



E.T.S.I.S. TELECOMUNICACIÓN

## PROYECTO FIN DE CARRERA PLAN 2000

**TEMA:** PUESTA A PUNTO DE UNA HERRAMIENTA EDUCACIONAL Y ANÁLISIS DE REFLECTORES Y BOCINAS EN ENTORNO MATLAB (SABOR)

**TÍTULO:** ANÁLISIS DE BOCINAS EN MATLAB

**AUTOR:** Javier Montero Carretero

**TUTOR:** Leandro de Haro Ariet

Vº Bº.

**DEPARTAMENTO:** TSC

**Miembros del Tribunal Calificador:**

**PRESIDENTE:** Lourdes López Santidrián

**VOCAL:** Leandro de Haro Ariet

**VOCAL SECRETARIO:** Florentino Jiménez Muñoz

**DIRECTOR:**

**Fecha de lectura:**

**Calificación:**

El Secretario,

### RESUMEN DEL PROYECTO:

En este proyecto se ha implementado un nuevo programa que parte de un software utilizado en el laboratorio de la asignatura Antenas y Compatibilidad Electromagnética llamado SABOR. Consiste en una herramienta educacional que pretende ser actualizada en versiones de sistemas operativos modernos utilizando la herramienta MATLAB. Se han analizado 6 tipos de antenas de bocinas: sectorial H, sectorial E, piramidal, cónica, piramidal corrugada y cónica corrugada.

La herramienta creada consiste en una interfaz gráfica que tiene como entradas las variables elementales de diseño de la apertura como por ejemplo: dimensiones de la propia bocina o parámetros generales comunes a todas ellas. A su vez, el software nos genera un conjunto de parámetros de salida fundamentales de las antenas de bocinas: Directividad, Ancho de haz, Centro de fase y Spillover.

El proyecto está desarrollado para que pueda servir de información teórico-práctico. Por ello, además de revisar la teoría de las bocinas, mostrará la información relacionada con la programación orientada a objetos en entorno de desarrollo MATLAB cuyo objetivo propio es adquirir una nueva forma de pensamiento acerca del proceso de descomposición de problemas.

Finalmente, se ha creado un manual de autoayuda para dar soporte al software.

## **RESUMEN DEL PROYECTO**

El presente proyecto parte de un programa utilizado en las prácticas de laboratorio en la asignatura Antenas y Compatibilidad Electromagnética del sexto semestre llamado SABOR, que pretende ser actualizado para que en las nuevas versiones de los sistemas operativos ofrecidos por la compañía Windows pueda ser operativo. El objetivo principal será diseñar e implementar nuevas funcionalidades así como desarrollar mejoras y corregir errores del mismo.

Para su mejor entendimiento se ha creado una herramienta en entorno MATLAB para analizar uno de los tipos más comunes de Apertura que se utilizan actualmente, las bocinas. Dicha herramienta es una interfaz gráfica que tiene como entradas las variables elementales de diseño de la apertura como por ejemplo: dimensiones de la propia bocina o los parámetros generales comunes a todas ellas. A su vez, el software nos genera algunos de los parámetros de salida fundamentales de las antenas: Directividad, Ancho de haz, Centro de fase y Spillover.

Para el correcto desarrollo del software se ha realizado numerosas pruebas con el fin de depurar y corregir errores con respecto a la anterior versión del SABOR. Por otra parte se ha hecho también hincapié en la funcionalidad del programa para que sea más intuitivo y evitar complejidades.

El tipo de antena que se pretende estudiar es la bocina que consiste en una guía de onda en la cual el área de la sección se va incrementando progresivamente hasta un extremo abierto, que se comporta como una apertura. Se utilizan extensamente en satélites comerciales para coberturas globales desde órbitas geoestacionarias, pero el uso más común es como elemento de radiación para reflectores de antenas. Los tipos de bocinas que se van a examinar en la herramienta son: Sectorial H, Sectorial E, Piramidal, Cónica, Cónica Corrugada y Piramidal Corrugada.

El proyecto está desarrollado de manera que pueda servir de información teórico-práctico de todo el software SABOR. Por ello, el documento además de revisar la teoría de las bocinas analizadas, mostrará la información relacionada con la programación orientado a objetos en entorno MATLAB cuyo objetivo propio es adquirir una nueva forma de pensamiento acerca del proceso de descomposición de problemas y desarrollo de soluciones de programación.

Finalmente se ha creado un manual de autoayuda para dar soporte al software y se han incluido los resultados de diversas pruebas realizadas para poder observar todos los detalles de su funcionamiento, así como las conclusiones y líneas futuras de acción.

## **PROJECT ABSTRACT**

This Project comes from a program used in the labs of the subject Antennas and Electromagnetic Compatibility in the sixth semester called SABOR, which aims to be updated in order to any type of computer running a Windows operating systems(Windows 7 and subsequent versions). The main objectives are design and improve existing functionalities and develop new features. In addition, we will correct mistakes in earlier versions.

For a better understanding a new custom tool using MATLAB environment has been created to analyze one of the most common types of apertura antenna which is used for the moment, horns. This tool is a graphical interface that has elementary design variables as a inputs, for example: Dimensions of the own horn or common general parameters of all horns. At the same time, the software generate us some of the fundamental parameters of antennas output like Directivity, Beamwidth, Phase centre and Spillover.

This software has been performed numerous tests for the proper functioning of the Software and we have been cared in order to debug and correct errors that were detected in earlier versions of SABOR. In addition, it has also been emphasized the program's functionality in order to be more intuitive and avoiding unnecessary barriers or complexities.

The type of antenna that we are going to study is the horn which consists of a waveguides which the section area has been gradually increasing to an open-ended, that behaves as an aperture. It is widely used in comercial satellites for global coverage from geostationary orbits. However, the most common use is radiating element for antenna reflectors. The types of horns which is going to be considered are: Rectangular H-plane sectorial, Rectangular E-plane sectorial, Rectangular Pyramidal, Circular, Corrugated Circular and Corrugated Pyramidal.

The Project is developed so that it can be used as practical-theoretical information around the SABOR software. Therefore, In addition to thoroughly reviewing the theory document of analyzed horns, it display information related to the object-oriented programming in MATLAB environment whose goal leads us to a new way of thinking about the process of decomposition of problems and solutions development programming.

Finally, it has been created a self-help manual in order to support the software and has been included the results of different tests to observe all the details of their operations, as well as the conclusions and future action lines.

## **AGRADECIMIENTOS**

Quiero dedicar este proyecto a mis padres, por apoyarme desde el primer minuto de la carrera y confiar siempre en mí tanto en los buenos como en los malos momentos. Sin olvidar a mi hermana Raquel cuyos ánimos y cariños me han motivado para seguir creciendo y dando energías para superar mis metas.

A mis amigos de la universidad por hacerme pasar uno de los mejores periodos de mi vida, ya que este libro no sería lo mismo sin esos ratos de charloteo en la cafetería, sin esas fiestas por Madrid que no olvidaré jamás, ni nuestras peculiares salidas. Momentos únicos que con vuestro apoyo y ayuda no hubiese sido lo mismo.

A mis amigos del barrio, agradecerles los momentos de ayuda en los que con esfuerzo y sacrificio podemos ser grandes con distintas formas de pensar. Momentos en el parque 88, en el centro comercial la vaguada no los olvidaré nunca.

A mi compañera de proyecto, Kathy por esas largas horas en la biblioteca de la vaguada, muy amenas con los hits del momento y del pasado. Agradecerla su esfuerzo y su adaptación a mis horarios de por la tarde. Y por qué no presumir, talento unido al talento equivale a talento, y solo así es posible conseguir los objetivos en esta vida.

Por último agradecer a nuestro tutor Leandro de Haro por habernos abierto la posibilidad de realizar este magnífico proyecto.

## ÍNDICE DE CONTENIDOS

### **Contenido**

CAPÍTULO I: INTRODUCCIÓN.....	1
CAPÍTULO II: TEORÍA DE BOCINAS .....	3
2.1 BOCINA SECTORIAL PLANO H .....	3
2.1.1 Introducción .....	3
2.1.2 Campo en la apertura. Parámetros $P_x(\theta, \phi)$ y $P_y(\theta, \phi)$ .....	4
2.1.3 Componentes polares normalizadas de campo lejano: $e_\theta$ y $e_\phi$ .....	4
2.1.4 Componentes polares de campo lejano: $E_\theta$ y $E_\phi$ .....	5
2.1.5 Diagrama normalizado de campo $f_E$ .....	5
2.1.6 Componentes copolar y contrapolar del campo (según polariz. en y): $E_{CP}$ y $E_{XP}$ .....	7
2.1.7 Potencia radiada .....	7
2.1.8 Directividad .....	8
2.1.9 Eficiencia de spillover .....	9
2.1.10 Centro de fase .....	9
2.2 BOCINA SECTORIAL PLANO E.....	10
2.2.1 Introducción .....	10
2.2.2 Campo en la apertura. Parámetros $P_x(\theta, \phi)$ y $P_y(\theta, \phi)$ .....	11
2.2.3 Componentes polares normalizadas de campo lejano: $e_\theta$ y $e_\phi$ .....	11
2.2.4 Componentes polares de campo lejano: $E_\theta$ y $E_\phi$ .....	12
2.2.5 Diagrama normalizado de campo $f_E$ .....	12
2.2.6 Componentes copolar y contrapolar del campo (según polariz. en y): $E_{CP}$ y $E_{XP}$ .....	13
2.2.7 Potencia radiada .....	14
2.2.8 Directividad .....	14
2.2.9 Eficiencia de spillover .....	15
2.2.10 Centro de fase .....	16
2.3 BOCINA PIRAMIDAL.....	17
2.3.1 Introducción .....	17
2.3.2 Campo en la apertura. Parámetros $P_x(\theta, \phi)$ y $P_y(\theta, \phi)$ .....	18
2.3.3 Componentes polares normalizadas de campo lejano: $e_\theta$ y $e_\phi$ .....	19
2.3.4 Componentes polares de campo lejano: $E_\theta$ y $E_\phi$ .....	20
2.3.5 Diagrama normalizado de campo $f_E$ .....	20
2.3.6 Componentes copolar y contrapolar del campo (según polariz. en y): $E_{CP}$ y $E_{XP}$ .....	21
2.3.7 Potencia radiada .....	22
2.3.8 Directividad .....	22

2.3.9 Eficiencia de spillover .....	23
2.3.10 Centro de fase .....	24
2.4 BOCINA CÓNICA.....	25
2.4.1 Introducción .....	25
2.4.2 Campo en la apertura. Parámetros $P_x(\theta, \phi)$ y $P_y(\theta, \phi)$ .....	26
2.4.3 Componentes polares normalizadas de campo lejano: $e_\theta$ y $e_\phi$ .....	29
2.4.4 Componentes polares de campo lejano: $E_\theta$ y $E_\phi$ .....	29
2.4.5 Diagrama normalizado de campo $f_E$ .....	30
2.4.6 Componentes copolar y contrapolar del campo (según polariz. en y): $E_{CP}$ y $E_{XP}$ .....	33
2.4.7 Potencia radiada .....	34
2.4.8 Directividad .....	35
2.4.9 Eficiencia de spillover .....	37
2.4.10 Centro de fase .....	38

## CAPÍTULO III: METÓDO DE INTEGRACIÓN GAUSS-LEGENDRE . 41

3.1 INTEGRALES SIMPLES.....	41
3.1.1 Límites de integración entre -1 y 1 .....	41
3.1.2 Límites de integración entre a y b .....	45
3.2 INTEGRALES DOBLES .....	45
3.3 OBTENCIÓN DE LAS RAÍCES Y PESOS DEL POLINOMIO DE LEGENDRE.....	47

## CAPÍTULO IV: METODOLOGÍA DE PROGRAMACIÓN ORIENTADA A OBJETOS 49

4.1 INTRODUCCIÓN A LA PROGRAMACIÓN ORIENTADA A OBJETOS.....	49
4.1.1 Origen .....	49
4.1.2 Conceptos en la programación orientada a objetos (POO) .....	50
4.1.3 Características de la POO.....	50
4.2 PROGRAMACIÓN ORIENTADA A OBJETOS EN MATLAB .....	51
4.2.1 Jerarquía de clases en MATLAB.....	51
4.2.2 Definición de clases en MATLAB .....	53
4.2.3 Trabajar con objetos en MATLAB .....	58

## CAPÍTULO V: SIMULADOR "SABOR" ..... 60

5.1 ESTRUCTURA DEL PROGRAMA .....	60
5.1.1 Diagrama de flujo general .....	60
5.1.2 Diagrama de clases UML de la aplicación .....	62
5.1.3 Clase SECTORIALE .....	63
5.1.4 Clase SECTORIALH.....	64
5.1.5 Clase PIRAMIDAL .....	65
5.1.6 Clase CIRCULAR.....	67

5.1.7 Clase CIRCULARCORRU .....	68
5.1.8 Clase PIRCORRU .....	69
5.2. MANUAL DE USUARIO .....	71
5.2.1 Ventana Inicio.....	71
5.2.2 Ventana Principal .....	72
5.3 ENTRADAS Y SALIDAS DE CADA DIRECTORIO .....	83
5.3.1 @SECTORIALE.....	83
5.3.2 @SECTORIALH .....	88
5.3.3 @PIRAMIDAL.....	93
5.3.4 @CIRCULAR .....	98
5.3.5 @CIRCULARCORRU.....	103
5.3.6 @PIRCURRU .....	107
5.4 COMPILADOR DE MATLAB. CREACIÓN DE EJECUTABLES.EXE .....	113
5.5 EJEMPLOS DE ANÁLISIS REALIZADOS EN SABOR .....	116
5.5.1 Sectorial E .....	116
5.5.2 Sectorial H.....	118
5.5.3 Sectorial Piramidal.....	120
5.5.4 Cónica .....	122
5.5.5 Circular corrugada.....	124
5.5.6 Piramidal .....	126
<b>CAPÍTULO VI CONCLUSIONES Y LINEAS FUTURAS.....</b>	<b>129</b>
<b>CAPÍTULO VII BIBLIOGRAFÍA .....</b>	<b>132</b>

## ÍNDICE DE FIGURAS

Fig 2.1 Bocina sectorial plano H.....	3
Fig 2.2 Bocina sectorial plano E .....	10
Fig 2.3 Bocina sectorial piramidal.....	17
Fig 2.4 Bocina corrugada piramidal .....	18
Fig 2.5 Bocina cónica .....	25
Fig 2.6 Bocina corrugada cónica .....	26
Fig 2.7 Bocina corrugada cónica (2).....	26
Fig 4.1 Jerarquía de clases en Matlab.....	51
Fig 4.2 Contenido de control toolbox.....	52
Fig 4.3 Contenido directorio @lti.....	52
Fig 4.4 Ayuda clase LTI.....	53
Fig 4.5 Creación de la clase MICLASE.....	53
Fig 4.6 Creación del constructor de la clase MICLASE.....	54
Fig 4.7 Creación de un método de la clase MICLASE .....	55
Fig 4.8 Representación UML de la clase MICLASE .....	56
Fig 4.9 Representación UML de la clase MICLASE (2).....	56
Fig 4.10 Constructor de la clase CLASEHIJA .....	57
Fig 5.1 Contenido de la carpeta .....	60
Fig 5.2 Diagrama de flujo general .....	61
Fig 5.3 Diagrama de clases de la aplicación .....	62
Fig 5.4 Pantalla de Inicio .....	71
Fig 5.5 Pantalla Principal .....	72
Fig 5.6 Pantalla Principal menú Bocinas.....	72
Fig 5.7 Dimensiones sectorial plano E .....	73
Fig 5.8 Dimensiones sectorial plano H.....	74
Fig 5.9 Dimensiones sectorial Piramidal.....	75
Fig 5.10 Dimensiones Cónica .....	76
Fig 5.11 Dimensiones cónica corrugada.....	77
Fig 5.12 Dimensiones Piramidal Corrugada.....	78
Fig 5.13 Reflector .....	79
Fig 5.14 Parámetros generales .....	79
Fig 5.15 Formato de Diagrama .....	80
Fig 5.16 Gráfica .....	81
Fig 5.17 Comparar .....	81
Fig 5.18 Ayuda .....	82
Fig 5.19 Creación de un proyecto .....	113
Fig 5.20 Creación del ejecutable .....	114
Fig 5.21 Creación del ejecutable(2).....	114



Fig 5.22 Creación del ejecutable(3).....	115
Fig 5.23 Creación del ejecutable(4).....	115
Fig 5.24 Diagrama y resultados sectorial plano E .....	116
Fig 5.25 Diagrama y resultados sectorial plano H.....	118
Fig 5.26 Diagrama y resultados sectorial piramidal.....	120
Fig 5.27 Diagrama y resultados cónica .....	122
Fig 5.28 Diagrama y resultados corrugada cónica .....	124
Fig 5.29 Diagrama y resultados corrugada piramidal .....	126

# CAPÍTULO I: INTRODUCCIÓN

---

## CAPÍTULO I: INTRODUCCIÓN

El proyecto de fin de carrera pretende analizar uno de los tipos más comunes de antenas: las bocinas, a partir de la implementación de un software básico. El programa será creado utilizando el entorno y lenguaje de programación orientada a objetos de MATLAB para que posteriormente pueda ser utilizado como uso didáctico en los laboratorios de la universidad.

La línea de desarrollo llevada a cabo en este proyecto será:

- Marco teórico de bocinas: En primero lugar se explicará la base teórica del proyecto, incluyendo la mayoría de ecuaciones utilizadas en el programa. Esta parte pretende analizar cada uno de los tipos de bocinas con sus características principales. La finalidad de esta parte es ofrecer al usuario la parte teórica que utilizaremos en el programa SABOR y que pueda servir como base para futuras actualizaciones de la herramienta.
- Marco teórico de la cuadratura Gauss-Legendre: Se hará un estudio teórico de cómo convertir una integral simple o doble en un simple o doble sumatorio. Se incluirá además las entradas y salidas programadas en MATLAB de la cuadratura que luego será aplicado a nuestro programa principal.
- Marco teórico de la programación orientada a objetos en entorno MATLAB: Nuestra herramienta SABOR se ha creado programando orientada a objetos y por ello se dará una explicación detallada de cómo implementarlo en MATLAB. Esta parte pretende introducir al usuario al mundo de la programación orientada a objetos en entorno de desarrollo MATLAB, para así poder implementar programas de cualquier tipo de una manera más estructurada de lo habitual. Además, también tiene por objeto servir de estudio teórico para futuras versiones del software.
- Guía del usuario y códigos de generación de resultados: Para dar mayor soporte al usuario, se ha detallado una guía de usuario de la herramienta SABOR por si se tuviera que realizar alguna consulta en cuanto al uso de determinadas ventanas en el GUIDE de MATLAB.
- Resultados: Una vez vistos el análisis teórico-práctico del proyecto se incluirá los resultados de los parámetros de salida del programa obtenidos a partir de los datos de entrada. También se ha decidido poner un ejemplo concreto de cada uno de los tipos de bocinas para poder ver el diagrama de radiación.
- Conclusiones y líneas futuras: Por último, se indicará las conclusiones obtenidas y líneas futuras planteadas en la realización del Proyecto de Fin de Carrera.

# CAPÍTULO II: TEORÍA DE BOCINAS

## CAPÍTULO II: TEORÍA DE BOCINAS

En esta parte del proyecto se va a realizar un estudio teórico de cada una de las bocinas. Además se va a mostrar las formulas necesarias para implementar el programa.

### 2.1 BOCINA SECTORIAL PLANO H

#### 2.1.1 Introducción

La bocina sectorial plano H es el resultado de "abocinar" en una guía rectangular la dimensión "a", hasta convertirse en "W". Para obtener la expresión del campo en la apertura, pondremos los ejes x, y, de forma análoga a como se hizo en la guía rectangular añadiendo el eje z, tal y como se indica en la figura, en la cual se muestran sus parámetros dimensionales:  $R_1$ , R,  $R_H$  y  $l_H$ .

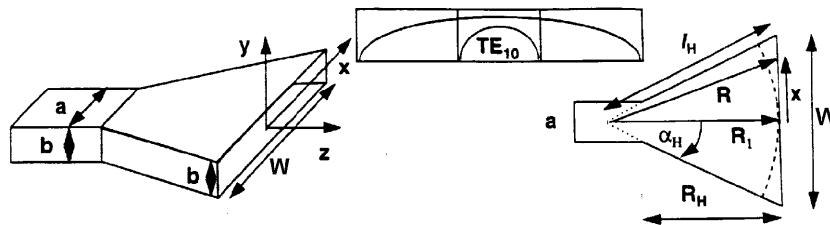


Fig. 2.1 Bocina sectorial plano H

Según la geometría mostrada, por triángulos semejantes se tiene que:

$$\frac{W/2}{R_1} = \frac{W/2 - a/2}{R_H}$$

De donde:

$$R_H = R_1 \left(1 - \frac{a}{W}\right)$$

También, aplicando el teorema de Pitágoras, se tiene que:

$$l_H = \sqrt{R_1^2 + \frac{W^2}{4}}$$

### 2.1.2 Campo en la apertura. Parámetros $P_x(\theta, \phi)$ y $P_y(\theta, \phi)$

El campo en la apertura, (añadimos subíndice para remarcarlo), será suponiéndolo polarizado linealmente según y:

$$\vec{E}_{a\_sectH}(x) = E_0 \cos\left(\pi \frac{x}{W}\right) e^{-j\delta(x)} \hat{y} = E_0 \cos\left(\pi \frac{x}{W}\right) e^{-jk_0 \frac{x^2}{2R_1}} \hat{y} = E_0 \cos\left(\pi \frac{x}{W}\right) e^{-j \frac{\pi x^2}{\lambda R_1}} \hat{y}$$

Es decir, el campo en la apertura, para la bocina sectorial plano H sólo tiene componente y, denominándola  $E_{ay\_sectH}$ , se tiene que vale:

$$E_{ay\_sectH}(x) = E_0 \cos\left(\pi \frac{x}{W}\right) e^{-jk_0 \frac{x^2}{2R_1}} = E_0 \cos\left(\pi \frac{x}{W}\right) e^{-j \frac{\pi x^2}{\lambda R_1}}$$

De lo visto anteriormente, no tenemos componente en x por lo tanto la potencia en x es 0:

$$P_{x\_sectH} = 0$$

Sustituyendo el valor de **E<sub>ay</sub>**, teniendo en cuenta la forma de la superficie de la apertura de la bocina  $S_{ap}$ , la cual determina los siguientes límites de integración  $-W/2 < x < W/2$ ,  $b/2 < y < b/2$  y utilizando el subíndice identificativo del tipo de bocina, resulta:

$$P_{y\_sectH}(\theta, \phi) = \int_{x=-W/2}^{W/2} \int_{y=-b/2}^{b/2} E_{ay\_sectH}(x) e^{-jk(ux+vy)} dx dy = \int_{x=-W/2}^{W/2} \int_{y=-b/2}^{b/2} \cos\left(\pi \frac{x}{W}\right) e^{-j \frac{\pi x^2}{\lambda R_1}} e^{-jk(ux+vy)} dx dy$$

### 2.1.3 Componentes polares normalizadas de campo lejano: $e_\theta$ y $e_\phi$

a) Según el primer principio de equivalencia.

Para la bocina sectorial plano H, en que  $P_x = 0$ , se convierte en:

$$\begin{bmatrix} e_\theta \\ e_\phi \end{bmatrix} = \frac{1 + \cos\theta}{2} \begin{bmatrix} \text{sen}\phi P_{y\_sectH} \\ \text{cos}\phi P_{y\_sectH} \end{bmatrix}$$

b) Según el segundo principio de equivalencia.

Para la bocina sectorial plano H, en que  $P_x = 0$ , se convierte en:

$$\begin{bmatrix} e_\theta \\ e_\phi \end{bmatrix} = \begin{bmatrix} \text{sen}\phi P_{y\_sec\ tH} \\ \text{cos}\phi \text{cos}\theta P_{y\_sec\ tH} \end{bmatrix}$$

### 2.1.4 Componentes polares de campo lejano: $E_\theta$ y $E_\phi$

Se obtienen a partir de las normalizadas  $e_\theta$  y  $e_\phi$  simplemente multiplicando por el siguiente factor que depende de la distancia  $r$  y de la longitud de onda:

$$\frac{je^{-jkr}}{\lambda r}$$

Es decir:

$$\begin{bmatrix} E_\theta \\ E_\phi \end{bmatrix} = \frac{je^{-jkr}}{\lambda r} \begin{bmatrix} e_\theta \\ e_\phi \end{bmatrix}$$

a) Según el primer principio de equivalencia, en bocinas sectoriales plano H ( $P_x=0$ ):

$$\begin{bmatrix} E_\theta \\ E_\phi \end{bmatrix} = \frac{je^{-jkr}}{\lambda r} \frac{1 + \text{cos}\theta}{2} \begin{bmatrix} \text{sen}\phi P_{y\_sec\ tH} \\ \text{cos}\phi P_{y\_sec\ tH} \end{bmatrix}$$

b) Según el segundo principio de equivalencia, en bocinas sectoriales plano H ( $P_x=0$ ):

$$\begin{bmatrix} E_\theta \\ E_\phi \end{bmatrix} = \frac{je^{-jkr}}{\lambda r} \begin{bmatrix} \text{sen}\phi P_{y\_sec\ tH} \\ \text{cos}\phi \text{cos}\theta P_{y\_sec\ tH} \end{bmatrix}$$

### 2.1.5 Diagrama normalizado de campo $f_E$

Para obtener el diagrama normalizado de campo debemos hallar el módulo del campo, el módulo del campo máximo y luego dividir estas dos cantidades.

a) Según el primer principio de equivalencia

Módulo del campo:

$$|\vec{E}| = \frac{1}{\lambda r} \frac{1 + \text{cos}\theta}{2} |P_{y\_sec\ tH}|$$

Módulo máximo del campo:

$$|\vec{E}|_{mx} = \frac{1}{\lambda r} |P_{y0\_sectH}|$$

Función normalizada de campo:

$$|f_{E1}| = \frac{1 + \cos \theta}{2} \frac{|P_{y\_sectH}|}{|P_{y0\_sectH}|}$$

Función normalizada de campo, expresada en dB:

$$f_{E1}|_{dB} = 20 \log \left( \frac{1 + \cos \theta}{2} \frac{|P_{y\_sectH}|}{|P_{y0\_sectH}|} \right)$$

b) Según el segundo principio de equivalencia:

Módulo del campo:

$$|\vec{E}| = \frac{1}{\lambda r} |P_{y\_sectH}| \sqrt{\text{sen}^2 \phi + \text{cos}^2 \phi \text{cos}^2 \theta}$$

Módulo máximo del campo:

$$|\vec{E}|_{mx} = \frac{1}{\lambda r} |P_{y0\_sectH}|$$

Función normalizada de campo:

$$|f_{E2}| = \frac{|P_{y\_sectH}|}{|P_{y0\_sectH}|} \sqrt{\text{sen}^2 \phi + \text{cos}^2 \phi \text{cos}^2 \theta}$$

Función normalizada de campo, expresada en dB:

$$|f_{E2}|_{dB} = 20 \log \left( \frac{|P_{y\_sectH}|}{|P_{y0\_sectH}|} \sqrt{\text{sen}^2 \phi + \text{cos}^2 \phi \text{cos}^2 \theta} \right)$$



### 2.1.6 Componentes copolar y contrapolar del campo (según polariz. en y): $E_{CP}$ y $E_{XP}$

Para la bocina sectorial plano H, en que  $P_x=0$ ,

a) Según el primer principio de equivalencia.

$$\begin{bmatrix} E_{CP} \\ E_{XP} \end{bmatrix} = \frac{je^{-jkr}}{\lambda r} \frac{1 + \cos\theta}{2} \begin{bmatrix} P_{y\_sectH} \\ 0 \end{bmatrix}$$

b) Según el segundo principio de equivalencia.

$$\begin{bmatrix} E_{CP} \\ E_{XP} \end{bmatrix} = \frac{je^{-jkr}}{\lambda r} \begin{bmatrix} (\text{sen}^2\phi + \cos^2\phi \cos\theta) P_{y\_sectH} \\ \text{sen}\phi \cos\phi (1 - \cos\theta) P_{y\_sectH} \end{bmatrix}$$

### 2.1.7 Potencia radiada

#### 2.1.7.1 Potencia radiada obtenida integrando el diagrama de radiación

Para la bocina sectorial plano H, en que  $P_x=0$ ,

a) Según el primer principio de equivalencia

$$P_{rad} = \frac{1}{2Z_0\lambda^2} \int_{\phi=0}^{2\pi} \int_{\theta=0}^{\pi} \frac{(1 + \cos\theta)^2}{4} |P_{y\_sectH}|^2 \text{sen}\theta d\theta d\phi$$

b) Según el segundo principio de equivalencia

$$P_{rad} = \frac{1}{2Z_0\lambda^2} \int_{\phi=0}^{2\pi} \int_{\theta=0}^{\pi} (\text{sen}^2\phi + \cos^2\phi \cos^2\theta) |P_{y\_sectH}|^2 \text{sen}\theta d\theta d\phi$$

#### 2.1.7.2 Potencia radiada obtenida integrando el campo en la apertura

El campo en la apertura de la bocina sectorial plano sólo tiene componente  $E_{ay}$ , teniendo en cuenta además que en la bocina sectorial plano H la forma de la superficie de la apertura de la bocina  $S_{ap}$  determina los siguientes límites de integración:  $-W/2 < x < W/2$ ,  $b/2 < y < b/2$  y utilizando el subíndice identificativo del tipo de bocina, se tiene:

$$P_{rad} = \frac{1}{2Z_0} \int_{x=-W/2}^{W/2} \int_{y=-b/2}^{b/2} |E_{ay\_sectH}(x)|^2 dx dy = \frac{1}{2Z_0} \int_{x=-W/2}^{W/2} \int_{y=-b/2}^{b/2} \left| \cos\left(\pi \frac{x}{W}\right) e^{-j\frac{\pi x^2}{\lambda R_1}} \right|^2 dx dy$$

### 2.1.8 Directividad

La expresión de la directividad viene dada por:

$$D = \frac{\frac{1}{2Z_0} \frac{1}{\lambda^2} (|P_{y0\_secIH}|^2)}{\frac{P_{rad}}{4\pi}}$$

Sustituyes la potencia radiada según el principio y el tipo de integración:

#### 2.1.8.1 Directividad calculando Prad por integración del diagrama

a) Según el primer principio de equivalencia.

$$D = 4\pi \frac{(|P_{y0\_secIH}|^2)}{\int_{\phi=0}^{2\pi} \int_{\theta=0}^{\pi} \frac{(1 + \cos\theta)^2}{4} |P_{y\_secIH}|^2 \text{sen}\theta d\theta d\phi} = 4\pi \frac{|\int_{x=-W/2, y=-b/2}^{W/2, b/2} E_{ay\_secIH}(x) dx dy|^2}{\int_{\phi=0}^{2\pi} \int_{\theta=0}^{\pi} \frac{(1 + \cos\theta)^2}{4} |P_{y\_secIH}|^2 \text{sen}\theta d\theta d\phi}$$

b) Según el segundo principio de equivalencia

$$D = 4\pi \frac{(|P_{y0\_secIH}|^2)}{\int_{\phi=0}^{2\pi} \int_{\theta=0}^{\pi} (\text{sen}^2\phi + \cos^2\phi \cos^2\theta) |P_{y\_secIH}|^2 \text{sen}\theta d\theta d\phi} = 4\pi \frac{|\int_{x=-W/2, y=-b/2}^{W/2, b/2} E_{ay\_secIH}(x) dx dy|^2}{\int_{\phi=0}^{2\pi} \int_{\theta=0}^{\pi} (\text{sen}^2\phi + \cos^2\phi \cos^2\theta) |P_{y\_secIH}|^2 \text{sen}\theta d\theta d\phi}$$

#### 2.1.8.2 Directividad calculando Prad por integración de los campos en la apertura

$$D = \frac{4\pi}{\lambda^2} \frac{(|P_{y0\_secIH}|^2)}{\int_{x=-W/2, y=-b/2}^{W/2, b/2} |E_{ay\_secIH}(x)|^2 dx dy} = \frac{4\pi}{\lambda^2} \frac{|\int_{x=-A/2, y=-b/2}^{W/2, b/2} E_{ay\_secIH}(x) dx dy|^2}{\int_{x=-W/2, y=-b/2}^{W/2, b/2} |E_{ay\_secIH}(x)|^2 dx dy}$$

### 2.1.9 Eficiencia de spillover

a) Según el primer principio de equivalencia,

$$\mathcal{E}_{spillover} = \frac{\int_{\phi=0}^{2\pi} \int_{\theta=0}^{\theta_0} \frac{(1 + \cos\theta)^2}{4} |P_{y\_sec\ tH}|^2 \text{sen}\theta d\theta d\phi}{\int_{\phi=0}^{2\pi} \int_{\theta=0}^{\pi} \frac{(1 + \cos\theta)^2}{4} |P_{y\_sec\ tH}|^2 \text{sen}\theta d\theta d\phi}$$

b) Según el primer principio de equivalencia,

$$\mathcal{E}_{spillover} = \frac{\int_{\phi=0}^{2\pi} \int_{\theta=0}^{\theta_0} (\text{sen}^2\phi + \cos^2\phi \cos^2\theta) |P_{y\_sec\ tH}|^2 \text{sen}\theta d\theta d\phi}{\int_{\phi=0}^{2\pi} \int_{\theta=0}^{\pi} (\text{sen}^2\phi + \cos^2\phi \cos^2\theta) |P_{y\_sec\ tH}|^2 \text{sen}\theta d\theta d\phi}$$

Dónde:

$$\theta_0 = \theta_{-XdB}$$

Con X entre 10 y 12 en la práctica

### 2.1.10 Centro de fase

La fórmula del centro de fase para un  $\phi$  genérico coincide para ambos principios de equivalencia y vale:

$$L_{ph} = \frac{\lambda}{2\pi(1 - \cos\theta)} \left\{ \arg[P_{y\_sec\ tH}(\theta_{-XdB}, \phi)] - \arg[P_{y0\_sec\ tH}] \right\}$$

Para el caso del **centro de fase medio**, coincide también para ambos principios de equivalencia:

$$L_{ph} = \frac{\lambda}{2\pi(1 - \cos\theta_{-XdB})} \cdot \frac{1}{2} \left\{ \arg[P_{y\_sec\ tH}(\theta_{-XdB}, 0^\circ)] + \arg[P_{y\_sec\ tH}(\theta_{-XdB}, 90^\circ)] - 2 \arg[P_{y0\_sec\ tH}] \right\}$$

## 2.2 BOCINA SECTORIAL PLANO E

### 2.2.1 Introducción

La bocina sectorial plano E es el resultado de "abocinar" en una guía rectangular la dimensión "b", hasta convertirse en "H" contenida en el plano E de radiación (el que contiene al vector **E**, es el plano vertical). Para obtener la expresión del campo en la apertura, pondremos los ejes x, y, de forma análoga a como se hizo en la guía rectangular añadiendo el eje z, en la cual se muestran sus parámetros dimensionales básicos:  $R_2$ , R,  $R_E$  y  $l_E$ .

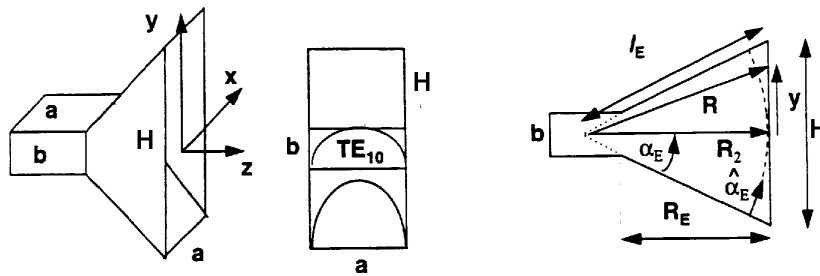


Fig. 2.2 Bocina sectorial plano E

Según la geometría mostrada, por triángulos semejantes se tiene que:

$$\frac{H/2}{R_2} = \frac{H/2 - b/2}{R_E}$$

De donde:

$$R_E = R_2 \left(1 - \frac{b}{H}\right)$$

También, aplicando el teorema de Pitágoras, se tiene que:

$$l_E = \sqrt{R_2^2 + \frac{H^2}{4}}$$

### 2.2.2 Campo en la apertura. Parámetros Px (θ,φ) y Py (θ,φ)

El campo en la apertura, (añadimos subíndice para remarcarlo), será suponiéndolo polarizado linealmente según y:

$$\vec{E}_{a\_sectE}(x) = E_0 \cos\left(\pi \frac{x}{a}\right) e^{-j\delta(y)} \hat{y} = E_0 \cos\left(\pi \frac{x}{a}\right) e^{-jk_0 \frac{y^2}{2R_2}} \hat{y} = E_0 \cos\left(\pi \frac{x}{a}\right) e^{-j \frac{\pi y^2}{\lambda R_2}} \hat{y}$$

Es decir, el campo en la apertura, para la bocina sectorial plano E, sólo tiene componente y, denominándola E<sub>ay\_ssectE</sub>, se tiene que vale:

$$E_{ay\_sectE}(x, y) = E_0 \cos\left(\pi \frac{x}{a}\right) e^{-jk_0 \frac{y^2}{2R_2}} = E_0 \cos\left(\pi \frac{x}{a}\right) e^{-j \frac{\pi y^2}{\lambda R_2}}$$

De lo visto anteriormente, no tenemos componente en x por lo tanto la potencia en x es 0:

$$P_{x\_sectEH} = 0$$

Sustituyendo el valor de **E<sub>ay</sub>**, teniendo en cuenta la forma de la superficie de la apertura de la bocina S<sub>ap</sub>, la cual determina los siguientes límites de integración  $-a/2 < x < a/2$ ,  $H/2 < y < H/2$  y utilizando el subíndice identificativo del tipo de bocina, resulta:

$$P_{y\_sectE}(\theta, \phi) = \int_{x=-a/2}^{a/2} \int_{y=-H/2}^{H/2} E_{ay\_sectE}(x, y) e^{-jk(ux+vy)} dx dy = \int_{x=-a/2}^{a/2} \int_{y=-H/2}^{H/2} \cos\left(\pi \frac{x}{W}\right) e^{-j \frac{\pi y^2}{\lambda R_2}} e^{-jk(ux+vy)} dx dy$$

### 2.2.3 Componentes polares normalizadas de campo lejano: e<sub>θ</sub> y e<sub>φ</sub>

a) Según el primer principio de equivalencia.

Para la bocina sectorial plano E, en que P<sub>x</sub> = 0, se convierte en:

$$\begin{bmatrix} e_{\theta} \\ e_{\phi} \end{bmatrix} = \frac{1 + \cos\theta}{2} \begin{bmatrix} \text{sen}\phi P_{y\_sectE} \\ \text{cos}\phi P_{y\_sectE} \end{bmatrix}$$

b) Según el segundo principio de equivalencia.

Para la bocina sectorial plano E, en que P<sub>x</sub> = 0, se convierte en:

$$\begin{bmatrix} e_{\theta} \\ e_{\phi} \end{bmatrix} = \begin{bmatrix} \text{sen}\phi P_{y\_sectE} \\ \text{cos}\phi \text{cos}\theta P_{y\_sectE} \end{bmatrix}$$

### 2.2.4 Componentes polares de campo lejano: $E_\theta$ y $E_\phi$

a) Según el primer principio de equivalencia, en bocinas sectoriales plano E ( $P_x=0$ ):

$$\begin{bmatrix} E_\theta \\ E_\phi \end{bmatrix} = \frac{je^{-jkr}}{\lambda r} \frac{1 + \cos\theta}{2} \begin{bmatrix} \text{sen}\phi P_{y\_sectE} \\ \text{cos}\phi P_{y\_sectE} \end{bmatrix}$$

b) Según el segundo principio de equivalencia, en bocinas sectoriales plano E ( $P_x=0$ ):

$$\begin{bmatrix} E_\theta \\ E_\phi \end{bmatrix} = \frac{je^{-jkr}}{\lambda r} \begin{bmatrix} \text{sen}\phi P_{Y\_sectE} \\ \text{cos}\phi \text{cos}\theta P_{Y\_sectE} \end{bmatrix}$$

### 2.2.5 Diagrama normalizado de campo $f_E$

Para obtener el diagrama normalizado de campo debemos hallar el módulo del campo, el módulo del campo máximo y luego dividir estas dos cantidades.

a) Según el primer principio de equivalencia

Módulo del campo:

$$|\vec{E}| = \frac{1}{\lambda r} \frac{1 + \cos\theta}{2} |P_{y\_sectE}|$$

Módulo máximo del campo:

$$|\vec{E}|_{mx} = \frac{1}{\lambda r} |P_{y0\_sectE}|$$

Función normalizada de campo:

$$|f_{E1}| = \frac{1 + \cos\theta}{2} \frac{|P_{y\_sectE}|}{|P_{y0\_sectE}|}$$

Función normalizada de campo, expresada en dB:

$$f_{E1}|_{dB} = 20 \log \left( \frac{1 + \cos\theta}{2} \frac{|P_{y\_sectE}|}{|P_{y0\_sectE}|} \right)$$

b) Según el segundo principio de equivalencia

Módulo del campo:

$$|\vec{E}| = \frac{1}{\lambda r} |P_{y\_sec tE}| \sqrt{\text{sen}^2 \phi + \text{cos}^2 \phi \text{cos}^2 \theta}$$

Módulo máximo del campo:

$$|\vec{E}|_{mx} = \frac{1}{\lambda r} |P_{y0\_sec tE}|$$

Función normalizada de campo:

$$|f_{E2}| = \frac{|P_{y\_sec tE}|}{|P_{y0\_sec tE}|} \sqrt{\text{sen}^2 \phi + \text{cos}^2 \phi \text{cos}^2 \theta}$$

Función normalizada de campo, expresada en dB:

$$|f_{E2}|_{dB} = 20 \log \left( \frac{|P_{y\_sec tE}|}{|P_{y0\_sec tE}|} \sqrt{\text{sen}^2 \phi + \text{cos}^2 \phi \text{cos}^2 \theta} \right)$$

**2.2.6 Componentes copolar y contrapolar del campo (según polariz. en y):  $E_{CP}$  y  $E_{XP}$**

Para la bocina sectorial plano E, en que  $P_x=0$ ,

a) Según el primer principio de equivalencia.

$$\begin{bmatrix} E_{CP} \\ E_{XP} \end{bmatrix} = \frac{je^{-jkr}}{\lambda r} \frac{1 + \text{cos} \theta}{2} \begin{bmatrix} P_{y\_sec tE} \\ 0 \end{bmatrix}$$

b) Según el segundo principio de equivalencia.

$$\begin{bmatrix} E_{CP} \\ E_{XP} \end{bmatrix} = \frac{je^{-jkr}}{\lambda r} \begin{bmatrix} (\text{sen}^2 \phi + \text{cos}^2 \phi \text{cos} \theta) P_{y\_sec tE} \\ \text{sen} \phi \text{cos} \phi (1 - \text{cos} \theta) P_{y\_sec tE} \end{bmatrix}$$

## 2.2.7 Potencia radiada

### 2.2.7.1 Potencia radiada obtenida integrando el diagrama de radiación

Para la bocina sectorial plano E, en que  $P_x=0$ ,

a) Según el primer principio de equivalencia

$$P_{rad} = \frac{1}{2Z_0\lambda^2} \int_{\phi=0}^{2\pi} \int_{\theta=0}^{\pi} \frac{(1 + \cos\theta)^2}{4} |P_{y\_sectE}|^2 \sin\theta d\theta d\phi$$

b) Según el segundo principio de equivalencia

$$P_{rad} = \frac{1}{2Z_0\lambda^2} \int_{\phi=0}^{2\pi} \int_{\theta=0}^{\pi} (\sin^2\phi + \cos^2\phi \cos^2\theta) |P_{y\_sectE}|^2 \sin\theta d\theta d\phi$$

### 2.2.7.2 Potencia radiada obtenida integrando la apertura

$$P_{rad} = \frac{1}{2Z_0} \int_{x=-a/2}^{a/2} \int_{y=-H/2}^{H/2} |E_{ay\_sectE}(x, y)|^2 dx dy = \frac{1}{2Z_0} \int_{x=-a/2}^{a/2} \int_{y=-H/2}^{H/2} \left| \cos\left(\pi \frac{x}{a}\right) e^{-j\frac{\pi y^2}{\lambda R_2}} \right|^2 dx dy$$

## 2.2.8 Directividad

La expresión de la directividad viene dada por:

$$D = \frac{\frac{1}{2Z_0} \frac{1}{\lambda^2} (|P_{y0\_sectE}|^2)}{\frac{P_{rad}}{4\pi}}$$

Sustituyes la potencia radiada según el principio y el tipo de integración:

### 2.2.8.1 Directividad calculando Prad por integración del diagrama

a) Según el primer principio de equivalencia,

$$D = 4\pi \frac{(|P_{y0\_sectE}|^2)}{\int_{\phi=0}^{2\pi} \int_{\theta=0}^{\pi} \frac{(1 + \cos\theta)^2}{4} |P_{y\_sectE}|^2 \sin\theta d\theta d\phi} = 4\pi \frac{\left| \int_{x=-a/2}^{a/2} \int_{y=-H/2}^{H/2} E_{ay\_sectE}(x, y) dx dy \right|^2}{\int_{\phi=0}^{2\pi} \int_{\theta=0}^{\pi} \frac{(1 + \cos\theta)^2}{4} |P_{y\_sectE}|^2 \sin\theta d\theta d\phi}$$



b) Según el primer principio de equivalencia

$$D = \frac{4\pi \int_{\phi=0}^{2\pi} \int_{\theta=0}^{\pi} (\sin^2 \phi + \cos^2 \phi \cos^2 \theta) |P_{y\_sec tE}|^2 \sin \theta d\theta d\phi}{\int_{x=-a/2}^{a/2} \int_{y=-H/2}^{H/2} |E_{ay\_sec tE}(x, y)|^2 dx dy} = 4\pi \frac{\int_{\phi=0}^{2\pi} \int_{\theta=0}^{\pi} (\sin^2 \phi + \cos^2 \phi \cos^2 \theta) |P_{y\_sec tE}|^2 \sin \theta d\theta d\phi}{\int_{x=-a/2}^{a/2} \int_{y=-H/2}^{H/2} |E_{ay\_sec tE}(x, y)|^2 dx dy}$$

**2.2.8.2 Directividad calculando Prad por integración de los campos en la apertura**

$$D = \frac{4\pi \int_{\phi=0}^{2\pi} \int_{\theta=0}^{\pi} (\sin^2 \phi + \cos^2 \phi \cos^2 \theta) |P_{y\_sec tE}|^2 \sin \theta d\theta d\phi}{\int_{x=-a/2}^{a/2} \int_{y=-H/2}^{H/2} |E_{ay\_sec tE}(x, y)|^2 dx dy} = \frac{4\pi \int_{\phi=0}^{2\pi} \int_{\theta=0}^{\pi} (\sin^2 \phi + \cos^2 \phi \cos^2 \theta) |P_{y\_sec tE}|^2 \sin \theta d\theta d\phi}{\lambda^2 \int_{x=-a/2}^{a/2} \int_{y=-H/2}^{H/2} |E_{ay\_sec tE}(x, y)|^2 dx dy}$$

**2.2.9 Eficiencia de spillover**

a) Siguiendo el primer principio de equivalencia

$$\mathcal{E}_{spillover} = \frac{\int_{\phi=0}^{2\pi} \int_{\theta=0}^{\theta_0} \frac{(1 + \cos \theta)^2}{4} |P_{y\_sec tE}|^2 \sin \theta d\theta d\phi}{\int_{\phi=0}^{2\pi} \int_{\theta=0}^{\pi} \frac{(1 + \cos \theta)^2}{4} |P_{y\_sec tE}|^2 \sin \theta d\theta d\phi}$$

b) Siguiendo el segundo principio de equivalencia

$$\mathcal{E}_{spillover} = \frac{\int_{\phi=0}^{2\pi} \int_{\theta=0}^{\theta_0} (\sin^2 \phi + \cos^2 \phi \cos^2 \theta) |P_{y\_sec tE}|^2 \sin \theta d\theta d\phi}{\int_{\phi=0}^{2\pi} \int_{\theta=0}^{\pi} (\sin^2 \phi + \cos^2 \phi \cos^2 \theta) |P_{y\_sec tE}|^2 \sin \theta d\theta d\phi}$$

Donde, recordemos que:

$$\theta_0 = \theta_{-XdB} , \text{ con X entre 10 y 12 en la práctica}$$

**2.2.10 Centro de fase**

La fórmula del centro de fase para un  $\phi$  genérico coincide para ambos principios de equivalencia y vale:

$$L_{ph} = \frac{\lambda}{2\pi(1 - \cos\theta)} \left\{ \arg\left[ P_{y\_sec tE}(\theta_{-XdB}, \phi) \right] - \arg\left[ P_{y0\_sec tE} \right] \right\}$$

Para el caso del **centro de fase medio**, coincide también para ambos principios de equivalencia:

$$L_{ph} = \frac{\lambda}{2\pi(1 - \cos\theta_{-XdB})} \cdot \frac{1}{2} \left\{ \arg\left[ P_{y\_sec tE}(\theta_{-XdB}, 0^\circ) \right] + \arg\left[ P_{y\_sec tE}(\theta_{-XdB}, 90^\circ) \right] - 2 \arg\left[ P_{y0\_sec tE} \right] \right\}$$

## 2.3 BOCINA PIRAMIDAL

### 2.3.1 Introducción

La bocina sectorial piramidal es el resultado de "abocinar" en una guía rectangular la dimensión "a", hasta convertirse en "W" contenida en el plano H de radiación (plano que contiene el vector **H**, es el plano horizontal) y la dimensión "b", hasta convertirse en "H" contenida en el plano E de radiación (plano que contiene el vector **E**, es el plano vertical). Para obtener la expresión del campo en la apertura, pondremos los ejes x,y, de forma análoga a como se hizo en la guía rectangular, añadiendo el z, tal y como se indica en la figura, en la cual se muestra en la foto (a) de la figura 2.3 la bocina piramidal con indicación de los parámetros geométricos a,b,W y H y en la foto (b) y foto (c) de la mismas figura, los cortes plano H ( $\phi=0^\circ$ ) y plano E ( $\phi=90^\circ$ ) respectivamente, con indicación de los parámetros geométricos anteriores además de  $R_1$ ,  $R_H$ ,  $l_H$ ,  $R_2$ ,  $R_E$  y  $l_E$ .

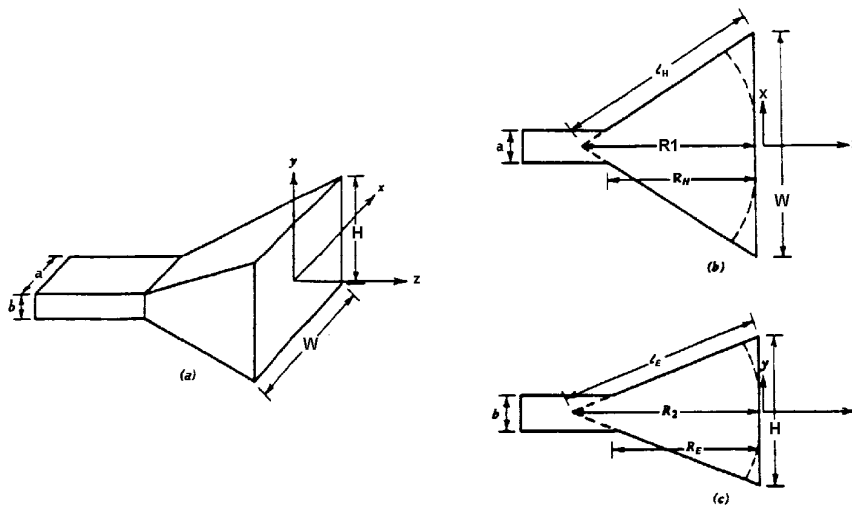


Fig. 2.3 Bocina sectorial piramidal

La bocina piramidal se considera una combinación de una bocina sectorial plano H y una bocina sectorial plano E. Los planos sectoriales H y E, mostrados en la figura anterior en (b) y (c) respectivamente, ponen de relieve la no existencia de un único vértice, hay dos, uno por cada plano, separados del centro de la apertura las distancias marcadas como  $R_1$  y  $R_2$  respectivamente. No obstante la distancias del centro de la apertura de la bocina al centro del plano de separación de la guía con la parte abocinada, lógicamente, deben coincidir en ambos planos, es decir:  $R_H = R_E$ . Las expresiones de  $R_H$ ,  $R_E$ ,  $l_H$ ,  $l_E$  son:

$$R_H = R_1 \left(1 - \frac{a}{W}\right)$$

$$R_E = R_2 \left(1 - \frac{b}{H}\right)$$

$$l_H = \sqrt{R_1^2 + \frac{W^2}{4}}$$

$$l_E = \sqrt{R_2^2 + \frac{H^2}{4}}$$

Si las paredes superior e inferior de parte abocinada se corrugan, tal y como se muestra en la figura, se tiene la **bocina corrugada piramidal**.

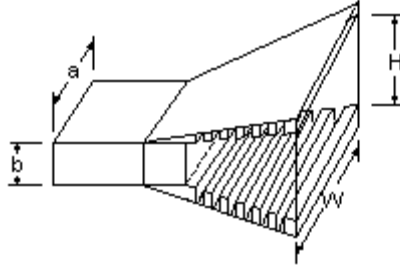


Fig. 2.4 Bocina corrugada piramidal

El objetivo de las corrugaciones es reducir las corrientes longitudinales sobre las paredes, que serán absorbidas por las corrugaciones, siempre que se diseñen de modo que su altura esté comprendida entre  $0.375 \lambda$  y  $\lambda/4$  (en la figura la altura de las corrugaciones se ha tomado como  $\lambda/4$ ), su ancho sea mucho menor que  $\lambda$  y su ancho más la separación con la corrugación contigua sea inferior a  $\lambda/4$ . Con ello, como luego veremos, se consigue una distribución de campo en la apertura con tapering también en el plano E, además de en el plano sectorial H (éste inducido por la forma de excitación de la guía), con el consiguiente aumento del plano E de la relación SLL, con respecto a la bocina piramidal normal. De este modo, en la bocina piramidal corrugada, la forma del diagrama sectorial plano E es como la del diagrama sectorial plano H (con la adecuada adaptación de las variables geométricas), o sea, el conocimiento de los diagramas de radiación plano H y plano E de la bocina piramidal corrugada, se reduce al del diagrama plano H de la bocina sectorial plano H.

### 2.3.2 Campo en la apertura. Parámetros $P_x(\theta, \phi)$ y $P_y(\theta, \phi)$

El campo en la apertura para la **bocina piramidal** (añadimos subíndice para remarcarlo), introduciendo la corrección de fase respecto a los frentes esféricos correspondientes a los planos H y E, suponiendo el campo polarizado linealmente según y, será:

$$\vec{E}_{a\_pir}(x) = E_0 \cos\left(\pi \frac{x}{W}\right) e^{-j[\delta(x)+\delta(y)]} \hat{y} = E_0 \cos\left(\pi \frac{x}{W}\right) e^{-j\frac{k_0}{2}\left(\frac{x^2}{R_1} + \frac{y^2}{R_2}\right)} \hat{y} = E_0 \cos\left(\pi \frac{x}{W}\right) e^{-j\frac{\pi}{\lambda}\left(\frac{x^2}{R_1} + \frac{y^2}{R_2}\right)} \hat{y}$$

El campo en la apertura para la **bocina piramidal corrugada** (añadimos subíndice para remarcarlo), suponiéndolo polarizado según y, será:

$$\vec{E}_{a\_cpir}(x) = E_0 \cos\left(\pi \frac{x}{W}\right) \cos\left(\pi \frac{y}{H}\right) e^{-j\frac{k_0}{2}\left(\frac{x^2}{R_1} + \frac{y^2}{R_2}\right)} \hat{y} = E_0 \cos\left(\pi \frac{x}{W}\right) \cos\left(\pi \frac{y}{H}\right) e^{-j\frac{\pi}{\lambda}\left(\frac{x^2}{R_1} + \frac{y^2}{R_2}\right)} \hat{y}$$

Como vemos, el campo en la apertura para las bocinas piramidales, debido a la restricción sobre la polarización, sólo tienen componentes  $y$ , denominándolas  $E_{ay\_pir}$ ,  $E_{ay\_cpir}$ , de acuerdo con lo anterior, se tiene que valen:

$$E_{ay\_pir}(x, y) = E_0 \cos\left(\pi \frac{x}{W}\right) e^{-j\frac{k_0}{2}\left(\frac{x^2}{R_1} + \frac{y^2}{R_2}\right)} = E_0 \cos\left(\pi \frac{x}{W}\right) e^{-j\frac{\pi}{\lambda}\left(\frac{x^2}{R_1} + \frac{y^2}{R_2}\right)}$$

$$E_{ay\_cpir}(x, y) = E_0 \cos\left(\pi \frac{x}{W}\right) \cos\left(\pi \frac{y}{H}\right) e^{-j\frac{k_0}{2}\left(\frac{x^2}{R_1} + \frac{y^2}{R_2}\right)} = E_0 \cos\left(\pi \frac{x}{W}\right) \cos\left(\pi \frac{y}{H}\right) e^{-j\frac{\pi}{\lambda}\left(\frac{x^2}{R_1} + \frac{y^2}{R_2}\right)}$$

De lo visto anteriormente, no tenemos componente en  $x$  por lo tanto La Potencia en  $x$  es 0:

$$P_{x\_pir/cpir} = 0$$

Sustituyendo el valor de **E<sub>ay</sub>**, teniendo en cuenta la forma de la superficie de la apertura de la bocina  $S_{ap}$ , la cual determina los siguientes límites de integración  $-W/2 < x < W/2$ ,  $H/2 < y < H/2$  y utilizando el subíndice identificativo del tipo de bocina, resulta:

$$P_{y\_pir}(\theta, \phi) = \int_{x=-W/2}^{W/2} \int_{y=-H/2}^{H/2} E_{ay\_pir}(x, y) e^{-jk(ux+vy)} dx dy = \int_{x=-W/2}^{W/2} \int_{y=-H/2}^{H/2} \cos\left(\pi \frac{x}{W}\right) e^{-j\frac{\pi}{\lambda}\left(\frac{x^2}{R_1} + \frac{y^2}{R_2}\right)} e^{-jk(ux+vy)} dx dy$$

$$P_{y\_cpir}(\theta, \phi) = \int_{x=-W/2}^{W/2} \int_{y=-H/2}^{H/2} E_{ay\_cpir}(x, y) e^{-jk(ux+vy)} dx dy = \int_{x=-W/2}^{W/2} \int_{y=-H/2}^{H/2} \cos\left(\pi \frac{x}{W}\right) \cos\left(\pi \frac{y}{H}\right) e^{-j\frac{\pi}{\lambda}\left(\frac{x^2}{R_1} + \frac{y^2}{R_2}\right)} e^{-jk(ux+vy)} dx dy$$

### 2.3.3 Componentes polares normalizadas de campo lejano: $e_\theta$ y $e_\phi$

a) Según el primer principio de equivalencia.

$$\begin{bmatrix} e_\theta \\ e_\phi \end{bmatrix} = \frac{1 + \cos\theta}{2} \begin{bmatrix} \text{sen}\phi P_{y\_pir/cpir} \\ \text{cos}\phi P_{y\_pir/cpir} \end{bmatrix}$$

b) Según el segundo principio de equivalencia.

$$\begin{bmatrix} e_\theta \\ e_\phi \end{bmatrix} = \begin{bmatrix} \text{sen}\phi P_{pir/cpir} \\ \text{cos}\phi \text{cos}\theta P_{pir/cpir} \end{bmatrix}$$

### 2.3.4 Componentes polares de campo lejano: $E_\theta$ y $E_\phi$

Recordemos que se obtienen a partir de las normalizadas  $e_\theta$  y  $e_\phi$  simplemente multiplicando por el siguiente factor que depende de la distancia  $r$  y de la longitud de onda:

$$\frac{je^{-jkr}}{\lambda r}$$

Es decir:

$$\begin{bmatrix} E_\theta \\ E_\phi \end{bmatrix} = \frac{je^{-jkr}}{\lambda r} \begin{bmatrix} e_\theta \\ e_\phi \end{bmatrix}$$

a) Según el primer principio de equivalencia.

$$\begin{bmatrix} E_\theta \\ E_\phi \end{bmatrix} = \frac{je^{-jkr}}{\lambda r} \frac{1 + \cos\theta}{2} \begin{bmatrix} \text{sen}\phi P_{y\_pir/cpir} \\ \text{cos}\phi P_{y\_pir/cpir} \end{bmatrix}$$

b) Según el primer principio de equivalencia.

$$\begin{bmatrix} E_\theta \\ E_\phi \end{bmatrix} = \frac{je^{-jkr}}{\lambda r} \begin{bmatrix} \text{sen}\phi P_{Y\_pir/cpir} \\ \text{cos}\phi \text{cos}\theta P_{Y\_pir/cpir} \end{bmatrix}$$

### 2.3.5 Diagrama normalizado de campo $f_E$

Para obtener el diagrama normalizado de campo debemos hallar el módulo del campo, el módulo del campo máximo y luego dividir estas dos cantidades.

a) Según el primer principio de equivalencia.

Módulo del campo:

$$|\vec{E}| = \frac{1}{\lambda r} \frac{1 + \cos\theta}{2} |P_{y\_pir/cpir}|$$

Módulo máximo del campo:

$$|\vec{E}|_{mx} = \frac{1}{\lambda r} |P_{y0\_pir/cpir}|$$

Función normalizada de campo:

$$|f_{E1}| = \frac{1 + \cos\theta}{2} \frac{|P_{y\_pir/cpir}|}{|P_{y0\_pir/cpir}|}$$

Función normalizada de campo, expresada en dB:

$$f_{E1}|_{dB} = 20 \log \left( \frac{1 + \cos \theta}{2} \frac{|P_{y\_pir/cpir}|}{|P_{y0\_pir/cpir}|} \right)$$

b) Según el segundo principio de equivalencia.

Módulo del campo:

$$|\vec{E}| = \frac{1}{\lambda r} |P_{y\_pir/cpir}| \sqrt{\text{sen}^2 \phi + \text{cos}^2 \phi \text{cos}^2 \theta}$$

Módulo máximo del campo:

$$|\vec{E}|_{mx} = \frac{1}{\lambda r} |P_{y0\_pir/cpir}|$$

Función normalizada de campo:

$$|f_{E2}| = \frac{|P_{y\_pir/cpir}|}{|P_{y0\_pir/cpir}|} \sqrt{\text{sen}^2 \phi + \text{cos}^2 \phi \text{cos}^2 \theta}$$

Función normalizada de campo, expresada en dB:

$$|f_{E2}|_{dB} = 20 \log \left( \frac{|P_{y\_pir/cpir}|}{|P_{y0\_pir/cpir}|} \sqrt{\text{sen}^2 \phi + \text{cos}^2 \phi \text{cos}^2 \theta} \right)$$

### 2.3.6 Componentes copolar y contrapolar del campo (según polariz. en y): $E_{CP}$ y $E_{XP}$

a) Según el primer principio de equivalencia.

$$\begin{bmatrix} E_{CP} \\ E_{XP} \end{bmatrix} = \frac{je^{-jkr}}{\lambda r} \frac{1 + \cos \theta}{2} \begin{bmatrix} P_{y\_pir/cpir} \\ 0 \end{bmatrix}$$

b) Según el primer principio de equivalencia.

$$\begin{bmatrix} E_{CP} \\ E_{XP} \end{bmatrix} = \frac{je^{-jkr}}{\lambda r} \begin{bmatrix} (\text{sen}^2 \phi + \text{cos}^2 \phi \text{cos} \theta) P_{y\_pir/cpir} \\ \text{sen} \phi \text{cos} \phi (1 - \text{cos} \theta) P_{y\_pir/cpir} \end{bmatrix}$$

### 2.3.7 Potencia radiada

#### 2.3.7.1 Potencia radiada obtenida integrando el diagrama de radiación

Para la bocina sectorial plano E, en que  $P_x=0$ ,

a) Según el primer principio de equivalencia

$$P_{rad} = \frac{1}{2Z_0\lambda^2} \int_{\phi=0}^{2\pi} \int_{\theta=0}^{\pi} \frac{(1 + \cos\theta)^2}{4} |P_{y\_pir/cpir}|^2 \sin\theta d\theta d\phi$$

b) Según el segundo principio de equivalencia

$$P_{rad} = \frac{1}{2Z_0\lambda^2} \int_{\phi=0}^{2\pi} \int_{\theta=0}^{\pi} (\sin^2\phi + \cos^2\phi \cos^2\theta) |P_{y\_pir/cpir}|^2 \sin\theta d\theta d\phi$$

#### 2.3.7.2 Potencia radiada obtenida integrando el campo en la apertura

$$P_{rad} = \frac{1}{2Z_0} \int_{x=-W/2}^{W/2} \int_{y=-H/2}^{H/2} |E_{ay\_pir/cpir}(x, y)|^2 dx dy$$

### 2.3.8 Directividad

En la bocina piramidal la expresión de la directividad vale:

$$D = \frac{\frac{1}{2Z_0} \frac{1}{\lambda^2} (|P_{y0\_pir/cpir}|^2)}{\frac{P_{rad}}{4\pi}}$$

Sustituyes la potencia radiada según el principio y el tipo de integración:

#### 2.3.8.1 Directividad calculando Prad por integración del diagrama

a) Según el primer principio de equivalencia

$$D = 4\pi \frac{(|P_{y0\_pir/cpir}|^2)}{\int_{\phi=0}^{2\pi} \int_{\theta=0}^{\pi} \frac{(1 + \cos\theta)^2}{4} |P_{y\_pir/cpir}|^2 \sin\theta d\theta d\phi} = 4\pi \frac{|\int_{x=-W/2}^{W/2} \int_{y=-H/2}^{H/2} E_{ay\_pir/cpir}(x, y) dx dy|^2}{\int_{\phi=0}^{2\pi} \int_{\theta=0}^{\pi} \frac{(1 + \cos\theta)^2}{4} |P_{y\_pir/cpir}|^2 \sin\theta d\theta d\phi}$$



b) Según el segundo principio de equivalencia

$$D = 4\pi \frac{(|P_{y0\_pir/cpir}|^2)}{\int_{\phi=0}^{2\pi} \int_{\theta=0}^{\pi} (\text{sen}^2 \phi + \cos^2 \phi \cos^2 \theta) |P_{y\_pir/cpir}|^2 \text{sen} \theta d\theta d\phi} = 4\pi \frac{|\int_{x=-W/2}^{W/2} \int_{y=-H/2}^{H/2} E_{ay\_pir/cpir}(x, y) dx dy|^2}{\int_{\phi=0}^{2\pi} \int_{\theta=0}^{\pi} (\text{sen}^2 \phi + \cos^2 \phi \cos^2 \theta) |P_{y\_pir/cpir}|^2 \text{sen} \theta d\theta d\phi}$$

### 2.3.8.2 Directividad calculando Prad por integración del diagrama

$$D = \frac{4\pi}{\lambda^2} \frac{(|P_{y0\_pir/cpir}|^2)}{\int_{x=-a/2}^{a/2} \int_{y=-H/2}^{H/2} |E_{ay\_pir/cpir}(x)|^2 dx dy} = \frac{4\pi}{\lambda^2} \frac{|\int_{x=-W/2}^{W/2} \int_{y=H/2}^{H/2} E_{ay\_pir/cpir}(x, y) dx dy|^2}{\int_{x=-a/2}^{a/2} \int_{y=-H/2}^{H/2} |E_{ay\_pir/cpir}(x, y)|^2 dx dy}$$

### 2.3.9 Eficiencia de spillover

a) Seguendo el primer principio de equivalencia

$$\mathcal{E}_{spillover} = \frac{\int_{\phi=0}^{2\pi} \int_{\theta=0}^{\theta_0} \frac{(1 + \cos \theta)^2}{4} |P_{y\_pir/cpir}|^2 \text{sen} \theta d\theta d\phi}{\int_{\phi=0}^{2\pi} \int_{\theta=0}^{\pi} \frac{(1 + \cos \theta)^2}{4} |P_{y\_pir/cpir}|^2 \text{sen} \theta d\theta d\phi}$$

b) Seguendo el primer principio de equivalencia

$$\mathcal{E}_{spillover} = \frac{\int_{\phi=0}^{2\pi} \int_{\theta=0}^{\theta_0} (\text{sen}^2 \phi + \cos^2 \phi \cos^2 \theta) |P_{y\_pir/cpir}|^2 \text{sen} \theta d\theta d\phi}{\int_{\phi=0}^{2\pi} \int_{\theta=0}^{\pi} (\text{sen}^2 \phi + \cos^2 \phi \cos^2 \theta) |P_{y\_pir/cpir}|^2 \text{sen} \theta d\theta d\phi}$$

Donde, recordemos que:

$$\theta_0 = \theta_{-XdB}, \quad \text{con X entre 10 y 12 en la práctica}$$

### 2.3.10 Centro de fase

La fórmula del centro de fase para un  $\phi$  genérico coincide para ambos principios de equivalencia y vale:

$$L_{ph} = \frac{\lambda}{2\pi(1 - \cos\theta)} \left\{ \arg\left[ P_{y\_pir/cpir}(\theta_{-XdB}, \phi) \right] - \arg\left[ P_{y0\_pir/cpir} \right] \right\}$$

Para el caso del **centro de fase medio**, coincide también para ambos principios de equivalencia:

$$L_{ph} = \frac{\lambda}{2\pi(1 - \cos\theta_{-XdB})} \cdot \frac{1}{2} \left\{ \arg\left[ P_{y\_pir/cpir}(\theta_{-XdB}, 0^\circ) \right] + \arg\left[ P_{y\_pir/cpir}(\theta_{-XdB}, 90^\circ) \right] - 2 \arg\left[ P_{y0\_pir/cpir} \right] \right\}$$

## 2.4 BOCINA CÓNICA

### 2.4.1 Introducción

Partiendo de una guía circular, si se agranda la boca siguiendo un cono de vértice situado en el eje tendremos una **bocina cónica**, como la mostrada en la figura, en la que se indican sus parámetros geométricos básicos:  $R_c$  (altura del cono),  $R_a$  (radio de la apertura),  $D_a$  (diámetro de la apertura),  $d_g$  (diámetro de la guía),  $L$  (distancia entre el plano que une la guía con la parte abocinada y el plano de la apertura de la bocina),  $R$  (distancia entre el vértice de la guía y un punto genérico de la apertura) y  $\alpha$  (ángulo de abocinamiento). Situando los ejes  $x$  e  $y$  sobre la apertura de la bocina en la forma indicada en la figura (el eje  $z$  se situará normal hacia nosotros en posteriores análisis); como también se muestra en la figura, un punto genérico de la apertura puede ser descrito además de por sus coordenadas cartesianas  $(x, y)$  por sus coordenadas polares  $(\rho, \phi)$

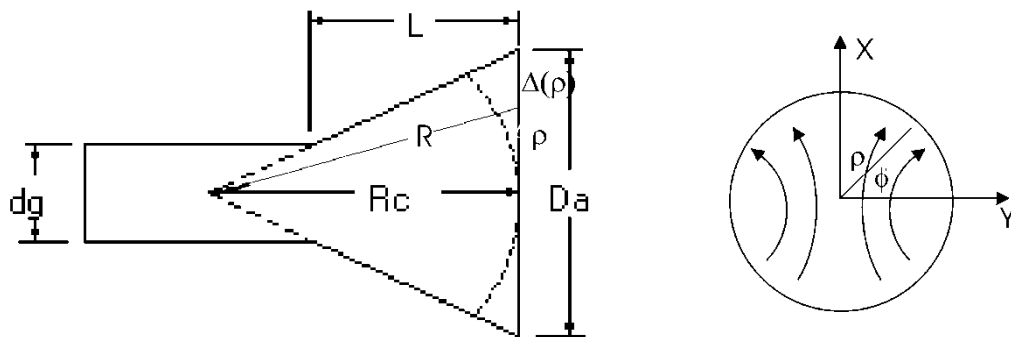


Fig. 2.5 Bocina cónica

Según la geometría mostrada, por triángulos semejantes se tiene que:

$$\frac{D_a / 2}{R_c} = \frac{D_a / 2 - d_g / 2}{L}$$

De donde:

$$L = R_c \left(1 - \frac{d_g}{D_a}\right)$$

Si la pared de la parte abocinada se corruga, tal y como se muestra en el corte de la bocina cónica de la figura, se tiene la **bocina corrugada cónica**

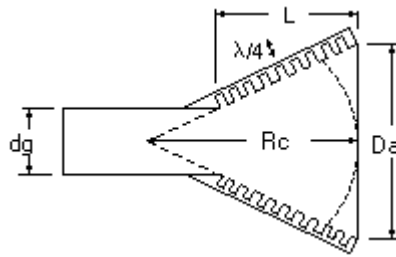


Fig. 2.6 Bocina corrugada cónica

El objetivo de las corrugaciones es reducir las corrientes longitudinales sobre las paredes, que serán absorbidas por las corrugaciones, siempre que se diseñen de modo que su altura esté comprendida  $0.375 \lambda$  y  $\lambda/4$  (en la figura la altura de las corrugaciones se ha tomado como  $\lambda/4$ ), su ancho sea mucho menor que  $\lambda$  y su ancho más la separación con la corrugación contigua sea inferior a  $\lambda/4$ . Con ello, como luego veremos, se consigue **modificar la distribución de campos**, pasándose del modo TE<sub>11</sub> al **modo híbrido HE<sub>11</sub>**, que se caracteriza por tener su amplitud, de forma rotacionalmente simétrica, decreciente del centro al borde (donde se anula) y tener líneas de campo rectas y paralelas en la dirección de polarización (como se muestra en la figura siguiente)

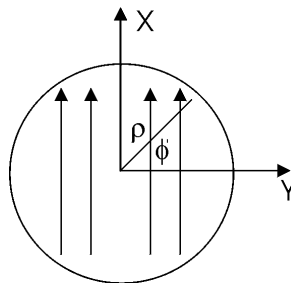


Fig. 2.7 Bocina corrugada cónica (2)

#### 2.4.2 Campo en la apertura. Parámetros $P_x(\theta, \phi)$ y $P_y(\theta, \phi)$

El campo en la apertura, para la bocina cónica (añadimos subíndice para remarcarlo), será análogo al de la guía circular; según el sistema de coordenadas de la figura, utilizando coordenadas cartesianas y polares, se podrá expresar de la siguiente forma:

$$\vec{E}_{a\_con} = E_{ax\_con}(\rho, \phi') \hat{x} + E_{ay\_con}(\rho, \phi') \hat{y}$$

Esta fórmula, debe escribirse sólo en coordenadas cartesianas para obtener Px y Py. Para ello, habría que aplicar las fórmulas de transformación de coordenadas polares a cartesianas sobre las variables polares:

$$\rho = \sqrt{x^2 + y^2}$$

$$\phi' = \arctg \frac{y}{x}$$

Con lo cual, la fórmula del campo en la apertura quedaría de la forma:

$$\vec{E}_{a\_con}(x, y) = E_{ax\_con}(x, y)\hat{x} + E_{ay\_con}(x, y)\hat{y}$$

Por simplicidad y comodidad en la escritura no se especificará la transformación de coordenadas polares a cartesianas. De acuerdo con ello, y en virtud de lo comentado sobre los campos en la apertura, en cuanto a que son como los del modo fundamental de la guía TE11 en la bocina cónica no corrugada y el modo híbrido HE11 en la bocina corrugada, tenemos que las componentes cartesianas del campo en la apertura para la bocina cónica lisa y corrugada tienen las siguientes expresiones:

**Bocina cónica lisa:**

$$E_{ax\_con}(x, y) = E_0 \frac{1'8412}{2R_a} J_2\left(\frac{1'8412\rho}{R_a}\right) \text{sen}(2\phi') e^{-j\frac{\pi\rho^2}{\lambda R_c}}$$

$$E_{ay\_con}(x, y) = E_0 \frac{1'8412}{2R_a} \left[ J_0\left(\frac{1'8412\rho}{R_a}\right) - J_2\left(\frac{1'8412\rho}{R_a}\right) \cos(2\phi') \right] e^{-j\frac{\pi\rho^2}{\lambda R_c}}$$

**Bocina cónica corrugada:**

$$E_{ax\_ccon}(x, y) = 0$$

$$E_{ay\_ccon}(x, y) = E_0 J_0\left(\frac{2'405\rho}{R_a}\right) e^{-j\frac{\pi\rho^2}{\lambda R_c}}$$

Donde  $J_0$  y  $J_2$  son las funciones de Bessel de orden 0 y orden 2 respectivamente.

Sustituyendo los valores **Eax**, **Eay** y teniendo en cuenta la forma de la superficie de la apertura de la bocina  $S_{ap}$ , la cual determina los siguientes límites de integración  $-a/2 < x < a/2$ ,  $H/2 < y < H/2$ ,  $P_x(\theta, \phi)$  y  $P_y(\theta, \phi)$  valen:

Para la **bocina cónica lisa**:

$$P_{x\_con}(\theta, \phi) = \int_{x=-Ra}^{Ra} \int_{y=-\sqrt{Ra^2-x^2}}^{\sqrt{Ra^2-x^2}} E_{ax\_con}(x, y) e^{-jk(ux+vy)} dx dy =$$

$$= \int_{x=-Ra}^{Ra} \int_{y=-\sqrt{Ra^2-x^2}}^{\sqrt{Ra^2-x^2}} \frac{1'8412}{2R_a} J_2\left(\frac{1'8412\rho}{R_a}\right) \text{sen}(2\phi') e^{-j\frac{\pi\rho^2}{\lambda R_c}} e^{-jk(ux+vy)} dx dy$$

$$P_{y\_con}(\theta, \phi) = \int_{x=-Ra}^{Ra} \int_{y=-\sqrt{Ra^2-x^2}}^{\sqrt{Ra^2-x^2}} E_{ay\_con}(x, y) e^{-jk(ux+vy)} dx dy =$$

$$= \int_{x=-Ra}^{Ra} \int_{y=-\sqrt{Ra^2-x^2}}^{\sqrt{Ra^2-x^2}} \frac{1'8412}{2R_a} \left[ J_0\left(\frac{1'8412\rho}{R_a}\right) - J_2\left(\frac{1'8412\rho}{R_a}\right) \cos(2\phi') \right] e^{-j\frac{\pi\rho^2}{\lambda R_c}} e^{-jk(ux+vy)} dx dy$$

Para la **bocina cónica corrugada**:

$$P_{x\_ccon}(\theta, \phi) = 0$$

$$P_{y\_ccon}(\theta, \phi) = \int_{x=-Ra}^{Ra} \int_{y=-\sqrt{Ra^2-x^2}}^{\sqrt{Ra^2-x^2}} E_{ay\_ccon}(x, y) e^{-jk(ux+vy)} dx dy =$$

$$= \int_{x=-Ra}^{Ra} \int_{y=-\sqrt{Ra^2-x^2}}^{\sqrt{Ra^2-x^2}} J_0\left(\frac{2'405\rho}{R_a}\right) e^{-j\frac{\pi x^2}{\lambda R_c}} e^{-jk(ux+vy)} dx dy$$

Donde, recordemos que para realizar la integración en coordenadas cartesianas hay que transformar las variables polares según las siguientes expresiones:

$$\rho = \sqrt{x^2 + y^2}$$

$$\phi' = \text{arctg} \frac{y}{x}$$

### 2.4.3 Componentes polares normalizadas de campo lejano: $e_\theta$ y $e_\phi$

a) Según el primer principio de equivalencia.

Para la **BOCINA CÓNICA LISA:**

$$\begin{bmatrix} e_\theta \\ e_\phi \end{bmatrix} = \frac{1 + \cos\theta}{2} \begin{bmatrix} \cos\phi P_{x\_con} + \text{sen}\phi P_{y\_con} \\ -\text{sen}\phi P_{x\_con} + \cos\phi P_{y\_con} \end{bmatrix}$$

Para la **BOCINA CÓNICA CORRUGADA (Px=0):**

$$\begin{bmatrix} e_\theta \\ e_\phi \end{bmatrix} = \frac{1 + \cos\theta}{2} \begin{bmatrix} \text{sen}\phi P_{y\_ccon} \\ \cos\phi P_{y\_ccon} \end{bmatrix}$$

b) Según el segundo principio de equivalencia

Para la **BOCINA CÓNICA LISA:**

$$\begin{bmatrix} e_\theta \\ e_\phi \end{bmatrix} = \begin{bmatrix} \cos\phi P_{x\_con} + \text{sen}\phi P_{y\_con} \\ -\text{sen}\phi \cos\theta P_{x\_con} + \cos\phi \cos\theta P_{y\_con} \end{bmatrix}$$

Para la **BOCINA CÓNICA CORRUGADA (Px=0):**

$$\begin{bmatrix} e_\theta \\ e_\phi \end{bmatrix} = \begin{bmatrix} \text{sen}\phi P_{y\_ccon} \\ + \cos\phi \cos\theta P_{y\_ccon} \end{bmatrix}$$

### 2.4.4 Componentes polares de campo lejano: $E_\theta$ y $E_\phi$

Recordemos que se obtienen a partir de las normalizadas  $e_\theta$  y  $e_\phi$  simplemente multiplicando por el siguiente factor que depende de la distancia  $r$  y de la longitud de onda:

$$\frac{je^{-jkr}}{\lambda r}$$

Es decir:

$$\begin{bmatrix} E_\theta \\ E_\phi \end{bmatrix} = \frac{je^{-jkr}}{\lambda r} \begin{bmatrix} e_\theta \\ e_\phi \end{bmatrix}$$

a) Según el **primer principio** de equivalencia.

Para la **BOCINA CÓNICA LISA**:

$$\begin{bmatrix} E_{\theta} \\ E_{\phi} \end{bmatrix} = \frac{je^{-jkr}}{\lambda r} \frac{1 + \cos\theta}{2} \begin{bmatrix} \cos\phi P_{x\_con} + \text{sen}\phi P_{y\_con} \\ -\text{sen}\phi P_{x\_con} + \cos\phi P_{y\_con} \end{bmatrix}$$

Para la **BOCINA CÓNICA CORRUGADA (Px=0)**:

$$\begin{bmatrix} E_{\theta} \\ E_{\phi} \end{bmatrix} = \frac{je^{-jkr}}{\lambda r} \frac{1 + \cos\theta}{2} \begin{bmatrix} \text{sen}\phi P_{y\_ccon} \\ \cos\phi P_{y\_ccon} \end{bmatrix}$$

b) Según el **segundo principio** de equivalencia

Para la **BOCINA CÓNICA LISA**:

$$\begin{bmatrix} E_{\theta} \\ E_{\phi} \end{bmatrix} = \frac{je^{-jkr}}{\lambda r} \begin{bmatrix} \cos\phi P_{x\_con} + \text{sen}\phi P_{y\_con} \\ -\text{sen}\phi \cos\theta P_{x\_con} + \cos\phi \cos\theta P_{y\_con} \end{bmatrix}$$

Para la **BOCINA CÓNICA CORRUGADA (Px=0)**:

$$\begin{bmatrix} E_{\theta} \\ E_{\phi} \end{bmatrix} = \frac{je^{-jkr}}{\lambda r} \begin{bmatrix} \text{sen}\phi P_{y\_ccon} \\ + \cos\phi \cos\theta P_{y\_ccon} \end{bmatrix}$$

#### 2.4.5 Diagrama normalizado de campo $f_E$

a) Según el **primer principio** de equivalencia.

Para la **BOCINA CÓNICA LISA**:

Módulo del campo:

$$|\vec{E}| = \frac{1}{\lambda r} \frac{1 + \cos\theta}{2} \sqrt{|\cos\phi P_{x\_con} + \text{sen}\phi P_{y\_con}|^2 + |-\text{sen}\phi P_{x\_con} + \cos\phi P_{y\_con}|^2}$$



Módulo máximo del campo:

$$|\vec{E}|_{mx} = \frac{1}{\lambda r} \sqrt{|P_{x0\_con}|^2 + |P_{y0\_con}|^2}$$

Función normalizada de campo:

$$f_{E1} = \frac{|\vec{E}|}{|\vec{E}|_{mx}} = \frac{1 + \cos\theta}{2} \frac{\sqrt{|\cos\phi P_{x\_con} + \text{sen}\phi P_{y\_con}|^2 + |-\text{sen}\phi P_{x\_con} + \cos\phi P_{y\_con}|^2}}{\sqrt{|P_{x0\_con}|^2 + |P_{y0\_con}|^2}}$$

Función normalizada de campo expresada en dB:

$$f_{E1}|_{dB} = 20 \log \frac{|\vec{E}|}{|\vec{E}|_{mx}} = 20 \log \left( \frac{1 + \cos\theta}{2} \frac{\sqrt{|\cos\phi P_{x\_con} + \text{sen}\phi P_{y\_con}|^2 + |-\text{sen}\phi P_{x\_con} + \cos\phi P_{y\_con}|^2}}{\sqrt{|P_{x0\_con}|^2 + |P_{y0\_con}|^2}} \right)$$

Para la **BOCINA CÓNICA CORRUGADA (Px=0)**:

Módulo del campo:

$$|\vec{E}| = \frac{1}{\lambda r} \frac{1 + \cos\theta}{2} |P_{y\_ccon}|$$

Módulo máximo del campo:

$$|\vec{E}|_{mx} = \frac{1}{\lambda r} |P_{y\_ccon}|$$

Función normalizada de campo:

$$|f_{E1}| = \frac{1 + \cos\theta}{2} \frac{|P_{y\_ccon}|}{|P_{y0\_ccon}|}$$

Función normalizada de campo, expresada en dB:

$$f_{E1}|_{dB} = 20 \log \left( \frac{1 + \cos\theta}{2} \frac{|P_{y\_ccon}|}{|P_{y0\_ccon}|} \right)$$

b) Según el **segundo principio** de equivalencia

Para la **BOCINA CÓNICA LISA**:

Módulo del campo:

$$|\vec{E}| = \frac{1}{\lambda r} \sqrt{|\cos\phi P_{x\_con} + \text{sen}\phi P_{y\_con}|^2 + |-\text{sen}\phi \cos\theta P_{x\_con} + \cos\phi \cos\theta P_{y\_con}|^2}$$

Módulo máximo del campo:

$$|\vec{E}|_{mx} = \frac{1}{\lambda r} \sqrt{|P_{x0\_con}|^2 + |P_{y0\_con}|^2}$$

Función normalizada de campo:

$$f_{E1} = \frac{|\vec{E}|}{|\vec{E}|_{mx}} = \frac{\sqrt{|\cos\phi P_{x\_con} + \text{sen}\phi P_{y\_con}|^2 + |-\text{sen}\phi \cos\theta P_{x\_con} + \cos\phi \cos\theta P_{y\_con}|^2}}{\sqrt{|P_{x0\_con}|^2 + |P_{y0\_con}|^2}}$$

Función normalizada de campo expresada en dB:

$$f_{E1}|_{dB} = 20 \log \frac{|\vec{E}|}{|\vec{E}|_{mx}} = 20 \log \left( \frac{\sqrt{|\cos\phi P_{x\_con} + \text{sen}\phi P_{y\_con}|^2 + |-\text{sen}\phi \cos\theta P_{x\_con} + \cos\phi \cos\theta P_{y\_con}|^2}}{\sqrt{|P_{x0\_con}|^2 + |P_{y0\_con}|^2}} \right)$$

Para la **BOCINA CÓNICA CORRUGADA (Px=0)**:

Módulo del campo:

$$|\vec{E}| = \frac{1}{\lambda r} |P_{y\_ccon}| \sqrt{\text{sen}^2\phi + \cos^2\phi \cos^2\theta}$$

Módulo máximo del campo:

$$|\vec{E}|_{mx} = \frac{1}{\lambda r} |P_{y0\_ccon}|$$

Función normalizada de campo:

$$|f_{E2}| = \frac{|P_{y\_ccon}|}{|P_{y0\_ccon}|} \sqrt{\text{sen}^2\phi + \cos^2\phi \cos^2\theta}$$

Función normalizada de campo, expresada en dB:

$$|f_{E2}|_{dB} = 20 \log \left( \frac{|P_{y\_ccon}|}{|P_{y0\_ccon}|} \sqrt{\text{sen}^2 \phi + \cos^2 \phi \cos^2 \theta} \right)$$

#### 2.4.6 Componentes copolar y contrapolar del campo (según polariz. en y): $E_{CP}$ y $E_{XP}$

a) Según el primer principio de equivalencia

Para la **BOCINA CÓNICA LISA**:

$$\begin{bmatrix} E_{CP} \\ E_{XP} \end{bmatrix} = \frac{je^{-jkr}}{\lambda r} \frac{1 + \cos \theta}{2} \begin{bmatrix} P_{y\_con} \\ P_{x\_con} \end{bmatrix}$$

Para la **BOCINA CÓNICA CORRUGADA ( $P_x=0$ )**:

$$\begin{bmatrix} E_{CP} \\ E_{XP} \end{bmatrix} = \frac{je^{-jkr}}{\lambda r} \frac{1 + \cos \theta}{2} \begin{bmatrix} P_{y\_ccon} \\ 0 \end{bmatrix}$$

b) Según el segundo principio de equivalencia.

Para la **BOCINA CÓNICA LISA**:

$$\begin{bmatrix} E_{CP} \\ E_{XP} \end{bmatrix} = \frac{je^{-jkr}}{\lambda r} \begin{bmatrix} \text{sen} \phi \cos \phi (1 - \cos \theta) P_{x\_con} + (\text{sen}^2 \phi + \cos^2 \phi \cos \theta) P_{y\_con} \\ (\text{sen}^2 \phi + \cos^2 \phi \cos \theta) P_{x\_con} + \text{sen} \phi \cos \phi (1 - \cos \theta) P_{y\_con} \end{bmatrix}$$

Para la **BOCINA CÓNICA CORRUGADA ( $P_x=0$ )**:

$$\begin{bmatrix} E_{CP} \\ E_{XP} \end{bmatrix} = \frac{je^{-jkr}}{\lambda r} \begin{bmatrix} (\text{sen}^2 \phi + \cos^2 \phi \cos \theta) P_{y\_ccon} \\ \text{sen} \phi \cos \phi (1 - \cos \theta) P_{y\_ccon} \end{bmatrix}$$

## 2.4.7 Potencia radiada

### 2.4.7.1 Potencia radiada obtenida integrando diagrama de radiación

a) Según el primer principio de equivalencia

Para la **BOCINA CÓNICA LISA**:

$$P_{rad} = \frac{1}{2Z_0\lambda^2} \int_{\phi=0}^{2\pi} \int_{\theta=0}^{\pi} \frac{(1 + \cos\theta)^2}{4} (|\cos\phi P_{x\_con} + \text{sen}\phi P_{y\_con}|^2 + |-\text{sen}\phi P_{x\_con} + \cos\phi P_{y\_con}|^2) \text{sen}\theta d\theta d\phi$$

Para la **BOCINA CÓNICA CORRUGADA (Px=0)**:

$$P_{rad} = \frac{1}{2Z_0\lambda^2} \int_{\phi=0}^{2\pi} \int_{\theta=0}^{\pi} \frac{(1 + \cos\theta)^2}{4} |P_{y\_ccon}|^2 \text{sen}\theta d\theta d\phi$$

b) Según el segundo principio de equivalencia.

Para la **BOCINA CÓNICA LISA**:

$$P_{rad} = \frac{1}{2Z_0\lambda^2} \int_{\phi=0}^{2\pi} \int_{\theta=0}^{\pi} (|\cos\phi P_{x\_con} + \text{sen}\phi P_{y\_con}|^2 + |-\text{sen}\phi \cos\theta P_{x\_con} + \cos\phi \cos\theta P_{y\_con}|^2) \text{sen}\theta d\theta d\phi$$

Para la **BOCINA CÓNICA CORRUGADA (Px=0)**:

$$P_{rad} = \frac{1}{2Z_0\lambda^2} \int_{\phi=0}^{2\pi} \int_{\theta=0}^{\pi} (\text{sen}^2\phi + \cos^2\phi \cos^2\theta) |P_{y\_ccon}|^2 \text{sen}\theta d\theta d\phi$$

### 2.4.7.2 Potencia radiada obtenida integrando el campo en la apertura

Para la **BOCINA CÓNICA LISA**:

$$P_{rad} = \frac{1}{2Z_0} \int_{x=-Ra}^{Ra} \int_{y=-\sqrt{Ra^2-x^2}}^{\sqrt{Ra^2-x^2}} (|E_{ax\_con}(x, y)|^2 + |E_{ay\_con}(x, y)|^2) dx dy$$

Para la **BOCINA CÓNICA CORRUGADA (Eax=0)**:

$$P_{rad} = \frac{1}{2Z_0} \int_{x=-Ra}^{Ra} \int_{y=-\sqrt{Ra^2-x^2}}^{\sqrt{Ra^2-x^2}} |E_{ay\_ccon}(x, y)|^2 dx dy$$

### 2.4.8 Directividad

Para la **BOCINA CÓNICA LISA**:

$$D = \frac{\frac{1}{2Z_0} \frac{1}{\lambda^2} (|P_{x0\_con}|^2 + |P_{y0\_con}|^2)}{\frac{P_{rad}}{4\pi}}$$

Para la **BOCINA CÓNICA CORRUGADA (Px=0)**:

$$D = \frac{\frac{1}{2Z_0} \frac{1}{\lambda^2} |P_{y0\_ccon}|^2}{\frac{P_{rad}}{4\pi}}$$

#### 2.4.8.1 Directividad calculando Prad por integración del diagrama

a) Según el primer principio de equivalencia

Para la **BOCINA CÓNICA LISA**:

$$D = 4\pi \frac{(|P_{x0\_con}|^2 + |P_{y0\_con}|^2)}{\int_{\phi=0}^{2\pi} \int_{\theta=0}^{\pi} \frac{(1 + \cos\theta)^2}{4} (|\cos\phi P_{x\_con} + \text{sen}\phi P_{y\_con}|^2 + |-\text{sen}\phi P_{x\_con} + \cos\phi P_{y\_con}|^2) \text{sen}\theta d\theta d\phi} =$$

$$= 4\pi \frac{|\int_{x=-Ra}^{Ra} \int_{y=-\sqrt{Ra^2-x^2}}^{\sqrt{Ra^2-x^2}} E_{ax\_con}(x, y) dx dy|^2 + |\int_{x=-Ra}^{Ra} \int_{y=-\sqrt{Ra^2-x^2}}^{\sqrt{Ra^2-x^2}} E_{ay\_con}(x, y) dx dy|^2}{\int_{\phi=0}^{2\pi} \int_{\theta=0}^{\pi} \frac{(1 + \cos\theta)^2}{4} (|\cos\phi P_{x\_con} + \text{sen}\phi P_{y\_con}|^2 + |-\text{sen}\phi P_{x\_con} + \cos\phi P_{y\_con}|^2) \text{sen}\theta d\theta d\phi}$$

Para la **BOCINA CÓNICA CORRUGADA (Px=0)**:

$$D = 4\pi \frac{|P_{y0\_ccon}|^2}{\int_{\phi=0}^{2\pi} \int_{\theta=0}^{\pi} \frac{(1 + \cos\theta)^2}{4} |P_{y\_ccon}|^2 \text{sen}\theta d\theta d\phi} = 4\pi \frac{|\int_{x=-Ra}^{Ra} \int_{y=-\sqrt{Ra^2-x^2}}^{\sqrt{Ra^2-x^2}} E_{ax\_con}(x, y) dx dy|^2}{\int_{\phi=0}^{2\pi} \int_{\theta=0}^{\pi} \frac{(1 + \cos\theta)^2}{4} |P_{y\_ccon}|^2 \text{sen}\theta d\theta d\phi}$$

b) Según el segundo principio de equivalencia

Para la **BOCINA CÓNICA LISA**:

$$D = \frac{4\pi \int_{\phi=0}^{2\pi} \int_{\theta=0}^{\pi} (|P_{x0\_con}|^2 + |P_{y0\_con}|^2) (|\cos\phi P_{x\_con} + \text{sen}\phi P_{y\_con}|^2 + |-\text{sen}\phi \cos\theta P_{x\_con} + \cos\phi \cos\theta P_{y\_con}|^2) \text{sen}\theta d\theta d\phi}{\int_{x=-Ra}^{Ra} \int_{y=-\sqrt{Ra^2-x^2}}^{\sqrt{Ra^2-x^2}} |E_{ax\_con}(x,y)|^2 dx dy + \int_{x=-Ra}^{Ra} \int_{y=-\sqrt{Ra^2-x^2}}^{\sqrt{Ra^2-x^2}} |E_{ay\_con}(x,y)|^2 dx dy}$$

$$= 4\pi \int_{\phi=0}^{2\pi} \int_{\theta=0}^{\pi} (|\cos\phi P_{x\_con} + \text{sen}\phi P_{y\_con}|^2 + |-\text{sen}\phi \cos\theta P_{x\_con} + \cos\phi \cos\theta P_{y\_con}|^2) \text{sen}\theta d\theta d\phi$$

Para la **BOCINA CÓNICA CORRUGADA (Px=0)**:

$$D = \frac{4\pi \int_{\phi=0}^{2\pi} \int_{\theta=0}^{\pi} (\text{sen}^2\phi + \cos^2\phi \cos^2\theta) |P_{y0\_con}|^2 \text{sen}\theta d\theta d\phi}{\int_{x=-Ra}^{Ra} \int_{y=-\sqrt{Ra^2-x^2}}^{\sqrt{Ra^2-x^2}} |E_{ax\_con}(x,y)|^2 dx dy}$$

$$= 4\pi \int_{\phi=0}^{2\pi} \int_{\theta=0}^{\pi} (\text{sen}^2\phi + \cos^2\phi \cos^2\theta) |P_{y\_con}|^2 \text{sen}\theta d\theta d\phi$$

### 2.4.8.2 Directividad calculando Prad por integración de los campos en la apertura

Para la **BOCINA CÓNICA LISA**:

$$D = \frac{4\pi}{\lambda^2} \frac{(|P_{x0\_con}|^2 + |P_{y0\_con}|^2)}{\int_{x=-Ra}^{Ra} \int_{y=-\sqrt{Ra^2-x^2}}^{\sqrt{Ra^2-x^2}} |E_{ax\_con}(x,y)|^2 dx dy + \int_{x=-Ra}^{Ra} \int_{y=-\sqrt{Ra^2-x^2}}^{\sqrt{Ra^2-x^2}} |E_{ay\_con}(x,y)|^2 dx dy}$$

$$= \frac{4\pi}{\lambda^2} \frac{|\int_{x=-Ra}^{Ra} \int_{y=-\sqrt{Ra^2-x^2}}^{\sqrt{Ra^2-x^2}} E_{ax\_con}(x,y) dx dy|^2 + |\int_{x=-Ra}^{Ra} \int_{y=-\sqrt{Ra^2-x^2}}^{\sqrt{Ra^2-x^2}} E_{ay\_con}(x,y) dx dy|^2}{\int_{x=-Ra}^{Ra} \int_{y=-\sqrt{Ra^2-x^2}}^{\sqrt{Ra^2-x^2}} |E_{ax\_con}(x,y)|^2 dx dy + \int_{x=-Ra}^{Ra} \int_{y=-\sqrt{Ra^2-x^2}}^{\sqrt{Ra^2-x^2}} |E_{ay\_con}(x,y)|^2 dx dy}$$

Para la **BOCINA CÓNICA CORRUGADA (Eax=Px=0)**:

$$D = \frac{4\pi}{\lambda^2} \frac{(|P_{y0\_pir/cpir}|^2)}{\int_{x=-Ra}^{Ra} \int_{y=-\sqrt{Ra^2-x^2}}^{\sqrt{Ra^2-x^2}} |E_{ay\_con}(x,y)|^2 dx dy} = \frac{4\pi}{\lambda^2} \frac{|\int_{x=-Ra}^{Ra} \int_{y=-\sqrt{Ra^2-x^2}}^{\sqrt{Ra^2-x^2}} E_{ay\_con}(x,y) dx dy|^2}{\int_{x=-Ra}^{Ra} \int_{y=-\sqrt{Ra^2-x^2}}^{\sqrt{Ra^2-x^2}} |E_{ay\_con}(x,y)|^2 dx dy}$$

#### 2.4.9 Eficiencia de spillover

En el caso de la bocina cónica, las fórmulas de eficiencia de spillover se transforman de la siguiente forma.

a) Según el primer principio de equivalencia

Para la **BOCINA CÓNICA LISA**:

$$D = \frac{\int_{\phi=0}^{2\pi} \int_{\theta=0}^{\theta_0} (|\cos\phi P_{x\_con} + \text{sen}\phi P_{y\_con}|^2 + |-\text{sen}\phi \cos\theta P_{x\_con} + \cos\phi \cos\theta P_{y\_con}|^2) \text{sen}\theta d\theta d\phi}{\int_{\phi=0}^{2\pi} \int_{\theta=0}^{\pi} (|\cos\phi P_{x\_con} + \text{sen}\phi P_{y\_con}|^2 + |-\text{sen}\phi \cos\theta P_{x\_con} + \cos\phi \cos\theta P_{y\_con}|^2) \text{sen}\theta d\theta d\phi}$$

Para la **BOCINA CÓNICA CORRUGADA (Px=0)**:

$$\mathcal{E}_{spillover} = \frac{\int_{\phi=0}^{2\pi} \int_{\theta=0}^{\theta_0} \frac{(1 + \cos\theta)^2}{4} |P_{y\_con}|^2 \text{sen}\theta d\theta d\phi}{\int_{\phi=0}^{2\pi} \int_{\theta=0}^{\pi} \frac{(1 + \cos\theta)^2}{4} |P_{y\_con}|^2 \text{sen}\theta d\theta d\phi}$$

b) Según el segundo principio de equivalencia

Para la **BOCINA CÓNICA LISA**:

$$\mathcal{E}_{spillover} = \frac{\int_{\phi=0}^{2\pi} \int_{\theta=0}^{\theta_0} (|\cos\phi P_{x\_con} + \text{sen}\phi P_{y\_con}|^2 + |-\text{sen}\phi \cos\theta P_{x\_con} + \cos\phi \cos\theta P_{y\_con}|^2) \text{sen}\theta d\theta d\phi}{\int_{\phi=0}^{2\pi} \int_{\theta=0}^{\pi} (|\cos\phi P_{x\_con} + \text{sen}\phi P_{y\_con}|^2 + |-\text{sen}\phi \cos\theta P_{x\_con} + \cos\phi \cos\theta P_{y\_con}|^2) \text{sen}\theta d\theta d\phi}$$

Para la **BOCINA CÓNICA CORRUGADA (Px=0)**:

$$\mathcal{E}_{spillover} = \frac{\int_{\phi=0}^{2\pi} \int_{\theta=0}^{\theta_0} (\text{sen}^2 \phi + \cos^2 \phi \cos^2 \theta) |P_{y\_ccon}|^2 \text{sen} \theta d\theta d\phi}{\int_{\phi=0}^{2\pi} \int_{\theta=0}^{\pi} (\text{sen}^2 \phi + \cos^2 \phi \cos^2 \theta) |P_{y\_ccon}|^2 \text{sen} \theta d\theta d\phi}$$

Donde, recordemos que:

$$\theta_0 = \theta_{-XdB} \quad , \text{ con X entre 10 y 12 en la práctica}$$

#### 2.4.10 Centro de fase

En el caso de las bocinas cónicas, las fórmulas del centro de fase se transforman de la siguiente forma.

a) Siguiendo el primer principio de equivalencia se transforma para la **BOCINA CÓNICA LISA / CORRUGADA** en:

$$L_{ph} = \frac{\lambda}{2\pi} \frac{\arg[P_{y\_con/ccon}(\theta_{-XdB}, \phi)] - \arg[P_{y0\_con/ccon}]}{(1 - \cos \theta_{-XdB})}$$

b) Siguiendo el segundo principio de equivalencia se transforma para la **BOCINA CÓNICA LISA** en:

$$L_{ph} = \frac{\lambda}{2\pi(1 - \cos \theta)} \left\{ \begin{aligned} &\arg[\text{sen} \phi \cos \phi (1 - \cos \theta_{-XdB}) P_{x\_con}(\theta_{-XdB}, \phi) + (\text{sen}^2 \phi + \cos^2 \phi \cos \theta_{-XdB}) P_{y\_con}(\theta_{-XdB}, \phi)] - \\ &- \arg[\text{sen} \phi \cos \phi (1 - \cos \theta_{-XdB}) P_{x0\_con} + (\text{sen}^2 \phi + \cos^2 \phi \cos \theta_{-XdB}) P_{y0\_con}] \end{aligned} \right\}$$

Para la **BOCINA CÓNICA CORRUGADA (Px=0)**: se obtiene igual expresión que con el principio de equivalencia, es decir (6.9.1)

$$L_{ph} = \frac{\lambda}{2\pi} \frac{\arg[P_{y\_con/ccon}(\theta_{-XdB}, \phi)] - \arg[P_{y0\_con/ccon}]}{(1 - \cos \theta_{-XdB})}$$



Recordemos que en el caso del **centro de fase medio**, se obtuvo, en general para cualquier bocina, igual expresión para ambos principios de equivalencia:

$$L_{ph} = \frac{\lambda}{2\pi(1 - \cos\theta_{-XdB})} \cdot \frac{1}{2} \left\{ \arg[P_{y\_con}(\theta_{-XdB}, 0^\circ)] + \arg[P_{y\_con}(\theta_{-XdB}, 90^\circ)] - 2 \arg[P_{y0\_con}] \right\}$$

# CAPÍTULO III: METODO DE INTEGRACIÓN GAUSS-LEGENDRE

---

## CAPÍTULO III: MÉTODO DE INTEGRACIÓN GAUSS-LEGENDRE

### 3.1 INTEGRALES SIMPLES

#### 3.1.1 Límites de integración entre -1 y 1

##### a) Fórmula de Gauss de 2 términos (2 parejas de parámetros)

El método Gauss-Legendre es un excelente método numérico para evaluar integrales definidas de funciones, por medio de sumatorios fáciles de implementar.

Sea una función  $f(t)$  que puede ser real o compleja la podemos integrar en un intervalo  $(-1,1)$  expresada del siguiente modo:

$$I = \int_{x=-1}^1 f(t) dt$$

Aproximándolo con el método de Gauss tenemos:

$$I = \int_{x=-1}^1 f(t) dx \approx \sum_{i=1}^n w_i f(t_i) + error$$

Siendo  $w_i$  los factores de ponderación y  $t_i$  los número de puntos, llamamos  $N$  al número total de parámetros y el número que va encima del sumatorio se calcula como  $n=N/2$ .

Si por ejemplo  $N$  valiese 4 tendríamos  $n=4/2= 2$  parejas de parámetros en el doble sumatorio, quedando la siguiente formula de la siguiente manera:

$$I = \int_{x=-1}^1 f(t) dt = \sum_{i=1}^2 w_i f(t_i) = w_1 f(t_1) + w_2 f(t_2)$$

Para determinar los 4 parámetros y sabiendo que  $f(t) = t^i$  con  $i=0,1,2,3$  se obtienen 4 ecuaciones que se pueden resolver igualando  $I$  a lo siguiente

$$I = \int_{x=-1}^1 t^k dx = \frac{2}{k+1}$$

Si  $k$  es par e  $I=0$  si  $k$  es impar.

El inconveniente que tiene es que para poder aplicarse requiere el conocimiento de la expresión de la función  $f$  a integrar que nos permita evaluarla en los puntos que requiere el método,

que serán diferentes según el número de términos elegido; cuanto mayor sea este número, se obtendrá mayor precisión en los resultados.

b) Polinomios de Legendre

El polinomio de Legendre de orden n se expresa de la siguiente manera

$$L_n(t) = \frac{1}{2^n n!} \frac{d^n}{dt^n} (t^2 - 1)^n$$

Si se desea obtener polinomios de orden mayor a 2 se aplicaría preferiblemente la siguiente ley de recurrencia:

$$L_n(t) = \frac{2n-1}{n} t \cdot L_{n-1}(t) - \frac{n-1}{n} L_{n-2}(t)$$

Los polinomios de Legendre son:

$$L_0(t) = 1$$

$$L_1(t) = t$$

$$L_2(t) = \frac{1}{2}(3t^2 - 1)$$

$$L_3(t) = \frac{t}{2}(5t^2 - 3)$$

$$L_4(t) = \frac{1}{8}(35t^4 - 30t^2 + 3)$$

$$L_5(t) = \frac{t}{8}(63t^4 - 70t^2 + 15)$$

$$L_6(t) = \frac{1}{16}(231t^6 - 315t^4 + 105t^2 - 5)$$

$$L_7(t) = \frac{t}{16}(429t^6 - 693t^4 + 315t^2 - 35)$$

$$L_8(t) = \frac{t}{128}(6435t^8 - 12012t^6 + 6930t^4 - 1260t^2 + 35)$$

c) Fórmulas de Gauss de más de 2 términos

Con más de dos términos la fórmula de Gauss no se resuelve con facilidad debido a la no linealidad con respecto a t. Para ello se emplea la teoría de los polinomios ortogonales. Se demuestra que los  $n=N/2$  que satisfacen las ecuaciones son las raíces del polinomio de Legendre de orden  $n=N/2$ .

Utilizando la técnica de Gauss arriba comentada se considera ahora en vez de funciones de la forma  $f(t)=tk$ , funciones de la forma  $f(t)=tk \text{Ln}(t)$ , donde  $\text{Ln}(t)$  es el polinomio de Legendre de orden  $n$ . La fórmula queda por tanto de la siguiente manera:

$$0 = \sum_{i=1}^n w_i \cdot f(t_i) = \sum_{i=1}^n w_i \cdot t_i^k L_n(t_i)$$

A continuación se muestran los valores de  $t_i$  y  $w_i$  para diferentes valores de  $n$  par.

<b>n</b>	<b>n/2</b>	<b><math>\pm t_i</math></b>	<b><math>w_i</math></b>
2	1	$\pm \frac{1}{\sqrt{3}}$	1
4	2	$\pm 0,33998\ 10435\ 84856$ $\pm 0,86113\ 63115\ 94053$	0,65214 51548 62546 0,34785 48451 37454
6	3	$\pm 0,23861\ 91860\ 83197$ $\pm 0,66120\ 93864\ 66265$ $\pm 0,93246\ 95142\ 03152$	0,46791 39345 72691 0,36076 15730 48139 0,17132 44923 79170
8	4	$\pm 0,18343\ 46424\ 95650$ $\pm 0,52553\ 24099\ 16239$ $\pm 0,79666\ 64774\ 13627$ $\pm 0,96028\ 98564\ 97536$	0,36268 37833 78362 0,31370 66458 77887 0,22238 10344 53374 0,10122 85362 90376
10	5	$\pm 0,14887\ 43389\ 81631$ $\pm 0,43339\ 53941\ 29247$ $\pm 0,67940\ 95682\ 99024$ $\pm 0,86506\ 33666\ 88985$ $\pm 9,97390\ 65285\ 17172$	0.29552 42247 14753 0,26926 67193 09996 0,21908 63625 15982 0,14945 13491 50581 0,06667 13443 98688

En cambio, para diferentes valores de n impar los valores de  $t_i$  y  $w_i$  son:

	<b>n</b>	<b>0</b> <b><math>\pm t_i</math></b>	<b><math>w_i</math></b>
3	1	0 $\pm \sqrt{\frac{3}{5}}$	8/9 5/9
5	2	0 $\pm 0,53846\ 93101\ 05683$ $\pm 0,90617\ 98459\ 38664$	0,56888 88888 88889 0,47862 86704 99366 0,23692 68850 56189
7	3	0 $\pm 0,40584\ 51513\ 77397$ $\pm 0,74153\ 11855\ 99394$ $\pm 0,94910\ 79123\ 42759$	0,41795 91836 73469 0,38183 00505 05119 0,27970 53914 89277 0,12948 49661 68870
9	4	0 $\pm 0,32425\ 34234\ 03890$ $\pm 0,61337\ 14327\ 00590$ $\pm 0,83603\ 11073\ 26636$ $\pm 0,96816\ 02395\ 07626$	0,33023 93550 01260 0,31234 70770 40003 0,26061 06964 02935 0,18064 81606 94857 0,08127 43883 61574
15	7	0 $\pm 0,20119\ 40939\ 97435$ $\pm 0,39415\ 13470\ 77563$ $\pm 0,57097\ 21726\ 08539$ $\pm 0,72441\ 77313\ 60170$ $\pm 0,84820\ 65834\ 10427$ $\pm 0,93727\ 33924\ 00706$ $\pm 0,98799\ 25180\ 20485$	0,20257 82419 25561 0,19843 14853 27111 0,18616 10001 15562 0,16626 92058 16994 0,13957 06779 26154 0,10715 92204 67172 0,07036 60474 88108 0,03075 32419 96117

### 3.1.2 Límites de integración entre a y b

Si por el contrario deseamos aplicar la integral en un intervalo (a,b) se debe realizar un cambio de variable. Como en la anterior ocasión, la integral viene expresada por:

$$I = \int_{x=a}^b f(x)dx$$

Para hacer el cambio de variable variamos la expresión de integración, en este caso x, entre -1 y 1 del siguiente modo:

$$\begin{aligned} x &= p \cdot t + q = \\ &= \frac{b-a}{2} \cdot t + \frac{b+a}{2} \end{aligned}$$

Siendo p un coeficiente que muestra la semidiferencia de los límites de integración superior e inferior y el término q es la semisuma de dichos límites, es decir:

Aproximando la integral con la fórmula de Gauss, tenemos que:

$$I = p \int_{t=-1}^1 f(pt + q)dt \approx p \sum_{i=1}^n w_i f(pt_i + q) + E_n$$

### 3.2 INTEGRALES DOBLES

Sea f(x,y) una función real de 2 variables, la integral doble extendida al recinto S(x,y) se puede escribir del siguiente modo, siguiendo la notación de Leibniz:

$$I_T = \iint_S f(x, y)dydx = \int_{x=x_a}^{x_b} \int_{y=y_b(x)}^{y_b(x)} f(x, y)dydx$$

Primeramente aplicamos el método numérico de integración de Gauss-Legendre en la primera integral respecto a y, siendo x constante:

$$I(x) = \int_{y=y_a(x)}^{y_b(x)} f(x, y)dy$$

Aplicamos el método entre -1 y 1. Para ello, se ha de realizar un cambio de variable de la forma  $y=ct+d$ , donde el coeficiente  $c$  es la semidiferencia de los límites de integración superior e inferior y el término  $d$  es la semisuma de dichos límites (en este caso serán funciones de  $x$ ), es decir:

$$y = c(x) \cdot t + d(x) = \frac{y_b(x) - y_a(x)}{2} \cdot t + \frac{y_b(x) + y_a(x)}{2}$$

Tras el cambio de variable señalado la primera integral se transforma en:

$$I(x) = c(x) \int_{t=-1}^1 f[x, c(x)t + d(x)] dt$$

Resolviendo numéricamente la integral se tiene que:

$$I(x) = c(x) \int_{t=-1}^1 f[x, c(x)t + d(x)] dt = c(x) \sum_{j=1}^n w_j \cdot f[x, c(x)t_j + d(x)] + E_{ny}$$

Ahora aplicamos el método numérico de integración de Gauss-Legendre en la segunda integración de la integral doble, respecto a  $x$ . El resultado de esta segunda integración será la integral buscada  $I_T$ , es decir:

$$I_T = \int_{x=x_a}^{x_b} I(x) dx$$

Como en la primera integral, hay que efectuar nuevamente un cambio de variable para que los límites de integración sea -1 y 1 de la forma  $x=pt+q$ , donde  $p$  es la semidiferencia de los límites de integración superior e inferior y el término  $q$  es la semisuma de dichos límites, es decir:

$$x = p \cdot t + q = \frac{x_b - x_a}{2} \cdot t + \frac{x_b + x_a}{2}$$

Tras el cambio de variable se transforma en:

$$I_T = p \int_{t=-1}^1 I(pt + q) dt$$

Aproximando la integral con la fórmula de Gauss, tenemos que:

$$I_T = p \int_{t=-1}^1 I(pt + q) dt = p \sum_{i=1}^n w_i I(pt_i + q) + E_{nx}$$



Una vez sustituido y simplificado, la fórmula final queda de la siguiente forma:

$$I_T = p \sum_{i=1}^n w_i c_i \sum_{j=1}^n w_j f(xg_i, yg_{ij}) + E_{ny} + E_{nx}$$

### 3.3 OBTENCIÓN DE LAS RAÍCES Y PESOS DEL POLINOMIO DE LEGENDRE

Las raíces ( $t_i$ ) y los pesos ( $w_i$ ) pueden ser obtenidos a partir de tablas de algún libro o a través de algún programa de ordenador que calcule  $t_i$  y  $w_i$  para un orden  $n$  dado. Preferiblemente resulta ventajoso utilizar un programa de ordenador para evitar introducir los valores manualmente. A continuación, se expone la descripción, entradas y salidas de las dos funciones en Matlab que calcula las raíces y pesos del polinomio de Legendre a partir de un orden  $n$  dado, y que utilizaremos en nuestras bocinas rectangulares y cónicas:

```
function [x,z,w1,w2]=lgwtRectangular2(N,a,b,c,d)
%DESCRIPCION:
% Función que permite calcular los puntos y pesos utilizando
%la cuadratura de Gauss-Legendre en el intervalo [a,b] en bocinas
rectangulares.
%PARAMETROS
%   ENTRADA:% Se utiliza ancho y altura en la piramidal, a y altura en la
sectorialE y b y ancho en la sectorialH
%       N: Número de puntos
%   ancho: anchura de la Apertura.
%   altura: Altura de la Apertura.%
%   SALIDA:
%       x2: puntos en coordenada x
%       y2: puntos en coordenada y
%       w1: pesos en x
%       w2: pesos en y
function [x,w]=lgwtCircular(N,a,b)
%DESCRIPCION:
% Función que permite calcular los puntos y pesos utilizando
%la cuadratura de Gauss-Legendre en el intervalo [a,b] en bocinas cónicas.
%PARAMETROS
%   ENTRADA:
%       N: Número de puntos
%   SALIDA:
%       x: raiz
%       w: peso
```

# CAPÍTULO IV: METODOLOGÍA DE PROGRAMACIÓN ORIENTADA A OBJETOS

---

## **CAPÍTULO IV: METODOLOGÍA DE PROGRAMACIÓN ORIENTADA A OBJETOS**

Este capítulo se va a centrar en la parte teórica de la programación orientada a objetos para poder aplicarlo en la aplicación de simulación "SABOR" y su estructura de software.

Hoy en día, no está muy extendida la idea de desarrollar las aplicaciones en lenguaje de programación orientada a objetos en MATLAB pese a sus múltiples beneficios. Antes de la versión 5.0 no se permitía desarrollar aplicaciones programando de esta manera.

La ventaja más importante de la programación orientada a objetos es su similitud a la forma de pensar de los humanos en unas entidades que tienen un estado, un comportamiento y una identidad, es por tanto, una metodología más natural.

### **4.1 INTRODUCCIÓN A LA PROGRAMACIÓN ORIENTADA A OBJETOS**

La programación orientada a objetos surge en la historia para evitar las complejidades que posee el software. Tradicionalmente, la forma de enfrentarse a esta complejidad ha sido empleando lo que llamamos programación estructurada, que consiste en descomponer el problema objeto de resolución en subproblemas y más subproblemas hasta llegar a acciones muy simples.

La programación orientada a objetos es otra forma de descomponer problemas. Para este caso, la descomposición es en objetos. Se trata de fijar no solo en lo que hay que hacer en el problema, sino en cuál es el escenario real del mismo, e intentar simularlo en nuestro programa.

Los lenguajes de programación tradicionales no orientados a objetos, como C o Pascal, basan su funcionamiento en el concepto de procedimiento o función. Una función es un conjunto de instrucciones que operan sobre unos argumentos y producen un resultado.

En el caso de los lenguajes orientados a objetos, como es el caso de C++ y Java, el elemento básico no es la función, sino un ente denominado objeto que a lo largo del proyecto se irá explicando su significado.

#### **4.1.1 Origen**

Podemos considerar como inicio de la Programación Orientada a Objetos(Object Oriented Programming), en la mitad de los años 60 en el Centro de Computación Noruego(The Norwegian Computing Center), surgió el lenguaje de programación Simula, desarrollado por Kristen Nygaard y Ole-Johan Dahl. Simula se definió como un lenguaje de programación orientado a la simulación de procesos, con el que se podían definir distintos tipos de actividades. En este lenguaje aparece por primera vez los conceptos de clases y objetos.

A partir de entonces casi todos los lenguajes de programación modernos comenzaron a utilizar los principios de orientación a objetos.

#### 4.1.2 Conceptos en la programación orientada a objetos (POO)

La programación orientada a objetos introduce nuevos conceptos. Entre ellos destacan los siguientes:

- Objeto: Un objeto es un conjunto de variables y métodos (funciones) relacionados entre sí.
- Clase: Define las variables y los métodos que son comunes para todos los objetos de un cierto tipo.
- Método: Acción que puede realizar un objeto.
- Evento: Un evento es una acción que ocurre en un momento determinado del tiempo, ya sea por acción del usuario, o porque la propia PC lo genera.
- Mensaje: Los objetos se comunican e interaccionan entre sí por medio de mensajes.
- Propiedad o atributo: Valores o características de los objetos. Permiten definir el estado del objeto u otras cualidades.
- Componentes de un objeto: Atributos, identidad, relaciones y métodos.

#### 4.1.3 Características de la POO

Las características principales de la programación orientada a objetos son las más importantes:

- Abstracción: La abstracción es una propiedad que permite en programación tomar información relevante, y descartar aquellos detalles que no son importantes en el ámbito del problema.
- Encapsulamiento: Significa reunir a todos los elementos que pueden pertenecer a una misma entidad, al mismo nivel de abstracción.
- Principio de ocultación: Hace referencia a que los atributos privados de un objeto no pueden ser modificados ni obtenidos a no ser que se haga a través del paso del mensaje (invocación a métodos, ya sean estas funciones o procedimientos).
- Polimorfismo: Es la capacidad que tienen los objetos de diferentes clases de responder a mensajes con el mismo nombre.
- Herencia: Es la relación entre una clase general y otra clase más específica. Cuando un objeto hereda más de una clase se dice que hay herencia múltiple; esta característica no está soportada por algunos lenguajes, por ejemplo, Java.

## 4.2 PROGRAMACIÓN ORIENTADA A OBJETOS EN MATLAB

En esta sección se hará una breve introducción a la programación orientada a objetos en MATLAB. Se explorará con algunos ejemplos los pasos para crear clases, constructores, métodos, herencia y la creación de objetos a partir de las clases. Este apartado es importante para poder entender el software desarrollado para crear la aplicación "SABOR".

### 4.2.1 Jerarquía de clases en MATLAB

En la siguiente figura se muestran árbol con los quince tipos de datos básicos definidos en Matlab que son los siguientes:

- double (double-precision floating-point number array)
- single (single-precision floating-point number array),
- char (character array),
- logical (array de valores verdadero y falso),
- int8 y uint8 (8-bit signed integer array, 8-bit unsigned integer array)
- int16 y uint16 (16-bit signed integer array, 16-bit unsigned integer array)
- int32 y uint32 (32-bit signed integer array, 32-bit unsigned integer array)
- int64 y uint64 (64-bit signed integer array, 64-bit unsigned integer array),
- cell (cell array)
- struct (struct array)
- function\_handle (array de valores para hacer llamadas a funciones)

A partir de ellos se pueden crear nuevos tipos de datos. Se ha remarcado user classes ya que son las clases creadas por los usuarios que heredan de la clase structure ya que ese es el punto de la jerarquía de clases en el que se insertan.

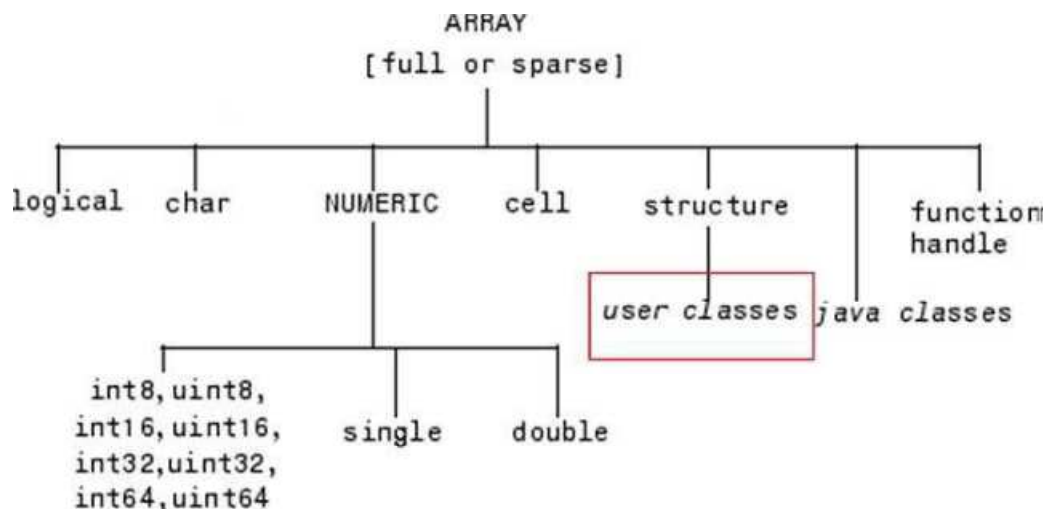


Fig 4.1 Jerarquía de clases en Matlab.

En el propio directorio de Matlab podemos encontrar algunos ejemplos de programación orientada a objetos. En la siguiente figura podemos ver el contenido del directorio {dirección donde tengas MATLAB}\R2009b\toolbox\control y pinchamos en control de nuevo y vemos cómo se organiza el código del toolbox control:

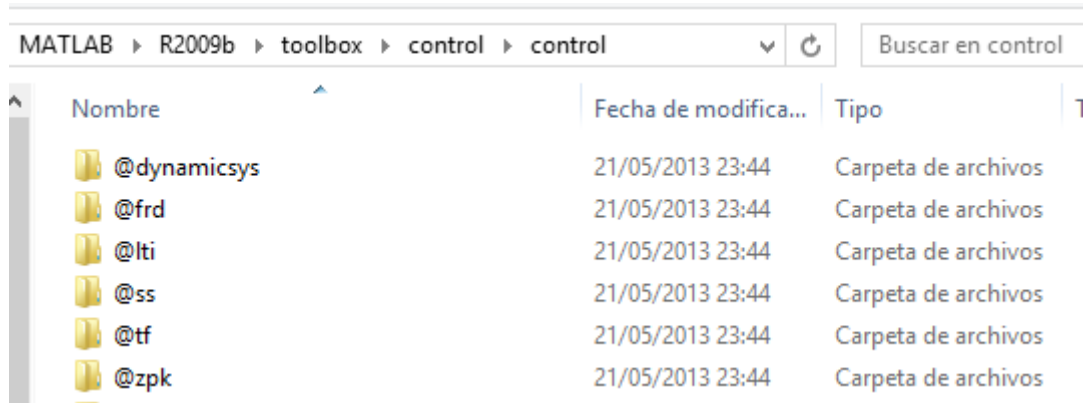


Fig 4.2 Contenido de control toolbox.

Como podemos apreciar en la figura, el directorio contiene una serie de carpetas cuyo nombre empieza por arroba. Eso significa que el directorio contiene la definición de una clase. Elegimos por ejemplo @lti, si mostramos el contenido tenemos un conjunto de ficheros.m:

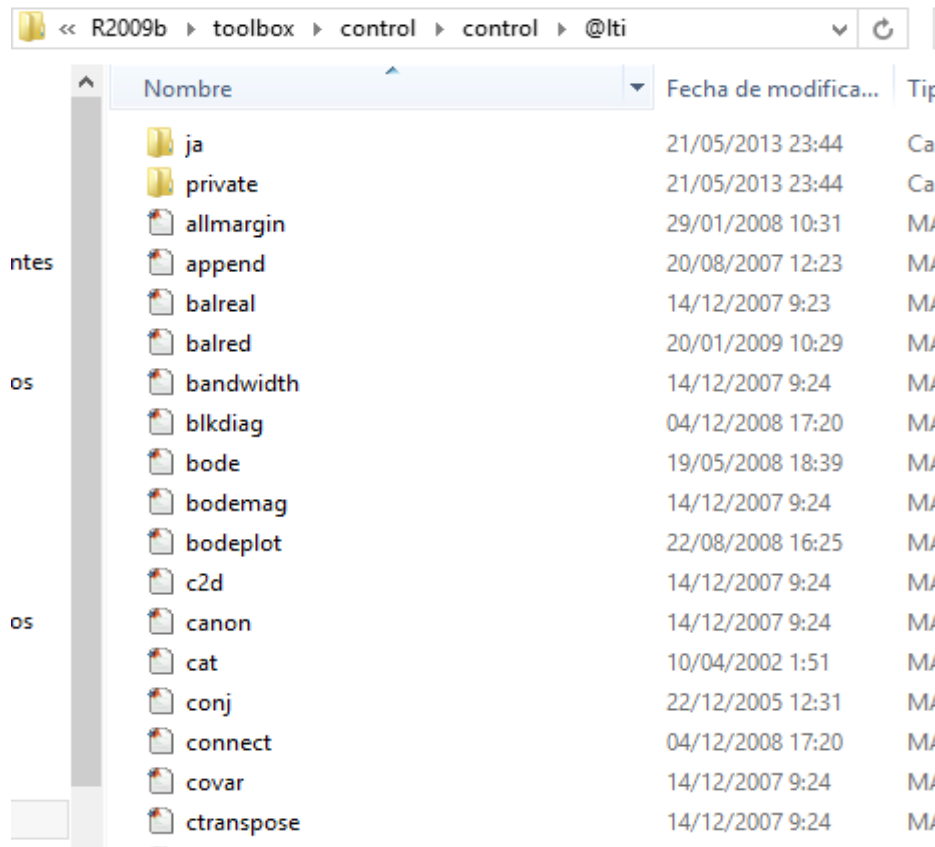


Fig 4.3 Contenido directorio @lti.

Extraemos la información de lti.m que corresponde al constructor de la clase lti para ver cómo se crea el objeto

```
function sys = lti(Ny,Nu)
if nargin==0,
    Ny = 0; Nu = 0; % for signature call by LOADOBJ
end
sys = struct(...
'Data',[]);

% Create parnet object
Parent = dynamicsys(Ny,Nu);

% Construccion del objeto
sys = class(sys,'lti',Parent);
```

Si consultamos la documentación de Matlab, podemos comprobar que LTI se ha implementado con metodología orientada a objetos.

```
>> help lti
LTI Constructor for the LTI parent object.

SYS = LTI(NY,NU) creates a LTI object with NY outputs
and NU inputs.
```

Figura 4.4 Ayuda clase LTI

### 4.2.2 Definición de clases en MATLAB

Para definir una clase se necesita crear un directorio que empiece por una arroba (@), seguida del nombre de la clase, quedando de la siguiente manera @nombreclase. En el ejemplo de la Figura 4.5, creamos la clase MICLASE.

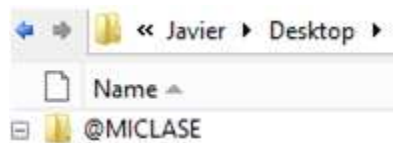


Fig 4.5 Creación de la clase MICLASE

#### 4.2.2.1 Crear el constructor de la clase

Pasamos a crear el constructor de la clase, que servirá para definir la estructura de datos del objeto (propiedades) e inicializarlo. Dentro del directorio anterior creamos un archivo llamado igual que la clase de extensión “.m” como se muestra en la Figura 4.6:

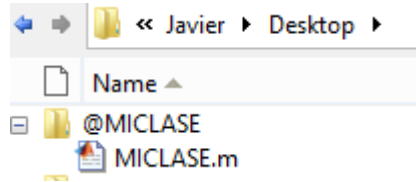


Fig 4.6 Creación del constructor de la clase MICLEASE

A continuación mostraremos el constructor de la clase MICLEASE:

```
function c = MICLEASE(argumento1, argumento2)
%DESCRIPCION:
% Constructor de la clase MICLEASE.
% PROPIEDADES
% propiedad1:
% propiedad2:
% SALIDA:
% c: Objeto de tipo MICLEASE
% MICLEASE= MICLEASE(argumento1,argumento2)
%Creamos la estructura de datos del objeto
c.propiedad1=argumento1;
c.propiedad2=argumento2;
%Devolvemos un objeto de tipo MICLEASE
c = class(c, 'MICLEASE');
```

Es importante resaltar que el constructor puede estar sobrecargado, es decir, es posible redefinir las reglas internas de una operación o función. En otras palabras, se puede crear un constructor que admita distinto número y tipos de parámetros de entrada.

#### 4.2.2.2 Añadir métodos a las clases

Para añadir un nuevo método a una clase, se crea en el directorio un archivo.m con el nombre del método. Para acceder dentro del método a los datos del objeto, es obligatorio pasar como parámetro de entrada el objeto sobre el que queremos ejecutar el método. Si el objeto es modificado dentro del método, hay que devolverlo como parámetro de salida. En la siguiente figura se muestra la localización del metodo1 y del metodo2 de la clase MICLEASE.



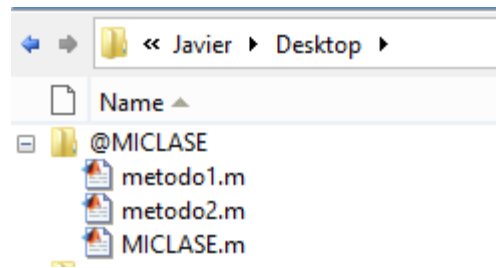


Fig4.7 Creación de un método de la clase MICLESE

A continuación se van a mostrar dos ejemplos, el método1 no va a modificar el objeto, pero en cambio el metodo2 sí que lo va a cambiar. El método1 tiene la siguiente función:

```
function [ salida ] = metodo1( miclase )
%METODO1
% DESCRIPCION
% Método que calcula la relación en porcentaje entre las propiedades:
propiedad1 ypropiedad2delobjetodetipoMICLESE
%PARAMETROS
% ENTRADA:
% (1)miclase: objeto de tipo MICLESE
% SALIDA:
% salida: parametro de salida
salida = (miclase.propiedad1/miclase.propiedad2)*100;
```

A continuación mostraremos el método método2 de la clase MICLESE, que modifica el objeto:

```
function [ objeto_modificado ] = metodo2( miclase )
%METODO1
% DESCRIPCION
% Método que incrementa el valor de las propiedades: propiedad1 y
propiedad2
%del objeto de tipo MICLESE en un 10 % y devuelve la relación entre lo
%PARAMETROS
% ENTRADA:
% (1)miclase: relación entre propiedad1 y propiedad2
% SALIDA:
% (2)objeto_modificado: objeto de tipo MICLESE, que se devuelve al haberse
modificado
%el valor de alguna de sus propiedades
miclase.propiedad1=miclase.propiedad1*1,1;
miclase.propiedad2=miclase.propiedad2*1,1;
objeto_modificado=miclase
```

Así mismo, podemos representar la clase MICLASE utilizando notación UML (Figura 4.8).

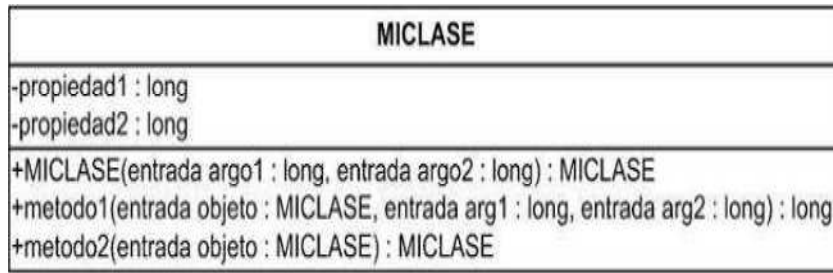


Fig 4.8 Representación UML de la clase MICLASE

Con la notación UML se puede ver de una manera gráfica cómo se ha creado la clase MICLASE. Se han definido 2 propiedades: *propiedad1* y *propiedad2* de tipo long, y la clase MICLASE posee 2 métodos: metodo1 y metodo2.

#### 4.2.2.3 Implementación de la herencia en MATLAB

La clase hija puede heredar la estructura de datos y los métodos del padre; de esta manera, se puede aprovechar la funcionalidad de la clase padre y a partir de ella, ampliarla o modificarla. Una clase hija puede heredar de una clase padre (herencia simple) o de varias (herencia múltiple).

Para ilustrar el concepto de herencia en Matlab, vamos a mostrar los pasos para crear la clase CLASEHIJA, que hereda de la clase MICLASE. En la clase CLASEHIJA se va a introducir el constructor de la clase CLASESHIJA, una propiedad llamada propiedad3 y un método llamado metodo3. Al igual que hicimos con la clase MICLASE, vamos a utilizar el diagrama de clases para facilitar la comprensión.

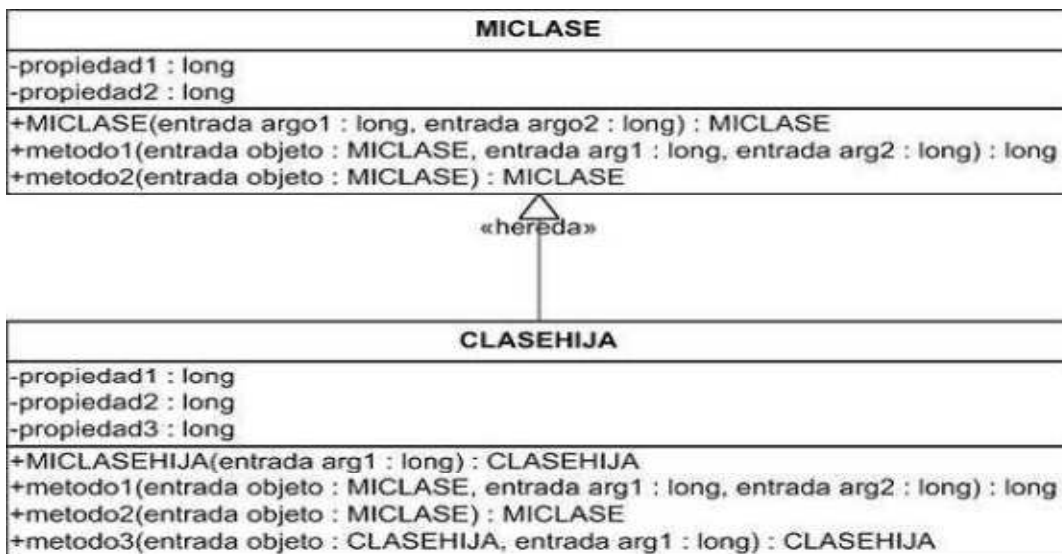


Fig 4.9 Representación UML de la clase MICLASE (2)

En la siguiente figura, se puede ver el contenido del directorio de la clase @CLASEHIJA, que está formado por dos ficheros: CLASEHIJA.m (el constructor) y metodo3.m (un método de la clase CLASEHIJA). A continuación, se ha proporcionado un listado de ambos archivos

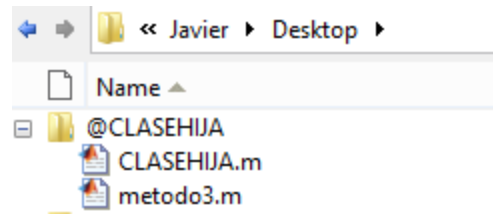


Fig 4.10 Constructor de la clase CLASEHIJA

El código correspondiente al contenido del constructor CLASEHIJA que hereda de la clase MICLASE es:

```
function [ hija ] = CLASEHIJA( input1 )

% DESCRIPCION
% Constructor de la clase CLASEHIJA que hereda de la clase MICLASE
%PARAMETROS
% ENTRADA:
% input1
% SALIDA:
% hija: objeto de clase CLASEHIJA
    hija.propiedad3=input1;
    c=MICLASE(50,78); %Llamamos al constructor de la clase padre

%childObj = class(childObj, 'childClass', parentObj)
hija = class(hija,'CLASEHIJA',c);
```

En el fichero CLASEHIJA.m debemos destacar que la última línea de código es la que implementa la herencia:

```
hija=class(objeto_hija, 'nombre_clase_hija', objeto_padre)
```

Donde objeto\_padre es una instancia de la clase padre, nombre\_clase\_hija es el nombre de la clase que hereda y objeto\_hija es una estructura que contiene las propiedades que no tiene la clase padre.

A continuación mostraremos el método método2 de la clase CLASEHIJA

```
function [ salida ] = metodo3( clasehija )
%METODO1
% DESCRIPCION
% Método que calcula la relación en porcentaje entre las propiedades:
propiedad1 ypropiedad3delobjetodetipoCLASEHIJA
%PARAMETROS
% ENTRADA:
% (1)clasehija: objeto de tipo CLASEHIJA
% SALIDA:
% salida: parametro de salida
salida = (miclase.propiedad1/miclase.propiedad3)*100;
```

No se han definido de nuevo los métodos de la clase padre ya que se han heredado. Si bien, existiría la posibilidad de modificarlos dentro de la clase que hereda en el caso de que fuera necesario.

### 4.2.3 Trabajar con objetos en MATLAB

Para crear un objeto de una clase tenemos que llamar al constructor de la clase y el parámetro de salida del constructor será nuestro objeto. En el siguiente ejemplo, creamos el objeto OBJECT de tipo MICLESE.

```
OBJECT= MICLESE(62,34);
```

#### 4.2.3.1 Llamar a un método del objeto MICLESE

Para llamar a los métodos metodo1 y al metodo2 del objeto MIOBJETO, creado anteriormente, utilizamos el siguiente código.

```
%Método metodo1 que NO modifica el objeto OBJECT
resultado=metodo1(OBJECT);
%Método metodo2 que SÍ modifica el objeto OBJECT
OBJECT=metodo2(OBJECT);
```

#### 4.2.3.2 Crear un objeto CLASEHIJA y llamar a un método heredado de MICLESE

El siguiente código crea un objeto de tipo CLASEHIJA y después llama al método metodo1 heredado de la clase MICLESE.

```
%Creación del objeto de tipo MICLESEHIJA
OBJECTHIJO=MICLESEHIJA(75);
resultado=metodo1(OBJECTHIJO);
```

## CAPÍTULO V: SIMULADOR “SABOR”

---

## CAPÍTULO V: SIMULADOR "SABOR"

### 5.1 ESTRUCTURA DEL PROGRAMA

En este apartado se describirá la estructura de la aplicación educativa "SABOR" el cual permite simular el comportamiento de las antenas de bocinas.

Si abrimos el contenido de la carpeta encontraremos todos los directorios que comiezan con una arroba excepto GUI. Este es el que contiene la interfaz gráfica, y los demás corresponden a directorios que contienen clases y métodos que implementan la funcionalidad de nuestra aplicación.

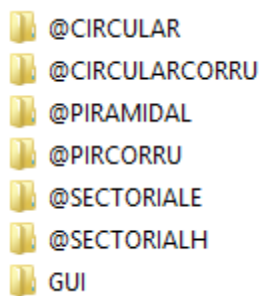


Fig 5.1 Contenido de la carpeta

#### 5.1.1 Diagrama de flujo general

En el directorio GUI los ficheros Inicio.fig e Inicio.m son los que permite generar la interfaz gráfica inicial. Para la mayor comprensión del programa se ha decidido realizar un diagrama de flujo general de la aplicación. La figura 5.2 que se mostrará a continuación, se aplica cuando se desea conocer los parámetros de salida para una o más veces, si deseas comparar entre dos diagramas de radiación pulsarías el menú diagrama del menú manteniendo la figura anterior. Cabe destacar que si necesitas información adicional del programa, se puede consultar el menú ayuda de la ventana principal un manual de autoayuda.

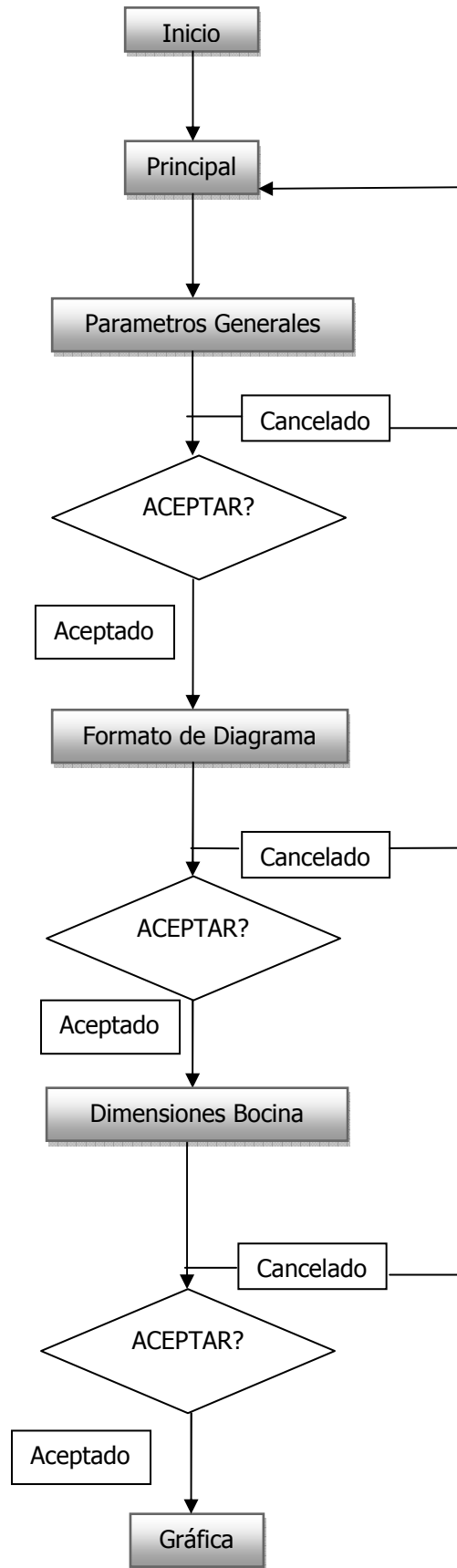


Fig 5.2 Diagrama de flujo general

### 5.1.2 Diagrama de clases UML de la aplicación

A continuación pasamos a describir la funcionalidad de cada una de las clases que hemos desarrollado así como sus propiedades y métodos

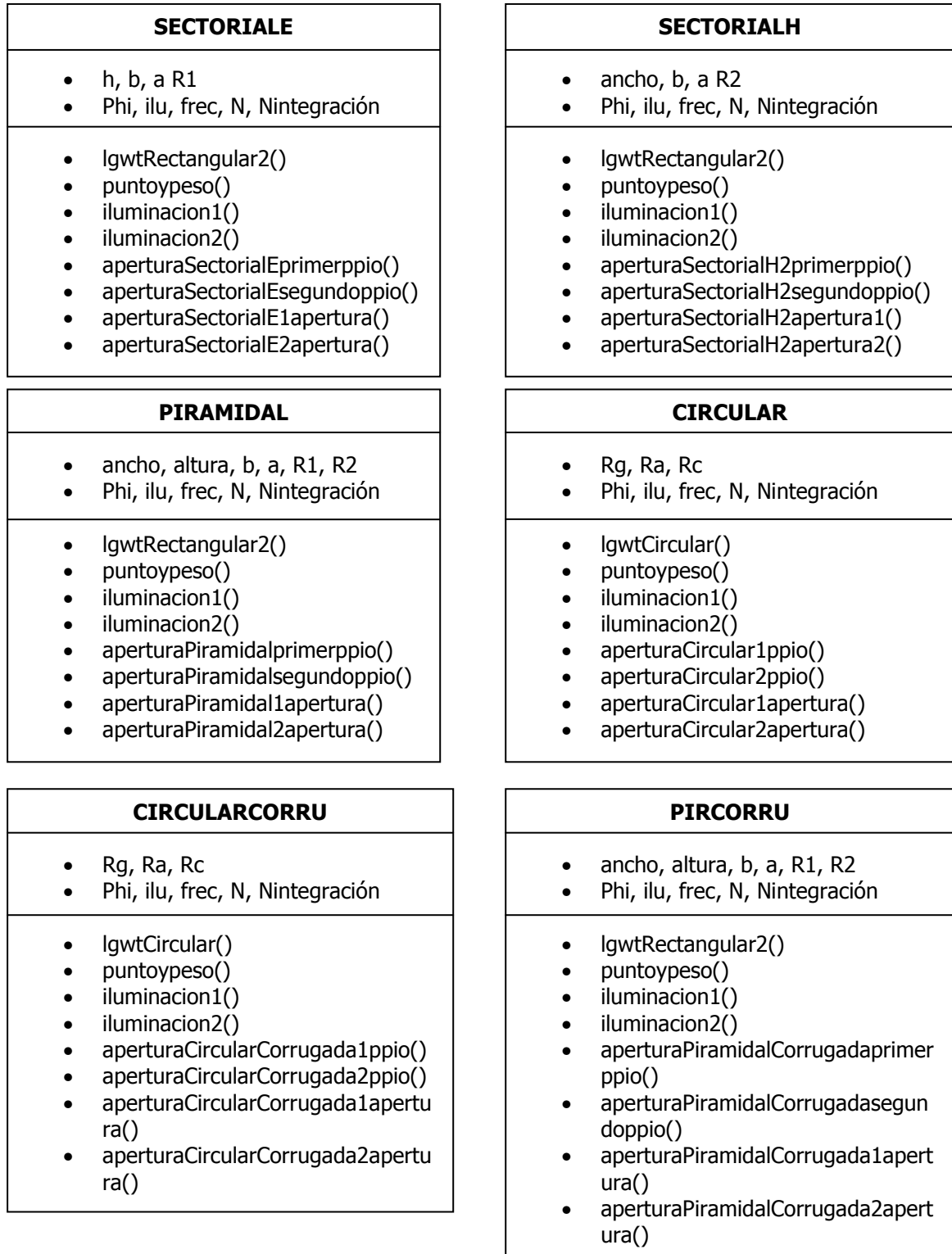


Fig 5.3 Diagrama de clases de la aplicación



### 5.1.3 Clase SECTORIALE

**Descripción:** Corresponde a la bocina sectorial plano E. El constructor de la clase SECTORIALE se le conoce como SECTORIALE.

#### Propiedades:

- **h:** Altura de la Apertura.
- **b:** Altura de la guía.
- **a:** ancho de la guía y de la Apertura.
- **R1:** Distancia del vértice que es generado al abocinar la guía con respecto al centro de la Apertura.
- **Phi:** Es el corte Phi. Permite conocer en qué plano deseamos realizar el análisis del diagrama de radiación.
- **ilu:** Es un número expresado en dB que se utilizará posteriormente para conocer el ancho de banda a un cierto nivel de iluminación.
- **Frec:** Es una variable que permite seleccionar la frecuencia de trabajo en GHz.
- **N:** Número de puntos que tiene la apertura.
- **Nintegracion:** Número de puntos empleados para dibujar posteriormente el diagrama de radiación.

#### Métodos:

- **[x2,y2,w1,w2]=lgwtRectangular2(sectoriale):** método que permite calcular los puntos y pesos utilizando la cuadratura de Gauss-Legendre en el intervalo [a,b]
- **puntoypeso(sectoriale):** Método que recoge los datos de salida del método lgwtRectangular2(sectoriale) y los procesa para convertir los puntos y pesos en una matriz de NxN.
- **[thetailu anchohaz centrofase]=iluminacion1(sectoriale):** Método que calcula el centro de fase y el ancho de haz a un cierto nivel de iluminación cuando analizas el primer principio de equivalencia. Además, se ha añadido como salida la variable thetailu que es la theta expresada en radianes a un cierto nivel de iluminación.
- **[thetailu anchohaz centrofase]=iluminacion2(sectoriale):** Método que calcula el centro de fase y el ancho de haz a un cierto nivel de iluminación cuando analizas el segundo principio de equivalencia. Además, se ha añadido como salida la variable thetailu que es la theta expresada en radianes a un cierto nivel de iluminación.
- **aperturaSectorialEprimerppio(sectoriale):** Método que calcula la Directividad y el Spillover para el primer principio de equivalencia e integrando en el diagrama. Además genera un archivo llamado bocina.mat que concentrará toda la información relativa a nuestra bocina.
- **aperturaSectorialEsegundoppio(sectoriale):** Método que calcula la Directividad y el Spillover para el segundo principio de equivalencia e integrando en el diagrama. Además

genera un archivo llamado bocina.mat que concentrará toda la información relativa a nuestra bocina.

- **aperturaSectorialE1apertura(sectoriale)**: Método que calcula la Directividad y el Spillover para el primer principio de equivalencia e integrando en la apertura. Además genera un archivo llamado bocina.mat que concentrará toda la información relativa a nuestra bocina.
- **aperturaSectorialE2apertura(sectoriale)**: Método que calcula la Directividad y el Spillover para el segundo principio de equivalencia e integrando en la apertura. Además genera un archivo llamado bocina.mat que concentrará toda la información relativa a nuestra bocina.

#### 5.1.4 Clase SECTORIALH

**Descripción:** Corresponde a la bocina sectorial plano H. El constructor de la clase SECTORIALH se le conoce como SECTORIALH.

##### Propiedades:

- **ancho**: Anchura de la Apertura.
- **b**: Altura de la guía.
- **a**: ancho de la guía.
- **R2**: Distancia del vértice que es generado al abocinar la guía con respecto al centro de la Apertura.
- **phi**: Es el corte Phi. Permite conocer en qué plano deseamos realizar el análisis del diagrama de radiación.
- **ilu**: Es un número expresado en dB que se utilizará posteriormente para conocer el ancho de banda a un cierto nivel de iluminación.
- **frec**: Es una variable que permite seleccionar la frecuencia de trabajo en GHz.
- **N**: Número de puntos que tiene la apertura.
- **Nintegracion**: Número de puntos empleados para dibujar posteriormente el diagrama de radiación.

##### Métodos:

- **[x2,y2,w1,w2]=lgwtRectangular2(sectorialh)**: Método que permite calcular los puntos y pesos utilizando la cuadratura de Gauss-Legendre en el intervalo [a,b]
- **puntoypeso(sectorialh)**: Método que recoge los datos de salida del método lgwtRectangular2(sectorialh) y los procesa para convertir los puntos y pesos en una matriz de NxN.
- **[thetailu anchohaz centrofase]=iluminacion1(sectorialh)**: Método que calcula el centro de fase y el ancho de haz a un cierto nivel de iluminación cuando analizas el primer principio de equivalencia. Además, se ha añadido como salida la variable thetailu que es la theta expresada en radianes a un cierto nivel de iluminación.

- **[thetailu anchohaz centrofase]=iluminacion2(sectorialh):** Método que calcula el centro de fase y el ancho de haz a un cierto nivel de iluminación cuando analizas el segundo principio de equivalencia. Además, se ha añadido como salida la variable thetailu que es la theta expresada en radianes a un cierto nivel de iluminación.
- **aperturaSectorialH2primerppio(sectorialh):** Método que calcula la Directividad y el Spillover para el primer principio de equivalencia e integrando en el diagrama. Además genera un archivo llamado bocina.mat que concentrará toda la información relativa a nuestra bocina.
- **aperturaSectorialH2segundoppio(sectorialh):** Método que calcula la Directividad y el Spillover para el segundo principio de equivalencia e integrando en el diagrama. Además genera un archivo llamado bocina.mat que concentrará toda la información relativa a nuestra bocina.
- **aperturaSectorialH2apertura1(sectorialh):** Método que calcula la Directividad y el Spillover para el primer principio de equivalencia e integrando en la apertura. Además genera un archivo llamado bocina.mat que concentrará toda la información relativa a nuestra bocina.
- **aperturaSectorialH2apertura2(sectorialh):** Método que calcula la Directividad y el Spillover para el segundo principio de equivalencia e integrando en la apertura. Además genera un archivo llamado bocina.mat que concentrará toda la información relativa a nuestra bocina.

### 5.1.5 Clase PIRAMIDAL

**Descripción:** Corresponde a la bocina sectorial Piramidal. El constructor de la clase PIRAMIDAL se le conoce como PIRAMIDAL.

#### Propiedades:

- **ancho:** Anchura de la Apertura.
- **altura:** Altura de la Apertura.
- **b:** Altura de la guía.
- **a:** ancho de la guía.
- **R1:** Distancia del vértice que es generado al abocinar la guía con respecto al centro de la Apertura.
- **R2:** Distancia del vértice que es generado al abocinar la guía con respecto al centro de la Apertura.
- **phi:** Es el corte Phi. Permite conocer en qué plano deseamos realizar el análisis del diagrama de radiación.
- **ilu:** Es un número expresado en dB que se utilizará posteriormente para conocer el ancho de banda a un cierto nivel de iluminación.
- **frec:** Es una variable que permite seleccionar la frecuencia de trabajo en GHz.
- **N:** Número de puntos que tiene la apertura.

- **Nintegracion:** Número de puntos empleados para dibujar posteriormente el diagrama de radiación.

**Métodos:**

- **[x2,y2,w1,w2]=lgwtRectangular2(piramidal):** método que permite calcular los puntos y pesos utilizando la cuadratura de Gauss-Legendre en el intervalo [a,b].
- **puntoypeso(piramidal):** Método que recoge los datos de salida del método lgwtRectangular2(piramidal) y los procesa para convertir los puntos y pesos en una matriz de NxN.
- **[thetailu anchohaz centrofase]=iluminacion1(piramidal):** Método que calcula el centro de fase y el ancho de haz a un cierto nivel de iluminación cuando analizas el primer principio de equivalencia. Además, se ha añadido como salida la variable thetailu que es la theta expresada en radianes a un cierto nivel de iluminación.
- **[thetailu anchohaz centrofase]=iluminacion2(piramidal):** Método que calcula el centro de fase y el ancho de haz a un cierto nivel de iluminación cuando analizas el segundo principio de equivalencia. Además, se ha añadido como salida la variable thetailu que es la theta expresada en radianes a un cierto nivel de iluminación.
- **aperturaPiramidalprimerppio(piramidal):** Método que calcula la Directividad y el Spillover para el primer principio de equivalencia e integrando en el diagrama. Además genera un archivo llamado bocina.mat que concentrará toda la información relativa a nuestra bocina.
- **aperturaPiramidalsegundoppio(piramidal):** Método que calcula la Directividad y el Spillover para el segundo principio de equivalencia e integrando en el diagrama. Además genera un archivo llamado bocina.mat que concentrará toda la información relativa a nuestra bocina.
- **aperturaPiramidal1apertura(piramidal):** Método que calcula la Directividad y el Spillover para el primer principio de equivalencia e integrando en la apertura. Además genera un archivo llamado bocina.mat que concentrará toda la información relativa a nuestra bocina.
- **aperturaPiramidal2apertura(piramidal):** Método que calcula la Directividad y el Spillover para el segundo principio de equivalencia e integrando en la apertura. Además genera un archivo llamado bocina.mat que concentrará toda la información relativa a nuestra bocina.

### 5.1.6 Clase CIRCULAR

**Descripción:** Corresponde a la bocina Cónica. El constructor de la clase CIRCULAR se le conoce como CIRCULAR.

#### Propiedades:

- **Rg:** Radio de la guía.
- **Ra:** Radio de la apertura.
- **Rc:** Distancia del vértice que es generado al abocinar la guía con respecto al centro de la Apertura.
- **phi:** Es el corte Phi. Permite conocer en qué plano deseamos realizar el análisis del diagrama de radiación.
- **ilu:** Es un número expresado en dB que se utilizará posteriormente para conocer el ancho de banda a un cierto nivel de iluminación.
- **frec:** Es una variable que permite seleccionar la frecuencia de trabajo en GHz.
- **N:** Número de puntos que tiene la apertura.
- **Nintegracion:** Número de puntos empleados para dibujar posteriormente el diagrama de radiación.

#### Métodos:

- **[x,w]=lgwtCircular(circular):** método que permite calcular los puntos y pesos utilizando la cuadratura de Gauss-Legendre en el intervalo [a,b]
- **puntoypeso(circular):** Método que recoge los datos de salida del método lgwtCircular (circular) y los procesa para convertir los puntos y pesos en una matriz de NxN.
- **[thetailu anchohaz centrofase]=iluminacion1(circular):** Método que calcula el centro de fase y el ancho de haz a un cierto nivel de iluminación cuando analizas el primer principio de equivalencia. Además, se ha añadido como salida la variable thetailu que es la theta expresada en radianes a un cierto nivel de iluminación.
- **[thetailu anchohaz centrofase]=iluminacion2(circular):** Método que calcula el centro de fase y el ancho de haz a un cierto nivel de iluminación cuando analizas el segundo principio de equivalencia. Además, se ha añadido como salida la variable thetailu que es la theta expresada en radianes a un cierto nivel de iluminación.
- **aperturaCircular1ppio(circular):** Método que calcula la Directividad y el Spillover para el primer principio de equivalencia e integrando en el diagrama. Además genera un archivo llamado bocina.mat que concentrará toda la información relativa a nuestra bocina.
- **aperturaCircular2ppio(circular):** Método que calcula la Directividad y el Spillover para el segundo principio de equivalencia e integrando en el diagrama. Además genera un archivo llamado bocina.mat que concentrará toda la información relativa a nuestra bocina.

- **aperturaCircular1apertura(circular):** Método que calcula la Directividad y el Spillover para el primer principio de equivalencia e integrando en la apertura. Además genera un archivo llamado bocina.mat que concentrará toda la información relativa a nuestra bocina.
- **aperturaCircular2apertura(circular):** Método que calcula la Directividad y el Spillover para el segundo principio de equivalencia e integrando en la apertura. Además genera un archivo llamado bocina.mat que concentrará toda la información relativa a nuestra bocina.

### 5.1.7 Clase CIRCULARCORRU

**Descripción:** Corresponde a la bocina Cónica Corrugada. El constructor de la clase CIRCULARCORRU se le conoce como CIRCULARCORRU.

#### Propiedades:

- **Rg:** Radio de la guía.
- **Ra:** Radio de la apertura.
- **Rc:** Distancia del vértice que es generado al abocinar la guía con respecto al centro de la Apertura.
- **phi:** Es el corte Phi. Permite conocer en qué plano deseamos realizar el análisis del diagrama de radiación.
- **ilu:** Es un número expresado en dB que se utilizará posteriormente para conocer el ancho de banda a un cierto nivel de iluminación.
- **frec:** Es una variable que permite seleccionar la frecuencia de trabajo en GHz.
- **N:** Número de puntos que tiene la apertura.
- **Nintegracion:** Número de puntos empleados para dibujar posteriormente el diagrama de radiación.

#### Métodos:

- **[x,w]=lgwtCircular(circularcorru):** método que permite calcular los puntos y pesos utilizando la cuadratura de Gauss-Legendre en el intervalo [a,b]
- **puntoypeso(circularcorru):** Método que recoge los datos de salida del método lgwtCircular (circularcorru) y los procesa para convertir los puntos y pesos en una matriz de NxN.
- **[thetailu anchohaz centrofase]=iluminacion1(circularcorru):** Método que calcula el centro de fase y el ancho de haz a un cierto nivel de iluminación cuando analizas el primer principio de equivalencia. Además, se ha añadido como salida la variable thetailu que es la theta expresada en radianes a un cierto nivel de iluminación.
- **[thetailu anchohaz centrofase]=iluminacion2(circularcorru):** Método que calcula el centro de fase y el ancho de haz a un cierto nivel de iluminación cuando analizas el segundo principio de equivalencia. Además, se ha añadido como salida la variable thetailu que es la theta expresada en radianes a un cierto nivel de iluminación.

- **aperturaCircularCorrugada1ppio(circularcorru)**: Método que calcula la Directividad y el Spillover para el primer principio de equivalencia e integrando en el diagrama. Además genera un archivo llamado bocina.mat que concentrará toda la información relativa a nuestra bocina.
- **aperturaCircularCorrugada2ppio(circularcorru)**: Método que calcula la Directividad y el Spillover para el segundo principio de equivalencia e integrando en el diagrama. Además genera un archivo llamado bocina.mat que concentrará toda la información relativa a nuestra bocina.
- **aperturaCircularCorrugada1apertura(circularcorru)**: Método que calcula la Directividad y el Spillover para el primer principio de equivalencia e integrando en la apertura. Además genera un archivo llamado bocina.mat que concentrará toda la información relativa a nuestra bocina.
- **aperturaCircularCorrugada2apertura(circularcorru)**: Método que calcula la Directividad y el Spillover para el segundo principio de equivalencia e integrando en la apertura. Además genera un archivo llamado bocina.mat que concentrará toda la información relativa a nuestra bocina.

#### 5.1.8 Clase PIRCORRU

**Descripción:** Corresponde a la bocina sectorial Piramidal Corrugada. El constructor de la clase PIRCORRU se le conoce como PIRCORRU.

##### Propiedades:

- **ancho:** Anchura de la Apertura.
- **altura:** Altura de la Apertura.
- **b:** Altura de la guía.
- **a:** ancho de la guía.
- **R1:** Distancia del vértice que es generado al abocinar la guía con respecto al centro de la Apertura.
- **R2:** Distancia del vértice que es generado al abocinar la guía con respecto al centro de la Apertura.
- **phi:** Es el corte Phi. Permite conocer en qué plano deseamos realizar el análisis del diagrama de radiación.
- **ilu:** Es un número expresado en dB que se utilizará posteriormente para conocer el ancho de banda a un cierto nivel de iluminación.
- **frec:** Es una variable que permite seleccionar la frecuencia de trabajo en GHz.
- **N:** Número de puntos que tiene la apertura.
- **Nintegracion:** Número de puntos empleados para dibujar posteriormente el diagrama de radiación.

**Métodos:**

- **[x2,y2,w1,w2]=lgwtRectangular2(pircorru):** método que permite calcular los puntos y pesos utilizando la cuadratura de Gauss-Legendre en el intervalo [a,b]
- **puntoypeso(pircorru):** Método que recoge los datos de salida del método lgwtRectangular2(pircorru) y los procesa para convertir los puntos y pesos en una matriz de NxN.
- **[thetailu anchohaz centrofase]=iluminacion1(pircorru):** Método que calcula el centro de fase y el ancho de haz a un cierto nivel de iluminación cuando analizas el primer principio de equivalencia. Además, se ha añadido como salida la variable thetailu que es la theta expresada en radianes a un cierto nivel de iluminación.
- **[thetailu anchohaz centrofase]=iluminacion2(pircorru):** Método que calcula el centro de fase y el ancho de haz a un cierto nivel de iluminación cuando analizas el segundo principio de equivalencia. Además, se ha añadido como salida la variable thetailu que es la theta expresada en radianes a un cierto nivel de iluminación.
- **aperturaPiramidalCorrugada1apertura(pircorru):** Método que calcula la Directividad y el Spillover para el primer principio de equivalencia e integrando en el diagrama. Además genera un archivo llamado bocina.mat que concentrará toda la información relativa a nuestra bocina.
- **aperturaPiramidalCorrugada2apertura(pircorru):** Método que calcula la Directividad y el Spillover para el segundo principio de equivalencia e integrando en el diagrama. Además genera un archivo llamado bocina.mat que concentrará toda la información relativa a nuestra bocina.
- **aperturaPiramidalCorrugada1apertura(pircorru):** Método que calcula la Directividad y el Spillover para el primer principio de equivalencia e integrando en la apertura. Además genera un archivo llamado bocina.mat que concentrará toda la información relativa a nuestra bocina.
- **aperturaPiramidalCorrugada2apertura(pircorru):** Método que calcula la Directividad y el Spillover para el segundo principio de equivalencia e integrando en la apertura. Además genera un archivo llamado bocina.mat que concentrará toda la información relativa a nuestra bocina.



## 5.2. MANUAL DE USUARIO

Recordamos que nuestro software se ha desarrollado buscando la mayor intuitividad y facilidad de aprendizaje para el usuario. Por ello proporcionaremos un manual de usuario explicando el uso de cada ventana.

### 5.2.1 Ventana Inicio

La ventana de Inicio es la primera ventana que aparece en la aplicación "SABOR". Se compone de dos botones en el que el usuario puede elegir entre analizar bocinas o reflectores. Como esta parte está dedicada al estudio de bocinas pulsaremos el botón correspondiente para acceder al estudio de dichas antenas:

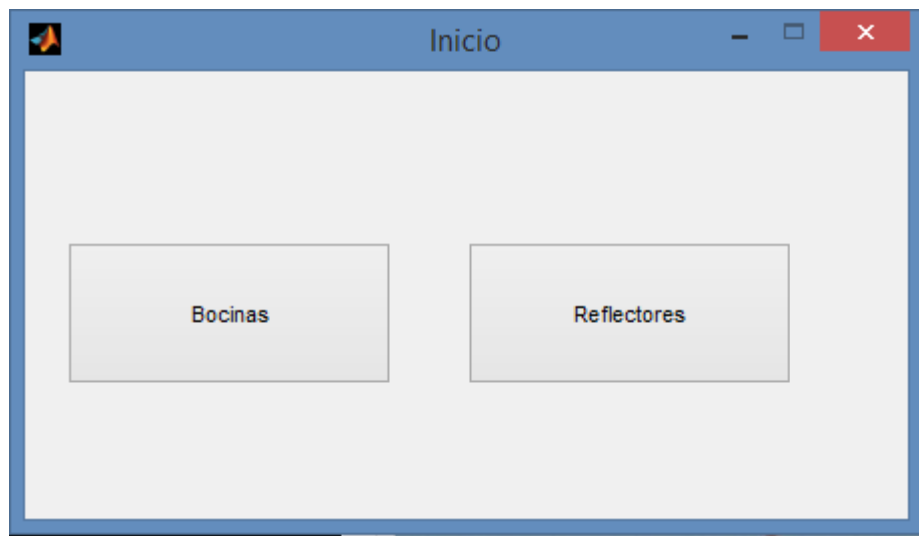


Fig 5.4 Pantalla de Inicio

### 5.2.2 Ventana Principal

En la siguiente figura mostraremos la pantalla principal que cerrará la de Inicio. Como podemos observar se compone de un menú de 7 elementos: Bocinas, Menú, Parámetros generales, Formato Diagrama, Gráfica, Comparar y Ayuda. Por defecto, siempre aparecerá la imagen de la bocina sectorial plano E.

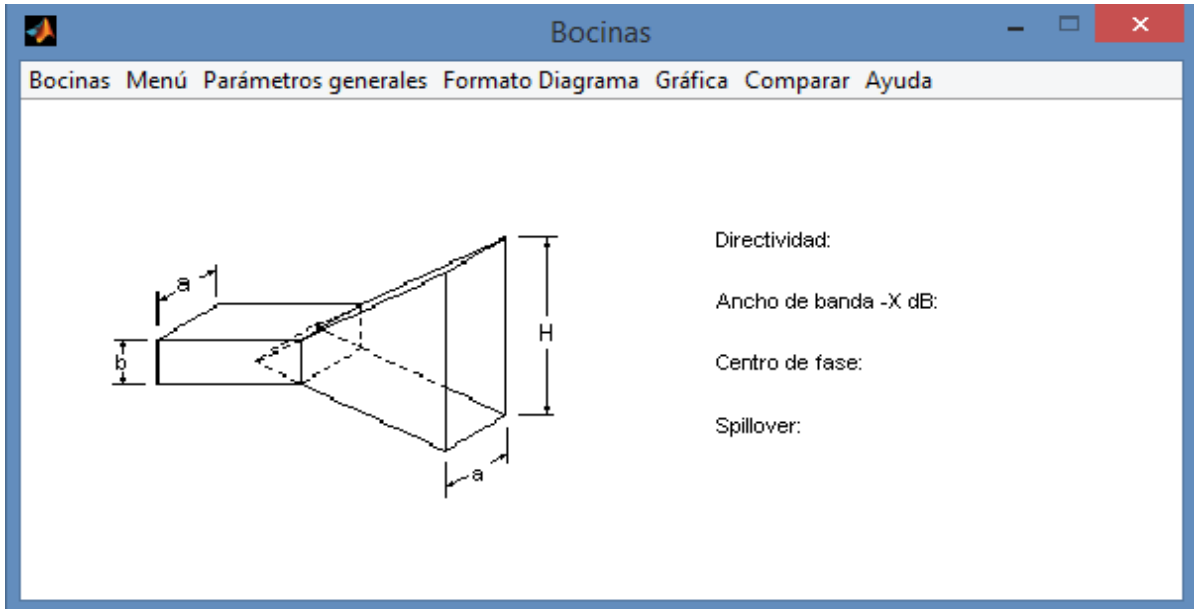


Fig 5.5 Pantalla Principal

A continuación mostraremos la imagen cuando pulsamos el menú Bocinas. En esta parte nos encontraremos las diferentes bocinas que deseamos analizar: Sectorial H, Sectorial E, Piramidal, Cónica, Corrugada Cónica y Corrugada Piramidal.

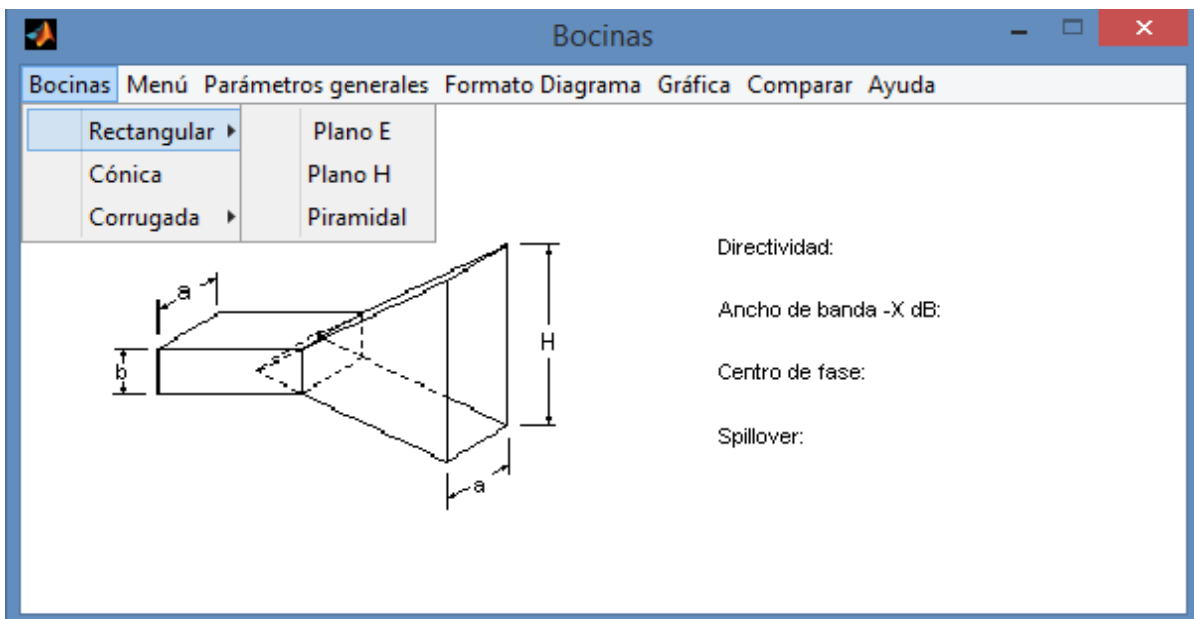


Fig 5.6 Pantalla Principal menú Bocinas

Según el tipo de bocina que se desea analizar aparecerá una imagen u otra. Además, saltará una nueva ventana cuando el usuario selecciona una bocina, apareciendo las dimensiones con las variables de entradas por defecto y el tipo de análisis que se quiere aplicar.

### 5.2.2.1 Menú bocina sectorial plano E

Empezaremos por la dimensiones de la bocina sectorial plano E. En la siguiente figura aparecen los datos de entrada y el análisis que se desea realizar.

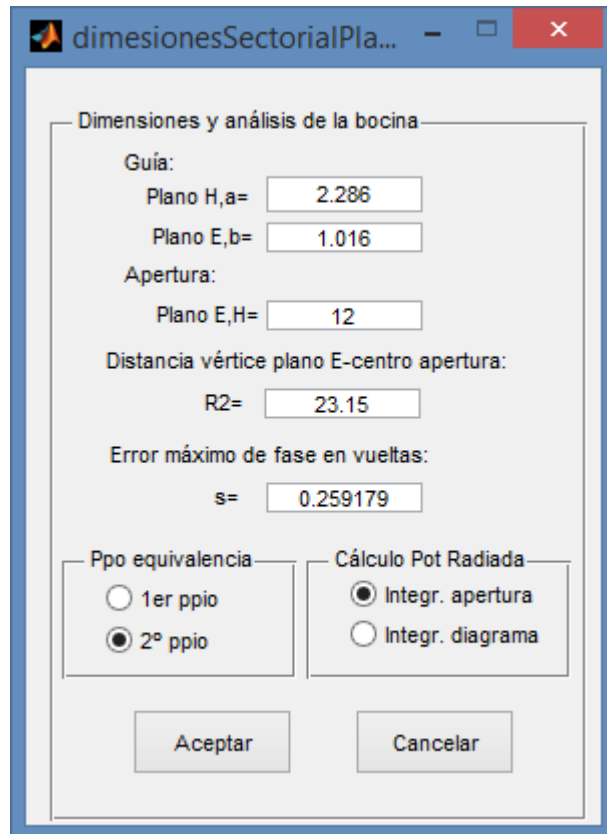


Fig 5.7 Dimensiones sectorial plano E

Como podemos observar mediante un selector se podrá seleccionar una opción u otra, dependiendo del análisis de principios o del tipo de integración que se desee ejecutar. Además también se podrá cambiar las variables de entrada a gusto del usuario.

En caso de que el usuario desee volver a la pantalla principal, se puede presionar el botón cancelar, pero antes se formulará una pregunta para confirmar si desea irse de la ventana.

### 5.2.2.2 Menú bocina sectorial plano H

Proseguimos con la ventana de dimensiones de la bocina sectorial plano H. En la siguiente figura aparecen los datos de entrada y el análisis que desea realizar.

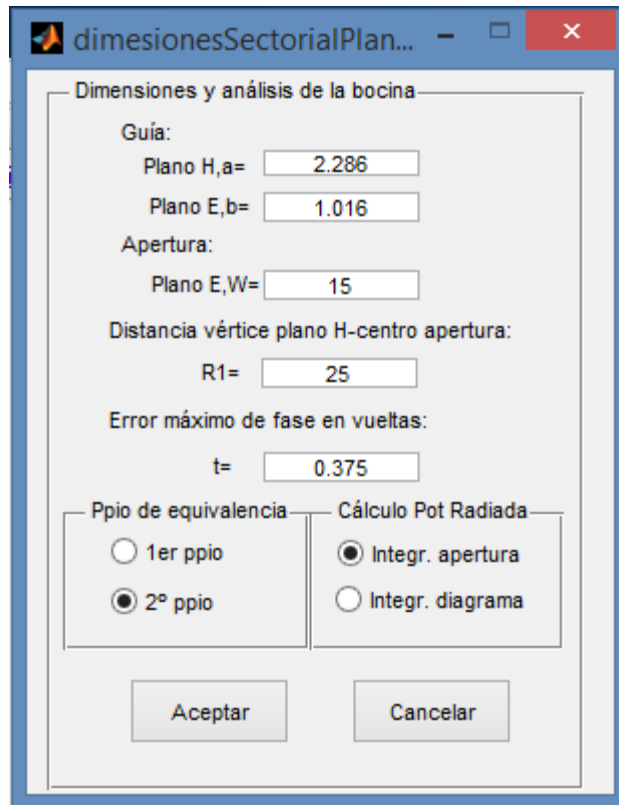


Fig 5.8 Dimensiones sectorial plano H

Como podemos observar mediante un selector se podrá seleccionar una opción u otra, dependiendo del análisis de principios o del tipo de integración que se desee ejecutar. Además también se podrá cambiar las variables de entrada a gusto del usuario.

En caso de que el usuario desee volver a la pantalla principal, se puede presionar el botón cancelar, pero antes se formulará una pregunta para confirmar si desea irse de la ventana.

### 5.2.2.3 Menú bocina sectorial plano Piramidal

La siguiente figura corresponde a la bocina sectorial Piramidal:

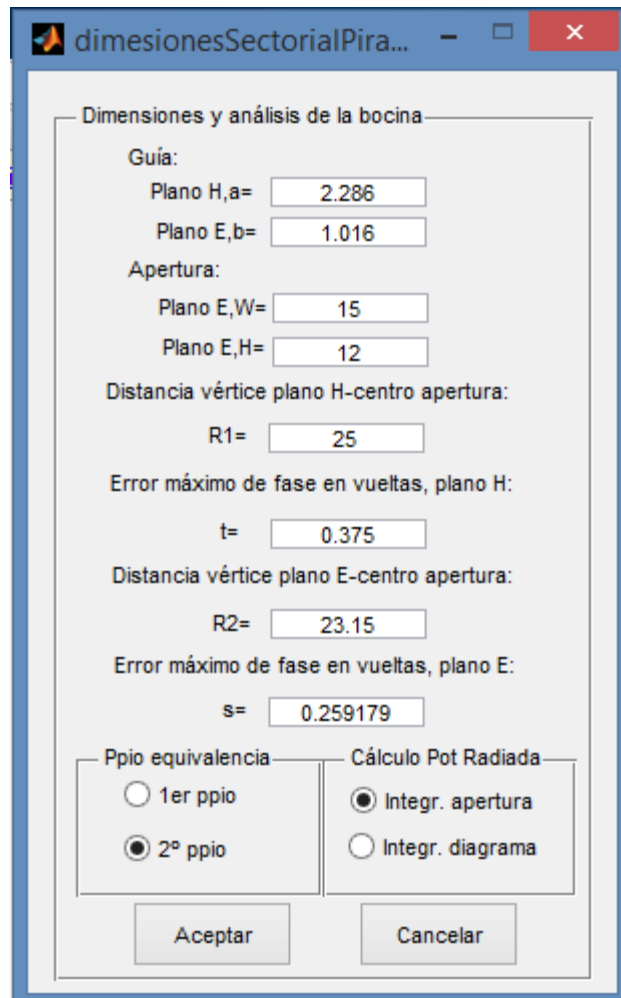


Fig 5.9 Dimensiones sectorial Piramidal

Como podemos observar mediante un selector se podrá seleccionar una opción u otra, dependiendo del análisis de principios o del tipo de integración que se desee ejecutar. Además también se podrá cambiar las variables de entrada a gusto del usuario.

En caso de que el usuario desee volver a la pantalla principal, se puede presionar el botón cancelar, pero antes se formulará una pregunta para confirmar si desea irse de la ventana.

### 5.2.2.4 Menú bocina Cónica

La siguiente figura corresponde a la bocina Cónica:

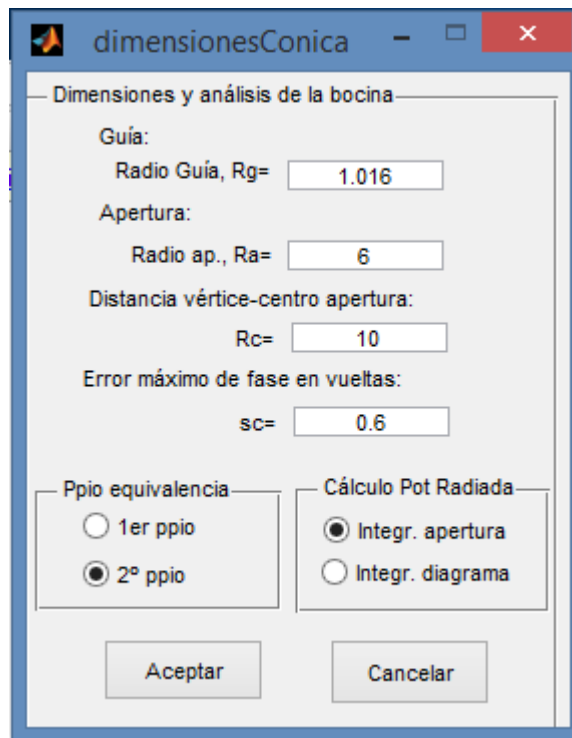


Fig 5.10 Dimensiones Cónica

Como podemos observar mediante un selector se podrá seleccionar una opción u otra, dependiendo del análisis de principios o del tipo de integración que se desee ejecutar. Además también se podrá cambiar las variables de entrada a gusto del usuario.

En caso de que el usuario desee volver a la pantalla principal, se puede presionar el botón cancelar, pero antes se formulará una pregunta para confirmar si desea irse de la ventana.

### 5.2.2.5 Menú bocina cónica corrugada

La siguiente figura corresponde a la bocina cónica corrugada:

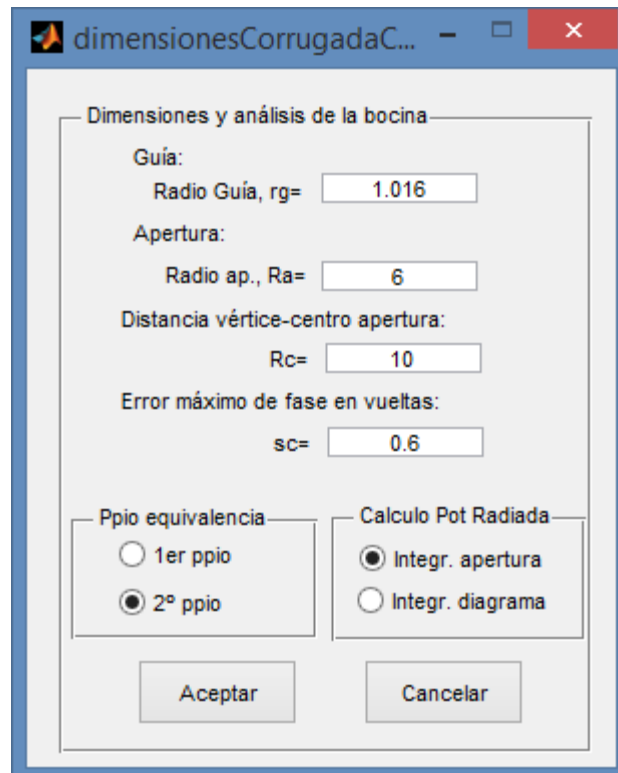


Fig 5.11 Dimensiones cónica corrugada

Como podemos observar mediante un selector se podrá seleccionar una opción u otra, dependiendo del análisis de principios o del tipo de integración que se desee ejecutar. Además también se podrá cambiar las variables de entrada a gusto del usuario.

En caso de que el usuario desee volver a la pantalla principal, se puede presionar el botón cancelar, pero antes se formulará una pregunta para confirmar si desea irse de la ventana.

### 5.2.2.6 Menú bocina Piramidal corrugada

La siguiente figura corresponde a la bocina Piramidal Corrugada:

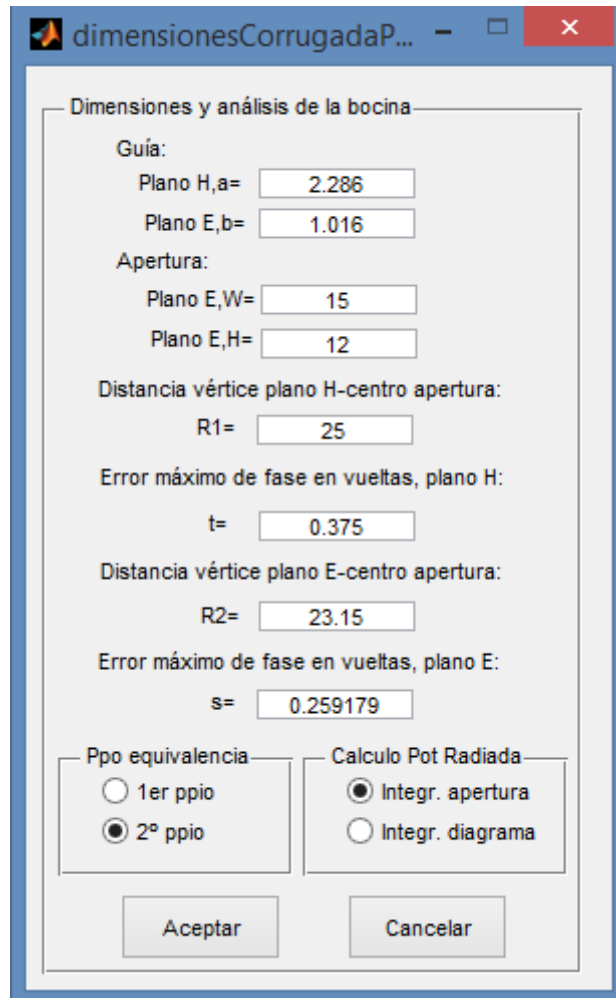


Fig 5.12 Dimensiones Piramidal Corrugada

Como podemos observar mediante un selector se podrá seleccionar una opción u otra, dependiendo del análisis de principios o del tipo de integración que se desee ejecutar. Además también se podrá cambiar las variables de entrada a gusto del usuario.

En caso de que el usuario desee volver a la pantalla principal, se puede presionar el botón cancelar, pero antes se formulará una pregunta para confirmar si desea irse de la ventana.



### 5.2.2.7 Menú Menú

Una vez analizadas las dimensiones de las bocinas volvemos a la pantalla principal. A continuación mostraremos la imagen cuando pulsamos el menú Menú:

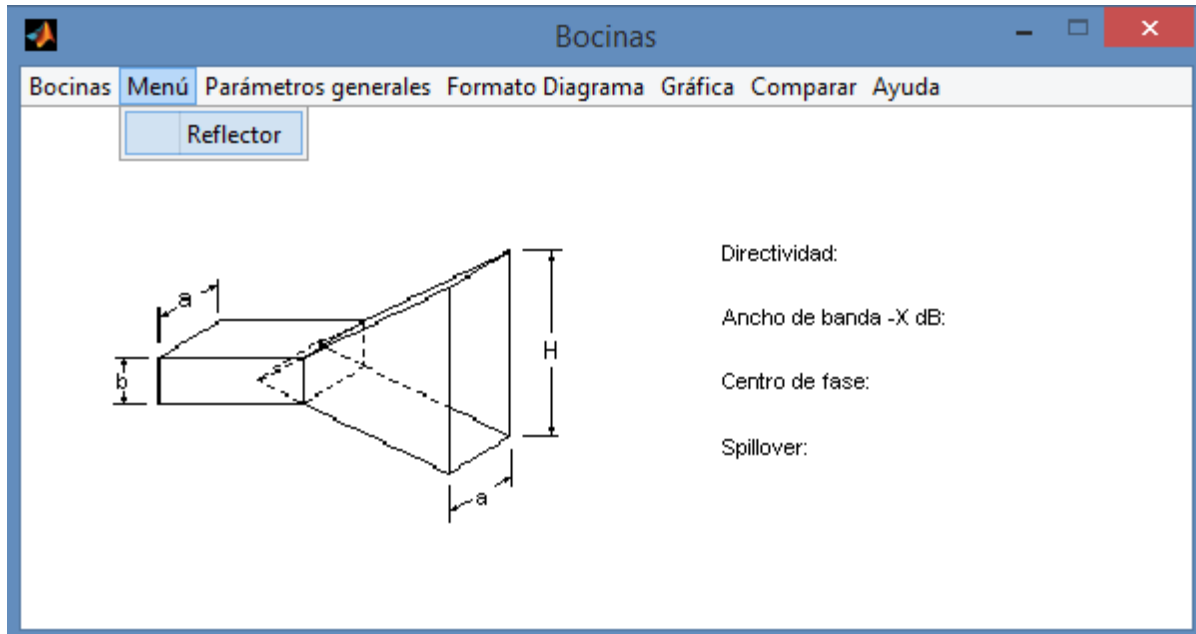


Fig 5.13 Reflector

No presenta código asociado ya que no están diseñados los reflectores en este proyecto, en el momento en el que tengamos la información se pulsará en el submenú Reflector

### 5.2.2.8 Menú Parámetros Generales

Como hicimos anteriormente volvemos a la pantalla principal. A continuación mostraremos la imagen cuando pulsamos el menú Parámetros Generales:

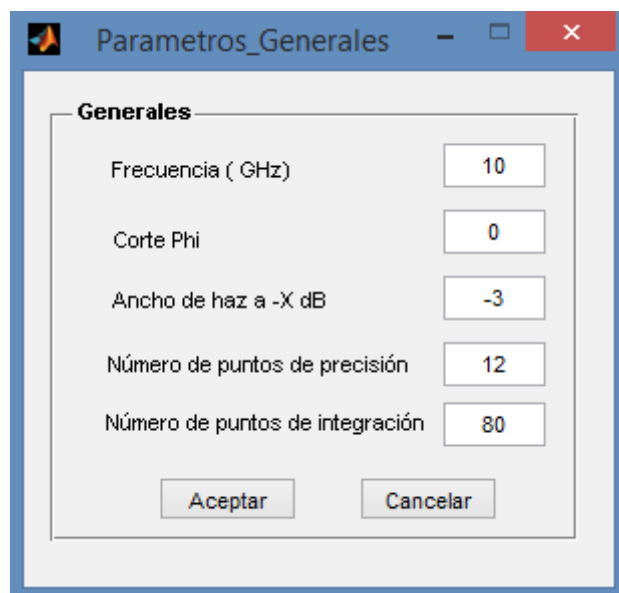


Fig 5.14 Parametros generales

Su código asociado trata de leer los parámetros generales que desea aplicar el usuario y guardarlos en un fichero.mat con el propósito de utilizarlo en el análisis de cada una de las bocinas.

En caso de que el usuario desee volver a la pantalla principal, se puede presionar el botón cancelar, pero antes se formulará una pregunta para confirmar si desea irse de la ventana.

### **5.2.2.9 Menú Formato Diagrama**

Como hicimos anteriormente volvemos a la pantalla principal. A continuación mostraremos la imagen cuando pulsamos el menú Formato Diagrama:

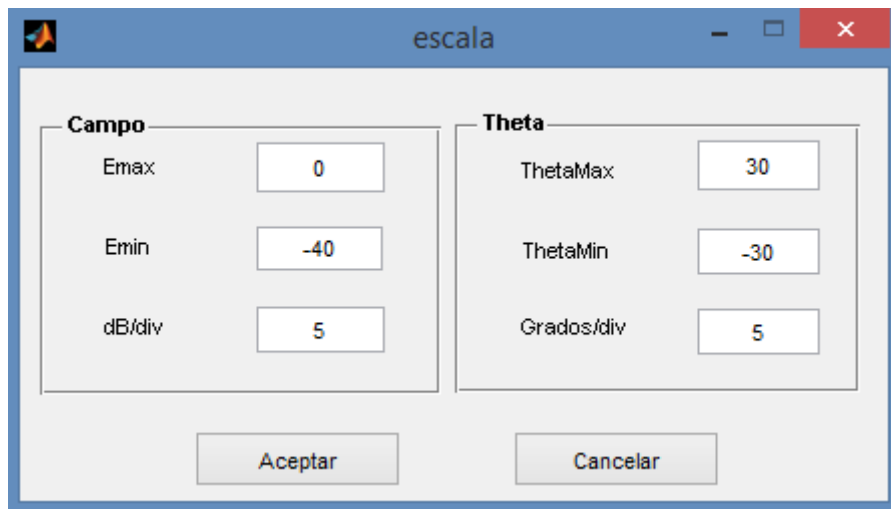


Fig 5.15 Formato de Diagrama

El objetivo de este menú es darle formato al diagrama de radiación. Su código asociado trata de leer los parámetros de escala que desea aplicar el usuario y guardarlos en un fichero.mat para posteriormente utilizarlo en el menú gráfica. El código del botón cancelar es exactamente el mismo que en casos anteriores, formulará una pregunta para confirmar si desea irse de la ventana.

### **5.2.2.10 Menú Gráfica**

A continuación volvemos al programa principal y pulsamos el menú gráfica. Rápidamente dibuja el diagrama de radiación y muestra los parámetros de salida a partir de las características de la bocina diseñada. Se abrirá una ventana donde podremos editar la gráfica a nuestro interés.

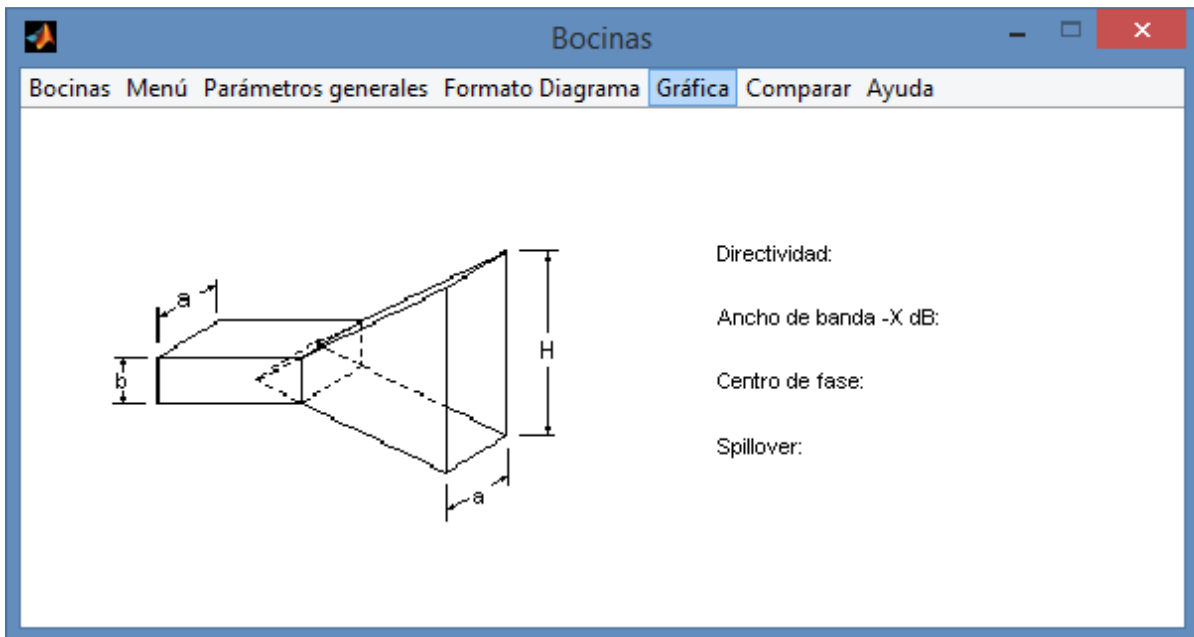


Fig 5.16 Gráfica

### 5.2.2.11 Menú Comparar

Si lo que queremos ver son las consecuencias del cambio de plano o de cualquier modificación en alguno de los parámetros de entradas, pulsaremos el botón Comparar para visualizarlos en el diagrama de radiación. Para ello, tenemos que mantener la gráfica anterior con el objeto de que no se borren los cambios.

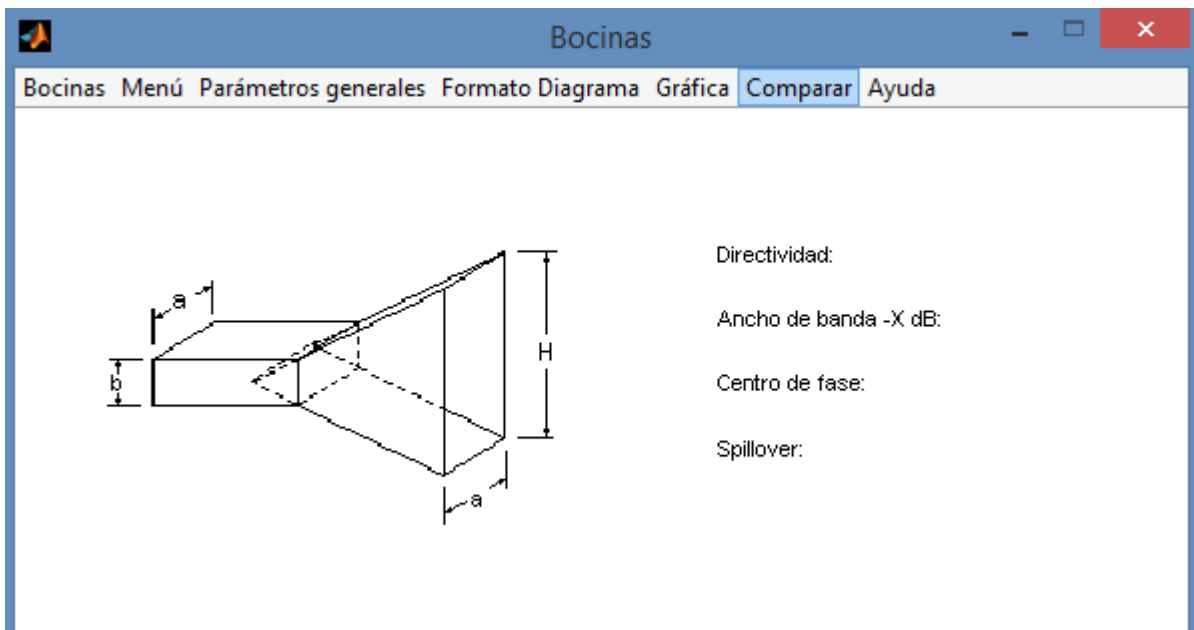


Fig 5.17 Comparar

### 5.2.2.12 Menú Ayuda

Por último, pulsando el botón de ayuda mostrará una guía de autoayuda en formato .pdf.

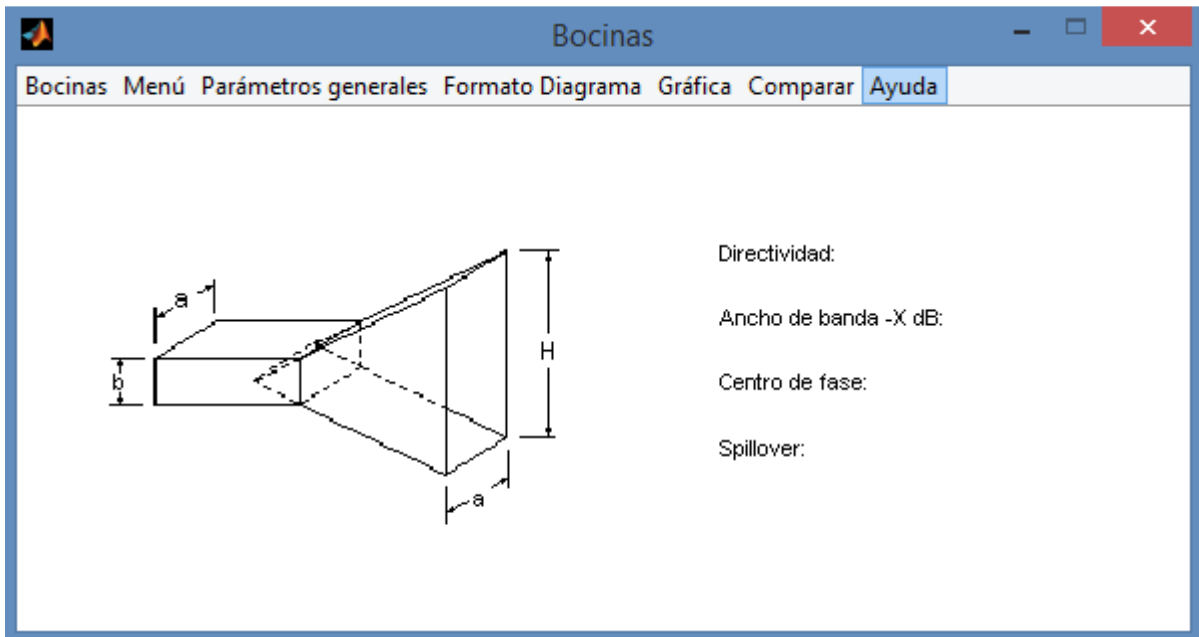


Fig 5.18 Ayuda

## 5.3 ENTRADAS Y SALIDAS DE CADA DIRECTORIO

En este apartado se va a ofrecer una visión más cercana de cada función de nuestro directorio. Como resumen, se ha hecho un breve comentario explicando qué hace la función y se ha suministrado los datos de entradas y de salida. Se proporcionarán tanto los constructores como los métodos de forma práctica en entorno de desarrollo MATLAB.

### 5.3.1 @SECTORIALE

#### 5.3.1.1 Constructor de la clase SECTORIALE

```
function se=SECTORIALE%(N,altura,a,b,R2,frec,phi,ilu)
%DESCRIPCION:
% Constructor de la clase SECTORIALE
%PARAMETROS
%  ENTRADA:
%  h: Altura de la Apertura.
%  b: Altura de la guía.
%  a: ancho de la guía y de la Apertura.
%  R2: Distancia del vértice que es generado al abocinar la guía con
respecto al centro de la Apertura.
%  Phi: Es el corte Phi. Permite conocer en qué plano deseamos realizar
el análisis del diagrama de radiación.
%  ilu: Es un número expresado en dB que se utilizará posteriormente para
conocer el ancho de banda a un cierto nivel de iluminación.
%  Frec: Es una variable que permite seleccionar la frecuencia de trabajo
en GHz.
%  N: Número de puntos que tiene la apertura.
%  Nintegracion: Número de puntos empleados para dibujar posteriormente
el diagrama de radiación.
%  SALIDA:
%  se: Objeto de tipo SECTORIALE
```

#### 5.3.1.2 Método lgwtRectangular2(sectoriale)

```
function [x2,y2,w1,w2]=lgwtRectangular2(sectoriale)
%DESCRIPCION:
% método que permite calcular los puntos y pesos utilizando
%la cuadratura de Gauss-Legendre en el intervalo [a,b]
%PARAMETROS
%  ENTRADA:
%      N: Número de puntos
%      h: Altura de la apertura
%      a: ancho de la guía
%  SALIDA:
```

```
%      x2: puntos en coordenada x
%      y2: puntos en coordenada y
%      w1: pesos en x
%      w2: pesos en y
```

### **5.3.1.3 Método puntospeso(sectoriale)**

```
function puntoypeso(sectoriale)
%DESCRIPCION:
%Método que recoge los datos de salida del método
lgwtRectangular2(sectoriale)
%y los procesa para convertir los puntos y pesos en una matriz de NxN.
%PARAMETROS
%  ENTRADA:
%      x2: puntos en coordenada x
%      y2: puntos en coordenada y
%      w1: pesos en x
%      w2: pesos en y
%      N: Número de puntos
%  SALIDA:
%      x: puntos en coordenada x
%      y: puntos en coordenada y
%      w: pesos
```

### **5.3.1.4 Método iluminacion1(sectoriale)**

```
function [thetailu anchohaz centrofase]=iluminacion1(sectoriale)
%DESCRIPCION:
%Método que calcula el centro de fase y el ancho de haz a un cierto
%nivel de iluminación cuando analizas el primer principio de equivalencia.
%Además, se ha añadido como salida la variable thetailu que es la theta
expresada
%en radianes a un cierto nivel de iluminación.
%PARAMETROS
%  ENTRADA:
%      N: Número de puntos
%      h: Altura de la apertura
%      a: ancho de la guía
%      R2: distancia
%      phi: corte Phi
%      frec: frecuencia de trabajo
%      ilu:iluminación
%  SALIDA:
%      thetailu: theta en radianes a un determinado nivel ilu dB
```

```
% anchohaz: Ancho de haz a X dBs
% centrofase: Centro de fase
```

### **5.3.1.5 Método iluminacion2(sectoriale)**

```
function [thetailu anchohaz centrofase]=iluminacion2(sectoriale)
%DESCRIPCION:
%Método que calcula el centro de fase y el ancho de haz a un cierto
%nivel de iluminación cuando analizas el segundo principio de
equivalencia.
%Además, se ha añadido como salida la variable thetailu que es la theta
expresada
%en radianes a un cierto nivel de iluminación.
%PARAMETROS
% ENTRADA:
%     N: Número de puntos
%     h: Altura de la apertura
%     a: ancho de la guía
%     R2: distancia
%     phi: corte Phi
%     frec: frecuencia de trabajo
%     ilu:iluminación
% SALIDA:
%     thetailu: theta en radianes a un determinado nivel ilu dB
%     anchohaz: Ancho de haz a X dBs
%     centrofase: Centro de fase
```

### **5.3.1.6 MétodoaperturaSectorialEprimerppio(sectoriale)**

```
function aperturaSectorialEprimerppio(sectoriale)
%DESCRIPCION:
%Método que calcula la Directividad y el Spillover para el primer
principio de equivalencia
%e integrando en el diagrama. Además genera un archivo llamado bocina.mat
%que concentrará toda la información relativa a nuestra bocina.
%PARAMETROS
% ENTRADA:
%     N: Número de puntos
%     h: Altura de la apertura
%     a: ancho de la guía
%     R2: distancia
%     phi: corte Phi
%     frec: frecuencia de trabajo
%     ilu:iluminación
```

```
%      Nintegracion:Número de puntos empleados para dibujar el diagrama
%      thetailu: theta en radianes a un determinado nivel ilu dB
%      SALIDA:
%      Directividad: Directividad
%      Spillover: Spillover
```

### **5.3.1.7 Método aperturaSectorialEsegundoppio(sectoriale)**

```
function aperturaSectorialEsegundoppio(sectoriale)
%DESCRIPCION:
%Método que calcula la Directividad y el Spillover para el segundo
principio de equivalencia
%e integrando en el diagrama. Además genera un archivo llamado bocina.mat
%que concentrará toda la información relativa a nuestra bocina.
%PARAMETROS
%      ENTRADA:
%      N: Número de puntos
%      h: Altura de la apertura
%      a: ancho de la guía
%      R2: distancia
%      phi: corte Phi
%      frec: frecuencia de trabajo
%      ilu:iluminación
%      Nintegracion:Número de puntos empleados para dibujar el diagrama
%      thetailu: theta en radianes a un determinado nivel ilu dB
%      SALIDA:
%      Directividad: Directividad
%      Spillover: Spillover
```

### **5.3.1.8 Método aperturaSectorialElapertura(sectoriale)**

```
function aperturaSectorialElapertura(sectoriale)
%DESCRIPCION:
%Método que calcula la Directividad y el Spillover para el primer
principio de equivalencia
%e integrando en la apertura. Además genera un archivo llamado bocina.mat
%que concentrará toda la información relativa a nuestra bocina.
%PARAMETROS
%      ENTRADA:
%      N: Número de puntos
%      h: Altura de la apertura
%      a: ancho de la guía
%      R2: distancia
%      phi: corte Phi
```



```
%      frec: frecuencia de trabajo
%      ilu:iluminación
%      Nintegracion:Número de puntos empleados para dibujar el diagrama
%      thetailu: theta en radianes a un determinado nivel ilu dB
%  SALIDA:
%      Directividad: Directividad
%      Spillover: Spillover
```

### **5.3.1.9 Método aperturaSectorialE2apertura(sectoriale)**

```
function aperturaSectorialE2apertura(sectoriale)
%DESCRIPCION:
%Método que calcula la Directividad y el Spillover para el segundo
principio de equivalencia
%e integrando en la apertura. Además genera un archivo llamado bocina.mat
%que concentrará toda la información relativa a nuestra bocina.
%PARAMETROS
%  ENTRADA:
%      N: Número de puntos
%      h: Altura de la apertura
%      a: ancho de la guía
%      R2: distancia
%      phi: corte Phi
%      frec: frecuencia de trabajo
%      ilu:iluminación
%      Nintegracion:Número de puntos empleados para dibujar el diagrama
%      thetailu: theta en radianes a un determinado nivel ilu dB
%  SALIDA:
%      Directividad: Directividad
%      Spillover: Spillover
```

## 5.3.2 @SECTORIALH

### 5.3.2.1 Constructor de la clase SECTORIALH

```
function sh=SECTORIALH
%DESCRIPCION:
% Constructor de la clase SECTORIALH
%PARAMETROS
%   ENTRADA:
%   ancho: anchura de la Apertura.
%   b: Altura de la guía.
%   a: ancho de la guía.
%   R1: Distancia del vértice que es generado al abocinar la guía con
respecto al centro de la Apertura.
%   Phi: Es el corte Phi. Permite conocer en qué plano deseamos realizar
el análisis del diagrama de radiación.
%   ilu: Es un número expresado en dB que se utilizará posteriormente para
conocer el ancho de banda a un cierto nivel de iluminación.
%   Frec: Es una variable que permite seleccionar la frecuencia de trabajo
en GHz.
%   N: Número de puntos que tiene la apertura.
%   Nintegracion: Número de puntos empleados para dibujar posteriormente
el
%   diagrama de radiación.
%   SALIDA:
%   sh: Objeto de tipo SECTORIALH
```

### 5.3.2.2 Método lgwtRectangular2(sectorialh)

```
function [x2,y2,w1,w2]=lgwtRectangular2(sectorialh)
%DESCRIPCION:
% método que permite calcular los puntos y pesos utilizando
%la cuadratura de Gauss-Legendre en el intervalo [a,b]
%PARAMETROS
%   ENTRADA:
%       N: Número de puntos
%   ancho: anchura de la Apertura.
%       b: Altura de la guía.
%   SALIDA:
%       x2: puntos en coordenada x
%       y2: puntos en coordenada y
%       w1: pesos en x
%       w2: pesos en y
```

**5.3.2.3 Método puntosypeso(sectorialh)**

```
function puntoypeso(sectorialh)
%DESCRIPCION:
%Método que recoge los datos de salida del método
lgwtRectangular2(sectorialh)
%y los procesa para convertir los puntos y pesos en una matriz de NxN.
%PARAMETROS
%   ENTRADA:
%       x2: puntos en coordenada x
%       y2: puntos en coordenada y
%       w1: pesos en x
%       w2: pesos en y
%       N: Número de puntos
%   SALIDA:
%       x: puntos en coordenada x
%       y: puntos en coordenada y
%       w: pesos
```

**5.3.2.4 Método iluminacion1(sectorialh)**

```
function [thetailu anchohaz centrofase]=iluminacion1(sectorialh)
%DESCRIPCION:
%Método que calcula el centro de fase y el ancho de haz a un cierto
%nivel de iluminación cuando analizas el primer principio de equivalencia.
%Además, se ha añadido como salida la variable thetailu que es la theta
expresada
%en radianes a un cierto nivel de iluminación.
%PARAMETROS
%   ENTRADA:
%       ancho: anchura de la Apertura.
%       b: Altura de la guía.
%       a: ancho de la guía.
%       R1: distancia
%       N: Número de puntos
%       phi: corte Phi
%       frec: frecuencia de trabajo
%       ilu:iluminación
%   SALIDA:
%       thetailu: theta en radianes a un determinado nivel ilu dB
%       anchohaz: Ancho de haz a X dBs
%       centrofase: Centro de fase
```

**5.3.2.5 Método iluminacion2(sectorialh)**

```
function [thetailu anchohaz centrofase]=iluminacion2(sectorialh)
%DESCRIPCION:
%Método que calcula el centro de fase y el ancho de haz a un cierto
%nivel de iluminación cuando analizas el segundo principio de
equivalencia.
%Además, se ha añadido como salida la variable thetailu que es la theta
expresada
%en radianes a un cierto nivel de iluminación.
%PARAMETROS
%  ENTRADA:
%      ancho: anchura de la Apertura.
%      b: Altura de la guía.
%      a: ancho de la guía.
%      R1: distancia
%      N: Número de puntos
%      phi: corte Phi
%      frec: frecuencia de trabajo
%      ilu:iluminación
%  SALIDA:
%      thetailu: theta en radianes a un determinado nivel ilu dB
%      anchohaz: Ancho de haz a X dBs
%      centrofase: Centro de fase
```

**5.3.2.6 Método aperturaSectorialH2primerppio(sectorialh)**

```
function aperturaSectorialH2primerppio(sectorialh)
%DESCRIPCION:
%Método que calcula la Directividad y el Spillover para el primer
principio de equivalencia
%e integrando en el diagrama. Además genera un archivo llamado bocina.mat
%que concentrará toda la información relativa a nuestra bocina.
%PARAMETROS
%  ENTRADA:
%      ancho: anchura de la Apertura.
%      b: Altura de la guía.
%      a: ancho de la guía.
%      R1: distancia
%      N: Número de puntos
%      phi: corte Phi
%      frec: frecuencia de trabajo
%      ilu:iluminación
%      Nintegracion:Número de puntos empleados para dibujar el diagrama
```

```
%      thetailu: theta en radianes a un determinado nivel ilu dB
%  SALIDA:
%      Directividad: Directividad
%      Spillover: Spillover
```

### **5.3.2.7 Método aperturaSectorialH2segundoppio (sectorialh)**

```
function aperturaSectorialH2segundoppio(sectorialh)
%DESCRIPCION:
%Método que calcula la Directividad y el Spillover para el segundo
principio de equivalencia
%e integrando en el diagrama. Además genera un archivo llamado bocina.mat
%que concentrará toda la información relativa a nuestra bocina.
%PARAMETROS
%  ENTRADA:
%      ancho: anchura de la Apertura.
%      b: Altura de la guía.
%      a: ancho de la guía.
%      R1: distancia
%      N: Número de puntos
%      phi: corte Phi
%      frec: frecuencia de trabajo
%      ilu:iluminación
%      Nintegracion:Número de puntos empleados para dibujar el diagrama
%      thetailu: theta en radianes a un determinado nivel ilu dB
%  SALIDA:
%      Directividad: Directividad
%      Spillover: Spillover
```

### **5.3.2.8 Método aperturaSectorialH2apertura1(sectorialh)**

```
function aperturaSectorialH2apertura1(sectorialh)
%DESCRIPCION:
%Método que calcula la Directividad y el Spillover para el primer
principio de equivalencia
%e integrando en la apertura. Además genera un archivo llamado bocina.mat
%que concentrará toda la información relativa a nuestra bocina.
%PARAMETROS
%  ENTRADA:
%      ancho: anchura de la Apertura.
%      b: Altura de la guía.
%      a: ancho de la guía.
%      R1: distancia
%      N: Número de puntos
```

```
% phi: corte Phi
% frec: frecuencia de trabajo
% ilu:iluminación
% Nintegracion:Número de puntos empleados para dibujar el diagrama
% thetailu: theta en radianes a un determinado nivel ilu dB
% SALIDA:
% Directividad: Directividad
% Spillover: Spillover
```

### **5.3.2.9 Método aperturaSectorialH2apertura2 (sectorialh)**

```
function aperturaSectorialH2apertura2(sectorialh)
%DESCRIPCION:
%Método que calcula la Directividad y el Spillover para el segundo
principio de equivalencia
%e integrando en la apertura. Además genera un archivo llamado bocina.mat
%que concentrará toda la información relativa a nuestra bocina.
%PARAMETROS
% ENTRADA:
% ancho: anchura de la Apertura.
% b: Altura de la guía.
% a: ancho de la guía.
% R1: distancia
% N: Número de puntos
% phi: corte Phi
% frec: frecuencia de trabajo
% ilu:iluminación
% Nintegracion:Número de puntos empleados para dibujar el diagrama
% thetailu: theta en radianes a un determinado nivel ilu dB
% SALIDA:
% Directividad: Directividad
% Spillover: Spillover
```

### 5.3.3 @PIRAMIDAL

#### 5.3.3.1 Constructor de la clase PIRAMIDAL

```
function pir=PIRAMIDAL
%DESCRIPCION:
% Constructor de la clase PIRAMIDAL
%PARAMETROS
%   ENTRADA:
%   ancho: anchura de la Apertura.
%   altura: altura de la Apertura.
%   b: Altura de la guía.
%   a: ancho de la guía.
%   R1: Distancia del vértice que es generado al abocinar la guía plano H
con respecto al centro de la Apertura.
%   R2: Distancia del vértice que es generado al abocinar la guía plano E
con respecto al centro de la Apertura.
%   Phi: Es el corte Phi. Permite conocer en qué plano deseamos realizar
el análisis del diagrama de radiación.
%   ilu: Es un número expresado en dB que se utilizará posteriormente para
conocer el ancho de banda a un cierto nivel de iluminación.
%   Frec: Es una variable que permite seleccionar la frecuencia de trabajo
en GHz.
%   N: Número de puntos que tiene la apertura.
%   Nintegracion: Número de puntos empleados para dibujar posteriormente
el
%   diagrama de radiación.
%   SALIDA:
%   pir: Objeto de tipo PIRAMIDAL
```

#### 5.3.3.2 Método *lgwtRectangular2(piramidal)*

```
function [x2,y2,w1,w2]=lgwtRectangular2(piramidal)
%DESCRIPCION:
% método que permite calcular los puntos y pesos utilizando
%la cuadratura de Gauss-Legendre en el intervalo [a,b]
%PARAMETROS
%   ENTRADA:
%       N: Número de puntos
%   ancho: anchura de la Apertura.
%   altura: Altura de la Apertura.
%   SALIDA:
%       x2: puntos en coordenada x
%       y2: puntos en coordenada y
%       w1: pesos en x
```

```
%      w2: pesos en y
```

### **5.3.3.3 Método puntosypeso(piramidal)**

```
function puntoypeso(piramidal)
%DESCRIPCION:
%Método que recoge los datos de salida del método
lgwtRectangular2(piramidal)
%y los procesa para convertir los puntos y pesos en una matriz de NxN.
%PARAMETROS
%  ENTRADA:
%      x2: puntos en coordenada x
%      y2: puntos en coordenada y
%      w1: pesos en x
%      w2: pesos en y
%      N: Número de puntos
%  SALIDA:
%      x: puntos en coordenada x
%      y: puntos en coordenada y
%      w: pesos
```

### **5.3.3.4 Método iluminacion1(piramidal)**

```
function [thetailu anchohaz centrofase]=iluminacion1(piramidal)
%DESCRIPCION:
%Método que calcula el centro de fase y el ancho de haz a un cierto
%nivel de iluminación cuando analizas el primer principio de equivalencia.
%Además, se ha añadido como salida la variable thetailu que es la theta
expresada
%en radianes a un cierto nivel de iluminación.
%PARAMETROS
%  ENTRADA:
%      ancho: anchura de la Apertura
%      altura: altura de la Apertura
%      b: Altura de la guía.
%      a: ancho de la guía.
%      R1: Distancia
%      R2: Distancia
%      N: Número de puntos
%      phi: corte Phi
%      frec: frecuencia de trabajo
%      ilu:iluminación
%  SALIDA:
%      thetailu: theta en radianes a un determinado nivel ilu dB
%      anchohaz: Ancho de haz a X dBs
```



```
%      centrofase: Centro de fase
```

### **5.3.3.5 Método iluminacion2(piramidal)**

```
function [thetailu anchohaz centrofase]=iluminacion2(piramidal)
%DESCRIPCION:
%Método que calcula el centro de fase y el ancho de haz a un cierto
%nivel de iluminación cuando analizas el segundo principio de
equivalencia.
%Además, se ha añadido como salida la variable thetailu que es la theta
expresada
%en radianes a un cierto nivel de iluminación.
%PARAMETROS
%  ENTRADA:
%      ancho: anchura de la Apertura
%      altura: altura de la Apertura
%      b: Altura de la guía.
%      a: ancho de la guía.
%      R1: Distancia
%      R2: Distancia
%      N: Número de puntos
%      phi: corte Phi
%      frec: frecuencia de trabajo
%      ilu:iluminación
%  SALIDA:
%      thetailu: theta en radianes a un determinado nivel ilu dB
%      anchohaz: Ancho de haz a X dBs
%      centrofase: Centro de fase
```

### **5.3.3.6 Método aperturaPiramidalprimerppio(piramidal)**

```
function aperturaPiramidalprimerppio(piramidal)
%DESCRIPCION:
%Método que calcula la Directividad y el Spillover para el primer
principio de equivalencia
%e integrando en el diagrama. Además genera un archivo llamado bocina.mat
%que concentrará toda la información relativa a nuestra bocina.
%PARAMETROS
%  ENTRADA:
%      ancho: anchura de la Apertura.
%      altura: altura de la Apertura.
%      b: Altura de la guía.
%      a: ancho de la guía.
%      R1: Distancia
%      R2: Distancia
```

```
%      N: Número de puntos
%      phi: corte Phi
%      frec: frecuencia de trabajo
%      ilu:iluminación
%      Nintegracion:Número de puntos empleados para dibujar el diagrama
%      thetailu: theta en radianes a un determinado nivel ilu dB
%  SALIDA:
%      Directividad: Directividad
%      Spillover: Spillover
```

### **5.3.3.7 Método aperturaPiramidalsegundoppio (piramidal)**

```
function aperturaPiramidalsegundoppio(piramidal)
%Método que calcula la Directividad y el Spillover para el segundo
principio de equivalencia
%e integrando en el diagrama. Además genera un archivo llamado bocina.mat
%que concentrará toda la información relativa a nuestra bocina.
%PARAMETROS
%  ENTRADA:
%      ancho: anchura de la Apertura.
%      altura: altura de la Apertura.
%      b: Altura de la guía.
%      a: ancho de la guía.
%      R1: Distancia
%      R2: Distancia
%      N: Número de puntos
%      phi: corte Phi
%      frec: frecuencia de trabajo
%      ilu:iluminación
%      Nintegracion:Número de puntos empleados para dibujar el diagrama
%      thetailu: theta en radianes a un determinado nivel ilu dB
%  SALIDA:
%      Directividad: Directividad
%      Spillover: Spillover
```

### **5.3.3.8 Método aperturaPiramidal1apertura(piramidal)**

```
function aperturaPiramidallapertura(piramidal)
%Método que calcula la Directividad y el Spillover para el primer
principio de equivalencia
%e integrando en la apertura. Además genera un archivo llamado bocina.mat
%que concentrará toda la información relativa a nuestra bocina.
%PARAMETROS
%  ENTRADA:
%      ancho: anchura de la Apertura.
```

```
% altura: altura de la Apertura.
% b: Altura de la guía.
% a: ancho de la guía.
% R1: Distancia
% R2: Distancia
% N: Número de puntos
% phi: corte Phi
% frec: frecuencia de trabajo
% ilu:iluminación
% Nintegracion:Número de puntos empleados para dibujar el diagrama
% thetailu: theta en radianes a un determinado nivel ilu dB
% SALIDA:
% Directividad: Directividad
% Spillover: Spillover
```

### **5.3.3.9 Método aperturaPiramidal2apertura (piramidal)**

```
function aperturaPiramidal2apertura(piramidal)
%DESCRIPCION:
%Método que calcula la Directividad y el Spillover para el segundo
principio de equivalencia
%e integrando en la apertura. Además genera un archivo llamado bocina.mat
%que concentrará toda la información relativa a nuestra bocina.
%PARAMETROS
% ENTRADA:
% ancho: anchura de la Apertura.
% altura: altura de la Apertura.
% b: Altura de la guía.
% a: ancho de la guía.
% R1: Distancia
% R2: Distancia
% N: Número de puntos
% phi: corte Phi
% frec: frecuencia de trabajo
% ilu:iluminación
% Nintegracion:Número de puntos empleados para dibujar el diagrama
% thetailu: theta en radianes a un determinado nivel ilu dB
% SALIDA:
% Directividad: Directividad
% Spillover: Spillover
```

### 5.3.4 @CIRCULAR

#### 5.3.4.1 Constructor de la clase CIRCULAR

```
function cir=CIRCULAR
%DESCRIPCION:
% Constructor de la clase CIRCULAR
%PARAMETROS
% ENTRADA:
% Rg: Radio de la guía.
% Ra: Radio de la apertura.
% Rc: Distancia del vértice que es generado al abocinar la guía con
respecto al centro de la Apertura.
% Phi: Es el corte Phi. Permite conocer en qué plano deseamos realizar
el análisis del diagrama de radiación.
% ilu: Es un número expresado en dB que se utilizará posteriormente para
conocer el ancho de banda a un cierto nivel de iluminación.
% Frec: Es una variable que permite seleccionar la frecuencia de trabajo
en GHz.
% N: Número de puntos que tiene la apertura.
% Nintegracion: Número de puntos empleados para dibujar posteriormente
el
% diagrama de radiación.
% SALIDA:
% cir: Objeto de tipo CIRCULAR
```

#### 5.3.4.2 Método lgwtCircular(circular)

```
function [x,w]=lgwtCircular(circular)
%DESCRIPCION:
% método que permite calcular los puntos y pesos utilizando
%la cuadratura de Gauss-Legendre en el intervalo [a,b]
%PARAMETROS
% ENTRADA:
% N: Número de puntos
% SALIDA:
% x: puntos en coordenada x
% w: pesos en coordenada x
```

**5.3.4.3 Método puntosypeso(circular)**

```
function puntoypeso(circular)
%DESCRIPCION:
%Método que recoge los datos de salida del método lgwtCircular (circular)
Y
%los procesa para convertir los puntos y pesos en una matriz de NxN.
%PARAMETROS
%   ENTRADA:
%       x: puntos en coordenada x
%       w: pesos en coordenada x
%       N: Número de puntos
%       Ra: Radio de la apertura
%   SALIDA:
%       x: puntos en coordenada x
%       y: puntos en coordenada y
%       w: pesos
```

**5.3.4.4 Método iluminacion1(circular)**

```
function [thetailu anchohaz centrofase]=iluminacion1(circular)
%DESCRIPCION:
%Método que calcula el centro de fase y el ancho de haz a un cierto
%nivel de iluminación cuando analizas el primer principio de equivalencia.
%Además, se ha añadido como salida la variable thetailu que es la theta
expresada
%en radianes a un cierto nivel de iluminación.
%PARAMETROS
%   ENTRADA:
%       Rg: Radio de la guía.
%       Ra: Radio de la apertura.
%       Rc: Distancia
%       phi: Es el corte Phi.
%       ilu: iluminación.
%       Frec: Es la frecuencia de trabajo en GHz.
%       N: Número de puntos.
%   SALIDA:
%       thetailu: theta en radianes a un determinado nivel ilu dB
%       anchohaz: Ancho de haz a X dBs
%       centrofase: Centro de fase
```

**5.3.4.5 Método iluminacion2(circular)**

```
function [thetailu anchohaz centrofase]=iluminacion2(circular)
%DESCRIPCION:
%Método que calcula el centro de fase y el ancho de haz a un cierto
%nivel de iluminación cuando analizas el segundo principio de
equivalencia.
%Además, se ha añadido como salida la variable thetailu que es la theta
expresada
%en radianes a un cierto nivel de iluminación.
%PARAMETROS
%  ENTRADA:
%      Rg: Radio de la guía.
%      Ra: Radio de la apertura.
%      Rc: Distancia
%      phi: Es el corte Phi.
%      ilu: iluminación.
%      Frec: Es la frecuencia de trabajo en GHz.
%      N: Número de puntos.
%  SALIDA:
%      thetailu: theta en radianes a un determinado nivel ilu dB
%      anchohaz: Ancho de haz a X dBs
%      centrofase: Centro de fase
```

**5.3.4.6 Método aperturaCircular1ppio(circular)**

```
function aperturaCircular1ppio(circular)
%DESCRIPCION:
%Método que calcula la Directividad y el Spillover para el primer
principio de equivalencia
%e integrando en el diagrama. Además genera un archivo llamado bocina.mat
%que concentrará toda la información relativa a nuestra bocina.
%PARAMETROS
%  ENTRADA:
%      Rg: Radio de la guía.
%      Ra: Radio de la apertura.
%      Rc: Distancia
%      Phi: Es el corte Phi.
%      ilu: iluminación.
%      Frec: Es la frecuencia de trabajo en GHz.
%      N: Número de puntos.
%      Nintegracion: Número de puntos diagrama.
%      thetailu: theta en radianes a un determinado nivel ilu dB
%  SALIDA:
```

```
% Directividad: Directividad
% Spillover: Spillover
```

### **5.3.4.7 Método aperturaCircular2ppio(circular)**

```
function aperturaCircular2ppio(circular)
%DESCRIPCION:
%Método que calcula la Directividad y el Spillover para el segundo
principio de equivalencia
%e integrando en el diagrama. Además genera un archivo llamado bocina.mat
%que concentrará toda la información relativa a nuestra bocina.
%PARAMETROS
% ENTRADA:
% Rg: Radio de la guía.
% Ra: Radio de la apertura.
% Rc: Distancia
% Phi: Es el corte Phi.
% ilu: iluminación.
% Frec: Es la frecuencia de trabajo en GHz.
% N: Número de puntos.
% Nintegracion: Número de puntos diagrama.
% thetailu: theta en radianes a un determinado nivel ilu dB
% SALIDA:
% Directividad: Directividad
% Spillover: Spillover
```

### **5.3.4.8 Método aperturaCircular1apertura(circular)**

```
function aperturaCircular1apertura(circular)
%DESCRIPCION:
%Método que calcula la Directividad y el Spillover para el primer
principio de equivalencia
%e integrando en la apertura. Además genera un archivo llamado bocina.mat
%que concentrará toda la información relativa a nuestra bocina.
%PARAMETROS
% ENTRADA:
% Rg: Radio de la guía.
% Ra: Radio de la apertura.
% Rc: Distancia
% Phi: Es el corte Phi.
% ilu: iluminación.
% Frec: Es la frecuencia de trabajo en GHz.
% N: Número de puntos.
% Nintegracion: Número de puntos diagrama.
% thetailu: theta en radianes a un determinado nivel ilu dB
```

```
% SALIDA:  
%     Directividad: Directividad  
%     Spillover: Spillover
```

#### **5.3.4.9 Método aperturaCircular2apertura(circular)**

```
function aperturaCircular2apertura(circular)  
%DESCRIPCION:  
%Método que calcula la Directividad y el Spillover para el segundo  
principio de equivalencia  
%e integrando en la apertura. Además genera un archivo llamado bocina.mat  
%que concentrará toda la información relativa a nuestra bocina.  
%PARAMETROS  
% ENTRADA:  
% Rg: Radio de la guía.  
% Ra: Radio de la apertura.  
% Rc: Distancia  
% Phi: Es el corte Phi.  
% ilu: iluminación.  
% Frec: Es la frecuencia de trabajo en GHz.  
% N: Número de puntos.  
% Nintegracion: Número de puntos diagrama.  
%   thetailu: theta en radianes a un determinado nivel ilu dB  
% SALIDA:  
%     Directividad: Directividad  
%     Spillover: Spillover
```



### 5.3.5 @CIRCULARCORRU

#### 5.3.5.1 Constructor de la clase CIRCULARCORRU

```
function circo=CIRCULARCORRU
%DESCRIPCION:
% Constructor de la clase CIRCULARCORRU
%PARAMETROS
%  ENTRADA:
%  Rg: Radio de la guía.
%  Ra: Radio de la apertura.
%  Rc: Distancia del vértice que es generado al abocinar la guía con
respecto al centro de la Apertura.
%  Phi: Es el corte Phi. Permite conocer en qué plano deseamos realizar
el análisis del diagrama de radiación.
%  ilu: Es un número expresado en dB que se utilizará posteriormente para
conocer el ancho de banda a un cierto nivel de iluminación.
%  Frec: Es una variable que permite seleccionar la frecuencia de trabajo
en GHz.
%  N: Número de puntos que tiene la apertura.
%  Nintegracion: Número de puntos empleados para dibujar posteriormente
el diagrama de radiación.
%  SALIDA:
%  circo: Objeto de tipo CIRCULARCORRU
```

#### 5.3.5.2 Método lgwtCircular(circularcorru)

```
function [x,w]=lgwtCircular(circularcorru)
%DESCRIPCION:
% método que permite calcular los puntos y pesos utilizando
%la cuadratura de Gauss-Legendre en el intervalo [a,b]
%PARAMETROS
%  ENTRADA:
%      N: Número de puntos
%  SALIDA:
%      x: puntos en coordenada x
%      w: pesos en coordenada x
```

#### 5.3.5.3 Método puntosypeso(pircorru)

```
function puntoypeso(circularcorru)
%Método que recoge los datos de salida del método lgwtCircular
(circularcorru) y
%los procesa para convertir los puntos y pesos en una matriz de NxN.
```

```
%PARAMETROS
%   ENTRADA:
%       x: puntos en coordenada x
%       w: pesos en coordenada x
%       N: Número de puntos
%       Ra: Radio de la apertura
%   SALIDA:
%       x: puntos en coordenada x
%       y: puntos en coordenada y
%       w: pesos
```

### **5.3.5.4 Método iluminacion1(circularcorru)**

```
function [thetailu anchohaz centrofase]=iluminacion1(circularcorru)
%DESCRIPCION:
%Método que calcula el centro de fase y el ancho de haz a un cierto
%nivel de iluminación cuando analizas el primer principio de equivalencia.
%Además, se ha añadido como salida la variable thetailu que es la theta
expresada
%en radianes a un cierto nivel de iluminación.
%PARAMETROS
%   ENTRADA:
%       Rg: Radio de la guía.
%       Ra: Radio de la apertura.
%       Rc: Distancia
%       phi: Es el corte Phi.
%       ilu: iluminación.
%       Frec: Es la frecuencia de trabajo en GHz.
%       N: Número de puntos.
%   SALIDA:
%       thetailu: theta en radianes a un determinado nivel ilu dB
%       anchohaz: Ancho de haz a X dBs
%       centrofase: Centro de fase
```

### **5.3.5.5 Método iluminacion2(circularcorru)**

```
function [thetailu anchohaz centrofase]=iluminacion2(circularcorru)
%DESCRIPCION:
%Método que calcula el centro de fase y el ancho de haz a un cierto
%nivel de iluminación cuando analizas el segundo principio de
equivalencia.
%Además, se ha añadido como salida la variable thetailu que es la theta
expresada
%en radianes a un cierto nivel de iluminación.
%PARAMETROS
```

```
% ENTRADA:
%   Rg: Radio de la guía.
%   Ra: Radio de la apertura.
%   Rc: Distancia
%   phi: Es el corte Phi.
%   ilu: iluminación.
%   Frec: Es la frecuencia de trabajo en GHz.
%   N: Número de puntos.
% SALIDA:
%   thetailu: theta en radianes a un determinado nivel ilu dB
%   anchohaz: Ancho de haz a X dBs
%   centrofase: Centro de fase
```

### **5.3.5.6 Método aperturaCircularCorrugada1ppio(circularcorru)**

```
function aperturaCircularCorrugada1ppio(circularcorru)
%DESCRIPCION:
%Método que calcula la Directividad y el Spillover para el primer
principio de equivalencia
%e integrando en el diagrama. Además genera un archivo llamado bocina.mat
%que concentrará toda la información relativa a nuestra bocina.
%PARAMETROS
% ENTRADA:
%   Rg: Radio de la guía.
%   Ra: Radio de la apertura.
%   Rc: Distancia
%   Phi: Es el corte Phi.
%   ilu: iluminación.
%   Frec: Es la frecuencia de trabajo en GHz.
%   N: Número de puntos.
%   Nintegracion: Número de puntos diagrama.
%   thetailu: theta en radianes a un determinado nivel ilu dB
% SALIDA:
%   Directividad: Directividad
%   Spillover: Spillover
```

### **5.3.5.7 Método aperturaCircularCorrugada2ppio(circularcorru)**

```
function aperturaCircularCorrugada2ppio(circularcorru)
%DESCRIPCION:
%Método que calcula la Directividad y el Spillover para el segundo
principio de equivalencia
%e integrando en el diagrama. Además genera un archivo llamado bocina.mat
%que concentrará toda la información relativa a nuestra bocina.
```

```
%PARAMETROS
%  ENTRADA:
%  Rg: Radio de la guía.
%  Ra: Radio de la apertura.
%  Rc: Distancia
%  Phi: Es el corte Phi.
%  ilu: iluminación.
%  Frec: Es la frecuencia de trabajo en GHz.
%  N: Número de puntos.
%  Nintegracion: Número de puntos diagrama.
%  thetailu: theta en radianes a un determinado nivel ilu dB
%  SALIDA:
%  Directividad: Directividad
%  Spillover: Spillover
```

### **5.3.5.8 Método aperturaCircularCorrugada1apertura(circularcorru)**

```
function aperturaCircularCorrugada1apertura(circularcorru)
```

```
%DESCRIPCION:
%Método que calcula la Directividad y el Spillover para el primer
principio de equivalencia
%e integrando en la apertura. Además genera un archivo llamado bocina.mat
%que concentrará toda la información relativa a nuestra bocina.
%PARAMETROS
%  ENTRADA:
%  Rg: Radio de la guía.
%  Ra: Radio de la apertura.
%  Rc: Distancia
%  Phi: Es el corte Phi.
%  ilu: iluminación.
%  Frec: Es la frecuencia de trabajo en GHz.
%  N: Número de puntos.
%  Nintegracion: Número de puntos diagrama.
%  thetailu: theta en radianes a un determinado nivel ilu dB
%  SALIDA:
%  Directividad: Directividad
%  Spillover: Spillover
```

### **5.3.5.9 Método aperturaCircularCorrugada2apertura(circularcorru)**

```
function aperturaCircularCorrugada2apertura(circularcorru)
```

```
%DESCRIPCION:
%Método que calcula la Directividad y el Spillover para el segundo
principio de equivalencia
%e integrando en la apertura. Además genera un archivo llamado bocina.mat
```

```
%que concentrará toda la información relativa a nuestra bocina.
%PARAMETROS
%  ENTRADA:
%  Rg: Radio de la guía.
%  Ra: Radio de la apertura.
%  Rc: Distancia
%  Phi: Es el corte Phi.
%  ilu: iluminación.
%  Frec: Es la frecuencia de trabajo en GHz.
%  N: Número de puntos.
%  Nintegracion: Número de puntos diagrama.
%  thetailu: theta en radianes a un determinado nivel ilu dB
%  SALIDA:
%  Directividad: Directividad
%  Spillover: Spillover
```

### 5.3.6 @PIRCURRU

#### 5.3.6.1 Constructor de la clase PIRCURRU

```
function pirco=PIRCORRU
%DESCRIPCION:
% Constructor de la clase PIRCORRU
%PARAMETROS
%  ENTRADA:
%  ancho: anchura de la Apertura.
%  altura: altura de la Apertura.
%  b: Altura de la guía.
%  a: ancho de la guía.
%  R1: Distancia del vértice que es generado al abocinar la guía plano H
con respecto al centro de la Apertura.
%  R2: Distancia del vértice que es generado al abocinar la guía plano E
con respecto al centro de la Apertura.
%  Phi: Es el corte Phi. Permite conocer en qué plano deseamos realizar
el análisis del diagrama de radiación.
%  ilu: Es un número expresado en dB que se utilizará posteriormente para
conocer el ancho de banda a un cierto nivel de iluminación.
%  Frec: Es una variable que permite seleccionar la frecuencia de trabajo
en GHz.
%  N: Número de puntos que tiene la apertura.
%  Nintegracion: Número de puntos empleados para dibujar posteriormente
el
%  diagrama de radiación.
```

```
% SALIDA:  
% pirco: Objecto de tipo PIRCORRU
```

### **5.3.6.2 Método *lgwtRectangular2(pircorru)***

```
function [x2,y2,w1,w2]=lgwtRectangular2(pircorru)  
%DESCRIPCION:  
% método que permite calcular los puntos y pesos utilizando  
%la cuadratura de Gauss-Legendre en el intervalo [a,b]  
%PARAMETROS  
% ENTRADA:  
%     N: Número de puntos  
% ancho: anchura de la Apertura.  
% altura: Altura de la Apertura.  
% SALIDA:  
%     x2: puntos en coordenada x  
%     y2: puntos en coordenada y  
%     w1: pesos en x  
%     w2: pesos en y
```

### **5.3.6.3 Método *puntosypeso(pircorru)***

```
function puntosypeso(pircorru)  
%DESCRIPCION:  
%Método que recoge los datos de salida del método  
lgwtRectangular2(pircorru)  
%y los procesa para convertir los puntos y pesos en una matriz de NxN.  
%PARAMETROS  
% ENTRADA:  
%     x2: puntos en coordenada x  
%     y2: puntos en coordenada y  
%     w1: pesos en x  
%     w2: pesos en y  
%     N: Número de puntos  
% SALIDA:  
%     x: puntos en coordenada x  
%     y: puntos en coordenada y  
%     w: pesos
```

**5.3.6.4 Método iluminacion1(pircorru)**

```
function [thetailu anchohaz centrofase]=iluminacion1(pircorru)
%DESCRIPCION:
%Método que calcula el centro de fase y el ancho de haz a un cierto
%nivel de iluminación cuando analizas el primer principio de equivalencia.
%Además, se ha añadido como salida la variable thetailu que es la theta
expresada
%en radianes a un cierto nivel de iluminación.
%PARAMETROS
%  ENTRADA:
%      ancho: anchura de la Apertura
%      altura: altura de la Apertura
%      b: Altura de la guía.
%      a: ancho de la guía.
%      R1: Distancia
%      R2: Distancia
%      N: Número de puntos
%      phi: corte Phi
%      frec: frecuencia de trabajo
%      ilu:iluminación
%  SALIDA:
%      thetailu: theta en radianes a un determinado nivel ilu dB
%      anchohaz: Ancho de haz a X dBs
%      centrofase: Centro de fase
```

**5.3.6.5 Método iluminacion2(pircorru)**

```
function [thetailu anchohaz centrofase]=iluminacion2(pircorru)
%DESCRIPCION:
%Método que calcula el centro de fase y el ancho de haz a un cierto
%nivel de iluminación cuando analizas el segundo principio de
equivalencia.
%Además, se ha añadido como salida la variable thetailu que es la theta
expresada
%en radianes a un cierto nivel de iluminación.
%PARAMETROS
%  ENTRADA:
%      ancho: anchura de la Apertura
%      altura: altura de la Apertura
%      b: Altura de la guía.
%      a: ancho de la guía.
%      R1: Distancia
%      R2: Distancia
```

```
%      N: Número de puntos
%      phi: corte Phi
%      frec: frecuencia de trabajo
%      ilu:iluminación
%  SALIDA:
%      thetailu: theta en radianes a un determinado nivel ilu dB
%      anchohaz: Ancho de haz a X dBs
%      centrofase: Centro de fase
```

### **5.3.6.6 Método aperturaPiramidalCorrugadaprimerppio(pircorru)**

```
function aperturaPiramidalCorrugadaprimerppio(pircorru)
%DESCRIPCION:
%Método que calcula la Directividad y el Spillover para el primer
principio de equivalencia
%e integrando en el diagrama. Además genera un archivo llamado bocina.mat
%que concentrará toda la información relativa a nuestra bocina.
%PARAMETROS
%  ENTRADA:
%      ancho: anchura de la Apertura.
%      altura: altura de la Apertura.
%      b: Altura de la guía.
%      a: ancho de la guía.
%      R1: Distancia
%      R2: Distancia
%      N: Número de puntos
%      phi: corte Phi
%      frec: frecuencia de trabajo
%      ilu:iluminación
%      Nintegracion:Número de puntos empleados para dibujar el diagrama
%      thetailu: theta en radianes a un determinado nivel ilu dB
%  SALIDA:
%      Directividad: Directividad
%      Spillover: Spillover
```

### **5.3.6.7 Método aperturaPiramidalCorrugadasegundoppio(pircorru)**

```
function aperturaPiramidalCorrugadasegundoppio(pircorru)
%DESCRIPCION:
%Método que calcula la Directividad y el Spillover para el segundo
principio de equivalencia
%e integrando en el diagrama. Además genera un archivo llamado bocina.mat
%que concentrará toda la información relativa a nuestra bocina.
```



```
%PARAMETROS
%   ENTRADA:
%       ancho: anchura de la Apertura.
%       altura: altura de la Apertura.
%       b: Altura de la guía.
%       a: ancho de la guía.
%       R1: Distancia
%       R2: Distancia
%       N: Número de puntos
%       phi: corte Phi
%       frec: frecuencia de trabajo
%       ilu:iluminación
%       Nintegracion:Número de puntos empleados para dibujar el diagrama
%       thetailu: theta en radianes a un determinado nivel ilu dB
%   SALIDA:
%       Directividad: Directividad
%       Spillover: Spillover
```

### **5.3.6.8 Método aperturaPiramidalCorrugada1apertura(pircorru)**

```
function aperturaPiramidalCorrugadalapertura(pircorru)
%DESCRIPCION:
%Método que calcula la Directividad y el Spillover para el primer
principio de equivalencia
%e integrando en la apertura. Además genera un archivo llamado bocina.mat
%que concentrará toda la información relativa a nuestra bocina.
%PARAMETROS
%   ENTRADA:
%       ancho: anchura de la Apertura.
%       altura: altura de la Apertura.
%       b: Altura de la guía.
%       a: ancho de la guía.
%       R1: Distancia
%       R2: Distancia
%       N: Número de puntos
%       phi: corte Phi
%       frec: frecuencia de trabajo
%       ilu:iluminación
%       Nintegracion:Número de puntos empleados para dibujar el diagrama
%       thetailu: theta en radianes a un determinado nivel ilu dB
%   SALIDA:
%       Directividad: Directividad
%       Spillover: Spillover
```

### **5.3.6.9 Método aperturaPiramidalCorrugada2apertura(pircorru)**

```
function aperturaPiramidalCorrugada2apertura(pircorru)
%DESCRIPCION:
%Método que calcula la Directividad y el Spillover para el segundo
principio de equivalencia
% integrando en la apertura. Además genera un archivo llamado bocina.mat
%que concentrará toda la información relativa a nuestra bocina.
%PARAMETROS
%  ENTRADA:
%      ancho: anchura de la Apertura.
%      altura: altura de la Apertura.
%      b: Altura de la guía.
%      a: ancho de la guía.
%      R1: Distancia
%      R2: Distancia
%      N: Número de puntos
%      phi: corte Phi
%      frec: frecuencia de trabajo
%      ilu:iluminación
%      Nintegracion:Número de puntos empleados para dibujar el diagrama
%      thetailu: theta en radianes a un determinado nivel ilu dB
%  SALIDA:
%      Directividad: Directividad
%      Spillover: Spillover
```

## 5.4 COMPILADOR DE MATLAB. CREACIÓN DE EJECUTABLES.EXE

Una vez nuestro proyecto se haya finalizado, podemos distribuirlo a los usuarios como un ejecutable estándar de Windows (archivo.exe). Esta posibilidad es muy interesante ya que libera al usuario de tener que instalar el entorno de desarrollo Matlab y permite distribuir de manera sencilla nuestra aplicación.

Antes de comenzar a crear el ejecutable debemos comprobamos qué compilador hay en nuestro ordenador con el siguiente comando:

```
>>mbuild -setup
```

Te aparecerá un listado de compiladores que tiene nuestro equipo.

Select a compiler:

[1] Lcc-win32 C 2.4.1 in C:\ARCHIV~1\MATLAB\R2009b\sys\lcc

[2] Microsoft Visual C++ 6.0 in C:\Archivos de programa\Microsoft Visual Studio

Seleccionamos cualquiera de los dos, por ejemplo el primero que es el que te viene cuando instalas el programa MATLAB.

Una vez elegido el compilador pasamos a crear el ejecutable. Introducimos el comando `deploytool` para crear el archivo de la aplicación y nos aparece la siguiente ventana:

```
>>deploytool
```

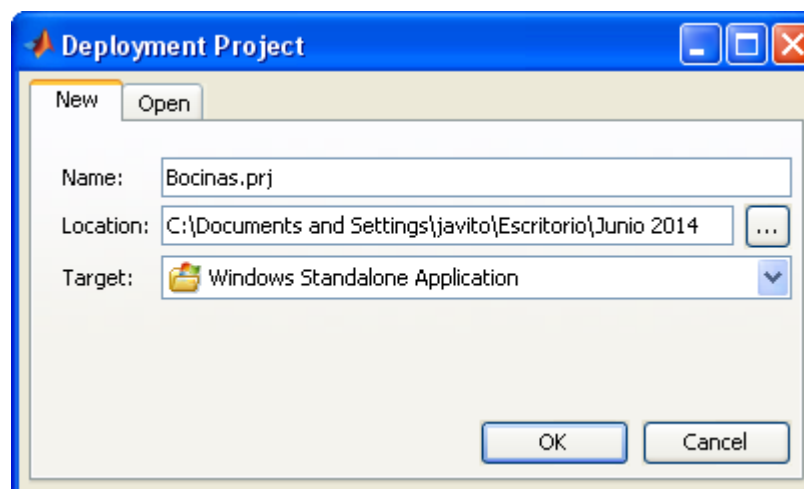


Fig 5.19 Creación de un proyecto

Aceptamos con el nombre y con la carpeta que desees y en Target seleccionamos Windows Standalone Application ya que se desea crear una aplicación en Windows.

Una vez hecho esto, automáticamente aparece la herramienta deploytool en el que tienes que introducir el archivo principal y los archivos secundarios del GUIDE incluyendo imágenes.

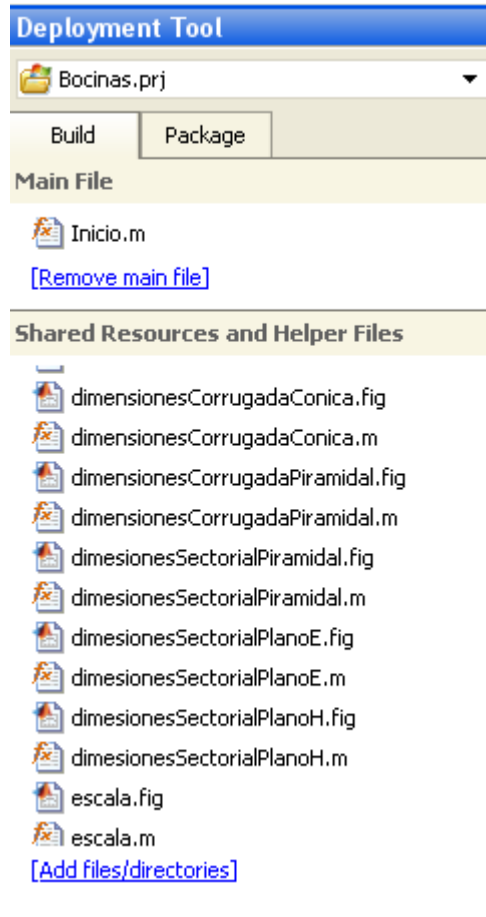


Fig 5.20 Creación del ejecutable

Para el empaquetamiento pasamos a la pestaña Package, y seleccionamos Add MCR para que funcione en cualquier ordenador que no tenga instalado Matlab. Sacará un ejecutable en el que una vez ejecutado instalará unas librerías propias de Matlab. En la siguiente figura se verá más claro:

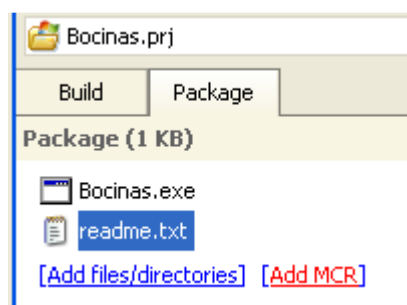


Fig 5.21 Creación del ejecutable(2)

Ahora lo que hay que hacer es compilar el programa, volvemos a la pestaña Build y pulsamos el siguiente botón:



Fig 5.22 Creación del ejecutable(3)

Una vez terminada la compilación pasamos a empaquetar el programa:



Fig 5.23 Creación del ejecutable(4)

Una vez empaquetado hay que instalar el MCRInstaller, y cuando termines se debe reiniciar el ordenador para su instalación óptima. A continuación ejecutas el archivo llamado nombreproyecto\_pkg.exe que haciendo dobleclick descomprimes el ejecutable deseado nombreproyecto.exe.

## 5.5 EJEMPLOS DE ANÁLISIS REALIZADOS EN SABOR

A continuación se presentan varios ejemplos de los seis tipos de bocinas con el software SABOR. En concreto, los ejemplos serán los siguientes:

### 5.5.1 Sectorial E

Parámetros de entrada:

Datos de entrada	
a	2.286
b	1.016
H	12
R2	23,15

Diagrama de radiación y resultados:

Para un ejemplo concreto mostraremos su diagrama de radiación:

- $\Phi=0$ ,
- Frecuencia= 10GHz
- 2º principio integrando en la apertura

Directividad:	13.8792
Ancho de banda -X dB:	59.4767
Centro de fase:	3.98778e-016
Spillover:	-5.02163

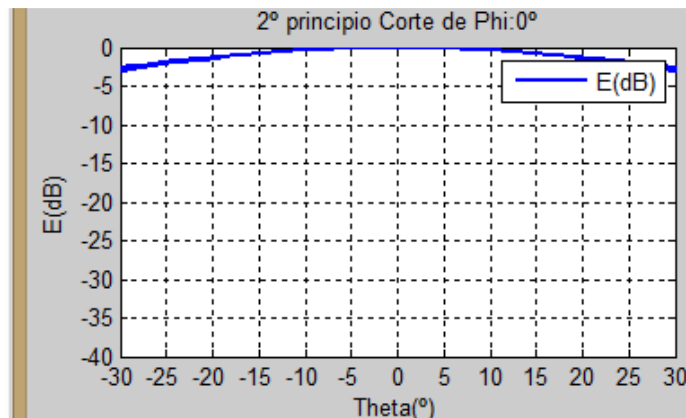


Fig 5.24 Diagrama y resultados sectorial plano E

A continuación mostramos los resultados en una tabla para  $\phi=0$  y  $\phi=90$ , la que está en la izquierda corresponde al análisis para una frecuencia de 10GHz. Por el contrario, en las tablas que están a la derecha corresponden a una frecuencia de 8GHz para ver su comportamiento:

INT. DIAGRAMA a 10 GHz

1er ppio	Phi = 0	Phi =90
Directividad(dB)	14,12	14,12
Centro de fase(cm)	0	13,48
Ancho de haz a -3dB(º)	78,70	13,36
Spillover(dB)	-1,78	-11,27

INT. DIAGRAMA a 8 GHz

1er ppio	Phi = 0	Phi =90
Directividad(dB)	13,01	13,01
Centro de fase(cm)	0	8,72
Ancho de haz a -3dB(º)	91,33	16,40
Spillover(dB)	-1,67	-10,61

Como hemos podido comprobar a mayor frecuencia, la bocina es más directiva.

INT. DIAGRAMA a 10GHz

2º ppio	Phi = 0	Phi =90
Directividad(dB)	12,00	12,00
Centro de fase(cm)	0,00	13,52
Ancho de haz a -3dB(º)	59,47	13,48
Spillover(dB)	-5,02	-13,32

INT. DIAGRAMA a 8GHz

2º ppio	Phi = 0	Phi =90
Directividad(dB)	11,07	11,07
Centro de fase(cm)	0,00	8,72
Ancho de haz a -3dB(º)	64,25	16,39
Spillover(dB)	-5,02	-12,55

Ahora vamos a ejecutar el análisis integrando en la apertura:

INT. APERTURA a 10GHz

1er ppio	Phi = 0	Phi =90
Directividad(dB)	13,88	13,88
Centro de fase(cm)	0	13,48
Ancho de haz a -3dB(º)	78,70	13,36
Spillover(dB)	-1,78	-11,27

INT.APERTURA a 8GHz

1er ppio	Phi = 0	Phi =90
Directividad(dB)	12,32	12,32
Centro de fase(cm)	0	8,72
Ancho de haz a 3dB(º)	91,33	16,40
Spillover(dB)	-1,67	-10,61

Como hemos podido comprobar a mayor frecuencia, la bocina es más directiva.

INT. APERTURA a 10GHz

2º ppio	Phi = 0	Phi =90
Directividad(dB)	13,88	13,88
Centro de fase(cm)	0,00	13,52
Ancho de haz a -3dB(º)	59,47	13,48
Spillover(dB)	-5,02	-13,32

INT.APERTURA a 8GHz

2º ppio	Phi = 0	Phi =90
Directividad(dB)	12,32	12,32
Centro de fase(cm)	8,72	8,72
Ancho de haz a 3dB(º)	16,40	16,39
Spillover(dB)	-10,61	-12,55

### 5.5.2 Sectorial H

Parámetros de entrada:

Datos de entrada	
a	2.286
b	1.016
w	15
R1	25

Diagrama de radiación y resultados:

Para un ejemplo concreto mostraremos su diagrama de radiación:

- $\Phi=0$ ,
- Frecuencia= 10GHz
- 2º principio integrando en la apertura

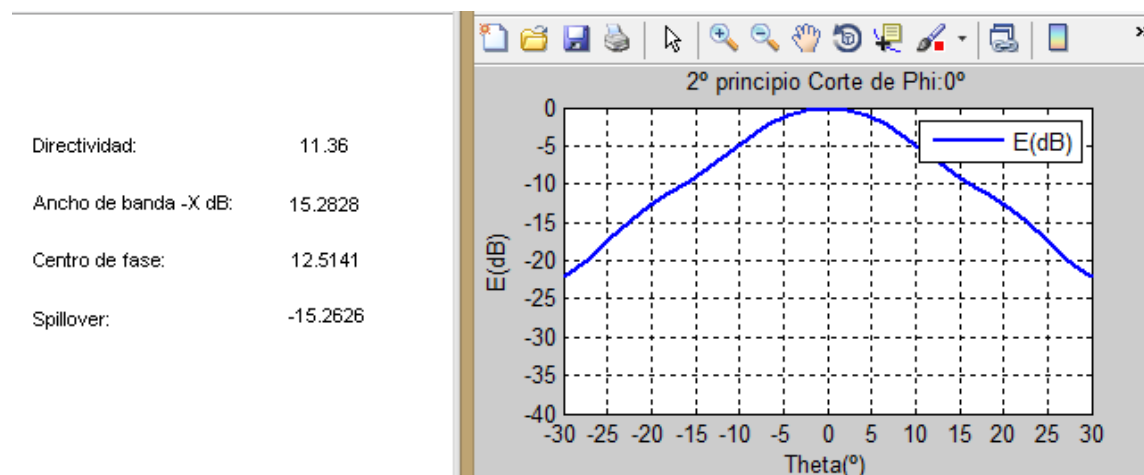


Fig 5.25 Diagrama y resultados sectorial plano H

A continuación mostramos los resultados en una tabla para  $\phi=0$  y  $\phi=90$ , la que está en la izquierda corresponde al análisis para una frecuencia de 10GHz. Por el contrario, en las tablas que están a la derecha corresponden a una frecuencia de 8GHz para ver su comportamiento:



INT. DIAGRAMA a 10GHz

1er ppio	Phi = 0	Phi =90
Directividad(dB)	13,02	13,02
Centro de fase(cm)	12,63	0
Ancho de haz a -3dB(°)	15,49	106,00
Spillover(dB)	-11,10	-1,38

INT. DIAGRAMA a 8GHz

1er ppio	Phi = 0	Phi =90
Directividad(dB)	12,19	12,19
Centro de fase(cm)	8,37	0
Ancho de haz a -3dB(°)	18,33	114,10
Spillover(dB)	-10,47	-1,28

Como hemos podido comprobar a mayor frecuencia, la bocina es más directiva.

INT. DIAGRAMA a 10GHz

2º ppio	Phi = 0	Phi =90
Directividad(dB)	8,96	8,96
Centro de fase(cm)	12,51	0
Ancho de haz a -3dB(°)	15,28	106,00
Spillover(dB)	-15,26	-4,94

INT. DIAGRAMA a 8GHz

2º ppio	Phi = 0	Phi =90
Directividad(dB)	8,09	8,09
Centro de fase(cm)	8,23	0
Ancho de haz a -3dB(°)	18,00	114,10
Spillover(dB)	-14,70	-4,79

Ahora vamos a ejecutar el análisis integrando en la apertura:

INT. APERTURA a 10GHz

1er ppio	Phi = 0	Phi =90
Directividad(dB)	11,36	11,36
Centro de fase(cm)	12,63	0
Ancho de haz a -3dB(°)	15,49	106,00
Spillover(dB)	-11,10	-1,38

INT. APERTURA a 8GHz

1er ppio	Phi = 0	Phi =90
Directividad(dB)	9,78	9,78
Centro de fase(cm)	8,37	0
Ancho de haz a -3dB(°)	18,33	114,10
Spillover(dB)	-10,47	-1,28

Como hemos podido comprobar a mayor frecuencia, la bocina es más directiva.

INT. APERTURA a 10GHz

2º ppio	Phi = 0	Phi =90
Directividad(dB)	11,36	11,36
Centro de fase(cm)	12,51	0
Ancho de haz a -3dB(°)	15,28	106,00
Spillover(dB)	-15,26	-4,94

INT. APERTURA a 8GHz

2º ppio	Phi = 0	Phi =90
Directividad(dB)	9,78	9,78
Centro de fase(cm)	8,23	0
Ancho de haz a -3dB(°)	18,00	114,10
Spillover(dB)	-14,70	-4,79

### 5.5.3 Sectorial Piramidal

Parámetros de entrada:

Datos de entrada	
a	2.286
b	1.016
W	15
H	12
R1	25
R2	23,15

Diagrama de radiación y resultados:

Para un ejemplo concreto mostraremos su diagrama de radiación:

- $\Phi=0$ ,
- Frecuencia= 10GHz
- 2º principio integrando en la apertura

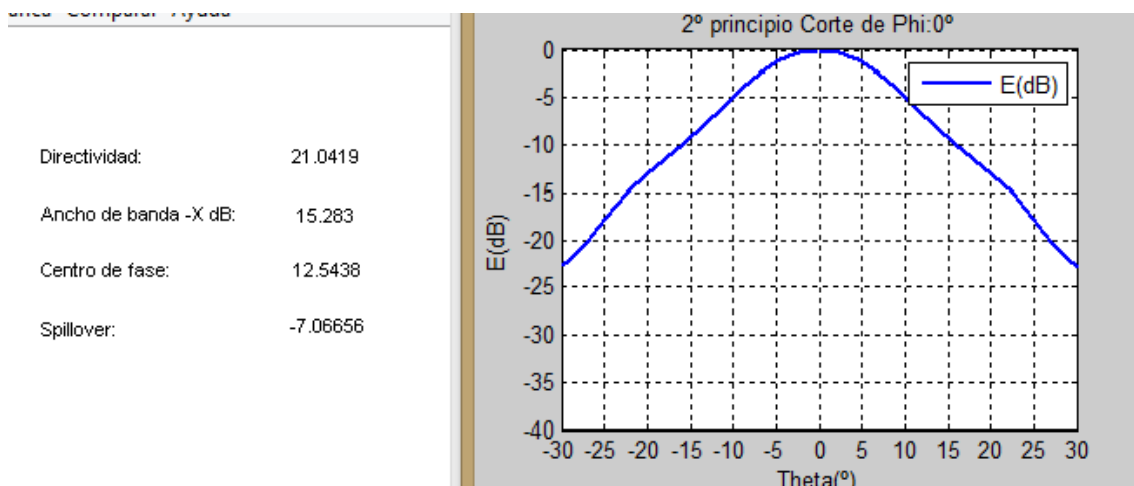


Fig 5.26 Diagrama y resultados sectorial piramidal

A continuación mostramos los resultados en una tabla para  $\phi=0$  y  $\phi=90$ , la que está en la izquierda corresponde al análisis para una frecuencia de 10GHz. Por el contrario, en las tablas que están a la derecha corresponden a una frecuencia de 8GHz para ver su comportamiento:

INT. DIAGRAMA a 10GHz

1er ppio	Phi = 0	Phi =90
Directividad(dB)	21,02	21,02
Centro de fase(cm)	12,59	13,52
Ancho de haz a -3dB(°)	15,49	13,50
Spillover(dB)	-3,90	-4,74

INT. DIAGRAMA a 8GHz

1er ppio	Phi = 0	Phi =90
Directividad(dB)	19,88	19,88
Centro de fase(cm)	8,3	8,72
Ancho de haz a -3dB(°)	18,33	16,39
Spillover(dB)	-3,55	-4,22

Como hemos podido comprobar a mayor frecuencia, la bocina es más directiva.

INT. DIAGRAMA a 10GHz

2º ppio	Phi = 0	Phi =90
Directividad(dB)	18,04	18,04
Centro de fase(cm)	12,54	13,63
Ancho de haz a -3dB(°)	15,28	13,48
Spillover(dB)	-7,06	-7,82

INT. DIAGRAMA a 8GHz

2º ppio	Phi = 0	Phi =90
Directividad(dB)	16,88	16,88
Centro de fase(cm)	8,3	8,72
Ancho de haz a -3dB(°)	18	16,40
Spillover(dB)	-6,74	-7,30

Ahora vamos a ejecutar el análisis integrando en la apertura:

INT. APERTURA a 10GHz

1er ppio	Phi = 0	Phi =90
Directividad(dB)	21,04	21,04
Centro de fase(cm)	12,59	13,52
Ancho de haz a -3dB(°)	15,49	13,50
Spillover(dB)	-3,90	-4,74

INT. APERTURA a 8GHz

1er ppio	Phi = 0	Phi =90
Directividad(dB)	19,84	19,84
Centro de fase(cm)	8,3	16,39
Ancho de haz a -3dB(°)	18,33	8,72
Spillover(dB)	-3,55	-4,22

Como hemos podido comprobar a mayor frecuencia, la bocina es más directiva.

INT. APERTURA a 10GHz

2º ppio	Phi = 0	Phi =90
Directividad(dB)	21,04	21,04
Centro de fase(cm)	12,54	13,63
Ancho de haz a -3dB(°)	15,28	13,48
Spillover(dB)	-7,07	-9,96

INT. APERTURA a 8GHz

2º ppio	Phi = 0	Phi =90
Directividad(dB)	19,84	19,84
Centro de fase(cm)	8,3	8,72
Ancho de haz a -3dB(°)	18	16,40
Spillover(dB)	-6,74	-7,30

### 5.5.4 Cónica

Parámetros de entrada:

Datos de entrada	
Rg	1.016
Ra	6
Rc	10

Diagrama de radiación y resultados:

Para un ejemplo concreto mostraremos su diagrama de radiación:

- $\Phi=0$ ,
- Frecuencia= 10GHz
- 2º principio integrando en la apertura

Directividad:	16.1821
Ancho de banda -X dB:	20.0043
Centro de fase:	9.54017
Spillover:	-9.46665

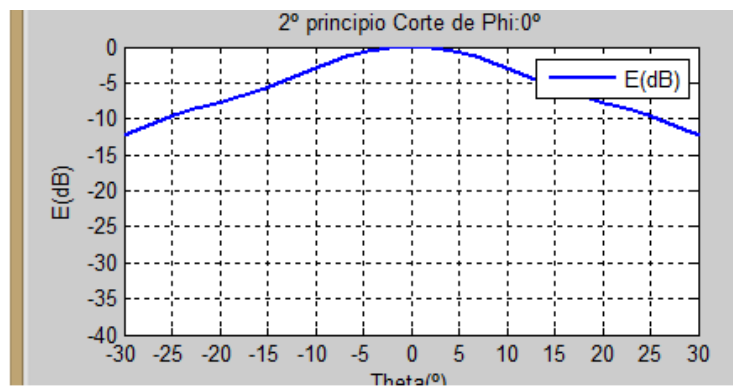


Fig 5.27 Diagrama y resultados cónica

A continuación mostramos los resultados en una tabla para  $\phi=0$  y  $\phi=90$ , la que está en la izquierda corresponde al análisis para una frecuencia de 10GHz. Por el contrario, en las tablas que están a la derecha corresponden a una frecuencia de 8GHz para ver su comportamiento:

INT. DIAGRAMA a 10GHz

1er ppio	Phi = 0	Phi =90
Directividad(dB)	16,32	16,21
Centro de fase(cm)	9,56	16,63
Ancho de haz a - 3dB(º)	20,51	28,27
Spillover(dB)	-6,19	-4,26

INT. DIAGRAMA a 8GHz

1er ppio	Phi = 0	Phi =90
Directividad(dB)	16,38	16,38
Centro de fase(cm)	5,59	12,18
Ancho de haz a - 3dB(º)	24,06	21,90
Spillover(dB)	-4,88	-5,45

INT. DIAGRAMA a 10GHz

2º ppio	Phi = 0	Phi =90
Directividad(dB)	13,20	13,20
Centro de fase(cm)	9,56	16,63
Ancho de haz a - 3dB(º)	20,0	28,32
Spillover(dB)	-9,47	-7,36

INT. DIAGRAMA a 8GHz

2º ppio	Phi = 0	Phi =90
Directividad(dB)	13,30	13,30
Centro de fase(cm)	5,55	12,18
Ancho de haz a - 3dB(º)	23,3	21,90
Spillover(dB)	-8,2	-8,54

Ahora vamos a ejecutar el análisis integrando en la apertura:

INT. APERTURA a 10GHz

1er ppio	Phi = 0	Phi =90
Directividad(dB)	16,18	16,18
Centro de fase(cm)	9,56	16,63
Ancho de haz a - 3dB(º)	20,51	28,27
Spillover(dB)	-6,19	-4,26

INT. APERTURA a 8GHz

1er ppio	Phi = 0	Phi =90
Directividad(dB)	16,24	16,24
Centro de fase(cm)	5,59	12,18
Ancho de haz a 3dB(º)	24,06	21,90
Spillover(dB)	-4,88	-5,45

INT. APERTURA a 10GHz

2º ppio	Phi = 0	Phi =90
Directividad(dB)	16,18	16,20
Centro de fase(cm)	9,56	16,63
Ancho de haz a - 3dB(º)	20,00	28,27
Spillover(dB)	-9,47	-7,37

INT. APERTURA a 8GHz

2º ppio	Phi = 0	Phi =90
Directividad(dB)	16,24	16,24
Centro de fase(cm)	5,55	12,18
Ancho de haz a 3dB(º)	23,3	21,90
Spillover(dB)	-8,2	-8,54

### 5.5.5 Circular corrugada

Parámetros de entrada:

Datos de entrada	
Rg	1.016
Ra	6
Rc	10

Diagrama de radiación y resultados:

Para un ejemplo concreto mostraremos su diagrama de radiación:

- $\Phi=0$ ,
- Frecuencia= 10GHz
- 2º principio integrando en la apertura

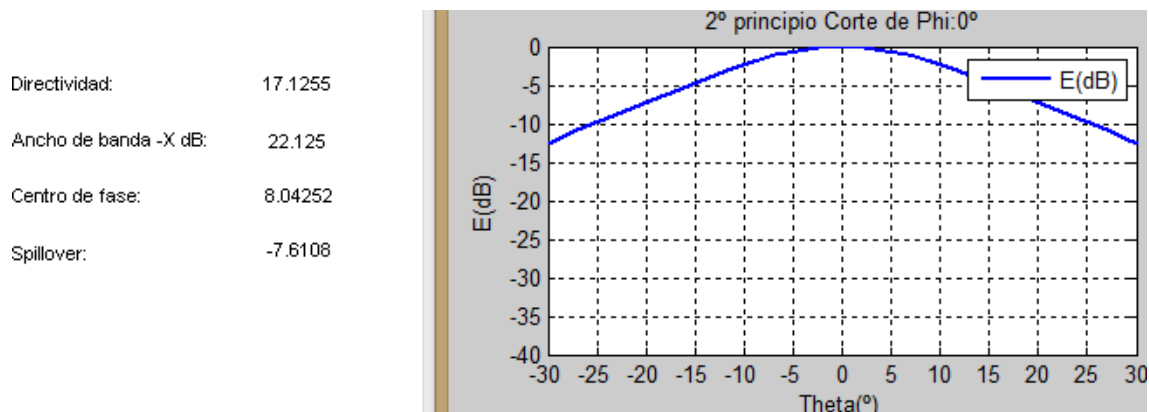


Fig 5.28 Diagrama y resultados corrugada cónica

A continuación mostramos los resultados en una tabla para  $\phi=0$  y  $\phi=90$ , la que está en la izquierda corresponde al análisis para una frecuencia de 10GHz. Por el contrario, en las tablas que están a la derecha corresponden a una frecuencia de 8GHz para ver su comportamiento:

INT. DIAGRAMA a 10GHz

1er ppio	Phi = 0	Phi =90
Directividad(dB)	17,10	17,10
Centro de fase(cm)	8,05	8,05
Ancho de haz a -3dB(°)	22,82	22,83
Spillover(dB)	-4,41	-4,41

INT. DIAGRAMA a 8GHz

1er ppio	Phi = 0	Phi =90
Directividad(dB)	16,34	16,34
Centro de fase(cm)	5,27	5,27
Ancho de haz a -3dB(°)	26,00	26,00
Spillover(dB)	-4,02	-4,02

Como hemos podido comprobar a mayor frecuencia, la bocina es más directiva.

INT. DIAGRAMA a 10GHz

2º ppio	Phi = 0	Phi =90
Directividad(dB)	14,09	14,09
Centro de fase(cm)	8,04	8,05
Ancho de haz a -3dB(°)	22,13	22,82
Spillover(dB)	-7,61	-7,42

INT. DIAGRAMA a 8GHz

2º ppio	Phi = 0	Phi =90
Directividad(dB)	13,33	13,33
Centro de fase(cm)	5,24	5,37
Ancho de haz a -3dB(°)	25,0	26,0
Spillover(dB)	-7,26	-70,03

Ahora vamos a ejecutar el análisis integrando en la apertura:

INT. APERTURA a 10GHz

1er ppio	Phi = 0	Phi =90
Directividad(dB)	17,12	17,13
Centro de fase(cm)	8,05	8,05
Ancho de haz a -3dB(°)	22,82	22,83
Spillover(dB)	-4,41	-4,41

INT. APERTURA a 8GHz

1er ppio	Phi = 0	Phi =90
Directividad(dB)	16,37	16,37
Centro de fase(cm)	5,27	5,27
Ancho de haz a -3dB(°)	26,00	26,00
Spillover(dB)	-4,02	-4,02

Como hemos podido comprobar a mayor frecuencia, la bocina es más directiva.

INT. APERTURA a 10GHz

2º ppio	Phi = 0	Phi =90
Directividad(dB)	17,12	17,12
Centro de fase(cm)	8,04	8,05
Ancho de haz a -3dB(°)	22,13	22,82
Spillover(dB)	-7,61	-7,42

INT. APERTURA a 8GHz

2º ppio	Phi = 0	Phi =90
Directividad(dB)	16,37	16,37
Centro de fase(cm)	5,24	5,37
Ancho de haz a -3dB(°)	25,0	26,0
Spillover(dB)	-7,26	-7,03

### 5.5.6 Piramidal

Parámetros de entrada:

Datos de entrada	
a	2.286
b	1.016
W	15
H	12
R1	25
R2	23,15

Diagrama de radiación y resultados:

Para un ejemplo concreto mostraremos su diagrama de radiación:

- Phi=0,
- Frecuencia= 10GHz
- 2º principio integrando en la apertura

Directividad:	20.6829
Ancho de banda -X dB:	15.2819
Centro de fase:	12.4776
Spillover:	-7.08

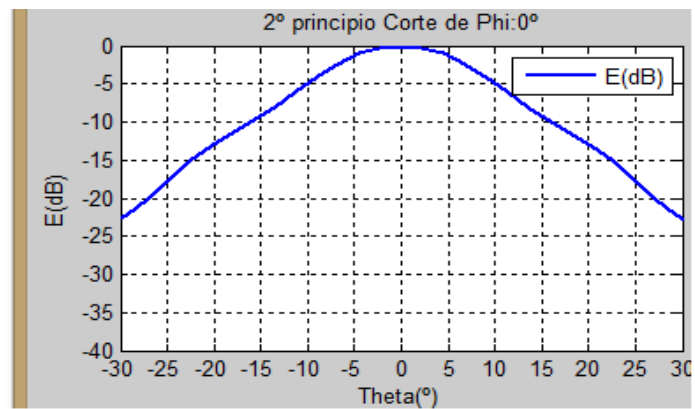


Fig 5.29 Diagrama y resultados corrugada piramidal

A continuación mostramos los resultados en una tabla para phi=0 y phi=90, la que está en la izquierda corresponde al análisis para una frecuencia de 10GHz. Por el contrario, en las tablas que están a la derecha corresponden a una frecuencia de 8GHz para ver su comportamiento.



INT. DIAGRAMA a 10GHz

1er ppio	Phi = 0	Phi =90
Directividad(dB)	20,68	20,68
Centro de fase(cm)	12,64	5,79
Ancho de haz a - 3dB(°)	15,49	17,94
Spillover(dB)	-3,98	-3,09

INT. DIAGRAMA a 8GHz

1er ppio	Phi = 0	Phi =90
Directividad(dB)	19,27	19,27
Centro de fase(cm)	8,38	3,78
Ancho de haz a - 3dB(°)	18,33	21,88
Spillover(dB)	-3,88	-2,82

Como hemos podido comprobar a mayor frecuencia, la bocina es más directiva.

INT. DIAGRAMA a 10GHz

2º ppio	Phi = 0	Phi =90
Directividad(dB)	17,66	17,66
Centro de fase(cm)	12,47	5,80
Ancho de haz a - 3dB(°)	15,28	17,94
Spillover(dB)	-7,08	-6,11

INT. DIAGRAMA a 8GHz

2º ppio	Phi = 0	Phi =90
Directividad(dB)	16,25	16,25
Centro de fase(cm)	8,23	3,8
Ancho de haz a - 3dB(°)	18,0	21,88
Spillover(dB)	-7,03	-5,84

Ahora vamos a ejecutar el análisis integrando en la apertura:

1er ppio	Phi = 0	Phi =90
Directividad(dB)	20,68	20,68
Centro de fase(cm)	12,64	5,79
Ancho de haz a - 3dB(°)	15,49	17,94
Spillover(dB)	-3,98	-3,09

1er ppio	Phi = 0	Phi =90
Directividad(dB)	19,28	19,27
Centro de fase(cm)	8,38	3,78
Ancho de haz a 3dB(°)	18,33	21,88
Spillover(dB)	-3,88	-2,82

Como hemos podido comprobar a mayor frecuencia, la bocina es más directiva.

INT. APERTURA a 10GHz

2º ppio	Phi = 0	Phi =90
Directividad(dB)	20,68	20,68
Centro de fase(cm)	12,47	5,80
Ancho de haz a - 3dB(°)	15,28	17,94
Spillover(dB)	-7,08	-6,11

INT. APERTURA a 8GHz

2º ppio	Phi = 0	Phi =90
Directividad(dB)	19,28	19,27
Centro de fase(cm)	8,23	3,8
Ancho de haz a 3dB(°)	18,0	21,88
Spillover(dB)	-7,03	-5,84

# CAPÍTULO VI: CONCLUSIONES Y LINEAS FUTURAS

---

## CAPÍTULO VI CONCLUSIONES Y LINEAS FUTURAS

En este proyecto se ha implementado una nueva versión del software SABOR para propósitos educativos. Se han analizado seis tipos comunes de antenas de bocinas: sectorial H, sectorial E, piramidal, cónica, piramidal corrugada y cónica corrugada. El programa fue creado con el entorno y lenguaje de programación orientada a objetos de MATLAB, utilizando los códigos provistos en el proyecto de fin de carrera de Francisco Egea Castejón titulado "SABOR 3.0 Software de análisis de bocinas y reflectores", del proyecto de fin de carrera de Jaime Calvo Tiemblo titulado "Estudio y simulación de técnicas de codificación en sistemas espacio-temporales con múltiples antenas en transmisión y recepción" y en libros como *Antenna Theory: Analysis and Design* de Constantine Balanis.

El programa consta de una interfaz gráfica que tiene como entradas las variables básicas para el diseño de las bocinas, y muestra como salidas los parámetros fundamentales de estas antenas como el diagrama de radiación, directividad, ancho de haz, centro de fase y spillover. Se han validado los resultados comparándolos con resultados de versiones anteriores al SABOR.

El documento incluye toda la información necesaria para dar soporte a la aplicación SABOR y pueda servir como base para futuras versiones, por lo que contiene, no sólo una nueva visión de la programación, más enfocada en una programación estructurada que es totalmente diferente a lo visto en clase, sino también fundamentos teóricos desarrollados en la anterior versión pero fundamentales para nuestro programa.

Una de las grandes ventajas que tiene nuestro software es su funcionamiento en cualquier equipo provisto del sistema operativo Windows 7 y Windows 8 ya que está soportado a partir de la plataforma MATLAB que se renueva frecuentemente. En versiones anteriores al SABOR, para su funcionamiento, es necesario tener un ordenador provisto de Windows XP, cuyo uso actual está en declive ya que Windows no da soporte técnico a su antiguo sistema operativo.

Como ya sabemos uno de los principales objetivos del proyecto ha sido solucionar errores que presentaba la anterior versión del programa (SABOR 3) y desarrollar mejoras a partir de una serie de pruebas exhaustivas. Algunos de los errores solucionados son:

- Algunos de los datos de salida en la versión SABOR 3 presentaban errores de cálculo. Pongamos un ejemplo concreto, la Directividad en la apertura para la bocina Sectorial E y bocina Circular presentan un error de cálculo que con la versión de Matlab han sido resueltos.
- La versión de SABOR en entorno MATLAB al ejecutar un análisis de cualquier bocina genera automáticamente un fichero.txt con los datos del diagrama de radiación. Esto permite mayor

flexibilidad y uso de la herramienta ya que con el fichero txt en mano, podemos exportarlo a Excel para su posterior tratamiento.

- Gracias a la plataforma de MATLAB podemos editar las gráficas a nuestro gusto. En versiones anteriores el tratamiento de imágenes era muy limitado.
- Una de las grandes mejoras y objetivos del proyecto es la mayor intuitividad y funcionalidad posible del programa. Por ello, se ha intentado reducir el número de ventanas para evitar complejidades que presentaban en versiones anteriores y se han integrado en una sola ventana.

El proyecto, por tanto, ha continuado una línea de trabajo y deja abierta la posibilidad de nuevas tareas como:

- Aumentar la rapidez de ejecución de los cálculos numéricos necesarios para el análisis de las antenas, y posibles mejoras en los resultados finales.
- Queda pendiente la inclusión del análisis de los reflectores en la herramienta SABOR.

## CAPÍTULO VII: BIBLIOGRAFÍA

---

## CAPÍTULO VII BIBLIOGRAFÍA

[1] Proyecto de fin de carrera titulado " Sabor v3.0 Software de análisis de bocinas y reflectores" (Año 2002) hecho por Francisco Egea Castejón del grupo de Radiación E.T.S.I. Telecomunicación Universidad Politécnica de Madrid.

[2] Stutzman, W. L., y Thiele, G. A. (1998). "Antenna Theory and Design (2da ed.). Hoboken, NJ: JohnWiley & Sons.

[3] Balanis, C. A. (2005). "Antenna Theory: Analysis and Design" (3ra ed.). Hoboken, NJ: John Wiley& Sons.

[4]Proyecto de fin de carrera titulado "Estudio y simulación de técnicas de codificación en sistemas espacio-temporales con múltiples antenas en transmisión y en recepción." Hecho por Jaime Calvo Tiemblo en Julio 2007.

[5]<http://www.mathworks.es/>

[6] help MATLAB R2012b.

[7] "SABOR: description of the methods applied for a fast analysis of horn and reflector antennas". IEEE Antennas and Propagation Magazine, Vol. 40, No., August 1998. Leandro de Haro.

[8] [www.youtube.es](http://www.youtube.es)

[9] Ángel Cardama Aznar, Lluís Jofre, Roca Juan Manuel, Rius Casals Jordi, Romeu Robert Sebastián, Blanch Boris UNIVERSITAT POLITÈCNICA DE CATALUNYA Miguel Ferrando Bataller UNIVERSITAT POLITÈCNICA DE VALÈNCIA. "Antenas" 2º edición año 2002".

[10] Apuntes de la asignatura Antenas y compatibilidad electromagnética de la EUITT.

[11] Apuntes de la asignatura Antenas del departamento SSR de la ETSIT.<http://www.gr.ssr.upm.es/docencia/grado/antenas/>

[12] [http://es.wikipedia.org/wiki/Programaci%C3%B3n\\_orientada\\_a\\_objetos](http://es.wikipedia.org/wiki/Programaci%C3%B3n_orientada_a_objetos)