



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA



ESCUELA TÉCNICA  
SUPERIOR INGENIEROS  
INDUSTRIALES VALENCIA

**TRABAJO FIN DE GRADO EN INGENIERÍA BIOMÉDICA**

# **DESARROLLO DE ESTRUCTURAS NANOFOTÓNICAS DE SENSADO DE ALTA SENSIBILIDAD Y MAYOR TOLERANCIA A DESVIACIONES DE FABRICACIÓN**

AUTORA: Maza Pino, Anny Michelle

TUTOR: Martínez Abiétar, Alejandro José

COTUTOR: García Rupérez, Jaime

Curso Académico: 2016-17

# RESUMEN

La nanotecnología es un campo que se encuentra actualmente en crecimiento constante. Una de sus aplicaciones, en el ámbito de la medicina, es el desarrollo de biosensores. Dentro de esta variedad de dispositivos, están cobrando especial importancia los de tipo nanofotónico gracias a sus prometedores resultados. Estos, a través del diseño de estructuras (normalmente dieléctricas) y del estudio de la variación de las propiedades de la luz cuando interactúa con el medio externo, permiten la detección de analitos, como proteínas o ADN, con un elevado nivel de sensibilidad que permitirá detectar una mínima cantidad de muestra. Sin embargo, una sensibilidad muy elevada puede significar también un problema, ya que podrá provocar que dichas estructuras sean también muy poco tolerantes a pequeños errores dimensionales durante su fabricación. Estas pequeñas desviaciones es algo muy común debido a las dimensiones a escala nanométrica que presentan las estructuras. Con la finalidad de encontrar estructuras de sensado que desvinculen en la medida de los posible estos dos parámetros, en este Trabajo Fin de Grado se desarrolla un sistema automatizado que permita obtener las configuraciones óptimas de las estructuras en términos de sensibilidad-tolerancia, haciendo uso del cálculo de las propiedades de éstas en función de sus dimensiones. Los resultados obtenidos han demostrado el correcto funcionamiento del sistema desarrollado, permitiendo determinar los intervalos de dimensiones óptimos en los que dicha relación se cumple.

**Palabras clave:** biosensor, sensor fotónico, nanofotónica, cristal fotónico, guía corrugada, *Photonic Band Gap*, sensibilidad, tolerancia, sistema de optimización, sistema automatizado.

# RESUM

La nanotecnologia és un camp que es troba actualment en creixement constant. Una de les seues aplicacions, en l'àmbit de la medicina, és el desenvolupament de biosensors. Dins d'aquesta varietat de dispositius, estan cobrant especial importància els de tipus nanofotònic gràcies als seus prometedors resultats. Aquests, a través del disseny d'estructures (normalment dielèctriques) i de l'estudi de la variació de les propietats de la llum quan interactua amb el medi extern, permeten la detecció d'analits, com proteïnes o ADN, amb un elevat nivell de sensibilitat que permetrà detectar una mínima quantitat de mostra. No obstant això, una sensibilitat molt elevada pot significar també un problema, ja que podrà provocar que aquestes estructures siguin també molt poc tolerants a xicotets errors dimensionals durant la seua fabricació. Aquestes xicotetes desviacions és una cosa molt comú a causa de les dimensions a escala nanomètrica que presenten les estructures. Amb la finalitat de trobar estructures de sensat que desvinculen en la mesura dels possible aquests dos paràmetres, en aquest Treball Fi de Grau es desenvolupa un sistema automatitzat que permeta obtenir les configuracions òptimes de les estructures en termes de sensibilitat-tolerància, fent ús del càlcul de les propietats d'aquestes en funció de les seues dimensions. Els resultats obtinguts han demostrat el correcte funcionament del sistema desenvolupat, permetent determinar els intervals de dimensions òptims en què aquesta relació es compleix.

**Paraules clau:** biosensor, sensor fotònic, nanofotònica, cristall fotònic, guia corrugada, *Photonic Band Gap*, sensibilitat, tolerància, sistema d'optimització, sistema automatitzat.

# SUMMARY

Nanotechnology is a field that is currently growing steadily. One of its applications, in the field of medicine, is the development of biosensors. Within this variety of devices, the nanophotonic type is gaining special importance thanks to their promising results. These, through the design of structures (usually dielectric) and the study of the variation of the properties of light when interacting with the external environment, allow the detection of analytes, such as proteins or DNA, with a high level of sensitivity that will allow detect a minimum amount of sample. However, a very high sensitivity may also mean a problem, since it may cause such structures to be also very little tolerant of small dimensional errors during their manufacture. These small deviations are very common due to the nanometric scale dimensions of the structures. To find structures of sensibility that separate as far as possible these two parameters, in this End-of-Grade Work an automated system is developed that allows to obtain the optimal configurations of the structures in terms of sensitivity-tolerance, making use of the calculation of the properties of these according to their dimensions. The obtained results have demonstrated the correct operation of the developed system, allowing to determine the intervals of optimal dimensions in which this relation is fulfilled.

**Keywords:** biosensor, photonic sensor, nanophotonic, photonic crystal, corrugated guide, Photonic Band Gap, sensitivity, tolerance, optimization system, automated system.

# CONTENIDOS

MEMORIA

PRESUPUESTO

ANEXOS

## MEMORIA

1.	INTRODUCCIÓN .....	1
1.1	Presentación del trabajo .....	1
1.2	Objetivos.....	2
1.2.1	Justificación y objetivos generales .....	2
1.2.2	Objetivos específicos .....	3
1.3	Materiales y métodos .....	4
2.	ANTECEDENTES .....	5
2.1	¿Qué son las estructuras nanofotónicas? ¿Y los cristales fotónicos? .....	5
2.2	Tipos de cristales fotónicos .....	6
2.3	Fundamentos teóricos de los cristales fotónicos.....	7
2.3.1	Teoría electromagnética: Ecuaciones de Maxwell .....	7
2.3.2	Física de estado sólido: simetrías.....	9
2.3.3	Photonic band gap (PBG).....	13
2.3.4	Sensado a partir de PBG .....	14
2.4	Aplicaciones en el ámbito biomédico .....	15
2.5	Estructuras fotónicas empleadas.....	16
2.5.1	Guía corrugada (corr3D) .....	18
2.5.2	Guía corrugada inversa (corr3Dinv) .....	19
3.	DESARROLLO DEL SISTEMA DE OPTIMIZACIÓN.....	20
3.1	Sistema automatizado general .....	21
3.2	Simulación MPB .....	24
3.2.1	Procesamiento de archivos .ctl y del script de simulación múltiple .....	24
3.2.2	Obtención de archivos .out , .fre , .ypar y .zpar .....	25
3.3	Detección de los PBGs .....	26
3.3.1	Obtención del diagrama de bandas y búsqueda de patrones del PBG .....	26

3.3.2	Búsqueda y extracción de anomalías presentes en cada PBG.....	29
3.3.3	Almacenamiento de datos (configuraciones, PBGs, anomalías...)	32
3.4	Representación de PBG.....	34
3.4.1	Extracción y representación de PBGs situados en $\lambda=1550$ nm .....	34
3.4.2	Representación de anomalías en PBG.....	41
3.5	Obtención de la sensibilidad.....	45
3.5.1	Cálculo y representación para todas las configuraciones.....	45
3.5.2	Representación para el plano de PBG en $\lambda=1550$ nm .....	48
3.6	Obtención de la tolerancia .....	50
3.6.1	Cálculo y representación para todas las configuraciones.....	50
3.6.2	Representación para el plano de PBG en $\lambda=1550$ nm .....	55
3.7	Funciones y representaciones de apoyo .....	58
3.7.1	Plano 3D de PBG en $\lambda=1550$ nm, mapas de color y líneas de contorno .....	58
4.	ANÁLISIS DE RESULTADOS.....	61
4.1	Guía corrugada.....	61
4.2	Guía corrugada inversa.....	63
5.	CONCLUSIONES.....	68
6.	POSIBLES LÍNEAS FUTURAS .....	69
7.	REFERENCIAS .....	70

## PRESUPUESTO

1.	INTRODUCCIÓN .....	73
2.	PRESUPUESTO .....	73
2.1	Presupuesto parcial.....	73
2.2	Presupuesto total.....	75
3.	REFERENCIAS .....	75

## ANEXOS

1.	CÓDIGO MATLAB DE LAS FUNCIONES DESARROLLADAS .....	78
1.1	Sistema automatizado general .....	78
1.1.1	funcion_general.m .....	78

1.2	Simulación MPB .....	80
1.2.1	filescreator3D.m.....	80
1.2.2	simulaciones_script_creator3D.m .....	82
1.3	Detección de PBG.....	84
1.3.1	dibbandasMPB_3Dslab_select_parity_wavelength.m.....	84
1.3.2	see_all_graphics.m.....	87
1.3.3	find_BG_3D.m.....	88
1.3.4	creator_matrizBG_3D.m.....	91
1.4	Representación PBG.....	93
1.4.1	creator_mapacolor_3D_rectas.m .....	93
1.4.2	creator_mapacolor_3D_plano1550.m .....	96
1.4.3	creator_mapacolor_3D_vol_dist.m .....	100
1.5	Obtención de sensibilidad y tolerancia .....	106
1.5.1	creator_vol_sensibilidad.m.....	106
1.5.2	creator_vol_tolerancia.m .....	108
1.6	Representación sensibilidad para PBG en $\lambda=1550$ nm .....	113
1.6.1	creator_mapacolor_4D_sensibilidad.m .....	113
1.7	Representación tolerancia para PBG en $\lambda=1550$ nm .....	116
1.7.1	creator_mapacolor_4D_tolerancia.m .....	116
1.8	Funciones de apoyo .....	119
1.8.1	interseccion_mapacolor_plano1550.m.....	119
2.	ARCHIVOS MODELO UTILIZADOS.....	122
2.1	corr3D-a380-wi100-we1500-h220-upper_DIW-w450-5bandas.ctl .....	122
2.2	simulaciones_scriptCorr3D.....	124

# MEMORIA

# 1. INTRODUCCIÓN

## 1.1 Presentación del trabajo

Los grandes avances de la ciencia en los últimos tiempos han impulsado la generación de nuevo conocimiento valioso en una gran diversidad de campos. Uno de los que ha sufrido un mayor crecimiento es la nanociencia cuya aplicación práctica tiene como resultado la nanotecnología. Esta rama de las ciencias aplicadas se encarga de estudiar, diseñar y manipular estructuras, materiales y sistemas a escala nanométrica ( $1nm = 10^{-9}m$ ) (Martí & Martínez, 2016).

Centrándose en un ámbito más concreto se encuentra la nanomedicina, cuya aplicación en el diagnóstico y terapia de enfermedades se encuentra en auge actualmente. Respecto al diagnóstico, los biosensores ofrecen la posibilidad de detectar la presencia de un analito (como una proteína o ADN) localizado en proporciones muy bajas gracias al diminuto tamaño que presentan y a la alta sensibilidad que puede proporcionar. Un tipo de nanosensor son los nanofotónicos. Estos hacen uso de las propiedades que presenta la luz cuando se emite a través de estructuras (normalmente dieléctricas) diseñadas a escala nanométrica (Luis Motellón & Bueren, 2010).

Así, una de las ventajas que presentan las estructuras nanofotónicas como biosensores es una mayor sensibilidad de detección haciendo uso de una cantidad de muestra muy inferior si se compara con dispositivos convencionales. Sin embargo, el desarrollo de sensores a esta escala también debe hacer frente a algunos problemas como, por ejemplo, las desviaciones que se producen en la fabricación. La desviación en las dimensiones geométricas de una nanoestructura puede afectar considerablemente a sus prestaciones por lo que es necesario reducir estos errores al mínimo posible. El principal problema que aparece al intentar reducirlo es que la sensibilidad y la tolerancia, por regla general, están inversamente correlacionadas, es decir, un aumento de la sensibilidad del sensor normalmente se traduce en una disminución de la tolerancia en la fabricación. Para intentar solucionar este conflicto, surge el planteamiento de este Trabajo Fin de Grado: el desarrollo de un sistema que permita diseñar estructuras nanofotónicas que presenten una alta sensibilidad de sensado y, a la vez, una mayor tolerancia a desviaciones de fabricación a través de la búsqueda de aquellas combinaciones de dimensiones estructurales más adecuadas.

El Trabajo de Fin de Grado (TFG) contiene varios apartados diferenciados en Memoria, Presupuestos y Anexos. En el apartado de Memoria se presentará, en primer lugar, el TFG en sí, incluyendo el planteamiento de los objetivos que se desean cumplir (apartado 1.2 de la Memoria) y los materiales y métodos necesarios para llevarlos a cabo (apartado 1.3 de la Memoria). A continuación, se explican los conceptos teóricos básicos necesarios sobre nanoestructuras y el comportamiento de la luz en ellas necesarios para el correcto entendimiento del resto del trabajo (apartado 2 de la Memoria). Seguidamente, se expondrá el desarrollo del sistema creado (apartado 3 de la Memoria), así como los resultados obtenidos con él (apartado 4 de la Memoria). Por último, se presentarán las conclusiones (apartado 5 de la Memoria) y las posibles líneas futuras y mejoras del proyecto (apartado 6 de la Memoria). Por otro lado, se añade, en el apartado de Presupuestos, los costos que conlleva el desarrollo del

sistema (apartado 2 de Presupuestos) y, en el apartado de Anexos, los códigos de las funciones (diseñadas en Matlab) que forman dicho sistema.

Finalmente cabe destacar que el TFG se ha llevado a cabo en el Centro de Tecnología Nanofotónica de Valencia (NTC). Este Instituto Universitario de Investigación posee una gran variedad de instalaciones y equipos de micro-nano fabricación (e.g. sala limpia, SEM y FTIR) que lo hace posicionarse como uno de los principales centros de investigación de la Universidad Politécnica de Valencia (UPV). Además, posee varias líneas de investigación a nivel europeo como PHOCNOSIS (desarrollo y validación de un sistema de análisis de diagnóstico inmediato para la detección precoz de enfermedades cardiovasculares) o SAPHELY (desarrollo y validación de un dispositivo capaz de realizar un diagnóstico temprano y mínimamente invasivo de enfermedades como el cáncer) de especial interés en el ámbito biomédico (Centro de Tecnología Nanofotónica de Valencia, s. f.).

## 1.2 Objetivos

### 1.2.1 Justificación y objetivos generales

Como se mencionará en el apartado siguiente, dentro de las estructuras fotónicas se encuentran los cristales fotónicos que presentan una gran versatilidad respecto a sus posibles aplicaciones. Cuando se utilizan como sensores, uno de los requisitos imprescindibles que deben cumplir es que sean capaces de detectar con mucha precisión cambios en su entorno. Es decir, que sean dispositivos con una sensibilidad de respuesta a variaciones muy elevada respecto a una respuesta modelo. Sin embargo, trabajar con dispositivos a escala micro-nanométrica supone también algunos problemas sobre todo durante los procesos de producción. Al tratarse de estructuras tan minúsculas cualquier variación de sus dimensiones (respecto al diseño) durante la fabricación puede suponer cambios importantes en las prestaciones del cristal fotónico. Es necesario, por tanto, que los modelos diseñados sean lo más tolerantes posible a las variaciones que pueden sufrir los parámetros de la estructura durante el proceso de fabricación.

Así, cada vez se requieren dispositivos que presenten una mayor sensibilidad a variaciones de entorno y a la vez una mayor tolerancia a posibles desviaciones de fabricación. No obstante, estos dos parámetros son difíciles de combinar. Una mayor sensibilidad de detección tendrá como consecuencia una mayor pérdida de prestaciones si se producen cambios en la geometría del dispositivo. Es decir, a mayor sensibilidad menor tolerancia.

Por todo lo dicho anteriormente, en este trabajo se pretende analizar y comprender el comportamiento de la luz en diferentes estructuras. La finalidad de dicho estudio será poder extraer parámetros (de manera automatizada) a partir de los cuales poder determinar aquellas configuraciones más adecuadas con relación al binomio sensibilidad-tolerancia.

### 1.2.2 Objetivos específicos

Los objetivos específicos del trabajo son los siguientes:

-Respecto al estudio del comportamiento de la luz cuando interactúa con cristales fotónicos:

- Comprender teóricamente cómo y porqué suceden ciertos fenómenos en los cristales fotónicos
- Búsqueda de patrones en dichos fenómenos para las mismas estructuras variando sus dimensiones (periodicidad, anchura de la guía, altura...)

-Respecto a la extracción de parámetros:

- Detección del *Photonic Band Gap* (PBG) para todas las combinaciones de configuraciones posibles de las diferentes estructuras
- Detección de las configuraciones que pueden presentar anomalías en su respuesta por la aparición del PBG bajo el cono de luz, por la inexistencia de PBG y por la presencia de longitudes de onda que sí se propagan dentro del PBG.
- Obtención y representación de aquellas configuraciones para las cuales el PBG se presenta en  $\lambda = 1550 \text{ nm}$
- Detección y representación de posibles anomalías para las configuraciones donde el PBG se encuentra en  $\lambda = 1550 \text{ nm}$

-Respecto al análisis de sensibilidad-tolerancia:

- Cálculo y representación de la sensibilidad de cada cristal fotónico bajo diferentes condiciones de contorno (con agua o con óxido de silicio por encima) para todas las posibles configuraciones
- Cálculo y representación de la tolerancia de cada cristal fotónico respecto a variaciones en las dimensiones de los parámetros de diseño para todas las posibles configuraciones
- Cálculo y representación de la sensibilidad de cada cristal fotónico bajo diferentes condiciones de contorno (con agua o con óxido de silicio por encima) para aquellas configuraciones donde el PBG se encuentra en  $\lambda = 1550 \text{ nm}$
- Cálculo y representación de la tolerancia de cada cristal fotónico respecto a variaciones en las dimensiones de los parámetros de diseño para aquellas configuraciones donde el PBG se encuentra en  $\lambda = 1550 \text{ nm}$
- Representar para cada cristal fotónico las posibles anomalías y la sensibilidad de las configuraciones donde el PBG se encuentra en  $\lambda = 1550 \text{ nm}$
- Representar para cada cristal fotónico las posibles anomalías y la tolerancia de las configuraciones donde el PBG se encuentra en  $\lambda = 1550 \text{ nm}$

-Otros objetivos:

- Automatización de todos los procesos anteriormente mencionados para facilitar y optimizar el estudio

- Comprensión de la herramienta software de simulación MIT Photonic-Bands (MPB)
- Simulación de las diferentes configuraciones de parámetros para las estructuras mediante MPB
- Automatización de la creación de scripts para simulación con MPB

### 1.3 Materiales y métodos

Para obtener aquellas estructuras que presentan las configuraciones más adecuadas respecto a la relación sensibilidad-tolerancia (objetivo principal del trabajo) la metodología empleada durante el desarrollo del sistema es la siguiente: en primer lugar, se realiza la simulación de las estructuras (con diferentes configuraciones) a través del servidor MPB. A partir de los datos obtenidos de estas se extrae la localización de los PBGs (en valores de longitud de onda), las configuraciones que los presentan y la presencia de posibles anomalías en estos. Dichos valores se almacenan en una matriz a partir de la cual se obtienen y representan aquellos PBGs presentes únicamente en  $\lambda=1550$  nm. Por otro lado, se realiza el cálculo de la sensibilidad y tolerancia para todas las posibles combinaciones donde las estructuras presentan PBG y, a partir de estas, se extraen únicamente aquellos PBG localizados en  $\lambda=1550$  nm. A partir de la comparación de estas últimas ilustraciones se obtienen los rangos de configuraciones donde la relación sensibilidad-tolerancia es más adecuada y, además, se propone la mejor combinación dentro de dicho rango de valores.

Se dispone de varias estructuras fotónicas diferentes sobre las que trabajar (explicadas en el apartado 2.5) con diferentes grados de dificultad. Sin embargo, para evitar repeticiones en la explicación y facilitar la lectura y el diseño de las funciones se decide emplear la más completa para el desarrollo del sistema (explicado en el apartado 3) que es la guía corrugada (corr3D). Se elige dicha estructura por ser la más genérica, pudiendo, tras realizar pequeños cambios en el código, utilizar este mismo sistema para la otra estructura estudiada (guía corrugada inversa 'corr3Dinv'). Esta última se empleará, por tanto, a modo de ejemplo del funcionamiento del sistema.

Por otro lado, las funciones se encuentran en los Anexos del TFG para que puedan ser consultadas en cualquier momento de manera sencilla. Además, para facilitar su comprensión, se ha añadido resumidas explicaciones del funcionamiento de cada una de ellas a modo de comentario.

Respecto a los materiales, se ha empleado fundamentalmente cuatro herramientas software:

- MPB (MIT Photonic-Bands): utilizado para realizar las simulaciones de las diferentes configuraciones elegidas para las estructuras fotónicas. El software proporciona datos de las propiedades y del comportamiento de los cristales fotónicos necesarios para su posterior análisis en MATLAB. Se utilizará a través de un servidor de manera remota.
- SSH Secure Shell y SSH Secure File Transfer Client: Ambos utilizan el protocolo Secure Shell (SSH) para permitir el acceso de forma remota a un servidor. En este

caso, se emplea para la conexión con el servidor MPB del NTC localizado en un ordenador de altas prestaciones en sus instalaciones en UPV.

- MayaVi: permite la visualización de datos científicos. Una vez se realizan las simulaciones con el servidor de MPB se puede emplear este programa para comprobar que las configuraciones simuladas son las correctas.
- MATLAB: empleado en el desarrollo de las funciones necesarias para conseguir los objetivos enumerados en la clasificación “Respecto a la extracción de parámetros”, “Respecto al análisis de sensibilidad-tolerancia” y los relacionados con la automatización de procesos del apartado 1.2.2.

## 2. ANTECEDENTES

### 2.1 ¿Qué son las estructuras nanofotónicas? ¿Y los cristales fotónicos?

Desde su nacimiento, el interés en la nanofotónica ha ido en aumento. La posibilidad de poder crear estructuras ópticas a escala nanométrica que posean las mismas o incluso mejores características que los dispositivos de dimensiones tradicionales ha abierto un gran abanico de posibilidades tanto en investigación como en la industria. Así, a través del estudio de la propagación de la luz en la materia y del diseño de nuevas configuraciones (para buscar la optimización de sus prestaciones) surgen las denominadas estructuras nanofotónicas.

Las estructuras nanofotónicas son, por tanto, los diferentes diseños posibles en los que la luz interacciona de manera óptima con la materia. Actualmente existe una gran variedad de estas estructuras como pueden ser los cristales fotónicos, anillos resonantes o interferómetros, por ejemplo. Estos, forman la base a partir de la cual se pueden diseñar una gran variedad de dispositivos como sensores biofotónicos, acopladores o multiplexores (Centro de Tecnología Nanofotónica de Valencia, s. f.).

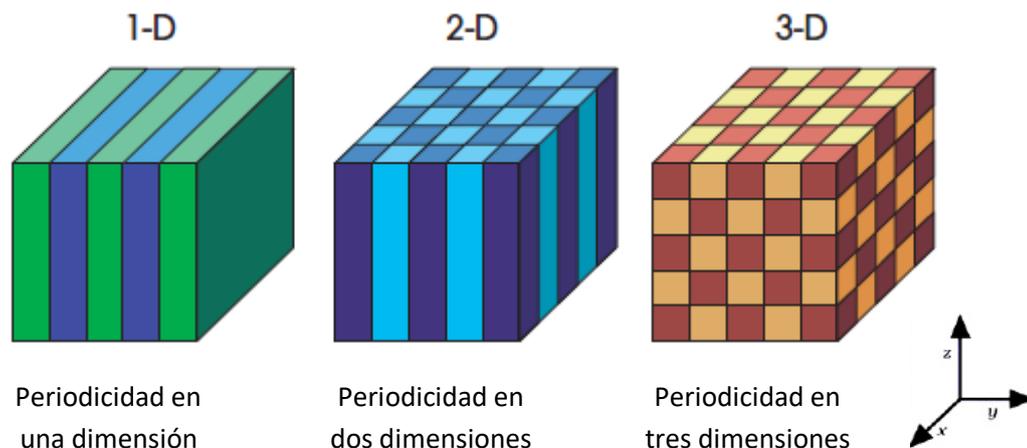
De entre todos los diferentes tipos de estructuras fotónicas existentes, este TFG se centrará en el análisis de los cristales fotónicos. Estos se forman mediante la creación de configuraciones que contienen diferentes materiales dieléctricos dispuestos en el espacio de manera periódica. Gracias a esta, se obtiene una modulación periódica del índice de refracción que verá la luz que incide en el cristal. La característica principal que hace destacar estas estructuras frente a otras es la aparición de lo que se denomina *Photonic Band Gap*. Este fenómeno “prohibirá” la transmisión de luz por el material para determinadas longitudes de onda. Los PBG (de los cuales se hablará de manera más extensa en el apartado 2.3.3) aparecerán cuando, entre otras características, las constantes dieléctricas que forman la red son lo suficientemente diferentes y, además, el material presenta un bajo índice de absorción de luz. Es decir, debido a sus propiedades ópticas, la utilización de cristales fotónicos permite controlar y diseñar por dónde se propagarán los modos de luz (Joannopoulos, Johnson, Winn, & Meade, 2008; Sukhoivanov & Guryev, 2009). Esta característica permitirá el diseño de una gran cantidad de cristales fotónicos en función de la aplicación que se desee.

Sin embargo, para poder explicar más en profundidad tanto el concepto de PBG como el resto de propiedades de los cristales fotónicos (que los hacen tan versátiles para el diseño de dispositivos en la actualidad) es necesario aclarar ciertos conceptos previamente.

## 2.2 Tipos de cristales fotónicos

En función de la periodicidad de los materiales dieléctricos que forman los cristales fotónicos y, por tanto, de su geometría estructural se pueden clasificar estas estructuras en tres categorías: unidimensionales (1D), bidimensionales (2D) y tridimensionales (3D) (Joannopoulos et al., 2008).

Como se puede observar en la Figura 1 en la imagen de la izquierda, los cristales fotónicos unidimensionales son los más sencillos puesto que únicamente están formados por dos materiales con diferente constante dieléctrica dispuestos de manera alterna. Es decir, en un eje se crea la periodicidad de las capas alternadas mientras que en los otros dos ejes dichas capas se presentan uniformes. En las estructuras bidimensionales (Figura 1 centro) la periodicidad se repite en dos ejes mientras que en el tercero se mantiene constante. Cuando los cambios en el índice de refracción se dan respecto a los tres ejes de coordenadas entonces denominamos a estas estructuras como tridimensionales (Figura 1 derecha). Se supone que la estructura se extiende al infinito en las tres direcciones espaciales para los tres tipos de cristales (Bellingeri, Chiasera, Kriegel, & Scotognella, 2017).



**Figura 1:** Representación de ejemplos de diferentes tipos de cristales fotónicos: unidimensionales (izquierda), bidimensionales (centro) y tridimensionales (derecha). La diferencia radica en la cantidad de ejes de coordenadas en los que se produce la periodicidad de los diferentes materiales dieléctricos. Los materiales con diferentes constantes dieléctricas se representan con colores distintos (Joannopoulos et al., 2008)

Debido a la mayor complejidad de estudio, optimización y fabricación que presentan las estructuras 2D y 3D, en este TFG únicamente se van a desarrollar los procesos de optimización para estructuras con periodicidad 1D. Sin embargo, es necesario tener presente que muchos de los conceptos y los resultados que se van a desarrollar, siempre y cuando se adapten a las dimensiones de periodicidad indicadas, podrán ser extrapolados para 2D y 3D.

## 2.3 Fundamentos teóricos de los cristales fotónicos

Antes de desarrollar este apartado cabe mencionar que la información utilizada en este apartado se ha extraído de los libros *Photonic crystals: Molding the flow of light* ((Joannopoulos et al., 2008)) y *Photonic crystals. Physics and Practical Modeling* (Sukhoivanov & Guryev, 2009) y, además, de los apuntes de la asignatura “Micro y Nano Tecnología” impartida en cuarto curso de Ingeniería Biomédica (Llorente Saez, 2017). Por otro lado, los fundamentos teóricos que engloban a los cristales fotónicos son muy extensos y relativamente complejos. Por este motivo, únicamente se realizará una visión general de los conceptos básicos necesarios para poder comprender el desarrollo del trabajo (localizado en el apartado 3).

Respecto a los cristales fotónicos, para poder entender cómo interactúa la luz con ellos es necesario tener presente tanto la Teoría Electromagnética, basada en las Ecuaciones de Maxwell, como la física de estado sólido. Las estructuras, por un lado, se pueden estudiar desde el punto de vista de cristales ordenados de manera periódica mientras que, por otro lado, los modos de luz (que interactúan con estos cristales) se interpretan como ondas electromagnéticas.

### 2.3.1 Teoría electromagnética: Ecuaciones de Maxwell

El comportamiento de la luz, cuando interactúa con una estructura de cristal fotónico se puede explicar a través de las Ecuaciones de Maxwell:

$$\begin{aligned}\nabla \cdot B &= 0 & \nabla \times E + \frac{\partial B}{\partial t} &= 0 \\ \nabla \cdot D &= \rho & \nabla \times H - \frac{\partial D}{\partial t} &= J\end{aligned}$$

**Ecuación 1:** Ecuaciones de Maxwell. Ley de Gauss para el campo magnético (arriba, izquierda), Ley de Faraday (arriba, derecha), Ley de Gauss aplicada en la materia (abajo, izquierda) y Ley de Ampère generalizada aplicada a la materia (abajo, derecha).  $B$  representa la inducción magnética,  $E$  el campo eléctrico,  $D$  el desplazamiento eléctrico,  $H$  el campo magnético,  $J$  las corrientes libres y  $\rho$  la densidad de cargas. Estas dos últimas son nulas en el caso de los cristales fotónicos.

Para hacer más sencillo el problema de hallar una solución aproximada a estas ecuaciones se imponen ciertas condiciones. En primer lugar, el régimen será lineal puesto que los campos se consideran pequeños. En segundo lugar, el material utilizado será macroscópico e isotrópico para poder relacionar  $E(r,\omega)$  y  $D(r,\omega)$  mediante  $\epsilon(r,\omega)$  (constante dieléctrica escalar). En tercer lugar, dicha constante será independiente de la frecuencia de manera que su valor será el de la constante dieléctrica del rango de frecuencias donde se encuentre. En último lugar, únicamente se analizará la parte real de  $\epsilon(r)$  puesto que se presupone que las pérdidas de los materiales dieléctricos serán bajas.

Por otro lado, gracias a la primera condición establecida en el párrafo anterior (linealidad de las ecuaciones de Maxwell), se podrá representar como un conjunto de armónicos la dependencia temporal que presentan los campos eléctrico y magnético (su dependencia espacial se mantiene). Dicho conjunto de armónicos recibe el nombre de modos. Tomando esto en cuenta, las soluciones aproximadas de las ecuaciones presentes en Ecuación 1 se podrán

obtener analizando las restricciones que éstas imponen a las variaciones de los modos de un campo con el tiempo. Así, los modos armónicos se pueden expresar como:

$$\mathbf{H}(\mathbf{r}, t) = \mathbf{H}(\mathbf{r})e^{j\omega t} \qquad \mathbf{E}(\mathbf{r}, t) = \mathbf{E}(\mathbf{r})e^{j\omega t}$$

**Ecuación 2:** Ecuación del modo armónico del campo magnético (izquierda) y del campo eléctrico (derecha).

siendo los valores de  $H(\mathbf{r})$  y  $E(\mathbf{r})$  en las ecuaciones anteriores unos patrones de campo multiplicados por una exponencial compleja.

A partir de la sustitución de Ecuación 2 en Ecuación 1, teniendo en cuenta las restricciones antes mencionadas y realizando una serie de operaciones adicionales, se obtiene la ‘ecuación maestra’ (Ecuación 3). Ésta solamente depende de  $H(\mathbf{r})$  por lo que se puede determinar el valor de sus modos. Sin embargo, también son de interés los modos  $E(\mathbf{r})$  por lo que en primer lugar se calculan los modos del campo magnético con la ecuación maestra y con estos resultados se obtienen los modos del campo eléctrico.

$$\nabla \times \left( \frac{1}{\epsilon(\mathbf{r})} \nabla \times \mathbf{H}(\mathbf{r}) \right) = \left( \frac{\omega}{c} \right)^2 \mathbf{H}(\mathbf{r})$$

**Ecuación 3:** Ecuación maestra a partir de la cual se puede obtener los modos de  $H(\mathbf{r})$ .

La ecuación maestra también permite distinguir los patrones de campo que pertenecen al mismo modo a través del denominado ‘problema de valor propio’. Para facilitar su explicación se redefine la Ecuación 3 como:

$$\Phi \mathbf{H}(\mathbf{r}) = \left( \frac{\omega}{c} \right)^2 \mathbf{H}(\mathbf{r})$$

**Ecuación 4:** Ecuación maestra representada como un problema de valor propio.

donde  $\Phi$  representa un operador. Así, basándose en la ecuación anterior, se realiza la operación indicada por  $\Phi$  sobre el modo  $H(\mathbf{r})$ . Si éste es válido, es decir, si realmente es un modo magnético que cumple las ecuaciones de Maxwell, el resultado de la ecuación será la función  $H(\mathbf{r})$  original multiplicada por un valor constante (número de veces que se repite dicha función). Este fenómeno se denomina ‘problema de valor propio’. En estos casos,  $H(\mathbf{r})$  se conoce como función propia o vector propio de  $\Phi$  y el valor constante recibe el nombre de valor propio.

Por otro lado, se conoce que  $\Phi$  presenta hermiticidad. Esta propiedad ofrece la posibilidad de distinguir los modos en función de su frecuencia. Dos modos normalizados  $H_1(\mathbf{r})$  y  $H_2(\mathbf{r})$  serán ortogonales si presentan frecuencias  $\omega_1$  y  $\omega_2$  diferentes. Si, por el contrario, presentan la misma frecuencia ( $\omega_1 = \omega_2$ ) serán modos degenerados que podrán ser o no ortogonales. Para poder obtener dichos modos degenerados se forzará la aparición de simetrías en las estructuras (de las cuales se hablará más en profundidad en el apartado 2.3.2).

También es necesario mencionar la existencia del teorema variacional electromagnético. Éste explica como la mayor concentración de energía de desplazamiento de un modo tiende a localizarse en aquellas regiones del cristal fotónico que presentan mayor

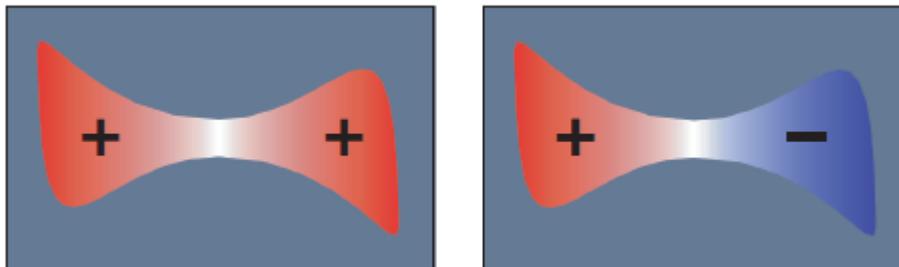
constante dieléctrica además de, a la vez, mantenerse ortogonales al resto de modos que presenten frecuencias más pequeñas. Así, aquel modo que posea la frecuencia más baja será el que minimice la función de energía electromagnética. Este teorema, además, permite conocer cómo se distribuyen los modos en el interior de los cristales fotónicos.

### 2.3.2 Física de estado sólido: simetrías

Como ya se ha mencionado en el apartado anterior, la aparición de simetrías tanto en las estructuras como en los modos de luz nos permite conocer propiedades de dichos modos de manera más sencilla.

#### SIMETRÍA DE INVERSIÓN

Si se tiene una estructura simétrica para la que se considera un modo con una distribución de campo  $H(r)$  (Figura 2) con una frecuencia dada  $\omega$  (siendo este modo no degenerado) entonces, debido a la simetría de la estructura, su modo invertido  $H(-r)$  deberá tener la misma frecuencia. Para que esto sea cierto es necesario que se cumpla  $\alpha^2 H(r) = H(r)$  donde  $\alpha$  valdrá 1 o -1.



**Figura 2:** Las estructuras de ambas imágenes presentan forma simétrica. Estructura con simetría par (izquierda) y estructura con simetría impar (derecha). Los colores representan la polaridad positiva (rojo) y negativa (azul) de los modos (Joannopoulos et al., 2008).

En función del valor de  $\alpha$  se pueden clasificar los modos no degenerados en:

- Simetría par: el modo es invariante a la inversión,  $H(-r) = H(r)$
- Simetría impar: el modo invertido es el opuesto al original,  $-H(-r) = H(r)$

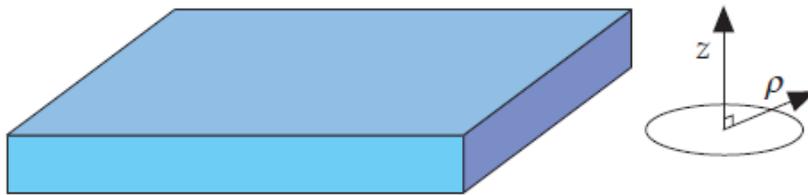
Además, es necesario tener en cuenta que si los modos son no degenerados únicamente será posible tener un tipo de simetría para esa frecuencia (la inversión del modo solo podrá tomar el valor de 1 o -1, no los dos a la vez).

Si, por el contrario, el campo  $H(r)$  es degenerado podrá existir otro modo (no necesariamente su inversión) que presente la misma frecuencia (por la propia definición de degeneración de modos). Sin embargo, mediante combinaciones lineales de este tipo de modos también se podrán obtener modos pares o impares.

### SIMETRÍA DE TRASLACIÓN

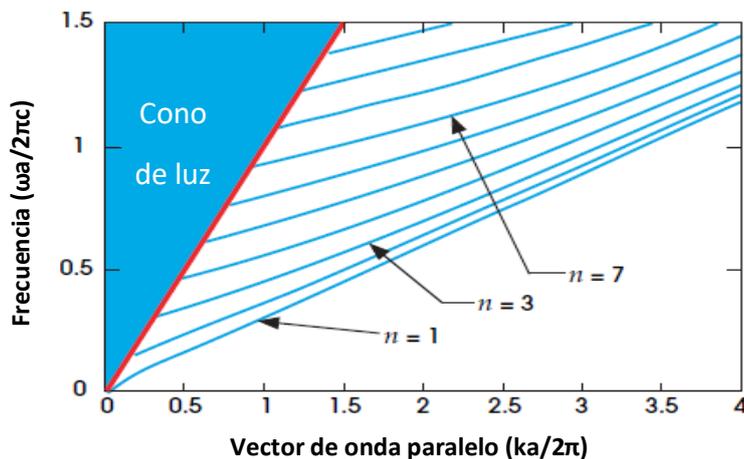
Se considera que un sistema  $\varepsilon(r)$  presenta simetría de traslación cuando al moverlo una distancia 'd' en una dirección concreta éste permanece invariante. Es decir, cuando se cumpla que  $\varepsilon(r + d) = \varepsilon(r)$ .

Existen dos tipos de simetrías de traslación en función del valor que pueda tomar 'd'. Si esta variable puede tomar cualquier valor, la simetría será independiente de la distancia que se tome y recibirá el nombre de simetría de traslación continua. Si, por lo contrario, dicha simetría depende de la distancia aplicada y ésta debe ser necesariamente múltiplo de un determinado valor constante (paso de distancia), la simetría será de traslación discreta.



**Figura 3:** Cristal planar con simetría traslacional continua para cualquier dirección en el plano (representado como  $\rho$  en el eje de coordenadas) comprendido entre los ejes x e y.

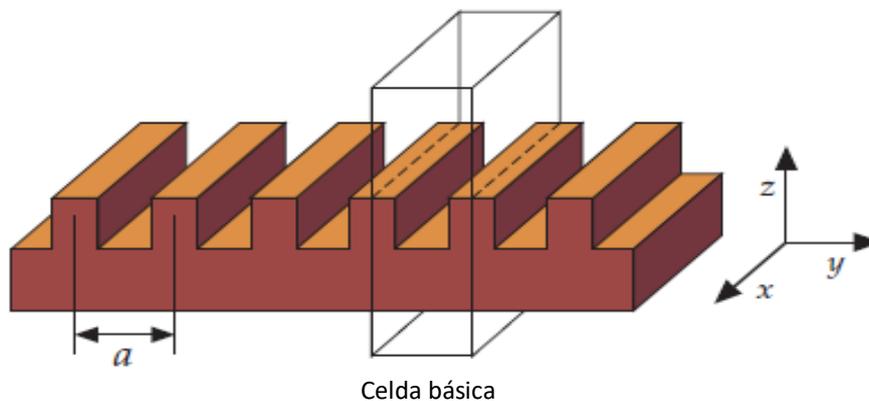
Respecto a la simetría de traslación continua, se puede tomar como ejemplo un cristal planar con forma de prisma rectangular (Figura 3). Su dimensión respecto al eje z es finita y discontinua (por encima y por debajo del cristal hay aire) mientras que respecto a los ejes x e y es infinita y continua. Debido a esto, la traslación respecto a cualquier dirección comprendida dentro del plano x-y será continua mientras que respecto al eje z no existirá. Es decir, el sistema sí puede ser trasladado en los ejes x e y a cualquier distancia en una dirección concreta sin sufrir perturbaciones mientras que respecto al eje z esto no sucede.



**Figura 4:** Representación de la evolución de la frecuencia en función del vector de onda en un cristal planar de altura 'a'. Las bandas generadas al realizar diferentes traslaciones continuas se representan como líneas azules. Cada una de las bandas se identifica con un valor de 'n' diferente. La línea roja representa la línea de luz, por encima de ésta los estados se extienden tanto por el cristal como por el aire. Esta región se conoce como 'cono de luz' (zona azul). Las bandas localizadas por debajo de la línea de luz representan los estados que quedarán confinados en el cristal.

El conjunto de ondas planas creadas por traslación continua se podrá clasificar mediante el denominado vector de onda 'k' que indica la variación del modo cuando éste se somete a traslación. Estos modos también se clasifican en función de sus frecuencias (de menor a mayor) con la variable n que recibe el nombre de número de banda. Este último, por tanto, indicará la posición (en orden creciente) de un modo respecto al resto. Como consecuencia, se podrá clasificar cada onda plana de manera distintiva mediante (k,n). También cabe mencionar la existencia de la 'línea de luz' que separa aquellos estados que únicamente se pueden encontrar en el cristal fotónico (localizados por debajo de la línea de luz) de aquellos que se extienden por el cristal y por el aire (por encima de la línea de luz), a esta región se la conoce como 'cono de luz'. Todo lo antes mencionado se ilustra en la Figura 4.

Por otra parte, también es necesario explicar varios conceptos clave respecto a la simetría de traslación discreta puesto que es este tipo de simetría es el que normalmente se presentará en los cristales fotónicos. Se toma como ejemplo ilustrativo de este tipo de traslación la estructura representada en la Figura 5.



**Figura 5:** Representación de una estructura periódica con simetría traslacional discreta respecto al eje y. La constante de red (que indica el paso de distancia entre una unidad básica de periodicidad y la siguiente) viene representada con la letra 'a'. La celda básica se ilustra mediante un prisma rectangular e indica la porción de la estructura que se repetirá para formar la red.

Así pues, el 'paso de distancia' que se nombró cuando se definió la simetría traslacional discreta (valor constante que marca la distancia entre una unidad básica de periodicidad y la siguiente) recibe el nombre de constante de red 'a'. El vector que indica la dirección y el desplazamiento mínimo se denomina vector de red primitivo y, en el caso de la Figura 5, corresponde a  $a = a\hat{y}$  (puesto que la simetría se muestra a lo largo del eje y). Este vector garantiza que la simetría existirá para cualquier distancia que sea el resultado de multiplicar su valor por un número entero. Es decir, existirá simetría discreta para todos aquellos sistemas  $\varepsilon(r) = \varepsilon(r + R)$  siendo  $R = l \cdot a$  donde 'l' puede ser cualquier número entero. También se define el término de celda básica como la unidad mínima de la estructura a partir de la cual se crea la red de periodicidad (en la Figura 5 representado como el prisma rectangular).

Por ende, se crea otro tipo de clasificación de los modos en función de sus valores de 'k' respecto a la simetría traslacional discreta. No obstante, es necesario tener presente que este tipo de simetría genera modos degenerados, es decir, pueden existir varios vectores de onda que pertenezcan al mismo conjunto de modos. Dichos vectores se pueden formular

genéricamente como  $k + m\left(\frac{2\pi}{a}\right)$  siendo  $m$  un número entero. En consecuencia, si se resolviesen las ecuaciones de Maxwell para cada uno de estos vectores de onda, el valor propio sería el mismo. Teniendo esto en cuenta y tomando  $b = 2\pi/a$  se define el vector de red recíproca primitivo como  $b = b\hat{y}$ . Así, de todos los modos degenerados, si únicamente se toman los modos originales y se define cada una de las funciones de onda, estas vendrán dadas por:

$$H(\dots, \mathbf{y}, \dots) \propto e^{jk_y y} \cdot \mathbf{u}_{k_y}(\mathbf{y}, \dots)$$

**Ecuación 5:** Función de onda tomando simetría traslacional discreta, por ejemplo, en el eje  $y$ . Se descompone en una onda plana  $e^{jk_y y}$  y una función periódica  $\mathbf{u}_{k_y}(\mathbf{y}, \dots)$  que la modula.

A este modo de formular las funciones de onda se llama estado de Bloch (cada estado es una función de onda). Gracias a la Ecuación 5 y a lo anteriormente explicado, se conoce que los vectores de onda  $k_y$  y  $k_y + mb$  son iguales por lo que también se debe cumplir que  $\omega(k_y) = \omega(k_y + mb)$ . Es decir, que las frecuencias de los modos sean periódicas para  $k_y$ . Debido a esto, para evitar valores redundantes de  $k_y$ , bastará con obtener las funciones de onda de las  $k_y$  que se encuentren dentro del intervalo  $[-\frac{\pi}{2}, \frac{\pi}{2}]$ . Este conjunto de valores de  $k_y$  se conoce como zona de Brillouin.

#### SIMETRÍAS DE ROTACIÓN, REFLEXIÓN ESPECULAR E INVERSIÓN

Se conoce como grupo puntual del cristal fotónico al conjunto de operaciones de simetría (rotaciones, reflexiones e inversiones) que se pueden aplicar sobre él dejando la estructura invariante.

Cuando la red presenta simetría rotacional, la zona de Brillouin contiene bandas de frecuencia  $\omega_n(k)$  redundantes. Además, si el cristal fotónico presenta alguna de las simetrías de las que forman el grupo puntual, las bandas de frecuencia correspondientes presentarán las mismas simetrías. Así, teniendo en cuenta la presencia de valores redundantes en la zona de Brillouin (si se consideran las simetrías del grupo puntual) se pueden descartar estos valores a la hora de realizar los cálculos. La nueva región generada por las bandas  $\omega_n(k)$  independientes entre sí (respecto a la simetría) recibe el nombre de zona irreducible de Brillouin. La zona de Brillouin, por tanto, estará formada por un número concreto de repeticiones de la zona irreducible.

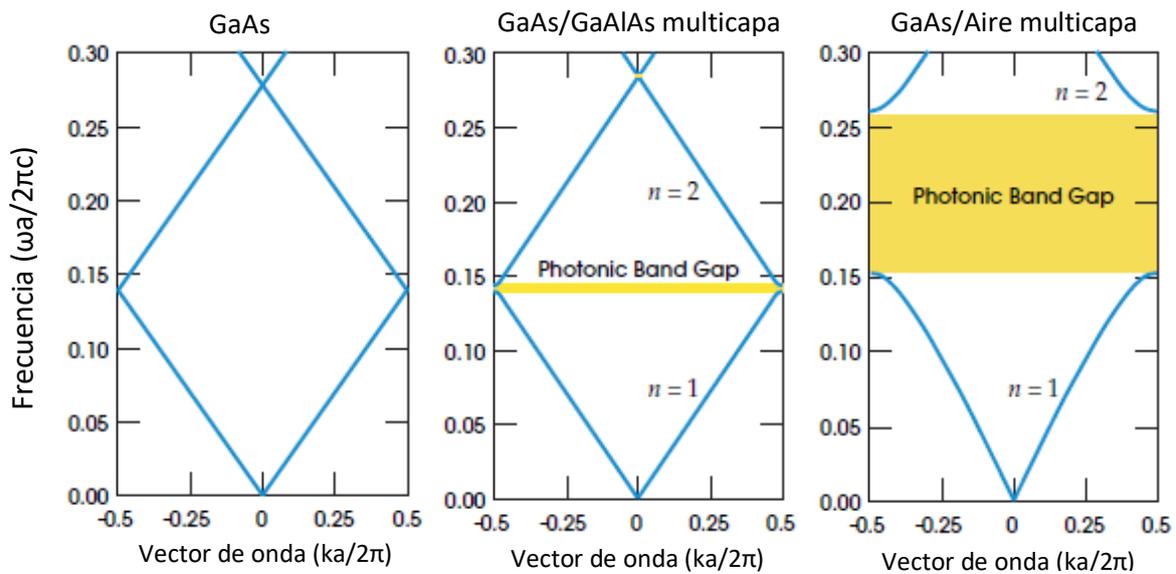
Otro tipo de simetría es la de reflexión especular, para que ésta se presente en una estructura es necesario que exista un plano que la defina, denominado plano de simetría o especular. Así, este plano dividirá dicha estructura en dos de manera que si se realiza una reflexión de ésta sobre el plano especular cada punto de ambas partes de la estructura coincidirá con su homólogo. Gracias a esta simetría se pueden clasificar los modos en función de la polarización del campo. Esta distinción es de gran utilidad ya que aporta información acerca de las propiedades de las simetrías que poseen y facilita la obtención de sus frecuencias (matemáticamente hablando).

En cristales fotónicos 2D (haciendo referencia a la cantidad de dimensiones dónde existe el cristal y no a las dimensiones donde este presenta simetría) con reflexión especular, todos sus modos se pueden clasificar en dos tipos: modo Transversal Eléctrico (TE) y modo Transversal Magnético (TM). Ambos modos se encuentran confinados en el plano xy (si consideramos reflexión especular en el eje z). Sin embargo, en los TE el campo eléctrico es definido como  $(E_x, E_y, H_z)$  mientras que en los TM el campo magnético viene dado por  $(H_x, H_y, E_z)$ . Por lo contrario, cuando se desea analizar cristales fotónicos 3D (haciendo referencia a la cantidad de dimensiones dónde existe el cristal y no a las dimensiones donde este presenta simetría), este tipo de clasificación no es útil puesto que presenta demasiadas restricciones a la hora de realizar cálculos.

### 2.3.3 Photonic band gap (PBG)

Como ya se introdujo en el apartado 2.1 la principal característica de los cristales fotónicos es la aparición de los denominados PBGs. Se define el PBG como la región formada por el conjunto de frecuencias 'prohibidas' para las cuales ningún modo de luz se propagará (siguiendo la periodicidad dieléctrica de la estructura).

Para profundizar más en este concepto se toma como ejemplo un cristal fotónico unidimensional multicapa (dos materiales distintos alternados) cuya periodicidad dieléctrica se genera respecto al eje z. Para esta misma estructura se modifican los materiales que la conforman. Variando las constantes dieléctricas se toma una estructura donde ambos materiales son el mismo (GaAs,  $\epsilon = 13$ ), otra donde las  $\epsilon$  tienen valores cercanos (GaAs,  $\epsilon = 13$ , y GaAlAs,  $\epsilon = 12$ ) y otra donde las constantes dieléctricas son muy distantes (GaAs,  $\epsilon = 13$ , y aire,  $\epsilon = 1$ ). Para estas estructuras se calculan y representan las bandas fotónicas (Figura 6). El cristal que únicamente está formado por GaAs (gráfica de la izquierda) no presenta PBG y, además, contiene información redundante debido a la representación de todos los valores de 'k' presentes dentro de la zona de Brillouin (no la zona irreducible de Brillouin). Por otro lado, las gráficas de las otras dos estructuras sí presentan PBG. Sin embargo, el tamaño de esta banda 'prohibida' varía en función de los materiales empleados (GaAs y GaAlAs en la gráfica central y GaAs y aire en la gráfica de la derecha de la Figura 6). La razón de esta diferencia se encuentra en el teorema variacional electromagnético (explicado en el apartado 2.3.1). Según éste, los modos que presentan frecuencias altas tienden a almacenar la mayor parte de su energía en aquellas regiones donde la constante dieléctrica es menor y viceversa. Es decir, los materiales que presenten valores de  $\epsilon$  muy pequeños únicamente podrán almacenar modos con frecuencias muy elevadas. De este modo, en la gráfica central las frecuencias bajas (localizadas en la banda n=1) almacenarán su energía en el material que mayor  $\epsilon$  presenta (GaAs  $\epsilon = 13$ ). De manera análoga sucederá con las frecuencias altas (banda n=2) y GaAlAs. Finalmente, en la gráfica de la izquierda se observa un mayor rango de frecuencias donde no se propaga la luz. Esto se debe al mayor contraste entre las constantes dieléctricas (GaAs,  $\epsilon = 13$ , y aire,  $\epsilon = 1$ ).



**Figura 6:** Representación de las bandas fotónicas y del PBG de tres estructuras fotónicas unidimensionales multicapa. En la izquierda la estructura donde todas las capas son del mismo material (GaAs,  $\epsilon = 13$ ). En el centro, estructura que presenta capas alternadas de GaAs y GaAlAs ( $\epsilon = 12$ ). En la derecha estructura que presenta capas alternadas de GaAs y aire ( $\epsilon = 1$ ).

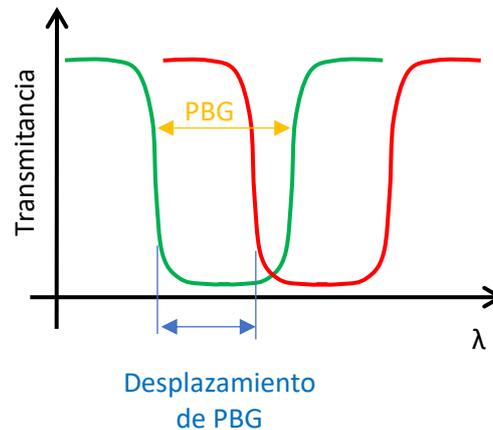
Es decir, el aire, al presentar una constante tan reducida, únicamente almacenará la energía de aquellos modos que presenten frecuencias muy altas (más elevadas que las que puede almacenar el GaAlAs). Como consecuencia, todas las frecuencias que el aire no puede almacenar serán las que no se propaguen por el cristal (generando un PBG).

Así, se puede concluir que el rango de frecuencias que abarca un PBG dependerá de la diferencia de constantes dieléctricas de la estructura: a mayor contraste, mayor PBG.

### 2.3.4 Sensado a partir de PBG

Como ya se ha mencionado en apartados anteriores, una de las características principales de los cristales fotónicos es la aparición de PBG. Estas 'bandas prohibidas' aparecen para determinadas longitudes de onda. La presencia en unas longitudes u otras depende del grado de interacción que posea la luz emitida con dicha estructura. Esto viene determinado, en gran medida, por el medio que rodea el cristal. De esta manera, el índice de refracción del medio condicionará la interacción de los modos de luz con la estructura fotónica de manera que si se produce un cambio en este las longitudes de onda que presentan PBG también se modificarán. Es decir, cuando se modifica el medio externo al cristal fotónico, se produce un desplazamiento del PBG hacia longitudes de onda superiores. Este cambio en el espectro del PBG es el principio utilizado para emplear los cristales fotónicos como sensores. Así, para una estructura determinada, si se conoce la longitud de onda a la que aparece el PBG para un medio definido se puede analizar su posible desplazamiento al cambiar el medio externo que rodea el cristal fotónico. Es decir, a través del análisis del cambio de longitud en el PBG estas estructuras son capaces de detectar cambios en el medio que las rodea (García Rupérez, 2008).

Un ejemplo de este desplazamiento se observa en la Figura 7 donde se muestra la evolución del PBG cuando se modifica el medio externo que lo rodea. La curva verde indica la aparición de un PBG para un valor de  $\lambda$  determinado en un medio externo con un índice de refracción conocido. La curva roja indica el desplazamiento del PBG a una longitud de onda superior debido al cambio del índice de refracción del medio externo (mayor que en el medio donde se mide la curva verde).



**Figura 7:** Evolución del PBG cuando se modifica el medio externo que rodea al cristal fotónico (García Rupérez, 2016).

Así pues, es necesario conocer la longitud de onda donde aparece el PBG para poder estudiar sus posibles variaciones. En este TFG se decide estudiar aquellos PBG localizados en  $\lambda=1550$  nm. A pesar de poder estudiar la evolución de cualquier PBG presente en las estructuras analizadas, la elección de esta longitud de onda se debe a que esta es la que se utiliza habitualmente en los sistemas de comunicaciones ópticas de 3ª ventana. Puesto que es una longitud de onda muy estudiada, la disponibilidad de equipos que tengan la capacidad de medirla es superior. Es por esto por lo que los equipos de medida presentes en los laboratorios del NTC analizan el comportamiento presente para este valor de  $\lambda$ . Así, el desarrollo expuesto en los apartados siguientes se realizará únicamente para el estudio de la evolución de los bordes de PBG presentes en esta longitud de onda en concreto.

## 2.4 Aplicaciones en el ámbito biomédico

Las estructuras fotónicas ofrecen la posibilidad de ser utilizadas para numerosas aplicaciones. Respecto al ámbito biomédico su uso más extendido es como biosensores. Puesto que las estructuras únicamente son sensibles a cambios en el índice de refracción del medio que las rodea es necesario biofuncionalizar la superficie de dichas estructuras para proporcionar la especificidad requerida en el sensado. Para esta funcionalización se añaden receptores capaces de detectar la presencia de un analito en concreto a partir de una muestra. De este modo, el objetivo final es que el sensor pueda manifestar la presencia de cierta sustancia gracias a los cambios producidos en el índice de refracción como consecuencia de la unión receptor-analito generada en la superficie de la estructura fotónica. Estos cambios en el índice de refracción se derivan en modificaciones en el espectro de los PBG presentes en dichas estructuras, esto es lo que se utiliza para detectar la presencia de un analito.

Por otro lado, la tecnología Lab-on-a-chip (LOC) se caracteriza por la búsqueda de los siguientes objetivos: poder acoplar en un mismo dispositivo varias pruebas de diagnóstico de manera que se realicen simultáneamente (detección multiparamétrica), una respuesta rápida, utilización de pequeños volúmenes de muestra, facilidad de uso y portabilidad y precios más económicos de los dispositivos, entre otras características. El desarrollo de sensores fotónicos está impulsando dicha tecnología.

Respecto a aplicaciones biomédicas específicas de este tipo de sensado basado en cristales fotónicos y, en general, en estructuras fotónicas se encuentran, dos proyectos que se están desarrollando en el NTC (que a su vez forman parte del proyecto europeo H2020): SHAPELY y PHOCNOSIS. En primer lugar, SHAPELY se centra en el desarrollo de un dispositivo de diagnóstico precoz de diferentes tipos de cáncer de manera mínimamente invasiva y en un único dispositivo. Es decir, se busca detectar lo más rápido posible (en cuestión de minutos) la presencia de partículas precursoras de la enfermedad a partir de una mínima muestra de sangre (2-3 gotas). Los marcadores empleados para la biofuncionalización son microRNA centrándose en el diagnóstico de cuatro tipos de cáncer: mama, próstata, pulmón y colorrectal. Para su diseño se utilizan estructuras de band gap fotónico biofuncionalizadas. Estas estructuras permiten la amplificación de la respuesta de sensado además de detectar las concentraciones de microRNA esperadas. Este dispositivo, por tanto, generará una disminución de gastos en esta enfermedad por parte de los sistemas de salud gracias a la posible realización de pruebas masivas a la población en búsqueda de fases tempranas del cáncer y a poder aumentar las tasas de supervivencia actuales referentes a esta enfermedad. En segundo lugar, se encuentra PHOCNOSIS. Este proyecto se basa en el desarrollo de un dispositivo de diagnóstico de situaciones agudas y crónicas de enfermedades cardiovasculares. Este proyecto también pretende obtener tecnología LOC a partir de la combinación de estructuras nanofotónicas con un sistema microfluídico. Las estructuras nanofotónicas se emplean a modo de sensado mientras que el sistema microfluídico permite inmovilizar los analitos sobre las estructuras nanofotónicas y aumentar su concentración (García Rupérez, 2016).

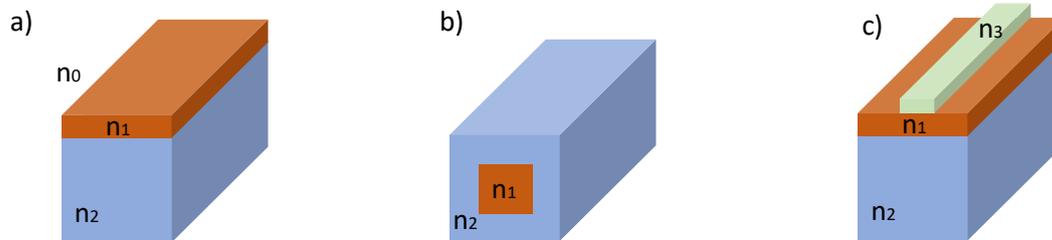
A parte de estas existen otras aplicaciones actuales de estructuras fotónicas como biosensores, por ejemplo, el desarrollo de cristales fotónicos en forma de guía para la detección de la concentración de glucosa (Arafa, Bouchemat, Bouchemat, Benmerkhi, & Hocini, 2017) o el diseño de microportadores formados por cristales fotónicos fabricados mediante hidrogel para la detección de marcadores de cáncer (Xu et al., 2017).

## 2.5 Estructuras fotónicas empleadas

Existe una gran variedad de estructuras fotónicas en función de su geometría, de los materiales empleados para su fabricación, de las propiedades de la luz en ellas, de las dimensiones de simetría, la finalidad de uso, etc. Un ejemplo son las estructuras en forma de guía en las que la luz queda atrapada en cierta área (denominada núcleo) pudiendo transportarse a una distancia determinada (Kilders & Brimont, 2009).

De igual manera que sucede con las estructuras fotónicas en general, existe una gran cantidad de tipos de estructuras guía. Se destacan las planas y las rectangulares (representadas en la Figura 8). Las primeras se caracterizan por ser las más simples, están formadas por una

cubierta superior, el núcleo y la cubierta inferior. Además, únicamente son capaces de confinar la luz en una dimensión. Las rectangulares ofrecen más variedad respecto al tipo de configuraciones diferentes como por ejemplo en canal (Figura 8, b) o en tira (Figura 8, c). Estas estructuras ofrecen el confinamiento de la luz en dos dimensiones (Cascón López & Universidad de Castilla-La Mancha., 2000).



**Figura 8:** Diferentes estructuras de guías: a) guía plana, b) guía en canal y c) guía en tira. b) y c) son guías rectangulares. En a)  $n_0$ ,  $n_1$  y  $n_2$  son los índices de refracción de la cubierta superior, núcleo y cubierta inferior respectivamente

Así, si a una guía planas se le añade periodicidad se obtendrá un cristal fotónico. Sin embargo, como ya se mencionó en el apartado 2.2, únicamente se van a estudiar aquellas que presentan simetría en una dimensión (cristales fotónicos 1D). Por este motivo, se decide desarrollar el proyecto sobre dos tipos de estructuras diferentes: guía corrugada y guía corrugada inversa.

Las estructuras diseñadas únicamente varían en la forma geométrica que adquieren presentando, por el contrario, los mismos materiales en cada parte del cristal y las mismas variables de geométricas.

Así, los materiales empleados en función de su localización para todas las configuraciones son:

- Cubierta superior: agua ( $H_2O$ , índice de refracción  $n= 1.333$ ) u óxido de silicio ( $SiO_2$ ,  $n = 1.444$ )
- Núcleo: Silicio ( $Si$ ,  $n=3.4777$ )
- Cubierta inferior: óxido de silicio ( $SiO_2$ ,  $n= 1.444$ )

Por otro lado, las variables geométricas que se van a modificar de las estructuras son:

- Periodicidad (a): indica el periodo de repetición de la celda básica de la estructura periódica
- Anchura de los elementos transversales de Si ( $w_i$ ): indica la anchura de los elementos transversales que se van añadiendo de manera periódica a la guía central de Si (para el caso de la guía corrugada convencional) o, por el contrario, la anchura de esta guía central que se debe eliminar de manera periódica para crear dichos elementos transversales (en el caso de la guía corrugada inversa)
- Longitud total de cada elemento transversal de Si ( $w_e$ ): indica la longitud de los elementos transversales que se van añadiendo de manera periódica a la guía

central de Si (para el caso de la guía corrugada convencional) o, por el contrario, la longitud que presenta la guía cuando se ha reducido su anchura de manera periódica (para el caso de la guía corrugada inversa)

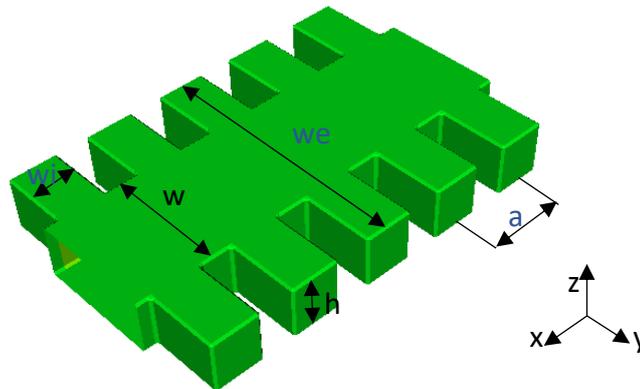
Las otras posibles variables geométricas que se han tomado como constantes son:

- Altura de la guía de Si (h): se toma como valor 220 nm
- Anchura de la guía de Si (w): se toma como valor 460 nm

### 2.5.1 Guía corrugada (corr3D)

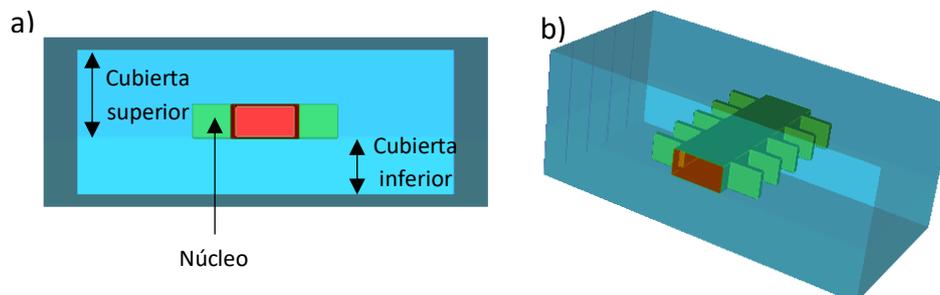
Este tipo de estructura también recibe el nombre de 'corr3D' (nombre de simulación).

Respecto a su geometría, parte de una guía central de Si (de anchura y altura constantes,  $w$  y  $h$ ) a la que se le va añadiendo de manera periódica ( $a$ ) elementos transversales (de anchura  $w_i$  y longitud  $w_e$ ) también de Si como se puede ver en la Figura 9.



**Figura 9:** Representación de una guía-onda corrugada (el núcleo). Las variables en azul ( $w_i$ ,  $w_e$  y  $a$ ) son las que se modificarán en las simulaciones mientras que las variables en negro ( $w$  y  $h$ ) son las que se mantendrán constantes.

La representación de la guía completa, es decir, la cubierta superior e inferior y el núcleo se muestran en la Figura 10. Tanto la Figura 9 como la Figura 10 se han obtenido a través del programa *MayaVi* para el fichero de datos .vtk de una simulación en MPB con valores de  $a=300$  nm,  $w_i=180$  nm y  $w_e=1000$  nm.

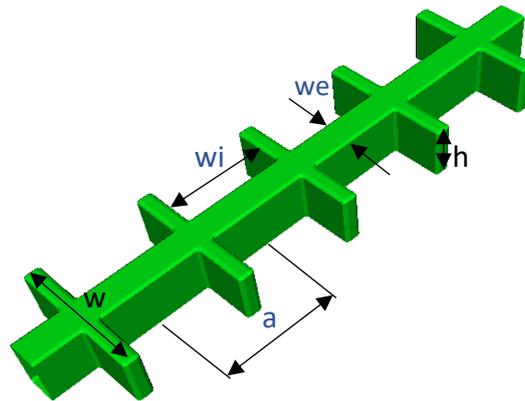


**Figura 10:** Representación de la guía-onda corrugada completa. En a) se ilustran las partes de la guía en un corte transversal y en b) se representa en 3D la guía

## 2.5.2 Guía corrugada inversa (corr3Dinv)

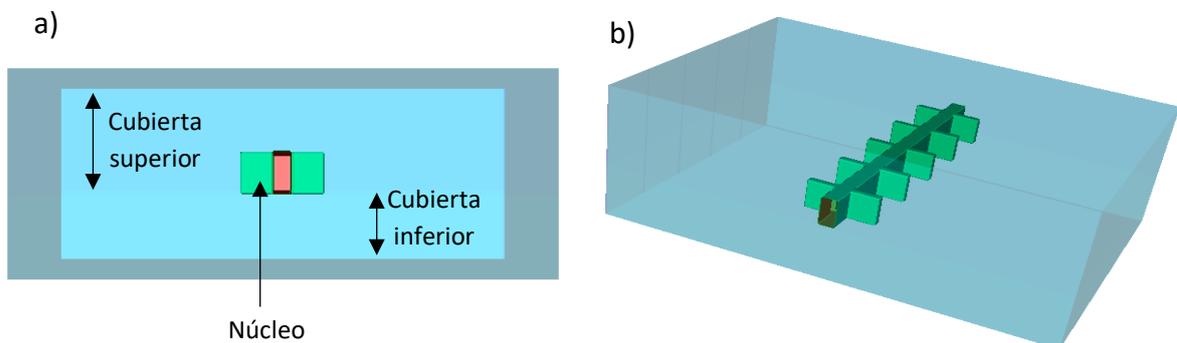
Este tipo de estructura también recibe el nombre de 'corr3Dinv' (nombre de simulación).

Respecto a su geometría, es muy similar a la de la guía corrugada vista en el apartado anterior. La diferencia principal que presenta es que los elementos transversales no se generan por adición de material a la guía si no que se producen de manera 'inversa' a través de la reducción de material de la guía central de Si a una determinada anchura (' $w_i$ ') y a lo largo de una determinada longitud (' $w_e$ '). Es decir, en este caso, se dispone de una guía central a la que se le va reduciendo su anchura de manera periódica formando elementos transversales invertidos. Un ejemplo de esta guía se muestra a continuación (Figura 11).



**Figura 11:** Representación de una guía corrugada inversa (el núcleo). Las variables en azul ( $w_i$ ,  $w_e$  y  $a$ ) son las que se modificarán en las simulaciones mientras que las variables en negro ( $w$  y  $h$ ) son las que se mantendrán constantes.

Al igual que en el apartado anterior, la representación de la guía-onda completa se muestran en la Figura 12. Tanto la Figura 11 como la Figura 12 se han obtenido a través del programa *MayaVi* para la el fichero de datos .vtk de una simulación en MPB con valores de  $a=400$  nm,  $w_i=350$  nm y  $w_e=100$  nm.



**Figura 12:** Representación de la guía-onda corrugada inversa completa. En a) se ilustran las partes de la guía en un corte transversal e en b) se representa en 3D la guía

### 3. DESARROLLO DEL SISTEMA DE OPTIMIZACIÓN

Como se ha mencionado en el apartado 1.2.1, el objetivo principal del trabajo es el desarrollo de un sistema que permita optimizar el proceso de extracción de parámetros y, a partir de su representación gráfica, poder obtener de manera sencilla y visual la relación sensibilidad-tolerancia de diferentes estructuras fotónicas.

Por otro lado, la localización de las funciones en los Anexos en relación con su explicación en este apartado se muestra a continuación (Tabla 1) para su fácil localización si se desean consultar:

Nombre de la función	Apartado de Memoria	Apartado de Anexos
<i>funcion_general.m</i>	3.1	1.1.1
<i>filescreator3D.m</i>	3.2.1	1.2.1
<i>simulaciones_script_creator3D.m</i>	3.2.1	1.2.2
<i>see_all_graphics_3D.m</i>	3.3.1	1.3.2
<i>find_BG_3D.m</i>	3.3.1 y 3.3.2	1.3.3
<i>creator_matrizBG_3D.m</i>	3.3.3	1.3.4
<i>dibbandasMPB_3Dslab_select_parity_wavelength.m (*)</i>	3.3.3	1.3.1
<i>creator_mapacolor_3D_rectas.m</i>	3.4.1	1.4.1
<i>creator_mapacolor_3D_plano1550.m</i>	3.4.1	1.4.2
<i>creator_mapacolor_3D_vol_dist.m</i>	3.4.2	1.4.3
<i>creator_vol_sensibilidad.m</i>	3.5.1	1.5.1
<i>creator_vol_tolerancia.m</i>	3.6.1	1.5.2
<i>creator_mapacolor_4D_sensibilidad.m</i>	3.5.2	1.6.1
<i>creator_mapacolor_4D_tolerancia.m</i>	3.6.1	1.7.1
<i>interseccion_mapacolor_plano1550.m</i>	3.7.1	1.8.1

**Tabla 1:** Localización en la Memoria y en los Anexos de cada función. La función marcada con un asterisco (\*) indica que esta función no se ha realizado en el TFG si no que ha sido proporcionada para facilitar este paso del sistema.

### 3.1 Sistema automatizado general

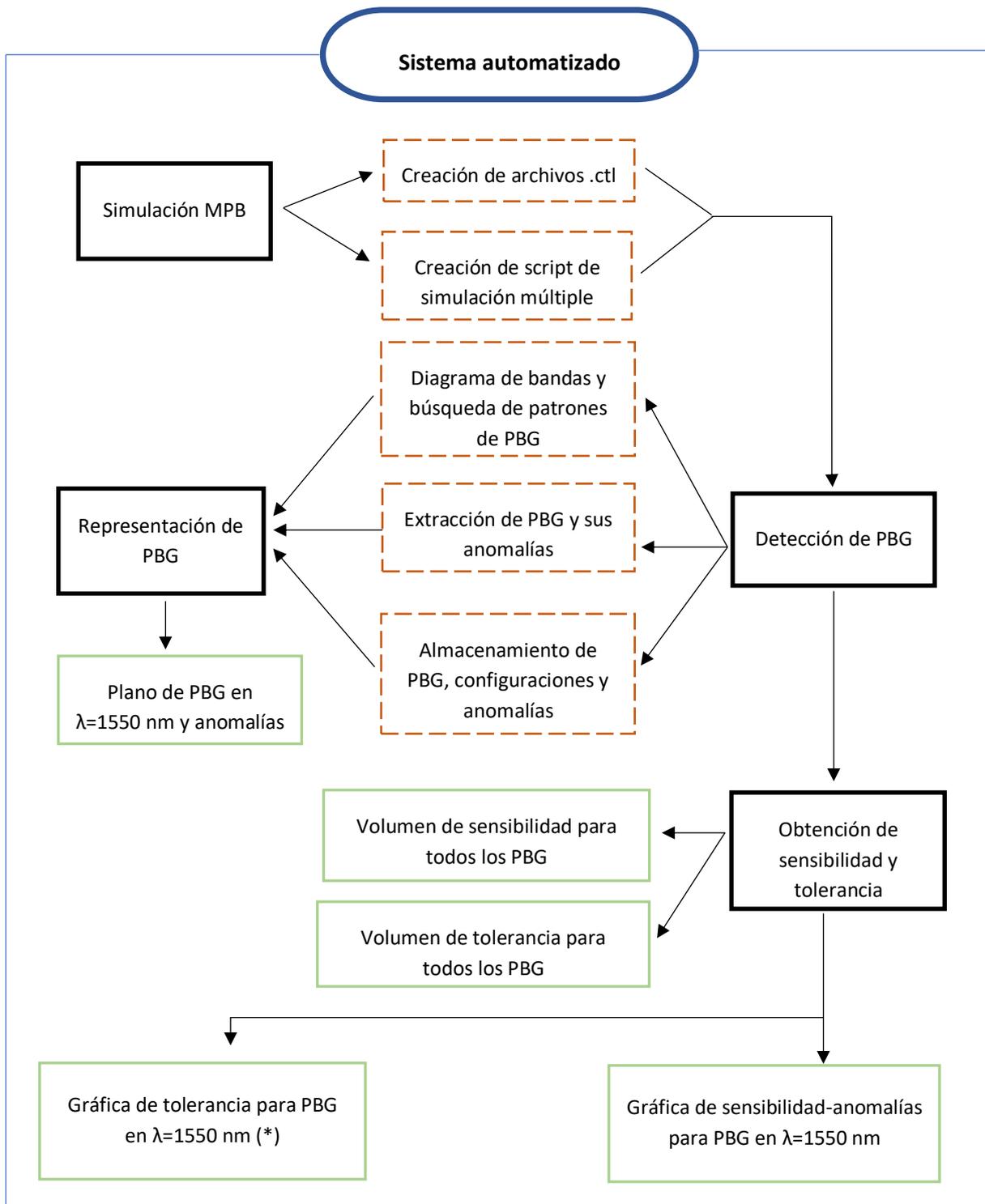
Para poder llevar a cabo el sistema de la manera más automatizada posible, éste se diseña a partir de una única función troncal llamada '*funcion\_general.m*' (apartado 1.1 de Anexos). Dicha función invoca a un conjunto de subprocesos en cascada en los que cada uno de ellos cumple una función específica.

El funcionamiento general del sistema, por tanto, consta de cuatro procesos principales que, enumerados en orden de ejecución, son: simulación en MPB, detección de PBG, representación de PBG y obtención de sensibilidad y tolerancia. Para llevar a cabo el primer proceso es necesaria la creación de los archivos. *ctl* y el script de simulación múltiple (apartado 3.2 de la Memoria) a partir de los cuales y, en adición con el servidor MPB, se consigue realizar las simulaciones de manera automática. Una vez se obtienen los datos de salida generados en dichas simulaciones se continúa con el proceso de detección de PBG en el que se representan los diagramas de bandas de todas las estructuras simuladas en búsqueda de patrones que ayuden a la extracción de estas regiones de interés (apartado 3.3 de la Memoria). Además, también se detectan aquellas configuraciones donde los PBG presentan anomalías por inexistencia de PBG, por solape con una banda (respecto a la representación del diagrama de bandas) o por encontrarse bajo el cono de luz. Todo esto se almacena para su posterior análisis. A partir de estos datos se procede, por un lado, a la representación de los PBG y sus anomalías (únicamente para aquellos bordes de PBG que se encuentren en  $\lambda=1550$  nm, apartado 3.4 de la Memoria) y, por otro lado, al cálculo de la sensibilidad y tolerancia para todas las configuraciones posibles simuladas (no únicamente para aquellas donde el PBG se presenta en el valor de  $\lambda$  antes mencionado, apartado 3.5.1 y 3.6.1 de la Memoria). Finalmente, a partir de la obtención de estos parámetros y de la localización de los PBG de interés (obtenido en el proceso de detección de PBG) se representan los valores de sensibilidad y tolerancia además de las anomalías para cada conjunto de bordes de PBG en  $\lambda=1550$  nm (apartados 3.5.2 y 3.6.2 de la Memoria). A partir de estas gráficas se puede obtener la relación sensibilidad-tolerancia buscada para cada borde de PBG estudiado (apartado 4 de la Memoria). Un resumen del funcionamiento general del sistema se muestra en la Figura 13.

Así, los pasos que sigue el algoritmo del sistema general únicamente son llamadas a otras funciones. Se realiza en el siguiente orden:

1. Llamada a la función '*filescreator3D.m*'
2. Llamada a la función '*simulaciones\_script\_creator3D.m*'
3. Pausa para simular en MPB los archivos generados en el paso 1 y 2
4. Llamada a la función '*creator\_matrizBG\_3D.m*'
5. Llamada a la función '*creator\_mapacolor\_3D\_plano1550.m*'
6. Llamada a la función '*creator\_mapacolor\_3D\_vol\_dist.m*'
7. Llamada a la función '*creator\_vol\_sensibilidad.m*'
8. Llamada a la función '*creator\_vol\_tolerancia.m*'
9. Llamada a la función '*creator\_mapacolor\_4D\_sensibilidad.m*'
10. Llamada a la función '*creator\_mapacolor\_4D\_tolerancia.m*'

Las características de cada función se explicarán a lo largo del apartado 3.



**Figura 13:** Esquema general del funcionamiento del sistema desarrollado. Los pasos principales se muestran encuadrados en negro, los pasos intermedios en línea discontinua naranja y los resultados finales se muestran en verde. (\*) En el caso de la tolerancia no se representan las anomalías puesto que, al realizar posteriormente la comparación entre sensibilidad y tolerancia se considera información repetida (en las gráficas de sensibilidad sí se representan las anomalías).

Por otro lado, los nombres que recibe cada input del sistema en la función troncal '*funcion\_general.m*', su correspondiente explicación y los valores que toman durante el desarrollo del sistema se muestran en la siguiente tabla (Tabla 2):

Variables de entrada	Definición	Valor (en nm) empleado (*)	
		Corr3D	Corr3Dinv
<i>amin</i>	Límite inferior del rango de valores de periodicidad que pueden presentar las guías	300	400
<i>ainterv</i>	Diferencia entre un valor de período y el siguiente	40	50
<i>amax</i>	Límite superior del rango de valores de periodicidad que pueden presentar las guías	420	550
<i>wimin</i>	Límite inferior del rango de valores de anchura de los elementos transversales que pueden presentar las guías	60	200
<i>wiinterv</i>	Diferencia entre un valor de anchura de los elementos transversales y el siguiente	40	50
<i>wimax</i>	Límite superior del rango de valores de anchura de los elementos transversales que pueden presentar las guías	180	350
<i>wemin</i>	Límite inferior del rango de valores de longitud de los elementos transversales que pueden presentar las guías	700	100
<i>weinterv</i>	Diferencia entre un valor de longitud de los elementos transversales y el siguiente	300	100
<i>wemax</i>	Límite superior del rango de valores de longitud de los elementos transversales que pueden presentar las guías	1600	400

**Tabla 2:** Explicación de los inputs de la '*funcion\_general.m*'. (\*) Para el desarrollo del sistema se emplea la guía corrugada 'corr3D', la guía corrugada inversa 'corr3Dinv' se utiliza una vez diseñado el proceso a modo de comprobación de su correcto funcionamiento.

Así, para poder obtener las configuraciones más adecuadas en la relación sensibilidad-tolerancia es necesario obtener datos de una gran cantidad de configuraciones diferentes para la misma estructura. Por este motivo, el procedimiento a seguir para obtener todas las configuraciones posibles es establecer los límites máximo y mínimo del valor de cada variable que se desea tener (inputs del sistema). Con esto, se crea un rango de valores equiespaciados por un intervalo también predefinido. Cada valor dentro del rango generará una configuración diferente de la estructura respecto a esa variable. Esto se realiza para las tres variables elegidas para el estudio: periodicidad (*a*), anchura y longitud de los elementos transversales (*wi* y *we* respectivamente) de manera que el conjunto de todas las configuraciones vendrá dado por todas las posibles combinaciones entre las tres variables.

## 3.2 Simulación MPB

Como se mencionó en el apartado 1.3, el software MPB (al que se tiene acceso mediante un servidor de manera remota) permite realizar simulaciones de las estructuras fotónicas establecidas y, por tanto, la obtención de las propiedades de interés de dichos cristales cuando interactúan con la luz.

Sin embargo, para poder realizar las simulaciones es necesario indicar, entre otras, las propiedades geométricas y los materiales que componen las estructuras. Para esto se definen una serie de archivos donde se describe todo lo necesario para crear y simular una estructura fotónica.

### 3.2.1 Procesamiento de archivos .ctl y del script de simulación múltiple

Son necesarios dos tipos de archivo para poder llevar a cabo la simulación en MPB: archivos .ctl y un script de simulación múltiple.

En primer lugar, para definir las diferentes configuraciones de la estructura fotónica es necesario crear archivos .ctl, el cuál definirá la geometría y el tipo de simulación a realizar. Se dispone de un archivo modelo (no realizado en el TFG, disponible en el apartado 2.1 de los Anexos) del que se irá modificando de manera automatizada las variables de interés. Éstas se pueden ver remarcadas en rojo en la Figura 14.

```
(define eps_Si (* 3.4777 3.4777)) ; Constante dieléctrica del Si (sacado de refractiveindex.info)
(define eps_SiO2 (* 1.444 1.444)) ; Constante dieléctrica del substrato de SiO2 (sacado de refractiveindex.info)
(define eps_aire 1) ; Constante dieléctrica del aire
(define eps_DIW (* 1.333 1.333)) ; Constante dieléctrica del agua
(define a 380) ; Periodo de la guía corrugada
(define h 220) ; Altura de la guía de Si
(define w 450) ; Anchura de la guía de Si
(define wi 100) ; Anchura de los elementos transversales de Si
(define we 1500) ; Longitud total del elemento transversal de Si

(define h-supercell (* 5 h)) ; Altura de la supercelda en nm
(define w-supercell 3000) ; Anchura de la supercelda en nm
```

**Figura 14:** Fragmento del archivo .ctl modelo. Se ha remarcado en rojo las variables que se van a modificar (a, wi y we)

Para llevar a cabo esta tarea se utiliza la función '*filescreator3D.m*'. En ella, se buscan las variables de interés dentro del archivo modelo .ctl y se modifica por el valor de las variables que corresponda. Esta modificación se guarda en un nuevo archivo .ctl cuyo título también debe seguir un formato específico, el cual se muestra en la Figura 15. Es importante que los títulos estén correctamente establecidos puesto que más adelante se extraerán datos de él.

**corr3D-a380-wi100-we1500-h220-upper\_DIW-w450-5bandas.ctl**

**Figura 15:** Ejemplo de título de un archivo .ctl modelo

Por otro lado, una vez se dispone de todos los documentos .ctl que se van a simular es necesario un script que automatice tanto las simulaciones como la obtención de los datos específicos de interés para evitar realizarlas una detrás de otra de manera manual. Se tiene como modelo un archivo (no realizado en el TFG, se puede ver en la Figura 16) donde se encuentran los comandos necesarios para extraer la información de interés de la simulación. Así, a partir de éste se diseña la función '*simulaciones\_script\_creator3D.m*' donde se genera de

manera automatizada la repetición de los comandos del archivo modelo variando únicamente los valores de las distintas configuraciones. El documento nuevo que se genera contiene, por tanto, los comandos que, cuando se realice la simulación, llamarán a los diferentes archivos .ctl que coincidan con las configuraciones establecidas. Es decir, este documento permitirá la posterior simulación en cadena todas las configuraciones posibles diseñadas.

```
mpb corr3D-a380-wi100-we1500-h220-upper_DIW-w450-5bandas.ctl > corr3D-a380-wi100-we1500-h220-upper_DIW-w450-5bandas.out
grep freq corr3D-a380-wi100-we1500-h220-upper_DIW-w450-5bandas.out > corr3D-a380-wi100-we1500-h220-upper_DIW-w450-5bandas.fre
grep zpar corr3D-a380-wi100-we1500-h220-upper_DIW-w450-5bandas.out > corr3D-a380-wi100-we1500-h220-upper_DIW-w450-5bandas.zpar
grep ypar corr3D-a380-wi100-we1500-h220-upper_DIW-w450-5bandas.out > corr3D-a380-wi100-we1500-h220-upper_DIW-w450-5bandas.ypar
mpb-data -r -x 5 epsilon.h5
h5tovtk -d data-new epsilon.h5
mv epsilon.vtk corr3D-a380-wi100-we1500-h220-upper_DIW-w450-5bandas-epsilon.vtk
```

**Figura 16:** Script modelo que contiene todos los comandos que se realizarán durante la simulación. Los cuadros azules marcan los diferentes archivos de salida que se generan tras ejecutar las líneas de código (.out, .fre, .zpar, .ypar, .h5 y .vtk).

Un ejemplo del script automatizado que se genera a través de la función ‘simulaciones\_script\_creator3D.m’ se muestra en la Figura 17. En este caso, se puede observar como el valor de la variable ‘we’ cambia a lo largo del script para simular estructuras con diferentes longitudes del elemento transversal.

```
mpb corr3D-a300-wi60-we700-h220-upper_DIW-w460-5bandas.ctl > corr3D-a300-wi60-we700-h220-upper_DIW-w460-5bandas.out
grep freq corr3D-a300-wi60-we700-h220-upper_DIW-w460-5bandas.out > corr3D-a300-wi60-we700-h220-upper_DIW-w460-5bandas.fre
grep zpar corr3D-a300-wi60-we700-h220-upper_DIW-w460-5bandas.out > corr3D-a300-wi60-we700-h220-upper_DIW-w460-5bandas.zpar
grep ypar corr3D-a300-wi60-we700-h220-upper_DIW-w460-5bandas.out > corr3D-a300-wi60-we700-h220-upper_DIW-w460-5bandas.ypar
mpb-data -r -x 5 epsilon.h5
h5tovtk -d data-new epsilon.h5
mv epsilon.vtk corr3D-a300-wi60-we700-h220-upper_DIW-w460-5bandas-epsilon.vtk

mpb corr3D-a300-wi60-we1000-h220-upper_DIW-w460-5bandas.ctl > corr3D-a300-wi60-we1000-h220-upper_DIW-w460-5bandas.out
grep freq corr3D-a300-wi60-we1000-h220-upper_DIW-w460-5bandas.out > corr3D-a300-wi60-we1000-h220-upper_DIW-w460-5bandas.fre
grep zpar corr3D-a300-wi60-we1000-h220-upper_DIW-w460-5bandas.out > corr3D-a300-wi60-we1000-h220-upper_DIW-w460-5bandas.zpar
grep ypar corr3D-a300-wi60-we1000-h220-upper_DIW-w460-5bandas.out > corr3D-a300-wi60-we1000-h220-upper_DIW-w460-5bandas.ypar
mpb-data -r -x 5 epsilon.h5
h5tovtk -d data-new epsilon.h5
mv epsilon.vtk corr3D-a300-wi60-we1000-h220-upper_DIW-w460-5bandas-epsilon.vtk

mpb corr3D-a300-wi60-we1300-h220-upper_DIW-w460-5bandas.ctl > corr3D-a300-wi60-we1300-h220-upper_DIW-w460-5bandas.out
grep freq corr3D-a300-wi60-we1300-h220-upper_DIW-w460-5bandas.out > corr3D-a300-wi60-we1300-h220-upper_DIW-w460-5bandas.fre
grep zpar corr3D-a300-wi60-we1300-h220-upper_DIW-w460-5bandas.out > corr3D-a300-wi60-we1300-h220-upper_DIW-w460-5bandas.zpar
grep ypar corr3D-a300-wi60-we1300-h220-upper_DIW-w460-5bandas.out > corr3D-a300-wi60-we1300-h220-upper_DIW-w460-5bandas.ypar
mpb-data -r -x 5 epsilon.h5
h5tovtk -d data-new epsilon.h5
mv epsilon.vtk corr3D-a300-wi60-we1300-h220-upper_DIW-w460-5bandas-epsilon.vtk
```

**Figura 17:** Ejemplo de script de simulación múltiple donde varía el valor de la longitud de los elementos transversales de la guía corrugada ‘we’. Los cuadros rojos marcan los valores que toma ‘we’ en las diferentes simulaciones.

Cabe mencionar que las funciones diseñadas ‘filescreator3D.m’ y ‘simulaciones\_script\_creator3D.m’ presentan los mismos inputs que la función troncal ‘funcion\_general.m’ (su explicación se encuentra en la Tabla 2 del apartado Sistema automatizado general 3.1).

### 3.2.2 Obtención de archivos .out, .fre, .ypar y .zpar

Cuando se ejecuta en el servidor el script de simulación múltiple se crean diferentes archivos de salida. En la Figura 16 se muestran estos archivos encuadrados en azul; cada uno de ellos contiene información diferente a cerca de la simulación:

- archivo.out: archivo de salida de la simulación con el servidor MPB. Este documento contiene todos los cálculos y pasos realizados como, por ejemplo, la resolución a las ecuaciones de Maxwell, el cálculo de las frecuencias de los

modos en función de la banda a la que corresponden y los valores de paridad también en función de las bandas.

- archivo.fre: almacena las filas de 'archivo.out' donde se encuentren los valores de las frecuencias.
- archivo.ypar: almacena las filas de 'archivo.out' donde se encuentren los valores de las paridades horizontales.
- archivo.zpar: almacena las filas de 'archivo.out' donde se encuentren los valores de las paridades verticales.
- archivo.h5: a partir del servidor MPB se genera un archivo gráfico con la distribución dieléctrica creada en la celda básica.
- archivo.vtk: a partir del archivo.h5 se genera información gráfica que se utilizará para representar la estructura simulada.

En los archivos .ctl de las estructuras se indica que, a la hora de simular, únicamente se realice el cálculo de los modos con paridad par en z (correspondientes a los modos TE). Es por este motivo por el cual los archivos .zpar aparecen vacíos. Esta elección de estudiar únicamente los modos pares se realiza para, de cara al futuro a la hora de realizar pruebas en el laboratorio, con prototipos cuyas dimensiones se extraen de este TFG, poder contar con mayores facilidades en el diseño del experimento. Los modos pares se excitan cuando la fibra óptica que emite la luz lo hace a la vez para toda la estructura mientras que los modos impares necesitan que la luz incida en una zona específica para poder propagarse y así excitar toda esta clase de modos. Por lo tanto, el estudio de modos impares conlleva una mayor dificultad de diseño de experimentos ya que, además de la fibra óptica, es necesario añadir lentes y demás instrumentos para conseguir su excitación mientras que para los modos pares únicamente se requiere que el haz de luz incidente sea de mayor tamaño que la estructura (para que incida por igual).

Así, a partir de los datos existentes en .out, .fre y .ypar se comienza el sistema de detección del PBG que se explica a continuación.

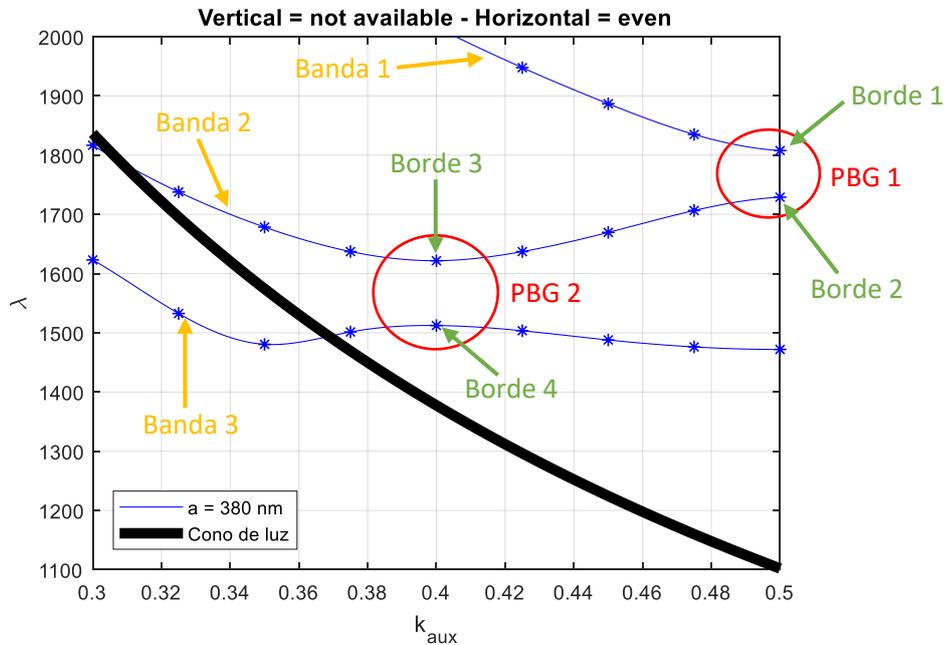
### 3.3 Detección de los PBGs

El conjunto de *Photonic Band Gaps* que presenta un cristal fotónico aporta información muy útil a la hora de decidir su configuración geométrica óptima. En los subapartados siguientes se explican los pasos realizados para su obtención.

#### 3.3.1 Obtención del diagrama de bandas y búsqueda de patrones del PBG

Una vez se han realizado las simulaciones correspondientes se utiliza el programa 'dibbandasMPB\_3Dslab\_select\_parity\_wavelength.m' (no realizado en el TFG) para representar el diagrama de bandas de cada posible configuración. Los valores escogidos como input en el sistema generan 64 posibles combinaciones por lo que se tendrán 64 diagramas de banda diferentes. Sin embargo, esta función únicamente representa un único diagrama de bandas por lo que se realiza una función denominada 'see\_all\_graphics.m' que represente todos los diagramas de todas las configuraciones de manera automática.

En la Figura 18 se representa el diagrama de bandas tipo a modo de ejemplo. En este se puede observar varios elementos importantes: las líneas y asteriscos azules marcan las bandas horizontales de paridad par para una configuración determinada, la línea gruesa negra indica por donde pasa la línea de luz formando el cono de luz por debajo de esta y, por último, se pueden observar entre las bandas contiguas ciertas zonas en las que no aparecen bandas guiadas, estas regiones son los PBG (marcados con círculos rojos).



**Figura 18:** Diagrama de bandas tipo. Las líneas y asteriscos azules marcan las bandas de la estructura, por debajo de la línea gruesa negra (línea de luz) se encuentra el cono de luz. Las zonas que presentan un círculo rojo marcan la existencia de un PBG en esa zona, cada borde de PBG recibe un nombre diferente marcado en verde. El orden de enumeración de las bandas es descendiente mostrándose en color naranja.

Las guías corrugadas, por lo general, poseen dos PBG. Cada uno de estos está formado por un borde superior y un borde inferior. Para facilitar la lectura a partir de ahora se denominará a dichos bordes del siguiente modo:

- Borde 1: hace referencia al borde superior del PBG1
- Borde 2: hace referencia al borde inferior del PBG1
- Borde 3: hace referencia al borde superior del PBG2
- Borde 4: hace referencia al borde inferior del PBG3

En la Figura 18 se puede observar el uso de estas abreviaciones. Sin embargo, se utilizará la nomenclatura larga cuando no dé lugar a posibles malinterpretaciones en la explicación. Además, se elige la enumeración de las bandas en orden descendiente.

Por otro lado, para poder detectar patrones en los bordes de PBG se representan todos los diagramas de banda mediante la función '*see\_all\_graphics.m*' (esta función únicamente llama a la función que genera los diagramas de bandas para todas las configuraciones establecidas) y se analiza visualmente el comportamiento de cada banda en las diferentes

configuraciones. Se consideran las siguientes pautas como aquellas condiciones que se deben cumplir para que un borde de PBG se considere óptimo:

- El borde 1 y el borde 2 se encuentran en la banda 1 y banda 2 respectivamente siempre en el último valor del vector de onda  $k_{aux}=0.5$
- El borde 3 corresponde al valor mínimo que presenta la banda 2
- El borde 4 corresponde con el pico (cambio de pendiente de la banda de creciente a decreciente) que presenta la banda 3
- Dentro del rango de longitudes de onda que no presentan respuesta por estar dentro de un PBG no debe existir ningún modo que se excite (i.e. que presente respuesta en la banda).

Estas pautas son las que se utilizan para crear la función '*find\_BG\_3D.m*' que detecta de manera automática los PBG existentes en los diagramas de bandas. Es decir, únicamente con el análisis de los valores de ciertas variables generadas en la función '*dibbandasMPB\_3Dslab\_select\_parity\_wavelength.m*' se puede conocer el valor de las longitudes de onda que definen un borde de PBG. Esta función se explicará más en profundidad en el apartado 3.3.2.

A continuación, se muestra una tabla donde se indican los inputs (variables de entrada) y outputs (variables de salida) de las funciones '*see\_all\_graphics.m*', y '*dibbandasMPB\_3Dslab\_select\_parity\_wavelength.m*' (Tabla 3). Las variables relacionadas con la función '*find\_BG\_3D.m*' se explican al final del siguiente apartado (Tabla 4).

Función		
<i>dibbandasMPB_3Dslab_select_parity_wavelength.m</i>		
		<b>Definición</b>
<b>Variable de entrada</b>	<i>titulo</i>	Nombre del archivo .ctl
<b>Variable de salida</b>	<i>kaux</i>	Vector que contiene los valores equiespaciados del vector ' <i>k_vector</i> ' que contiene los posibles vectores de onda
	<i>a</i>	Período de la estructura
	<i>sal</i>	Matriz que almacena los valores de cada borde de PBG para las diferentes bandas. Cada fila de dicha variable corresponde a una banda determinada
<i>see_all_graphics.m</i>		
<b>Variable de entrada</b>	Coincide con las explicadas en la Tabla 2	
<b>Variable de salida</b>	Diagrama de bandas	Representación gráfica del comportamiento de la luz en una estructura fotónica determinada.

**Tabla 3:** Resumen de los inputs y outputs de las funciones '*dibbandasMPB\_3Dslab\_select\_parity\_wavelength.m*' y '*see\_all\_graphics.m*'.

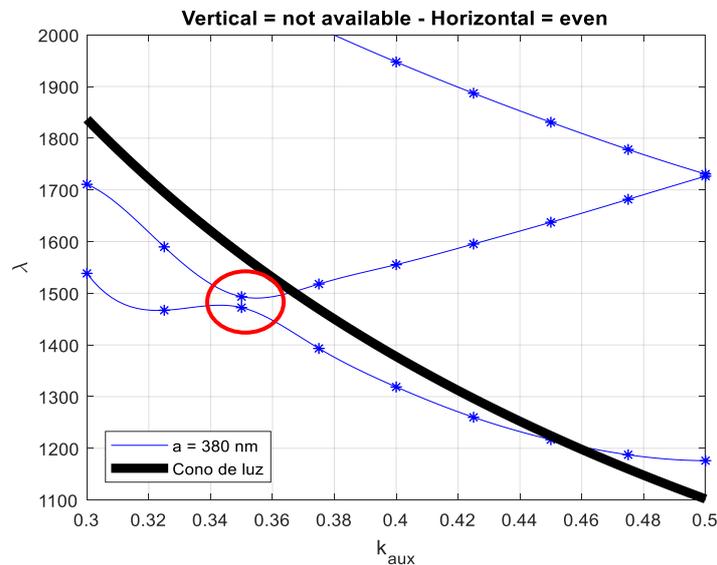
### 3.3.2 Búsqueda y extracción de anomalías presentes en cada PBG

A pesar de lo explicado en el apartado anterior, los diagramas de bandas pueden ser muy diversos y, en consecuencia, presentar anomalías que indican pérdidas en la efectividad de guiado de la luz para dicha estructura.

Teniendo esto en cuenta, se observa que existen bordes de PBG que no cumplen algunos de los requisitos descritos anteriormente; esto es lo que se denomina a partir de ahora como anomalías. A continuación, se explicará más en detalle las más frecuentes.

#### PBG DEBAJO DEL CONO DE LUZ

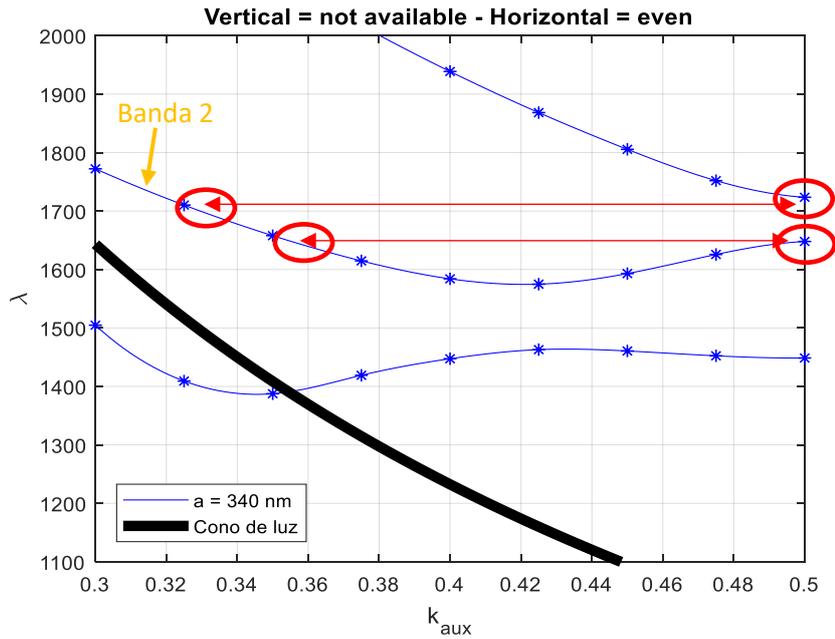
Como su propio nombre indica, aparece cuando el PBG se encuentra por debajo del cono de luz. Para las frecuencias y vectores de onda estudiadas únicamente podrá aparecer esta anomalía en el PBG2. La presencia de esta anomalía provoca una reducción en la exactitud en la que las longitudes de onda no se transmitirán respecto al mismo PBG si estuviera por encima del cono de luz. Un ejemplo de este tipo de anomalía se presenta en la siguiente ilustración (Figura 19):



**Figura 19:** Diagrama de bandas con PBG2 por debajo del cono de luz (marcado con un círculo rojo)

#### BORDE DE PBG CON SOLAPE DE $\lambda$

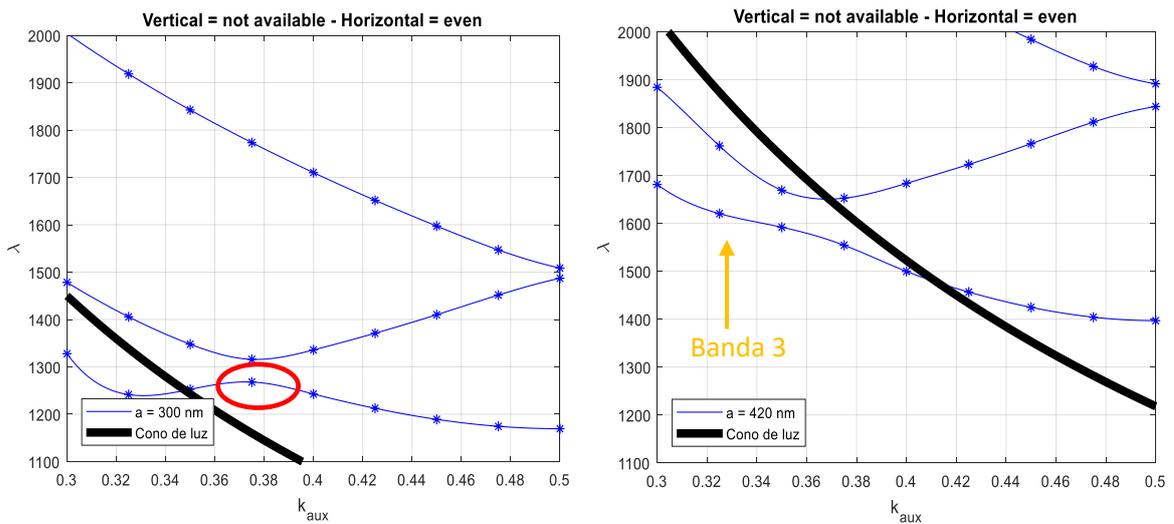
Este tipo de anomalía surge cuando dentro del PBG existen valores de  $\lambda$  que sí transmiten la luz. Sin embargo, para simplificar el problema se va a tomar como anomalía únicamente aquellos valores de  $\lambda$  donde existe un borde de PBG y un modo de luz excitado al mismo tiempo (como se muestra en la Figura 20) por solape de una banda. Estas anomalías generan distorsión del PBG puesto que hay una zona que sí transmite luz dentro de la banda prohibida. Se realiza la búsqueda de las longitudes de onda que generan distorsión para todos los bordes de PBG, pero, en el caso de las guías corrugadas, esta anomalía únicamente aparece en los bordes 1 y 2. También cabe mencionar que puede suceder que uno de los bordes de un PBG presente distorsión y el otro no, todo depende de la distribución de la banda 2 en el diagrama.



**Figura 20:** Diagrama de bandas con borde 1 y borde 2 distorsionados por solape de la banda 2. Se marca con círculos y flechas rojas los puntos donde los bordes presentan solape con la banda 2. Esta banda se indica con texto naranja.

**NO EXISTE BORDE DE PBG**

Esta anomalía solamente aparece en el borde 4 cuando la banda 3 no presenta ningún pico. En estos casos la función *'find\_BG\_3D.m'* marca el valor del borde como 'NaN' para su posterior análisis. En la Figura 21 se muestra una comparación de dos diagramas de banda, a la izquierda uno que sí presenta borde 4 y a la derecha otro diagrama donde la banda 3 no presenta ningún borde.



**Figura 21:** Comparación entre diagramas de bandas. En el diagrama de la izquierda se muestra con un círculo rojo el borde 4 presente en la banda 3. En el diagrama de la derecha se observa como para esta misma banda no aparece ningún pico (por lo que no existe borde 4)

Teniendo en cuenta las posibles anomalías en los bordes de PBG se diseña parte de la función '*find\_BG\_3D.m*' para que las detecte y genere una respuesta de manera que las configuraciones que las presenten sean identificadas y puedan ser estudiadas más adelante.

Los pasos que sigue el algoritmo de esta función son los siguientes:

- Respecto a la detección de los bordes de PBG:
  1. A partir de los inputs de la función (las salidas de la función que genera los diagramas de bandas) se crean diferentes matrices que contienen las coordenadas del cono de luz, y las coordenadas de los valores de cada banda por separado
  2. Se localiza el borde 1 de PBG; este valor se encuentra en la banda 1 y se conoce que siempre estará en  $k_{aux}=0.5$
  3. Se localiza el borde 2 de PBG; este valor se encuentra en la banda 2 y se conoce que siempre estará en  $k_{aux}=0.5$
  4. Se localiza el borde 3 de PBG; se encuentran en la banda 2 y se conoce que será el mínimo de todas las longitudes de onda que dicha banda presenta
  5. Se localiza el borde 4 de PBG; se encuentra en la banda 3 y se conoce que será el valor que presente un pico en la evolución de la banda respecto a las longitudes de onda que la componen
  
- Respecto a la detección de anomalías
  6. Se detectan la posible distorsión en el borde 1. Se extraen los valores de la banda 2 que se encuentran por encima del cono de luz. A partir de este conjunto de datos se compara si el valor del borde 1 en algún momento es inferior a algún punto de esta banda. De ser así significará que para este punto el borde 1 presenta distorsión por solape con la banda 2
  7. Se detecta la distorsión presente en el borde 2 de PBG del mismo modo que obtiene para el borde 1 en el paso anterior.
  8. La detección de distorsiones en los bordes 3 y 4 se realiza del mismo modo que para los bordes 1 y 2. Se cambia la comparación de la banda 2 (utilizada en los bordes 1 y 2) por la banda 3.
  9. Para detectar si el borde 3 y 4 se encuentran por debajo del cono de luz se compara su valor con la parte de la banda 2 y 3 respectivamente que se encuentra por encima del cono de luz. Si el borde se encuentra en dicha parte de la banda no estará por debajo del cono de luz
  10. La detección de la inexistencia del borde 4 se realiza de manera automática cuando se busca el pico de la banda 3. Si no se encuentra ningún pico significa que no existirá borde 4 y este tomará el valor de 'NaN'

Todos estos pasos se repiten para todas las configuraciones posibles, en este caso 64 veces.

Las variables de entrada y salida de esta función se muestran en la Tabla 4 siguiente.

Función		
<i>find_BG_3D.m</i>		
Definición		
<b>Variable de entrada</b>	Coincide con las variables de salida de la función <i>dibbandasMPB_3Dslab_select_parity_wavelength.m</i>	
<b>Variable de salida</b>	<i>borde1</i> <i>borde2</i> <i>borde3</i> <i>borde4</i>	Longitud de onda donde se encuentra el borde 1,2,3 y 4 respectivamente
	<i>borde3_bajo_cono</i> <i>borde4_bajo_cono</i>	Variable lógica. Vale '1' si el borde 3 y 4 respectivamente está por debajo del cono de luz y '0' si no
	<i>banda3_sinBG</i>	Variable lógica. Vale '1' si no existe el borde 4 y '0' si no
	<i>borde1_dist</i> <i>borde2_dist</i> <i>borde3_dist</i> <i>borde4_dist</i>	Variable lógica. Vale '1' si el borde 1, 2, 3 y 4 respectivamente presenta distorsión y '0' si no

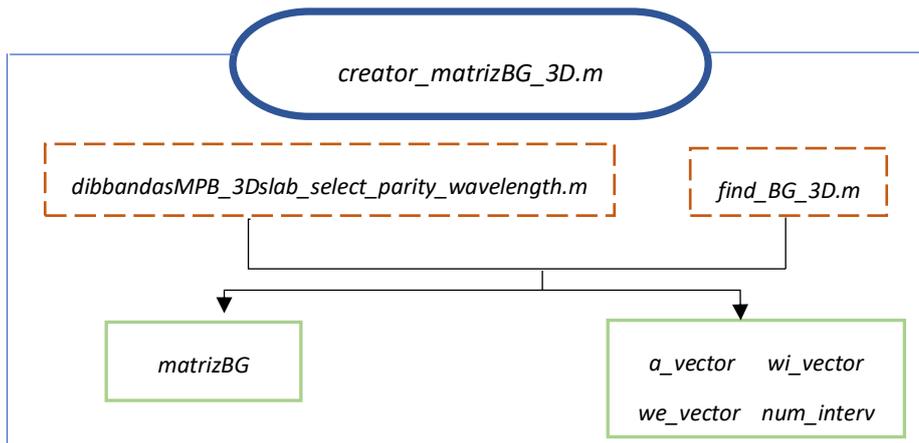
**Tabla 4:** Resumen de los inputs y outputs de la función '*find\_BG\_3D.m*'

### 3.3.3 Almacenamiento de datos (configuraciones, PBGs, anomalías...)

Es importante almacenar todos los datos previamente explicados: bordes de PBG, anomalías y las configuraciones donde se encuentran para el posterior análisis de estos datos. Esta tarea se lleva a cabo mediante la función '*creator\_matrizBG\_3D.m*'. Esta función contiene llamadas internas en primer lugar a la función '*dibbandasMPB\_3Dslab\_select\_parity\_wavelength.m*' y seguidamente a '*find\_BG\_3D.m*' (como se observa en la Figura 22).

Los pasos del algoritmo de esta función son los siguientes:

1. Llamada a la función que genera el diagrama de bandas
2. Llamada a '*find\_BG\_3D.m*'
3. Almacenamiento en '*matrizBG*' de los outputs generados en el paso 2
4. Se repite el proceso para todas las configuraciones posibles
5. Se crea la interpolación de los valores de las coordenadas originales de la simulación. Estos vectores servirán en gran parte de las funciones diseñadas en el sistema.



**Figura 22:** Resumen del funcionamiento interno de la función '*creator\_matrizBG\_3D.m*'. Las funciones que invoca están enmarcadas en naranja y los outputs de la función se indican en verde.

Cuando se ejecuta la función de almacenamiento esta genera cinco outputs mostrados en color verde en la Figura 22. El significado de los inputs y outputs de la función se explica a continuación en la Tabla 5.

Función		
<i>creator_matrizBG_3D.m</i>		
		Definición
Variable de entrada	Coincide con las variables de entrada de ' <i>funcion_general.m</i> ' explicadas en la Tabla 2	
Variable de salida	<i>matrizBG</i>	Matriz que contiene todos los datos originales de cada simulación. Cada columna representa una combinación de configuraciones posibles. Los valores de cada fila se explican en la Tabla 4
	<i>a_vector</i>	Interpolación de los valores originales de ' <i>a</i> ' utilizados en la simulación
	<i>wi_vector</i>	Interpolación de los valores originales de ' <i>wi</i> ' utilizados en la simulación
	<i>we_vector</i>	Interpolación de los valores originales de ' <i>we</i> ' utilizados en la simulación
	<i>num_interv</i>	Indica el tamaño de equiespaciado de las interpolaciones de ' <i>a_vector</i> ', ' <i>wi_vector</i> ' y ' <i>we_vector</i> '

**Tabla 5:** Resumen de los inputs y outputs de la función '*creator\_matrizBG\_3D.m*'

El output '*matrizBG*' contiene información muy relevante a la hora de realizar estudios posteriores por lo que es necesario aclarar cómo se han distribuido estos datos en la matriz. Ésta se organiza por filas de manera que cada fila contiene la información de una variable diferente (la explicación detallada se muestra en la Figura 23). Así, finalmente se obtendrá una matriz de 14 filas (correspondientes a 14 variables) y 64 columnas (correspondientes a todas las estructuras simuladas) para el caso de las guías corrugadas.

	1	2	3
1	300	300	300
2	60	60	60
3	700	1000	1300
4	1.4948e+03	1.5045e+03	1.5076e+03
5	1.4869e+03	1.4871e+03	1.4871e+03
6	1.2568e+03	1.2785e+03	1.3035e+03
7	NaN	1.2406e+03	1.2531e+03
8	1	0	0
9	0	1	0
10	1	0	0
11	0	0	0
12	0	0	0
13	0	0	0
14	0	0	0

Periodicidad ' $a$ '  
 Anchura de los elementos transversales ' $w_i$ '  
 Longitud de los elementos transversales ' $w_e$ '

Borde 1, 2, 3 y 4 de PBG

Borde 3 y 4 bajo cono de luz

Borde 4 no existe

Borde 1, 2, 3 y 4 distorsionados

**Figura 23:** Distribución por filas de la matriz denominada '*matrizBG*' creada a partir de la función '*creator\_matrizBG\_3D.m*'. A partir de un fragmento de dicha matriz se diferencia por colores la variable que corresponde a cada fila.

De esta manera, cada columna corresponderá a una combinación específica de configuraciones donde se muestran los valores de los bordes y las anomalías que presenta.

### 3.4 Representación de PBG

Una vez se recoge toda la información en '*matrizBG*' se procede a su análisis. En primer lugar, se obtiene y representa aquellas configuraciones donde el borde de PBG se encuentra en  $\lambda=1550$  nm y después se extraen aquellos bordes donde los existen anomalías.

#### 3.4.1 Extracción y representación de PBGs situados en $\lambda=1550$ nm

La detección y representación de las configuraciones para las que el PBG se encuentra en  $\lambda=1550$  nm se realiza mediante el diseño de dos funciones: '*creator\_mapacolor\_3D\_rectas.m*' y '*creator\_mapacolor\_3D\_plano1550.m*'.

La función '*creator\_mapacolor\_3D\_rectas.m*' es la primera que se creó. Ésta detecta la longitud de onda de interés para cada borde de PBG y crea líneas (en 3D manteniendo para cada línea el valor de ' $a$ ' constante) que marcan aquellas configuraciones de la estructura fotónica que la presentan. Además, también se añaden aquellos puntos que presentan anomalías en la representación.

Los pasos que sigue el algoritmo de esta función son los siguientes:

- Para la obtención y representación de los bordes de PBG en  $\lambda=1550$  nm
  1. Se crea un mallado a partir de los vectores que contienen los parámetros de la estructura interpolados ' $a\_vector$ ', ' $w_i\_vector$ ' y ' $w_e\_vector$ '. Es decir, para cada

vector se genera una matriz que repita sus valores respecto a una dimensión determinada. Así, la matriz de mallado de 'a' repetirá las configuraciones contenidas en 'a\_vector' por filas, la de 'wi' lo hará por columnas y la de 'we' repetirá cada valor en diferentes matrices. Así, se consigue obtener una matriz tridimensional en la que se presentan todas las combinaciones posibles entre todas las variables

2. Se elimina los posibles valores 'NaN' existentes en los bordes de PBG (sobre todo para el caso de los bordes 4) puesto que su presencia genera fallos y se almacena cada valor de los bordes en las configuraciones correspondientes creadas en el mallado del paso 1 (se utiliza interpolación lineal)
  3. A partir de los valores de la matriz de bordes que se ha generado en el paso 2 se almacenan en una nueva matriz únicamente las combinaciones de dimensiones de la estructura fotónica que proporciona un borde de PBG en  $\lambda=1550$  nm y representan mediante líneas
- Para la representación de las configuraciones originales (las que se han simulado con MPB) de los bordes de PBG que presentan anomalías:
    4. Se busca en la matriz original de datos 'matrizBG' las configuraciones donde los diferentes bordes de PBG presentan anomalías. Se almacenan dichas posiciones y se representan como puntos inconexos entre si

Con respecto a las variables de entrada y salida que presenta esta función y su correspondiente explicación se encuentra a continuación (Tabla 6).

Función			
<i>creator_mapacolor_3D_rectas.m</i>			
<b>Definición</b>			
<b>Variable de entrada</b>	Coincide con las variables de salida (excepto <i>num_interv</i> ) de 'creator_matrizBG_3D.m' explicadas en la Tabla 5		
<b>Variable de salida</b>	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%; vertical-align: top;"> <i>mapacolor3D_rectas_borde1</i>  <i>mapacolor3D_rectas_borde2</i>  <i>mapacolor3D_rectas_borde3</i>  <i>mapacolor3D_rectas_borde4</i> </td> <td style="width: 50%; vertical-align: top;">           Representación de todas las configuraciones en las que el borde (1, 2, 3 y 4 respectivamente) presenta PBG en <math>\lambda=1550</math> nm y de las posibles anomalías presentes         </td> </tr> </table>	<i>mapacolor3D_rectas_borde1</i> <i>mapacolor3D_rectas_borde2</i> <i>mapacolor3D_rectas_borde3</i> <i>mapacolor3D_rectas_borde4</i>	Representación de todas las configuraciones en las que el borde (1, 2, 3 y 4 respectivamente) presenta PBG en $\lambda=1550$ nm y de las posibles anomalías presentes
<i>mapacolor3D_rectas_borde1</i> <i>mapacolor3D_rectas_borde2</i> <i>mapacolor3D_rectas_borde3</i> <i>mapacolor3D_rectas_borde4</i>	Representación de todas las configuraciones en las que el borde (1, 2, 3 y 4 respectivamente) presenta PBG en $\lambda=1550$ nm y de las posibles anomalías presentes		

**Tabla 6:** Resumen de los inputs y outputs de la función 'creator\_mapacolor\_3D\_rectas.m'

Los resultados para los diferentes bordes de PBG se muestran en las siguientes ilustraciones: Figura 24, Figura 25, Figura 26 y Figura 27. En ellas se observa como para cada borde de PBG se crea un "plano de líneas" de aquellas configuraciones que presentan el PBG en  $\lambda=1550$  nm. Además, se añade en cada representación las anomalías (indicadas en la leyenda) que ese tipo de borde presenta (para todas las configuraciones, no únicamente para aquellas que poseen  $\lambda=1550$  nm). También se representa la leyenda de colores, la cual muestra

el valor exacto de  $\lambda$  que representa la línea dibujada (la tolerancia que se ha elegido es de  $1550 \pm 1 \text{ nm}$ ).

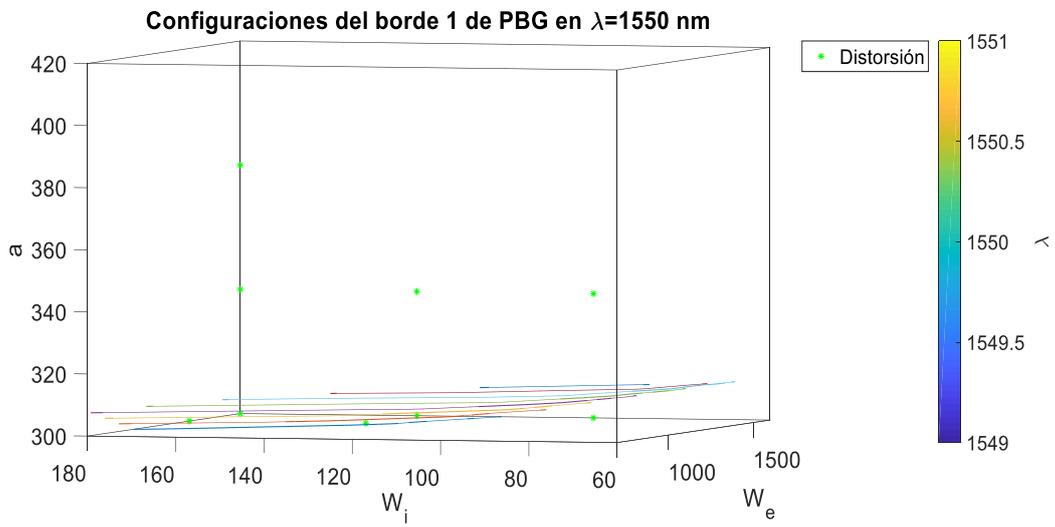


Figura 24: Representación de todas las configuraciones en las que el borde 1 presenta PBG en  $\lambda=1550 \text{ nm}$

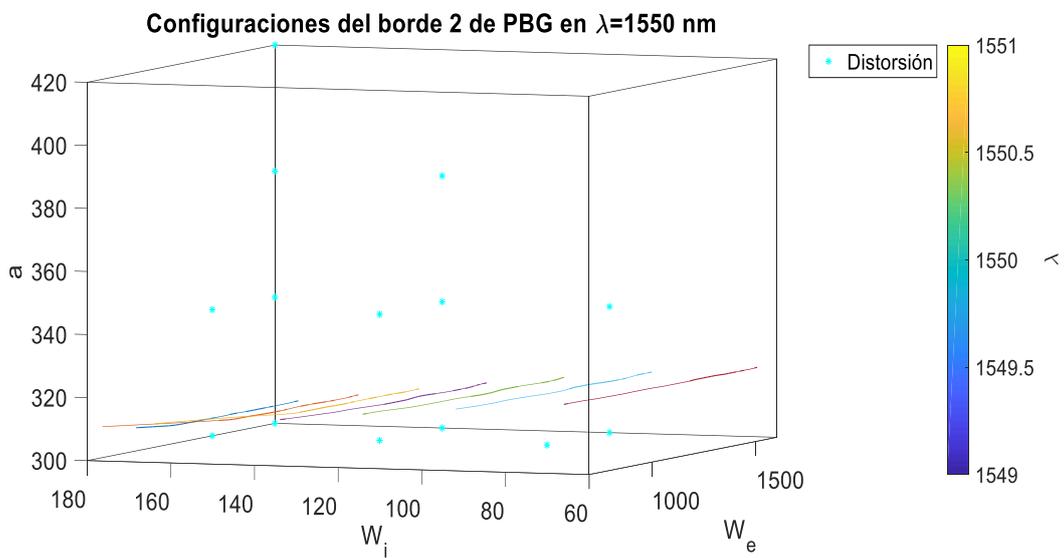


Figura 25: Representación de todas las configuraciones en las que el borde 2 presenta PBG en  $\lambda=1550 \text{ nm}$

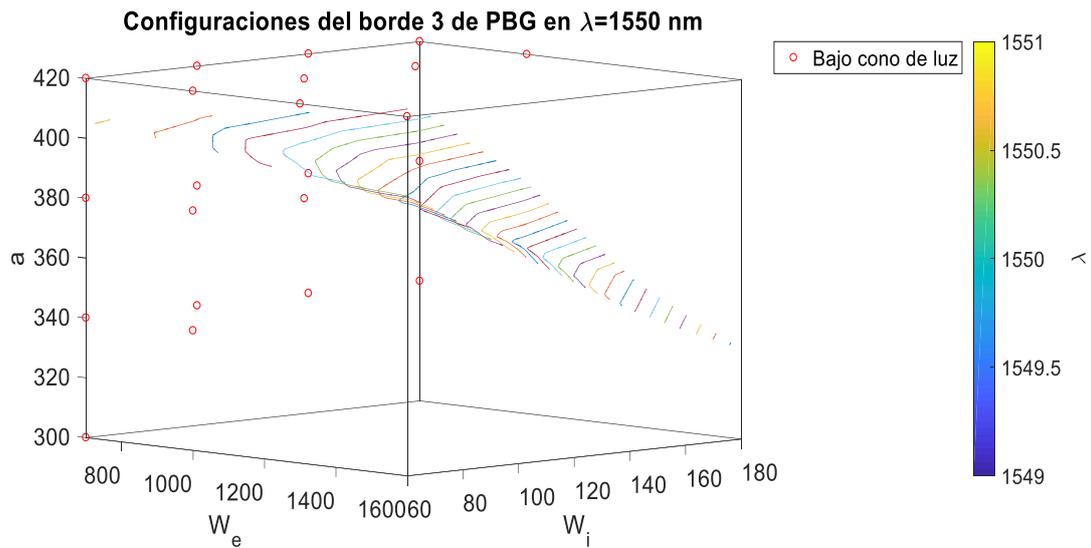


Figura 26: Representación de todas las configuraciones en las que el borde 3 presenta PBG en  $\lambda=1550$  nm

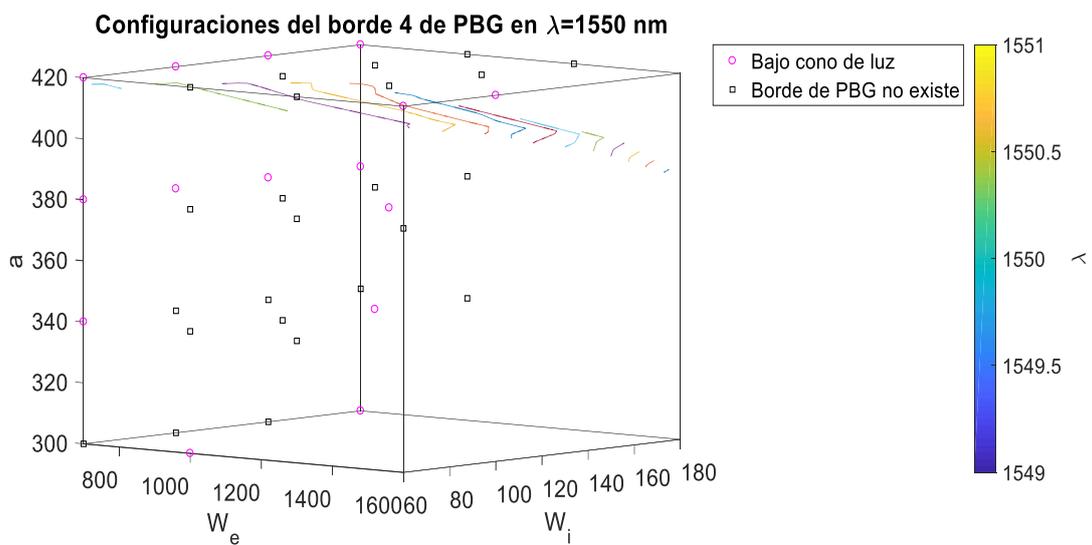


Figura 27: Representación de todas las configuraciones en las que el borde 4 presenta PBG en  $\lambda=1550$  nm

No obstante, vistos los resultados obtenidos con *'creator\_mapacolor\_3D\_rectas.m'* se decide mejorar la representación en búsqueda del plano que forman todas las rectas. Esta mejora se consigue con el diseño de la función *'creator\_mapacolor\_3D\_plano1550.m'*.

Los pasos del algoritmo de esta función son los mismos que los explicados para *'creator\_mapacolor\_3D\_rectas.m'* excluyendo la representación de las líneas y añadiendo el siguiente punto:

- Para la obtención del plano donde los bordes de PBG se encuentran en  $\lambda=1550$  nm:

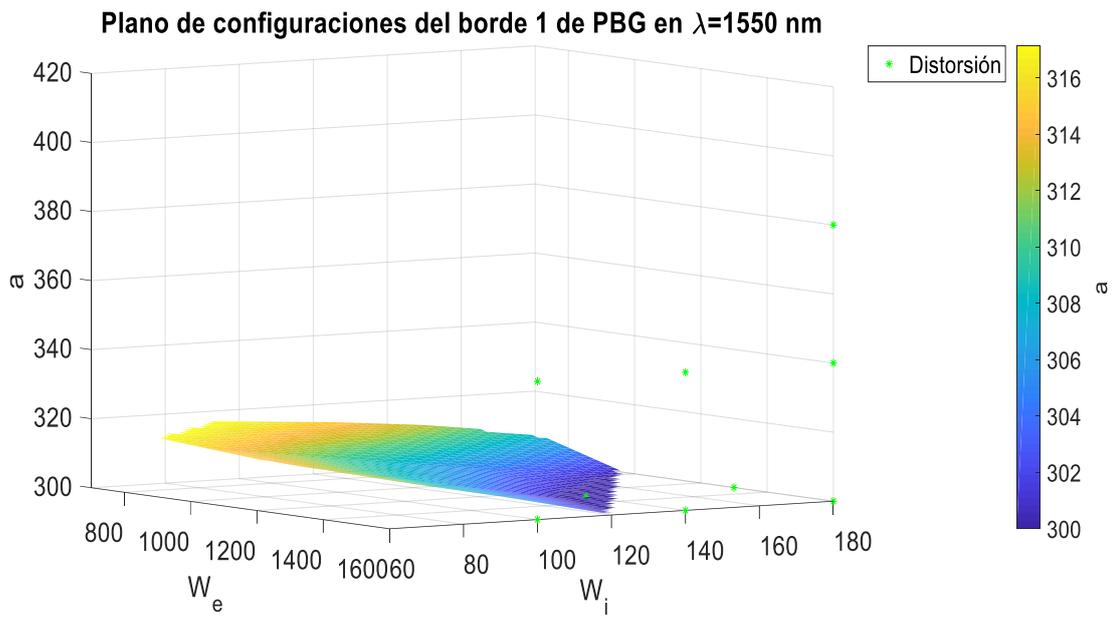
1. Se interpolan los valores de las configuraciones de bordes de PBG en  $\lambda=1550$  nm y se vuelve a crear un mallado con éstas para crear un plano de estas configuraciones (en lugar de un conjunto de rectas que forman un plano como sucede en la función '*creator\_mapacolor\_3D\_rectas.m*')
2. Representación del mallado creado en el paso anterior

Las variables de entrada y salida que presenta esta función se explican en la Tabla 7.

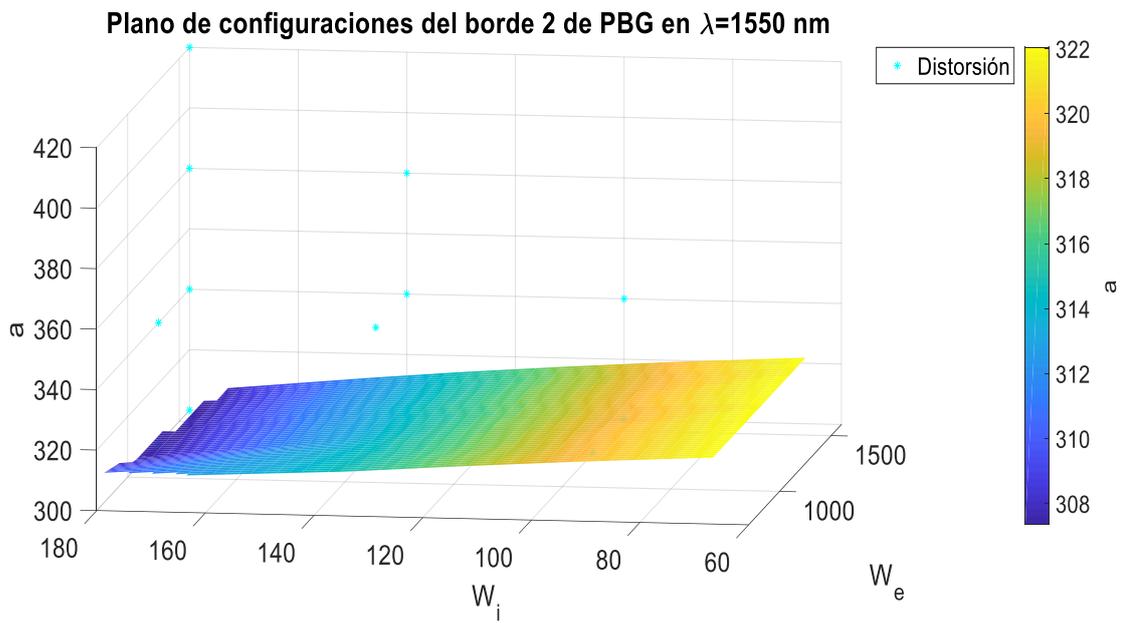
Función	
<i>creator_mapacolor_3D_plano1550.m</i>	
Definición	
Variable de entrada	Coincide con las variables de salida de ' <i>creator_matrizBG_3D.m</i> ' explicadas en la Tabla 5
Variable de salida	<i>mapacolor3D_plano_borde1</i> <i>mapacolor3D_plano_borde2</i> <i>mapacolor3D_plano_borde3</i> <i>mapacolor3D_plano_borde4</i>
	Representación del plano de todas las configuraciones en las que el borde (1, 2, 3 y 4 respectivamente) presenta PBG en $\lambda=1550$ nm y de las posibles anomalías presentes
	<i>borde1_dist</i> <i>borde2_dist</i> <i>borde3_dist</i> <i>borde4_dist</i>
	Matriz que contiene las coordenadas de los bordes 1, 2, 3 y 4 respectivamente que presentan distorsión
	<i>borde3_bajo_cono</i> <i>borde4_bajo_cono</i>
	Matriz que contiene las coordenadas de los bordes 3 y 4 respectivamente que se encuentran bajo el cono de luz
	<i>borde4_sinBG</i>
	Matriz que contiene las coordenadas donde no existe borde 4 de PBG

**Tabla 7:** Resumen de los inputs y outputs de la función '*creator\_mapacolor\_3D\_plano1550.m*'

Los resultados generados con esta función se muestran a continuación en las ilustraciones: Figura 28, Figura 29, Figura 30 y Figura 31.



**Figura 28:** Representación del plano de todas las configuraciones en las que el borde 1 presenta PBG en  $\lambda=1550$  nm



**Figura 29:** Representación del plano de todas las configuraciones en las que el borde 2 presenta PBG en  $\lambda=1550$  nm

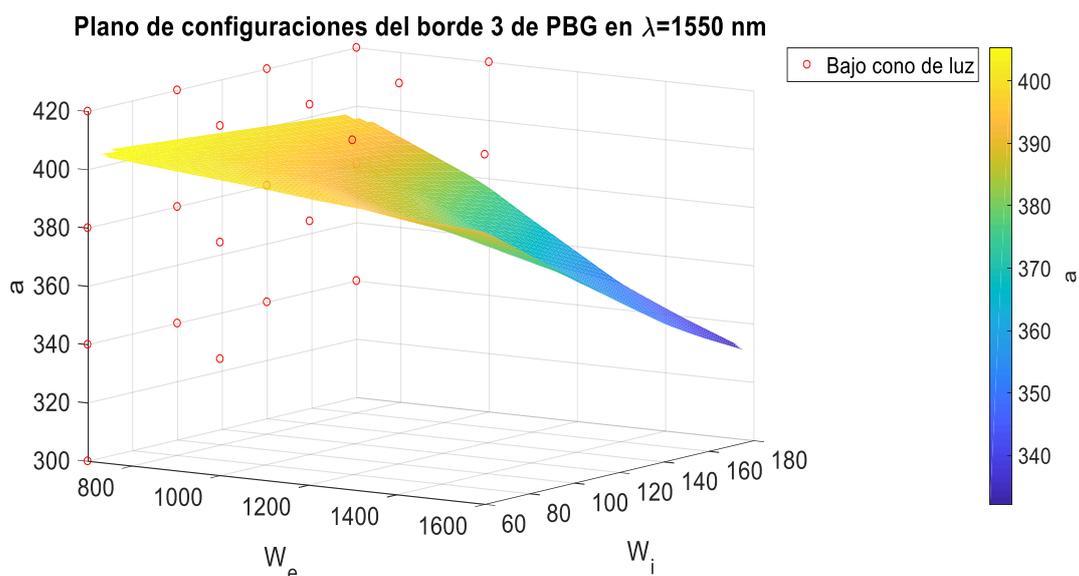


Figura 30: Representación del plano de todas las configuraciones en las que el borde 3 presenta PBG en  $\lambda=1550$  nm

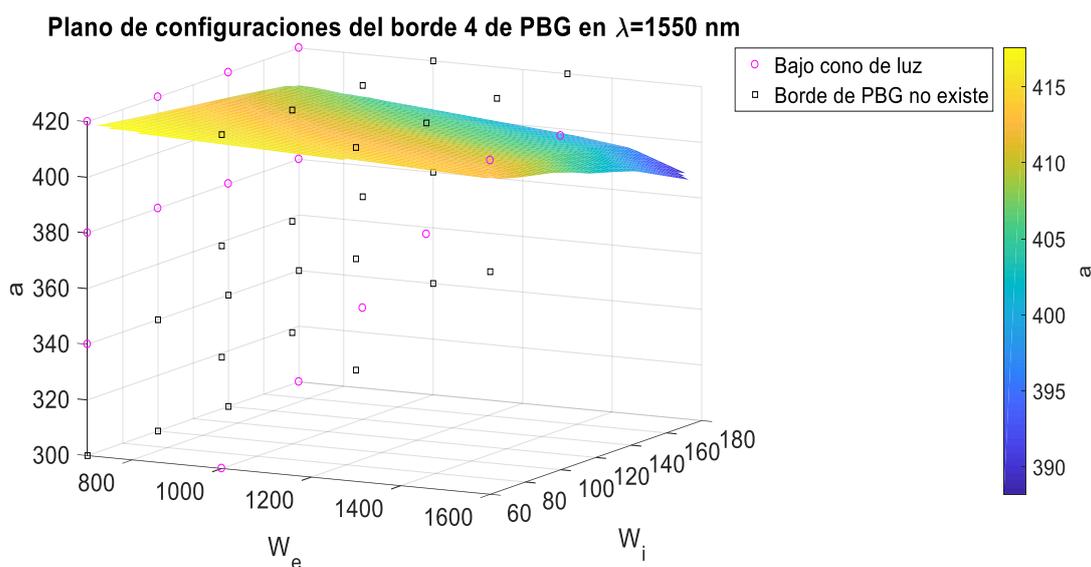


Figura 31: Representación del plano de todas las configuraciones en las que el borde 4 presenta PBG en  $\lambda=1550$  nm

Así, mediante la interpolación lineal de las rectas generadas con la primera función se obtienen cuatro planos que indican todas las configuraciones de interés. También se añaden marcas que indican los bordes que presentan anomalías. En este caso la leyenda de color únicamente indica el valor de la periodicidad de cada estructura.

En relación con el sistema general desarrollado cabe destacar que éste únicamente ejecutará la función *'creator\_mapacolor\_3D\_plano1550.m'* ya que aporta información más completa que *'creator\_mapacolor\_3D\_rectas.m'*.

### 3.4.2 Representación de anomalías en PBG

Hasta ahora los puntos conflictivos presentes en los bordes de PBG se han presentado como puntos inconexos presentes en las figuras 3D. Estos puntos representan anomalías originales presentes en las estructuras que se han simulado, es decir, aquellas anomalías detectadas en '*matrizBG*'. Sin embargo, al igual que sucede en el caso anterior donde las líneas 'forman' en su conjunto un plano, se puede interpolar los puntos que presentan anomalías para obtener, de manera aproximada, aquellos puntos que generan distorsión en las configuraciones que no se han simulado con MPB.

La función '*creator\_mapacolor\_3D\_vol\_dist.m*' consigue este objetivo, es decir, obtiene el volumen de configuraciones que presentan una respuesta no deseada a partir de las coordenadas de los puntos que presentan anomalías. Además, superpone este volumen con los planos de los bordes de PBG donde  $\lambda=1550$  nm facilitando así, de manera visual, la detección de la zona del plano que presentará anomalías.

El algoritmo que presenta '*creator\_mapacolor\_3D\_vol\_dist.m*' sigue los siguientes pasos:

- Para la obtención del plano de bordes de PBG en  $\lambda=1550$  nm

Los pasos que se deben seguir para la obtención del plano coinciden con los del algoritmo de la función '*creator\_mapacolor\_3D\_plano1550.m*' localizada en el apartado 3.4.1

- Para la representación de las coordenadas originales (las que se han simulado con MPB) de los bordes de PBG que presentan anomalías:

1. Se utilizan los inputs de la función. Las posiciones de cada anomalía se encuentran almacenadas en matrices por lo que únicamente hay que representarlas

- Para la creación y representación del volumen de anomalías:

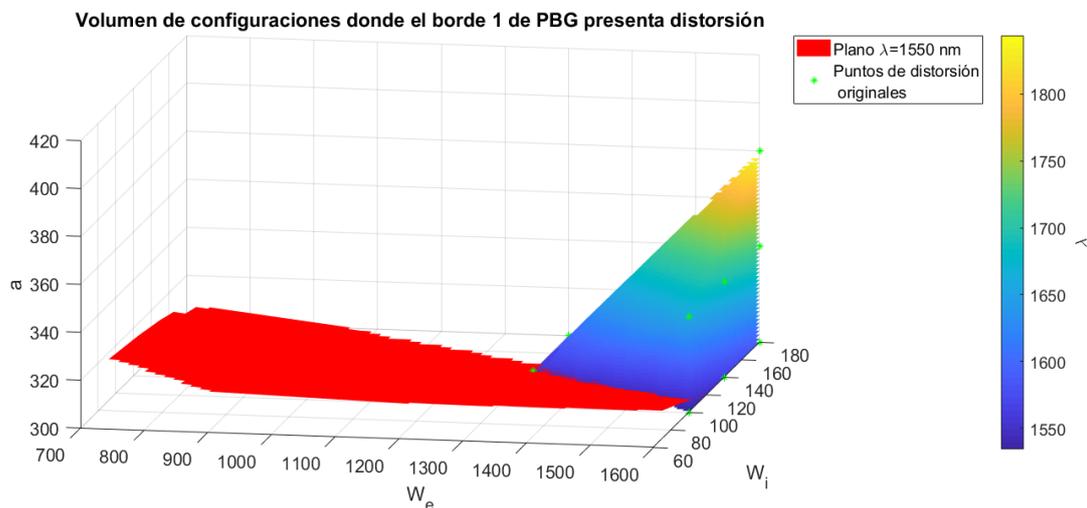
2. Se vuelve a realizar el mallado en este caso de los datos obtenidos en el paso 1
3. A partir de estos valores se crea el volumen de anomalías mediante la utilización de 'slices'. Es decir, se crean planos que mantienen el valor de ' $\alpha$ ' constante y en función de la variación de ' $w_e$ ' y ' $w_i$ ' para este plano se rellenan los valores de las longitudes de onda que presentan anomalías (excepto en el caso donde el borde 4 no existe, en este la representación del volumen se realiza en un único color aleatorio).

Las variables de entrada y salida de esta función se muestran en la Tabla 8.

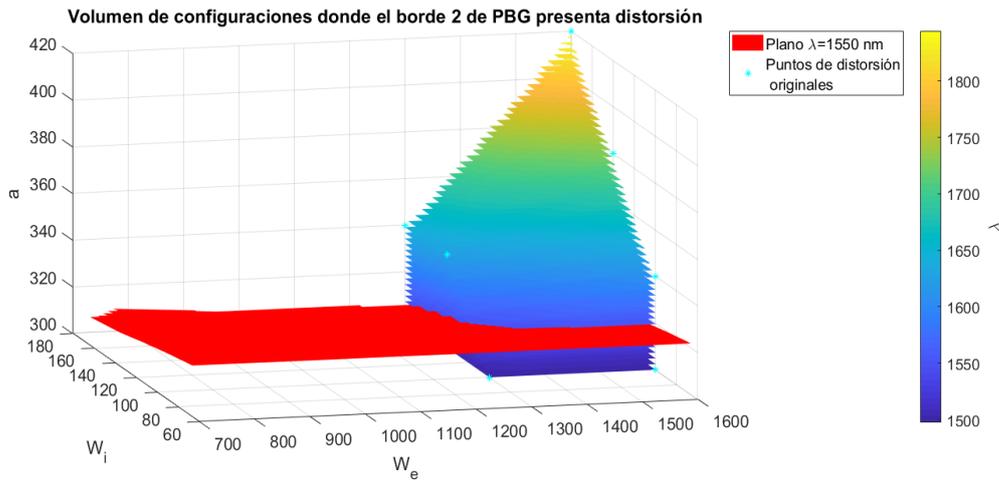
Función		
<i>creator_mapacolor_3D_vol_dist.m</i>		
Definición		
<b>Variable de entrada</b>	Coincide con las variables de salida de ' <i>creator_matrizBG_3D.m</i> ' explicadas en la Tabla 5 y con las variables de salida de ' <i>creator_mapacolor_3D_plano1550.m</i> ' (excepto las representaciones gráficas) explicadas en la Tabla 7	
<b>Variable de salida</b>	<i>mapacolor3D_vol_dist1</i> <i>mapacolor3D_vol_dist2</i> <i>mapacolor3D_vol_dist3</i> <i>mapacolor3D_vol_dist4</i>	Representación del volumen de configuraciones que presenta distorsión en el borde 1, 2, 3 y 4 respectivamente superpuesto al plano donde dicho borde de PBG se encuentra en $\lambda=1550$ nm
	<i>mapacolor3D_vol_bajoCono3</i> <i>mapacolor3D_vol_bajoCono4</i>	Representación del volumen de configuraciones que presenta el borde 3 y 4 bajo el cono de luz respectivamente superpuesto al plano donde dicho borde de PBG se encuentra en $\lambda=1550$ nm
	<i>mapacolor3D_vol_sinBG4</i>	Representación del volumen de configuraciones que no presenta el borde 4 superpuesto al plano donde dicho borde de PBG se encuentra en $\lambda=1550$ nm

**Tabla 8:** Resumen de los inputs y outputs de la función '*creator\_mapacolor\_3D\_vol\_dist.m*'

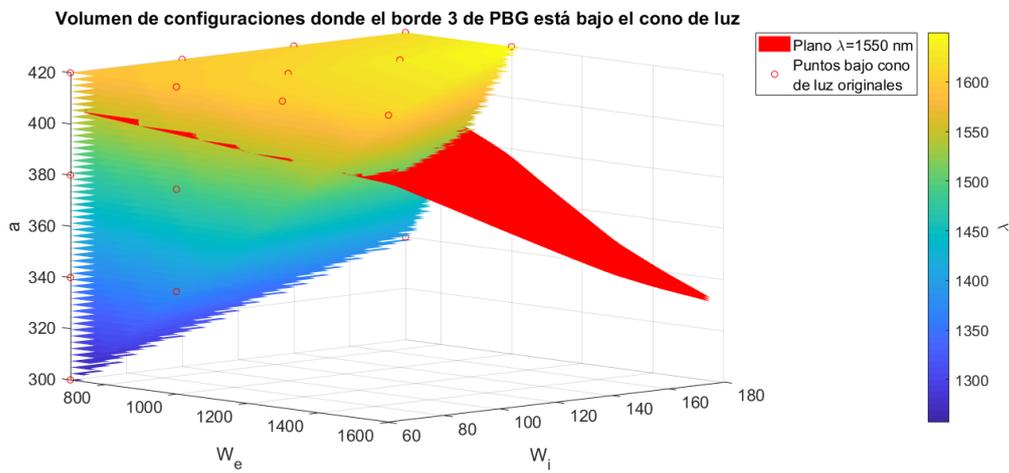
Las figuras obtenidas mediante este algoritmo se muestran a continuación.



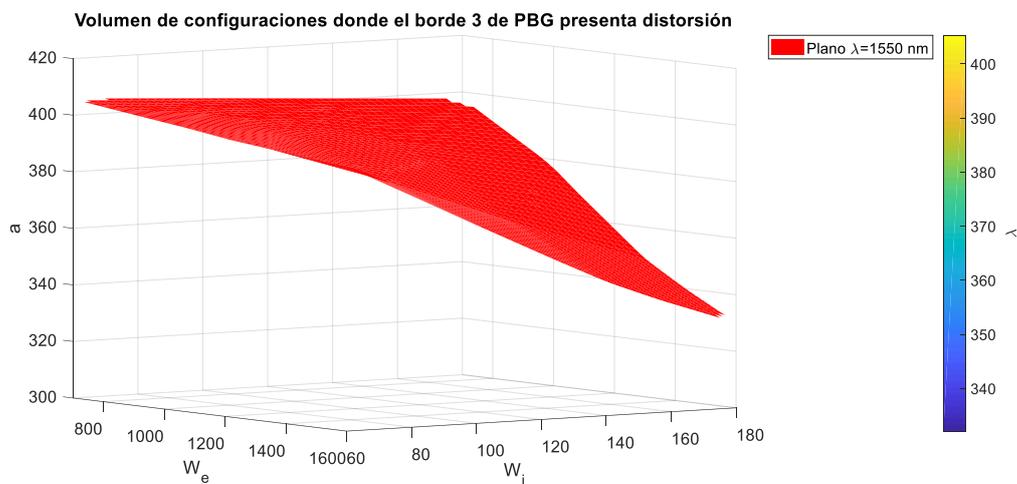
**Figura 32:** Representación del volumen de configuraciones que presenta distorsión en el borde 1 superpuesto al plano donde el borde 1 de PBG se encuentra en  $\lambda=1550$  nm



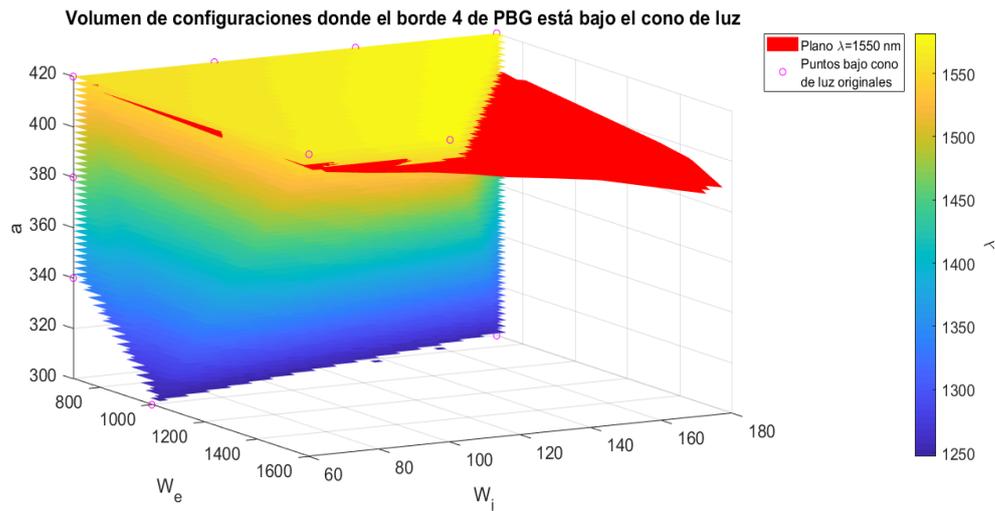
**Figura 33:** Representación del volumen de configuraciones que presenta distorsión en el borde 2 superpuesto al plano donde el borde 2 de PBG se encuentra en  $\lambda=1550$  nm



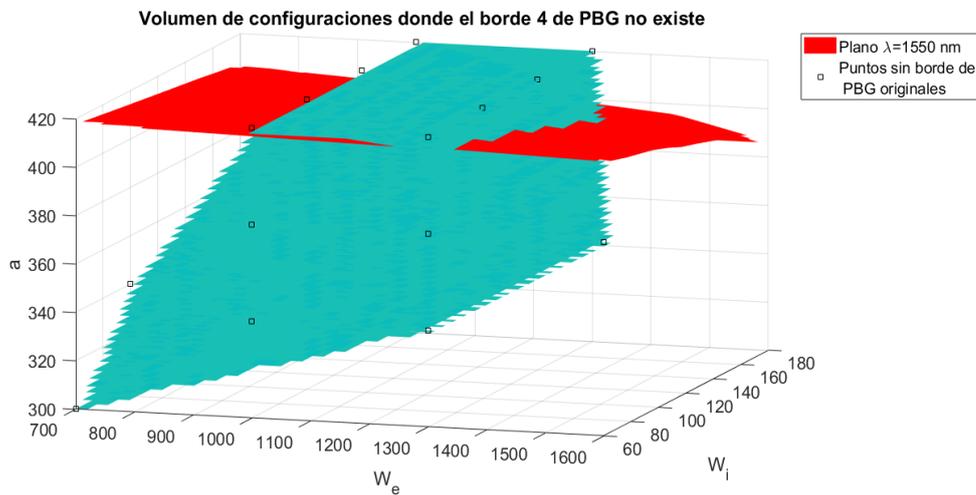
**Figura 34:** Representación del volumen de configuraciones que presenta el borde 3 de PBG bajo el cono de luz superpuesto al plano donde el borde 3 de PBG se encuentra en  $\lambda=1550$  nm



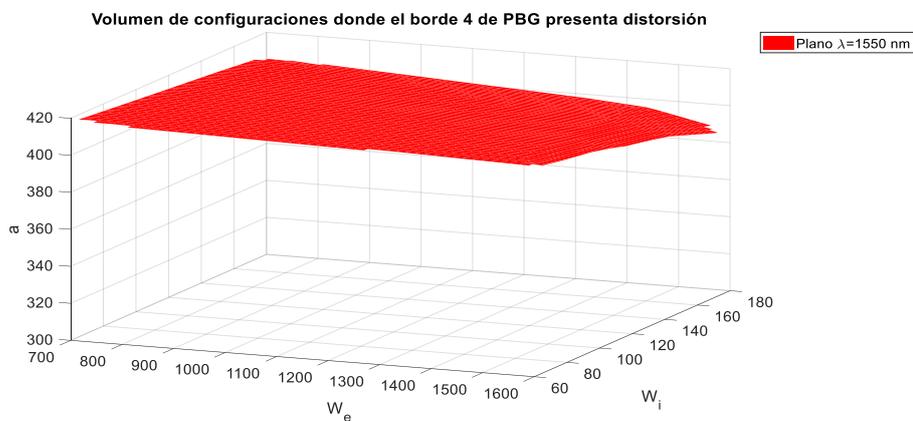
**Figura 35:** Inexistencia de volumen de configuraciones que presenta distorsión para el borde 3 de PBG por lo que únicamente se representa el plano donde el borde 3 de PBG se encuentra en  $\lambda=1550$  nm



**Figura 36:** Representación del volumen de configuraciones que presenta el borde 4 de PBG bajo el cono de luz superpuesto al plano donde el borde 4 de PBG se encuentra en  $\lambda=1550$  nm



**Figura 37:** Representación del volumen de configuraciones que no presenta borde 4 de PBG superpuesto al plano donde el borde 4 de PBG se encuentra en  $\lambda=1550$  nm



**Figura 38:** Inexistencia de volumen de configuraciones que presenta distorsión para el borde 4 de PBG por lo que únicamente se representa el plano donde el borde 4 de PBG se encuentra en  $\lambda=1550$  nm

Observando de la Figura 32 a la Figura 38 se observa que, además de los volúmenes de las anomalías, también se representan los valores originales de éstas a modo de comprobación de la correcta representación del volumen. Además, otra manera de verificar que las representaciones son correctas es analizar los valores de las longitudes de onda del volumen de anomalías cuando se superpone al plano rojo. Así, la intersección se presenta siempre en  $\lambda=1550$  nm. Con esto se comprueba que el plano es correcto. También cabe destacar que en aquellos bordes donde no se presenten anomalías, la representación gráfica únicamente mostrará el plano indicado (sin volumen) como es el caso de la Figura 35 y la Figura 38.

### 3.5 Obtención de la sensibilidad

A partir de la obtención de los planos donde los bordes de PBG se encuentran en  $\lambda=1550$  nm se puede obtener información muy útil a la hora de diseñar estructuras lo más eficientes posible y que presenten el mayor grado de robustez respecto a posibles desviaciones de fabricación. Es decir, que las estructuras presenten la mayor sensibilidad posible (a cambios en el entorno) y que sean tolerantes a la posible inexactitud geométrica de la fabricación.

Para conocer cuáles son las configuraciones que cumplen estas características a continuación se estudia la sensibilidad de las guías corrugadas simuladas y, en el apartado 3.6, se se analiza la tolerancia.

#### 3.5.1 Cálculo y representación para todas las configuraciones

La obtención de la sensibilidad de una estructura requiere analizar las variaciones que ésta sufre cuando se cambian las condiciones de su entorno. Así, en este caso, se estudian los cambios que se generan en la posición del PBG cuando se modifica el medio que se encuentra por encima de la guía corrugada. Es decir, se toman dos medios que presentan índices de refracción parecidos para poder observar los cambios mínimos entre ambos. Los materiales utilizados son agua (índice de refracción=1.333) y  $\text{SiO}_2$  (índice de refracción=1.444). Todas las simulaciones explicadas en los apartados anteriores se han realizado tomando la guía corrugada con agua por encima. Es necesario, por tanto, volver a simular las mismas estructuras cambiando este medio por  $\text{SiO}_2$  (repetir los pasos explicados al principio del apartado 3 hasta conseguir la 'matrizBG' correspondiente a  $\text{SiO}_2$  por encima).

Por otro lado, el cálculo de la sensibilidad se realiza a través de la fórmula siguiente:

$$\text{Sensibilidad} = \frac{\lambda_{\text{SiO}_2} - \lambda_{\text{H}_2\text{O}}}{n_{\text{SiO}_2} - n_{\text{H}_2\text{O}}} \left( \frac{\text{nm}}{\text{RIU}} \right)$$

**Ecuación 6:** Ecuación de la sensibilidad aplicada al caso de simulaciones con  $\text{SiO}_2$  y  $\text{H}_2\text{O}$  por encima de la guía. Los  $\lambda$  representan los diferentes valores de los bordes de PBG para ambos materiales y las  $n$  simbolizan el índice de refracción de cada material. Las unidades en que se mide la sensibilidad son nm/RIU (Refractive Index Unit)

Teniendo todo esto en cuenta se diseña la función '*creator\_vol\_sensibilidad.m*' que genera el volumen de sensibilidades para todos los bordes de PBG. Los pasos que sigue el algoritmo de esta función son:

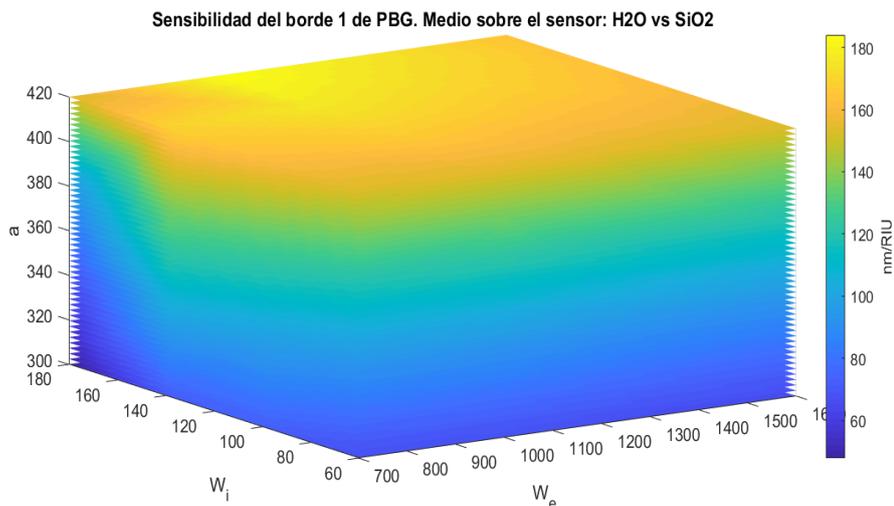
1. Definición de las constantes dieléctricas utilizadas
2. Definición de los vectores '*a\_vector*', '*wi\_vector*' y '*we\_vector*' mediante interpolación y mallado de estos vectores
3. Cálculo de la sensibilidad aplicando la Ecuación 6 sobre cada borde de PBG
4. Creación de una matriz que contiene todos los resultados de sensibilidad obtenidos en el paso 3
5. Para esta matriz se elimina los valores que presentan 'NaN' como sensibilidad y mallado de esta matriz (se genera una matriz en la que cada valor de sensibilidad se corresponde con una combinación de configuraciones diferente)
6. Representación de la sensibilidad mediante 'slices'. Entre cada slice se realiza una pausa para poder observar la evolución de la sensibilidad.

Las variables de entrada y salida de esta función se explican a continuación (Tabla 9).

Función		
<i>creator_vol_sensibilidad.m</i>		
		Definición
<b>Variable de entrada</b>	<i>matrizBG_SiO2</i> <i>matrizBG_H2O</i>	Matriz que contiene todos los datos originales de la simulación con SiO <sub>2</sub> y H <sub>2</sub> O respectivamente por encima
<b>Variable de salida</b>	<i>vol4D_sensibilidad_borde1</i> <i>vol4D_sensibilidad_borde2</i> <i>vol4D_sensibilidad_borde3</i> <i>vol4D_sensibilidad_borde4</i>	Representación del volumen de sensibilidad en el borde 1, 2, 3 y 4 de PBG respectivamente

**Tabla 9:** Resumen de los inputs y outputs de la función '*creator\_vol\_sensibilidad.m*'

Las gráficas que genera esta función se pueden observar en las siguientes ilustraciones:



**Figura 39:** Representación del volumen de sensibilidad en el borde 1 de PBG

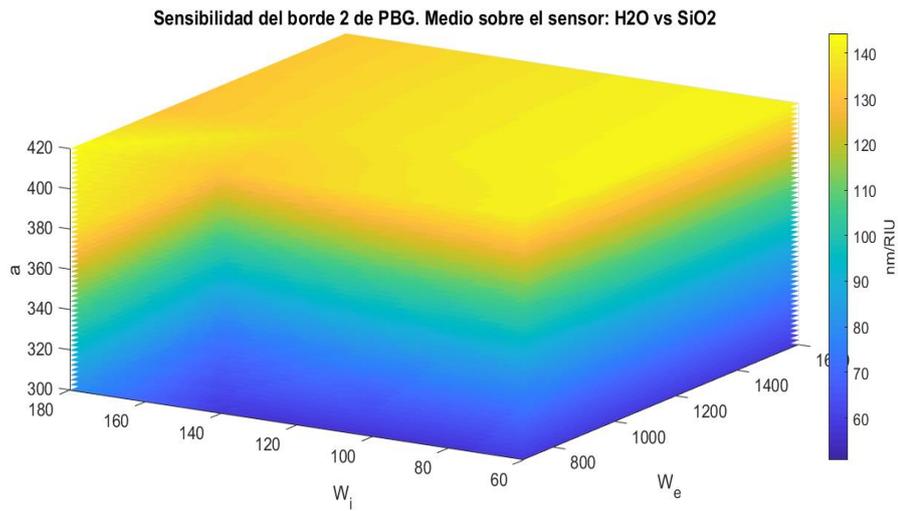


Figura 40: Representación del volumen de sensibilidad en el borde 2 de PBG

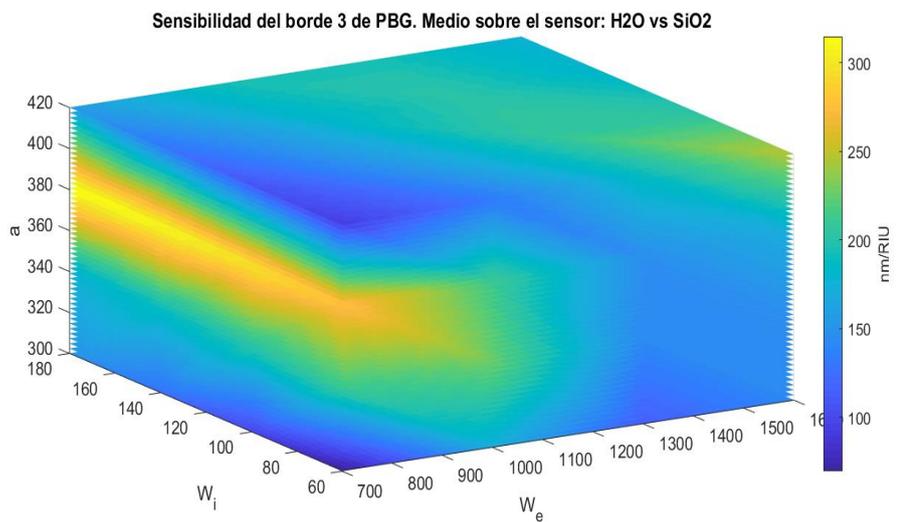


Figura 41: Representación del volumen de sensibilidad en el borde 3 de PBG

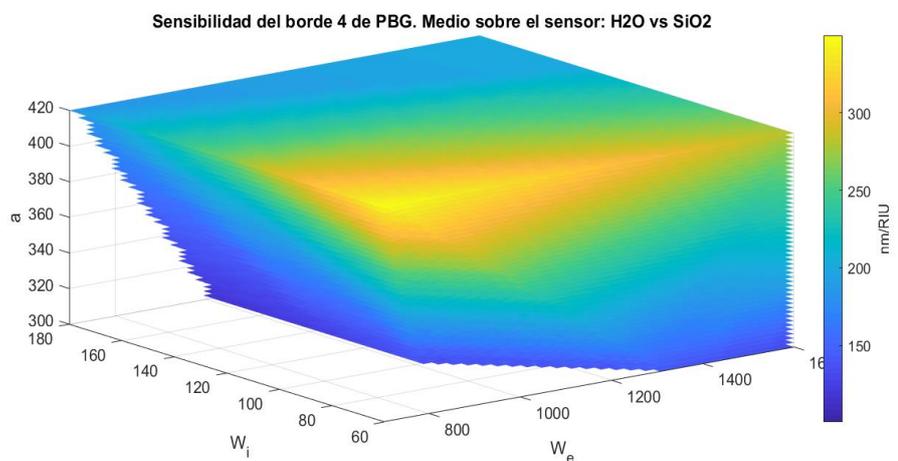


Figura 42: Representación del volumen de sensibilidad en el borde 4 de PBG

A partir de las gráficas anteriores (Figura 39, Figura 40, Figura 41 y Figura 42) se observa como los colores más cálidos representan una mayor sensibilidad a cambios en el medio. Así, los mayores valores de sensibilidad se presentan para el borde 3 y 4 de PBG. Las configuraciones que no presentan valor de sensibilidad en la Figura 42 marcan aquellos valores donde el borde 4 no existe.

Por otro lado, cabe destacar que en estas figuras se ha representado todos los 'slices' para cada borde de PBG formando así 'cubos de sensibilidades'. En estos, no se puede apreciar aquellos valores de sensibilidad que poseen las configuraciones localizadas en la zona 'interna' de cada cubo si no únicamente aquellas que se encuentran en las superficies más 'externas' de estos. Sin embargo, cuando se ejecuta esta función en Matlab sí se pueden ver estos valores 'internos' puesto que entre la representación de un slice y el siguiente se añade una 'pausa' controlada de manera manual.

### 3.5.2 Representación para el plano de PBG en $\lambda=1550$ nm

Los volúmenes de sensibilidad obtenidos en el subapartado anterior son de gran ayuda para poder observar de manera general la evolución de la sensibilidad a lo largo de todas las configuraciones. Sin embargo, los valores de sensibilidad que realmente son de interés conocer son los de las configuraciones que forman el plano de borde de PBG en  $\lambda=1550$  nm. Con esta finalidad se diseña la función '*creator\_mapacolor\_4D\_sensibilidad.m*'.

Los pasos que sigue el algoritmo de esta función son los siguientes:

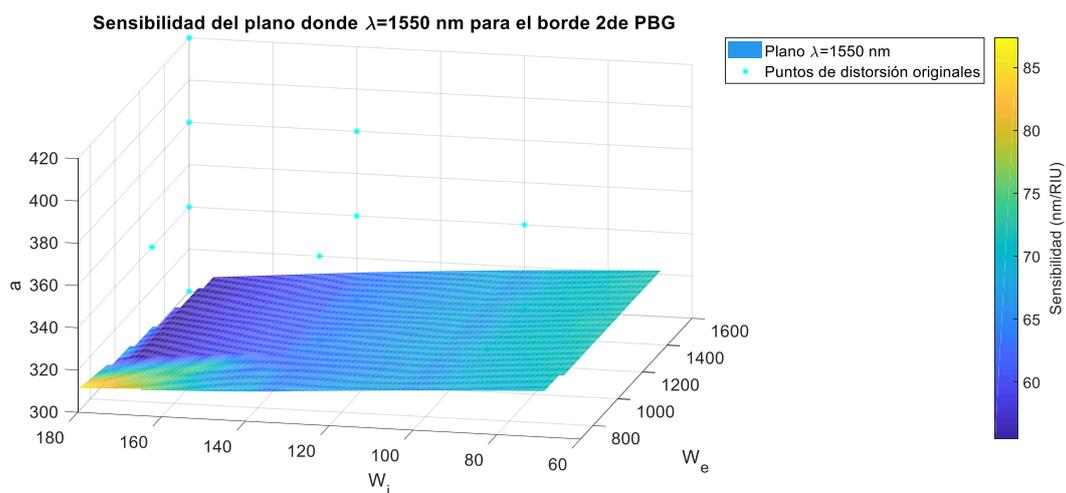
1. Para la obtención de los datos que forman el plano de PBG en  $\lambda=1550$  nm se realizan los pasos ya explicados en el apartado 3.4.1 del algoritmo de la función '*creator\_mapacolor\_3D\_plano1550.m*'. Estos pasos se realizan para la matriz '*matrizBG\_H2O*'
2. Se eliminan aquellos valores de los bordes que sean 'NaN' de '*matrizBG\_SiO2*' para evitar problemas de funcionamiento en el desarrollo del algoritmo y se crea una nueva matriz con estos datos sin 'NaN'
3. Se realiza el mallado de las configuraciones de PBG presentes en la matriz creada en el paso anterior únicamente para aquellas configuraciones que estén presentes en el plano de los bordes de PBG obtenidos en el paso 1
4. Se calcula la sensibilidad a partir de la matriz generada en el paso 3 y aplicando la fórmula presente en la Ecuación 6.
5. Se representa el plano donde los bordes de PBG se encuentran en  $\lambda=1550$  nm añadiendo en la cuarta dimensión (leyenda de color) los valores de sensibilidad para cada configuración del plano
6. Se representan además los puntos originales de bordes de PBG que presentan anomalías

Las variables de entrada que presenta esta función se muestran en la tabla siguiente (Tabla 10).

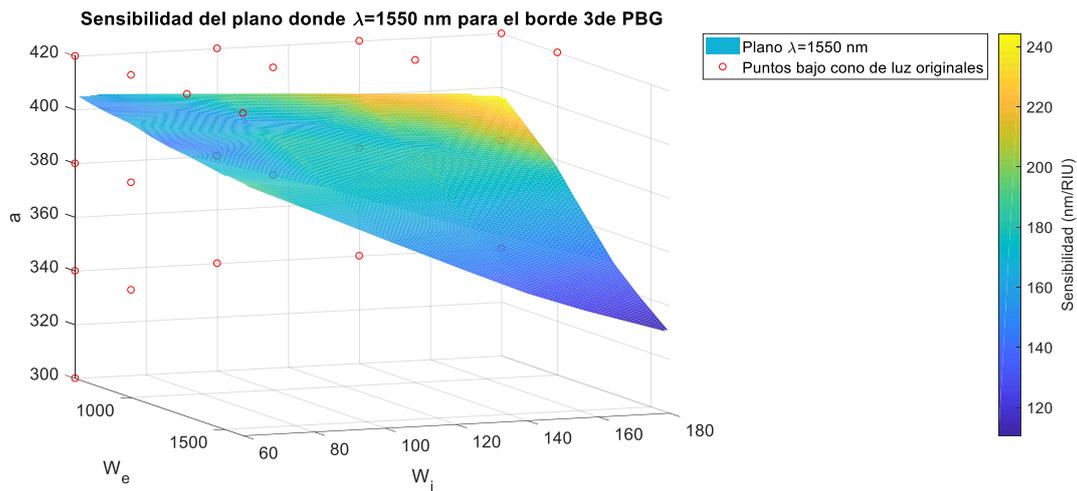
Función		
<i>creator_mapacolor_4D_sensibilidad.m</i>		
		Definición
Variable de entrada	<i>matrizBG</i> <i>matrizBG_SiO2</i>	Matriz que contiene todos los datos originales de la simulación con H <sub>2</sub> O y SiO <sub>2</sub> por encima respectivamente
	Las variables de salida de la Tabla 7 (excepto las referidas a representaciones gráficas)	
Variable de salida	<i>mapacolor4D_plano1_sens</i> <i>mapacolor4D_plano2_sens</i> <i>mapacolor4D_plano3_sens</i> <i>mapacolor4D_plano4_sens</i>	Representación de la sensibilidad de los valores del plano donde $\lambda=1550$ nm para los bordes 1, 2, 3 y 4 de PBG respectivamente

**Tabla 10:** Resumen de los inputs y outputs de la función '*creator\_mapacolor\_4D\_sensibilidad.m*'

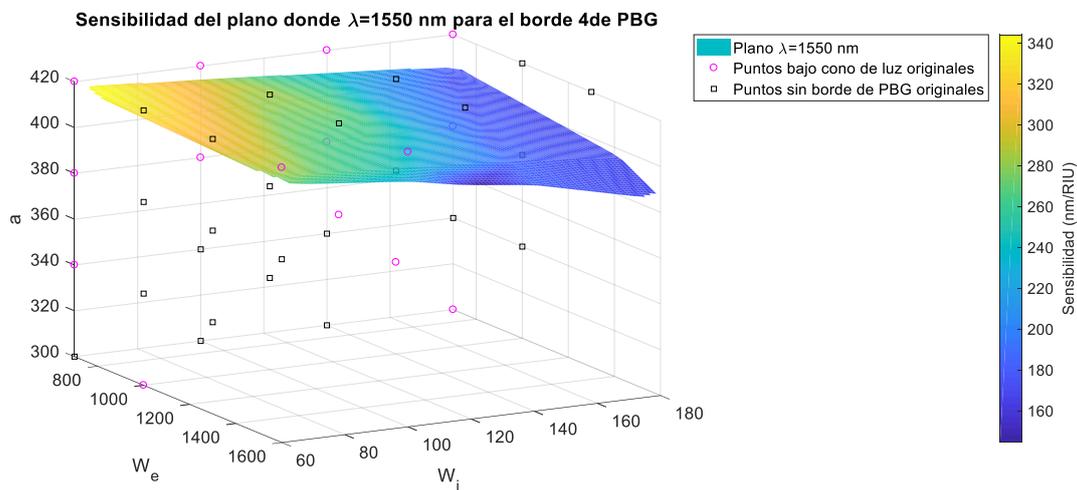
Las representaciones que se obtienen de esta función se muestran a continuación.



**Figura 43:** Representación de la sensibilidad de los valores del plano donde  $\lambda=1550$  nm para los bordes 2 de PBG



**Figura 44:** Representación de la sensibilidad de los valores del plano donde  $\lambda=1550$  nm para los bordes 3 de PBG



**Figura 45:** Representación de la sensibilidad de los valores del plano donde  $\lambda=1550$  nm para los bordes 4 de PBG

Cabe destacar que la figura que debería representar el plano para el borde 1 de PBG no se añade por no presentar datos concluyentes debido a un fallo desconocido. Los valores de sensibilidad para este plano se estimarán a partir del volumen de sensibilidades obtenido en el apartado 3.5.1 para el borde 1. El resto de planos, con las sensibilidades correspondientes a los bordes 2,3 y 4 respectivamente se representan en las figuras: Figura 43, Figura 44 y Figura 45 junto a los puntos originales que presentan distorsión.

## 3.6 Obtención de la tolerancia

### 3.6.1 Cálculo y representación para todas las configuraciones

La obtención de combinaciones de configuraciones que presenten un mayor nivel de tolerancia a desviaciones de fabricación también es un requisito importante a la hora de diseñar una estructura.

Para poder obtener el nivel de tolerancia de cada estructura simulada se decide analizar la variación que sufre cada borde de PBG cuando se varía en una unidad cada parámetro geométrico estudiado respecto a una dimensión determinada. Es decir, en el caso de las guías corrugadas (como se puede ver en la Figura 9) el período se representa a lo largo del eje z, la longitud de los elementos transversales en el eje y y la anchura de los elementos transversales en el eje x. Teniendo esto en cuenta, se estudia la variación del valor de los bordes de PBG respecto a la variaciones de sus parámetros en un eje determinado. Esto se realiza mediante el cálculo de las derivadas parciales primeras de los valores de  $\lambda$  respecto a cada eje de coordenadas. Puesto que las derivadas generan tres respuestas diferentes, éstas se multiplican entre sí para poder observar la variación que sufre un valor de borde de PBG no únicamente respecto a una dimensión si no respecto a las tres. La fórmula utilizada se muestra a continuación:

$$\text{Variabilidad} = \frac{\partial \lambda}{\partial x} \cdot \frac{\partial \lambda}{\partial y} \cdot \frac{\partial \lambda}{\partial z}$$

**Ecuación 7:** Fórmula utilizada para la obtención de la tolerancia, término antagonista a variabilidad, de las diferentes estructuras

Es importante tener en cuenta que la variabilidad de las estructuras (obtenida mediante la Ecuación 7) obtiene aquellas zonas que presentan mayor variación cuando se modifica su geometría. Este término es antagonista a la definición de tolerancia por lo que, en las representaciones obtenidas más adelante, los puntos de interés serán aquellos que presenten la variabilidad más baja puesto que son los más tolerantes.

Con esta finalidad se diseña la función '*creator\_vol\_tolerancia.m*' cuyo algoritmo sigue los pasos mostrados a continuación:

1. Se eliminan los valores 'NaN' presentes en los bordes 4 de PBG de 'matrizBG'.
2. A diferencia de las funciones anteriores, en este caso es necesario realizar dos interpolaciones diferentes con sus respectivos mallados. En primer lugar, se realiza una interpolación lineal de los datos originales de simulación y después a partir del mallado resultante de esta interpolación se realiza otra interpolación en este caso cúbica
3. A través de las matrices obtenidas mediante el mallado del paso 1 se calcula el paso 'h' de las derivadas en función de la dimensión estudiada
4. Se calculan las derivadas parciales primeras de los valores de los bordes de PBG respecto a cada dimensión estudiada
5. Se combinan las tres matrices correspondientes a cada derivada parcial obtenidas en el paso 3 empleando la Ecuación 7. Es decir, se obtiene la variabilidad respecto a las tres dimensiones para cada configuración posible
6. Representación mediante 'slices' del volumen de variabilidad (y por ende de tolerancia) de los diferentes bordes de PBG

La interpolación cúbica realizada en el paso 2 tiene como finalidad poder representar correctamente el volumen de tolerancias. Durante el proceso de diseño de esta función se observó que únicamente utilizando la interpolación lineal no se obtenían resultados adecuados

para su correcta interpretación. Esto sucede debido a que, a la hora de realizar las derivadas parciales, si los datos sobre los que se trabaja presentan interpolación lineal, el algoritmo únicamente detecta los cambios de pendiente que suceden con los datos originales de 'matrizBG' por lo que el cálculo y la representación de la tolerancia se basaba en 'bloques de triángulos' de diferentes valores con vértices en los datos originales de la simulación. Sin embargo, si se realizan las derivadas sobre datos que han sido interpolados cúbicamente los cambios de pendiente se realizan sobre todos los puntos que se han interpolado y no únicamente sobre los originales. Es decir, existe cambio de pendiente entre los puntos interpolados lo que permite el cálculo de las derivadas en cada punto calculado cúbicamente consiguiendo, por tanto, unos valores de tolerancia mucho más acertados y una representación mejor que la conseguida con la interpolación lineal. La razón de realizar en el algoritmo de la función 'creator\_vol\_tolerancia' en primer lugar la lineal y después la cúbica es por que la función 'interp3' de Matlab necesita la introducción de datos primeramente interpolados linealmente. Para evitar el problema antes mencionado la interpolación lineal únicamente obtiene los datos originales de simulación por lo que, realmente, no se está realizando interpolación lineal ninguna. En la Figura 46 se muestra la diferencia entre los resultados obtenidos en función de la interpolación elegida.

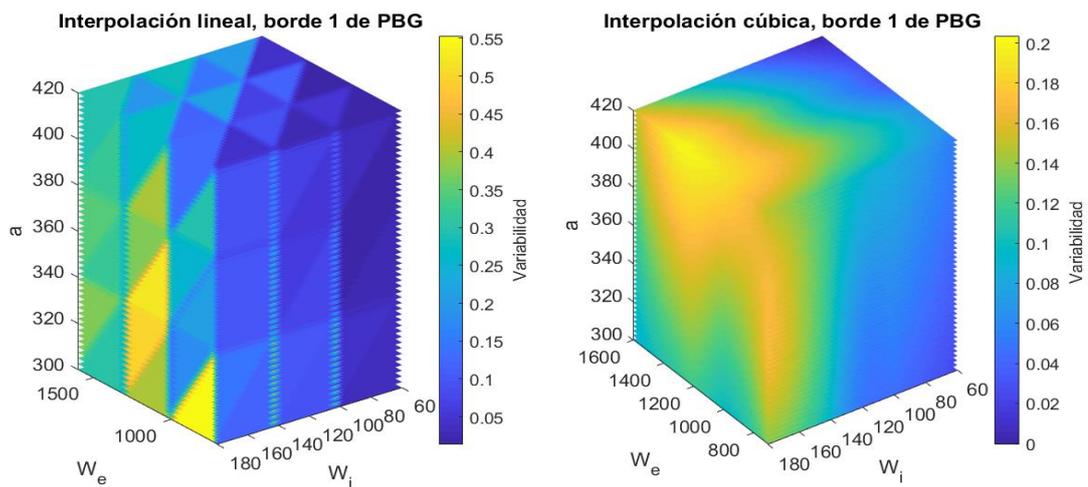


Figura 46: Interpolación lineal (izquierda) vs interpolación cúbica (derecha) de la tolerancia del borde 1 de PBG

Vistos los resultados de cada tipo de interpolación se decide diseñar el algoritmo realizando la interpolación cúbica.

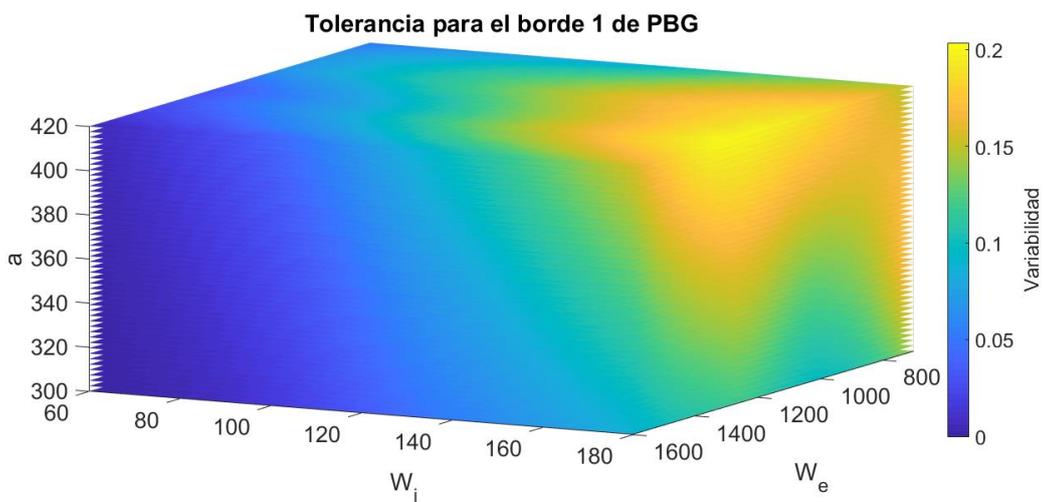
Las variables de entrada y salida que presenta la función 'creator\_vol\_tolerancia' se muestran a continuación (Tabla 11).

Función		
<i>creator_vol_tolerancia.m</i>		
<b>Definición</b>		
<b>Variable de entrada</b>	<i>matrizBG</i>	Matriz que contiene todos los datos originales de la simulación con H <sub>2</sub> O por encima

Variable de salida	$vol4D\_tolerancia\_borde1$ $vol4D\_tolerancia\_borde2$ $vol4D\_tolerancia\_borde3$ $vol4D\_tolerancia\_borde4$	Representación del volumen de tolerancias en el borde 1, 2, 3 y 4 de PBG respectivamente
	$X\_int\_borde1$ $Y\_int\_borde1$ $Z\_int\_borde1$ $W\_int\_borde1$  $X\_int\_borde2$ $Y\_int\_borde2$ $Z\_int\_borde2$ $W\_int\_borde2$  $X\_int\_borde3$ $Y\_int\_borde3$ $Z\_int\_borde3$ $W\_int\_borde3$  $X\_int\_borde4$ $Y\_int\_borde4$ $Z\_int\_borde4$ $W\_int\_borde4$	Matriz de mallado de las coordenadas de 'we', 'wi' y 'a' (para $X\_int$ , $Y\_int$ y $Z\_int$ respectivamente) para aquellas configuraciones donde exista el borde 1, 2, 3 y 4 respectivamente.  $W\_int$ presenta la interpolación cúbica de los valores de $\lambda$ del borde 1, 2, 3 y 4 respectivamente
	$tolerancia\_borde1$ $tolerancia\_borde2$ $tolerancia\_borde3$ $tolerancia\_borde4$	Tolerancia que presentan los valores de $\lambda$ del borde 1, 2, 3 y 4 respectivamente

**Tabla 11:** Resumen de los inputs y outputs de la función 'creator\_vol\_tolerancia.m'

Las gráficas de tolerancia obtenidas mediante esta función se muestran a continuación:



**Figura 47:** Representación del volumen de tolerancias en el borde 1 de PBG

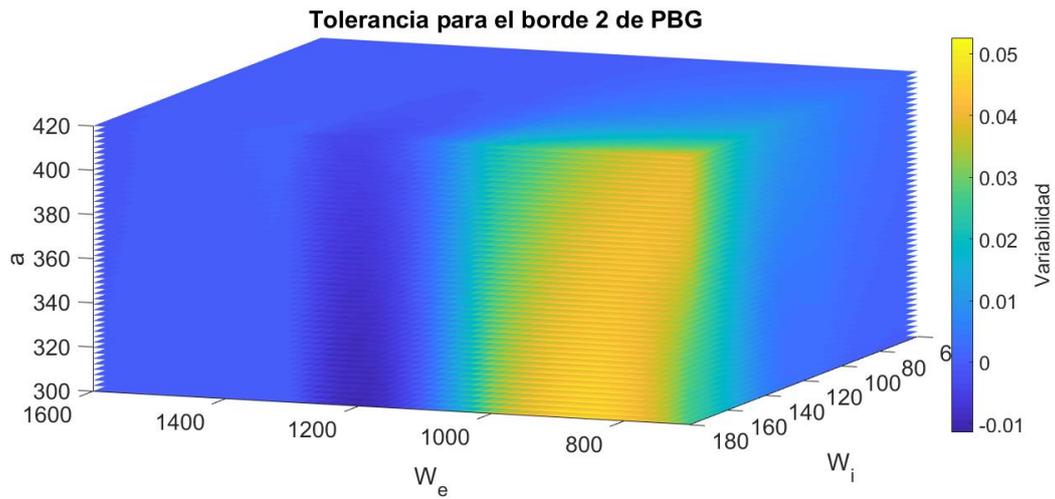


Figura 48: Representación del volumen de tolerancias en el borde 2 de PBG

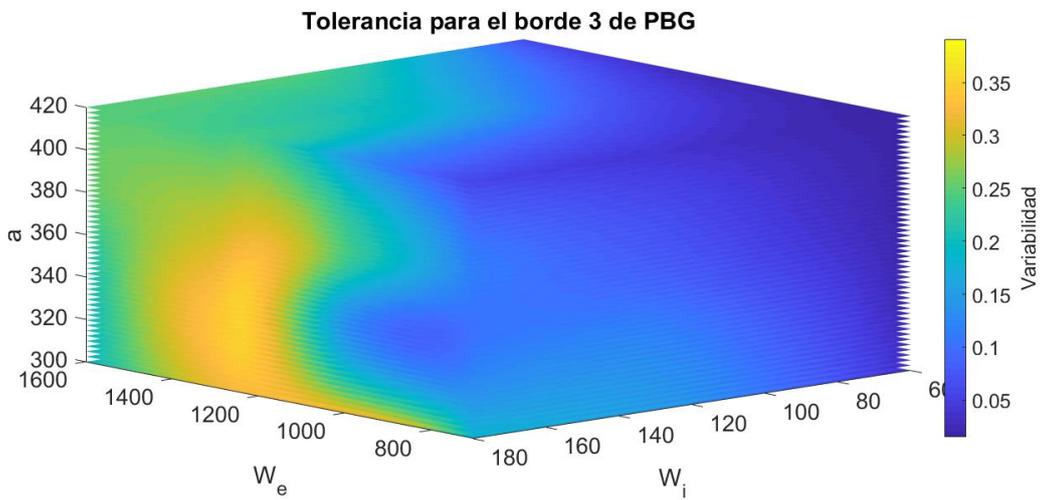


Figura 49: Representación del volumen de tolerancias en el borde 3 de PBG

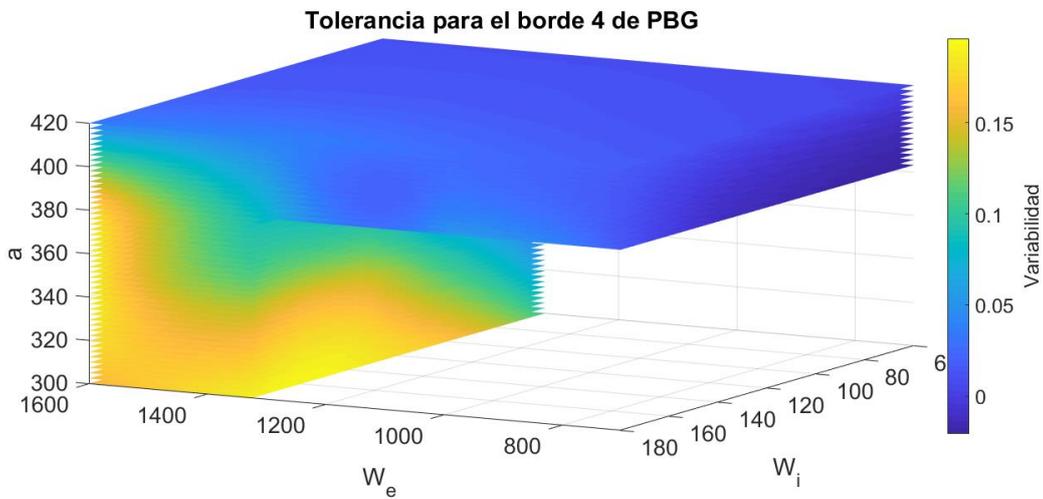


Figura 50: Representación del volumen de tolerancias en el borde 4 de PBG

En las figuras mostradas arriba ( Figura 47, Figura 48, Figura 49 y Figura 50) se muestra en tonos más oscuros aquellas configuraciones que presentan una mayor tolerancia a desviaciones dimensionales durante la fabricación. Los tonos más claros muestran aquellas donde la variabilidad es máxima y, por tanto, la tolerancia es mínima.

Al igual que sucede en el apartado 3.5.1 en la representación de las sensibilidades, la representación del volumen de tolerancias se realiza mediante 'slices' de manera que no se pueden observar los valores que forman la parte 'interna' del cubo de tolerancias si no únicamente las tolerancias de aquellas configuraciones que se encuentran en la parte más 'externa' del cubo. En su representación en Matlab mediante el comando 'pause' se pueden observar dichos valores internos de cada 'slice' que forma el cubo.

### 3.6.2 Representación para el plano de PBG en $\lambda=1550$ nm

De la misma forma que sucede en el apartado 3.5.1, la representación del volumen de tolerancias obtenido en el apartado 3.6.1 para cada banda de PBG presenta una visión general. Sin embargo, es más útil conocer los valores de tolerancia que se presentan únicamente para el plano de bordes de PBG en  $\lambda=1550$  nm puesto que ésta es la longitud de onda que se desea estudiar. Con esta finalidad se diseña la función '*creator\_mapacolor\_4D\_tolerancia.m*'.

Los pasos del algoritmo de esta función son los siguientes:

1. Se introducen los inputs del sistema '*X\_int*', '*Y\_int*', '*Z\_int*' y '*W\_int*' (contienen todos los valores del mallado e interpolación cúbica realizados en la función '*creator\_vol\_tolerancia.m*')
2. Se redondea la matriz '*W\_int*' para facilitar la búsqueda de datos en ella (si los datos son demasiado exactos son más diferentes entre si). Se buscan las posiciones de esta matriz que presentan  $\lambda=1550$  nm
3. Se extraen los valores presentes en las posiciones obtenidas en el paso 2 para las matrices de mallado '*X\_int*', '*Y\_int*', '*Z\_int*'. Estas tres variables en conjunto marcan las posiciones de todos aquellos bordes donde  $\lambda=1550$  nm
4. A partir de las posiciones obtenidas en el paso 2 se extrae, para estas mismas coordenadas, los valores de la matriz 'tolerancia' (introducida también como input)
5. Se crea un mallado a partir de los valores obtenidos en los pasos 3 y 4 de manera que se obtiene una matriz donde, para cada posición donde el borde de PBG está en  $\lambda=1550$  nm, se coloca su valor de tolerancia correspondiente
6. Se representa el plano añadiendo como cuarta dimensión (leyenda de color) la tolerancia

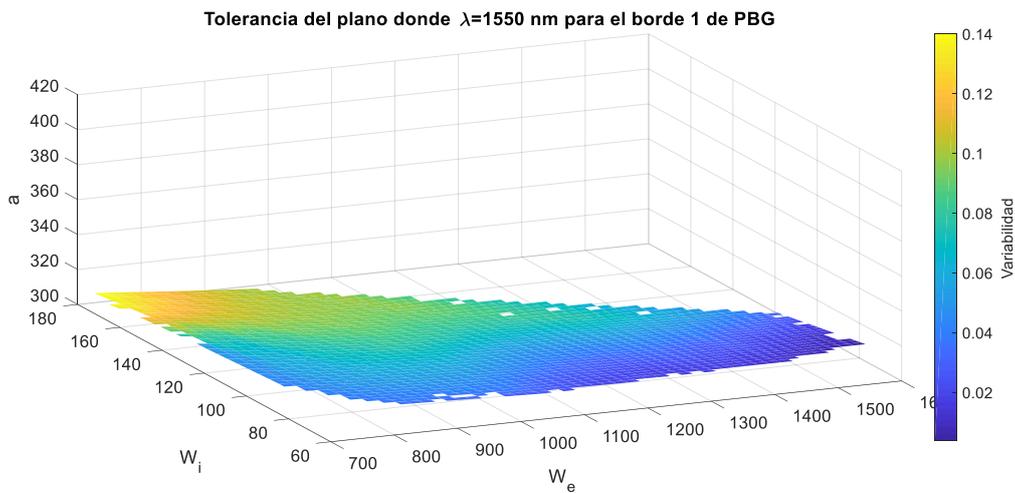
Para evitar complicaciones innecesarias en esta función se decide no representar las posiciones que presentan anomalías. Puesto que al final del desarrollo es necesario comparar los planos de sensibilidades y de tolerancias únicamente con la representación de las anomalías en uno de los dos es suficiente para saber su localización y tenerla en cuenta en la selección de configuraciones.

Las variables de entrada y salida de esta función se muestran a continuación (Tabla 12):

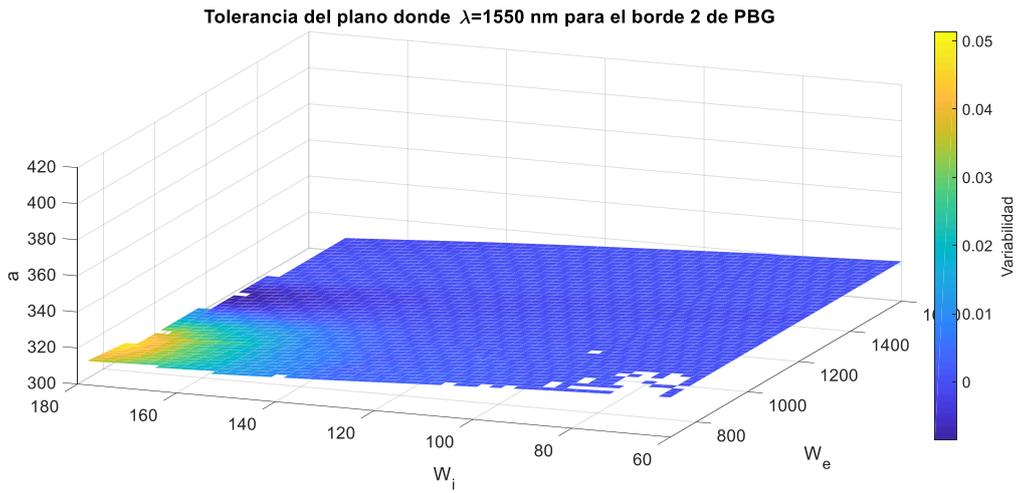
Función		
<i>creator_mapacolor_4D_tolerancia.m</i>		
		Definición
Variable de entrada	<i>num_interv</i>	Valor que indica el tamaño de equiespaciado de las interpolaciones de ' <i>a_vector</i> ', ' <i>wi_vector</i> ' y ' <i>we_vector</i> '
	Las variables de salida de la Tabla 11 (excepto las referidas a representaciones gráficas)	
Variable de salida	<i>mapacolor4D_plano1_tol</i> <i>mapacolor4D_plano2_tol</i> <i>mapacolor4D_plano3_tol</i> <i>mapacolor4D_plano4_tol</i>	Representación de la tolerancia de los valores del plano donde $\lambda=1550$ nm para los bordes 1, 2, 3 y 4 de PBG respectivamente

**Tabla 12:** Resumen de los inputs y outputs de la función '*creator\_mapacolor\_4D\_tolerancia.m*'

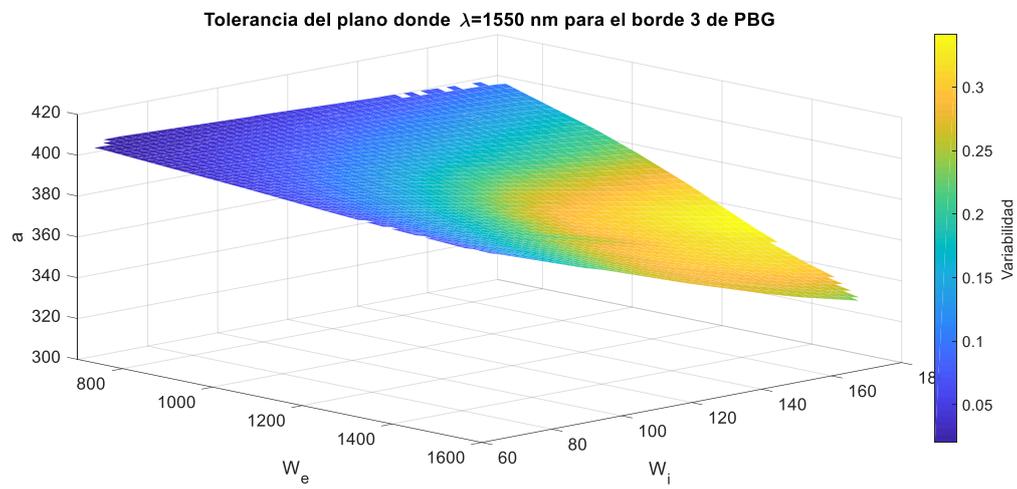
Las representaciones que genera esta función se muestran a continuación:



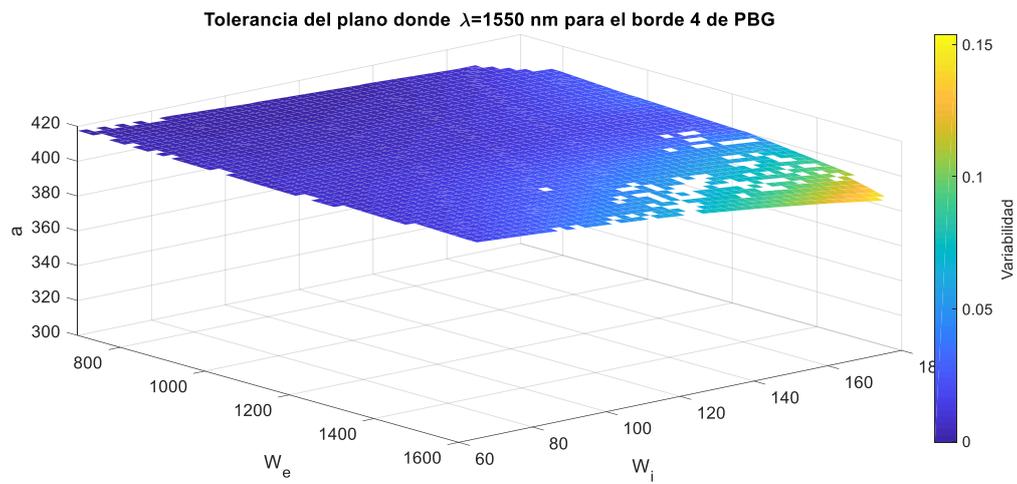
**Figura 51:** Representación de la tolerancia de los valores del plano donde  $\lambda=1550$  nm para el borde 1 de PBG



**Figura 52:** Representación de la tolerancia de los valores del plano donde  $\lambda=1550$  nm para el borde 2 de PBG



**Figura 53:** Representación de la tolerancia de los valores del plano donde  $\lambda=1550$  nm para el borde 3 de PBG



**Figura 54:** Representación de la tolerancia de los valores del plano donde  $\lambda=1550$  nm para el borde 4 de PBG

En las gráficas anteriores ( Figura 51, Figura 52, Figura 53 y Figura 54) se observa como los planos también presentan cuadrados blancos pero en este caso en mayor cantidad. Se presupone que estos valores equivalen a zonas donde  $\lambda$  no es 1550 nm exactamente. También se puede deber a la presencia de una gran cantidad de valores 'NaN' en las matrices que se representan y que generan fallos en estas figuras.

### 3.7 Funciones y representaciones de apoyo

#### 3.7.1 Plano 3D de PBG en $\lambda=1550$ nm, mapas de color y líneas de contorno

Con la finalidad de ampliar la explicación de cómo se obtienen los diferentes planos donde los bordes de PBG se encuentran en  $\lambda=1550$  nm (obtenidos en el apartado 3.4.1) se crea la función '*interseccion\_mapacolor\_plano1550.m*'.

En ésta, se representan varios conceptos: los colormap 3D indican la variación de  $\lambda$  (borde de PBG) variando dos parámetros ('*w<sub>e</sub>*' y '*w<sub>i</sub>*' en este caso) y manteniendo el otro constante ('*a*' en este caso), el plano 3D donde  $\lambda=1550$  nm para el borde de PBG y las líneas rojas de intersección entre los colormap y el plano 3D.

Los pasos que sigue el algoritmo de esta función son los siguientes:

- Para la representación del plano 3D:
  1. Los pasos realizados para la obtención de los planos de los diferentes bordes de PBG en  $\lambda=1550$  nm coincide con los pasos realizados en la función '*creator\_mapacolor\_3D\_plano1550.m*' explicados en el apartado 3.4.1
- Para el diseño de los colormap 3D:
  2. A partir de las coordenadas obtenidas en el paso 1 se representan tres colormaps equiespaciados mediante diferentes 'slices' localizados para diferentes valores de '*a*'
- Para la representación de las líneas de intersección entre los colormap y el plano:
  3. A partir de las coordenadas obtenidas en el paso 1 se representan tres líneas que indican la posición de los colormap donde el valor de  $\lambda=1550$  nm

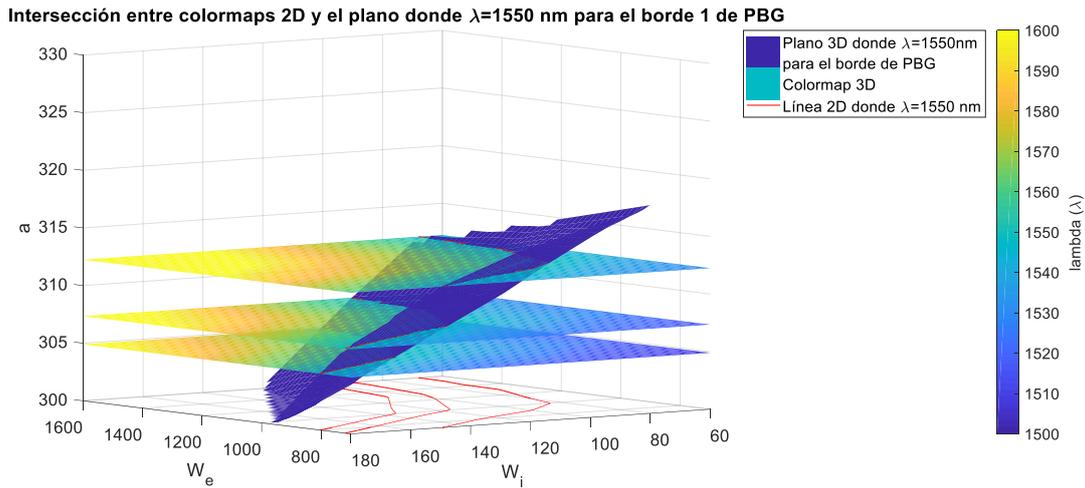
Las variables de entrada y salida de la función se muestran a continuación (Tabla 13):

Función	
<i>interseccion_mapacolor_plano1550.m</i>	
Definición	
<b>Variable de entrada</b>	Las variables de salida de la Tabla 5

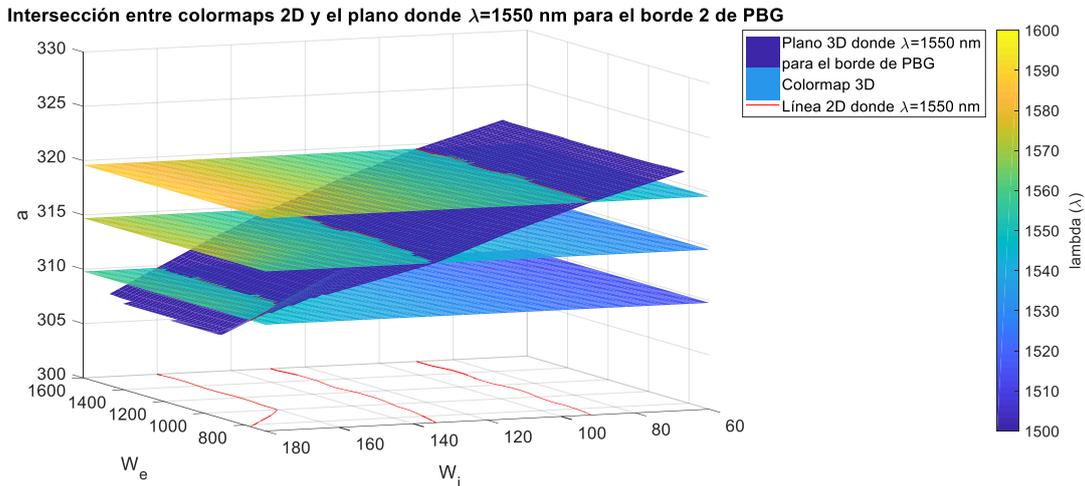
<b>Variable de salida</b>	<i>plano3D_slice_borde1</i> <i>plano3D_slice_borde2</i> <i>plano3D_slice_borde3</i> <i>plano3D_slice_borde4</i>	Representación de la intersección entre diferentes colormaps 3D con el plano donde $\lambda=1550$ nm para los bordes 1, 2, 3 y 4 de PBG respectivamente
---------------------------	--	---

**Tabla 13:** Resumen de los inputs y outputs de la función 'interseccion\_mapacolor\_plano1550.m'

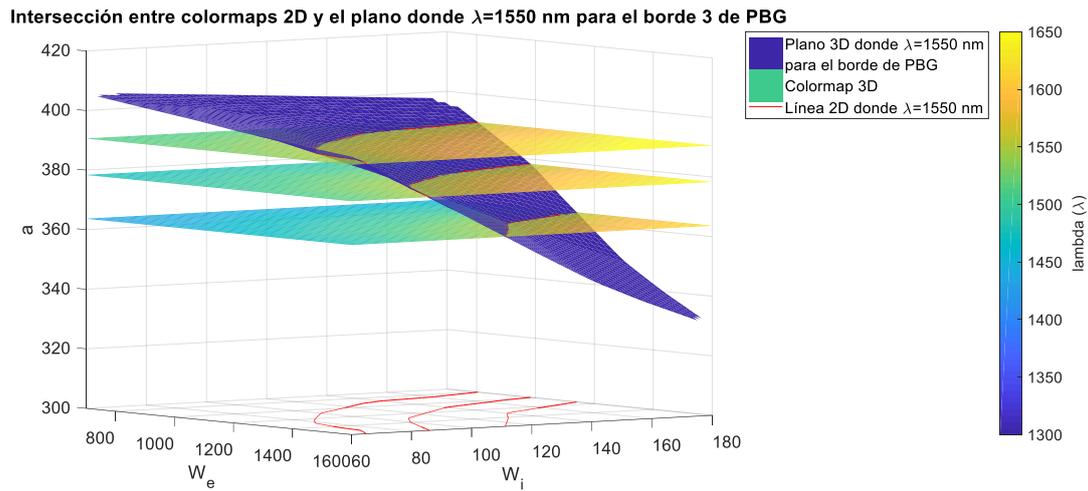
Las diferentes representaciones que se obtienen son las siguientes:



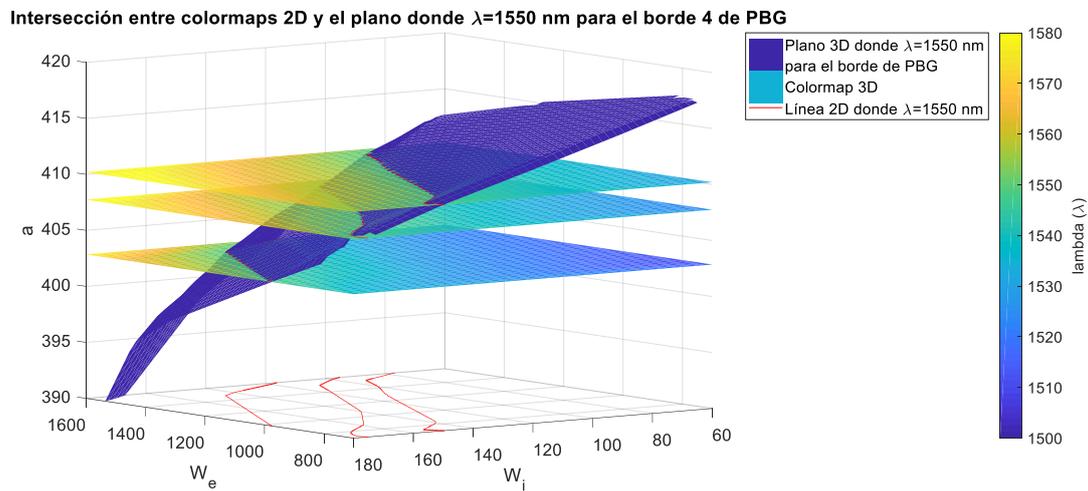
**Figura 55:** Representación de la intersección entre diferentes colormaps 3D con el plano donde  $\lambda=1550$  nm para el borde 1 de PBG



**Figura 56:** Representación de la intersección entre diferentes colormaps 3D con el plano donde  $\lambda=1550$  nm para el borde 2 de PBG



**Figura 57:** Representación de la intersección entre diferentes colormaps 3D con el plano donde  $\lambda=1550$  nm para el borde 3 de PBG



**Figura 58:** Representación de la intersección entre diferentes colormaps 3D con el plano donde  $\lambda=1550$  nm para el borde 4 de PBG

En las gráficas anteriores ( Figura 55, Figura 56, Figura 57 y Figura 58) se puede observar como los colormaps intersectan con el plano. Esto se marca con una línea roja en cada intersección. Dicha intersección, si se observa la leyenda de colores, se da cuando, para todos los valores de longitud de onda que representa cada colormap, su valor es  $\lambda = 1550$  nm.

Teniendo esto en cuenta, lo que se realiza en el apartado 3.4.1 realmente es extraer, de todos colormap generados a partir de la variación del valor de ' $a$ ', la línea que representa las diferentes configuraciones donde la longitud de onda es 1550 nm (línea roja en las figuras anteriores). A partir de estas líneas se crea, mediante interpolación, el plano donde  $\lambda=1550$  nm para cada borde de PBG.

Por lo tanto, volviendo a las representaciones anteriores se puede comprobar la correcta generación de dicho plano puesto que las líneas rojas coinciden con la intersección que se da con los colormap y, además, esta intersección se da en valores de  $\lambda=1550$  nm.

## 4. ANÁLISIS DE RESULTADOS

### 4.1 Guía corrugada

Los resultados gráficos obtenidos con la guía corrugada para cada función se han ido presentando a lo largo de su explicación correspondiente.

Para cada borde de PBG existen ciertos rangos de configuraciones donde la relación sensibilidad-tolerancia es óptima. Estos rangos se explican a continuación y, además, a modo de resumen en la Tabla 14 se encuentra toda la información de interés al respecto. Cabe indicar que estos son datos aproximados y extraídos a través de la visualización de las gráficas obtenidas en el apartado 3.5.2 y 3.6.2 en Matlab. Para obtener datos más precisos sería necesario hacer un estudio en más profundidad de todos los valores extraídos. Por otro lado, los rangos de configuraciones a partir de ahora se expresarán como (amin-amax, wimin-wimax, wemin,wemax) por mayor comodidad y la combinación de configuraciones elegida dentro de este rango se expresa como (a, wi, we).

Respecto al borde 1, puesto que los datos de la gráfica correspondiente al plano de sensibilidades para los bordes 1 de PBG que se encuentran en  $\lambda=1550$  nm no son concluyentes debido a un fallo desconocido, los datos aproximados de las sensibilidades se obtienen a partir de la Figura 39 (volumen de todas las sensibilidades para todas las configuraciones posibles del borde 1 de PBG). Analizando las tolerancias para el plano de dicho borde se observa que la mejor tolerancia, con valores de variabilidad de 0-0.02, se da para configuraciones de rangos (300-310, 75-120, 1300-1600). A partir de estos valores se busca en el volumen de sensibilidades dichos rangos y se obtiene que esta rondará los 60-100 nm/RIU. No existen anomalías en este rango. La configuración más adecuada dentro de este rango se elige en (309, 82, 1582).

Respecto al borde 2, se observa como para los mayores rangos de sensibilidad los rangos de tolerancia son mínimos (presentan los mayores niveles de variabilidad posibles). Se detectan dos posibles zonas de interés. La primera se encuentra entre los rangos de configuraciones (308-311, 170-178, 718-900) con sensibilidades de 80-85 nm/RIU y variabilidad de 0.03-0.05. La segunda zona, a pesar presentar una sensibilidad menor (70-75 nm/RIU), presenta una tolerancia máxima (variabilidad de 0-0.1). Esta zona será la elegida, encontrándose en rangos de (320-321, 67-84, 735-1590). Para ambos rangos no se presentan posibles distorsiones cerca. Dentro de la zona elegida se elige como combinación de configuraciones a (321,77,1491).

Respecto al borde 3, se analiza que, en esta ocasión, las configuraciones con mayor sensibilidad a su vez son las que mejor tolerancia presentan. Éstas se encuentran entre los rangos de parámetros (380-400, 140-180, 700-836) con sensibilidades de 220-240 nm/RIU y variabilidad de 0-0.1. Sin embargo, es necesario tener presente que para esta zona existen puntos donde el borde se encuentra bajo el cono de luz por lo que no es aconsejable su elección. Otro rango de sensibilidad óptimo se encuentra en los rangos (382-384, 93-140, 1064-1364) con 180-200 nm/RIU de sensibilidad y 0.1-0.25 de variabilidad. Para este rango no existen bordes que se encuentren bajo el cono de luz. Se elige como combinación de configuraciones a (383, 140, 1045).

Respecto al borde 4, se observa que, al igual que pasa en el borde 3, la mayor sensibilidad se da en aquellas configuraciones con mayor tolerancia. Los rangos donde se encuentra son (415-417, 60-97, 720-1310) con valores de sensibilidad de 300-340 nm/RIU y

variabilidad 0-0.05. Sin embargo, para estos rangos existe una gran cantidad de valores que se pueden encontrar bajo el cono de luz o donde no exista PBG pudiendo generar anomalías en un futuro. Por este motivo se elige otro rango donde, a pesar de presentar valores de sensibilidad menores, no existen posibles anomalías. Este rango se encuentra en (403-407, 118-146, 1355-1482) donde se presentan valores de sensibilidad de 200-240 nm/RIU y variabilidad de 0.05-0.08. La configuración elegida dentro de este rango es (406, 126, 1400).

Gracias a la siguiente tabla se conoce que la mejor relación tolerancia-sensibilidad se da para el borde 3 y 4. Además, se observa como existen rangos de valores donde las prestaciones son mejores que las elegidas, sin embargo, al presentar anomalías se descartan.

Resultados guía corrugada (corr3D)						
	Rango de configuraciones más adecuadas (amin-amax, wimin-wimax, wemin-wemax)			Sensibilidad (nm/RIU)	Variabilidad (inversa de Tolerancia)	Posible anomalía
	Configuración elegida (a, wi, we)					
<b>Borde 1</b>	(300-310, 75-120, 1300-1600)			60-100	0-0.02	No
	(309, 82, 1582)			80	0	No
<b>Borde 2</b>	(308-311, 170-178, 718-900)			80-85	0.03-0.05	No
	(320-321, 67-84, 735-1590)			70-75	0-0.1	No
	(321,77,1491)			75	0	No
<b>Borde 3</b>	(380-400, 140-180, 700-836)			220-240	0-0.1	Sí
	(382-384, 93-140, 1064-1364)			180-200	0.1-0.25	No
	(383, 140, 1045)			200	0.25	No
<b>Borde 4</b>	(415-417, 60-97, 720-1310)			300-340	0-0.05	Sí
	(403-407, 118-146, 1355-1482)			200-240	0.05-0.08	No
	(406, 126, 1400)			220	0.05	No

**Tabla 14:** Resumen de los rangos de configuraciones posibles donde existe una buena relación sensibilidad-tolerancia (marcado en verde aquellos rangos elegidos como más adecuados) así como de la configuración concreta elegida para aquellos rangos (marcada en rojo). Se añaden los valores de sensibilidad y tolerancia para cada rango y la existencia o no de posibles anomalías.

## 4.2 Guía corrugada inversa

Para comprobar el correcto funcionamiento del sistema se simula en él la guía corrugada inversa (explicada en el apartado 2.5.2). Una de las diferencias respecto a sus propiedades en comparación con la guía corrugada normal es que solo presenta un PBG en vez de dos. Teniendo esto en cuenta, es necesario realizar ciertos cambios en el código para que detecte de manera correcta las características de esta estructura. Los cambios se han incluido en el código del sistema localizado en el apartado de Anexos. Los resultados obtenidos con este tipo de estructura se presentan a continuación:

- Respecto a la obtención de los planos donde el PBG está en  $\lambda=1550$  nm:

Se representan en dichas gráficas ( Figura 59 y Figura 60) a la vez los puntos que presentan distorsión. Para el borde 1 no se observa la existencia de puntos que puedan generar anomalías mientras que para el borde 2 existen puntos que se encuentran bajo el cono de luz y puntos que presentan distorsión por solape con la banda 2.

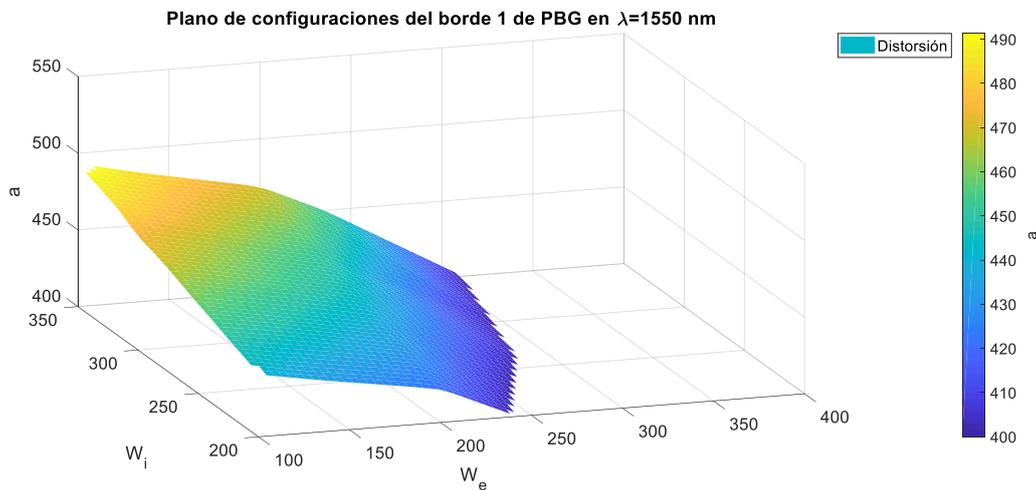


Figura 59: Representación obtenida a través de 'creator\_mapacolor\_3D\_plano1550.m'

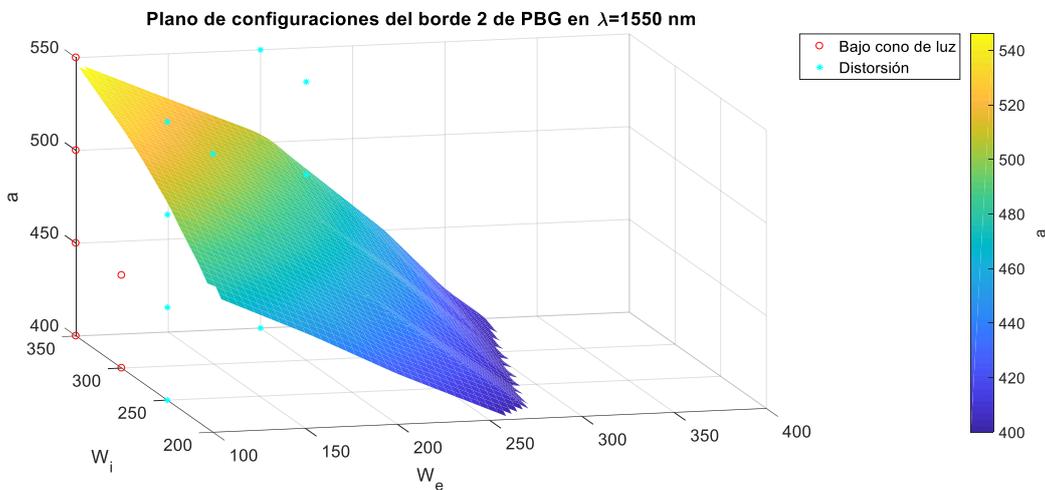
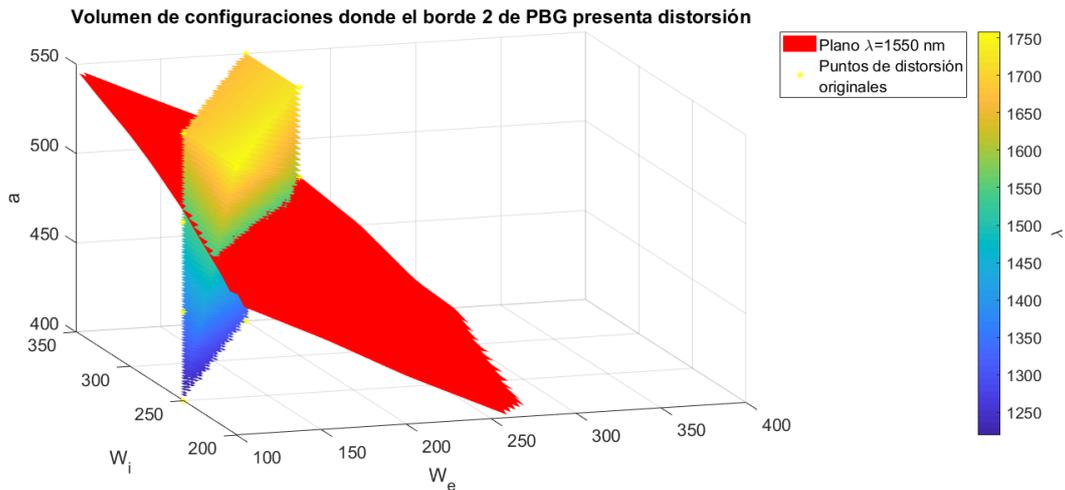
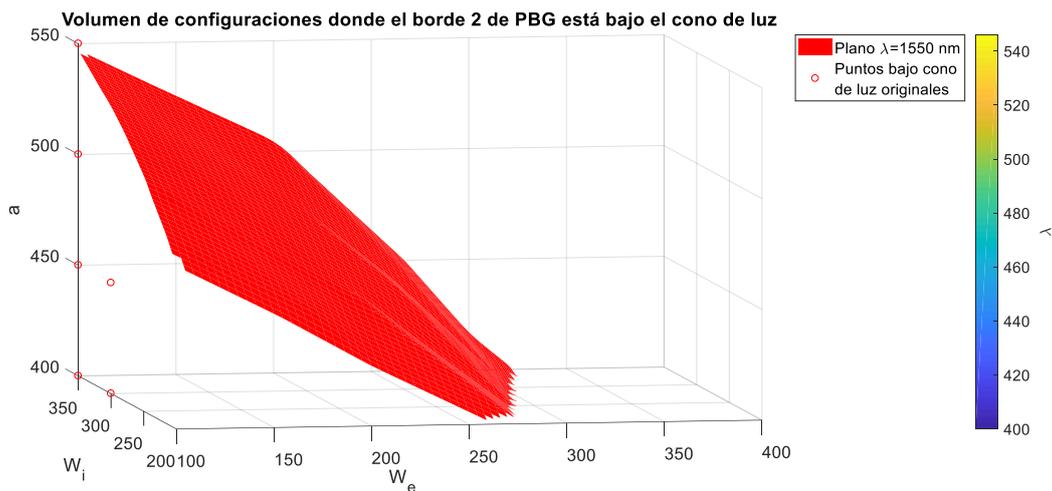


Figura 60: Representación obtenida a través de 'creator\_mapacolor\_3D\_plano1550.m'

- Respecto al cálculo de los volúmenes de configuraciones que presentan distorsión:



**Figura 61:** Representación obtenida a través de 'creator\_mapacolor\_3D\_vol\_dist.m'



**Figura 62:** Representación obtenida a través de 'creator\_mapacolor\_3D\_vol\_dist.m'

Puesto que el borde 1 no presenta puntos de distorsión su plano no se representa. En la Figura 61 no se ha creado un volumen puesto que los puntos por debajo de la línea de luz únicamente se localizan en un plano. En la Figura 62 se observa el volumen de distorsión para todas las configuraciones posibles del borde 2 de PBG.

- Respecto a la obtención de la sensibilidad:

Los valores de sensibilidad del borde 1 ( Figura 63) y 2 ( Figura 64) presentan rangos parecidos, sin embargo, en el borde 1 existe una mayor cantidad de configuraciones que presentan el valor de sensibilidad máxima (600-650 nm/RIU).

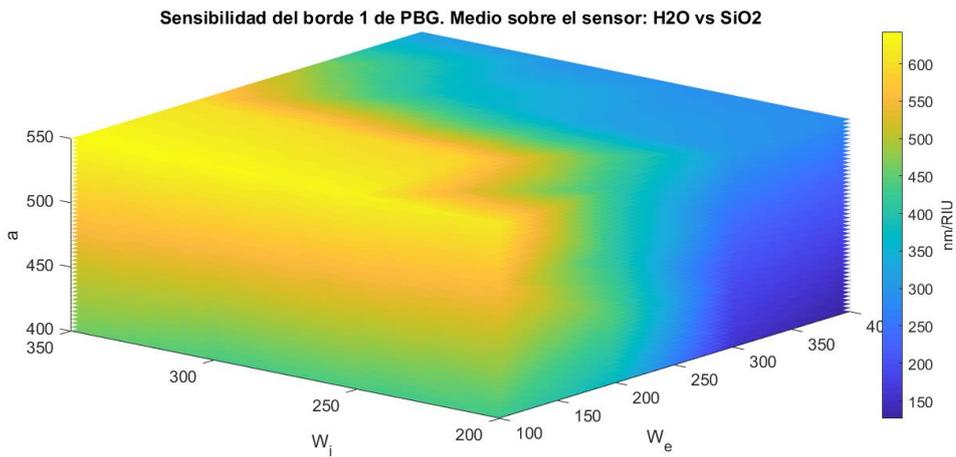


Figura 63: Representación obtenida a través de 'creator\_vol\_sensibilidad.m'

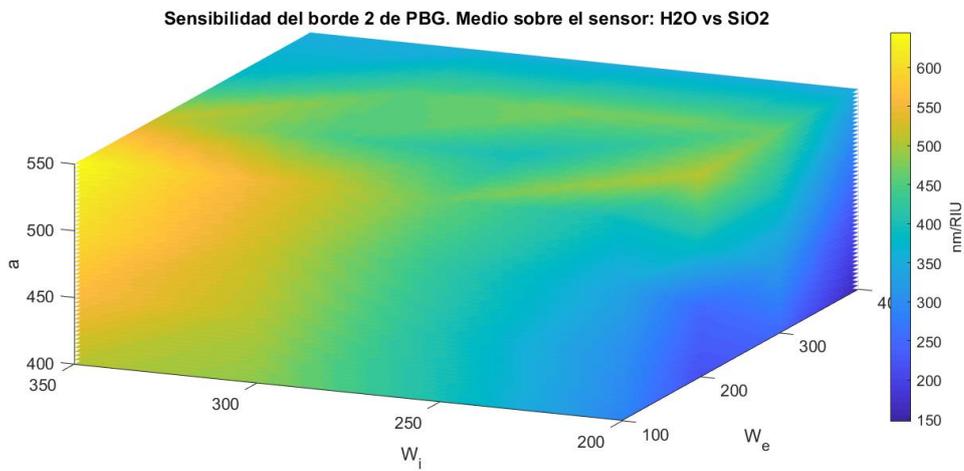


Figura 64: Representación obtenida a través de 'creator\_vol\_sensibilidad.m'

- Respecto a la sensibilidad en los bordes de PBG en  $\lambda=1550$  nm:

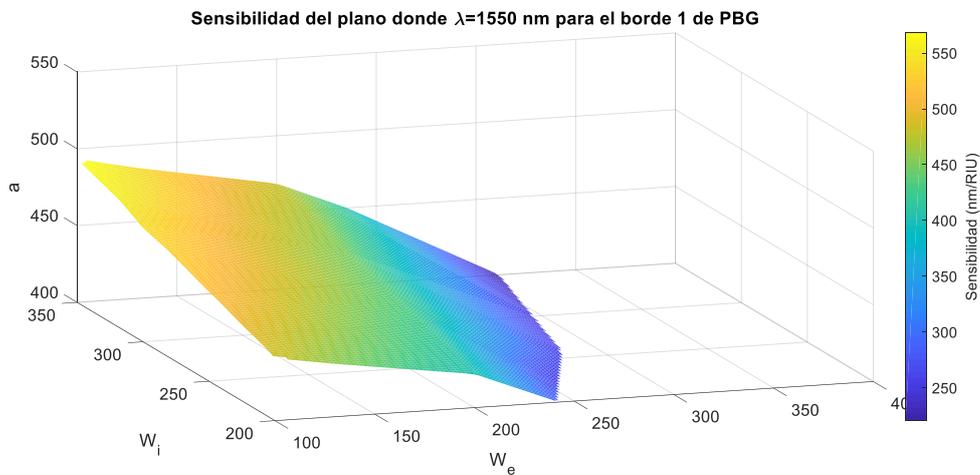
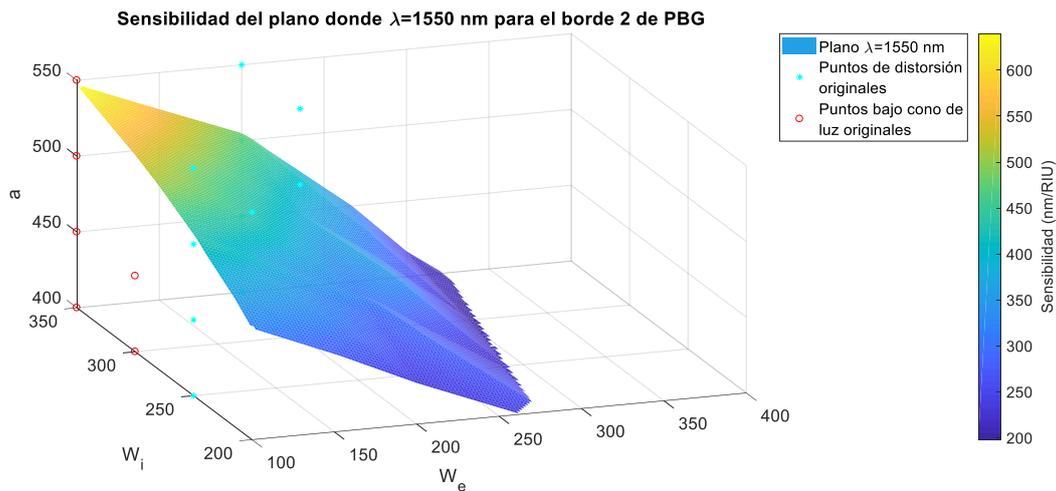


Figura 65: Representación obtenida a través de 'creator\_mapacolor\_4D\_sensibilidad.m'

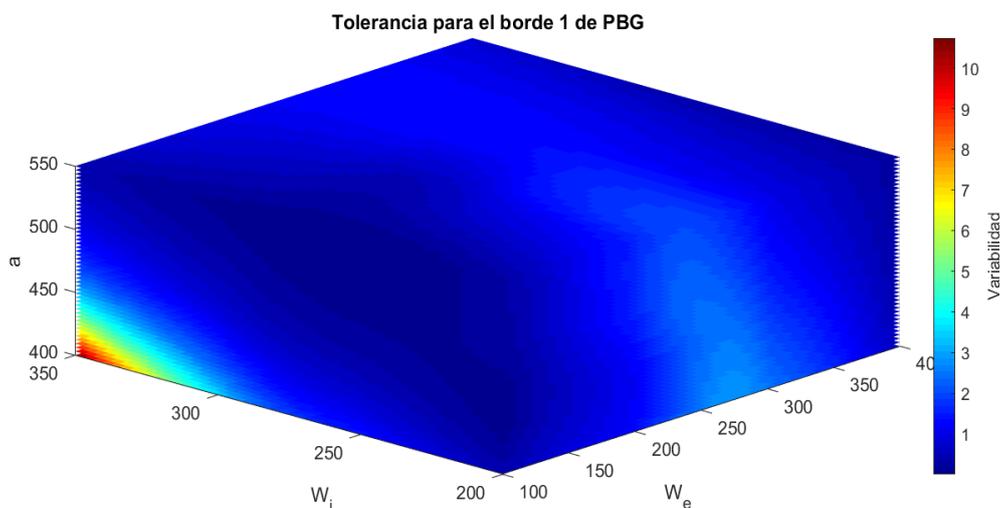


**Figura 66:** Representación obtenida a través de *'creator\_mapacolor\_4D\_sensibilidad.m'*

La Figura 65 y Figura 66 representan los valores de sensibilidad para los bordes 1 y 2 respectivamente. Además, se representan las configuraciones donde los bordes pueden presentar distorsión (en el borde 1 no existen dichos puntos).

- Respecto a la obtención de la tolerancia:

En las representaciones de las tolerancias ( Figura 67 y Figura 68) se decide cambiar la escala de color para una mejor visualización. En ambas ilustraciones se observa como, a pesar de presentar un rango de variabilidad muy elevado (0-16) la mayoría de las configuraciones se encuentran en los valores más inferiores (variabilidad entre 0-5). Sin embargo, es necesario tener en cuenta que estos valores son mucho mayores a los obtenidos en la guía corrugada normal (donde los rangos oscilan de 0-0.30) por lo que se deduce que la guía corrugada inversa será mucho más sensible a variaciones en la fabricación que la otra guía estudiada.



**Figura 67:** Representación obtenida a través de *'creator\_vol\_tolerancia.m'*

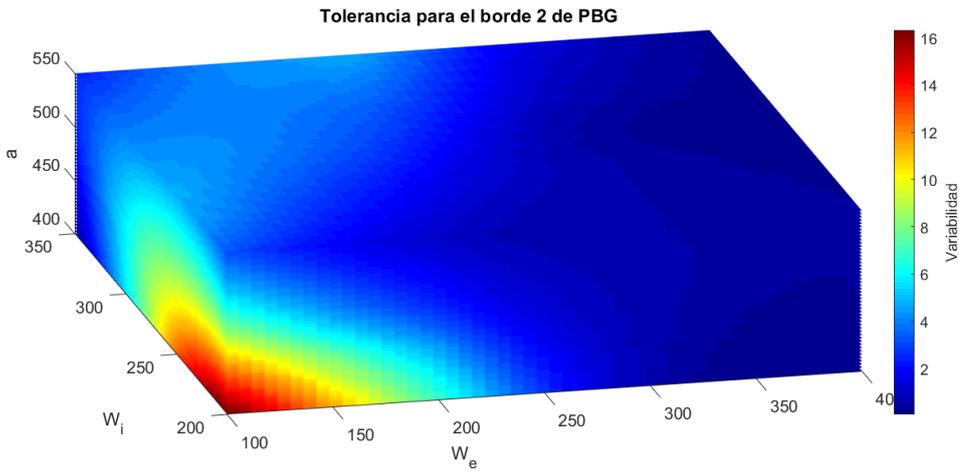


Figura 68: Representación obtenida a través de 'creator\_vol\_tolerancia.m'

- Respecto a la tolerancia en los bordes de PBG en  $\lambda=1550$  nm:

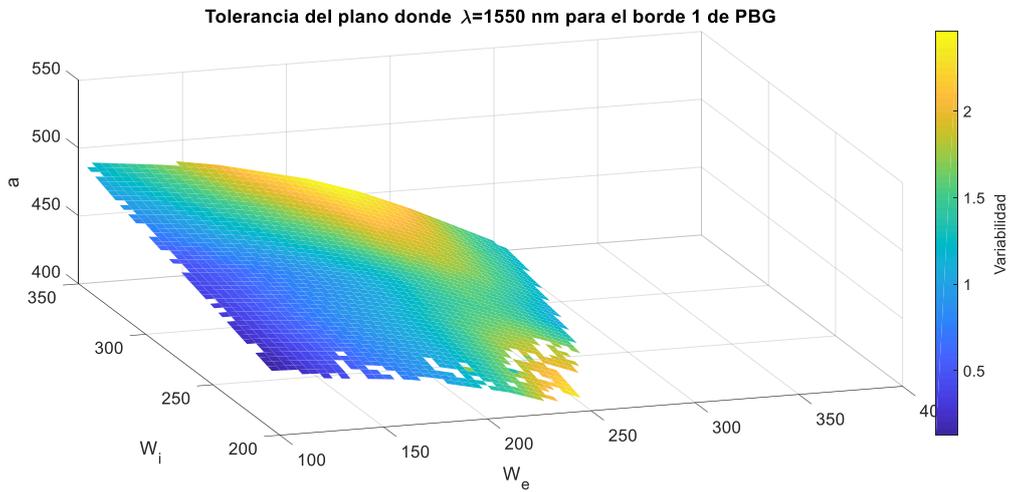


Figura 69: Representación obtenida a través de 'creator\_mapacolor\_4D\_tolerancia.m'

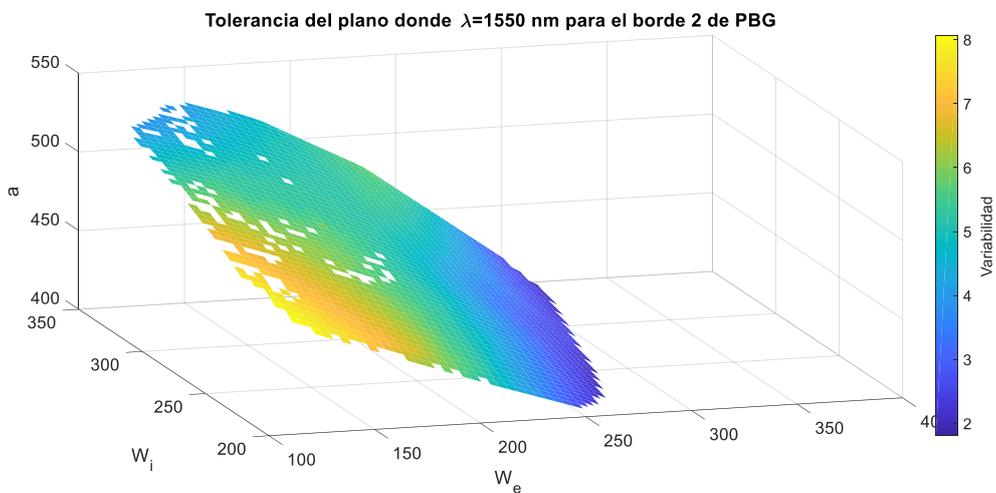


Figura 70: Representación obtenida a través de 'creator\_mapacolor\_4D\_tolerancia.m'

En la Figura 69 y Figura 70 se ilustran las tolerancias de ambos bordes. Se observa la presencia de los cuadrados blancos como errores de representación.

Finalmente, los rangos donde la relación sensibilidad-tolerancia es más adecuada para los bordes 1 y 2 se muestran en la Tabla 15 a continuación. La mejor relación tolerancia-sensibilidad se da para el borde 1.

Resultados guía corrugada (corr3Dinv)						
	Rango de configuraciones más adecuadas (amin-amax, wimin-wimax, wemin-wemax)			Sensibilidad (nm/RIU)	Variabilidad (inversa de Tolerancia)	Posible anomalía
	Configuración elegida (a, wi, we)					
<b>Borde 1</b>	(480-490,	330-350,	100-140)	525-550	1-1.5	No
	(423-455,	200-243,	100-150)	425-500	0-1	No
	(441, 202,100)			500	0	No
<b>Borde 2</b>	(505-527,	265-350,	100-200)	450-550	4-6	No
	(525,346,147)			550	5	No

**Tabla 15:** Rangos de configuraciones posibles donde existe una buena relación sensibilidad-tolerancia (marcado en verde aquellos rangos elegidos como más adecuados) así como de la configuración concreta elegida para aquellos rangos (marcada en rojo). Se añaden los valores de sensibilidad y tolerancia para cada rango y la existencia o no de posibles anomalías

## 5. CONCLUSIONES

Gracias a los resultados obtenidos y en función de los objetivos planteados se puede concluir que el objetivo general se ha conseguido. Es decir, se ha logrado desarrollar un sistema que permita obtener aquellas configuraciones que presentan la mayor relación sensibilidad-tolerancia de los diferentes bordes de PBG teniendo en cuenta la posible presencia de anomalías. Sin embargo, es preciso tener en cuenta que se necesitan ciertas nociones básicas no estudiadas en la carrera para poder desarrollarlo. A pesar de esto, se considera que el sistema diseñado obtiene resultados óptimos pese a ser mejorables.

Por otro lado, haciendo referencia a los objetivos específicos planteados en el trabajo se concluye que también se han cumplido satisfactoriamente todos:

- Se ha adquirido nuevos conocimientos teóricos acerca de la nanotecnología y nanofotónica que han permitido el desarrollo del trabajo. La observación y comprensión de los patrones que siguen los PBG ha sido muy importante a la hora de desarrollar un sistema lo más robusto posible

- La extracción de parámetros ha sido exitosa pudiendo detectar la presencia de cada PBG y extraer su localización. Además, se ha podido indicar aquellos bordes de PBG que presentan alguna anomalía (por solape con una banda, por localizarse el PBG por debajo de la línea de luz o por inexistencia de borde de PBG). Puesto que las anomalías generan distorsión en la transmisión de la luz en la estructura nanofotónica y, en consecuencia, provocan una pérdida de prestaciones de ésta es importante conocer dónde se producen para evitar el diseño de estructuras con estas configuraciones en la medida de lo posible.
- La obtención y representación de los planos de cada borde de PBG cuando se encuentra en  $\lambda=1550$  nm también se ha realizado de manera correcta. La generación de este plano facilita la determinación de aquellas configuraciones donde el borde de PBG se encuentra en la longitud de onda deseada. Además, se añade la representación de las anomalías presentes en cada borde de PBG
- El cálculo y la representación de la sensibilidad y tolerancia de todas las configuraciones simuladas se ha llevado a cabo correctamente y como medio para finalmente obtener por separado los valores de estos parámetros cuando el borde de PBG se encuentra en  $\lambda=1550$  nm (excepto para el borde 1 de *corr3D*). Cabe mencionar que estas representaciones pueden ser mejoradas con un estudio más profundo del tema
- A partir de las gráficas de tolerancias y sensibilidades para los bordes de PBG donde  $\lambda=1550$  nm (excepto para el borde 1 de *corr3D*) se consigue extraer los rangos para todas las variables geométricas estudiadas (periodicidad, anchura de los elementos transversales de la guía y longitud de los elementos transversales de la guía) donde la relación sensibilidad-tolerancia es más adecuada.
- Todo el proceso se ha automatizado lo máximo posible de manera que únicamente sea necesario la introducción de ciertos parámetros de entrada y la ejecución de las simulaciones en el servidor MPB de manera manual. Una vez esto se haya introducido el sistema obtendrá todas las representaciones explicadas en el apartado 3 que se encuentren dentro de la función general de manera autónoma.

## 6. POSIBLES LÍNEAS FUTURAS

Existen diferentes caminos a seguir una vez finalizado el desarrollo del sistema. En función de la finalidad que se busque se pueden diferenciar del siguiente modo:

- Respecto a la obtención de mejoras técnicas del sistema:
  1. Automatizar las simulaciones MPB: En el sistema actual es necesario ejecutar el script de simulación manualmente. Es decir, el sistema crea un script que contiene las simulaciones múltiples que el servidor ejecutará; sin embargo, la acción de ejecutar dicho script se debe realizar de manera manual. Para poder crear un sistema aún más autónomo se podría estudiar la manera de, a través de Matlab, poder ejecutar dicho script en el servidor automáticamente sin necesidad de programas externos (SSH Secure File Transfer Client y SSH Secure Shell Client)

2. Resolver los problemas de representación de sensibilidad y tolerancia para los planos donde el borde de PBG se encuentra en  $\lambda=1550$  nm: Como se ha visto en los apartados 3.5.2 y 3.6.2 al representar dichos planos aparecen zonas blancas. Se ha supuesto que esto se debe a fallos en la representación debido a la gran cantidad de valores 'NaN' en la matriz que se representa o por la inexistencia de borde en  $\lambda=1550$  nm en esa localización. Al suponer que se debe a un fallo se estudiaron, por ejemplo, diversas maneras de extrapolar estos datos por sus cercanos sin éxito. Por este motivo, dedicar más tiempo a la búsqueda de evitar estos errores de representación podría ser otro objetivo futuro de mejora
  3. Ampliar los métodos de detección de PBG a más tipos de estructuras: entre otras cosas, en función de la forma geométrica de las estructuras, la cantidad y la localización de los bordes de PBG que poseen varía. En este caso el desarrollo del sistema se realizó para unas estructuras concretas (*corr3D* y *corr3Dinv*) de manera que si se introduce otra geometría el sistema no funcionará correctamente. Una manera de conseguir un sistema más robusto sería añadir la detección de una mayor variedad de estructuras
  4. Ampliar la cantidad de variables estudiadas en el sistema: actualmente el proceso se realiza para una comparación de cuatro variables ( $\alpha$ ,  $w_i$ ,  $w_e$  y material sobre la guía); sin embargo, esta cantidad se podría aumentar para conseguir la mayor cantidad de información posible sobre el comportamiento de las estructuras cuando estos varían
  5. Obtención de la sensibilidad si se deposita una pequeña capa (5-10 nm de espesor) correspondiente a un analito a detectar en lugar de modificar todo el medio que se encuentra alrededor de la estructura (como se ha realizado en este TFG)
- Respecto a los pasos posteriores al diseño:
    6. Fabricación de las estructuras óptimas: a través de los resultados obtenidos del sistema, una línea futura posible es elegir aquellas que se consideren más adecuadas y fabricar dichas estructuras. Con esto se podría comprobar la validez del sistema gracias a la comparación de los resultados de diseño y los físicos

## 7. REFERENCIAS

- Arafa, S., Bouchemat, M., Bouchemat, T., Benmerkhi, A., & Hocini, A. (2017). Infiltrated photonic crystal cavity as a highly sensitive platform for glucose concentration detection. *Optics Communications*, 384, 93-100. <http://doi.org/10.1016/j.optcom.2016.10.019>
- Bellingeri, M., Chiasera, A., Kriegel, I., & Scotognella, F. (2017). Optical properties of periodic, quasi-periodic, and disordered one-dimensional photonic structures. *Optical Materials*, 72, 403-

421. <http://doi.org/10.1016/j.optmat.2017.06.033>

Cascón López, J., & Universidad de Castilla-La Mancha. (2000). *Filtros ópticos canónicos reconfigurables para señales de radiofrecuencia análisis y síntesis*. Edic. de la Universidad de Castilla - La Mancha. Recuperado a partir de <https://books.google.es/books?id=u4ffY6OA6LMC&pg=PA21&lpg=PA21&dq=partes+de+una+guiaonda&source=bl&ots=vWuojjZKNf&sig=NHSJzeINehJ1BaMvR2DH5omBEwc&hl=es&sa=X&ved=0ahUKEwjeh-jdz-nVAhXH1xQKHZXC7MQ6AEIRjAJ#v=onepage&q=partes+de+una+guiaonda&f=false>

Centro de Tecnología Nanofotónica de Valencia. (s. f.). Micro/Nanofabricación. Recuperado 14 de agosto de 2017, a partir de <https://www.ntc.upv.es/nanofabricacion.html>

García Rupérez, J. (2008). *Estudio de los fenómenos de onda lenta y dispersión en estructuras periódicas de fotónica integrada*.

García Rupérez, J. (2016). Desarrollo de biosensores nanofotónicos para sistemas de análisis lab-on-a-chip.

Joannopoulos, J. J. D., Johnson, S., Winn, J. N. J., & Meade, R. R. D. (2008). *Photonic crystals: molding the flow of light*. *Time*. <http://doi.org/10.1063/1.1586781>

Kilders, P. S., & Brimont, A. (2009). *Análisis y caracterización de guías corrugadas de onda lenta para aplicaciones de modulación electro-óptica en silicio*.

Llorente Saez, R. (2017). *Apuntes de la asignatura de Micro-nano tecnología. 4º GIB*.

Luis Motellón, J., & Bueren, J. (2010). *9ª Edición del curso de biotecnología aplicada a la salud humana*. (S. L. Edikamed, Ed.). Barcelona. Recuperado a partir de [https://s3-eu-west-1.amazonaws.com/farmavet/amgen.es/web/pdf/Biotecnologia\\_2010.pdf](https://s3-eu-west-1.amazonaws.com/farmavet/amgen.es/web/pdf/Biotecnologia_2010.pdf)

Martí, J., & Martínez, A. (2016). *Apuntes de la asignatura Biotecnología y Nanotecnología 3º GIB*.

Sukhoivanov, I. A., & Guryev, I. V. (2009). *Photonic crystals : physics and practical modeling*. Springer.

Xu, Y., Zhang, X., Luan, C., Wang, H., Chen, B., & Zhao, Y. (2017). Hybrid hydrogel photonic barcodes for multiplex detection of tumor markers. *Biosensors and Bioelectronics*, 87, 264-270. <http://doi.org/10.1016/j.bios.2016.08.063>

# **PRESUPUESTO**

# 1. INTRODUCCIÓN

El presupuesto del proyecto supone una parte a tener en cuenta a la hora de, en un futuro, poder obtener su rentabilidad de uso. Para poder obtenerlo es necesario conocer el coste que presenta cada recurso utilizado en él. Así, en este apartado se realiza un desglose del presupuesto inicial del proyecto donde se indica el coste de cada recurso empleado así como el salario del personal que desarrolla el sistema. Es necesario tener en cuenta que, excepto el ordenador, todos los materiales empleados son software, al tratarse de un proyecto que pretende ser utilizado previamente a los procesos de fabricación.

## 2. PRESUPUESTO

Se realiza un desglose del presupuesto en función de su referencia a recursos para el desarrollo del sistema (Tabla 16) o a salarios del personal (Tabla 17). Finalmente se expone el presupuesto total obtenido a partir de los presupuestos parciales (Tabla 18). La información acerca de la estructura general y composición del presupuesto se ha extraído de los apuntes de la asignatura Proyectos y Fabricación de 4º de Grado en Ingeniería Biomédica (Pastor Ferrando, 2016).

### 2.1 Presupuesto parcial

- Presupuesto parcial 1: Recursos

RECURSOS					
	CONCEPTO	DENOMINACIÓN	CANTIDAD (unidad)	COSTE UNITARIO (€/unidad)	COSTE TOTAL (€)
MATERIALES	Ordenador	Herramienta física que, en combinación con los recursos software, permite el desarrollo del sistema	1	850	850
	Ordenador de altas prestaciones	Herramienta física que contiene el servidor en el que está instalado el simulador MPB	1	3199	3199
SOFTWARE	Licencia Matlab 2017	Herramienta de lenguaje matemático	1	2144	2144
	MIT Photonic Bands (MPB)	Programa de simulación de estructuras fotónicas. Posee licencia GNU	1	0	0

		General Public Licence. Software libre			
	Licencia MayaVi	Visualizador de datos científicos. Posee licencia BSD. Software libre	1	0	0
	Licencia SSH Secure File Transfer Client	Permite la conexión entre el servidor de MPB del NTC (Centro de Nanofotónica de valència). Software libre	1	0	0
	Licencia SSH Secure Shell Client	Permite la conexión entre el servidor de MPB del NTC (Centro de Nanofotónica de valència). Software libre	1	0	0
	Licencia Office 2016	Conjunto de herramientas software para el desarrollo escrito del proyecto	1	149	149
<b>TOTAL</b>					6342

**Tabla 16:** Presupuesto desglosado de recursos del proyecto

- Presupuesto parcial 2: Salarios

<b>SALARIOS</b>					
	<b>CONCEPTO</b>	<b>DENOMINACIÓN</b>	<b>CANTIDAD (horas)</b>	<b>COSTE UNITARIO (€/h)</b>	<b>COSTE TOTAL (€)</b>
PERSONALES	1 Ingeniero Biomédico	Graduado en Ingeniería Biomédica	300	13	3900
<b>TOTAL</b>					3900

**Tabla 17:** Presupuesto desglosado de salarios del proyecto

Se ha elegido el ordenador portátil de trabajo Asus F756UV-TY204T cuyo precio aparece en (fnac, s. f.), el ordenador de altas prestaciones es un Mountain Titanium cuyo precio aparece en («Titanium: Ordenador de Sobremesa potente y de alto rendimiento.», s. f.) la licencia de Matlab 2017 se obtiene de (MathWorks, s. f.), la licencia de MIT MPB se obtiene de (AbInitio, s. f.), la licencia de MayaVi se obtiene de («The MayaVi Data Visualizer», s. f.), la licencia de SSH

Secure File Transfer Client y Licencia SSH Secure Shell Client se obtienen de («SSH client alternatives and comparison», s. f.) y la licencia de Office 2016 se obtiene de («Comparación de todos los productos de Microsoft Office | Microsoft Office», s. f.).

Respecto al cálculo del salario bruto de un ingeniero biomédico recién graduado se ha realizado una estimación a partir del presupuesto estimado de una beca de investigación predoctoral ofrecida por la *Generalitat Valenciana* (GVA) (Conselleria d' Educació Investigació Cultura i Esport, 2017). En esta, el salario bruto máximo de un investigador es de 22192.80 €/año sin descontar las retenciones generadas por la Seguridad Social e IRPF de dicha subvención. Así se obtiene 1849.4 €/mes y se realiza una aproximación de 35 h/semana trabajadas por lo que se obtendría aproximadamente un salario bruto de 13 €/h. Este valor es el que se añade en el presupuesto parcial de 'Salarios' a modo de aproximación de los costes generados como salario de un ingeniero biomédico como investigador predoctoral.

## 2.2 Presupuesto total

PRESUPUESTO TOTAL		
		COSTE TOTAL (€)
RECURSOS		6342
SALARIOS		3900
<b>TOTAL</b>		10242

**Tabla 18:** Presupuesto total del proyecto

## 3. REFERENCIAS

Ablnitio. (s. f.). MIT Photonic Bands - Ablnitio. Recuperado 20 de agosto de 2017, a partir de [http://ab-initio.mit.edu/wiki/index.php/MIT\\_Photonic\\_Bands](http://ab-initio.mit.edu/wiki/index.php/MIT_Photonic_Bands)

Comparación de todos los productos de Microsoft Office | Microsoft Office. (s. f.). Recuperado 20 de agosto de 2017, a partir de <https://products.office.com/es-es/compare-all-microsoft-office-products?tab=1>

Conselleria d' Educació Investigació Cultura i Esport. (2017). Diari oficial de la Generalitat Valenciana. Recuperado a partir de [http://www.dogv.gva.es/datos/2017/01/30/pdf/2017\\_647.pdf](http://www.dogv.gva.es/datos/2017/01/30/pdf/2017_647.pdf)

fnac. (s. f.). Portátil Asus F756UV-TY204T Blanco - PC Portátil. Recuperado 20 de agosto de 2017, a partir de <https://www.fnac.es/Portatil-Asus-F756UV-TY204T-Blanco-Ordenador-portatil-PC-Portatil/a1339800>

MathWorks. (s. f.). New License for MATLAB Home R2017a - MathWorks España. Recuperado 20 de agosto de 2017, a partir de <https://es.mathworks.com/store/link/products/home/new>

Pastor Ferrando, J. P. (2016). *Apuntes de la asignatura Proyectos y Fabricación 4º GIB*.

SSH client alternatives and comparison. (s. f.). Recuperado 20 de agosto de 2017, a partir de <https://www.ssh.com/ssh/client>

The MayaVi Data Visualizer. (s. f.). Recuperado 20 de agosto de 2017, a partir de <http://mayavi.sourceforge.net/index.html>

Titanium: Ordenador de Sobremesa potente y de alto rendimiento. (s. f.). Recuperado 20 de agosto de 2017, a partir de <https://www.mountain.es/sobremesa/titanium>

## ANEXOS

# 1. CÓDIGO MATLAB DE LAS FUNCIONES DESARROLLADAS

## 1.1 Sistema automatizado general

### 1.1.1 *funcion\_general.m*

```
1 function funcion_general(amin,aintervl,amax,wimin,wiintervl,wimax,wemin,...
2 weintervl,wemax)
3 %
4 %
5 % %%%Función principal a partir de la cual se llama al resto de funciones
6 % %%%necesarias para realizar el proceso de optimización del sensor.
7 %
8 % %%%NOTA: En esta función se explica únicamente a grandes rasgos los pasos
9 % llevados a cabo durante
10 % %%%el proceso de optimización. La información más detallada acerca de los
11 % %%%pasos de cada función se
12 % %%%encuentra en los comentarios de dichas funciones
13 %
14 %%%Entrada:
15 % amin: valor mínimo que tomará la variable a
16 % aintervl: valor constante que indica la diferencia de tamaño entre un
17 % valor de a y su siguiente (separación equiespaciada entre amin y amax)
18 %
19 % amax: valor máximo que tomará la variable a
20 % wimin: valor mínimo que tomará la variable wi
21 % wiintervl: valor constante que indica la diferencia de tamaño entre un
22 % valor de wi y su siguiente (separación equiespaciada entre wimin y wimax)
23 %
24 % wimax: valor máximo que tomará la variable wi
25 % wemin: valor mínimo que tomará la variable we
26 % weintervl: valor constante que indica la diferencia de tamaño entre un
27 % valor de we y su siguiente (separación equiespaciada entre wemin y wemax)
28 %
29 % wemax: valor máximo que tomará la variable we
30 %
31 %%%Salida:
32 %
33 % Todos los outputs de las funciones siguientes
34 %
35 estructura=input('Introduzca el nombre de la estructura (corr3D o
36 corr3Dinv) \n',...
37 's');
38 %% Paso 1. Creación de los archivos .ctl
39 filescreator3D(amin,aintervl,amax,wimin,wiintervl,wimax,wemin,weintervl,...
40 wemax,estructura);
41 %% Paso 2. Creación del archivo de simulación múltiple
42 simulaciones_script_creator3D(amin,aintervl,amax,wimin,wiintervl,wimax,...
43 wemin,weintervl,wemax,estructura);
44
45 %% Paso 3. Utilización del simulador MPB
46 %Se obtienen los archivos .ypar, .zpar y .fre (entre otros) necesarios para
47 %poder continuar el análisis
48
49 pause
50
51 %% Paso 4. Obtención de 'matrizBG'
52
53 %Dentro de la siguiente función se llama a otras dos funciones:
54 %'dibbandasMPB_3Dslab_select_parity_wavelength' calcula las bandas de cada
55 %configuración y 'find_BG_3D' obtiene cada columna de 'matrizBG'
56
57 [matrizBG,a_vector,wi_vector,we_vector,num_intervl]= creator_matrizBG_3D...
58 (amin,aintervl,amax,wimin,wiintervl,wimax,wemin,weintervl,wemax,...
59 estructura);
```

```

60
61 %% Paso 5. Representación de los planos de posibles configuraciones donde
62 %% los bordes de PBG se corresponden con lambda=1550 nm
63
64
[mapacolor3D_plano_borde1,mapacolor3D_plano_borde2,mapacolor3D_plano_borde3,..
.
65 mapacolor3D_plano_borde4,borde1_dist,borde2_dist,borde3_dist,...
66 borde4_dist,borde3_bajo_cono,borde4_bajo_cono,borde4_sinBG,u,...
67 borde2_bajo_cono]=creator_mapacolor_3D_plano1550(matrizBG,a_vector,...
68 wi_vector,we_vector,num_interv,estructura);
69
70 %% Paso 6. Representación de los volúmenes de anomalías que se generan a
71 %% partir de los puntos originales de anomalías obtenidos en el
72 %% paso 5. Además, se superponen en las gráficas dichos puntos y
73 %% el plano obtenido en el paso 5
74
75 %A partir de estas gráficas se puede conocer qué configuraciones no son
76 %óptimas a la hora de fabricar por presentar anomalías
77 [mapacolor3D_vol_dist1,mapacolor3D_vol_dist2,mapacolor3D_vol_dist3,...
78
mapacolor3D_vol_dist4,mapacolor3D_vol_bajoCono3,mapacolor3D_vol_bajoCono4,...
79 mapacolor3D_vol_sinBG4,u,mapacolor3D_vol_bajoCono2]=...
80 creator_mapacolor_3D_vol_dist(matrizBG,a_vector,wi_vector,we_vector,...
81 num_interv,borde1_dist, borde2_dist,borde3_dist,borde4_dist,...
82 borde3_bajo_cono,borde4_bajo_cono,borde4_sinBG,borde2_bajo_cono,u,...
83 estructura);
84
85 %% Paso 7. Representación de la sensibilidad de todas las configuraciones
86 %% posibles de los bordes de PBG a modo de "volumen" mediante 'slices'
87
88 %Para poder calcular la sensibilidad es necesario tener los datos de
89 %simulación del sensor con SiO2 por encima y con H2O por encima. Estos
90 %datos se cargan para poder operar con ellos
91 load('matrizBG_SiO2.mat')
92 load('matrizBG_H2O.mat')
93
94 [vol4D_sensibilidad_borde1,vol4D_sensibilidad_borde2,...
95 vol4D_sensibilidad_borde3,vol4D_sensibilidad_borde4,u]=...
96 creator_vol_sensibilidad(matrizBG_SiO2,matrizBG_H2O,u,estructura);
97
98 %% Paso 8. Representación de la tolerancia de todas las configuraciones
99 %% posibles de los bordes de PBG a modo de "volumen" mediante
100 %% 'slices'
101
102
[vol4D_tolerancia_borde1,vol4D_tolerancia_borde2,vol4D_tolerancia_borde3,...
103 vol4D_tolerancia_borde4,X_int_borde1,Y_int_borde1,Z_int_borde1,...
104 W_int_borde1,tolerancia_borde1,X_int_borde2,Y_int_borde2,Z_int_borde2,...
105 W_int_borde2,tolerancia_borde2,X_int_borde3,Y_int_borde3,Z_int_borde3,...
106 W_int_borde3,tolerancia_borde3,X_int_borde4,Y_int_borde4,Z_int_borde4,...
107 W_int_borde4,tolerancia_borde4,u]=creator_vol_tolerancia...
108 (matrizBG,num_interv,u,estructura);
109
110 %% Paso 9. Representación de los planos donde los bordes de PBG se
111 %% corresponden con lambda=1550 nm añadiendo como cuarto parámetro
112 %% la sensibilidad
113
114
[mapacolor4D_plano1_sens,mapacolor4D_plano2_sens,mapacolor4D_plano3_sens,...
115 mapacolor4D_plano4_sens,u]=creator_mapacolor_4D_sensibilidad(matrizBG,...
116 matrizBG_SiO2,borde1_dist,borde2_dist,borde3_dist,borde4_dist,...
117 borde3_bajo_cono,borde4_bajo_cono,borde4_sinBG,borde2_bajo_cono,u,...
118 estructura);
119
120 %% Paso 10. Representación de los planos donde los bordes de PBG se
121 %% corresponden con lambda=1550 nm añadiendo como cuarto parámetro

```

```

122 %% la tolerancia
123
124 [mapacolor4D_plano1_tol,mapacolor4D_plano2_tol,mapacolor4D_plano3_tol,...
125 mapacolor4D_plano4_tol,u]=creator_mapacolor_4D tolerancia(X_int_borde1,...
126 Y_int_borde1,Z_int_borde1,W_int_borde1,tolerancia_borde1,X_int_borde2,...
127 Y_int_borde2,Z_int_borde2,W_int_borde2,tolerancia_borde2,X_int_borde3,...
128 Y_int_borde3,Z_int_borde3,W_int_borde3,tolerancia_borde3,X_int_borde4,...
129 Y_int_borde4,Z_int_borde4,W_int_borde4,tolerancia_borde4,num_interv,u,...
130 estructura);
131
132 end

```

## 1.2 Simulación MPB

### 1.2.1 *filescreator3D.m*

```

1 function filescreator3D(amin,aintervl,amax,wimin,wiintervl,wimax,wemin,...
2 weintervl,wemax,estructura)
3 %
4 %% Creación de archivos .ctl de simulación.
5 %% Para realizar la simulación de las diferentes configuraciones para
6 %% un mismo sensor se automatiza la creación de los ficheros .ctl para
7 %% todas las combinaciones de configuraciones posibles. En cada archivo
8 %% .ctl se define las características de diseño correspondientes.
9 %
10 %% Entrada:
11 % amin: valor mínimo que tomará la variable a
12 % aintervl: valor constante que indica la diferencia de tamaño entre un
13 % valor de a y su siguiente (separación equiespaciada entre amin y amax)
14 %
15 % amax: valor máximo que tomará la variable a
16 % wimin: valor mínimo que tomará la variable wi
17 % wiintervl: valor constante que indica la diferencia de tamaño entre un
18 % valor de wi y su siguiente (separación equiespaciada entre wimin y wimax)
19 %
20 % wimax: valor máximo que tomará la variable wi
21 % wemin: valor mínimo que tomará la variable we
22 % weintervl: valor constante que indica la diferencia de tamaño entre un
23 % valor de we
24 %
25 % y su siguiente (separación equiespaciada entre wemin y wemax)
26 %
27 % wemax: valor máximo que tomará la variable we
28 %
29 %% Salida:
30 % La salida de la función corresponde a cada uno de los archivos .ctl
31 % creados
32 %
33 %
34 %
35 %% Creación de los archivos .ctl a partir de la lectura y modificación de
36 %% un 'archivo modelo'
37
38 % El bucle genera todas las posibles combinaciones a partir de los datos
39 % de entrada
40 for i=amin:aintervl:amax
41 for j=wimin:wiintervl:wimax
42 for k=wemin:weintervl:wemax
43 % Se crea el título del archivo
44 switch estructura

```

```

45 case 'corr3D'
46 titulo=['corr3D-a',num2str(i),'-wi',num2str(j),'-we',...
47 num2str(k),'-h220-upper_DIW-w460-5bandas.ct1'];
48 case 'corr3Dinv'
49 titulo=['corr3Dinv-a',num2str(i),'-wi',num2str(j),...
50 '-we',num2str(k),'-h220-upper_DIW-w460-5bandas.ct1'];
51 end
52
53 %Se genera el archivo y su correspondiente identificador
54 %'fileID'. 'w+' indica que el archivo creado presenta
55 %propiedades de lectura y modificación
56 fileID = fopen(titulo,'w+');
57
58 %A partir del 'archivo modelo' se busca la parte del texto
59 %donde se definen las variables. La expresión 'a\s\d{3}' en la
60 %función regexp busca en 'text' aquellas 'a' seguidas de un
61 %espacio en blanco y tres números
62 switch estructura
63 case 'corr3D'
64 text=fileread('corr3D-a380-wi100-we1500-h220-upper_DIW-w460-5bandas.ct1');
65 case 'corr3Dinv'
66 text=fileread('corr3Dinv-a400-wi200-we100-h220-upper_DIW-w460-
67 5bandas.ct1');
68
69 expr1='a\s\d{3}';
70 expr2='wi\s\d{3}';
71 expr3='we\s\d{3}';
72 posicion_a=regexp(text,expr1);
73 posicion_wi=regexp(text,expr2);
74 posicion_we=regexp(text,expr3);
75 a_string=(num2str(i));
76
77 %Se sustituye los valores de las variables en las posiciones
78 %antes obtenidas y se ajusta el texto en función del tamaño
79 %de dichos números
80 if length(a_string)==3
81 text(posicion_a+2:posicion_a+4)=a_string;
82 elseif length(a_string)==4
83 text(posicion_a+2:posicion_a+6)=[a_string,')'];
84 elseif length(a_string)==2
85 text(posicion_a+2:posicion_a+5)=[a_string,') '];
86 elseif length(a_string)==1
87 text(posicion_a+2:posicion_a+5)=[a_string,') '];
88 end
89
90 wi_string=(num2str(j));
91 if length(wi_string)==3
92 text(posicion_wi+3:posicion_wi+5)=wi_string;
93 elseif length(wi_string)==4
94 text(posicion_wi+3:posicion_wi+7)=[wi_string,')'];
95 elseif length(wi_string)==2
96 text(posicion_wi+3:posicion_wi+6)=[wi_string,') '];
97 elseif length(wi_string)==1
98 text(posicion_wi+3:posicion_wi+6)=[wi_string,') '];
99 end
100
101 we_string=(num2str(k));
102 if length(we_string)==4
103 text(posicion_we+3:posicion_we+6)=we_string;
104 elseif length(we_string)==3
105 text(posicion_we+3:posicion_we+7)=[we_string,') '];
106 elseif length(we_string)==2
107 text(posicion_we+2:posicion_we+7)=[we_string,') '];
108 elseif length(we_string)==1
109 text(posicion_we+3:posicion_we+7)=[we_string,') '];
110 end

```

```

111
112 %Se guarda el texto modificado en el archivo creado
113 %anteriormente y se cierra
114 fprintf(fileID,text);
115 fclose(fileID);
116 end
117 end
118 end
119 end

```

## 1.2.2 simulaciones\_script\_creator3D.m

```

1 function
simulaciones_script_creator3D(amin,aintervl,amax,wimin,wiintervl,...
2 wimax,wemin,weintervl,wemax,estructura)
3 %
4 % %%Creación del archivo que, a partir de los ficheros .ctl y de la
5 % %%simulación, obtiene,entre otros, los archivos .ypar, .zpar y .fre
6 % %%necesarios para su posterior estudio.
7 %
8 % %%Se crea una matriz 3D con todas las combinaciones posibles. El tamaño
9 % %%de la tercera dimensión indica la cantidad de repeticiones de la
10 % %%simulación que se realizará para los diferentes valores de las
11 % %%variables. A partir de esta matriz se modifica el texto%a conveniencia
12 % %%para en un único archivo poder realizar todas las simulaciones en
13 % %%cadena(sin necesidad de simular manualmente cada combinación de
14 % %%parámetros).
15 %
16 %%Entrada:
17 % amin: valor mínimo que tomará la variable a
18 % aintervl: valor constante que indica la diferencia de tamaño entre un
19 % valor de a y su siguiente (separación equiespaciada entre amin y amax)
20 %
21 % amax: valor máximo que tomará la variable a
22 % wimin: valor mínimo que tomará la variable wi
23 % wiintervl: valor constante que indica la diferencia de tamaño entre un
24 % valor de wi y su siguiente (separación equiespaciada entre wimin y wimax)
25 %
26 % wimax: valor máximo que tomará la variable wi
27 % wemin: valor mínimo que tomará la variable we
28 % weintervl: valor constante que indica la diferencia de tamaño entre un
29 % valor de we y su siguiente (separación equiespaciada entre wemin y wemax)
30 %
31 % wemax: valor máximo que tomará la variable we
32 %
33 %%Salida:
34 % Script de simulación
35 %
36 %
37 %
38 %% Paso 1. Determinación de variables internas de la función
39
40 varfun=[amin,aintervl,amax,wimin,wiintervl,wimax,wemin,weintervl,wemax];
41 %Se obtiene el número de parámetros diferentes con los que se está
42 %trabajando
43 numvar=length(varfun)/3;
44 %Se calcula el número de combinaciones posibles entre parámetros
45 y=((amax-amin)/aintervl+1)*((wimax-wimin)/wiintervl+1)*((wemax-
wemin)...
46 /weintervl+1);
47 %El valor de 'z' se conoce previamente, representa la cantidad de veces
48 %que se va a requerir el valor de un parámetro
49 z=9;

```

```

50 %Se crea la matriz A en un primer momento rellena de ceros. En las líneas
51 %de código siguientes se irá rellenoando dicha matriz con todas las
52 %posibles combinaciones entre parámetros.
53 A=zeros(numvar,z,y);
54 s=1;
55 t=1;
56
57 %% Paso 2. Rellenado de la matriz A
58
59 %Cada submatriz de A repite el mismo valor por fila, esto se realiza así
60 %debido a que la forma de leer una matriz de la función 'fprintf' es por
61 %columnas. Es por esto que,para poder nombrar en el texto los parámetros
62 %de manera correcta, se elige esta metodología. Por otro lado, las
63 %diferentes combinaciones de las variables m,n,p se guardan en las
64 %diferentes submatrices de A
65 for m=amin:aintervl:amax
66 for n=wimin:wiintervl:wimax
67 for p=wemin:weintervl:wemax
68 %Mientras s sea menor o igual a la cantidad de matrices creadas
69 %en la tercera dimensión de A se ejecutarán las sentencias
70 %siguientes
71 while s<=y
72 for r=1:numvar
73 %Se rellena la primera fila de la matriz A con el valor
74 %de la variable m
75 if r==1
76 A(r,:,s)=m;
77
78 %Se rellena la segunda fila de la matriz A con el valor
79 %de la variable n
80 elseif r==2
81 A(r,:,s)=n;
82
83 %Se rellena la tercera fila de la matriz A con el valor
84 %de la variable p
85 else
86 A(r,:,s)=p;
87 end
88 end
89 s=s+1;
90 break
91 end
92 end
93 end
94 end
95
96 %% Paso 3. Creación del scrip de simulación
97
98 %Se genera el archivo y su correspondiente identificador 'fileID' mediante
99 %la función fopen. 'w+' indica que el archivo creado presenta propiedades
100 %de lectura y modificación
101 switch estructura
102 case 'corr3D'
103 fileID = fopen('simulaciones_multiple_scriptCorr3D','w+');
104 case 'corr3Dinv'
105 fileID = fopen('simulaciones_multiple_scriptCorr3Dinv','w+');
106 end
107
108 %Para poder utilizar todas las combinaciones de matrices que se han creado
109 %anteriormente se extraen dichas matrices de manera individual en una
nueva
110 %matriz B. Los valores almacenados en esta se van sustituyendo por orden
en
111 %las variables de formatSpec definidas como %d. El porcentaje de '%d'
112 %indica la creación de un operador de formato,la 'd' pertenece a un
113 %conjunto de letras ya asignadas que muestran el tipo de formato de la
114 %variable. En este caso, indica que dicha variable será un número entero

```

```

115 %de base 10.
116 while t<=y
117 B=A(:, :, t);
118 switch estructura
119 case 'corr3D'
120 formatSpec=['\nmpb corr3D-a%d-wi%d-we%d-h220-upper_DIW-w460-5bandas.ctl >
'...'
121 'corr3D-a%d-wi%d-we%d-h220-upper_DIW-w460-5bandas.out '...'
122 '\ngrep freq corr3D-a%d-wi%d-we%d-h220-upper_DIW-w460-5bandas.out > '...'
123 'corr3D-a%d-wi%d-we%d-h220-upper_DIW-w460-5bandas.fre '...'
124 '\ngrep zpar corr3D-a%d-wi%d-we%d-h220-upper_DIW-w460-5bandas.out > '...'
125 'corr3D-a%d-wi%d-we%d-h220-upper_DIW-w460-5bandas.zpar '...'
126 '\ngrep ypar corr3D-a%d-wi%d-we%d-h220-upper_DIW-w460-5bandas.out > '...'
127 'corr3D-a%d-wi%d-we%d-h220-upper_DIW-w460-5bandas.ypar '...'
128 '\nmpb-data -r -x 5 epsilon.h5 \nh5tovtk -d data-new epsilon.h5 '...'
129 '\nmv epsilon.vtk corr3D-a%d-wi%d-we%d-h220-upper_DIW-w460-5bandas-
epsilon.vtk \n '];
130 case 'corr3Dinv'
131 formatSpec=['\nmpb corr3Dinv-a%d-wi%d-we%d-h220-upper_DIW-w460-5bandas.ctl
> '...'
132 'corr3Dinv-a%d-wi%d-we%d-h220-upper_DIW-w460-5bandas.out '...'
133 '\ngrep freq corr3Dinv-a%d-wi%d-we%d-h220-upper_DIW-w460-5bandas.out >
'...'
134 'corr3Dinv-a%d-wi%d-we%d-h220-upper_DIW-w460-5bandas.fre '...'
135 '\ngrep zpar corr3Dinv-a%d-wi%d-we%d-h220-upper_DIW-w460-5bandas.out >
'...'
136 'corr3Dinv-a%d-wi%d-we%d-h220-upper_DIW-w460-5bandas.zpar '...'
137 '\ngrep ypar corr3Dinv-a%d-wi%d-we%d-h220-upper_DIW-w460-5bandas.out >
'...'
138 'corr3Dinv-a%d-wi%d-we%d-h220-upper_DIW-w460-5bandas.ypar '...'
139 '\nmpb-data -r -x 5 epsilon.h5 \nh5tovtk -d data-new epsilon.h5 '...'
140 '\nmv epsilon.vtk corr3Dinv-a%d-wi%d-we%d-h220-upper_DIW-w460-5bandas-
epsilon.vtk \n '];
141 end
142
143 %Se modifica el archivo en aquellos lugares del string 'formatSpec'
144 %donde se ha creado un operador de formato '%d'
145 fprintf(fileID, formatSpec, B);
146 t=t+1;
147 end
148
149 %Se cierra el archivo
150 fclose(fileID);
151 end

```

## 1.3 Detección de PBG

### 1.3.1 *dibbandasMPB\_3Dslab\_select\_parity\_wavelength.m*

Esta función NO se ha realizado en el TFG

```

1 function [kaux,a,sal]=dibbandasMPB_3Dslab_select_parity_wavelength(titulo)
2 filename=titulo;
3 %%%%%%%%% we indicate the horizontal and vertical parities that we want to
4 %%%%%%%%% extract (even, odd or both)
5 %%%%%%%%% we also indicate the parity threshold for extracting the band
6 selected_parity_hor='even';
7 selected_parity_vert='even';
8 parity_threshold_hor=0.05; % always a positive value
9 parity_threshold_vert=0.05; % always a positive value
10 %%%% important --> the even horizontal parity has minus sign
11

```

```

12 %%%%%%%%% we look for the period value within the filename (looking for the
'-a' string)
13 pos1=strfind(filename, '-a');
14 pos_aux1=strfind(filename, '-');
15 pos_aux2=find(pos_aux1>pos1);
16 pos2=pos_aux1(pos_aux2(1));
17 a=str2num(filename(pos1+2:pos2-1));
18
19 file_info=importdata([filename, '.fre']);
20 try % if the file hasn't text, it will provide an error
21 X=file_info.data; % in this way, we can ignore the text on the file
22 catch
23 X=file_info;
24 end
25 X=X';
26 k_vector=X(5,:);
27 bands=X(6:end,:);
28
29 %%%%%%%%% we check the parities file and change the sign if it's not even
30 try % if the file is empty (we haven't calculated the parities), it will
provide an error
31 file_info=importdata([filename, '.ypar']);
32 PAR_HOR=file_info.data; % in this way, we can ignore the text on the file
33 catch
34 PAR_HOR=load([filename, '.ypar']);
35 end
36
37 if ~isempty(PAR_HOR)
38 PAR_HOR=PAR_HOR(:,2:end)';
39 else
40 PAR_HOR=ones(size(bands));
41 selected_parity_hor='not available';
42 end
43
44 try % if the file is empty (we haven't calculated the parities), it will
provide an error
45 file_info=importdata([filename, '.zpar']);
46 PAR_VERT=file_info.data; % in this way, we can ignore the text on the file
47 catch
48 PAR_VERT=load([filename, '.zpar']);
49 end
50
51 if ~isempty(PAR_VERT)
52 PAR_VERT=PAR_VERT(:,2:end)';
53 else
54 PAR_VERT=ones(size(bands));
55 selected_parity_vert='not available';
56 end
57
58 pos1=strfind(filename, '-a');
59 if ~isempty(strfind(selected_parity_hor, 'even'))
60 PAR_HOR=-PAR_HOR;
61 disp('horizontal even')
62 elseif ~isempty(strfind(selected_parity_hor, 'odd'))
63 disp('horizontal odd')
64 elseif ~isempty(strfind(selected_parity_hor, 'both'))
65 PAR_HOR=abs(PAR_HOR);
66 disp('horizontal both')
67 end
68
69 if ~isempty(strfind(selected_parity_vert, 'even'))
70 disp('vertical even')
71 elseif ~isempty(strfind(selected_parity_vert, 'odd'))
72 PAR_VERT=-PAR_VERT;
73 disp('vertical odd')
74 elseif ~isempty(strfind(selected_parity_vert, 'both'))
75 PAR_VERT=abs(PAR_VERT);

```

```

76 disp('vertical both')
77 end
78
79
80 pos=find((PAR_HOR>parity_threshold_hor)&(PAR_VERT>parity_threshold_vert));
81 bands_parity=zeros(size(bands));
82 bands_parity(pos)=bands(pos);
83
84 for ind1=1:size(bands_parity,1)
85 for ind2=1:size(bands_parity,2)
86 if bands_parity(ind1,ind2)==0
87 pos=find(bands_parity((ind1+1):end,ind2)~=0);
88 if length(pos)~=0
89 bands_parity(ind1,ind2)=bands_parity(pos(1)+ind1,ind2);
90 bands_parity(pos(1)+ind1,ind2)=0;
91 end
92 end
93 end
94 end
95
96 pos=find(sum(bands_parity,2)>0);
97 bands_parity=bands_parity(pos,:);
98 if length(find(bands_parity(end,:))~=1) % the last file of the matrix only
has one element, so it's not possible to make a spline
99 bands_parity=bands_parity(1:end-1,:);
100 end
101 bands_parity(bands_parity==0)=NaN;
102
103 %%%%%%%%% we make a spline of the data to plot it with higher resolution
104 s=size(bands_parity);
105 sal=zeros(s(1),100);
106
107 kaux=linspace(k_vector(1),k_vector(end),100);
108 for ind=1:s(1)
109 sal(ind,:)=spline(k_vector,bands_parity(ind,:),kaux);
110 end
111
112 figure
113 p1=plot(kaux,a./sal(:,1:100),'b-');
114 hold on
115 plot(k_vector,a./bands_parity,'b*')
116 grid
117 title(['Vertical = ' selected_parity_vert ' - Horizontal = '
selected_parity_hor])
118 xlabel('k_{aux}'); ylabel('\lambda');
119
120 p2=plot(kaux,a./kaux*1.45,'LineWidth',5,'Color',0*[1 1 1]); % maximum
period for SiO2 upper cladding = 535 nm
121 % plot(kaux,a./kaux*1,'LineWidth',5,'Color',0*[1 1 1])
122
123 legend([p1(1,1) p2],['a = ',num2str(a),' nm'],'Cono de
luz','Location','SouthWest');
124 % axis([min(kaux) max(kaux) 1450 1700])
125 axis([0.3 max(kaux) 1100 2000])
126 end

```

### 1.3.2 see\_all\_graphics.m

```
1 function
see_all_graphics(amin,aintervl,amax,wimin,wiintervl,wimax,wemin,weintervl,wema
x,estructura)
2 %
3 % %%Representación de todas las bandas de PBG para cada configuración
4 % %%simulada
5 %
6 %%Entrada:
7 % amin: valor mínimo que tomará la variable a
8 % aintervl: valor constante que indica la diferencia de tamaño entre un
9 % valor de a y su siguiente (separación equiespaciada entre amin y amax)
10 % amax: valor máximo que tomará la variable a
11 % wimin: valor mínimo que tomará la variable wi
12 % wiintervl: valor constante que indica la diferencia de tamaño entre un
13 % valor de wi y su siguiente (separación equiespaciada entre wimin y wimax)
14 % wimax: valor máximo que tomará la variable wi
15 % wemin: valor mínimo que tomará la variable we
16 % weintervl: valor constante que indica la diferencia de tamaño entre un
17 % valor de we y su siguiente (separación equiespaciada entre wemin y wemax)
18 % wemax: valor máximo que tomará la variable we
19 % estructura: tipo de estructura estudiada
20 %
21 %%Salida:
22 % Diagrama de bandas para todas las configuraciones posibles introducidas
23 % mediante los inputs
24 %
25 %
26 %
27 %% Se crea un bucle que genera todas las posibles combinaciones entre 'a',
28 %% 'wi, e 'we'. Se invoca a los ficheros generados durante la simulación
29 %% con MPB llamando a la función
'dibbandasMPB_3Dslab_select_parity_wavelength'
30 %% que representa las bandas de cada posible configuración
31
32 for m=amin:aintervl:amax
33 for n=wimin:wiintervl:wimax
34 for p=wemin:weintervl:wemax
35
36 %Se modifica del titulo del archivo las variables 'a','wi','we'
37 switch estructura
38 case 'corr3D'
39 titulo=['corr3D-a',num2str(m),'-wi',num2str(n),'-we',...
40 'num2str(p),'-h220-upper_DIW-w460-5bandas'];
41 case 'corr3Dinv'
42 titulo=['corr3Dinv-a',num2str(m),'-wi',num2str(n),'-we',...
43 'num2str(p),'-h220-upper_DIW-w460-5bandas'];
44 end
45 dibbandasMPB_3Dslab_select_parity_wavelength(titulo);
46 end
47 end
48 end
```

### 1.3.3 find\_BG\_3D.m

```
1 function [borde1,borde2,borde3,borde4,borde3_bajo_cono,borde4_bajo_cono,...
2 banda3_sinBG,borde1_dist,borde2_dist,borde3_dist,borde4_dist,...
3 borde2_bajo_cono]=find_BG_3D(kaux,a,sal,estructura)
4 %
5 % %%Extracción de los bordes superiores e inferiores del PBG 1 y 2 y
6 % %%detección de anomalías en dichos bordes para cada combinación de
7 % %%configuraciones diseñado. Las anomalías estudiadas son: borde de PBG
8 % %%bajo el cono de luz, inexistencia del borde de PBG o distorsión
9 % %%de este por solape con otra lambda)
10 %
11 %%Entrada
12 % kaux: vector que contiene los valores equiespaciados del vector
'k_vector'
13 % a: valor de la variable 'a' (periodo)
14 % sal: almacena los valores de cada borde de PBG para las diferentes
bandas.
15 % Cada fila de dicha variable corresponde a una banda determinada
16 %
17 %%Salida:
18 % borde1: Borde superior de PBG1
19 % borde2: Borde inferior de PBG1
20 % borde3: Borde superior de PBG2
21 % borde4: Borde inferior de PBG2
22 % borde3_bajo_cono: Vale '1' si el borde superior de PBG2 está por debajo
23 % del cono de luz
24 % borde4_bajo_cono: Vale '1' si el borde inferior de PBG2 está por debajo
25 % del cono de luz
26 % banda3_sinBG: Vale '1' si no existe el borde inferior de PBG2 (al no
27 % presentarse ningún pico en la banda 3)
28 % borde1_dist: Vale '1' si el borde superior de PBG1 está distorsionado
29 % borde2_dist: Vale '1' si el borde inferior de PBG1 está distorsionado
30 % borde3_dist: Vale '1' si el borde superior de PBG2 está distorsionado
31 % borde4_dist: Vale '1' si el borde inferior de PBG2 está distorsionado
32 %
33 %NOTA: Únicamente se estudian los casos donde el borde de PBG se encuentra
34 % bajo el cono de luz para borde3 y 4. Lo mismo sucede en el estudio de las
35 % bandas que no presentan BG. Después de un análisis del comportamiento de
36 % las bandas y de la extracción de sus patrones se llega a la conclusión de
37 % que solamente interesa el estudio de dichas bandas en concreto puesto que
38 % son las que presentan anomalías
39 %
40 %
41 %
42 %% Paso 1. Se crea una matriz con las coordenadas del cono de luz
43
44 %Los valores de la primera fila corresponden al eje x (kaux), los valores
45 %de la segunda fila corresponden al eje y (lamda)
46 coord_conoluz(1,:)=kaux;
47 coord_conoluz(2,:)=a./kaux*1.45;
48
49 %% Paso 2. Se localiza el borde superior del primer PBG. Este se encuentra
50 %% en la primera banda (de arriba abajo en las gráficas)
51
52 %Se almacenan las coordenadas de la primera banda
53 coord_banda1(1,:)=kaux;
54 coord_banda1(2,:)=a./sal(1,:);
55
56 %Se conoce que la banda 1 siempre estará por encima del cono de luz por lo
57 %que localizar el borde de band gap únicamente se reduce a encontrar
58 %el valor de lambda al que le corresponde kaux=0.5
59 borde1=coord_banda1(2,find(coord_banda1(1,')==0.5));
60
```

```

61 %% Paso 3. Se localiza el borde inferior del primer PBG y y el borde
62 %% superior del segundo PBG. Se encuentran en la segunda banda
63
64 %Se almacenan las coordenadas de la segunda banda
65 coord_banda2(1,:)=kaux;
66 coord_banda2(2,:)=a./sal(2,:);
67
68 %Se obtiene el borde inferior del primer PBG (valor de lambda al que
69 %corresponde kaux=0.5). Después se localiza el valor del borde del segundo
70 %PBG, este será el valor mínimo de todos los que presenta la segunda banda
71 borde2=coord_banda2(2,find(coord_banda2(1,')==0.5));
72
73 switch estructura
74 case 'corr3D'
75 borde3=min(coord_banda2(2,:));
76
77 %% Paso 4. Detección de anomalías en los bordes de PBG 'borde1','borde2' y
78 %% 'borde3'
79
80 %Se separan los valores de la banda que se encuentran por encima y
81 %por debajo del cono de luz puesto que presentan respuestas
82 %diferentes en función de su localización. Si por encima del cono
83 %de luz existe algún valor de lambda superior al PBG este no será
84 %completo por lo que su respuesta se mostrará distorsionada
85 posiciones_sup=find(coord_banda2(2,*)>coord_conoluz(2,));
86
87 %Es necesario eliminar el valor de 'borde2' para evitar que después
88 %se genere un falso positivo en la detección de lambdas IGUALES a
89 %esta
90 sup_banda2=coord_banda2(2,posiciones_sup(1:length(posiciones_sup)-1));
91
92 %Si el vector 'sup_banda2' no está vacío se procede a calcular,
93 %para cada valor, su posición relativa respecto al PBG
94 if isempty(sup_banda2)==0
95 comparacion1=sup_banda2(1,find(sup_banda2>=borde1));
96 comparacion2=sup_banda2(1,find(sup_banda2>=borde2));
97
98 %Se marca con 1 aquellos valores de la banda que presentan
99 %distorsión
100 if isempty(comparacion1)
101 borde1_dist=0;
102 else
103 borde1_dist=1;
104 end
105
106 if isempty(comparacion2)
107 borde2_dist=0;
108 else
109 borde2_dist=1;
110 end
111
112 %Se detecta si el borde superior del PBG2 se encuentra por
113 %encima o por debajo del cono de luz. Cuando se da este último
114 %caso, se añade un 1
115 d=find(borde3==sup_banda2,1);
116 if isempty(d)
117 borde3_bajo_cono=1;
118 else
119 borde3_bajo_cono=0;
120 end
121
122 %Cuando todos los valores de la banda2 se encuentran por debajo del
123 %cono de luz 'borde3' se encontrará bajo dicho cono. El tipo de
124 %distorsión de 'borde1' y 'borde2' no tienen por qué verse afectado
125 %por lo que se marca como 0
126 else
127 borde3_bajo_cono=1;

```

```

128 borde1_dist=0;
129 borde2_dist=0;
130 end
131
132 %% Paso 5. Se localiza el borde inferior del segundo posible PBG. Se
133 %% encuentra en la tercera banda
134
135 coord_banda3(1,:)=kaux;
136 coord_banda3(2,:)=a./sal(3,:);
137
138 %Para obtener 'borde4' se busca el pico máximo de la tercera banda
139 try borde4=findpeaks(coord_banda3(2,:));
140
141 %% Paso 6. Detección de anomalías en los bordes de PBG 'borde3' y 'borde4'
142
143 posiciones_sup=find(coord_banda3(2,*)>coord_conoluz(2,:));
144 sup_banda3=coord_banda3(2,posiciones_sup(1:length(posiciones_sup)));
145 if isempty(sup_banda3)==0
146 comparacion1=sup_banda3(1,find(sup_banda3>=borde3));
147 %Solo se usan valores mayores a borde4 para evitar las
148 %repeticiones de dicho valor que suceden cuando el pico
149 %máximo de la banda 3 se mantiene en varias posiciones
150 comparacion2=sup_banda3(1,find(sup_banda3>borde4));
151 if isempty(comparacion1)
152 borde3_dist=0;
153 else
154 borde3_dist=1;
155 end
156
157 if isempty(comparacion2)
158 borde4_dist=0;
159 else
160 borde4_dist=1;
161 end
162
163 d=find(borde4==sup_banda3);
164 if isempty(d)
165 borde4_bajo_cono=1;
166 else
167 borde4_bajo_cono=0;
168 end
169
170 else
171 borde4_bajo_cono=1;
172 borde3_dist=0;
173 borde4_dist=0;
174 end
175 %Si la banda presenta PBG la siguiente variable valdrá 0, si no
176 %tiene PBG valdrá 1
177 banda3_sinBG=0;
178
179 %Si la banda no presenta ningún pico máximo no existirá el borde
180 %inferior del PBG2
181 catch borde4=NaN;
182 borde4_bajo_cono=0;
183 banda3_sinBG=1;
184 borde3_dist=0;
185 borde4_dist=0;
186 end
187
188 case {'corr3Dinv','corr3Dslot'}
189 borde3=NaN;
190 borde4=NaN;
191 borde3_bajo_cono=NaN;
192 borde4_bajo_cono=NaN;
193 banda3_sinBG=NaN;
194 borde3_dist=NaN;

```

```

195 borde4_dist=NaN;
196
197 posiciones_sup=find(coord_banda2(2,:)>coord_conoluz(2,:));
198
199 sup_banda2=coord_banda2(2,posiciones_sup(1:length(posiciones_sup)));
200
201 if isempty(sup_banda2)==0
202 comparacion1=sup_banda2(1,find(sup_banda2(1:end-1)>=borde1));
203 comparacion2=sup_banda2(1,find(sup_banda2(1:end-1)>=borde2));
204
205 if isempty(comparacion1)
206 borde1_dist=0;
207 else
208 borde1_dist=1;
209 end
210
211 if isempty(comparacion2)
212 borde2_dist=0;
213 else
214 borde2_dist=1;
215 end
216
217 d=find(borde2==sup_banda2,1);
218 if isempty(d)
219 borde2_bajo_cono=1;
220 else
221 borde2_bajo_cono=0;
222 end
223
224 else
225 borde2_bajo_cono=1;
226 borde1_dist=0;
227 borde2_dist=0;
228 end
229
230 end
231 end

```

### 1.3.4 *creator\_matrizBG\_3D.m*

```

1 function [matrizBG,a_vector,wi_vector,we_vector,num_interv]= ...
2 creator_matrizBG_3D(amin,aintervl,amax,wimin,wiintervl,wimax,wemin,...
3 weintervl,wemax,estructura)
4 %
5 % %%Obtención de los bordes de photonic band gap existentes para las 4
6 % %%primeras bandas de simulación de cada posible configuración. Además,
7 % %%se detectan aquellos bordes de PBG que se encuentran bajo el cono de
8 % %%luz, aquellos donde hay distorsión por solape de lambdas estimuladas
9 % %%a la vez y también se señalan aquellas bandas que no presentan PBG.
10 % %%Finalmente se obtienen las interpolaciones de los valores originales
11 % %%de 'a','wi' e 'we'
12 %
13 % %%NOTA: En esta función se llama a otras dos funciones:
14 % 'dibbandasMPB_3Dslab_select_parity_wavelength' que calcula las bandas de
15 % cada configuración y 'find_BG_3D' que obtiene cada columna de 'matrizBG'
16 %
17 %%Entrada:
18 % amin: valor mínimo que tomará la variable a
19 % aintervl: valor constante que indica la diferencia de tamaño entre un
20 % valor de a y su siguiente (separación equiespaciada entre amin y amax)
21 % amax: valor máximo que tomará la variable a

```

```

22 % wimin: valor mínimo que tomará la variable wi
23 % wiintervl: valor constante que indica la diferencia de tamaño entre un
24 % valor de wi y su siguiente (separación equiespaciada entre wimin y wimax)
25 % wimax: valor máximo que tomará la variable wi
26 % wemin: valor mínimo que tomará la variable we
27 % weintervl: valor constante que indica la diferencia de tamaño entre un
28 % valor de we y su siguiente (separación equiespaciada entre wemin y wemax)
29 % wemax: valor máximo que tomará la variable we
30 %
31 %%%Salida:
32 % matrizBG: matriz que contiene todos los datos originales de la simulación
33 % en base a los cuales se realizarán los posteriores análisis. En esta
34 % matriz cada columna representa una combinación de configuraciones
35 % posibles. Los valores de cada fila en orden descendiente son
36 % respectivamente: 'a','wi','we',borde superior de PBG1 (en banda 1),borde
37 % inferior de PBG1 (en banda 2),borde superior de PBG2 (en banda 3),borde
38 % inferior de PBG2 (en banda 4), borde superior de PBG1 bajo cono de luz,
39 % borde inferior de PBG1 bajo cono de luz, borde superior de PBG2 bajo cono
40 % de luz, borde inferior de PBG2 bajo cono de luz, banda 3 sin PBG, borde
41 % superior de PBG1 distorsionado, borde inferior de PBG1 distorsionado,
42 % borde superior de PBG2 distorsionado y borde inferior de PBG2
distorcionado
43
44 % a_vector: interpolación de los valores originales de 'a' utilizados en
45 % la simulación
46 % wi_vector: interpolación de los valores originales de 'wi'
47 % utilizados en la simulación
48
49 % we_vector: interpolación de los valores originales de 'we' utilizados en
50 % la simulación
51
52 % num_interv: indica el tamaño de equiespaciado de las interpolaciones de
53 % 'a_vector','wi_vector' y 'we_vector'
54 %
55 % Paso 1. Ejecución de la función
'dibbandasMPB_3Dslab_select_parity_wavelength'
56 % para la obtención de las bandas
57
58 u=1;
59 for p=amin:aintervl:amax
60 for q=wimin:wiintervl:wimax
61 for r=wemin:weintervl:wemax
62 switch estructura
63 case 'corr3D'
64 titulo=['corr3D-a',num2str(p),'-wi',num2str(q),'-we',num2str(r),...
65 '-h220-upper_DIW-w460-5bandas'];
66 case 'corr3Dinv'
67 titulo=['corr3Dinv-a',num2str(p),'-wi',num2str(q),'-we',num2str(r),...
68 '-h220-upper_DIW-w460-5bandas'];
69
70 end
71
72 [kaux,a,sal]=dibbandasMPB_3Dslab_select_parity_wavelength(titulo);
73
74 % Paso 2. Ejecución de la función 'find_BG_3D' para la obtención de cada
75 % columna de 'matrizBG'
76
77 [borde1,borde2,borde3,borde4,borde3_bajo_cono,borde4_bajo_cono,...
78 banda3_sinBG,borde1_dist,borde2_dist,borde3_dist,...
79 borde4_dist,borde2_bajo_cono]=find_BG_3D(kaux,a,sal,estructura);
80
81 % Paso 3. Concatenación de las columnas generadas en el paso 2 en una
82 % única matriz denominada 'matrizBG'
83
84 switch estructura
85 case 'corr3D'
86 matrizBG(:,u)=[p;q;r;borde1;borde2;borde3;borde4;...

```

```

87 borde3_bajo_cono;borde4_bajo_cono;banda3_sinBG;...
88 borde1_dist;borde2_dist;...
89 borde3_dist;borde4_dist];
90 case 'corr3Dinv'
91 matrizBG(:,u)=[p;q;r;borde1;borde2;borde2_bajo_cono;...
92 borde1_dist;borde2_dist];
93 end
94
95 u=u+1;
96 end
97 end
98 end
99
100 %% Paso 4. Interpolación de las coordenadas originales de la simulación
101
102 %Se crean tres vectores fila que contendrán la interpolación de los
valores
103 %de las variables originales equiespaciadas en el número de intervalos
104 %definido en 'num_interv'
105
106 num_interv=50;
107 a_vector=linspace(min(matrizBG(1,:)),max(matrizBG(1,:)),num_interv);
108 wi_vector=linspace(min(matrizBG(2,:)),max(matrizBG(2,:)),num_interv);
109 we_vector=linspace(min(matrizBG(3,:)),max(matrizBG(3,:)),num_interv);
110
111 end

```

## 1.4 Representación PBG

### 1.4.1 *creator\_mapacolor\_3D\_rectas.m*

```

1 function [mapacolor3D_rectas_borde1,mapacolor3D_rectas_borde2,...
2 mapacolor3D_rectas_borde3,mapacolor3D_rectas_borde4]=...
3 creator_mapacolor_3D_rectas(matrizBG,a_vector,wi_vector,we_vector)
4 %
5 % %%% Representación 3D mediante rectas de aquellas configuraciones para
6 % %%%las cuales los diferentes bordes de PBG se encuentran en lambda=1550
nm.
7 % %%% Además se añade la representación de las anomalías que presentan los
8 % %%%diferentes bordes.
9 %
10 %%%Entrada:
11 % matrizBG: matriz que contiene todos los datos originales de la simulación
12 % a_vector: Interpolación de los valores originales de 'a' utilizados en
13 % la simulación
14 %
15 % wi_vector: Interpolación de los valores originales de 'wi' utilizados en
16 % la simulación
17 %
18 % we_vector: Interpolación de los valores originales de 'we' utilizados en
19 %la simulación
20 %
21 %
22 %%%Salida:
23 % mapacolor3D_rectas_borde1: Representación 3D de aquellas configuraciones
24 % en las que el borde superior del PBG1 se encuentra en lambda=1550 nm
25
26 % mapacolor3D_rectas_borde2: Representación 3D de aquellas configuraciones
27 % en las que el borde inferior del PBG1 se encuentra en lambda=1550 nm
28
29 % mapacolor3D_rectas_borde3: Representación 3D de aquellas configuraciones
30 % en las que el borde superior del PBG2 se encuentra en lambda=1550 nm
31 %

```

```

32 % mapacolor3D_rectas_borde4: Representación 3D de aquellas configuraciones
33 % en las que el borde inferior del PBG2 se encuentra en lambda=1550 nm
34 %
35 %
36 %
37 %% Paso 1. Mallado de las coordenadas we,wi,a en las matrices X,Y,Z para
38 %% sus correspondientes valores de lambda almacenado en W
39
40 u=1;
41 [X,Y,Z]=meshgrid(we_vector,wi_vector,a_vector);
42 for m=4:4
43 %Eliminación de los posibles 'NaN' existentes en los bordes de PBG
44 pos_sinNaN=find(isnan(matrizBG(m,:))==0);
45 matrizBG_sinNaN=matrizBG(:,pos_sinNaN);
46 W=griddata(matrizBG_sinNaN(3,:),matrizBG_sinNaN(2,:),...
47 matrizBG_sinNaN(1,:),matrizBG_sinNaN(m,:),X,Y,Z);
48
49 %% Paso 2. Representación de las líneas donde los bordes de PBG se
50 %% encuentran en lambda=1550 nm
51
52 %Se obtienen las coordenadas deseadas a partir de su búsqueda en el mapa
53 %de color 2D. Se añade la tercera dimensión para poder representar las
54 %líneas en 3D
55 figure(u)
56 for n=1:size(W,3)
57 [coordenadas]=contour(X(:,:,1),Y(:,:,1),W(:,:,n),1550*[1 1], 'w');
58 coordenadas=coordenadas(:,2:length(coordenadas));
59 a=Z(1,1,n)*ones(1,length(coordenadas));
60 coordenadas(3,:)=a;
61 plot3(coordenadas(1,:),coordenadas(2,:),coordenadas(3,:))
62 box on
63 ax=gca;
64 ax.BoxStyle = 'full';
65 axis([700 1600 60 180 300 420])
66 hold on
67
68 end
69
70 %% Paso 3. Obtención y representación de las anomalías presentes en los
71 %% bordes de PBG
72
73 if m==4
74 %Coordenadas donde el borde 1 de PBG presenta distorsión
75 a_borde1_dist=matrizBG(1,find(matrizBG(11,:)));
76 wi_borde1_dist=matrizBG(2,find(matrizBG(11,:)));
77 we_borde1_dist=matrizBG(3,find(matrizBG(11,:)));
78
79 p2=plot3(we_borde1_dist,wi_borde1_dist,a_borde1_dist,'g*');
80 legend(p2,'Distorsión');
81
82 %Almacenamiento en la variable de salida de su gráfica
83 %correspondiente
84 mapacolor3D_rectas_borde1=figure(u);
85 elseif m==5
86 %Coordenadas donde el borde 2 de PBG presenta distorsión
87 a_borde2_dist=matrizBG(1,find(matrizBG(12,:)));
88 wi_borde2_dist=matrizBG(2,find(matrizBG(12,:)));
89 we_borde2_dist=matrizBG(3,find(matrizBG(12,:)));
90
91 p3=plot3(we_borde2_dist,wi_borde2_dist,a_borde2_dist,'c*');
92 legend(p3,'Distorsión')
93
94 mapacolor3D_rectas_borde2=figure(u);
95
96 elseif m==6
97 %Coordenadas donde el borde 3 de PBG se encuentra bajo el cono de
98 %luz

```

```

99 a_borde3_bajo_cono=matrizBG(1,find(matrizBG(8,:)));
100 wi_borde3_bajo_cono=matrizBG(2,find(matrizBG(8,:)));
101 we_borde3_bajo_cono=matrizBG(3,find(matrizBG(8,:)));
102
103 %Coordenadas donde el borde 3 de PBG presenta distorsión
104 a_borde3_dist=matrizBG(1,find(matrizBG(13,:)));
105 wi_borde3_dist=matrizBG(2,find(matrizBG(13,:)));
106 we_borde3_dist=matrizBG(3,find(matrizBG(13,:)));
107
108 p4=plot3(we_borde3_dist,wi_borde3_dist,a_borde3_dist,'y*');
109 p5=plot3(we_borde3_bajo_cono,wi_borde3_bajo_cono,a_borde3_bajo_cono,'ro');
110
111 legend([p5 p4], 'Bajo cono de luz', 'Distorsión');
112
113 mapacolor3D_rectas_borde3=figure(u);
114
115 elseif m==7
116 %Coordenadas donde el borde 4 de PBG se encuentra bajo el cono de
117 %luz
118 a_borde4_bajo_cono=matrizBG(1,find(matrizBG(9,:)));
119 wi_borde4_bajo_cono=matrizBG(2,find(matrizBG(9,:)));
120 we_borde4_bajo_cono=matrizBG(3,find(matrizBG(9,:)));
121
122 %Coordenadas donde el borde 4 de PBG no existe
123 a_borde4_sinBG=matrizBG(1,find(matrizBG(10,:)));
124 wi_borde4_sinBG=matrizBG(2,find(matrizBG(10,:)));
125 we_borde4_sinBG=matrizBG(3,find(matrizBG(10,:)));
126
127 %Coordenadas donde el borde 4 de PBG presenta distorsión
128 a_borde4_dist=matrizBG(1,find(matrizBG(14,:)));
129 wi_borde4_dist=matrizBG(2,find(matrizBG(14,:)));
130 we_borde4_dist=matrizBG(3,find(matrizBG(14,:)));
131
132 p6=plot3(we_borde4_dist,wi_borde4_dist,a_borde4_dist,'r*');
133 p7=plot3(we_borde4_bajo_cono,wi_borde4_bajo_cono,a_borde4_bajo_cono,'mo');
134 p8=plot3(we_borde4_sinBG,wi_borde4_sinBG,a_borde4_sinBG,'ks');
135
136 legend([p7 p8 p6], 'Bajo cono de luz', 'Borde de PBG no existe', ...
137 'Distorsión');
138
139 mapacolor3D_rectas_borde4=figure(u);
140
141 end
142
143 %Propiedades de las gráficas
144 titulo=['Configuraciones del borde ', num2str(u), ' de PBG en \lambda=1550
nm'];
145 title(titulo); xlabel('W_{e}'); ylabel('W_{i}'); zlabel('a');
146 c=colorbar;
147 c.Label.String = '\lambda';
148 set(gcf, 'position', get(groot, 'ScreenSize'));
149 u=u+1;
150 end
151 end

```

### 1.4.2 *creador\_mapacolor\_3D\_plano1550.m*

```
1 function [mapacolor3D_plano_borde1,mapacolor3D_plano_borde2,...
2 mapacolor3D_plano_borde3,mapacolor3D_plano_borde4,borde1_dist,...
3 borde2_dist,borde3_dist,borde4_dist,borde3_bajo_cono,borde4_bajo_cono,...
4 borde4_sinBG,u,borde2_bajo_cono]=creador_mapacolor_3D_plano1550...
5 (matrizBG,a_vector,wi_vector,we_vector,num_interv,estructura)
6 %
7 % %%Representación gráfica 3D del plano de cada conjunto de bordes de PBG
8 % %%que se estimulan para lambda=1550 nm
9 % %%Se obtienen las líneas 2D para las que lambda=1550 y se añade la
10 % %% tercera dimensión (la variable que falta). A continuación se
interpolan
11 % %%las coordenadas 3D para poder crear un mallado y finalmente el plano
12 % %%3D deseado. Además se marcan los bordes que presentan anomalías.
13 %
14 %%Entrada:
15 % matrizBG: matriz que contiene todos los datos originales de la simulación
16 % a_vector: Interpolación de los valores originales de 'a' utilizados en la
17 % simulación
18 %
19 % wi_vector: Interpolación de los valores originales de 'wi' utilizados en
20 % la simulación
21 %
22 % we_vector: Interpolación de los valores originales de 'we' utilizados en
23 % la simulación
24 %
25 % num_interv: indica el tamaño de equiespaciado de las interpolaciones de
26 % 'a_vector', 'wi_vector' y 'we_vector'
27 %
28 % estructura: nombre de la estructura de simulación
29 %
30 %%Salida:
31 % mapacolor3D_plano_borde1: Representación 3D de aquellas configuraciones
32 % en las que el borde superior del PBG1 se encuentra en lambda=1550 nm
33 %
34 % mapacolor3D_plano_borde2: Representación 3D de aquellas configuraciones
35 % en las que el borde inferior del PBG1 se encuentra en lambda=1550 nm
36 %
37 % mapacolor3D_plano_borde3: Representación 3D de aquellas configuraciones
38 % en las que el borde superior del PBG2 se encuentra en lambda=1550 nm
39 %
40 % mapacolor3D_plano_borde4: Representación 3D de aquellas configuraciones
41 % en las que el borde inferior del PBG2 se encuentra en lambda=1550 nm
42 %
43 % borde1_dist: matriz que contiene las coordenadas del borde superior del
44 % PBG1 que presentan distorsión
45 %
46 % borde2_dist: matriz que contiene las coordenadas del borde inferior del
47 % PBG1 que presentan distorsión
48 %
49 % borde3_dist: matriz que contiene las coordenadas del borde superior del
50 % PBG2 que presentan distorsión
51 %
52 % borde4_dist: matriz que contiene las coordenadas del borde inferior del
53 % PBG2 que presentan distorsión
54 %
55 % borde3_bajo_cono: matriz que contiene las coordenadas del borde superior
56 % del PBG2 que se encuentran bajo el cono de luz
57 %
58 % borde4_bajo_cono: matriz que contiene las coordenadas del borde inferior
59 % del PBG2 que se encuentran bajo el cono de luz
60 %
61 % borde4_sinBG: matriz que contiene las coordenadas donde no existe el
62 % borde inferior del PBG2
63 %
```

```

64 % borde2_bajo_cono: matriz que contiene las coordenadas del borde inferior
65 % del PBG1 que se encuentran bajo el cono de luz
66 %
67 %% Paso 1. Mallado de las coordenadas we,wi,a en las matrices X,Y,Z para
68 %% sus correspondientes valores de lambda almacenado en W
69
70 u=1;
71 [X,Y,Z]=meshgrid(we_vector,wi_vector,a_vector);
72
73 %Se elige el tipo de estructura
74 switch estructura
75 case 'corr3D'
76 f=4:7;
77 case 'corr3Dinv'
78 f=4:5;
79 end
80
81 for m=f
82
83 %Eliminación de los posibles 'NaN' existentes en los bordes de PBG
84 pos_sinNaN=find(isnan(matrizBG(m,:))==0);
85 matrizBG_sinNaN=matrizBG(:,pos_sinNaN);
86
W=griddata(matrizBG_sinNaN(3,:),matrizBG_sinNaN(2,:),matrizBG_sinNaN(1,:) ...
87 ,matrizBG_sinNaN(m,:),X,Y,Z);
88
89 %% Paso 2. Almacenamiento de las coordenadas que forman las líneas donde
90 %% los bordes de PBG se encuentran en lambda=1550 nm
91
92 %Se obtienen las coordenadas deseadas a partir de su búsqueda en el
93 %mapa de color 2D. Se añade la tercera dimensión y se almacenan dichas
94 %coordenadas en 'coordenadas_total'
95 coordenadas_total=[];
96 figure(u)
97 for n=1:size(W,3)
98 [coordenadas]=contour(X(:,:,1),Y(:,:,1),W(:,:,n),1550*[1 1]);
99 if isempty(coordenadas)==0
100 coordenadas=coordenadas(:,2:length(coordenadas));
101 a=Z(1,1,n)*ones(1,length(coordenadas));
102 coordenadas(3,:)=a;
103 coordenadas_total=[coordenadas_total coordenadas];
104 else
105 end
106 end
107
108 %% Paso 3. Interpolación de los valores de wi e we de 'coordenadas_total'
109 %% para el posterior mallado de a,wi e we y representación del
110 %% plano 3D
111
112
vectorA=linspace(min(coordenadas_total(1,:)),max(coordenadas_total(1:,:), ...
113 num_interv);
114
vectorB=linspace(min(coordenadas_total(2,:)),max(coordenadas_total(2:,:), ...
115 num_interv);
116 [A,B]=meshgrid(vectorA,vectorB);
117 C=griddata(coordenadas_total(1,:),coordenadas_total(2:,:), ...
118 coordenadas_total(3,:),A,B);
119 surf(A,B,C);shading flat
120 hold on;
121
122 %% Paso 4. Obtención y representación de las anomalías presentes en los
123 %% bordes de PBG
124
125 if m==4
126 %Se indican las coordenadas de de matrizBG donde se encuentra la
127 %información

```

```

128 switch estructura
129 case 'corr3D'
130 a=11;
131 case 'corr3Dinv'
132 a=7;
133 end
134
135 %Coordenadas donde el borde 1 de PBG presenta distorsión
136 borde1_dist(1,:)=matrizBG(1,find(matrizBG(a,:)));
137 borde1_dist(2,:)=matrizBG(2,find(matrizBG(a,:)));
138 borde1_dist(3,:)=matrizBG(3,find(matrizBG(a,:)));
139
140 p2=plot3(borde1_dist(3,:),borde1_dist(2,:),borde1_dist(1:),'g*');
141 legend(p2,'Distorsión');
142
143 %Almacenamiento en la variable de salida de su gráfica correspondiente
144 mapacolor3D_plano_borde1=figure(u);
145
146 elseif m==5
147 switch estructura
148 case 'corr3D'
149 a=12;
150 case 'corr3Dinv'
151 a=8;
152 borde2_bajo_cono(1,:)=matrizBG(1,find(matrizBG(6,:)));
153 borde2_bajo_cono(2,:)=matrizBG(2,find(matrizBG(6,:)));
154 borde2_bajo_cono(3,:)=matrizBG(3,find(matrizBG(6,:)));
155
156 p4=plot3(borde2_bajo_cono(3,:),borde2_bajo_cono(2,:),...
157 borde2_bajo_cono(1:),'ro');
158
159
160 end
161 %Coordenadas donde el borde 2 de PBG presenta distorsión
162 borde2_dist(1,:)=matrizBG(1,find(matrizBG(a,:)));
163 borde2_dist(2,:)=matrizBG(2,find(matrizBG(a,:)));
164 borde2_dist(3,:)=matrizBG(3,find(matrizBG(a,:)));
165
166 p3=plot3(borde2_dist(3,:),borde2_dist(2,:),borde2_dist(1:),'c*');
167 switch estructura
168 case 'corr3D'
169 legend(p3,'Distorsión')
170 case 'corr3Dinv'
171 legend([p4 p3],'Bajo cono de luz','Distorsión');
172 end
173
174 mapacolor3D_plano_borde2=figure(u);
175
176 elseif m==6
177 %Coordenadas donde el borde 3 de PBG se encuentra bajo el cono de
178 %luz
179 borde3_bajo_cono(1,:)=matrizBG(1,find(matrizBG(8,:)));
180 borde3_bajo_cono(2,:)=matrizBG(2,find(matrizBG(8,:)));
181 borde3_bajo_cono(3,:)=matrizBG(3,find(matrizBG(8,:)));
182
183 %Coordenadas donde el borde 3 de PBG presenta distorsión
184 borde3_dist(1,:)=matrizBG(1,find(matrizBG(13,:)));
185 borde3_dist(2,:)=matrizBG(2,find(matrizBG(13,:)));
186 borde3_dist(3,:)=matrizBG(3,find(matrizBG(13,:)));
187
188 p4=plot3(borde3_bajo_cono(3,:),borde3_bajo_cono(2,:),...
189 borde3_bajo_cono(1:),'ro');
190 p5=plot3(borde3_dist(3,:),borde3_dist(2,:),borde3_dist(1:),'y*');
191
192
193 legend([p4 p5],'Bajo cono de luz','Distorsión');
194

```

```

195 mapacolor3D_plano_borde3=figure(u);
196
197 elseif m==7
198 %Coordenadas donde el borde 4 de PBG se encuentra bajo el cono de
199 %luz
200 borde4_bajo_cono(1,:)=matrizBG(1,find(matrizBG(9,:)));
201 borde4_bajo_cono(2,:)=matrizBG(2,find(matrizBG(9,:)));
202 borde4_bajo_cono(3,:)=matrizBG(3,find(matrizBG(9,:)));
203
204 %Coordenadas donde el borde 4 de PBG no existe
205 borde4_sinBG(1,:)=matrizBG(1,find(matrizBG(10,:)));
206 borde4_sinBG(2,:)=matrizBG(2,find(matrizBG(10,:)));
207 borde4_sinBG(3,:)=matrizBG(3,find(matrizBG(10,:)));
208
209 %Coordenadas donde el borde 4 de PBG presenta distorsión
210 borde4_dist(1,:)=matrizBG(1,find(matrizBG(14,:)));
211 borde4_dist(2,:)=matrizBG(2,find(matrizBG(14,:)));
212 borde4_dist(3,:)=matrizBG(3,find(matrizBG(14,:)));
213
214 p6=plot3(borde4_dist(3,:),borde4_dist(2,:),borde4_dist(1:,:), 'r*');
215 p7=plot3(borde4_bajo_cono(3,:),borde4_bajo_cono(2,:), ...
216 borde4_bajo_cono(1,:), 'mo');
217 p8=plot3(borde4_sinBG(3,:),borde4_sinBG(2,:),borde4_sinBG(1,:), 'ks');
218
219 legend([p7 p8 p6], 'Bajo cono de luz', 'Borde de PBG no existe', ...
220 'Distorsión');
221
222 mapacolor3D_plano_borde4=figure(u);
223
224 end
225
226 %Propiedades de las gráficas
227 titulo=['Plano de configuraciones del borde ', num2str(u), ' de PBG en ' ...
228 '\lambda=1550 nm'];
229 title(titulo); xlabel('W_{e}'); ylabel('W_{i}'); zlabel('a');
230 c=colorbar;
231 c.Label.String = 'a';
232 switch estructura
233 case 'corr3D'
234 axis([700 1600 60 180 300 420])
235 case 'corr3Dinv'
236 axis([100 400 200 350 400 550])
237 end
238 set(gcf, 'position', get(groot, 'ScreenSize'));
239 u=u+1;
240 end
241
242 %Puesto que para ambas estructuras no existen algunas figuras estas se
243 %definen como 'NaN'
244 switch estructura
245 case 'corr3Dinv'
246 mapacolor3D_plano_borde3=NaN;
247 mapacolor3D_plano_borde4=NaN;
248 borde3_dist=NaN;
249 borde4_dist=NaN;
250 borde3_bajo_cono=NaN;
251 borde4_bajo_cono=NaN;
252 borde4_sinBG=NaN;
253 case 'corr3D'
254 borde2_bajo_cono=NaN;
255 end
256 end

```

### 1.4.3 *creator\_mapacolor\_3D\_vol\_dist.m*

```
1 function
2 [mapacolor3D_vol_dist1,mapacolor3D_vol_dist2,mapacolor3D_vol_dist3,...
3 mapacolor3D_vol_dist4,mapacolor3D_vol_bajoCono3,...
4 mapacolor3D_vol_bajoCono4,mapacolor3D_vol_sinBG4,u,...
5 mapacolor3D_vol_bajoCono2]=creator_mapacolor_3D_vol_dist(matrizBG,...
6 a_vector,wi_vector,we_vector,num_interv,borde1_dist,borde2_dist,...
7 borde3_dist,borde4_dist,borde3_bajo_cono,borde4_bajo_cono,borde4_sinBG,...
8 borde2_bajo_cono,u,estructura)
9 %
10 % %%%Representación gráfica de los volúmenes generados por los puntos
11 % %%%conflictivos de los diferentes bordes de PBG. Además se superpone con
12 % %%%la representación de los planos de los bordes a los que les
13 % corresponde
14 % %%%lambda=1550 nm y los puntos originales de las anomalías.
15 %
16 % %%%Entrada:
17 % matrizBG: matriz que contiene todos los datos originales de la simulación
18 % a_vector: Interpolación de los valores originales de 'a' utilizados en la
19 % simulación
20 % wi_vector: Interpolación de los valores originales de 'wi' utilizados en
21 % la simulación
22 % we_vector: Interpolación de los valores originales de 'we' utilizados en
23 % la simulación
24 %
25 % num_interv: indica el tamaño de equiespaciado de las interpolaciones de
26 % 'a_vector','wi_vector' y 'we_vector'
27 %
28 % borde1_dist: matriz que contiene los valores del borde superior del PBG1
29 % que presentan distorsión
30 %
31 % borde2_dist: matriz que contiene los valores del borde inferior del PBG1
32 % que presentan distorsión
33 % borde3_dist: matriz que contiene los valores del borde superior del PBG2
34 % que presentan distorsión
35 %
36 % borde4_dist: matriz que contiene los valores del borde inferior del PBG2
37 % que presentan distorsión
38 %
39 % borde3_bajo_cono: matriz que contiene los valores del borde superior del
40 % PBG2 que se encuentran bajo el cono de luz
41 %
42 % borde4_bajo_cono: matriz que contiene los valores del borde inferior del
43 % PBG2 que se encuentran bajo el cono de luz
44 %
45 % borde4_sinBG: matriz que contiene las coordenadas donde no existe el
46 % borde inferior del PBG2
47 %
48 % borde2_bajo_cono: matriz que contiene las coordenadas del borde inferior
49 % del PBG1 que se encuentran bajo el cono de luz
50 %
51 % u: numero de figura que se representa
52 % estructura: nombre de la estructura de simulación
53 %
54 %%%Salida:
55 % mapacolor3D_vol_dist1: Representación del volumen generado por los bordes
56 % superiores del PBG1 que presentan distorsión
57 %
58 % mapacolor3D_vol_dist2: Representación del volumen generado por los bordes
59 % inferiores del PBG1 que presentan distorsión
60 %
61 % mapacolor3D_vol_dist3: Representación del volumen generado por los bordes
```

```

62 % superiores del PBG2 que presentan distorsión
63 %
64 % mapacolor3D_vol_dist4: Representación del volumen generado por los bordes
65 % inferiores del PBG2 que presentan distorsión
66 %
67 % mapacolor3D_vol_bajoCono3: Representación del volumen generado por los
68 % bordes superiores del PBG2 localizados bajo el cono de luz
69 %
70 % mapacolor3D_vol_bajoCono4: Representación del volumen generado por los
71 % bordes inferiores del PBG2 localizados bajo el cono de luz
72 %
73 % mapacolor3D_vol_sinBG4: Representación del volumen generado por las
74 % coordenadas donde no existen los bordes inferiores del PBG2
75 % mapacolor3D_vol_bajoCono2: Representación del volumen generado por los
76 % bordes inferiores del PBG1 localizados bajo el cono de luz
77 %
78 % u: numero de figura que se representa
79 %
80 %
81 %% Paso 1. Mallado de las coordenadas we,wi,a en las matrices X,Y,Z para
82 %% sus correspondientes valores de lambda almacenado en W
83
84 % u=1;
85 q=1;
86 r=1;
87 [X,Y,Z]=meshgrid(we_vector,wi_vector,a_vector);
88
89 switch estructura
90 case 'corr3D'
91 f=[4 5 6 6 7 7 7];
92 case 'corr3Dinv'
93 f=[4 5 5];
94 end
95
96 for m=f
97 %Eliminación de los posibles 'NaN' existentes en los bordes de PBG
98 pos_sinNaN=find(isnan(matrizBG(m,:))==0);
99 matrizBG_sinNaN=matrizBG(:,pos_sinNaN);
100
W=griddata(matrizBG_sinNaN(3,:),matrizBG_sinNaN(2,:),matrizBG_sinNaN(1,:),...
101 matrizBG_sinNaN(m,:),X,Y,Z);
102
103 %% Paso 2. Almacenamiento de las coordenadas de las líneas donde los
bordes
104 %% de PBG se encuentran en lambda=1550 nm
105 coordenadas_total=[];
106 figure(u)
107 for n=1:size(W,3)
108 [coordenadas]=contour(X(:,:,1),Y(:,:,1),W(:,:,n),1550*[1 1]);
109 if isempty(coordenadas)==0
110 coordenadas=coordenadas(:,2:length(coordenadas));
111 a=Z(1,1,n)*ones(1,length(coordenadas));
112 coordenadas(3,:)=a;
113 coordenadas_total=[coordenadas_total coordenadas];
114 else
115 end
116 end
117
118 %% Paso 3. Interpolación de los valores de wi e we de 'coordenadas_total'
119 %% para el posterior mallado de a,wi e we y representación del
120 %% plano 3D
121
122
vectorA=linspace(min(coordenadas_total(1,:)),max(coordenadas_total(1:)),...
123 num_interv);
124
vectorB=linspace(min(coordenadas_total(2,:)),max(coordenadas_total(2:)),...

```

```

125 num_interv);
126 [A,B]=meshgrid(vectorA,vectorB);
127 C=griddata(coordenadas_total(1,:),coordenadas_total(2,:),...
128 coordenadas_total(3,:),A,B);
129 p1=surf(A,B,C);shading flat
130 hold on;
131 switch estructura
132 case 'corr3D'
133 axis([700 1600 60 180 300 420])
134 case 'corr3Dinv'
135 axis([100 400 200 350 400 550])
136 end
137
138 %% Paso 4. Representación de las coordenadas de las anomalías presentes en
139 %% los bordes de PBG y elección de las variables 'ejeZ','ejeY','ejeX'
140
141 switch m
142 case 4
143 p2=plot3(borde1_dist(3,:),borde1_dist(2,:),borde1_dist(1:),'g*');
144
145 %Los valores que tomen las variables que se presentan a
146 %continuación definirá el volumen que se va a crear en el paso 5
147 ejeZ=borde1_dist(1,:);
148 ejeY= borde1_dist(2,:);
149 ejeX=borde1_dist(3,:);
150 switch estructura
151 case 'corr3D'
152 fila=11;
153 case 'corr3Dinv'
154 fila=7;
155 end
156
157 case 5
158 switch estructura
159 case 'corr3D'
160 p3=plot3(borde2_dist(3,:),borde2_dist(2,:),...
161 borde2_dist(1:),'c*');
162
163 ejeZ=borde2_dist(1,:);
164 ejeY= borde2_dist(2,:);
165 ejeX=borde2_dist(3,:);
166 fila=12;
167
168 case 'corr3Dinv'
169 if q==1
170 p4=plot3(borde2_bajo_cono(3,:),borde2_bajo_cono(2,:),...
171 borde2_bajo_cono(1:),'ro');
172 ejeZ=borde2_bajo_cono(1,:);
173 ejeY= borde2_bajo_cono(2,:);
174 ejeX=borde2_bajo_cono(3,:);
175 fila=6;
176
177 elseif q==2
178 p5=plot3(borde2_dist(3,:),borde2_dist(2,:),...
179 borde2_dist(1:),'y*');
180 ejeZ=borde2_dist(1,:);
181 ejeY= borde2_dist(2,:);
182 ejeX=borde2_dist(3,:);
183 fila=8;
184 end
185 end
186
187 case 6
188
189 if q==1
190 p4=plot3(borde3_bajo_cono(3,:),borde3_bajo_cono(2,:),...
191 borde3_bajo_cono(1:),'ro');

```

```

192 ejeZ=borde3_bajo_cono(1,:);
193 ejeY= borde3_bajo_cono(2,:);
194 ejeX=borde3_bajo_cono(3,:);
195 fila=8;
196
197 elseif q==2
198 p5=plot3(borde3_dist(3,:),borde3_dist(2,:),borde3_dist(1,:),...
199 'y*');
200 ejeZ=borde3_dist(1,:);
201 ejeY= borde3_dist(2,:);
202 ejeX=borde3_dist(3,:);
203 fila=13;
204
205 end
206
207 case 7
208
209 if r==1
210 p6=plot3(borde4_bajo_cono(3,:),borde4_bajo_cono(2,:),...
211 borde4_bajo_cono(1,:), 'mo');
212 ejeZ=borde4_bajo_cono(1,:);
213 ejeY= borde4_bajo_cono(2,:);
214 ejeX=borde4_bajo_cono(3,:);
215 fila=9;
216
217
218 elseif r==2
219 p7=plot3(borde4_sinBG(3,:),borde4_sinBG(2,:),...
220 borde4_sinBG(1,:), 'ks');
221 ejeZ=borde4_sinBG(1,:);
222 ejeY=borde4_sinBG(2,:);
223 ejeX=borde4_sinBG(3,:);
224 fila=10;
225
226
227 elseif r==3
228 p8=plot3(borde4_dist(3,:),borde4_dist(2,:),borde4_dist(1,:),...
229 'r*');
230 ejeZ=borde4_dist(1,:);
231 ejeY=borde4_dist(2,:);
232 ejeX=borde4_dist(3,:);
233 fila=14;
234 end
235
236 end
237
238 %% Paso 5. Creación del volumen de anomalías mediante la utilización de
239 %% 'slices'
240
241 %Puesto que es posible que hayan bordes de PBG que no presenten
242 %anomalías es necesario utilizar 'try'. Los valores de 'ejeZ','ejeY','ejeX'
243 %irán cambiando en función del valor de 'm'
244 try
245 vector_a_prom=linspace(min(ejeZ),max(ejeZ),num_interv);
246 vector_wi_prom=linspace(min(ejeY),max(ejeY),num_interv);
247 vector_we_prom=linspace(min(ejeX),max(ejeX),num_interv);
248
249 borde_prom=matrizBG(m,find(matrizBG(fila,:)));
250
251 if m==7 && r==2
252 %Para el caso del volumen donde no existen bordes de PBG es
253 %necesario dar un valor (cualquiera) para poder representarlo
254 %(la cuarta dimensión que es el valor de lambda de los puntos
255 %anómalos en este caso no tendría ningún valor puesto que no
256 %existe).
257 borde_prom(:, :)=1;
258 else

```

```

259 end
260
261 [X_prom,Y_prom,Z_prom]=meshgrid(vector_we_prom,vector_wi_prom,...
262 vector_a_prom);
263 W_prom=griddata(ejeX,ejeY,ejeZ,borde_prom,X_prom,Y_prom,Z_prom);
264 p9=slice(X_prom,Y_prom,Z_prom,W_prom,[],[],vector_a_prom);
265 %figure;
vol3d('CData',W_prom,'XData',X_prom,'YData',Y_prom,'ZData',Z_prom);
266 shading flat
267 set(gcf,'position',get(groot,'Screensize'));
268
269 catch
270 end
271
272 %% Paso 6. Ajuste de propiedades de las gráficas y almacenamiento de estas
273 %% en las variables de salida de la función
274
275 %Cambio de color del plano de lambda=1550 nm a rojo
276 p1.FaceColor='r';
277 c=colorbar;
278 c.Label.String ='\lambda';
279
280 if m==4
281 titulo=['Volumen de configuraciones donde el borde 1 de PBG presenta'...
282 'distorsión'];
283 title(titulo); xlabel('W_{e}'); ylabel('W_{i}'); zlabel('a');
284 legend([p1 p2],'Plano \lambda=1550 nm','Puntos de distorsión '...
285 'originales');
286 set(gcf,'position',get(groot,'Screensize'));
287
288 mapacolor3D_vol_dist1=figure(u);
289
290 elseif m==5
291
292 switch estructura
293 case 'corr3D'
294 titulo=['Volumen de configuraciones donde el borde 2 de PBG '...
295 'presenta distorsión'];
296 title(titulo); xlabel('W_{e}'); ylabel('W_{i}'); zlabel('a');
297 legend([p1 p3],'Plano \lambda=1550 nm','Puntos de distorsión '...
298 'originales')
299 set(gcf,'position',get(groot,'Screensize'));
300
301 mapacolor3D_vol_dist2=figure(u);
302
303 case 'corr3Dinv'
304 if q==1
305 titulo=['Volumen de configuraciones donde el borde 2 de'...
306 'PBG está bajo el cono de luz'];
307 title(titulo); xlabel('W_{e}'); ylabel('W_{i}'); zlabel('a');
308 legend([p1 p4],'Plano \lambda=1550 nm','Puntos bajo '...
309 'cono de luz originales')
310 set(gcf,'position',get(groot,'Screensize'));
311
312 mapacolor3D_vol_bajoCono2=figure(u);
313
314 q=q+1;
315 elseif q==2
316 titulo=['Volumen de configuraciones donde el borde 2 de'...
317 'PBG presenta distorsión'];
318 title(titulo); xlabel('W_{e}'); ylabel('W_{i}'); zlabel('a');
319 legend([p1 p5],'Plano \lambda=1550 nm','Puntos de '...
320 'distorsión originales')
321 set(gcf,'position',get(groot,'Screensize'));
322
323 mapacolor3D_vol_dist2=figure(u);
324 %close (mapacolor3D_vol_dist3);

```

```

325 end
326
327 end
328
329 elseif m==6
330 if q==1
331 titulo=['Volumen de configuraciones donde el borde 3 de PBG '...
332 'está bajo el cono de luz'];
333 title(titulo); xlabel('W_{e}'); ylabel('W_{i}'); zlabel('a');
334 legend([p1 p4], 'Plano \lambda=1550 nm', 'Puntos bajo cono de '...
335 luz originales')
336
337 mapacolor3D_vol_bajoCono3=figure(u);
338
339 elseif q==2
340 titulo=['Volumen de configuraciones donde el borde 3 de PBG '...
341 'presenta distorsión'];
342 title(titulo); xlabel('W_{e}'); ylabel('W_{i}'); zlabel('a');
343 legend([p1 p5], 'Plano \lambda=1550 nm', 'Puntos de distorsión '...
344 originales')
345 set(gcf, 'position', get(groot, 'Screensize'));
346
347 mapacolor3D_vol_dist3=figure(u);
348 % close (mapacolor3D_vol_dist3);
349 end
350 q=q+1;
351
352 elseif m==7
353 if r==1
354 titulo=['Volumen de configuraciones donde el borde 4 de PBG '...
355 'está bajo el cono de luz'];
356 title(titulo); xlabel('W_{e}'); ylabel('W_{i}'); zlabel('a');
357 legend([p1 p6], 'Plano \lambda=1550 nm', 'Puntos bajo cono de luz
originales')
358
359 % mapacolor3D_vol_bajoCono4=figure(5);
360 mapacolor3D_vol_bajoCono4=figure(u);
361
362 elseif r==2
363 titulo=['Volumen de configuraciones donde el borde 4 de PBG no '...
364 existe'];
365 title(titulo); xlabel('W_{e}'); ylabel('W_{i}'); zlabel('a');
366 legend([p1 p7], 'Plano \lambda=1550 nm', 'Puntos sin borde de PBG
originales')
367
368 % mapacolor3D_vol_sinBG4=figure(6);
369 mapacolor3D_vol_sinBG4=figure(u);
370
371 else
372 titulo=['Volumen de configuraciones donde el borde 4 de PBG'...
373 'presenta distorsión'];
374 title(titulo); xlabel('W_{e}'); ylabel('W_{i}'); zlabel('a');
375 legend([p1 p8], 'Plano \lambda=1550 nm', 'Puntos de distorsión originales')
376 set(gcf, 'position', get(groot, 'Screensize'));
377
378 % mapacolor3D_vol_dist4=figure(7);
379 mapacolor3D_vol_dist4=figure(u);
380 % close (mapacolor3D_vol_dist4);
381 end
382 r=r+1;
383 end
384 u=u+1;
385 end
386
387 switch estructura
388
389 case 'corr3Dinv'

```

```

390 mapacolor3D_vol_dist3=NaN;
391 mapacolor3D_vol_dist4=NaN;
392 mapacolor3D_vol_bajoCono3=NaN;
393 mapacolor3D_vol_bajoCono4=NaN;
394 mapacolor3D_vol_sinBG4=NaN;
395
396 case 'corr3D'
397 mapacolor3D_vol_bajoCono2=NaN;
398 end
399
400 end

```

## 1.5 Obtención de sensibilidad y tolerancia

### 1.5.1 *creator\_vol\_sensibilidad.m*

```

1 function [vol4D_sensibilidad_borde1,vol4D_sensibilidad_borde2,...
2 vol4D_sensibilidad_borde3,vol4D_sensibilidad_borde4,u]=...
3 creator_vol_sensibilidad(matrizBG_SiO2,matrizBG_H2O,u,estructura)
4 %
5 % %%Cálculo de la sensibilidad del sensor cuando varía el medio que hay
6 % %%encima de este. Se compara el sensor con H2O por encima
7 % %% (cte dieléctrica=1,333) con el sensor con SiO2 por encima
8 % %% (cte dieléctrica=1,444). La representación se realiza mediante
9 % %% múltiples 'slice' con mapas de color propios.
10 %
11 %%Entrada:
12 % matrizBG_SiO2: matriz que contiene todos los datos originales de la
13 % simulación con SiO2 por encima
14 %
15 % matrizBG_H2O: matriz que contiene todos los datos originales de la
16 % simulación con H2O por encima
17 %
18 % u: número de figura que se representa
19 % estructura: nombre de la estructura de simulación
20 %
21 %%Salida:
22 % vol4D_sensibilidad_borde1: representación del volumen de sensibilidad
23 % para el borde superior de PBG1
24 %
25 % vol4D_sensibilidad_borde2: representación del volumen de sensibilidad
26 % para el borde inferior de PBG1
27 %
28 % vol4D_sensibilidad_borde3: representación del volumen de sensibilidad
29 % para el borde superior de PBG2
30 %
31 % vol4D_sensibilidad_borde4: representación del volumen de sensibilidad
32 % para el borde inferior de PBG2
33
34 % u: número de figura que se representa
35 %
36 %
37 %
38 %% Paso 1. Definición de las constantes dieléctricas utilizadas y mallado
39 %% de las coordenadas we,wi,a en las matrices X,Y,Z para ambas
40 %% simulaciones (SiO2 y H2O).
41
42 constante_dielectrica_SiO2=1.444;
43 constante_dielectrica_H2O=1.333;
44 s=1;
45
46 %Las matrices X,Y,Z coinciden para ambas simulaciones por lo que su cálculo

```

```

47 %únicamente se realizará una vez utilizándose los mismos para ambos
sensores
48 a_vector=linspace(min(matrizBG_SiO2(1,:)),max(matrizBG_SiO2(1,:)),50);
49 wi_vector=linspace(min(matrizBG_SiO2(2,:)),max(matrizBG_SiO2(2,:)),50);
50 we_vector=linspace(min(matrizBG_SiO2(3,:)),max(matrizBG_SiO2(3,:)),50);
51 [X,Y,Z]=meshgrid(we_vector,wi_vector,a_vector);
52
53 %% Paso 2. Cálculo de la sensibilidad para los diferentes bordes de PBG
54
55 switch estructura
56 case 'corr3D'
57 f=4:7;
58 case 'corr3Dinv'
59 f=4:5;
60 end
61
62 for m=f
63 %Se realiza la resta de los valores correspondientes a lambda en la
64 %simulación de SiO2 menos la de H2O y se divide por la resta entre las
65 %constantes dieléctricas de dichos materiales
66 sensibilidad_original=(matrizBG_SiO2(m,:)-matrizBG_H2O(m,:))/...
67 (constante_dielectrica_SiO2-constante_dielectrica_H2O);
68
69 %Se localizan las posiciones donde la fórmula anterior es NaN y se
70 %crea una matriz sin estos valores. Se realiza el mallado de la
71 %sensibilidad
72 posiciones=find(isnan(sensibilidad_original)==0);
73 sensibilidad=[matrizBG_H2O(1:3,posiciones);...
74 sensibilidad_original(posiciones)];
75 sensibilidad_grid=griddata(sensibilidad(3,:),sensibilidad(2,:),...
76 sensibilidad(1,:),sensibilidad(4,:),X,Y,Z);
77
78 %% Paso 3. Representación gráfica de la sensibilidad de las diferentes
79 %% configuraciones para cada borde de PBG mediante múltiples 'slice'
80 %% y ajuste de las propiedades de dichas gráficas
81
82 figure(u)
83 for n=1:length(a_vector)
84 slice(X,Y,Z,sensibilidad_grid,[],[],a_vector(n))
85 shading flat;
86 hold on
87 c=colorbar;
88 c.Label.String = 'nm/RIU';
89 titulo=['Sensibilidad del borde ',num2str(s),' de PBG. Medio sobre'...
90 'el sensor: H2O vs SiO2'];
91 title(titulo); xlabel('W_{e}'); ylabel('W_{i}'); zlabel('a');
92 set(gcf,'position',get(groot,'ScreenSize'));
93
94 switch estructura
95 case 'corr3D'
96 axis([700 1600 60 180 300 420])
97 case 'corr3Dinv'
98 axis([100 400 200 350 400 550])
99 end
100
101 %Se utiliza la función 'pause' para poder visualizar mejor el
102 %cambio de sensibilidad en cada 'slice' en función de las
103 %configuraciones existentes
104 pause()
105 end
106
107 %% Paso 4. Almacenamiento en las variables de salida de sus gráficas
108 %% correspondientes
109
110 if m==4
111 vol4D_sensibilidad_borde1=figure(u);
112 elseif m==5

```

```

113 vol4D_sensibilidad_borde2=figure(u);
114 elseif m==6
115 vol4D_sensibilidad_borde3=figure(u);
116 else
117 vol4D_sensibilidad_borde4=figure(u);
118 end
119
120 u=u+1;
121 s=s+1;
122
123 end
124
125 switch estructura
126 case 'corr3Dinv'
127 vol4D_sensibilidad_borde3=NaN;
128 vol4D_sensibilidad_borde4=NaN;
129 end
130
131 end

```

### 1.5.2 *creator\_vol\_tolerancia.m*

```

1 function [vol4D_tolerancia_borde1,vol4D_tolerancia_borde2,...
2 vol4D_tolerancia_borde3,vol4D_tolerancia_borde4,X_int_borde1,...
3 Y_int_borde1,Z_int_borde1,W_int_borde1,tolerancia_borde1,X_int_borde2,...
4 Y_int_borde2,Z_int_borde2,W_int_borde2,tolerancia_borde2,X_int_borde3,...
5 Y_int_borde3,Z_int_borde3, W_int_borde3,tolerancia_borde3,X_int_borde4,...
6 Y_int_borde4,Z_int_borde4,W_int_borde4,tolerancia_borde4,u]=...
7 creator_vol_tolerancia(matrizBG,num_interv,u,estructura)
8 %
9 % %%Cálculo de tolerancias.
10 % %%Para poder obtener las variaciones que sufre lambda en función de
11 % %%a,we y wi es necesario el cálculo del producto de las derivadas
12 % %%parciales de primer orden de dichos valores de lambda respecto
13 % %%a a,we y wi. Cada una de estas derivadas se realiza en la dirección
14 % %%de derivación correspondiente. Es decir, la derivada respecto a a se
15 % %%realiza respecto al eje Z, wi respecto al eje Y e we respecto al eje
16 % %%Dichos valores se representan mediante un "volumen" de splines para
17 % %%cada borde de PBG.
18 %
19 % %%NOTA: Se realiza una interpolación cúbica de los valores obtenidos de
20 % %%'matrizBG' para mejorar la calidad de la representación del volumen
21 %
22 %%Entrada:
23 % matrizBG: matriz que contiene todos los datos originales de la simulación
24 % num_interv: indica el tamaño de equiespaciado de las interpolaciones de
25 % 'a_vector_int','wi_vector_int' y 'we_vector_int'
26
27 % u: numero de figura que se representa
28 % estructura: nombre de la estructura de simulación
29 %
30 %%Salida:
31 % vol4D_tolerancia_borde1: representación del volumen de tolerancias para
32 % el borde superior de PBG1
33
34 % vol4D_tolerancia_borde2: representación del volumen de tolerancias para
35 % el borde inferior de PBG1
36
37 % vol4D_tolerancia_borde3: representación del volumen de tolerancias para
38 %el borde superior de PBG2
39

```

```

40 % vol4D_tolerancia_borde4: representación del volumen de tolerancias para
41 % el borde inferior de PBG2
42 %
43 % X_int_borde1: Matriz de mallado de las coordenadas de 'we' para aquellas
44 % configuraciones donde exista el borde superior del PBG1
45 %
46 % Y_int_borde1: Matriz de mallado de las coordenadas de 'wi' para aquellas
47 % configuraciones donde exista el borde superior del PBG1
48 %
49 % Z_int_borde1: Matriz de mallado de las coordenadas de 'a' para aquellas
50 % configuraciones donde exista el borde superior del PBG1
51 %
52 % W_int_borde1: Interpolación cúbica de los valores de lambda del borde
53 % superior del PBG1
54 %
55 % tolerancia_borde1: Tolerancia que presentan los valores de lambda del
56 % borde superior del PBG1
57 %
58 % X_int_borde2: Matriz de mallado de las coordenadas de 'we' para aquellas
59 % configuraciones donde exista el borde inferior del PBG1
60 %
61 % Y_int_borde2: Matriz de mallado de las coordenadas de 'wi' para aquellas
62 % configuraciones donde exista el borde inferior del PBG1
63 %
64 % Z_int_borde2: Matriz de mallado de las coordenadas de 'a' para aquellas
65 % configuraciones donde exista el borde inferior del PBG1
66 %
67 % W_int_borde2: Interpolación cúbica de los valores de lambda del borde
68 % inferior del PBG1
69 % tolerancia_borde2: Tolerancia que presentan los valores de lambda del
70 % borde inferior del PBG1
71 %
72 % X_int_borde3: Matriz de mallado de las coordenadas de 'we' para aquellas
73 % configuraciones donde exista el borde superior del PBG2
74 %
75 % Y_int_borde3: Matriz de mallado de las coordenadas de 'wi' para aquellas
76 % configuraciones donde exista el borde superior del PBG2
77 % Z_int_borde3: Matriz de mallado de las coordenadas de 'a' para aquellas
78 % configuraciones donde exista el borde superior del PBG2
79 % W_int_borde3: Interpolación cúbica de los valores de lambda del borde
80 % superior del PBG2
81 % tolerancia_borde3: Tolerancia que presentan los valores de lambda del
82 % borde superior del PBG2
83 %
84 % X_int_borde4: Matriz de mallado de las coordenadas de 'we' para aquellas
85 % configuraciones donde exista el borde inferior del PBG2
86 %
87 % Y_int_borde4: Matriz de mallado de las coordenadas de 'wi' para aquellas
88 % configuraciones donde exista el borde inferior del PBG2
89 %
90 % Z_int_borde4: Matriz de mallado de las coordenadas de 'a' para aquellas
91 % configuraciones donde exista el borde superior del PBG2
92 %
93 % W_int_borde4: Interpolación cúbica de los valores de lambda del borde
94 % inferior del PBG2
95 %
96 % tolerancia_borde4: Tolerancia que presentan los valores de lambda del
97 % borde inferior del PBG2
98 %
99 % u: numero de figura que se representa
100 %
101 %
102 %
103 %% Paso 1. Mallado de las coordenadas we,wi,a en las matrices X,Y,Z para
104 %% sus correspondientes valores de lambda almacenado en W
105 %% (interpolación lineal) y mallado de las coordenadas we_int,

```

```

106 %% wi_int,a_int en las matrices X_int,Y_int,Z_int para sus
107 %% correspondientes valores de lambda almacenado en W_int
108 %% (interpolación cúbica)
109
110 %Para poder realizar la interpolación cúbica primero es necesario realizar
111 %una interpolación lineal. Puesto que no conviene realizar la
interpolación
112 %lineal esta únicamente extrae los valores originales de 'matrizBG'
113 a_vector=linspace(min(matrizBG(1,:)),max(matrizBG(1,:)),4);
114 wi_vector=linspace(min(matrizBG(2,:)),max(matrizBG(2,:)),4);
115 we_vector=linspace(min(matrizBG(3,:)),max(matrizBG(3,:)),4);
116
117 s=1;
118
119 switch estructura
120 case 'corr3D'
121 f=4:7;
122 case 'corr3Dinv'
123 f=4:5;
124 end
125
126 for m=f
127
128 %Para el borde de band gap 4 es necesario realizar un tratamiento
129 %especial que elimine los valores NaN de borde4 en 'matrizBG'
130 if m==7
131 posiciones=find(isnan(matrizBG(7,:))~=0);
132 coordenadas_borde4_BG=[matrizBG(1:3,posiciones);matrizBG(7,posiciones)];
133
134 %Puesto que se desconoce si todas las configuraciones aparecerán
135 %para el borde 4 de PBG es necesario realizar un tratamiento
136 %especial en el que se tenga en cuenta únicamente aquellos valores
137 %que aparecen. Se indica para cada valor del vector 'a_vector,
138 %we_vector,wi_vector' las repeticiones que aparecen en las filas
139 %1,2 y 3 de coordenadas_borde4_BG respectivamente y se indica las
140 %posiciones de aquellas que se repiten
141 a_repetidos=find(histc(coordenadas_borde4_BG(1,:),a_vector)~=0);
142 wi_repetidos=find(histc(coordenadas_borde4_BG(2,:),wi_vector)~=0);
143 we_repetidos=find(histc(coordenadas_borde4_BG(3,:),we_vector)~=0);
144
145
146 a_vector=linspace(min(coordenadas_borde4_BG(1,:)),max(coordenadas_borde4_BG(1,
:)),length(a_repetidos));
147
148 wi_vector=linspace(min(coordenadas_borde4_BG(2,:)),max(coordenadas_borde4_BG(2
,)),length(wi_repetidos));
149
150 we_vector=linspace(min(coordenadas_borde4_BG(3,:)),max(coordenadas_borde4_BG(3
,)),length(we_repetidos));
151 [X,Y,Z]=meshgrid(we_vector,wi_vector,a_vector);
152
153 %Interpolación lineal
154
155 W=griddata(coordenadas_borde4_BG(3,:),coordenadas_borde4_BG(2,:),coordenadas_b
orde4_BG(1,:),coordenadas_borde4_BG(4,:),X,Y,152
156
157 a_vector_int=linspace(min(coordenadas_borde4_BG(1,:)),max(coordenadas_borde4_B
G(1,)),num_interv);
158
159 wi_vector_int=linspace(min(coordenadas_borde4_BG(2,:)),max(coordenadas_borde4_
BG(2,)),num_interv);
160
161 we_vector_int=linspace(min(coordenadas_borde4_BG(3,:)),max(coordenadas_borde4_
BG(3,)),num_interv);
162
163 else

```

```

158 [X,Y,Z]=meshgrid(we_vector,wi_vector,a_vector);
159 W=griddata(matrizBG(3,:),matrizBG(2,:),matrizBG(1,:),matrizBG(m,:),X,Y,Z);
160
161 a_vector_int=linspace(min(matrizBG(1,:)),max(matrizBG(1:)),num_interv);
162 wi_vector_int=linspace(min(matrizBG(2,:)),max(matrizBG(2:)),num_interv);
163 we_vector_int=linspace(min(matrizBG(3,:)),max(matrizBG(3:)),num_interv);
164 end
165
166 %Interpolación cúbica
167 [X_int,Y_int,Z_int]=meshgrid(we_vector_int,wi_vector_int,a_vector_int);
168 W_int=interp3(X,Y,Z,W,X_int,Y_int,Z_int,'cubic');
169
170 %% Paso 2. Cálculo de derivadas
171
172 %Se calcula el paso 'h' de las derivadas. Puesto que este va a ser siempre
173 %constante basta con restar, respecto al eje indicado, un valor cualquiera
174 %de la matriz respecto a su valor anterior.
175 hX_int=X_int(1,2,1)-X_int(1,1,1);
176 hY_int=Y_int(2,1,1)-Y_int(1,1,1);
177 hZ_int=Z_int(1,1,2)-Z_int(1,1,1);
178
179 %%Derivadas primeras parciales de W respecto a los tres ejes de
coordenadas
180 %a partir del gradiente.
181
[X_derivada1_int,Y_derivada1_int,Z_derivada1_int]=gradient(W_int,hX_int,hY_int
,hZ_int);
182
183 %% Paso 3. Cálculo de la tolerancia a través de la obtención de la
184 %variabilidad de los bordes de PBG
185
186 %Se obtiene mediante el producto de las derivadas primeras respecto al
187 %eje X, Y e Z
188 tolerancia=abs(X_derivada1_int.*Y_derivada1_int.*Z_derivada1_int);
189
190 %% Paso 4. Representación de la tolerancia de las diferentes
191 %% configuraciones para cada borde de PBG
192
193 %Creación del volumen mediante múltiples 'slice'. Ajuste de rangos de
194 %representación de ejes de coordenadas, rangos de colorbar y adición de
195 %leyenda y título
196 figure(u)
197 for n=1:length(a_vector_int)
198
199 slice(X_int,Y_int,Z_int,tolerancia,[],[],a_vector_int(n))
200 hold on, shading flat
201 c=colorbar;
202 c.Label.String = 'Variabilidad';
203 titulo=['Tolerancia para el borde ',num2str(s),' de PBG'];
204 title(titulo); xlabel('W_{e}'); ylabel('W_{i}'); zlabel('a');
205 set(gcf,'position',get(groot,'Screensize'));
206
207 switch estructura
208 case 'corr3D'
209 axis([700 1600 60 180 300 420])
210 case 'corr3Dinv'
211 axis([100 400 200 350 400 550])
212 end
213
214 pause(.5)
215
216 end
217
218 %% Paso 5. Almacenamiento en las variables de salida de sus gráficas
219 %% correspondientes
220

```

```

221 if m==4
222 vol4D_tolerancia_borde1=figure(u);
223 X_int_borde1=X_int;
224 Y_int_borde1=Y_int;
225 Z_int_borde1=Z_int;
226 W_int_borde1=W_int;
227 tolerancia_borde1=tolerancia;
228 elseif m==5
229 vol4D_tolerancia_borde2=figure(u);
230 X_int_borde2=X_int;
231 Y_int_borde2=Y_int;
232 Z_int_borde2=Z_int;
233 W_int_borde2=W_int;
234 tolerancia_borde2=tolerancia;
235 elseif m==6
236 vol4D_tolerancia_borde3=figure(u);
237 X_int_borde3=X_int;
238 Y_int_borde3=Y_int;
239 Z_int_borde3=Z_int;
240 W_int_borde3=W_int;
241 tolerancia_borde3=tolerancia;
242 else
243 vol4D_tolerancia_borde4=figure(u);
244 X_int_borde4=X_int;
245 Y_int_borde4=Y_int;
246 Z_int_borde4=Z_int;
247 W_int_borde4=W_int;
248 tolerancia_borde4=tolerancia;
249 end
250
251 u=u+1;
252 s=s+1;
253
254 switch estructura
255 case 'corr3Dinv'
256 vol4D_tolerancia_borde3=NaN;
257 X_int_borde3=NaN;
258 Y_int_borde3=NaN;
259 Z_int_borde3=NaN;
260 W_int_borde3=NaN;
261 tolerancia_borde3=NaN;
262 vol4D_tolerancia_borde4=NaN;
263 X_int_borde4=NaN;
264 Y_int_borde4=NaN;
265 Z_int_borde4=NaN;
266 W_int_borde4=NaN;
267 tolerancia_borde4=NaN;
268 end
269
270 end
271 end

```

## 1.6 Representación sensibilidad para PBG en $\lambda=1550$ nm

### 1.6.1 *creator\_mapacolor\_4D\_sensibilidad.m*

```
1 function [mapacolor4D_plano1_sens,mapacolor4D_plano2_sens,...
2 mapacolor4D_plano3_sens,mapacolor4D_plano4_sens,u]=...
3 creator_mapacolor_4D_sensibilidad(matrizBG,matrizBG_SiO2,borde1_dist,...
4 borde2_dist,borde3_dist,borde4_dist,borde3_bajo_cono,borde4_bajo_cono,...
5 borde4_sinBG,borde2_bajo_cono,u,estructura)
6 %
7 % %%Representación gráfica de los planos que forman los diferentes bordes
8 % %%de PBG cuando estos valen lambda=1550 nm. La cuarta dimensión añadida
9 % %%es el valor de la sensibilidad (en nm/RIU) para dichos bordes. Además,
10 % %%se añaden los puntos que presentan anomalías.
11 %
12 %%%Entrada:
13 % matrizBG: matriz que contiene todos los datos originales de la simulación
14 % con H2O por encima del sensor
15 %
16 % matrizBG_SiO2: matriz que contiene todos los datos originales de la
17 % simulación con SiO2 por encima del sensor
18 %
19 % borde1_dist: matriz que contiene los valores del borde superior del PBG1
20 % que presentan distorsión
21 %
22 % borde2_dist: matriz que contiene los valores del borde inferior del PBG1
23 % que presentan distorsión
24 %
25 % borde3_dist: matriz que contiene los valores del borde superior del PBG2
26 % que presentan distorsión
27 %
28 % borde4_dist: matriz que contiene los valores del borde inferior del PBG2
29 % que presentan distorsión
30 %
31 % borde3_bajo_cono: matriz que contiene los valores del borde superior del
32 %PBG2 que se encuentran bajo el cono de luz
33 %
34 % borde4_bajo_cono: matriz que contiene los valores del borde inferior del
35 % PBG2 que se encuentran bajo el cono de luz
36 %
37 % borde4_sinBG: matriz que contiene las coordenadas donde no existe el
38 %borde inferior del PBG2
39 %
40 % u: numero de figura que se representa
41 % estructura: nombre de la estructura de simulación
42 %
43 %%%Salida:
44 % mapacolor4D_plano1_sens: Representación de la sensibilidad de los valores
45 % del plano donde lambda=1550 nm para los bordes superiores del PBG1
46 %
47 % mapacolor4D_plano2_sens: Representación de la sensibilidad de los valores
48 % del plano donde lambda=1550 nm para los bordes inferiores del PBG1
49 %
50 % mapacolor4D_plano3_sens: Representación de la sensibilidad de los valores
51 % del plano donde lambda=1550 nm para los bordes superiores del PBG2
52 %
53 % mapacolor4D_plano4_sens: Representación de la sensibilidad de los valores
54 % del plano donde lambda=1550 nm para los bordes inferiores del PBG2
55 %
56 %
57 %
58 %% Paso 1. Mallado de las coordenadas we,wi,a en las matrices X,Y,Z para
59 %% sus correspondientes valores de lambda almacenado en W
60
```

```

61 %En este caso sí es necesario recalcular los valores de 'a_vector',
62 %'wi_vector' y 'we_vector' puesto que el 'num_interv' es mayor
63 %(se desea mayor precisión)
64 num_interv=100;
65
66 a_vector=linspace(min(matrizBG(1,:)),max(matrizBG(1:)),num_interv);
67 wi_vector=linspace(min(matrizBG(2,:)),max(matrizBG(2:)),num_interv);
68 we_vector=linspace(min(matrizBG(3,:)),max(matrizBG(3:)),num_interv);
69 s=1;
70 [X,Y,Z]=meshgrid(we_vector,wi_vector,a_vector);
71
72 switch estructura
73 case 'corr3D'
74 f=4:7;
75 case 'corr3Dinv'
76 f=4:5;
77 end
78
79 for m=f
80
81 %Eliminación de los posibles 'NaN' existentes en los bordes de PBG
82 pos_sinNaN=find(isnan(matrizBG(m,:))==0);
83 matrizBG_sinNaN=matrizBG(:,pos_sinNaN);
84 W=griddata(matrizBG_sinNaN(3,:),matrizBG_sinNaN(2:),...
85 matrizBG_sinNaN(1,:),matrizBG_sinNaN(m,:),X,Y,Z);
86
87 %% Paso 2. Almacenamiento de las coordenadas de las líneas donde los bordes
88 %% de PBG se encuentran en lambda=1550 nm
89
90 coordenadas_total=[];
91 figure
92 for n=1:size(W,3)
93 [coordenadas]=contour(X(:,:,1),Y(:,:,1),W(:,:,n),1550*[1 1]);
94 if isempty(coordenadas)==0
95 coordenadas=coordenadas(:,2:length(coordenadas));
96 a=Z(1,1,n)*ones(1,length(coordenadas));
97 coordenadas(3,:)=a;
98 coordenadas_total=[coordenadas_total coordenadas];
99 else
100 end
101 end
102 %% Paso 3. Interpolación de los valores de wi e we de 'coordenadas_total'
103 %% para el posterior mallado de a,wi e we que forma el plano donde
104 %% lambda=1550 nm
105
106
107 vectorA=linspace(min(coordenadas_total(1,:)),max(coordenadas_total(1:)),...
108 num_interv);
109
110 vectorB=linspace(min(coordenadas_total(2,:)),max(coordenadas_total(2:)),...
111 num_interv);
112 [A,B]=meshgrid(vectorA,vectorB);
113 C=griddata(coordenadas_total(1,:),coordenadas_total(2:),...
114 coordenadas_total(3,:),A,B);
115
116 %% Paso 4. Cálculo de la sensibilidad para el plano de lambda=1550 nm
117
118 %Antes de calcular la sensibilidad es necesario descartar los valores
119 %de los bordes de 'matrizBG_siO2' cuyo valor es NaN. Esto se realiza
120 %para evitar problemas de representación después
121 matriz_siO2_coincidente=matrizBG_siO2(:,pos_sinNaN);
122 pos_matrizBG_siO2_sinNaN=find(isnan(matriz_siO2_coincidente(m,:))==0);
123 matrizBG_siO2_sinNaN=matrizBG_siO2_coincidente(:,pos_matrizBG_siO2_sinNaN);
124 D=griddata(matrizBG_siO2_sinNaN(3,:),matrizBG_siO2_sinNaN(2:),...
125 matrizBG_siO2_sinNaN(1,:),matrizBG_siO2_sinNaN(m,:),A,B,C);

```

```

126 %Cálculo de la sensibilidad. Se utiliza la siguiente fórmula
127  $(\lambda_{SiO_2} - \lambda_{H_2O}) / (cte_{DielectricaSiO_2} - cte_{DielectricaH_2O})$ 
128 sensibilidad=(D-1550*ones(num_interv,num_interv))/(1.444-1.333);
129
130 %% Paso 5. Representación del plano añadiendo la sensibilidad como
parámetro
131 p1=surf(A,B,C,sensibilidad);
132 shading flat; hold on;
133
134 %% Paso 6. Representación de los bordes de PBG que presentan anomalías y
135 %% almacenamiento de las gráficas en las variables de salida
136 %% correspondientes
137 if m==4
138 p2=plot3(borde1_dist(3,:),borde1_dist(2,:),borde1_dist(1:),'g*');
139 legend([p1 p2], 'Plano \lambda=1550 nm',...
140 'Puntos de distorsión originales');
141 mapacolor4D_plano1_sens=figure(u);
142
143 elseif m==5
144 switch estructura
145 case 'corr3D'
146 p3=plot3(borde2_dist(3,:),borde2_dist(2,:),borde2_dist(1:),'c*');
147 legend([p1 p3], 'Plano \lambda=1550 nm',...
148 'Puntos de distorsión originales');
149 mapacolor4D_plano2_sens=figure(u);
150
151 case 'corr3Dinv'
152 p4=plot3(borde2_bajo_cono(3,:),borde2_bajo_cono(2,:),...
153 borde2_bajo_cono(1:),'ro');
154 p3=plot3(borde2_dist(3,:),borde2_dist(2,:),borde2_dist(1:),'c*');
155 legend([p1 p3 p4], 'Plano \lambda=1550 nm',...
156 'Puntos de distorsión originales',...
157 'Puntos bajo cono de luz originales');
158 mapacolor4D_plano2_sens=figure(u);
159 end
160
161 elseif m==6
162 p4=plot3(borde3_bajo_cono(3,:),borde3_bajo_cono(2,:),...
163 borde3_bajo_cono(1:),'ro');
164 p5=plot3(borde3_dist(3,:),borde3_dist(2,:),borde3_dist(1:),'y*');
165 legend([p1 p4 p5], 'Plano \lambda=1550 nm',...
166 'Puntos bajo cono de luz originales',...
167 'Puntos de distorsión originales');
168 mapacolor4D_plano3_sens=figure(u);
169
170 elseif m==7
171 p6=plot3(borde4_bajo_cono(3,:),borde4_bajo_cono(2,:),...
172 borde4_bajo_cono(1:),'mo');
173 p7=plot3(borde4_sinBG(3,:),borde4_sinBG(2,:),borde4_sinBG(1:),'ks');
174 p8=plot3(borde4_dist(3,:),borde4_dist(2,:),borde4_dist(1:),'r*');
175 legend([p1 p6 p7 p8], 'Plano \lambda=1550 nm',...
176 'Puntos bajo cono de luz originales',...
177 'Puntos sin borde de PBG originales',...
178 'Puntos de distorsión originales');
179 mapacolor4D_plano4_sens=figure(u);
180
181 else
182 end
183
184 %% Paso 7. Ajuste de propiedades de las gráficas
185
186 titulo=['Sensibilidad del plano donde \lambda=1550 nm para el borde ',...
187 num2str(s), ' de PBG'];
188 title(titulo); xlabel('W_{e}'); ylabel('W_{i}'); zlabel('a');
189 c=colorbar;
190 c.Label.String ='Sensibilidad (nm/RIU)';
191

```

```

192 switch estructura
193 case 'corr3D'
194 axis([700 1600 60 180 300 420])
195 case 'corr3Dinv'
196 axis([100 400 200 350 400 550])
197 end
198
199 set(gcf, 'position', get(groot, 'Screensize'));
200 u=u+1;
201 s=s+1;
202
203 end
204
205 switch estructura
206 case 'corr3Dinv'
207 mapacolor4D_plano3_sens=NaN;
208 mapacolor4D_plano4_sens=NaN;
209 end
210
211 end

```

## 1.7 Representación tolerancia para PBG en $\lambda=1550$ nm

### 1.7.1 *creator\_mapacolor\_4D\_tolerancia.m*

```

1 function [mapacolor4D_plano1_tol,mapacolor4D_plano2_tol,...
2 mapacolor4D_plano3_tol,mapacolor4D_plano4_tol,u]=...
3 creator_mapacolor_4D_tolerancia(X_int_borde1,Y_int_borde1,...
4 Z_int_borde1,W_int_borde1,tolerancia_borde1,X_int_borde2,Y_int_borde2,...
5 Z_int_borde2,W_int_borde2,tolerancia_borde2,X_int_borde3,Y_int_borde3,...
6 Z_int_borde3,W_int_borde3,tolerancia_borde3,X_int_borde4,Y_int_borde4,...
7 Z_int_borde4,W_int_borde4,tolerancia_borde4,num_interv,u,estructura)
8 %
9 % %%Representación gráfica (en escala de color) de la tolerancia que
10 % %%presentan los planos donde lambda=1550 nm para los diferentes bordes
11 % %%de PBG. Además, se añaden los puntos que presentan anomalías.
12 %
13 %%Entrada:
14 % X_int_borde1: Matriz de mallado de las coordenadas de 'we' para aquellas
15 % configuraciones donde exista el borde superior del PBG1
16 %
17 % Y_int_borde1: Matriz de mallado de las coordenadas de 'wi' para aquellas
18 % configuraciones donde exista el borde superior del PBG1
19 %
20 % Z_int_borde1: Matriz de mallado de las coordenadas de 'a' para aquellas
21 % configuraciones donde exista el borde superior del PBG1
22 %
23 % W_int_borde1: Interpolación cúbica de los valores de lambda del borde
24 % superior del PBG1
25 %
26 % tolerancia_borde1: Tolerancia que presentan los valores de lambda del
27 % borde superior del PBG1
28 %
29 % X_int_borde2: Matriz de mallado de las coordenadas de 'we' para aquellas
30 % configuraciones donde exista el borde inferior del PBG1
31 %
32 % Y_int_borde2: Matriz de mallado de las coordenadas de 'wi' para aquellas
33 % configuraciones donde exista el borde inferior del PBG1
34 %
35 % Z_int_borde2: Matriz de mallado de las coordenadas de 'a' para aquellas
36 % configuraciones donde exista el borde inferior del PBG1
37 %

```

```

38 % W_int_borde2: Interpolación cúbica de los valores de lambda del borde
39 % inferior del PBG1
40 %
41 % tolerancia_borde2: Tolerancia que presentan los valores de lambda del
42 % borde inferior del PBG1
43 %
44 % X_int_borde3: Matriz de mallado de las coordenadas de 'we' para aquellas
45 % configuraciones donde exista el borde superior del PBG2
46 %
47 % Y_int_borde3: Matriz de mallado de las coordenadas de 'wi' para aquellas
48 % configuraciones donde exista el borde superior del PBG2
49 %
50 % Z_int_borde3: Matriz de mallado de las coordenadas de 'a' para aquellas
51 % configuraciones donde exista el borde superior del PBG2
52 % W_int_borde3: Interpolación cúbica de los valores de lambda del borde
53 % superior del PBG2
54 % tolerancia_borde3: Tolerancia que presentan los valores de lambda del
55 % borde superior del PBG2
56 %
57 % X_int_borde4: Matriz de mallado de las coordenadas de 'we' para aquellas
58 % configuraciones donde exista el borde inferior del PBG2
59 %
60 % Y_int_borde4: Matriz de mallado de las coordenadas de 'wi' para aquellas
61 % configuraciones donde exista el borde inferior del PBG2
62 %
63 % Z_int_borde4: Matriz de mallado de las coordenadas de 'a' para aquellas
64 % configuraciones donde exista el borde superior del PBG2
65 %
66 % W_int_borde4: Interpolación cúbica de los valores de lambda del borde
67 % inferior del PBG2
68 %
69 % tolerancia_borde4: Tolerancia que presentan los valores de lambda del
70 % borde inferior del PBG2
71 %
72 % num_interv: indica el tamaño de equiespaciado de las interpolaciones de
73 % 'a_vector', 'wi_vector' y 'we_vector'
74 % u: numero de figura que se representa
75 % estructura: nombre de la estructura de simulación
76 %
77 %%%Salida:
78 % mapacolor4D_plano1_tol: Representación de la tolerancia de los valores
79 % del plano donde lambda=1550 nm para los bordes superiores del PBG1
80 %
81 % mapacolor4D_plano2_tol: Representación de la tolerancia de los valores
82 % del plano donde lambda=1550 nm para los bordes inferiores del PBG1
83 %
84 % mapacolor4D_plano3_tol: Representación de la tolerancia de los valores
85 % del plano donde lambda=1550 nm para los bordes superiores del PBG2
86 %
87 % mapacolor4D_plano4_tol: Representación de la tolerancia de los valores
88 % del plano donde lambda=1550 nm para los bordes inferiores del PBG2
89 %
90 %
91 %
92 %%% Paso 1. Definición de parámetros en función del borde de PBG
93
94 s=1;
95
96 switch estructura
97 case 'corr3D'
98 f=4:7;
99 case 'corr3Dinv'
100 f=4:5;
101 end
102
103 for m=f
104

```

```

105 figure(u);
106 if m==4
107 X_int=X_int_borde1;
108 Y_int=Y_int_borde1;
109 Z_int=Z_int_borde1;
110 W_int=W_int_borde1;
111 tolerancia=tolerancia_borde1;
112 mapacolor4D_plano1_tol=figure(u);
113 elseif m==5
114 X_int=X_int_borde2;
115 Y_int=Y_int_borde2;
116 Z_int=Z_int_borde2;
117 W_int=W_int_borde2;
118 tolerancia=tolerancia_borde2;
119 mapacolor4D_plano2_tol=figure(u);
120 elseif m==6
121 X_int=X_int_borde3;
122 Y_int=Y_int_borde3;
123 Z_int=Z_int_borde3;
124 W_int=W_int_borde3;
125 tolerancia=tolerancia_borde3;
126 mapacolor4D_plano3_tol=figure(u);
127 else
128 X_int=X_int_borde4;
129 Y_int=Y_int_borde4;
130 Z_int=Z_int_borde4;
131 W_int=W_int_borde4;
132 tolerancia=tolerancia_borde4;
133 mapacolor4D_plano4_tol=figure(u);
134 end
135
136 %% Paso 2. Obtención de las coordenadas de 'we','wi' e 'a' donde los
137 %% valores de borde de PBG vale lambda=1550 nm
138
139 %Para poder obtener los valores exactos es necesario redondear los
140 %datos iniciales
141 W_round=round(W_int,4,'significant');
142 X_1550=X_int(find(W_round==1550));
143 Y_1550=Y_int(find(W_round==1550));
144 Z_1550=Z_int(find(W_round==1550));
145
146 %% Paso 3. Obtención de la tolerancia para los valores posibles del borde
147 %% de PBG donde lambda=1550 nm
148
149 pos_1550=find(W_round==1550);
150 tolerancia_1550=tolerancia(pos_1550);
151
152 %% Paso 4. Mallado de las coordenadas obtenidas en el paso 2 y de la
153 %% tolerancia obtenida en el paso 3
154
155 vectorA=linspace(min(X_1550),max(X_1550),num_interv);
156 vectorB=linspace(min(Y_1550),max(Y_1550),num_interv);
157 [A,B]=meshgrid(vectorA,vectorB);
158 C=griddata(X_1550,Y_1550,Z_1550,A,B);
159
160 %Mallado de la tolerancia
161 G=griddata(X_1550,Y_1550,Z_1550,tolerancia_1550,A,B,C);
162
163 %% Paso 5. Representación del plano de tolerancias y ajuste de las
164 %% propiedades de las gráficas
165
166 surf(A,B,C,G);shading flat
167 c=colorbar;
168 c.Label.String = 'Variabilidad';
169 titulo=['Tolerancia del plano donde \lambda=1550 nm para el borde ',...
170 num2str(s), ' de PBG'];

```

```

171 title(titulo); xlabel('W_{e}'); ylabel('W_{i}'); zlabel('a');
172 set(gcf, 'position', get(groot, 'Screensize'));
173
174 switch estructura
175 case 'corr3D'
176 axis([700 1600 60 180 300 420])
177 case 'corr3Dinv'
178 axis([100 400 200 350 400 550])
179 end
180
181 u=u+1;
182 s=s+1;
183 end
184
185 switch estructura
186 case 'corr3Dinv'
187 mapacolor4D_plano3_tol=NaN;
188 mapacolor4D_plano4_tol=NaN;
189 end
190 end
191

```

## 1.8 Funciones de apoyo

### 1.8.1 interseccion\_mapacolor\_plano1550.m

```

1 function [plano3D_slice_borde1,plano3D_slice_borde2,plano3D_slice_borde3,...
2 plano3D_slice_borde4]=interseccion_mapacolor_plano1550(matrizBG,...
3 a_vector,wi_vector,we_vector)
4 %
5 % %%Representación gráfica del plano 3D que forma todas las posibles
6 % %%configuraciones entre las variables a,we,wi que presentan estimulación
7 % %%para lambda=1550 nm. Dichos planos se separan en función del borde de
8 % %%band gap de estudio. Para facilitar su comprensión se añaden planos
9 % %%2D horizontales denominados 'slice'. Cada uno de estos representa,
10 % %%mediante un mapa de color, el valor de lambda que se estimula conforme
11 % %%varía we y wi para un valor de a constate. Además, se dibuja una línea
12 % %%roja en cada 'slice' que marca las configuraciones para las cuales
13 % %%lambda=1550 nm. Dichas líneas también se representan juntas en la
14 % %%"base" del plano formado por los ejes de coordenadas x e y (we y wi
15 % %%respectivamente) Las intersecciones entre el plano 3D y los slice
16 % %%corroboran la correcta representación del plano 3D puesto que el plano
17 % %%los atraviesa por la línea roja.
18 %
19 %%Entrada:
20 % matrizBG: matriz que contiene todos los datos originales de la simulación
21 % a_vector: interpolación de los posibles valores de 'a' para un número
22 % conocido de puntos equiespaciados
23 %
24 % wi_vector: interpolación de los posibles valores de 'wi' para un número
25 % conocido de puntos equiespaciados
26 %
27 % we_vector: interpolación de los posibles valores de 'we' para un número
28 % conocido de puntos equiespaciados
29 %
30 %%Salida:
31 % plano3D_slice_borde1: representación gráfica del plano 3D, tres slice y
32 % sus correspondientes líneas rojas para el borde superior de PBG1
33 %
34 % plano3D_slice_borde2: representación gráfica del plano 3D, tres slice y
35 % sus correspondientes líneas rojas para el borde inferior de PBG1
36 %

```

```

37 % plano3D_slice_borde3: representación gráfica del plano 3D, tres slice y
38 % sus correspondientes líneas rojas para el borde superior de PBG2
39 %
40 % plano3D_slice_borde4: representación gráfica del plano 3D, tres slice y
41 % sus correspondientes líneas rojas para el borde inferior de PBG2
42 %
43 %
44 %
45 %% Paso 1. Mallado de las coordenadas we,wi,a en las matrices X,Y,Z para
46 %% sus correspondientes valores de lambda almacenado en W
47
48 u=1;
49 [X,Y,Z]=meshgrid(we_vector,wi_vector,a_vector);
50 for m=4:7
51 %Eliminación de los posibles 'NaN' existentes en los bordes de PBG
52 pos_sinNaN=find(isnan(matrizBG(m,:))==0);
53 matrizBG_sinNaN=matrizBG(:,pos_sinNaN);
54 W=griddata(matrizBG_sinNaN(3,:),matrizBG_sinNaN(2,:),...
55 matrizBG_sinNaN(1,:),matrizBG_sinNaN(m,:),X,Y,Z);
56
57 %% Paso 2. Obtención de las coordenadas para las cuales lambda=1550 nm
58
59 coordenadas_total=[];
60 for n=1:size(W,3)
61 [coordenadas]=contour(X(:,:,1),Y(:,:,1),W(:,:,n),1550*[1 1],'k');
62 if isempty(coordenadas)==0
63 coordenadas=coordenadas(:,2:length(coordenadas));
64 a=Z(1,1,n)*ones(1,length(coordenadas));
65 coordenadas(3,:)=a;
66 coordenadas_total=[coordenadas_total coordenadas];
67 else
68 end
69 end
70
71 %% Paso 3. Representación de los planos 3D 'slice'
72
73 figure(u)
74 long=size(coordenadas_total,2);
75 long=[long/4 long*.5 long*3/4]; %Los planos se eligen aproximadamente
76 %equiespaciados
77 long=round(long); %Se redondea para encontrar la posición deseada en
78 %la matriz (las posiciones son números enteros)
79 a_slice=[coordenadas_total(3,long(1)) coordenadas_total(3,long(2))
coordenadas_total(3,long(3))];
80 p2=slice(X,Y,Z,W,[],[],a_slice);
81 shading flat
82 hold on
83
84 %% Paso 4. Representación de las líneas donde lambda=1550 nm para cada
85 %% plano 'slice' realizado en el paso 3
86 for n=a_slice
87 posiciones=find(coordenadas_total(3,:)==n);
88 p3=plot3(coordenadas_total(1,posiciones),...
89 coordenadas_total(2,posiciones),coordenadas_total(3,posiciones),...
90 'r');
91 if m==4 || m==5 || m==6
92 plot3(coordenadas_total(1,posiciones),...
93 coordenadas_total(2,posiciones),300*ones(1,size(posiciones,2)),...
94 'r')
95 else
96 plot3(coordenadas_total(1,posiciones),...
97 coordenadas_total(2,posiciones),390*ones(1,size(posiciones,2)),...
98 'r')
99 end
100 hold on
101 end
102

```

```

103 %% Paso 5. Representación del plano 3D cuyas coordenadas marcan las
posibles
104 %% configuraciones para las cuales lambda=1550 nm
105
106 % Es necesario realizar un mallado únicamente con las coordenadas de
interés
107 vectorA=linspace(min(coordenadas_total(1,:)),...
108 max(coordenadas_total(1,:),50);
109 vectorB=linspace(min(coordenadas_total(2,:)),...
110 max(coordenadas_total(2,:),50);
111 [A,B]=meshgrid(vectorA,vectorB);
112 C=griddata(coordenadas_total(1,:),coordenadas_total(2,:),...
113 coordenadas_total(3,:),A,B);
114 p1=surf(A,B,C);shading flat
115 hold on
116 c=colorbar;
117 c.Label.String='lambda (\lambda)';
118
119 %% Paso 6. Almacenamiento en las variables de salida de las gráficas
120 %% obtenidas, ajuste de rangos de representación de ejes de
121 %% coordenadas, rangos de colorbar y adición de leyenda y título
122
123 if m==4
124 plano3D_slice_borde1=figure(u);
125 caxis([1500 1600]); %se ajusta el rango de la escala de colores
126 axis([700 1600 60 180 300 330]) %se ajustan los rangos de
127 %representación de los ejes de coordenadas para una óptima
128 %visualización
129 axis([700 1600 60 180 300 330])
130 elseif m==5
131 plano3D_slice_borde2=figure(u);
132 caxis([1500 1600]);
133 axis([700 1600 60 180 300 330])
134 elseif m==6
135 plano3D_slice_borde3=figure(u);
136 caxis([1300 1650]);
137 axis([700 1600 60 180 300 420])
138 else
139 plano3D_slice_borde4=figure(u);
140 caxis([1500 1580]);
141 axis([700 1600 60 180 390 420])
142 end
143 legend([p1 p2(1,1) p3], 'Plano 3D donde \lambda=1550 nm para el borde de
PBG ',...
144 'Colormap 3D', 'Línea 2D donde \lambda=1550 nm')
145 set(gcf, 'position', get(groot, 'ScreenSize'));
146 titulo=['Intersección entre colormaps 2D y el plano donde \lambda=1550
'...
147 'nm para el borde ', num2str(u), ' de PBG'];
148 title(titulo); xlabel('W_{e}'); ylabel('W_{i}'); zlabel('a');
149 u=u+1;
150 end
151 end

```

## 2. ARCHIVOS MODELO UTILIZADOS

### 2.1 corr3D-a380-wi100-we1500-h220-upper\_DIW-w450-5bandas.cti

```
; Guía corrugada 3D de SOI con h = 220 nm

(define grid-step 10) ; Fijo un step para el grid, que cogeré de 10 nm para que
se ajusten todas las dimensiones

; definiré los elementos de la celda básica para que los bordes caigan de forma exacta
en el mallado

; Utilizo una supercelda cúbica --> eje z para la altura, eje y la dirección de los
elementos transversales, eje x la dirección de propagación

(define eps_Si (* 3.4777 3.4777)) ; Constante dieléctrica del Si (sacado de
refractiveindex.info)
(define eps_SiO2 (* 1.444 1.444)) ; Constante dieléctrica del substrato de SiO2
(sacado de refractiveindex.info)
(define eps_aire 1) ; Constante dieléctrica del aire
(define eps_DIW (* 1.333 1.333)) ; Constante dieléctrica del agua
(define a 380) ; Periodo de la guía corrugada
(define h 220) ; Altura de la guía de Si
(define w 450) ; Anchura de la guía de Si
(define wi 100) ; Anchura de los elementos transversales de Si
(define we 1500) ; Longitud total del elemento transversal de Si

(define h-supercell (* 5 h)) ; Altura de la supercelda en nm
(define w-supercell 3000) ; Anchura de la supercelda en nm

; Calculo los parámetros normalizados
(define grid-step_norm (/ grid-step a))
(define h_norm (/ h a))
(define w_norm (/ w a))
(define wi_norm (/ wi a))
(define we_norm (/ we a))
(define h-supercell_norm (/ h-supercell a))
(define w-supercell_norm (/ w-supercell a))
```

```

(set! geometry-lattice (make lattice (size 1 w-supercell_norm h-supercell_norm)))
; Hacemos una supercelda rectangular de altura h-supercell y anchura w-supercell

(set! default-material (make dielectric (epsilon eps_DIW))) ; considero que
tenemos DIW por encima

(set! geometry (list
  (make block (center 0 0 0) (size 1 w_norm h_norm) (material (make dielectric
(epsilon eps_Si))))
  (make block (center 0 0 0) (size wi_norm we_norm h_norm) (material (make
dielectric (epsilon eps_Si))))
)) ; Guía de Si con elementos transversales

(set! geometry (append geometry (list
  (make block (center 0 0 (/ (+ h_norm h-supercell_norm) -4)) (size 1 w-
supercell_norm (/ (- h-supercell_norm h_norm) 2)) (material (make dielectric (epsilon
eps_SiO2))))
))) ; Lower cladding de SiO2

(set! grid-size (vector3 (+ (/ a grid-step) 1) (+ (/ w-supercell grid-step) 1) (+ (/
h-supercell grid-step) 1)))
(set! num-bands 5)

;(define Gamma (cartesian->reciprocal (vector3 0 0 0)))
(define Kp1 (cartesian->reciprocal (vector3 0.3 0 0)))
(define Kp (cartesian->reciprocal (vector3 0.5 0 0)))

;(set! k-points (list Gamma Kp))
(set! k-points (list Kp1 Kp))

(set! k-points (interpolate 7 k-points))

; Calculo las bandas y miro sus simetrías
(run-zeven display-yparities)

```

## 2.2 simulaciones\_scriptCorr3D

```
mpb corr3D-a380-wi100-we1500-h220-upper_DIW-w450-5bandas.ctl > corr3D-a380-wi100-  
we1500-h220-upper_DIW-w450-5bandas.out  
grep freq corr3D-a380-wi100-we1500-h220-upper_DIW-w450-5bandas.out > corr3D-a380-  
wi100-we1500-h220-upper_DIW-w450-5bandas.freq  
grep zpar corr3D-a380-wi100-we1500-h220-upper_DIW-w450-5bandas.out > corr3D-a380-  
wi100-we1500-h220-upper_DIW-w450-5bandas.zpar  
grep ypar corr3D-a380-wi100-we1500-h220-upper_DIW-w450-5bandas.out > corr3D-a380-  
wi100-we1500-h220-upper_DIW-w450-5bandas.ypar  
mpb-data -r -x 5 epsilon.h5  
h5tovtk -d data-new epsilon.h5  
mv epsilon.vtk corr3D-a380-wi100-we1500-h220-upper_DIW-w450-5bandas-epsilon.vtk
```