well-defined independent stable spatial modes. Nevertheless, although such cylindrical structures are relatively simple, the classification of the modes and the understanding of the observed effects here is far from being trivial. The situation becomes more complicated in the case of CNTs [13–16]. First, the random factor is important here, because in view of technological peculiarities of CNT preparation, the length of nanotubes, as well as the plasma frequency of their walls, varies within rather wide limits. The latter leads to the fact that the frequencies of the surface plasmons-polaritons turn out to be located in a rather wide range. Second, at the small distance between nanotubes in the CNT system, the internal cross-coupling between the CNTs

may arise at the moderate localization of plasmons (when the wavelength is comparable to the average distance between nanotubes), which leads to new effects associated with collective field oscillations in such a CNT system. The Cherenkov radiation arises when the electron velocity exceeds the speed of light in a transparent medium [17,18]. The induced charges and currents near inhomogeneous systems give rise to radiative losses. The frequency spectrum of Cherenkov radiation by a particle in an infinite dielectric is given by

$$\frac{d^2 E}{dx d\omega} = \frac{q^2}{4\pi} \omega \left(1 - \frac{c^2}{v^2 n^2(\omega)} \right), \quad (1)$$

which describes the amount of energy emitted from the Cherenkov radiation, per wavelength, per unit length traveled. Here $n(\omega)$ is the index of refraction of the material the charge particle moves through with velocity v . Cherenkov radiation is a frequency continuous and around the visible spectrum; the relative intensity per unit frequency is approximately proportional to the frequency ω . That is, higher frequencies are more intense in Cherenkov radiation. In a transparent material, this radiation is emitted in an angle θ interval. $\sin \theta d\theta = c/vn^2(\omega)(dn(\omega)/d\omega)$. Recently, Cherenkov radiation has been widely investigated in case of various materials, metamaterials, and different setups [19–25]. The paper is organized as follows. In Section 2, we formulate the field equations that allow simple analytical description for conducting infinite cylindrical nanotubes. In Section 3, we study the excitation of field in CNTs

Plasmon-polariton oscillations in three-dimensional disordered nanotubes excited by a moving charge

Cite as: *J. Appl. Phys.* **126**, 013101 (2019); doi: [10.1063/1.5098019](https://doi.org/10.1063/1.5098019)

Submitted: 30 March 2019 · Accepted: 11 June 2019 ·

Published Online: 1 July 2019



Gennadiy Burlak,^{1,a)} Cecilia Cuevas-Arteaga,^{1,b)} Gustavo Medina-Ángel,^{1,c)} Erika Martínez-Sánchez,^{2,d)} and Yessica Y. Calderón-Segura^{1,e)}

AFFILIATIONS

¹CIICAp, Universidad Autónoma del Estado de Morelos, Av. Universidad 1001, Cuernavaca, Morelos 62210, Mexico

²Facultad de Ingeniería, Universidad Autónoma de Coahuila, Blvd. Fundadores Km. 13 Ciudad Universitaria, Arteaga, Coahuila 25354, Mexico

^{a)}Electronic mail: gburlak@uaem.mx

^{b)}Electronic mail: ccuevas@uaem.mx

^{c)}Electronic mail: gustavo.isc@hotmail.com

^{d)}Electronic mail: erika.martinez@uadec.edu.mx

^{e)}Electronic mail: ycaiderons@uaem.mx

ABSTRACT

We systematically investigated the plasmon-polariton oscillations generated by a fast radiating charge (Cherenkov radiation) in a three-dimensional (3D) strongly disordered nanostructure. We studied the dynamic properties of an optical field in a random composition of empty single-wall nanotubes by using a 3D numerical finite-difference time domain technique. In our approach, only parameters of nanotube structures are fixed. The dynamic spectrum of internal field excitations was left to be defined as a result of numerical simulation. The patterns of total field (charge + carbon nanotubes) are determined by the interference of a moving charge field and the spectrum of surface plasmon-polaritons in disordered nanotubes. We found that the field energy losses, as a function of the charge velocity, has a clearly pronounced maximum when the characteristic frequency scale (defined by a charge velocity) is close to the frequency of the surface plasmon-polariton resonances generated in coupled nanotubes, even at a significant level of disorder. Our studies show that the shape of the resonance peak, depending on the charge velocity, is similar for carbon and TiO₂ nanostructures, but, only for frequencies from the range of the surface plasmon polaritons of respective materials. The TiO₂ nanostructure films for a dassic cylindrical polytetrafluoroethylene cell was synthesized in our experiments too.

Published under license by AIP Publishing. <https://doi.org/10.1063/1.5098019>

I. INTRODUCTION

The development of spectroscopic techniques on a spatial scale of a few nanometres has been a topic of increasing interest in recent years.¹ The rapid progress in new material technologies has led to many advantages in the new field of nanoplasmonics. Nowadays, nanoplasmonics is an area of research where surface plasmons, generated as a result of the interaction of electromagnetic radiation and conduction electrons in dielectric materials, can be applied² on the polaritons that detach from the surface of a dielectric material. Such excitations can serve as a basis for the generation of new optical circuits at a nanoscale level that can carry signals in different frequency ranges.^{3–11} The conduction properties of the carbon nanotubes (CNTs) allow for the generation of optical

fields that allow the collective oscillations of the conduction electrons. The theory of conducting dispersion cylinders is used to model such CNTs.^{12–14} The CNT, which has important properties for nanotechnology (and also for optical communication), takes advantage of the narrow band that allows for the formation of independent and well-defined spatial modes for nanostructures.^{16–19} In Ref. 20, the excitation of collective surface plasmons by a charge moving with uniform velocity in regular 3D carbon nanotubes (CNTs) was studied in the case of a very small level of disorder. However, in general, the level of disorder of nanotubes is significantly high. In particular, such a situation (with a significant level of 3D disorder) is studied in this paper. In this case, not only the physical parameters (length and radius) of the nanotubes are random, but also can be random in both mutual orientation and distance. The latter leads to

Software para calcular la probabilidad crítica de percolación en un sistema de nanotubos desordenados aplicado al estudio de sus radios en materiales conductores

Software to calculate the probability of percolation in a disordered nanotube system applied to the study of its radios in conductive materials

Gustavo Medina Ángel^{1*}, Gennadiy Burlak¹

¹Centro de Investigación en Ingeniería y Ciencias Aplicadas,
Universidad Autónoma del Estado de Morelos.

Avenida Universidad 1001, Colonia Chamilpa, Cuernavaca, Morelos, México, C.P. 62209.

*Correo-e: gustavo.isc@hotmail.com

PALABRAS CLAVE:

Rejilla, Hilos, Desviación, CNT.

RESUMEN

Generamos un software en donde simulamos un sistema 3D, en el cual implementamos nanotubos variando sus grosores y longitudes mediante una distribución normal.

Asociamos nuestro sistema de simulación a la conductividad en sistemas que pueden o no conducir la energía eléctrica de acuerdo a la cantidad de elementos que lo conforman y con dependencia directa de su probabilidad crítica. Consideramos los grosores o radios de los nanotubos para generar clústeres infinitos que logran conducir la energía eléctrica a través del sistema simulando en un material conductor.

Este estudio nos permite conocer los detalles en dimensiones nanométricas y ayuda a tener una noción más cercana del comportamiento en la construcción de materiales conductores.

KEYWORDS:

Grid, Threads, Desviation, CNT.

ABSTRACT

We generate a software in which we simulate a 3D system, in which we implement nanotubes varying their radios and lengths with normal distribution.

We associate our simulation system with the conductivity in systems that may or may not conduct electrical energy according to the number of elements that conform it and with direct dependence on its critical probability. We considered the thicknesses or radios of the nanotubes to generate infinite clusters that managed to conduce the electrical energy through the system simulating in a conductor of materials.

This study allows us to know the details in nanometric dimensions and help to have a notion closer to the behavior in the construction of conductive materials.

Study of the critical probability of percolation in a 3D system with pores of random radius for variable grids

G. Medina-Ángel^{a,b}, Y. Y. Calderón-Segura^b, G. Burlak^a, and J. A. Hernández-Aguilar^b

^aCentro de Investigación en Ingeniería y Ciencias Aplicadas,

^bFacultad de Contaduría, Administración e Informática, Universidad Autónoma del Estado de Morelos, Av. Universidad 1001, Cuernavaca, Morelos 62210, México.

Received 4 December 2019; accepted 19 February 2020

We numerically study the percolation in 3D porous materials, populated by pores with random sizes on 3D grid of variable sizes. We identify the clusters for each grid as well as the infinite cluster that is defined by the critical probability through the neighborhood hybrid structure method. We also determine the characteristic size of each cluster in the material as well as the volume of the infinite cluster that allows optimizing the percolation step at our simulation. In this work, several tests were performed changing the size of the grid. This allows us to determine the optimal size and how it affects the percolation by the simulating grids. Our main results show that in systems with pores having random radii the critical probability increases when size of grid $L > 40$ (that correspond to typical size system about 4000 nm) with respect of the case with uniform pores.

Keywords: Clustering; percolation; pores; grid.

PACS: 21.60.Gx; 07.05.Tp; 61.43.Bn; 87.15.Zg; 36.40.Ei

DOI: <https://doi.org/10.31349/RevMexFis.66.315>

1. Introduction

Percolation theory is a general mathematical theory of transport and connectivity in random complex systems. Percolation is a random process developed from the observation of connectivity between microscopic elements and their effects on macroscopic properties [1]. Percolation looks for a trajectory between the porosity of the medium in a random way, in this mode the porosity of the medium from one end to another. The fluid can be a liquid, steam, heat flow, electric current [2], infection or any fluid or property that can move through a porous medium. For example, water on the ground under the action of gravity, sewage treatment plants. Currently, different resistant and low-cost materials are being developed where the material must be light enough, but with a degree of porosity that does not allow liquids to seep through it. Contrary to the manufacture of membranes or materials, that filters fluids in order to eliminate impurities [3]. In this work, we study the percolation in nano-structures [4–7] composed of nano-particles ($N - P$) such as carbon-gel glucose ($AG - CNT$) [8] which vary from a size of 1 nm to nano-particles of SiO_2 with a size of 100 nm [9]. In this work, we apply a porosity level in a simulated material to know if there is percolation in the material. In Fig. 1, the simulation of a porous material is shown on a cubic grid of size $L = 10$ [10, 11] with the value of $L \times L \times L = 1000$, comprising a grid of the size of $100,000 \text{ nm}^3$.

2. Probability and critical probability “ P_c ”

For a 3D grid G , we use pores of various sizes [12]. To generate the radii of the pores we use a normal distribution, see Eq. (1), with $\mu_r = 0.5$ and $\sigma_r = 0.2$ in the simulated grid

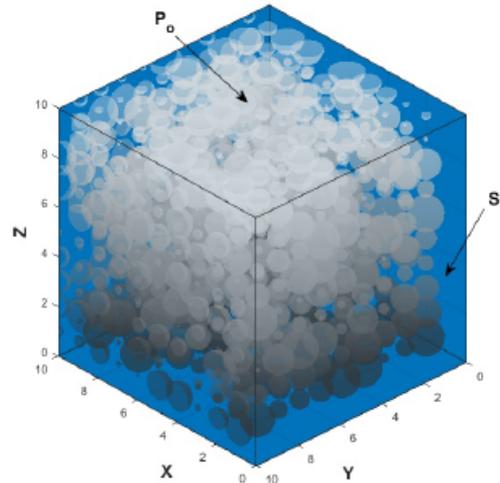


FIGURE 1. (Color online) 3D simulation of a porous material, with variable radii in its pores, where P_o is the hollow pore inside the grid and S is the solid material.

where r is normalized to the typical system size. Here P is the probability of pores that were generated in the system. The number of pores is determined by the level of porosity called probability P in the system, the larger the probability P in the system.

$$\frac{1}{\sqrt{2\pi}\sigma_r} e^{-\frac{(r-\mu_r)^2}{2\sigma_r^2}}. \quad (1)$$

The critical probability known as P_c in a material is referred to as the Probability P necessary to achieve percolation in a material.

Bibliografía

- [1] M. Loos, Carbon nanotube reinforced composites: CNT Polymer Science and Technology, Amsterdam: Elsevier., 2015.
- [2] Coleman, J. N., S. Curran, A. B. Dalton, A. P. Davey, B. McCarthy, W. Blau, and R. C. Barklie, "Percolation-dominated conductivity in a conjugated-polymer-carbon-nanotube composite," *Phys. Rev. B*, Vol. 58, 7462-7495, 1998.
- [3] Berhan, L., and A. M. Sastry, "Modeling percolation in high-aspect-ratio fiber systems. I. Soft-core versus hard-core models," *Phys. Rev. E*, Vol. 75, 1-8, 2007.
- [4] Eletsii, A. V., A. A. Knizhnik, B. V. Potapkin, J. M. Kenny, "Electrical characteristics of carbon nanotube-doped composites," *Physics - Uspekhi*, Vol. 58, 225-270, 2015.
- [5] Foygel, M., R. D. Morris, D. Anez, S. French, and V.L. Sobolev, "Theoretical and computational studies of carbon nanotube composites and suspensions: Electrical and thermal conductivity", *Phys. Rev. B.*, Vol. 71, 104201.1-104201.8, 2005.
- [6] Ma, H.M., X.-L. Gao, "A three-dimensional Monte Carlo model for electrically conductive polymer matrix composites filled with curved fibers," *Polymer*, Vol. 49, 4230-4238, 2008.
- [7] Heng Gu, Jiaojiao Wang, Choongho Yu, "Three-dimensional Modeling of Percolation Behavior of Electrical Conductivity in Segregated Network Polymer Nanocomposites Using Monte Carlo Method," *Advances in Materials*, Vol. 5, 1-8, 2016.
- [8] Kuo-Chi Lin, Daniel Lee, Linan An, Young Hoon Joo, "Finite-Size Scaling Features of Electric Conductivity Percolation in Nanocomposites", *Nanoscience and Nanoengineering*, Vol. 1, 15-22, 2013.

- [9] Ning Hu, Zen Masuda, Cheng Yan, Go Yamamoto, "The electrical properties of polymernanocomposites with carbon nanotubefillers", *Nanotechnology*, Vol. 19, 215701, 2008.
- [10] Sagalianov, I., L. Vovchenko, L. Matzui, and O. Lazarenko, "Synergistic Enhancement of the Percolation Threshold in Hybrid Polymeric Nanocomposites Based on Carbon Nanotubes and Graphite Nanoplatelets", *Nanoscale Research Letters*, Vol. 12, No. 140, 2017.
- [11] Xueshen Wang, Qunqing Li, Jing Xie, Zhong Jin, Jinyong Wang, Yan Li, "Fabrication of Ultralong and Electrically Uniform Single-Walled Carbon Nanotubes on Clean Substrates", *Nano lett.*, Vol. 9, 3137-3141, 2009.
- [12] A. M. Attiya, "Lower frequency limit of carbon nanotube antenna," *Progress In Electromagnetics Research*, Vol. 94, 419-433, 2009.
- [13] M. Aidi and T. Aguil, ".Electromagnetic modeling of coupled carbon nanotube dipole antennas based on integral equations system," *Progress In Electromagnetics Research M*, Vol. 40, 179-183, 2014.
- [14] S. M. Mikki and A. A. Kishk, "Derivation of the carbon nanotube susceptibility tensor using lattice dynamics formalism," *Progress In Electromagnetics Research B*, Vol. 9, 1-26, 2008.
- [15] D. Bychanok, G. Gorokhov, D. Meisak, P. Kuzhir, S. A. Maksimenko, Y. Wang, Z. Han, X. Gao, and H. Yue, "Design of carbon nanotube-based broadband radar absorber for ka-band frequency range," *Progress In Electromagnetics Research M*, Vol. 53, 9-16, 2017.
- [16] Q. Dai, H. Butt, R. Rajasekharan, T. D. Wilkinson, and G. A. J. Amaratunga, "fabrication of carbon nanotubes on inter - digitated metal electrode for switchable nanophotonic devices ," *Progress In Electromagnetics Research*, Vol. 127, 65-77, 2012.
- [17] P. Savi, M. Yasir, M. Giorcelli, and A. Tagliaferro, "The effect of carbon nanotubes concentration on complex permittivity of nanocomposites," *Progress In Electromagnetics Research M*, Vol. 55, 203-209, 2017.
- [18] Kane S.Yee. (1966) Numerical Solution of initial Boundary Value Problem Involving Maxwell's Equatios in Isotropic Media *IEEE Trans. on Antennas and propagation.* **14**,302-307.
- [19] A. Taflove and S. C. Hagness, *Computational Electrodynamics: the finite-difference time-domain method*. Boston: Artech House, Inc., 2000.
- [20] N. C. Frateschi and A. F. J. Levi, The spectrum of microdisk lasers, *J. Appl. Phys.*, vol. 80, p. 644, 1996.

- [21] J. M. Gérard, D. Barrier, J. Y. Marzin, R. Kuszelewicz, L. Manin, E. Costard, V. Thierry-Mieg, and T. Rivera, Quantum boxes as active probes for photonic microstructures: The pillar microcavity case, *Appl. Phys. Lett.*, vol. 69, p. 499, 1996.
- [22] H. Y. Ryu, M. Notomi, and Y. H. Lee, High-quality-factor and small-mode-volume hexapole modes in photonic-crystal-slab nanocavities, *Appl. Phys. Lett.*, vol. 83, p. 4294, 2003.
- [23] Y. Akahane, T. Asano, B. S. Song, and S. Noda, *Nature*, vol. 425, p. 944, 2003.
- [24] P. R. Villeneuve, S. Fan, and J. D. Joannopoulos, Microcavities in photonic crystals: Mode symmetry, tunability, and coupling efficiency, *Phys. Rev. B*, vol. 54, p. 7837, 1996.
- [25] P. R. Villeneuve, S. Fan, and J. D. Joannopoulos, Numerical solution of initial boundary value problems involving maxwell's equations in isotropic media, *IEEE Trans. Antennas and Propagation*, vol. 14, p. 302, 1966.
- [26] Y. Xu, J. Vukovic, R. K. Lee, O. J. Painter, A. Scherer, and A. Yariv, Finite-difference time-domain calculation of spontaneous emission lifetime in a microcavity, *J. Opt. Soc. Am. B*, vol. 16, p. 465, 1999.
- [27] J. D. Jackson, *Classical Electrodynamics*. New York: Wiley, 1974.
- [28] O. Painter and K. Srinivasan, Polarization properties of dipolelike defect modes in photonic crystal nanocavities, *Opt. Lett.*, vol. 27, p. 339, 2002.
- [29] S. H. Kim and Y. H. Lee, Symmetry relations of two-dimensional photonic crystal cavity modes, *IEEE J. Quantum Electron.*, vol. 39, p. 1081, 2003.
- [30] R. B. Laughlin, *A Different Universe: Reinventing Physics from the Bottom Down*. New York: A Member of the Perseus Books Group, 2005.
- [31] J. Vukovic, M. Loncar, H. Mabuchi, and A. Scherer, *IEEE J. Quantum Electron.*, vol. 38, p. 850, 2002.
- [32] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling, *Numerical Recipes in C: The Art of Scientific Computing*, 2nd ed. Cambridge University Press, 1992.
- [33] Kim, S. H., Kim, S. K., and Lee, Y. H. (2006). Vertical beaming of wavelength-scale photonic crystal resonators. *Physical Review B*, 73(23), 235117.
- [34] Sullivan, D. M. (2013). *Electromagnetic simulation using the FDTD Sullivan method*. John Wiley and Sons.
- [35] Anguera, J. and Andújar, A. Antenas: Elementos Indispensables de las Telecomunicaciones, *Cienciaprop* 1(1), 2018.

- [36] J M. Pitarke, V.M. Silkin, E.V. Chulkov, P.M. Echenique, "Theory of surface plasmons and surface-plasmon, Rep. Prog. Phys. 70, 1-87 (2007).
- [37] W.L. Barnes, A. Dereux and T.W. Ebbesen, "Surface plasmon subwavelength optics," Nature 424, 824 (2003).
- [38] N. Wataru , O. Motoichi , Y. Takashi, "Nanodot coupler with a surface plasmon polariton condenser for optical far/near-field conversion, .Appl. Phys. Lett. 86, 181108 (2005).
- [39] W. L. Barnes, W. A. Murray, J. Dintinger, E. Devaux, and T.W. Ebbesen, "Surface plasmon polaritons and their role in the enhanced transmission of light through periodic arrays of subwavelength holes in a metal film," Phys. Rev. Lett. 92, 107401 (2004).
- [40] M. A. Kaliteevski, S. Brand, R. A. Abram, V. V. Nikolaev, M. V. Maximov, N. N. Ledentsov, C. M. Sotomayor Torres, and A. V. Kavokin, ".Exciton polaritons in a cylindrical microcavity with an embedded quantum wire," Phys. Rev. B 61, 13791 (2000).
- [41] V. Federico , B. Markus , S. Stephan , W. Nadine , H. Detlef, and M. Stefan , "Strong coupling between surface plasmon polariton and laser dye rhodamine 800, .Appl. Phys. Lett. 99, 051110 (2011).
- [42] X. Sanshui , Z. Xiaolong , L. Bo-Hong, and N. Asger Mortensen, "Graphene-plasmon polaritons: From fundamental properties to potential applications," Front. Phys. 11, 117801 (2016).
- [43] G.Y. Slepyan, S. A. Maksimenko, A. Lakhtakia, O. Yevtushenko, and A. V. Gusakov, ".Electrodynamics of carbon nanotubes: Dynamic conductivity, impedance boundary conditions, and surface wave propagation," Physical Review B 60, 17136 (1999).
- [44] M. V. Shuba, A. G. Paddubskaya, A. O. Plyushch, P. P. Kuzhir, G. Ya. Slepyan, S. A. Maksimenko, V. K. Ksenevich, P. Buka, D. Seliuta, I. Kasalynas, J. Macutkevicius, G. Valusis, C. Thomsen, and A. Lakhtakia, ".Experimental evidence of localized plasmon resonance in composite materials containing single-wall carbon nanotubes," Phys. Rev. B 85, 165435 (2012).
- [45] C. A. Pfeiffer, E. N. Economou, and K. L. Ngai, "Surface polaritons in a circularly cylindrical interface: surface plasmons" Phys. Rev. B 10, 3038 (1974).
- [46] S. Ursula and D. Alain, "Surface plasmon polaritons on metal cylinders with dielectric core," Phys. Rev. B 64, 125420 (2001).
- [47] L. Novotny and C. Hafner, "Light propagation in a cylindrical waveguide with a complex, metallic, dielectric function," Phys. Rev. E 50, 4094 (1994).

- [48] M. Foygel, R. D. Morris, D. Anez, S. French, and V.L. Sobolev, "Theoretical and computational studies of carbon nanotube composites and suspensions:Electrical and thermal conductivity,"*Phys. Rev. B* 71, 104201.1–104201.8. (2005).
- [49] G. Heng, W. Jiaojiao and Y. Choongho, "Three-dimensional modeling of percolation behavior of electrical conductivity in segregated network polymer nanocomposites using Monte Carlo method,,"*Advances in Materials* 5, 1–8 (2016).
- [50] H. Ning, M. Zen, Y. Cheng and Y. Go, "The electrical properties of polymer nanocomposites with carbon nanotube fillers,"*Nanotechnology* 19, 21571 (2008).
- [51] J. N. Coleman, S. Curran, A.B. Dalton, A.P. Davey, B. McCarthy, W. Blau, and R.C. Barklie, "Percolation-dominated conductivity in a conjugated-polymer-carbon-nanotube composite,"*Phys. Rev.B* 58, 7462-7495 (1998).
- [52] J.D. Jackson, *Classical Electrodynamics*, "John Willey and Sons, (1998).
- [53] G.N. Afanasiev, *Cherenkov Radiation in a Dispersive Medium, Vavilov-Cherenkov and Synchrotron Radiation*, "Kluwer Academic Publishers, (2004).
- [54] C. Luo, M. Ibanescu, S.G. Johnson, and J.D. Joannopoulos, "Cherenkov radiation in photonic crystals,"*Science* 299, 368–371 (2003).
- [55] G. Chang, L.J. Chen, and F.X. Kurtner, "Highly efficient Cherenkov radiation in photonic crystal fibers for broadband visible wavelength generation", *Optics Letters* 35, 2361–2363 (2010).
- [56] G. Burlak, "Spectrum of Cherenkov radiation in dispersive metamaterials with negative refraction index,"*PIER* 132, 149–158 (2012).
- [57] G. Burlak, and E. Martínez-Sánchez, "Change of structure of the Cherenkov emission at modulated source in dispersive metamaterials,"*PIER* 139, 277–288 (2013).
- [58] S. Xi, H. Chen, T. Jiang, L. Ran, J. Huangfu, B.L. Wu, J.A. Kong, and M. Chen, "Experimental verification of reversed Cherenkov radiation in left-handed metamaterial,"*Phys. Rev. Lett.* 103, 194801 (2009).
- [59] F.J. García de Abajo, A.G. Pattantyus-Abraham, N. Zabala, A. Rivacoba, M.O. Wolf, and P.M. Echenique, "Cherenkov effect as a probe of photonic nanostructures,"*Phys. Rev. Lett.* 91, 143902 (2003).
- [60] G. Burlak, E. Martinez-Sanchez, "The Cherenkov Emission in Regular and Random Photonic Crystals,"*PIER M* 47, 77–86 (2016).
- [61] K.A. Nicholas, W. Alvin, T. AW, "Principles of Plasma Physics,"*McGraw-Hill Inc.,US*; (1973).

- [62] A. Taflove, S.C. Hagness, Computational Electrodynamics: The Finite-Difference Time-Domain Method, Artech House, Boston, (2005).
- [63] J.D. Joannopoulos, S.G. Johnson, J.N. Winn, and R.D. Meade, "Photonic Crystals Molding the Flow of Light," Princeton University Press, (2008).
- [64] Ogasawara, T., Moon, S. Y., Inoue, Y., and Shimamura, Y. (2011). Mechanical properties of aligned multi-walled carbon nanotube/epoxy composites processed using a hot-melt prepreg method. Composites science and technology, 71(16), 1826-1833.
- [65] McCullen, Seth D. and Stevens, Derrick R. and Roberts, Wesley A. and Ojha, Satyajeet S. and Clarke, Laura I. and Gorga, Russell E., "Morphological, Electrical, and Mechanical Characterization of Electrospun Nanofiber Mats Containing Multiwalled Carbon Nanotubes", Macromolecules, 40(4), 997-1003(2007).
- [66] F. D. Angelis, G. Das, P. Candeloro, M. Patrini, M. Galli, A. Bek, M. Lazzarino, I. Maksymov, C. Liberale, L. C. Andreani and E. D. Fabrizio, Nanoscale chemical mapping using three-dimensional adiabatic compression of surface plasmon polaritons, Nature Nanotechnology **5**, 67-72 (2010).
- [67] A.F. Aleksandrov, L. S. Bogdankevich, and A.A. Rukhadze, Principles of Plasma Electrodynamics, Springer-Verlag Berlin Heidelberg **40**, 552-553 (1985).
- [68] G. Burlak, G. Medina-Ángel, Structure of optical Cherenkov radiation in a three-dimensional compound medium with carbon nanotubes, Journal of the Optical Society of America B, **36**, 187-193 (2019).
- [69] Burlak, G., Cuevas-Arteaga, C., Medina-Ángel, G., Martínez-Sánchez, E., and Calderón-Segura, Y. Y. (2019). Plasmon-polariton oscillations in three-dimensional disordered nanotubes excited by a moving charge. Journal of Applied Physics, 126(1), 013101.
- [70] Reuven Y. Rubinstein, Dirk P. Kroese, *Simulation and the Monte Carlo Method*, Wiley Series in Probability and Statistics (Book 10), Wiley; 3 edition (2016).
- [71] N. Wataru, O. Motoichi and Y. Takashi, Nanodot coupler with a surface plasmon polariton condenser for optical far/near-field conversion, Appl. Phys. Lett. **86**, 181108 (2005).
- [72] Downing, C. A., Mariani, E., Weick, G., Retardation effects on the dispersion and propagation of plasmons in metallic nanoparticle chains, J. Phys.: Condens. Matter **30**, 025301 (2018)
- [73] Panagiotopoulos, P., Kolesik, M., Koch, S. W., Wright, E. M., Tochitsky, S. and Moloney, J. V. Control of the filament dynamics of 10 μm pulses via designer pulse trains. *JOSA B*, **36**(10), G33-G39.(2019)

- [74] Oripov, B., Bieler, T., Ciovati, G., Calatroni, S., Dhakal, P., Junginger, T., and Wilde, S. High-Frequency Nonlinear Response of Superconducting Cavity-Grade Nb Surfaces. *Physical Review Applied*, **11**(6), 064030.(2019)
- [75] Zhang, Y., Xu, Y., Tian, C., Xu, Q., Zhang, X., Li, Y., and Zhang, W. Terahertz spoof surface-plasmon-polariton subwavelength waveguide. *Photonics Research*, **6**(1), 18-23.(2018)
- [76] Sivarajah, P., Steinbacher, A., Dastrup, B., Lu, J., Xiang, M., Ren, W., and Nelson, K. A. THz-frequency magnon-phonon-polaritons in the collective strong-coupling regime. *Journal of Applied Physics*, **125**(21), 213103.(2019)
- [77] Dukhopelnykov, S. V., Sauleau, R., Garcia-Vigueras, M., and Nosich, A. I. Combined plasmon-resonance and photonic-jet effect in the THz wave scattering by dielectric rod decorated with graphene strip. *Journal of Applied Physics*, **126**(2), 023104.(2019)
- [78] Berrutti, P., Giaccone, B., Martinello, M., Grassellino, A., Khabiboulline, T., Doleans, M., and Ross, M. Plasma ignition and detection for in-situ cleaning of 1.3 GHz 9-cell cavities. *Journal of Applied Physics*, **126**(2), 023302.(2019)
- [79] Marquès, J. R., Lancia, L., Gangolf, T., Blecher, M., Bolaños, S., Fuchs, J., and Weber, S. Joule-level high-efficiency energy transfer to subpicosecond laser pulses by a plasma-based amplifier. *Physical Review X*, **9**(2), 021008.(2019)
- [80] De Angelis, F., Patrini, M., Das, G., Maksymov, I., Galli, M., Businaro, L., and Di Fabrizio, E. A hybrid plasmonic photonic nanodevice for label-free detection of a few molecules. *Nano letters*, **8**(8), 2321-2327.(2008)
- [81] Yelon, A., Piyakis, K. N., and Sacher, E. Surface plasmons in Drude metals. *Surface science*, **569**(1-3), 47-55.(2004)
- [82] Liu, S., Zhang, P., Liu, W., Gong, S., Zhong, R., Zhang, Y., and Hu, M. Surface polariton Cherenkov light radiation source. *Physical review letters*, **109**(15), 153902.(2012)
- [83] Magic Collaboration. Observation of pulsed γ -Rays above 25 GeV from the Crab Pulsar with MAGIC. *Science*, **322**(5905), 1221-1224.(2008)
- [84] McCullen, S. D., Stevens, D. R., Roberts, W. A., Ojha, S. S., Clarke, L. I., and Gorga, R. E. Morphological, electrical, and mechanical characterization of electrospun nanofiber mats containing multiwalled carbon nanotubes. *Macromolecules*, **40**(4), 997-1003.(2007)
- [85] Loos, J., Alexeev, A., Grossiord, N., Koning, C. E., and Regev, O. Visualization of single-wall carbon nanotube (SWNT) networks in conductive polystyrene nanocomposites by charge contrast imaging. *Ultramicroscopy*, **104**(2), 160-167.(2005)

- [86] Yuen, S. M., Ma, C. C. M., Chiang, C. L., and Teng, C.C. Morphology and properties of aminosilane grafted MWCNT/polyimide nanocomposites. *Journal of Nanomaterials*, **2008**.(2008)
- [87] Natarajan, B., Lachman, N., Lam, T., Jacobs, D., Long, C., Zhao, M., and Liddle, J. The evolution of carbon nanotube network structure in unidirectional nanocomposites resolved by quantitative electron tomography. *ACS nano*, **9**(6), 6050-6058.(2015)
- [88] Ma, H. Y., Tong, L. F., Xu, Z. B., and Fang, Z. P. Functionalizing carbon nanotubes by grafting on intumescent flame retardant: nanocomposite synthesis, morphology, rheology, and flammability. *Advanced functional materials*, **18**(3), 414-421.(2008)
- [89] Oliva-Avilés, A. I., Avilés, F., and Sosa, V. Electrical and piezoresistive properties of multi-walled carbon nanotube/polymer composite films aligned by an electric field. *Carbon*, **49**(9), 2989-2997.(2011)
- [90] Kashiwagi, T., Du, F., Winey, K. I., Groth, K. M., Shields, J. R., Bellayer, S. P., and Douglas, J. F. Flammability properties of polymer nanocomposites with single-walled carbon nanotubes: effects of nanotube dispersion and concentration. *Polymer*, **46**(2), 471-481.(2005)
- [91] Ogasawara, T., Moon, S. Y., Inoue, Y., and Shimamura, Y. Mechanical properties of aligned multi-walled carbon nanotube/epoxy composites processed using a hot-melt prepreg method. *Composites science and technology*, **71**(16), 1826-1833.(2011)
- [92] Liu, T., Phang, I. Y., Shen, L., Chow, S. Y., and Zhang, W. D. Morphology and mechanical properties of multiwalled carbon nanotubes reinforced nylon-6 composites. *Macromolecules*, **37**(19), 7214-7222.(2004)
- [93] Lin, L., Liu, S., Fu, S., Zhang, S., Deng, H., and Fu, Q. Fabrication of highly stretchable conductors via morphological control of carbon nanotube network. *Small*, **9**(21), 3620-3629.(2013)
- [94] Burlak, G., and Medina-Angel, G. The Optimization at Studying of Electrical Conductivity in the Dielectric Nanocomposites with Disordered Nanotubes. *Progress In Electromagnetics Research*, **74**, 77-82.(2018)
- [95] Burlak, G., and Medina-Ángel, G. Structure of optical Cherenkov radiation in a three-dimensional compound medium with carbon nanotubes. *JOSA B*, **36**(2), 187-193.(2019)
- [96] Burlak, G., Cuevas-Arteaga, C., Medina-Ángel, G., Martínez-Sánchez, E., and Calderón-Segura, Y.Y. Plasmon-polariton oscillations in three-dimensional disordered nanotubes excited by a moving charge. *Journal of Applied Physics*, **126**(1), 013101.(2019)

- [97] Macleod, A. J., Noble, A., and Jaroszynski, D. A. Cherenkov radiation from the quantum vacuum. *Physical review letters*, **122**(16), 161601.(2019)
- [98] Willis, K. J., Hagness, S. C., and Knezevic, I. Terahertz conductivity of doped silicon calculated using the ensemble Monte Carlo/finite-difference time-domain simulation technique. *Applied Physics Letters*, **96**(6), 062106.(2010)
- [99] R. del Coso, J. Requejo-Isidro, J. Solis, J. Gonzalo and C.N. Afonso, "Third order nonlinear optical susceptibility of Cu: Al₂O₃ nanocomposites: from spherical nanoparticles to the percolation threshold,"*J.Appl. Phys.* **95**, 2755-2762 (2004).
- [100] J. Li and J.K. Kim, "Percolation threshold of conducting polymer composites containing 3D randomly distributed graphite nanoplatelets,"*Compos. Sci. Technol.* **67**, 2114-2120 (2007). [3]
- [101] J.E.P. Monteagudo and P.L.C. Lage, "Cross-properties relations in 3D percolation networks: I. Network characteristic length determination,"*Transport Porous Med.* **61**, 143-156 (2005).
- [102] S. Nakaya and K. Nakamura, "Percolation conditions in fractured hard rocks: A numerical approach using the three-dimensional binary fractal fracture network (3D BFFN) model,"*J. Geophys.Res.* **112**, (2007).
- [103] S. Davis, P. Trapman, H. Leirs, M. Begon, and J. A. P. Heesterbeek, "The abundance threshold for plague as a critical percolation phenomenon,"*Nature* **454**, 634-637 (2008).
- [104] S. Miyazima, "A new percolation model with two threshold points,"*Progress Theoret. Phys.* **157**, 152-155 (2005).
- [105] K. Yamamoto, H. Yoshinaga and S. Miyazima, "Site and Bond Percolation Problem for Construction of Macroscopic Surface in a Cubic Lattice,"*Fractals* **17**, 131-135 (2009).
- [106] G. Burlak, M. Vlasova, P. Marquez-Aguilar, M. Kakazey and L. Xixitla-Cheron, "Optical percolation in ceramics assisted by porous clusters,"*Opt.Commun.* **282**, 2850-2856 (2009).
- [107] G. Burlak, A. Diaz-de-Anda, Yu. Karlovich and A.B. Klimov "Critical behavior of nanoemitter radiation in a percolation material,"*Phys. Lett. A.* **373**, 1492-1499 (2009).
- [108] V. Astratov and S. Ashili, "Percolation of light through whispering gallery modes in 3D lattices of coupled microspheres,"*Optics Express* **25**, 17351-17361 (2007).
- [109] M. Isichenko, "Percolation, statistical topography, and transport in random media,"*Rev. Mod. Phys.* **64**, 961-1043 (1992).

- [110] D. Stauffer, A. Aharony, *Introduction to Percolation Theory* Second ed. (CRC, 1992).
- [111] C. Domb, *Phase Transitions and Critical Phenomena* Volume **18** (Academic Press, 2000).
- [112] G. Grimmett, *Percolation*, second ed., (Springer Verlag, 1999).
- [113] S. M. Stishov, "Quantum phase transitions," *Phys. Usp.* **47**, 789 (2004).
- [114] J. S. Lundeen, B. Sutherland, A. Patel, C. Stewart and C. Bamber, "Direct measurement of the quantum wavefunction," *Nature* **474**, 188-191 (2011).
- [115] L. Li, Y. Yang, D. Zhang, Zuo-Guang Ye, S. Jesse, S.V. Kalinin and R. K. Vasudevan, "Machine learning-enabled identification of material phase transitions based on experimental data," *Sci. Adv.* **4**, (2018).
- [116] J. Urban, A. Menon, Z. Tian, A. Jain and K. Hippalgaonkar, "New horizons in thermoelectric materials: Correlated electrons, organic transport, machine learning, and more," *J. Appl. Phys.* **125**, 180902 (2019).
- [117] D. Vodenicarevic, N. Locatelli, J. Grollier and D. Querlioz, "Nano-oscillator-based classification with a machine learning-compatible architecture," *J. Appl. Phys.* **124**, 152117 (2018).
- [118] L. Piloizzi, F. Farrelly, G. Marcucci and C. Conti, "Machine learning inverse problem for topological photonics," *Comm. phys.* **1**, 1-7 (2018).
- [119] J. Carrasquilla and R. Melko, "Machine learning phases of matter," *Nature Phys* **13**, 431-434 (2017).
- [120] J. Salvail, M. Agnew, A. Johnson, E. Bolduc, J. Leach and R. Boyd, "Full characterization of polarization states of light via direct measurement," *Nature Photonics* **7**, 316-321 (2013).
- [121] M. Krenn, M. Malik, R. Fickler, R. Lapkiewicz and A. Zeilinger, "Automated search for new quantum experiments," *Phys. Rev. Lett.* **116**, 090405 (2016).
- [122] A.A. Melnikov, H.P. Nautrup, M. Krenn, V. Dunjko, M. Tiersch, A. Zeilinger and H. J. Briegel, "Active learning machine learns to create new quantum experiments," *PNAS* **115**, 1221-1226 (2018).
- [123] G. Burlak and Y. Calderón-Segura, "Percolation and lasing in real 3D crystals with inhomogeneous distributed random pores," *Physica B* **453**, 8-13 (2014).
- [124] K. Vogel and H. Risken, "Determination of quasiprobability distributions in terms of probability distributions," *Phys. Rev. A* **40**, 2847-2849 (1989).

- [125] D.T. Smithey, M. Beck, M. G. Raymer and A. Faridani, "Measurement of the Wigner distribution and the density matrix of a light mode using optical homodyne tomography," *Phys. Rev. Lett.* **70**, 1244-1247 (1993).
- [126] G. Breitenbach, S. Schiller and J. Mlynek, "Measurement of the quantum states of squeezed light," *Nature* **387**, 471-475 (1997).
- [127] A.G. White, D.F.V. James, P.H. Eberhard and P.G. Kwiat, "Nonmaximally entangled states," *Phys. Rev. Lett.* **83**, 3103-3107 (1999).
- [128] M. Hofheinz H. Wang, M. Ansmann, et al., "Synthesizing arbitrary quantum states in a superconducting resonator," *Nature* **459**, 546-549 (2009).
- [129] G. Steinbrecher, J. Olson, D. Englund and J. Carolan, "Quantum optical neural networks," *npj Quantum Inf* **5**, 60 (2019).
- [130] G. Burlak, "Evaluation of the spectrum of a quantum system using machine learning based on incomplete information about the wavefunctions," *Appl. Phys. Lett.* **116**, 024101 (2020).
- [131] G. Burlak and Y. G. Rubo, "Mirrorless lasing from light emitters in percolating clusters," *Phys. Rev. A* **92**, 013812 (2015).
- [132] S. Haykin, *Neural networks and learning machines*, 3rd ed. (Pearson Education, Inc. 2009).
- [133] P. Roy, S. Berger, P. Schmuki, *TiO₂ Nanotubes: Synthesis and Applications*, *Angew. Chem. Int. Ed.* **50**, 2904-2939 (2011).
- [134] I. Paramasivam, H. Jha, N. Liu, P. Schmuki, A review of photocatalysis using self-organized *TiO₂* nanotubes and other ordered oxide nanostructures, *Small* **8**, 3073-3103 (2012).
- [135] S. Kment, F. Riboni, S. Pausova, L. Wang, L. Wang, H. Han, Z. Hubicka, J. Krysa, P. Schmuki, R. Zboril, Photoanodes based on *TiO₂* and $\alpha - Fe_2O_3$ for solar water splitting-superior role of 1D nanoarchitectures and of combined heterostructures, *Chem. Soc. Rev.*
- [136] Goodfellow, I., Bengio, Y. and Courville, A. (2016). *Deep learning*. MIT press. 200-2020.
- [137] Nielsen, M. A. (2015). *Neural networks and deep learning* (Vol. 2018). San Francisco, CA, USA:: Determination press.
- [138] Medina, G., Calderón, Y. Y., Burlak, G., and Hernández, J. A. (2020). Study of the critical probability of percolation in a 3D system with pores of random radius for variable grids. *Revista Mexicana de Física*, 66(3 May-Jun), 315-321.