



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA

– **TELECOM** ESCUELA  
TÉCNICA **VLC** SUPERIOR  
DE INGENIERÍA DE  
TELECOMUNICACIÓN

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Técnica Superior de Ingeniería de  
Telecomunicación

Herramienta de análisis de grandes agrupaciones de  
antenas a partir de agrupaciones más pequeñas

Trabajo Fin de Grado

Grado en Ingeniería de Tecnologías y Servicios de  
Telecomunicación

AUTOR/A: Serrano Callejas, Adrián

Tutor/a: Valero Nogueira, Alejandro

CURSO ACADÉMICO: 2021/2022



## Resumen

El siguiente trabajo pretende desarrollar una herramienta que facilite el análisis aproximado de grandes agrupaciones de antenas a partir del análisis de agrupaciones de menor tamaño, que permitan ser estudiadas de una manera más sencilla, que la agrupación objetivo.

El coste computacional asociado al análisis de *arrays* de grandes dimensiones (a partir de 16x16 elementos) es demasiado elevado y resulta demasiado complejo.

Para resolver este inconveniente, la solución expuesta en este proyecto, utiliza el análisis de agrupaciones más pequeñas, cuyas características de radiación (ya conocidas), van a permitir obtener una aproximación de las características de la agrupación buscada.

El proceso, se basa en apoyarse en el análisis de *arrays* de (generalmente), 4x4 u 8x8 elementos, para extrapolar los parámetros de interés. El trabajo realizará las rutinas de análisis, apoyándose en el software *Feko*, que nos permitirá comparar los resultados obtenidos y se completará con una herramienta *Matlab* para facilitar el manejo de todas las variables implicadas.

Finalmente, se compararán los resultados de ambas soluciones obtenidas tanto en *Feko*, como en *Matlab*, para tratar de concluir si la herramienta realiza una aproximación adecuada.

## Resum

El següent treball pretén desenvolupar una eina que facilite l'anàlisi aproximat de grans agrupacions d'antenes a partir de l'anàlisi d'agrupacions de menor grandària, que permeten ser estudiades d'una manera més senzilla, que l'agrupació objectiu.

El cost computacional associat a l'anàlisi d'*arrays* de grans dimensions (a partir de 16x16 elements) és massa elevat i resulta massa complex.

Per a resoldre aquest inconvenient, la solució exposada en aquest projecte, utilitza l'anàlisi d'agrupacions més xicotetes, les característiques de radiació de les quals (ja conegudes), permetran obtindre una aproximació de les característiques de l'agrupació buscada.

El procés, es basa a secundar-se en l'anàlisi d'*arrays* de (generalment), 4x4 o 8x8 elements, per a extrapolar els paràmetres d'interés. El treball realitzarà les rutines d'anàlisi, secundant-se en el programa *Feko*, que ens permetrà comparar els resultats obtinguts i es completarà amb una eina *Matlab* per a facilitar el maneig de totes les variables implicades.

Finalment, es compararan els resultats de totes dues solucions obtingudes tant en *Feko*, com en *Matlab*, per a tractar de concloure si l'eina realitza una aproximació adequada.

## Abstract

The following work aims to develop a tool that facilitates the approximate analysis of large groups of antennas from the analysis of smaller groups, which allow them to be studied in a simpler way than the target group.

The computational cost associated with the analysis of large *arrays* (from 16x16 elements) is

---

too high and too complex.

To solve this inconvenience, the solution presented in this project uses the analysis of smaller groups, whose radiation characteristics (already known), will allow obtaining an approximation of the characteristics of the group sought.

The process is based on relying on the analysis of *arrays* of (generally), 4x4 or 8x8 elements, to extrapolate the parameters of interest. The work will carry out the analysis routines, supported by the *Feko* software, which will allow us to compare the results obtained and will be completed with a *Matlab* tool to facilitate the handling of all the variables involved.

Finally, the results of both solutions obtained both in *Feko* and in *Matlab* will be compared, to try to conclude if the tool performs an adequate approximation.

---

## Agradecimientos

A mis padres, por apoyarme en los momentos más complicados de la carrera, y por haberme enseñado los valores con los que viviré toda mi vida. Por enseñarme que, una persona que se esfuerza puede con absolutamente todo, aunque tenga que dedicarle más tiempo que el resto.

Dar un agradecimiento especial a mi tutor Alejandro, no solo por la oportunidad de haber desarrollado este trabajo con él, sobre un tema tan apasionante para mí, sino también por demostrarme, que una de las asignaturas más complicadas de la carrera, podría llegar a ser, una de las más interesantes de mi vida como alumno.

Por otra parte, al resto de profesores que me han ayudado a aprender, a esforzarme y sobretodo, a creer en mí mismo, cuando la poca confianza afectaba de manera negativa a mis estudios.

Por último, a todos aquellos compañeros que me han ayudado y me han apoyado, para llegar hasta aquí.

# Índice general

## I Memoria

<b>1. Introducción</b>	<b>1</b>
1.1. Contexto del trabajo . . . . .	1
1.2. Objetivos . . . . .	2
1.3. Estructura . . . . .	2
<b>2. Principios teóricos</b>	<b>5</b>
2.1. <i>Array</i> sin acoplos . . . . .	6
2.2. <i>array</i> con acoplos . . . . .	7
2.3. Impedancia activa y corriente . . . . .	9
2.4. Factor de <i>array</i> . . . . .	11
2.5. Proceso para el análisis de agrupaciones a partir de otras formadas con menos elementos: <i>Array lineal</i> . . . . .	12
2.6. Proceso para el análisis de agrupaciones a partir de otras formadas con menos elementos: <i>Array plano</i> . . . . .	14
<b>3. Creación del entorno de simulación</b>	<b>17</b>
3.1. Appdesigner . . . . .	17
3.2. Diagrama de bloques . . . . .	18
3.3. Bloque 1: Matriz de admitancias . . . . .	19
3.4. Bloque 2: Matriz mapeada . . . . .	22
3.5. Bloque 3: Corrientes . . . . .	23
3.6. Bloque 4: Diagrama de radiación . . . . .	24
3.7. Bloque 5: Campo total . . . . .	26
3.8. Bloque 6: Visualización . . . . .	26
3.9. Bloque 7: Remuestreo . . . . .	29
<b>4. Resultados</b>	<b>31</b>
4.1. Análisis de resultados: Agrupación 8x8 . . . . .	31
4.2. Análisis de resultados: Agrupación 16x16 . . . . .	35
4.3. Análisis de resultados: Antena tipo parche . . . . .	38
<b>5. Conclusiones</b>	<b>45</b>
5.1. Aplicaciones . . . . .	46
5.2. Mejoras futuras . . . . .	46
<b>Bibliografía</b>	<b>49</b>

**Bibliografía**

**49**

# Índice de figuras

2.1. Representación de las interacciones que se producen entre dos antenas de cierta proximidad (Imagen extraída de [3]) . . . . .	5
2.2. Representación de tres agrupaciones distintas con sus fuentes de tensión . . . . .	5
2.3. Circuito serie sin acoplos . . . . .	6
2.4. Cuadripolo representando la matriz de admitancias que surge debido a las interacciones con elementos próximos . . . . .	7
2.5. Circuito con la influencia de los acoplos . . . . .	8
2.6. Factor de <i>array</i> de una agrupación de 10 elementos (Imagen extraída de [3]) . . . . .	11
2.7. Diagrama representando como pasar de una agrupación de 4 elementos a una de 8 . . . . .	12
2.8. Aspecto de la matriz de admitancias de la agrupación 4x4 (Imagen tomada del artículo [1]) . . . . .	13
2.9. Aspecto de la matriz de admitancias objetivo de 8x8 elementos (Imagen tomada a partir del artículo [1] ) . . . . .	13
2.10. Forma de representar el <i>array</i> plano . . . . .	14
2.11. Forma de representar el <i>array</i> plano con la matriz de una de sus casillas . . . . .	15
3.1. Diagrama de bloques del proyecto . . . . .	18
3.2. Primero de los bloques que conforman el programa . . . . .	20
3.3. Gráfica que muestra el aspecto de la matriz de admitancias . . . . .	21
3.4. Gráficas que muestran el módulo y la fase del archivo importado . . . . .	21
3.5. Matriz mapeada . . . . .	22
3.6. Zoom en uno de los bloques de la matriz mapeada . . . . .	23
3.7. Dipolo . . . . .	24
3.8. Segunda pestaña del programa desarrollado con <i>Appdesigner</i> . . . . .	25
3.9. Ventana emergente tras seleccionar el botón de importar diagrama de radiación . . . . .	25
3.10. Diagrama de radiación del dipolo visto desde el programa desarrollado . . . . .	25
3.11. Datos necesarios para realizar el proceso . . . . .	27
3.12. Aspecto de las diferentes distribuciones de corrientes . . . . .	27
3.13. Tercera pestaña de nuestro programa completa . . . . .	28
3.14. Visualización final del diagrama de radiación objetivo . . . . .	28
3.15. Datos solicitados por el programa desarrollado para llevar a cabo el remuestreo (si fuera necesario) . . . . .	29
4.1. Primera pestaña de nuestro programa . . . . .	32
4.2. Tercera pestaña de nuestro programa correspondiente a la agrupación extrapolada . . . . .	33
4.3. Diagrama de radiación de la agrupación objetivo de 8x8 elementos . . . . .	33
4.4. Resultados obtenidos para un corte de 0° . . . . .	34
4.5. Resultados obtenidos para un corte de 90° . . . . .	35



4.6. Tercera pestaña de nuestro programa correspondiente al <i>array</i> extrapolado . . . .	36
4.7. Diagrama de radiación de la agrupación objetivo de 16x16 elementos . . . . .	37
4.8. Resultado para un corte de 0° . . . . .	37
4.9. Resultado para un corte de 90° . . . . .	38
4.10. Aspecto de una antena de tipo parche generada desde <i>Feko</i> . . . . .	39
4.11. Diagrama de radiación de la antena de tipo parche . . . . .	40
4.12. Primera pestaña del programa para una antena tipo Parche . . . . .	41
4.13. Diagrama de radiación de la antena de tipo parche en la segunda pestaña del programa	41
4.14. <i>Array</i> extrapolado 8x8 utilizando una antena tipo parche . . . . .	42
4.15. Diagrama de radiación final de la agrupación objetivo . . . . .	43
4.16. Comparación de resultados antena parche 0° . . . . .	43
4.17. Comparación de resultados antena parche 90° . . . . .	44



**Parte I**

**Memoria**



# Capítulo 1

## Introducción

### 1.1. Contexto del trabajo

Uno de los elementos más importantes dentro de las telecomunicaciones, por sus aplicaciones en campos tan diversos como la aeronáutica hasta las comunicaciones móviles e inalámbricas, son las agrupaciones de antenas.

Analizar las características de una antena aislada a partir de las diferentes *suites* de simulación que podemos encontrar disponibles para estudiantes, es una tarea sencilla, ya que algunos programas, como el entorno de análisis y desarrollo de antenas *Feko*, nos permiten obtener información muy diversa sobre un *array*: Diagrama de radiación, directividad, cortes en diferentes ejes, corrientes, matrices de impedancias y admitancias... Además, nos aporta toda esta información de una manera muy intuitiva y sencilla para el alumno.

El problema surge cuando hemos de analizar una agrupación de antenas de grandes dimensiones y el coste computacional se dispara.

El acoplo mutuo entre los elementos de un *array* es una parte fundamental a tener en cuenta. Si no se tuviera en cuenta este efecto, obtener las características principales de la agrupación sería algo, más que sencillo, pero el resultado obtenido no se correspondería con la realidad. Los inconvenientes que generan los acoplos afectan tanto a la impedancia de entrada como al diagrama de radiación del *array*.

Según artículos que tratan de buscar soluciones a este inconveniente [1], [2], dos métodos de análisis han sido utilizados durante estos años para poder solventar el problema:

- Análisis basado en la aproximación mediante *arrays* infinitos y la periodicidad de la estructura: Este modelo es el más usado pero no realiza una buena aproximación de los efectos de los acoplos mutuos en los bordes, por lo que no resulta la mejor opción para obtener el diagrama de radiación de la agrupación objetivo
- Método basado en el tamaño real o finito del *array*: Método que basa sus limitaciones en el coste computacional tan elevado que posee, que lo convierte en un método inviable para *arrays* muy grandes.

Ante estas limitaciones, basadas en el coste computacional para compilar y para poder observar

las características de la agrupación, (resultando en una gran cantidad de tiempo invertido y bajo rendimiento del entorno empleado) diferentes soluciones se han planteado para cubrir esa necesidad de un método, que permita calcular las características de una agrupación con un gran número de elementos, entre ellas, métodos que combinan los dos, comentados anteriormente, aunque no se ha conseguido el resultado deseado.

La solución planteada en este proyecto, será aquella en la que a partir de un *array* de menor tamaño, consigamos mapear la matriz de admitancias característica de este y así obtener las características de la agrupación objetivo de forma aproximada.

## 1.2. Objetivos

Para conseguir el propósito comentado anteriormente, el primero de los objetivos que tratará de abordar esta memoria, es detallar de forma minuciosa los procesos y cálculos teóricos necesarios, tratando de facilitar el orden y la justificación del proyecto al lector.

Además, a lo largo del desarrollo del programa que nos ayudará a obtener la respuesta final, se van a realizar diferentes casos, siguiendo un proceso que involucra los dos programas comentados anteriormente.

En primer lugar, deberemos crear un proyecto *Feko* que nos permita conocer el diagrama de radiación y las características de la antena específica aislada.

Para comprobar el correcto funcionamiento del programa que se desarrollará posteriormente, las antenas aisladas empleadas serán un dipolo (ya que es el principal tipo de antena que suele utilizarse como antena aislada debido al diagrama de radiación tan característico y sencillo que posee), y una antena de tipo parche (utilizada sobretodo para tecnología *microstrip* y para numerosas aplicaciones actuales), cuya dificultad aumenta considerablemente, en comparación al dipolo (sobretodo a nivel de coste computacional, como se verá posteriormente). Por tanto, el siguiente objetivo será demostrar la diferencia en el proceso, en la respuesta, pero sobre todo, en el coste computacional que el uso de cada antena supone.

A partir del programa desarrollado en *Matlab* (cuyo proceso se mostrará en los diferentes puntos de esta memoria), el cual utiliza como datos el archivo correspondiente al diagrama de radiación de la antena aislada y la matriz de admitancias del *array* de menor tamaño, conseguiremos obtener una aproximación del diagrama de radiación buscado, siendo este el principal objetivo de nuestro trabajo.

Finalmente, se realizará una comparación del diagrama de radiación objetivo y aquel que ha sido generado en un principio con el entorno de desarrollo *Feko*, para comprobar, si nuestro programa funciona de forma adecuada.

## 1.3. Estructura

En esta memoria, se va a dividir el proyecto en diferentes fases, según la línea temporal y de trabajo que se ha seguido a la hora de realizar este proceso.

Se hará un repaso de las ideas teóricas principales, haciendo hincapié en como conseguir dividir

la agrupación en agrupaciones más sencillas y se hará una descripción minuciosa del proceso que se ha seguido para desarrollar el programa en *Matlab*, utilizando una herramienta de diseño, incluida en el propio entorno y de la cual se hablará posteriormente.

De igual forma, se describirá como se han creado todos los ejemplos desde el entorno de desarrollo de *Feko*, para finalmente dar una conclusión.

Los diferentes capítulos en los que se dividirá el proyecto, son los siguientes:

- **Capítulo 1. Introducción:** En primer lugar, se muestran las directrices que seguirá el proyecto, detallando tanto el contexto del trabajo expuesto, así como sus objetivos o principales motivaciones y detallando la estructura que seguirá el mismo.
- **Capítulo 2. Principios teóricos:** En este capítulo, se tratará de dar una explicación teórica del proyecto en cuestión, donde se explicará el método necesario y los conceptos fundamentales, que nos permitirán estudiar el comportamiento de la agrupación objetivo, a partir de agrupaciones más sencillas.
- **Capítulo 3. Creación del entorno de simulación:** Este capítulo basa su contenido en desarrollar el programa que nos permitirá obtener los resultados finales. Para ello, se mostrará el funcionamiento del programa, tratando de completar la explicación teórica del capítulo anterior y plasmarla en código *Matlab*, detallando como se han convertido las ideas teóricas, en cada uno de los bloques que formarán la versión final.
- **Capítulo 4. Resultados:** Una vez desarrollado el programa, será momento de tratar el contenido de este capítulo, mostrando los resultados obtenidos a partir de las diferentes funciones que forman el programa, para realizar un análisis y demostrar si el funcionamiento del programa es el correcto.
- **Capítulo 5. Conclusiones:** Finalmente, en el último capítulo se realizará el análisis y conclusión de los resultados obtenidos, se detallarán las principales aplicaciones que podría tener este programa, o esta idea, hablando generalmente, y finalmente, las líneas de trabajo futuras que podrían aplicarse al proyecto.

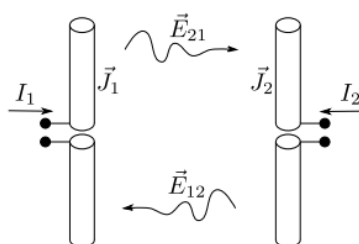




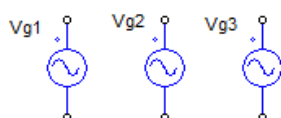
## Capítulo 2

# Principios teóricos

A la hora de estudiar un *array* de antenas, hemos de tener en cuenta el acoplamiento que sufren los diferentes elementos del *array*, debido a la influencia del resto de elementos de este. La interacción entre los diferentes elementos repercute en su impedancia de entrada y por tanto en la corriente de alimentación, siendo esta distinta a la esperada en un principio [3].



**Figura 2.1:** Representación de las interacciones que se producen entre dos antenas de cierta proximidad (Imagen extraída de [3])



**Figura 2.2:** Representación de tres agrupaciones distintas con sus fuentes de tensión

Si observamos la figura 2.1, que representa dos dipolos de cierta proximidad, podemos entender a que nos referimos. El comportamiento ideal de la antena (y sus características) cuando se encuentra aislada, van a modificarse al encontrar una antena en las proximidades de su posición.

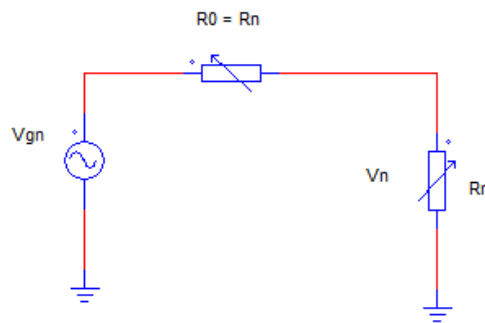
En la imagen 2.2, se trata de mostrar esta influencia ya que, el elemento de la segunda posición y de la tercera posición van a ejercer una influencia en el de la primera, del mismo modo que el primer elemento y el tercero van a ejercer una influencia sobre el segundo y así sucesivamente.

El propósito del proyecto, es tener en cuenta los acoplos que van a influenciar el comportamiento del resto de elementos, y de esta forma, extraer todos los cálculos y procesos necesarios para llevar a cabo el objetivo principal del trabajo: conseguir obtener las características de una agrupación de un número determinado de elementos a partir de otra de menor tamaño.

Pero para llevar a cabo el proceso, teniendo en cuenta el acoplo mutuo, es necesario observar como se comporta la agrupación sin tener en cuenta este fenómeno.

## 2.1. Array sin acoplos

Si no se tiene en cuenta el acoplamiento entre elementos, el esquema de nuestra agrupación puede entenderse como un circuito serie donde intervienen una tensión aplicada, que será nuestra tensión dato ( $V_{gn}$ ) y una tensión efectiva ( $V_n$ ), así como una resistencia interna, característica de cualquier circuito que incluya una fuente de tensión ( $R_0$ ) y una resistencia efectiva ( $R_N$ ). El circuito, quedaría como muestra la figura 2.3:



**Figura 2.3: Circuito serie sin acoplos**

Donde podemos observar, si realizamos un simple análisis del circuito, mediante la teoría de mallas y teniendo en cuenta la suposición, de que la resistencia efectiva es idéntica a la resistencia interna (ya que en este caso no existen acoplos y por tanto no existirá influencia de ningún otro elemento en el circuito), que la forma de poder obtener la tensión efectiva, básicamente se basa en una simple ecuación:

$$V_n = \frac{V_{gn}}{2} \quad (2.1)$$

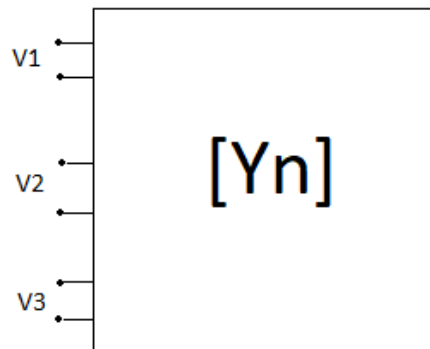
Pero realmente, la forma de entender el funcionamiento interno de la agrupación cambia si tenemos en cuenta el acoplo que se produce entre los diferentes elementos del *array*, como se verá en los siguientes puntos, donde en este caso, si que será necesario considerar un elemento que produce una influencia en el circuito característico de nuestra agrupación.

## 2.2. array con acoplos

A la hora de explicar como se modifica el proceso, teniendo en cuenta el fenómeno de los acoplos, debemos apoyarnos en la teoría de cuadripolos, pues nos será de vital importancia a la hora de realizar los cálculos necesarios para entender el proceso.

Los dos dipolos que se representan en la figura 2.1, se afectan mutuamente de modo que las reacciones mutuas entre campos y densidades de corriente, pueden representarse mediante los parámetros  $[Z]$  de los cuadripolos. De este modo, tanto las tensiones como las corrientes de los puertos de los dipolos se pueden relacionar entre sí, empleando la teoría y las expresiones relacionadas con los parámetros  $[Z]$  de dichos cuadripolos.

Por tanto, si se tiene en cuenta el fenómeno del acoplo entre elementos y la teoría de cuadripolos, tendremos la representación correspondiente a la figura 2.4 (para el cuadripolo que representa la matriz de admitancias) y a la figura 2.5 (para el esquema modificado).



**Figura 2.4: Cuadripolo representando la matriz de admitancias que surge debido a las interacciones con elementos próximos**

Donde en este caso, el cuadripolo representado con la expresión  $[Y]$ , hace referencia a la admitancia que surgirá de la influencia de los elementos próximos a un dipolo.

Por tanto, y teniendo en cuenta que la forma de relacionar las corrientes de los diferentes elementos del *array*, mediante la teoría de cuadripolos se obtiene mediante la siguiente expresión:

$$I_n = Y \cdot V_n \quad (2.2)$$

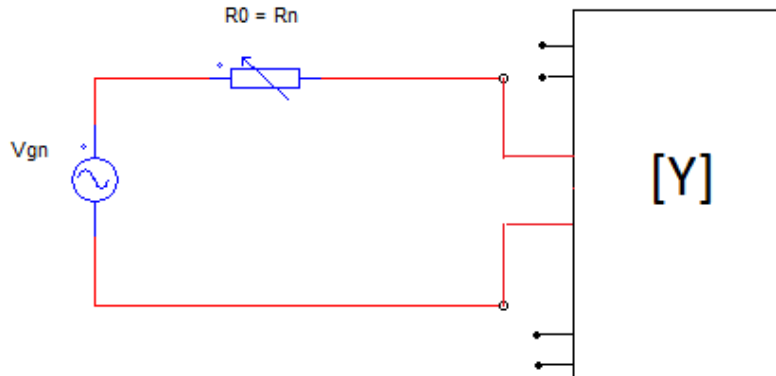
Es sencillo entender que, las ecuaciones correspondientes a las diferentes corrientes, debidas a la interacción del resto de elementos del *array* van a tener el siguiente aspecto:

$$I_1 = Y_{11} \cdot V_1 + Y_{12} \cdot V_2 + Y_{13} \cdot V_3 \quad (2.3)$$

$$I_2 = Y_{21} \cdot V_1 + Y_{22} \cdot V_2 + Y_{23} \cdot V_3 \quad (2.4)$$

$$I_3 = Y_{31} \cdot V_1 + Y_{32} \cdot V_2 + Y_{33} \cdot V_3 \quad (2.5)$$

Donde hemos considerado que se trabaja con una agrupación de 4x4 elementos, de ahí los sub-índices indicados en las expresiones correspondientes a las admitancias (considerando una matriz,



**Figura 2.5: Circuito con la influencia de los acoplos**

tendremos la admitancia debida al elemento de la primera fila y la primera columna, el elemento de la primera fila y la segunda columna, y así sucesivamente, hasta completar una matriz de 16 elementos en total).

A partir de las ecuaciones obtenidas anteriormente, y teniendo en cuenta que la expresión de la admitancia de entrada (aquella importante para continuar desarrollando los cálculos), se obtiene según la teoría, del siguiente cociente:

$$Y_{in1} = \frac{I_1}{V_1} \quad (2.6)$$

Igualando las expresiones 2.3, 2.4 y 2.5 con la expresión anterior, obtendremos finalmente la siguiente expresión (caracterizada únicamente para obtener la admitancia del elemento de la primera fila y la primera columna de la matriz):

$$Y_{in1} = \frac{I_1}{V_1} = Y_{11} + \left( Y_{12} \cdot \frac{V_2}{V_1 + Y_{13} \cdot \frac{V_3}{V_1}} \right) \quad (2.7)$$

Por otra parte, si se analiza el circuito característico, aplicando de nuevo la teoría de mallas, tendremos el siguiente desarrollo:

$$V_{gn} = R_n \cdot I_n + V_n \quad (2.8)$$

Donde, despejando:

$$I_n = \frac{V_{gn} - V_n}{R_n} \quad (2.9)$$

Este, nos permitirá calcular las corrientes efectivas, a partir de la tensión dato, la tensión efectiva y finalmente, a partir de la resistencia efectiva.

Una vez concluido este análisis, podemos expresarlo de forma matricial, para ser más comprensible y para poder obtener la conclusión deseada. Si expresamos en forma matricial:

$$[I_n] = \begin{bmatrix} \ddots & 0 & 0 \\ 0 & 1/R_n & 0 \\ 0 & 0 & \ddots \end{bmatrix} \cdot ([V_{gn}] - [V_n]) \quad (2.10)$$

Como puede observarse, esta expresión en forma matricial, coincide con los cálculos desarrollados en la expresión 2.9, donde tendremos que la diferencia entre la tensión real o eficaz y la tensión de la fuente, multiplicada por una matriz diagonal con el cociente  $1/R_n$ , nos dará el valor de la corriente.

Teniendo en cuenta la relación entre la corriente, la admitancia y la tensión real, expresada en la ecuación 2.2, podremos obtener de forma matricial, la siguiente ecuación:

$$[I_n] = [Y_n] \cdot [V_n] \quad (2.11)$$

$$[Y_n] \cdot [V_n] = \begin{bmatrix} \ddots & 0 & 0 \\ 0 & 1/R_n & 0 \\ 0 & 0 & \ddots \end{bmatrix} \cdot ([V_{gn}] - [V_n]) \quad (2.12)$$

Si comenzamos a despejar la ecuación anterior, podremos obtener los resultados necesarios, teniendo en cuenta que se conoce como dato a la tensión aplicada. De ese modo, podremos conocer la relación que existe entre la tensión aplicada y la tensión real o efectiva:

$$[V_{gn}] = \left( \begin{bmatrix} \ddots & 0 & 0 \\ 0 & R_n & 0 \\ 0 & 0 & \ddots \end{bmatrix} \cdot [Y_n] + \begin{bmatrix} \ddots & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \ddots \end{bmatrix} \right) \cdot [V_n] \quad (2.13)$$

Siendo la matriz I, la matriz identidad que, como sabemos es una matriz de ceros salvo en la diagonal principal, que se ocupa con 1.

Con todo lo anterior, podemos concluir, que la matriz de admitancias va a ser la herramienta fundamental para poder analizar agrupaciones de un tamaño mayor, a partir de agrupaciones más sencillas (gracias a la teoría de cuádrupolos), ya que, únicamente mapeando esta matriz, será posible obtener matrices de tamaños superiores, como se explicará posteriormente.

## 2.3. Impedancia activa y corriente

De la misma forma que hemos podido obtener los valores de las corrientes y admitancias de nuestro circuito, también es posible calcular el valor de la impedancia activa y expresar los cálculos llevados a cabo, de una forma matricial, para su posterior desarrollo con *Matlab*, así como también pueden calcularse las corrientes efectivas, que serán de vital importancia para el proyecto.

De nuevo, debemos recaer en la teoría de cuadripolos (necesaria, si vamos a realizar cálculos con parámetros  $[Y]$  o  $[Z]$ ), que nos va a permitir obtener las herramientas necesarias para poder calcular cada una de las incógnitas. De nuevo, reutilizaremos expresiones comentadas en secciones anteriores, con el fin de que la comprensión sea lo más satisfactoria posible.

Comenzaremos este proceso, obteniendo en primer lugar la impedancia activa, y para ello, debemos tener en cuenta, de nuevo, el circuito característico de 2.5. Realizando un análisis de este, teniendo en cuenta que la impedancia puede calcularse como el cociente de la tensión y la corriente, tendremos el siguiente desarrollo:

$$Z_{n(activa)} = \frac{V_n}{I_n} = R_n \cdot \left( \frac{V_n}{V_{gn} - V_n} \right) \quad (2.14)$$

Como ya se ha hecho anteriormente, la idea fundamental de estos cálculos teóricos es llevar las ecuaciones a un plano matricial, donde podamos entender de una forma más sencilla su funcionamiento, y que facilite la posterior implementación en *Matlab*.

Como sabemos por lo deducido anteriormente, y como puede comprobarse en la ecuación 2.13, que representa la relación existente entre la tensión aplicada y la efectiva, podemos despejar esta tensión efectiva, únicamente pasando el resto de términos de la ecuación al otro lado de la igualdad (de ahí la importancia de desarrollar la expresión, mostrando la relación entre ambas tensiones).

$$[V_n] = \left( \left( \begin{bmatrix} \ddots & 0 & 0 \\ 0 & R_n & 0 \\ 0 & 0 & \ddots \end{bmatrix} \cdot [Y_n] + \begin{bmatrix} \ddots & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \ddots \end{bmatrix} \right)^{-1} \cdot [V_{gn}] = [T_n] \cdot [V_{gn}] \quad (2.15)$$

Sí además, tenemos en cuenta que todo el primer término del producto (aquel que se encuentra entre paréntesis y elevado a un exponente -1), correspondería a la corriente efectiva (como puede observarse en la ecuación 2.12, después de haber aplicado la ecuación 2.11) podemos ahora escribir la impedancia activa de forma matricial, teniendo en cuenta lo desarrollado en todo este punto, como sigue:

$$[Z_a] = \begin{bmatrix} \ddots & 0 & 0 \\ 0 & R_n & 0 \\ 0 & 0 & \ddots \end{bmatrix} \cdot \frac{[T_n] \cdot [V_{gn}]}{\left( \begin{bmatrix} \ddots & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \ddots \end{bmatrix} - [T_n] \right) \cdot [V_{gn}]} \quad (2.16)$$

Como se puede observar, se aplica el resultado de la ecuación 2.14, solo que aplicando todo lo explicado anteriormente.

Para acabar el apartado que nos ocupa, debemos comentar como obtener las corrientes de nuestra agrupación.

Estas, nos serán de gran importancia a la hora de calcular, posteriormente, nuestro factor de *array*, que constata uno de los elementos más importantes de las agrupaciones de antenas.

Para ello, únicamente debemos saber que la corriente, puede calcularse, teniendo en cuenta la expresión 2.2, con una única modificación, y es que, en nuestro caso, para obtener la corriente real o efectiva, debemos recurrir a la impedancia activa y a la tensión activa (ambas calculadas anteriormente).

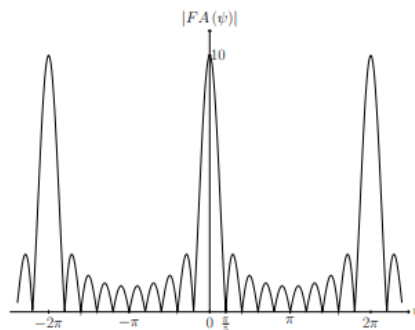
La solución es sencilla, pues como sabemos, la relación entre la impedancia y la admitancia, se basa en que una, es la inversa de la otra, con lo que, podemos calcular la corriente deseada, aplicando un cociente de tensión, entre impedancia:

$$I_n = \frac{V_a}{Z_a} \quad (2.17)$$

Todos los cálculos desarrollados hasta ahora, serán de gran importancia a la hora de desarrollar el programa final, como se verá posteriormente, en el diagrama de bloques que estructurará el capítulo 3. De momento, debemos centrarnos en este último cálculo, pues, como se ha comentado anteriormente, es importante para poder caracterizar al elemento más importante de una agrupación de antenas, del tipo que sea: El factor de *array*.

## 2.4. Factor de *array*

Uno de los elementos más importantes (por no decir el más importante), de las agrupaciones de antenas es el denominado factor de *array*.



**Figura 2.6:** Factor de *array* de una agrupación de 10 elementos (Imagen extraída de [3])

El factor de *array*, es independiente de la antena que lo forma y aporta la información de la interferencia como si los radiadores fueran isótropos. Por tanto, se puede considerar que el factor de *array* actúa como el diagrama de radiación de un *array* de elementos isótropos. La expresión fundamental de este factor, se representa a continuación, siendo el sumatorio de las amplitudes de corriente y siendo el parámetro exponencial, aquel que representa el desfase entre los campos radiados por dos antenas contiguas debido a la diferencia de caminos recorridos y a la diferencia de fase en la alimentación [3].

$$F(\theta, \phi) = \sum_{n=0}^N a_n \cdot e^{jn\psi_n} \quad (2.18)$$

En nuestro caso, la importancia de este elemento característico de las agrupaciones de antenas, se basa en que, gracias al producto de este factor de *array* junto con el diagrama de radiación de la antena aislada, nos va a permitir obtener la forma del campo total, necesario para los posteriores cálculos de nuestro programa a desarrollar.

$$\vec{E}_{total} = F(\theta, \phi) \cdot \vec{E}_{elemento} \quad (2.19)$$

Donde  $\vec{E}_{elemento}$ , hace referencia al campo eléctrico de la antena aislada.

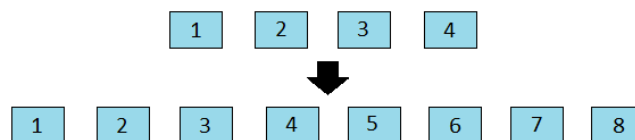
La forma de poder obtener este factor, es gracias a las corrientes comentadas en el apartado anterior, pues a partir de la extracción de estas, podemos ser capaces de construir nuestro factor de *array*, para posteriormente calcular el campo total, necesario para obtener la respuesta final de nuestro programa.

Como ya se ha comentado, las corrientes, podrán calcularse a partir de la expresión de la ecuación 2.17, a partir de la tensión y la impedancia efectivas.

## 2.5. Proceso para el análisis de agrupaciones a partir de otras formadas con menos elementos: *Array lineal*

Llegados a esta sección del capítulo correspondiente al desarrollo teórico, es momento de expresar como llevar a cabo el proceso de mapeo por el cual, gracias a la teoría que se comentará a continuación, seremos capaces de obtener esa agrupación de mayor tamaño a partir de una menor. Para ello, debemos recurrir a la matriz de admitancias comentada en apartados anteriores.

Si tomamos como ejemplo lo expuesto en el artículo [1], y en el artículo [2] podemos entender el proceso por el cual podemos realizar este procedimiento. Si consideramos una agrupación de cuatro elementos, que queremos mapear o extrapolar, para obtener las características de una de ocho elementos, podríamos expresarlo como una serie de “casillas”, como se indica en la figura 2.7:



**Figura 2.7:** Diagrama representando como pasar de una agrupación de 4 elementos a una de 8

Tomando como ejemplo un *array* de 8x8 elementos como *array* de mayor tamaño y tratando de analizarlo a partir de un *array* 4x4, tendremos que el aspecto de la matriz de admitancias de la agrupación menor, será aquel que se representa en la imagen 2.8:



1,1	1,2	1,3	1,4
2,1	2,2	2,3	2,4
3,1	3,2	3,3	3,4
4,1	4,2	4,3	4,4

**Figura 2.8: Aspecto de la matriz de admitancias de la agrupación 4x4 (Imagen tomada del artículo [1])**

Únicamente mapeando esta matriz, transformándola en una matriz de 8x8 elementos, tendremos el resultado de la figura 2.9:

1,1	1,2	1,3	1,4	0	0	0	0
2,1	2,2	2,3	2,4	1,4	0	0	0
3,1	3,2	3,3	3,4	2,4	1,4	0	0
4,1	4,2	4,3	4,4	3,4	2,4	1,4	0
0	0	2,4	3,4	3,3	3,4	2,4	1,4
0	0	1,4	2,4	3,4	3,3	3,4	2,4
0	0	0	1,4	2,4	2,3	2,2	3,4
0	0	0	0	1,4	1,3	1,2	1,1

**Figura 2.9: Aspecto de la matriz de admitancias objetivo de 8x8 elementos (Imagen tomada a partir del artículo [1] )**

Teniendo en cuenta el espaciado entre elementos (generalmente representado con la letra minúscula *d*), podemos observar como los elementos de la matriz que superan el triple de este espaciado, quedan representados por un 0, dejando aquellos elementos de la matriz 4x4 que no superan este espaciado, idénticos.

Con este análisis, queda demostrado como a partir de un *array* de 4x4 elementos y a partir de un mapeo de los elementos de la matriz, es posible obtener la matriz de admitancias de agrupaciones de 8x8, 16x16, 32x32, o incluso más elementos.

Con todo lo desarrollado anteriormente, el objetivo fundamental va a ser tratar de plasmar estas ideas teóricas en un código *Matlab* que nos permita obtener las características de la agrupación objetivo.

Para ello, se crearán una serie de funciones que nos permitan realizar todos los cálculos anteriores a partir de los datos necesarios, en este caso, tanto la matriz de admitancias dato, como el diagrama de radiación de la antena aislada.

Posteriormente, trataremos de sintetizar este código mediante la herramienta *Appdesigner*, para poder visualizar todos los resultados a medida que se compilan para, finalmente, exportarlo de nuevo, y comprobar los resultados, mediante el entorno de análisis de *Feko*.

## 2.6. Proceso para el análisis de agrupaciones a partir de otras formadas con menos elementos: *Array* plano

Todo el proceso comentado anteriormente para una agrupación lineal puede extenderse al caso de un *array* plano, ya que deseamos que el programa que se desarrollará a continuación, ofrezca la posibilidad de calcular las características de este tipo de agrupaciones (además de las lineales). Para ello, debemos tener en cuenta que el proceso de mapeo, es similar al anterior, solo que en este caso, algo más complejo.

Para empezar, debemos tener en cuenta que el fenómeno que ocurre elemento a elemento en la agrupación lineal, en un *array* plano, ocurrirá fila a fila, ya que cada fila es un *array*.

Las matrices de las agrupaciones lineales, en este tipo de agrupaciones, son casillas, existiendo además una interacción mutua entre filas.

Para entender algo mejor lo comentado, se representa el siguiente ejemplo, en el cual se entiende, como en un *array* plano, las casillas o elementos, son directamente, matrices completas (Observar figuras 2.10 y 2.11). Tomando, por tanto, como ejemplo, un *array* plano de 4x4 elementos, la forma de representar la matriz mapeada, será la expuesta en las figuras comentadas.

1	4		
		F1,F1	F1,F2
		F1,F3	F1,F4
4		F2,F1	F2,F2
		F2,F3	F2,F4
		F3,F1	F3,F2
		F3,F3	F3,F4
		F4,F1	F4,F2
		F4,F3	F4,F4

**Figura 2.10: Forma de representar el *array* plano**

Como podemos observar, cada una de las casillas de la matriz, es una matriz de menor tamaño, con lo que cada una de las casillas, está formada por un número de matrices más pequeñas.

Los números representados (los dos cuatros y el uno), representan las filas del *array* que se encuentra en la figura anterior, por lo que se puede observar que cada una de las casillas está formada por cuatro filas en horizontal y cuatro filas en vertical, describiendo el comportamiento comentado en la parte superior.

Para entenderlo de otra forma, cada una de las casillas que, en un *array* lineal, representaba un único elemento, en este caso, representa una matriz completa. Es decir, una matriz de 4x4, en un

1				4			
1,1	1,2	1,3	1,4	F1,F2	F1,F3	F1,F4	
2,1	2,2	2,3	2,4				
3,1	3,2	3,3	3,4				
4,1	4,2	4,3	4,4				
				F2,F1	F2,F2	F2,F3	F2,F4
				F3,F1	F3,F2	F3,F3	F3,F4
				F4,F1	F4,F2	F4,F3	F4,F4

**Figura 2.11: Forma de representar el *array* plano con la matriz de una de sus casillas**

*array* plano, tendrá a su vez, una matriz 4x4 en cada uno de sus elementos.

Con esta explicación finaliza el capítulo de desarrollo teórico. En los siguientes puntos, trataremos de automatizar todos los cálculos comentados en este apartado, llevando todo lo explicado a un código que nos permita obtener la respuesta final. Es importante comentar que, a lo largo del siguiente capítulo, se hará referencia en muchos aspectos al proceso teórico desarrollado en este apartado.



## Capítulo 3

# Creación del entorno de simulación

En este apartado, comienza el proceso para desarrollar un programa que nos permita automatizar los cálculos detallados en el apartado anterior.

Para ello, se van explicar cada uno de los procesos realizados durante el trabajo, para poder llevar esta idea a la práctica, y poder conseguir un programa que permita obtener los resultados deseados a través de los dos entornos principales: *Matlab* y *Feko*.

A la hora de comenzar a desarrollar nuestro programa, y teniendo en cuenta los diferentes aspectos comentados anteriormente, es de vital importancia conocer los diferentes archivos y procesos que nos servirán como datos a la hora de obtener la respuesta final.

Para completar este apartado, serán necesarios una serie de archivos que se generarán con el entorno de desarrollo *Feko* y corresponderán al diagrama de radiación de la antena aislada y a la matriz de admitancias del *array*, y, por otra parte, se detallarán las principales características, procesos y líneas de código que corresponden a las distintas funciones que nos permitirán completar cada uno de los procesos comentados en el desarrollo teórico y que nos permitirán obtener la respuesta final.

A continuación, pasaremos a sintetizar el programa *Matlab* y a tratar de llevar a cabo un diseño más limpio y profesional, utilizando la herramienta *Appdesigner*, de *Matlab*, que nos permitirá configurar los diferentes diagramas y archivos, para un análisis completo de los resultados.

Con todo ello, obtendremos las respuestas finales de nuestro programa, y podremos visualizar los resultados en el propio entorno (se irá detallando a lo largo de este capítulo, en que pestañas podremos visualizar los resultados).

### 3.1. Appdesigner

Para comenzar, debemos explicar de forma breve, las principales características del entorno de simulación en el cual vamos a desarrollar nuestro programa final.

Este entorno, no será otro que *Appdesigner*, una herramienta del entorno *Matlab*, que permite crear aplicaciones profesionales de una forma intuitiva y clara, sin necesidad de tener conocimientos previos a la hora de realizar entornos programados por uno mismo (tal y como nos explica la guía de usuario [4]).

La utilidad de este entorno, que lo hace tan intuitivo y sencillo, reside en que podemos crear el aspecto de nuestro programa a partir de componentes visuales de uso sencillo y eficaz, para que, tanto el usuario, como el programador, no necesiten de una formación para realizar el programa.

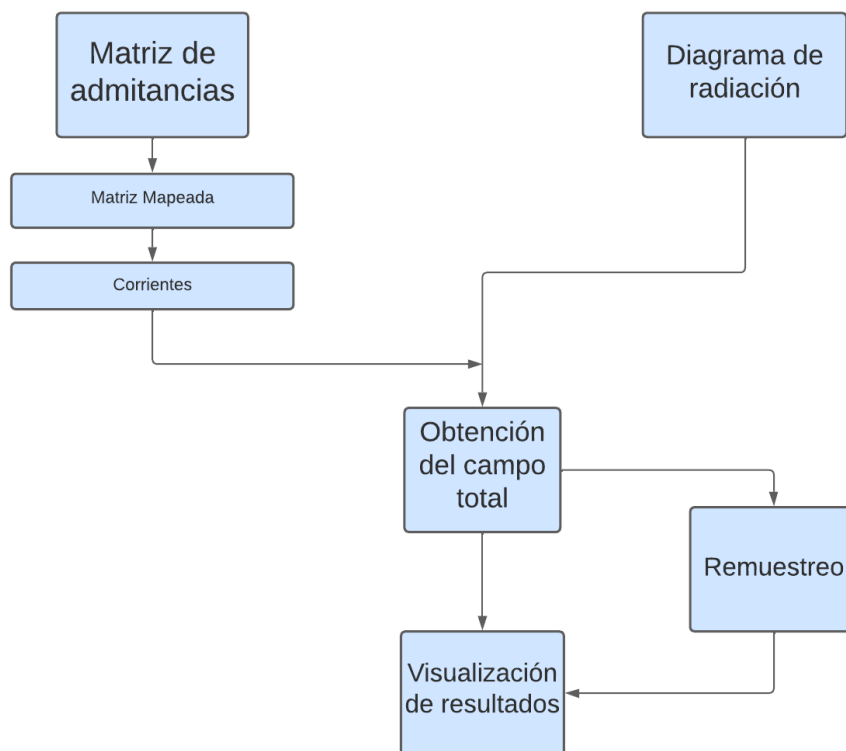
*Appdesigner*, también integra las dos tareas principales de la creación de aplicaciones: diseñar los componentes visuales, como se ha comentado anteriormente y programar el comportamiento de la aplicación, de una forma bastante sencilla y muy similar a como se realizaría con el entorno *Matlab*, familiar para nosotros.

Es, por todas estas funciones y por ser el entorno recomendado para crear aplicaciones en *Matlab*, que este entorno será el elegido, para crear nuestra versión del programa final.

A continuación detallaremos paso por paso, que es aquello que vamos a programar con esta herramienta del entorno *Matlab*, que nos permitirá automatizar lo desarrollado en el capítulo 2, y como vamos a dividir cada uno de los bloques que van a formar el desarrollo de nuestro proyecto

## 3.2. Diagrama de bloques

Para tener una visión más clara y esquematizada de los procesos que se van a llevar a cabo, para automatizar los cálculos teóricos, debemos dividir el trabajo en bloques, como vemos en la figura 3.1.



**Figura 3.1: Diagrama de bloques del proyecto**

Este diagrama de bloques coincide con los pasos desarrollados en el apartado teórico y co-

mienza con un primer bloque, cuya finalidad es obtener la matriz de admitancias de la agrupación de menor tamaño (que, como veremos a continuación, se obtendrá mediante un proceso que se ha de llevar a cabo en *Feko*), mientras que, en paralelo, se obtiene el diagrama de radiación del elemento radiante; ambos datos necesarios para obtener el posterior producto del factor de *array* por el campo total.

A continuación, y a partir del archivo correspondiente a la matriz de admitancias, lograremos obtener la matriz mapeada, a partir de funciones del entorno *Matlab*, teniendo en cuenta el proceso desarrollado en los procedimientos teóricos, para, posteriormente, obtener las corrientes correspondientes a los terminales de la agrupación, del mismo modo.

Posteriormente, gracias a la teoría de antenas, debemos saber que tendremos que realizar el producto del factor de *array* y del campo de una antena aislada, para obtener finalmente, las características de la agrupación deseada.

Una vez concluido este bloque, el programa pasará por dos bloques. Uno de visualización directa del resultado del producto comentado anteriormente, y un paso opcional de remuestreo, en el caso de que sea necesario (posteriormente se explicará de manera clara, en que consiste este proceso, y se entenderá el porqué, de que se trate de un bloque opcional).

A lo largo de este capítulo se explicarán cada uno de los procesos implicados en la resolución de los bloques expuestos, detallando, además, como se incluyen en el programa final desarrollado con la herramienta *Matlab, Appdesigner*

Con todo ello, pasamos a desarrollar cada uno de los bloques que componen nuestro trabajo, tratando de completar las explicaciones basándonos en los procesos teóricos que se han desarrollado en el capítulo 2.

### 3.3. Bloque 1: Matriz de admitancias

Como bien hemos comentado en el apartado teórico, el principal elemento que nos va a permitir mapear las características de un *array* de menor tamaño, para obtener las características de una agrupación mayor, va a ser la matriz de admitancias.

Por tanto, debemos describir en primer lugar como obtener este archivo correspondiente a la matriz de admitancias que servirá como dato para las funciones posteriores.

Para ello, el proceso de creación de la agrupación, estará a cargo del entorno *Feko*, que nos permite tanto crear, como simular y visualizar diferentes configuraciones de antenas, diagramas de radiación, características de los diferentes tipos de antenas y a su vez, características de los *arrays*.

El proceso se basa en que los elementos del *array* se han de crear de manera independiente y a mano. Para ello, creamos un dipolo y lo copiamos trasladándolo a cada posición correspondiente a los elementos de esta agrupación.

A continuación, deberemos solicitar que el programa nos proporcione los parámetros [S] de la agrupación, ya que, como hemos visto en el apartado teórico, los cálculos necesarios para obtener los objetivos del proyecto, no serían posibles sin aplicar la teoría de cuadripolos y de los parámetros [S],

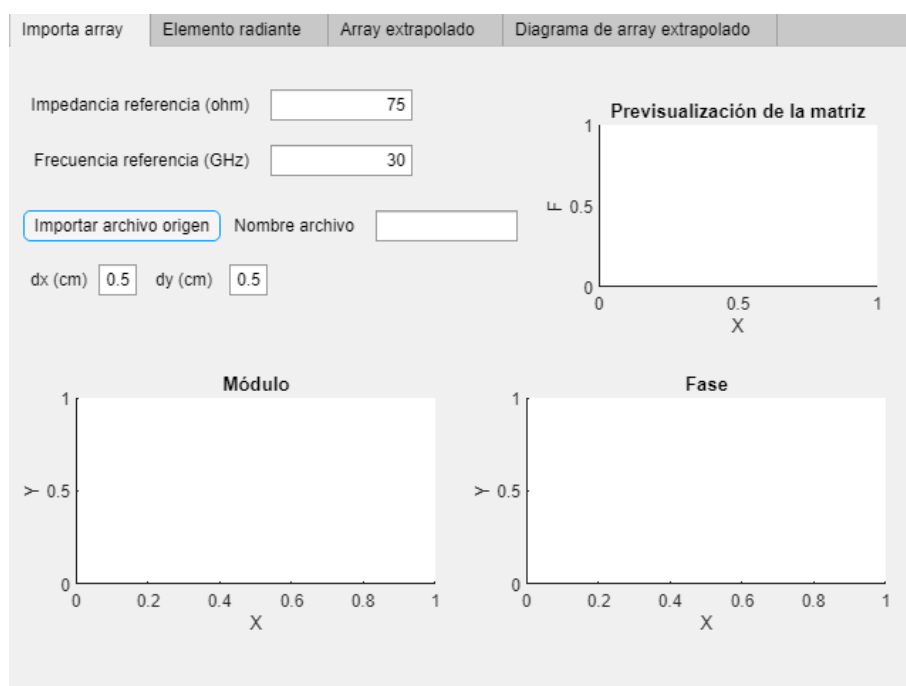
Sin embargo, una vez calculados los parámetros, para nuestro propósito, deberemos solicitar

que *PostFeko* nos proporcione la matriz de admitancias a partir de los parámetros [S]. Los parámetros [Y], deberemos exportarlos sin normalizar y en formato parte real/imaginaria.

Con todo ello, se obtiene el archivo necesario correspondiente a las admitancias de la agrupación, que nos servirá como dato para el resto de funciones y que tiene el formato *.yNf*, siendo “N” un valor que coincide con el número de elementos del *array*.

Una vez comprendido cómo obtener los diferentes archivos correspondientes a la matriz de admitancias, debemos describir cómo este archivo se incluye dentro del programa de *Appdesigner*, que hemos desarrollado.

En la imagen 3.2 se aprecia la primera pestaña de nuestro programa, correspondiente a la importación y visualización de los datos más importantes.



**Figura 3.2: Primero de los bloques que conforman el programa**

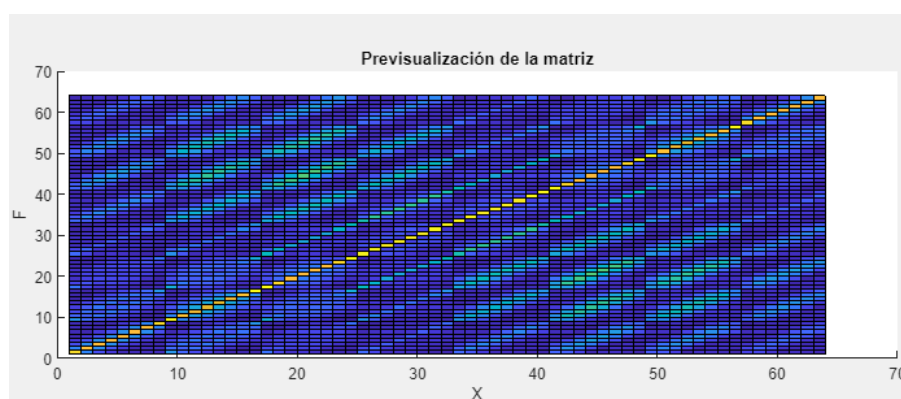
En esta pestaña encontramos diferentes campos que el usuario ha de especificar y que aparecen con valores predeterminados (que podrán modificarse si el usuario así lo considera). Estos campos son los siguientes:

- Frecuencia de referencia: Uno de los datos más importantes de nuestro diseño, que en general estará expresado en el orden de Gigahercios y que en nuestro programa toma un valor por defecto de 30 GHz.
- Impedancia de referencia: En nuestro caso especificado en ohmios y con un valor de referencia de 75 ohmios por defecto. Será otro de los datos principales para los cálculos posteriores (como se puede comprobar en el apartado teórico)
- Botón que importa el archivo de origen: Este archivo de origen será el correspondiente a la matriz de admitancias dato, obtenida en este mismo apartado mediante *Feko* y que está formado por la extensión *.yNp*.



- Espaciado en el eje x y en el eje y: Este valor de espaciado entre elementos (distancia entre los diferentes elementos de la agrupación), estará especificado en centímetros y tomará un valor de referencia de 0.5 cm. Tendremos que especificarlo tanto en el eje x como en el eje y.

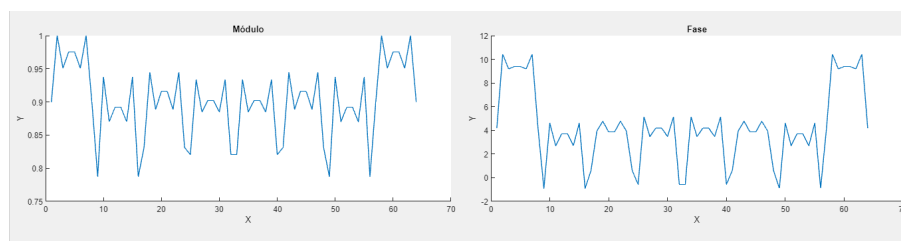
Una vez incluidos todos los datos, necesarios para poder realizar los procesos de obtención de la matriz mapeada y la posterior extracción de las corrientes, podemos observar una primera gráfica que previsualiza el aspecto de la matriz de admitancias del archivo seleccionado, para, en la parte inferior, dejar paso a la representación del módulo y la fase de las tensiones efectivas (figura 3.3).



**Figura 3.3: Gráfica que muestra el aspecto de la matriz de admitancias**

Como puede observarse por el aspecto que tiene la matriz de admitancias de ejemplo, de la figura anterior, podemos observar, que se trata de un *array* plano donde, como hemos visto en la teoría del capítulo 2, cada una de las casillas, se comportará como una matriz más pequeña, afirmación que puede verificarse en la figura, donde además, observamos como, los elementos de la diagonal de cada una de las casillas (matrices en este caso), tienen un color distinto al resto de elementos, y esto se debe a que las amplitudes de los pesos de las corrientes, son de mayor tamaño, siendo en la diagonal principal de toda la matriz completa, el mayor tamaño de todas las amplitudes.

De igual forma, podemos visualizar el módulo y la fase de las tensiones efectivas, como vemos en la figura 3.4:



**Figura 3.4: Gráficas que muestran el módulo y la fase del archivo importado**

Estas, se han obtenido gracias al procedimiento especificado en el apartado teórico, pero llevando a cabo un proceso automatizado que se detallará posteriormente.

De este modo, concluye el primer bloque de nuestro diagrama, donde se ha generado el archivo

correspondiente a la matriz de admitancias, se ha visualizado su aspecto, y se han solicitado los principales datos que serán necesarios para continuar con el desarrollo del programa.

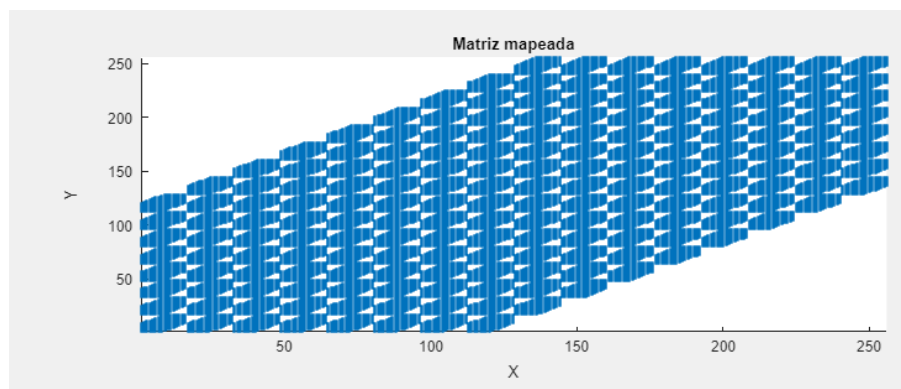
### 3.4. Bloque 2: Matriz mapeada

Si se observa el diagrama de bloques del apartado anterior, podremos observar como, el siguiente paso para llegar a visualizar los resultados finales, consiste en obtener la matriz mapeada a partir del archivo correspondiente a la matriz de admitancias obtenida en el primer bloque.

Para ello, se hará uso de una función *Matlab*, que permitirá mapear una matriz (que corresponde a la matriz de impedancias/admitancias de un *array* plano 4x4 u 8x8), en otro de dimensiones mayores. Esta función además está desarrollada teniendo en cuenta la posibilidad de que el *array* objetivo pueda tener distinto número de elementos en cada dimensión.

Los únicos datos necesarios para que la función realice su labor, es la matriz de entrada (en nuestro caso la matriz de admitancias obtenida en el apartado anterior), la dimensión en el eje x del *array* de salida, es decir, la dimensión correspondiente a las filas del *array* y la dimensión en el eje y de la agrupación de salida, o dicho de otro modo, la dimensión correspondiente a las columnas de la agrupación.

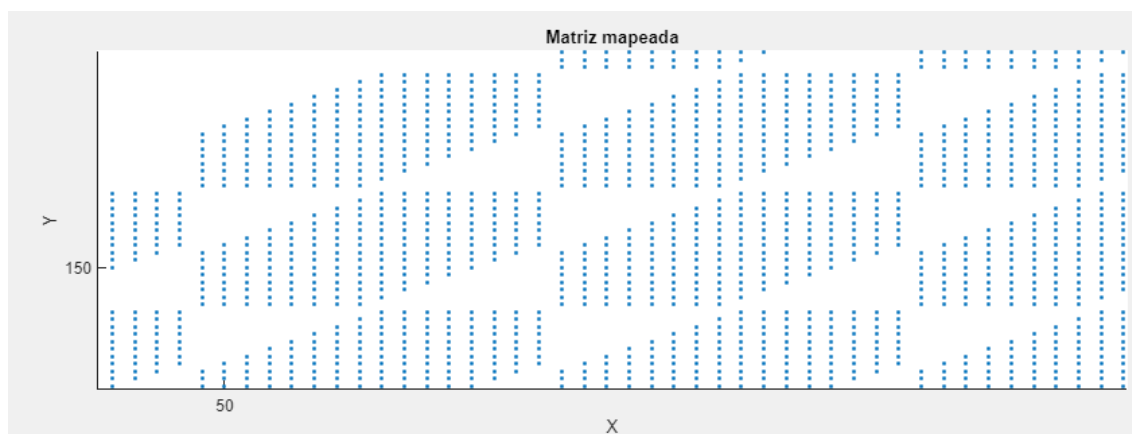
Con todo ello, el resultado de la función, será la matriz objetivo (o de salida) de NxN elementos, que corresponderá a nuestra matriz mapeada (figura 3.5).



**Figura 3.5: Matriz mapeada**

De nuevo, podemos observar como, al igual que la matriz de admitancias, a partir de la cual, se ha realizado el mapeo, esta matriz también es diagonal, solo que en este caso tiene la particularidad de que, cada elemento, es una matriz diagonal más, como puede apreciarse en la figura 3.6:

Conviene aclarar que, por claridad en la visualización, el programa dibuja únicamente un punto, y no un valor numérico, en aquellos elementos de la matriz que son distintos de cero y nada donde la matriz es cero.



**Figura 3.6: Zoom en uno de los bloques de la matriz mapeada**

### 3.5. Bloque 3: Corrientes

Como ya se ha comentado en los apartados teóricos, una vez se ha obtenido la matriz mapeada, gracias a las respectivas funciones del programa y al archivo correspondiente a la matriz de admitancias, el siguiente paso, consiste en obtener las corrientes en los terminales del *array*, gracias a esta matriz mapeada. Para ello, se ha desarrollado una nueva función en *Matlab*, que nos permite llevar a cabo dicha tarea.

Para ello, debemos, en primer lugar, obtener tanto las tensiones, como las impedancias efectivas del *array* (como bien se ha comentado en el apartado 2.3 del capítulo 2). Para ello, la función principal que se ha utilizado, nos permite calcular las tensiones y las impedancias efectivas de un *array*, a partir de la matriz de admitancias de los elementos radiantes que lo forman.

Por tanto, el primer paso para comenzar con la función, ha sido leer la matriz de admitancias dato (obtenida en los apartados anteriores), para, a continuación automatizar todos los cálculos teóricos comentados en el “Capítulo 2: Principios teóricos”. Además, todo este proceso, ha sido aquel que nos ha permitido expresar tanto el módulo como la fase de las tensiones eficaces, correspondientes a la primera pestaña de nuestro programa, comentada en el apartado 3.3 (3.4).

Finalmente, como sabemos por teoría, teniendo en cuenta que la corriente efectiva, no es más que el cociente entre la tensión efectiva y la impedancia efectiva, podemos obtener de forma sencilla las corrientes del *array* objetivo.

Comentar además, que también se ha incluido en esta función, el proceso para pasar el resultado a la forma matricial, teniendo en cuenta la tensión efectiva y las dimensiones de destino tanto en el eje “x”, como en el eje “y”.

Con estos dos procesos definidos en los puntos 3.4 y 3.5, se completa una de las ramas del diagrama de bloques, pero para poder completar nuestro proceso de desarrollo en bloques, de forma paralela a los procesos detallados en las páginas anteriores, hay que sumar un nuevo proceso: la obtención del diagrama de radiación de la antena aislada.

Este proceso es de vital importancia, ya que sin él, no podría desarrollarse el paso siguiente, que consiste en poder calcular el campo total, que se obtiene del producto del factor de *array* y de este diagrama de radiación.

### 3.6. Bloque 4: Diagrama de radiación

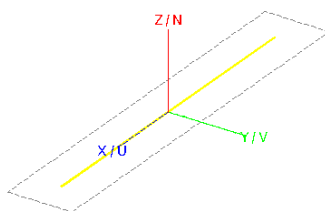
Teniendo en cuenta la teoría de antenas y las diferentes explicaciones comentadas en el apartado teórico (teniendo en cuenta los resultados obtenidos en el apartado anterior), debemos saber que para obtener el campo total de la agrupación y de ese modo, poder obtener y representar la agrupación objetivo, hemos de recurrir al producto del factor de *array* de la agrupación de menor tamaño y el diagrama de radiación del elemento radiante.

Gracias al archivo correspondiente a la matriz de admitancias de la agrupación de menor tamaño, obtenida en el apartado anterior, podemos realizar este producto extrayendo las corrientes de cada uno de los terminales del nombrado archivo, como se ha completado anteriormente.

Es por ello, que el siguiente paso para poder avanzar en nuestro proyecto, es obtener el archivo correspondiente al diagrama de radiación del elemento radiante.

En este capítulo se mostrará el proceso y el resultado característico del diagrama de radiación de, por ejemplo, un dipolo, por ser el elemento principal de este tipo de procesos, a pesar de que el proceso que se detalla a continuación, es común para cualquier antena aislada, y, para demostrar esta última afirmación, comprobaremos los resultados obtenidos en el capítulo 4, para una antena aislada de tipo parche.

Si suponemos que comenzamos por el dipolo, únicamente es necesario crear una nueva antena de tipo *wire* en *Feko*, para colocar nuestro dipolo en los ejes correspondientes (figura 3.7).



**Figura 3.7: Dipolo**

Después de especificar los datos, que en este caso introducirá el usuario, y después de crear los diferentes puertos, fuentes de voltaje y fuente de frecuencias, tendremos que analizar el resultado de nuestra simulación, abriendo el programa *PostFeko*, donde podremos visualizar los resultados del diagrama de radiación, con su forma característica (Un lóbulo con forma de anillo o rosquilla).

A continuación, debemos exportar el fichero correspondiente al diagrama de radiación, en este caso con una extensión ".ffe" o *FarField*, ya que esta extensión será la necesaria para poder importar el archivo a nuestro programa desarrollado en *Appdesigner*, el cual, abordará todo este proceso en su segunda pestaña.

Como podemos observar en la figura 3.8, se ha desarrollado el entorno, para poder importar los archivos teniendo en cuenta su origen, tanto desde *Feko*, como desde *CST*.

Al accionar el botón correspondiente para importar el archivo, emerge una ventana (figura 3.9), que nos permitirá seleccionar el archivo correspondiente, con la extensión característica de este tipo de archivos.

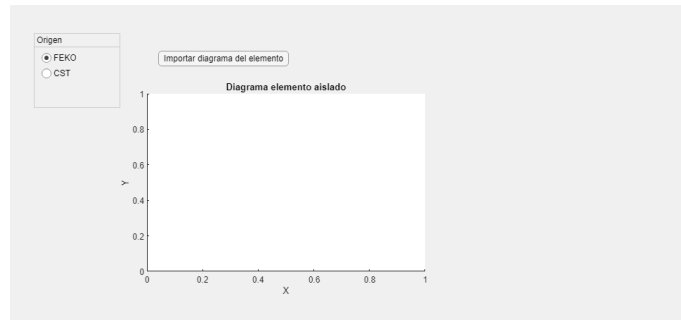


Figura 3.8: Segunda pestaña del programa desarrollado con *Appdesigner*

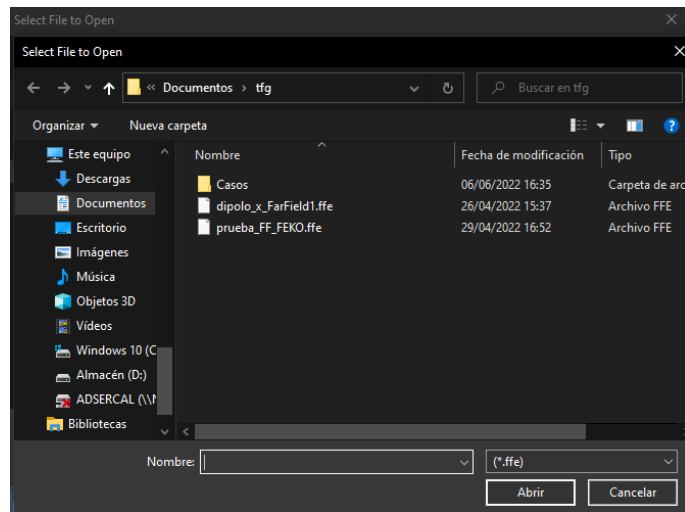


Figura 3.9: Ventana emergente tras seleccionar el botón de importar diagrama de radiación

Por otra parte, tendremos una gráfica que nos permitirá visualizar este diagrama de radiación de la antena aislada y así comprobar que hemos importado el diagrama correcto (3.10).

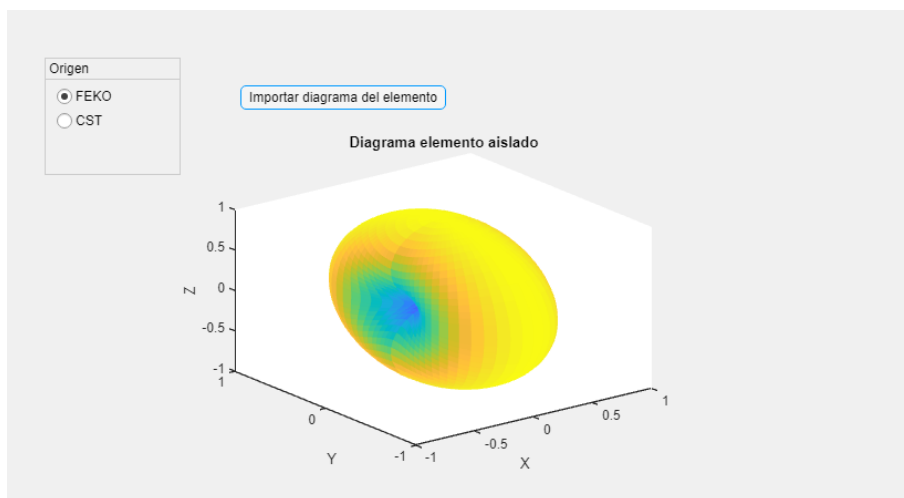


Figura 3.10: Diagrama de radiación del dipolo visto desde el programa desarrollado

Con el archivo importado y guardado en una variable, se puede pasar al siguiente bloque de nuestro diagrama, el cual, se basará en llevar a cabo el cálculo del campo total, teniendo en cuenta el factor de *array* que se obtendrá con los procesos explicados en los primeros bloques, y el diagrama de radiación, explicado en este apartado.

### 3.7. Bloque 5: Campo total

Como se ha comentado anteriormente, uno de los elementos característicos y el más importante dentro de las agrupaciones de antenas es el conocido factor de *array*.

Dentro de la múltiple información que nos permite obtener este elemento de los *arrays*, la más útil para este proyecto es aquella que nos permite obtener el campo total de la agrupación objetivo a partir del producto de este, por el diagrama de radiación de la antena aislada.

Para ello, de nuevo, hemos recurrido a una función desarrollada en *Matlab* y que nos va a permitir automatizar el proceso de cálculo comentado.

En primer lugar, debemos calcular el factor de *array*, a partir del archivo correspondiente a la matriz de admitancias obtenida anteriormente. Para ello, utilizamos una función que nos permitirá calcular este factor, para elementos equiespaciados en 1D o 2D.

Para ello, las entradas necesarias serán, el vector fila de amplitudes de corriente en el eje x en el eje y, el desfase progresivo en x y en y (grados), el espaciado de nuestra agrupación, tanto en el eje x como en el eje y, la frecuencia normalizada de nuestro *array*, y finalmente, las matrices de los ángulos de observación  $\theta$  y  $\phi$ .

Con todo ello, podremos obtener nuestro factor de *array*, y únicamente deberíamos realizar el producto con el diagrama de radiación de la antena aislada, y, para ello, se extraerá el campo del archivo correspondiente al diagrama de radiación, para finalmente, obtener el campo total, que será el elemento fundamental, para poder realizar la posterior representación final, del diagrama de radiación de nuestra agrupación objetivo.

### 3.8. Bloque 6: Visualización

Una vez realizado todo el proceso para obtener los resultados finales, debemos especificar una nueva serie de datos, que nos permitirán ajustar el resultado a los nuevos parámetros introducidos por el usuario (figura 3.11).

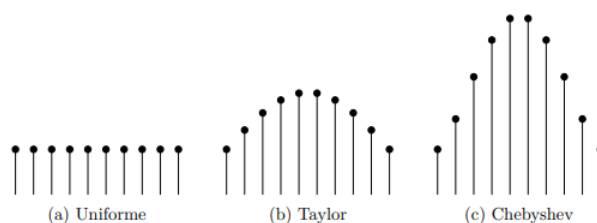
- N° elementos en “x” y en “y”: Para comenzar, será necesario especificar el número de elementos que tendrá la agrupación objetivo tanto en el eje “x” como en el eje “y”. Los valores por defecto que toman estos campos son de 16 elementos en cada eje.
- Distribución en “x” y en “y”: En este caso, tendremos dos desplegables que nos permitirán seleccionar el tipo de distribución que deseamos en cada uno de los ejes, pasando por las tres distribuciones características de las agrupaciones de antenas: Uniforme, Taylor y Chebyshev. Posteriormente se especificarán las características de cada una de ellas. En este caso, por defecto, se encuentra la distribución uniforme como opción.

N°Elementos en X	<input type="text" value="16"/>
N°Elementos en Y	<input type="text" value="16"/>
Distribución en X	Uniforme ▼
Distribución en Y	Uniforme ▼
NLPS en X (dB)	<input type="text" value="13.5"/>
NLPS en Y (dB)	<input type="text" value="13.5"/>

**Figura 3.11: Datos necesarios para realizar el proceso**

- NLPS “x” y en “y”: Finalmente, nuestro programa ofrece la posibilidad de especificar el NLPS, en cada uno de los ejes, ya que la especificación de un determinado NLPS, para el diagrama de radiación de un *array* se traduce en una distribución de amplitudes con un determinado perfil de caída, desde el elemento central hacia los elementos de los bordes de la agrupación. La unidad fundamental será el decibelio y en nuestro caso el valor predeterminado será de 13.5 dB en ambos ejes.

Sobre las distintas distribuciones comentadas anteriormente, de todas ellas, la uniforme es la que proporciona haces más estrechos a costa de mostrar lóbulos secundarios más altos. Además, las distribuciones de Taylor y la de Chebyshev permiten obtener niveles de lóbulos muy bajos, así como, ofrecen la posibilidad de prefijar de antemano el NLPS deseado. Por otra parte, la distribución de Chebyshev tiene la particularidad de que todos los lóbulos secundarios presentan el mismo nivel (3.12).

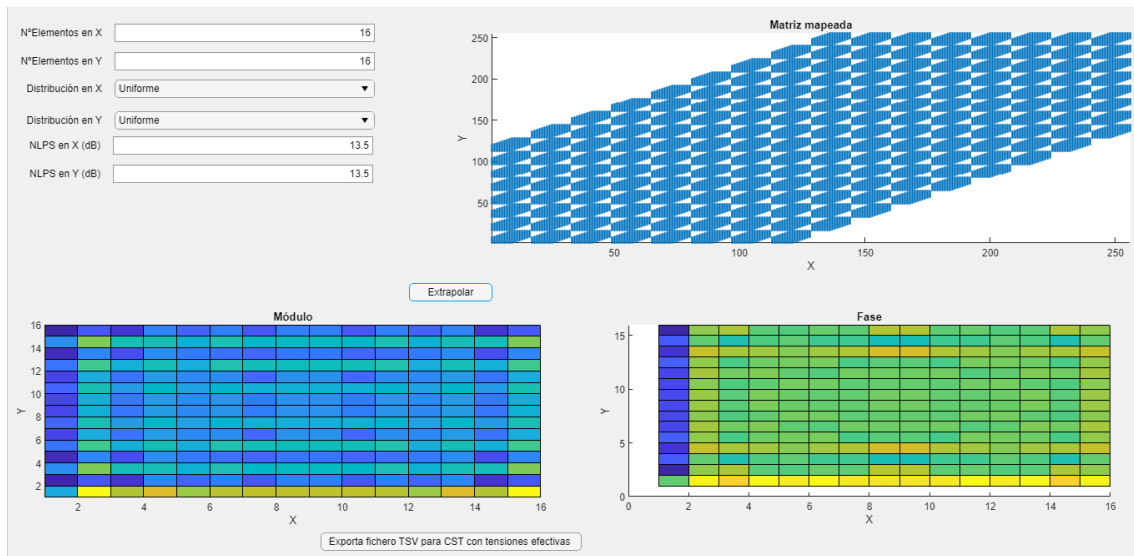


**Figura 3.12: Aspecto de las diferentes distribuciones de corrientes**

Con todos los datos explicados, en la pestaña completa nos encontramos con una serie de gráficas, para visualizar todos los resultados posibles. Entre ellas se encuentra la gráfica correspondiente a la matriz mapeada 3.5 y las gráficas que se detallan en la figura 3.13.

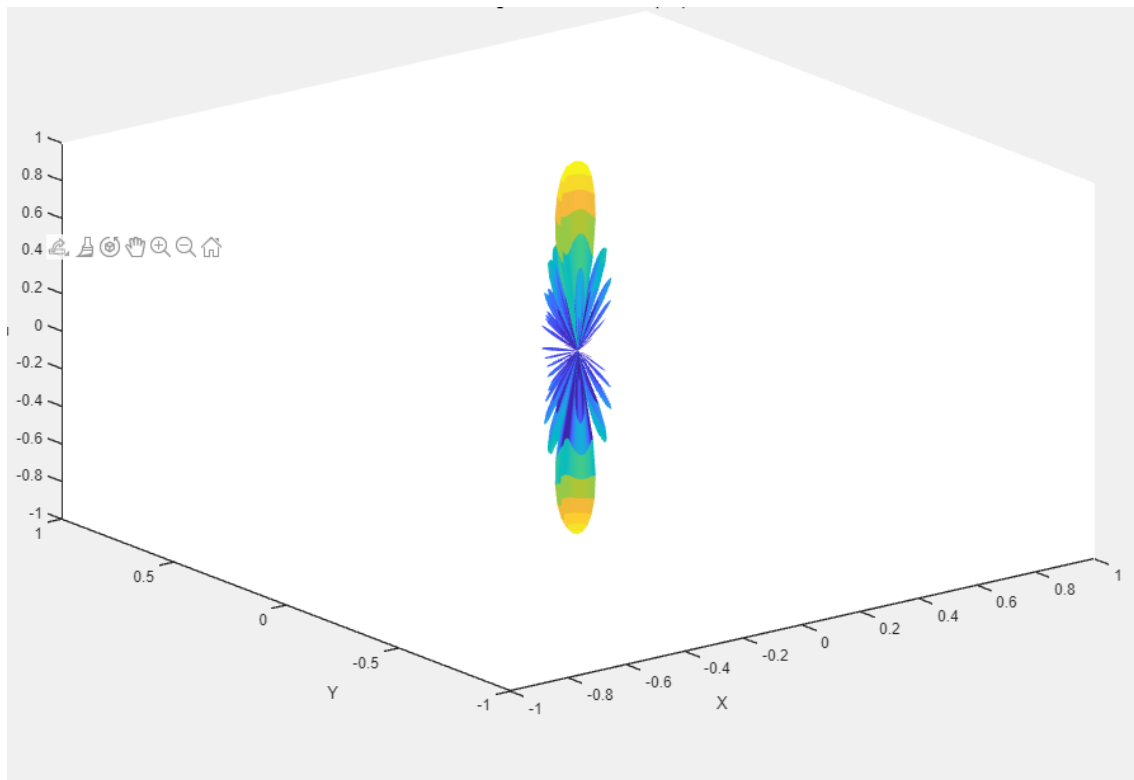
Estas dos gráficas, representan aquello que se ha comentado en la primera pestaña de nuestro programa, representan tanto el módulo como la fase de las amplitudes de los pesos de las corrientes (solo que, en este caso, para la matriz mapeada), donde de nuevo, podemos apreciar ese cambio de colores, fríos para amplitudes de menor tamaño, y más cálidos para amplitudes de un mayor tamaño.

Por último, y para poder visualizar el diagrama de radiación final, hemos de pasar a la siguiente pestaña (tomando la posibilidad de que sea necesario un remuestreo, que se comentará en el



**Figura 3.13: Tercera pestaña de nuestro programa completa**

apartado siguiente), donde encontraremos una gráfica que nos dará el resultado final de todos los procesos de nuestro programa desarrollado en *Matlab* (3.14).



**Figura 3.14: Visualización final del diagrama de radiación objetivo**



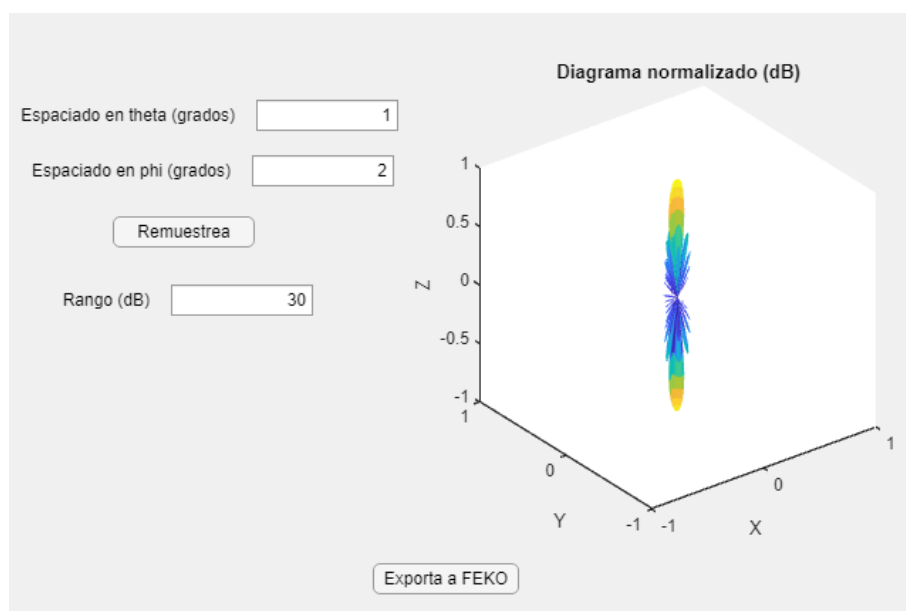
### 3.9. Bloque 7: Remuestreo

Una vez obtenemos nuestra matriz mapeada, debemos pasar al proceso de visualización de esta, pero en ciertas ocasiones, si nos encontramos ante diagramas de radiación con anchos de haz muy estrechos, que quedan muy por debajo del muestreo especificado inicialmente, muchos de los lóbulos del factor de *array*, quedarán ocultos.

Es por ello, que el próximo bloque del que hablaremos (el cual solo se lleva a cabo, si ocurre el fenómeno citado anteriormente), es el proceso de remuestreo.

Si tomamos como ejemplo un diagrama de radiación con un ancho de haz de 1 grado sexagesimal y nuestro muestreo principal se encuentra a 5 grados sexagesimales, van a quedar ocultos un gran número de lóbulos secundarios.

Para solventar este problema, la última pestaña de nuestro programa, solicita los datos presentes en la figura 3.15, para saber si es necesario ajustar el muestreo principal.



**Figura 3.15: Datos solicitados por el programa desarrollado para llevar a cabo el remuestreo (si fuera necesario)**

Entre los diferentes datos especificados, nos encontramos con el espaciado en los ejes fundamentales (ya que al tratarse de una modificación en el diagrama de radiación, estamos hablando de coordenadas esféricas), y el rango en decibelios.

Los datos predeterminados que, al igual que todos los anteriores, pueden ser modificados por el usuario, se especifican en 1 grado, para el espaciado en  $\theta$  y 2 grados en  $\phi$ , con un rango de 30 decibelios.

Para llevar a cabo el proceso, se utiliza una de las funciones desarrolladas durante el proyecto. Esta, nos permite muestrear más fino el diagrama de radiación a partir de la modificación en los datos comentados anteriormente.

Las entradas de esta función serán, la matriz de ángulos a interpolar en  $\phi$  (grados), la matriz de ángulos a interpolar en  $\theta$  (grados), matriz de ángulos original  $\phi$  (grados), la matriz de ángulos original  $\theta$  (grados) y finalmente, la amplitud de la función a remuestrear.

Por otra parte, las salidas de la función serán la matriz de ángulos  $\phi$  de salida (grados), la matriz de ángulos  $\theta$  de salida (grados) y finalmente, la matriz de amplitudes tras remuestrear.

Este proceso será de vital importancia si queremos que nuestro diagrama de radiación muestre por pantalla el mayor número de lóbulos posible.

Para finalizar, nos encontramos ante dos botones, que al ser accionados realizan las funciones de remuestreo (es decir, todo el proceso comentado anteriormente) y de exportación a *Feko*, para poder comprobar los resultados finales.

Esta comprobación se realizará en el próximo capítulo de esta memoria, donde comprobaremos si los procesos desarrollados durante las líneas anteriores, han sido los correctos.

## Capítulo 4

# Resultados

Después de haber desarrollado nuestro entorno de simulación, tratando de automatizar los cálculos y procesos teóricos, que se incluyen en el segundo capítulo de esta memoria, es el momento de obtener, analizar y comprobar si los resultados obtenidos por nuestro programa, son los esperados y si realmente se ha conseguido crear un entorno que permita llevar a cabo el objetivo principal de este proyecto.

Para analizar los diferentes resultados obtenidos, hemos de obtener la respuesta de nuestro programa ante una serie de datos, siendo los más importantes, aquellos que corresponden a la matriz de admitancias de la agrupación de origen y el diagrama de radiación de la antena aislada, y comparar los resultados con aquellos que el entorno de simulación *Feko*, obtiene de manera directa.

Podría entenderse por tanto, que la forma de comprobar si nuestro programa funciona correctamente, es comparando la respuesta que se obtiene, en forma de diagrama de radiación de la agrupación objetivo, del *array* extrapolado, con nuestro programa, con la respuesta obtenida mediante las diferentes herramientas que *Feko* tiene a su alcance.

Este proceso de obtención de resultados, constará también de un apartado destinado a concluir que los cálculos, principios y suposiciones tomadas durante el desarrollo de toda la memoria, no solo se entienden para un dipolo (como antena aislada), sino que puede extenderse a cualquier tipo de antena, y, para ello, se obtendrán los resultados correspondientes, empleando como antena aislada, una antena de tipo parche, para posteriormente comentar estos resultados.

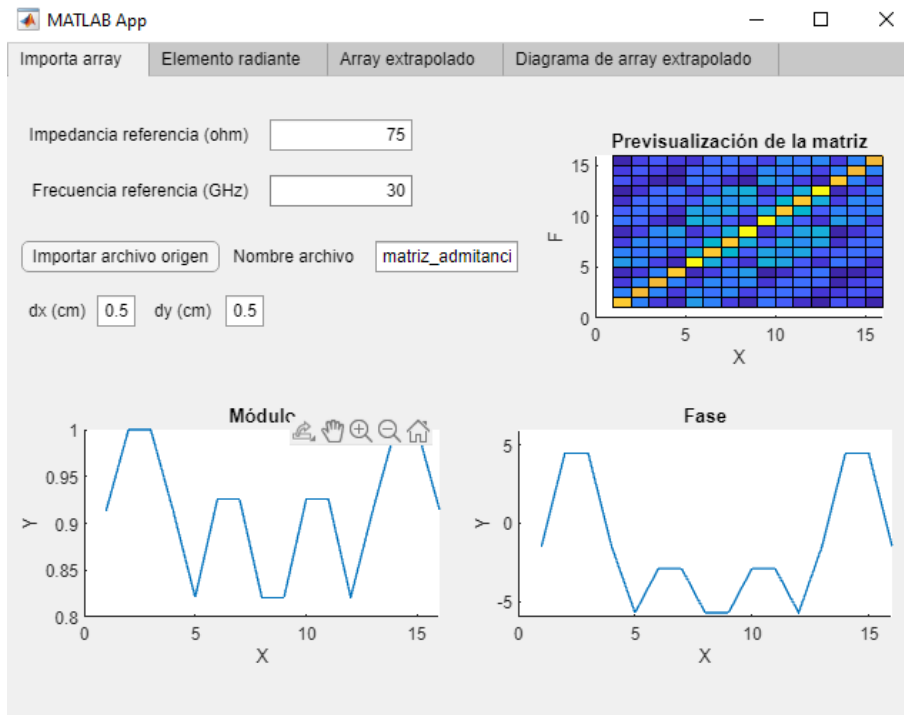
### 4.1. Análisis de resultados: Agrupación 8x8

Comenzamos a analizar los resultados obtenidos con nuestro entorno de simulación con el caso en el que, deseamos conseguir el diagrama de radiación de una agrupación objetivo de 8x8 elementos.

Para poder llevarlo a cabo, necesitaremos generar una matriz de admitancias dato, a partir de un *array* de 4x4 elementos (como vimos en el capítulo 3, dentro del bloque correspondiente a la matriz de admitancia).

En los siguientes párrafos, se comentará paso por paso el proceso de obtención de resultados, que comienza con la primera pestaña de nuestro programa, correspondiente al *array* importado

(figura 4.1).



**Figura 4.1: Primera pestaña de nuestro programa**

Como puede observarse en la imagen, los datos correspondientes a la frecuencia e impedancia de referencia, así como el espaciado entre elementos, se mantendrán con los valores predeterminados y, al importar la matriz de admitancias correspondiente a una agrupación 4x4, podremos observar la previsualización de la matriz, así como el módulo y la fase de las tensiones efectivas en los puertos de entrada de los 16 elementos del array.

A continuación importamos el archivo correspondiente al diagrama de radiación de la antena aislada, en este primer caso un dipolo, y pasaremos a calcular el *array* extrapolado, gracias a la tercera pestaña de nuestro programa desarrollado (figura 4.2), especificando que los elementos, en ambos ejes de la agrupación objetivo, van a ser 8 (el resto de datos quedan por defecto).

Observamos en la figura, tanto la matriz mapeada que se obtenía gracias a los datos especificados en la primera pestaña, como el módulo y la fase del *array* extrapolado.

Finalmente, el resultado obtenido en la cuarta pestaña de nuestro entorno, correspondiente al diagrama de radiación de nuestra agrupación objetivo de 8x8 elementos, a partir de una matriz de admitancias dato de una agrupación de 4x4 elementos y una antena aislada correspondiente a un dipolo, será el mostrado en la figura 4.3.

Comparemos ahora el resultado obtenido, con aquel que *Feko*, obtiene de forma exacta directamente sobre un *array* 8x8.

Para ello, exportaremos el fichero *FarField* que se obtiene tras la simulación aproximada en nuestro programa. Posteriormente, generaremos una agrupación de 8x8 elementos mediante la opción que lo permite en el entorno de desarrollo y simulación *Feko*. Una vez especificados las fuentes de tensión, puertos, *Mesh*, frecuencia... Simulamos la agrupación y observamos sus resultados en

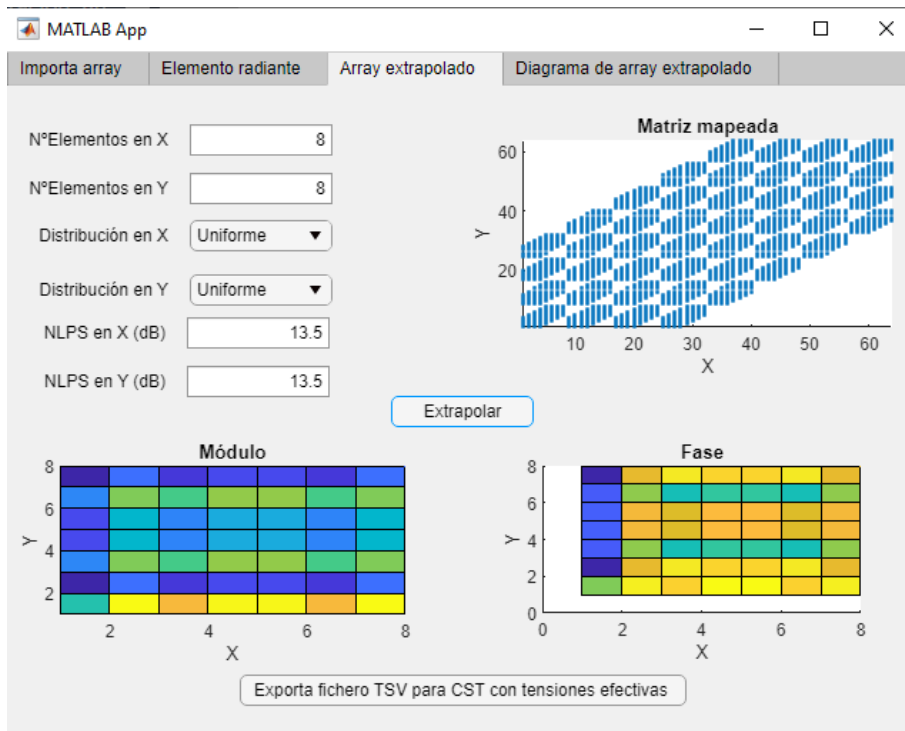


Figura 4.2: Tercera pestaña de nuestro programa correspondiente a la agrupación extrapolada

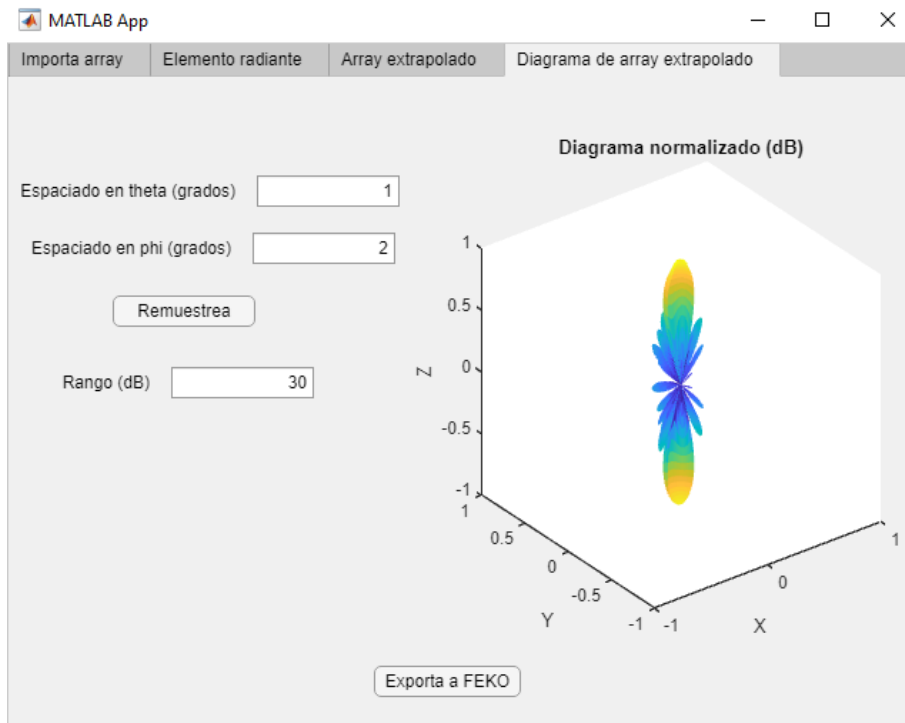
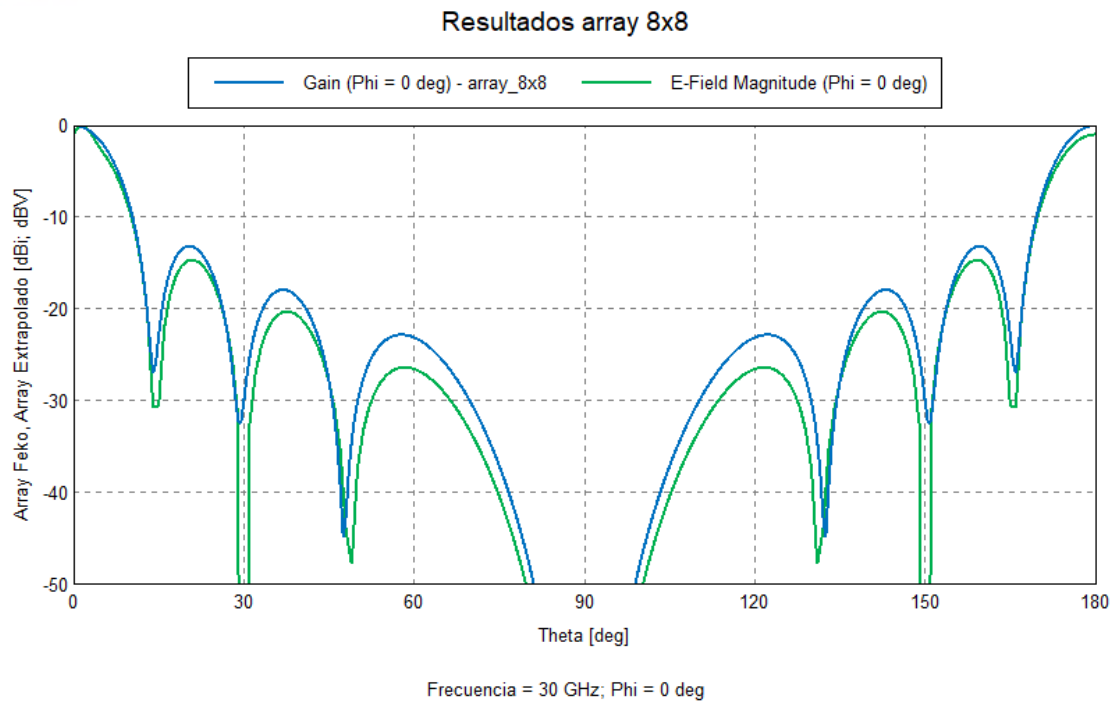


Figura 4.3: Diagrama de radiación de la agrupación objetivo de 8x8 elementos

*PostFeko*.

En nuestro caso, vamos a superponer ambos resultados en la misma gráfica para que la comparación sea lo más acertada posible y que pueda apreciarse de manera correcta. Para ello, elegiremos representar dos cortes dentro de *PostFeko*, dentro del eje  $\phi$ , de  $0^\circ$  y de  $90^\circ$ , para comparar los resultados en los dos cortes fundamentales del diagrama de radiación y así observar si ambos resultados coinciden en sus respuestas (4.4).



**Figura 4.4: Resultados obtenidos para un corte de  $0^\circ$**

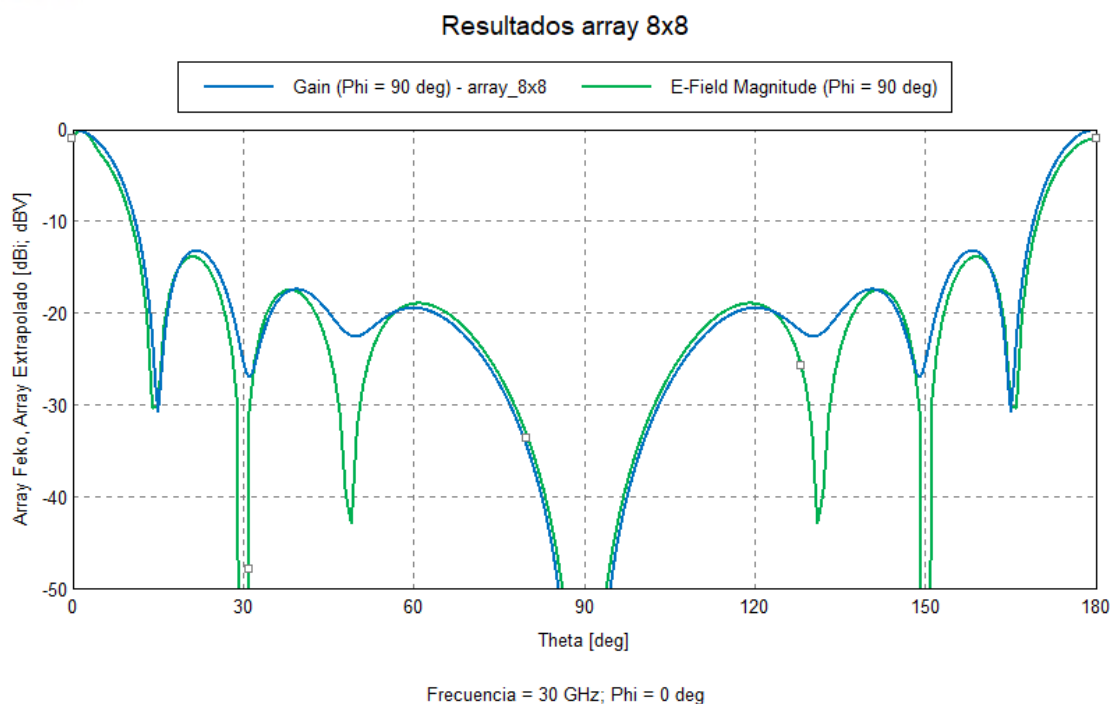
Como puede observarse, tendremos dos gráficas, representadas por colores azul y verde. El color azul representa la respuesta del diagrama de radiación para el *array* 8x8 obtenido directamente de *Feko*, mientras que el color verde representa la agrupación extrapolada desde un *array* 4x4 por nuestro entorno de desarrollo.

Como se puede comprobar, las respuestas de ambas, en este corte de  $0^\circ$  dentro del eje  $\phi$ , son prácticamente idénticas, dando por correctas todas las afirmaciones y todos los procesos que se han comentado en esta memoria.

Podemos observar además, que los lóbulos principales y los primeros lóbulos secundarios de ambas respuestas (aquellos más importantes), tienen un error de coincidencia muy pequeño, permitiendo así demostrar que nuestro entorno de desarrollo funciona correctamente, dentro de las limitaciones que presenta el fenómeno de los acoplos mutuos.

Para entender esta última idea, es de vital importancia fijarnos en los resultados obtenidos, en este caso, para el corte de  $90^\circ$  dentro del eje  $\phi$ , que se muestra en 4.5

En este caso, podemos observar (comparando con el anterior), que, de nuevo, el lóbulo principal y los primeros lóbulos secundarios coinciden en su respuesta de manera satisfactoria, sin embargo, a medida que nos alejamos del lóbulo principal, la respuesta de la agrupación extrapolada, tiende



**Figura 4.5: Resultados obtenidos para un corte de 90°**

a diferir de la obtenida automáticamente con *Feko*.

Esto tiene una sencilla explicación y son los acoplos mutuos. Si bien la influencia de los acoplos en el corte de 0° es prácticamente nula debido a que los diferentes dipolos se encuentran enfrentados en sus extremos (produciendo un nulo entre ellos y por tanto, reduciendo al máximo el acoplo mutuo), en el caso del corte de 90°, nos encontramos que los diferentes dipolos se encuentran en la posición que favorece la aparición de este fenómeno, produciendo que, a medida que nos alejamos del lóbulo principal, la respuesta ideal, es decir, aquella obtenida con *Feko*, no coincida tanto con la respuesta obtenida con nuestra agrupación extrapolada.

Con todo ello, podemos afirmar que la respuesta obtenida por parte de nuestro entorno desarrollado en *Appdesigner*, nos proporciona una muy buena respuesta ante el fenómeno de acoplos mutuos, permitiendo así, demostrar, que el programa podría servir para aligerar el coste computacional asociado al cálculo de *arrays* de gran tamaño.

Esta demostración deberá afirmarse al aumentar el número de elementos a algo más considerable, como 16 elementos en cada uno de los ejes, caso, que estudiaremos a continuación.

## 4.2. Análisis de resultados: Agrupación 16x16

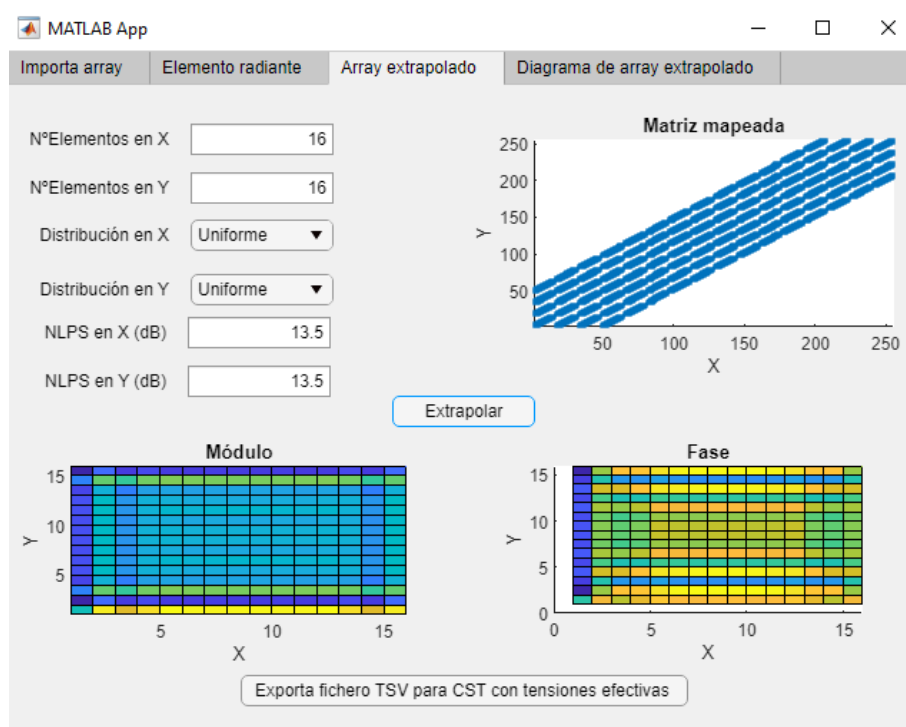
Visto lo anterior, es el momento de aumentar el número de elementos de nuestra agrupación objetivo, en este caso, tratando de alcanzar los 16 elementos en cada eje. Es decir, en este apartado se estudiarán y analizarán los resultados correspondientes a una agrupación objetivo 16x16.

De nuevo, y siguiendo el proceso que se siguió en el punto anterior, debemos pasar por la

primera de nuestras pestañas, dentro del entorno desarrollado e importar la matriz de admitancias dato, correspondiente a una agrupación de 4x4 elementos.

A continuación, será necesario importar el diagrama de radiación de la antena aislada, siendo de nuevo, en este caso, un dipolo (cuyo diagrama de radiación puede apreciarse en el capítulo 3, dentro del apartado en el que se comenta este aspecto).

Finalmente, llegamos a la tercera pestaña, donde en este caso, encontramos la diferencia fundamental con respecto al caso anterior, en este, debemos especificar que nuestra agrupación objetivo conste de 16 elementos en cada uno de los ejes (4.6).



**Figura 4.6:** Tercera pestaña de nuestro programa correspondiente al *array* extrapolado

Una vez más comprobamos el aspecto que tendría la matriz mapeada y el módulo y la fase de las amplitudes de los pesos de las corrientes para nuestra agrupación extrapolada, para finalmente, dejar paso a la última de las pestañas, que nos aporta el diagrama de radiación de la agrupación objetivo de 16x16 elementos, a partir de una matriz de admitancias de la agrupación dato de 4x4 elementos y con una antena aislada correspondiente a un dipolo (4.7).

De nuevo, en este caso, será necesario comparar el resultado con el que obtiene automáticamente *Feko* con sus herramientas. Para ello de nuevo, se superpondrán ambos diagramas de radiación para tener una visión más clara de la comparación (4.8).

Podemos observar que se demuestra finalmente lo afirmado en el caso anterior, el entorno desarrollado nos proporciona unos resultados considerablemente correctos que nos permiten comprobar que los objetivos del proyecto se cumplen, ya que, si observamos la gráfica (que sigue el mismo patrón de colores comentado anteriormente), la respuesta sigue siendo prácticamente idéntica.

Observamos de nuevo, que el lóbulo principal y los principales lóbulos secundarios son prácticamente coincidentes y que a medida que nos alejamos del lóbulo principal comienza a aparecer



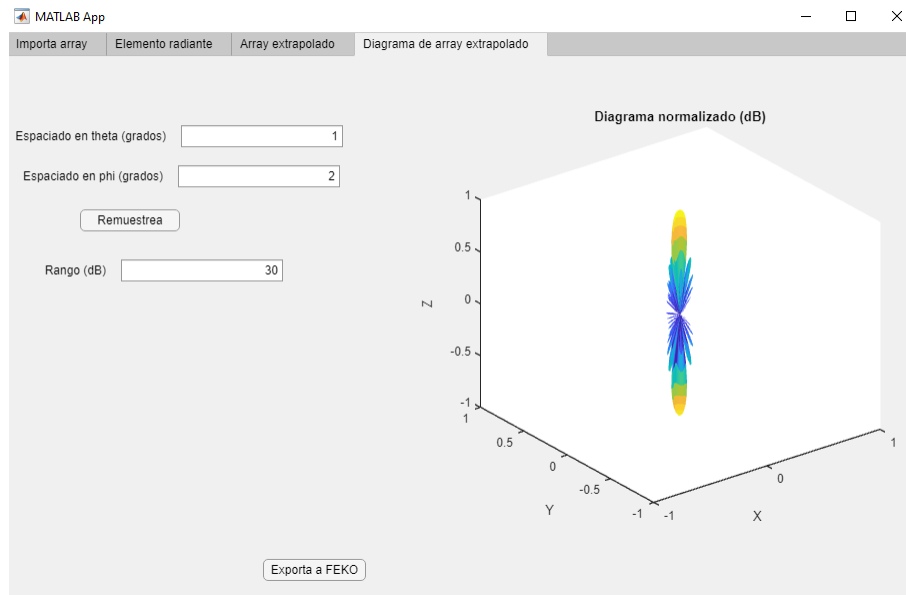


Figura 4.7: Diagrama de radiación de la agrupación objetivo de 16x16 elementos

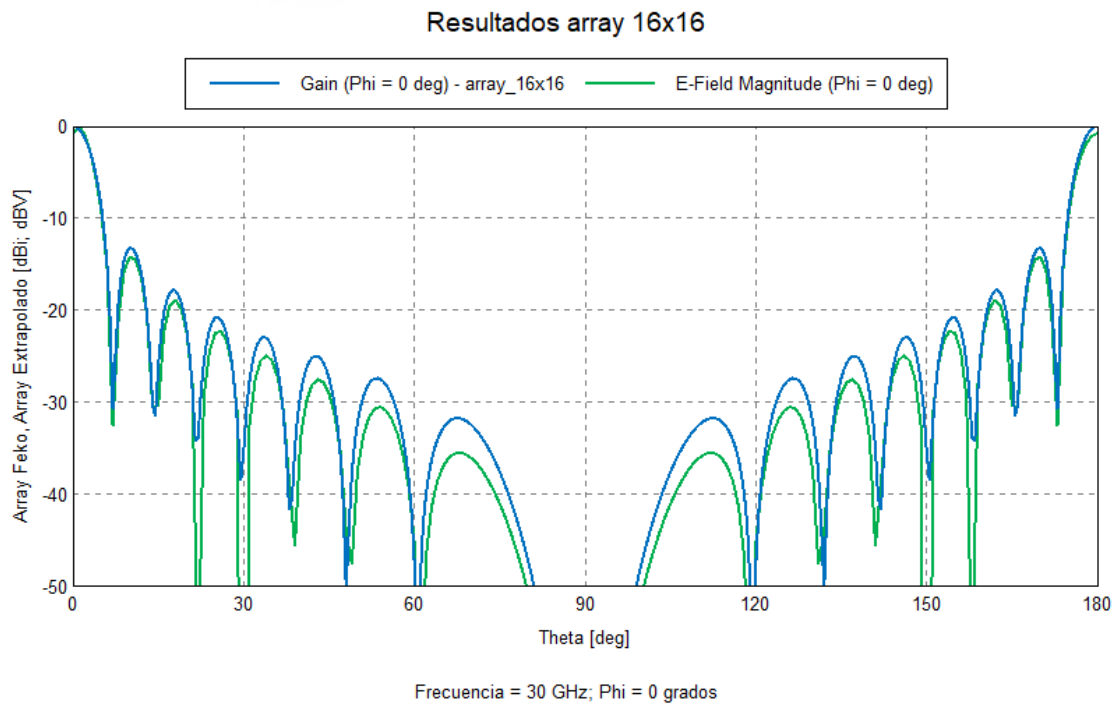
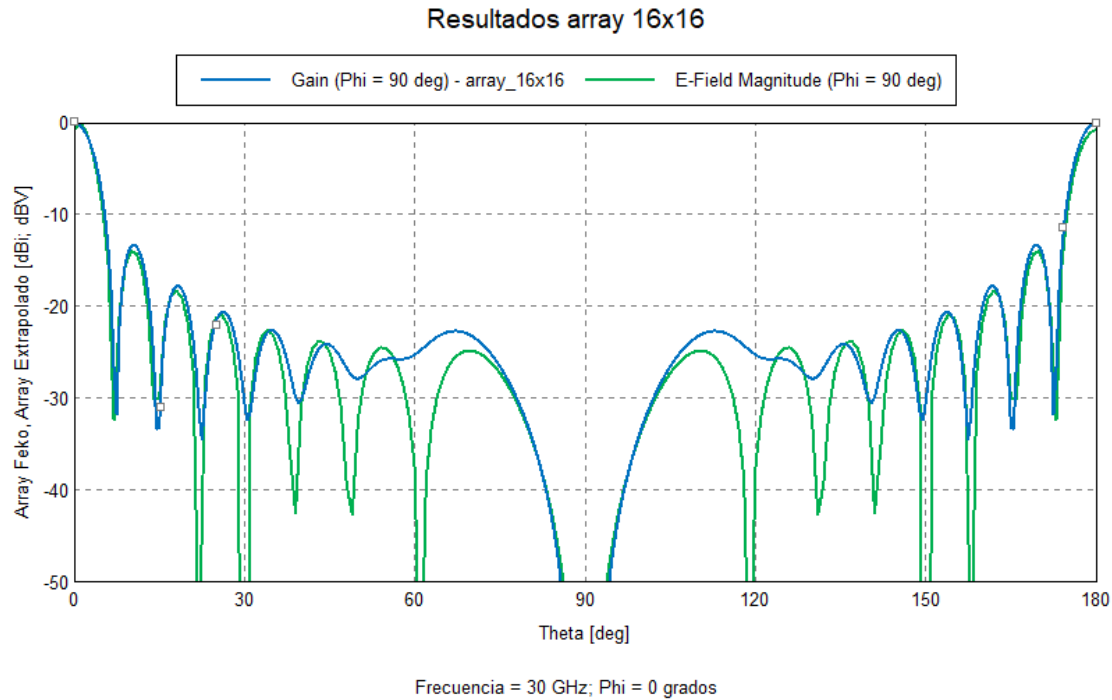


Figura 4.8: Resultado para un corte de  $0^\circ$

(en muy menor medida, en comparación con el corte de  $90^\circ$ ), el fenómeno de los acoplos mutuos.

Como ya se hizo en el caso anterior, debemos observar, a su vez, como se comportan los resultados de los diagramas de radiación representados en el corte correspondiente al eje de  $90^\circ$  (4.9).

Y, una vez más, vuelve a cumplirse lo comentado anteriormente, para una agrupación de 8x8



**Figura 4.9: Resultado para un corte de 90°**

elementos. En el caso del corte en 90°, al encontrarnos en la posición de mayor influencia de los acoplos mutuos, observamos como la respuesta, a medida que nos alejamos del lóbulo principal, tiende a cambiar en comparación al resultado obtenido con *Feko*, pero, podemos comprobar como el lóbulo principal y los principales lóbulos secundarios siguen manteniendo una coincidencia prácticamente total, demostrando el buen funcionamiento de nuestro entorno desarrollado en *App-designer*.

Sin embargo, a pesar de haber demostrado el correcto funcionamiento del programa, en casos de agrupaciones objetivo de 8x8 y 16x16 elementos, empleando para ello una matriz de admitancias de un *array* de 4x4 elementos, y utilizando como antena aislada un dipolo; decidimos realizar una prueba más con un elemento radiante más complejo que un simple dipolo, para comprobar si nos encontrábamos en lo cierto, y podíamos afirmar completamente, que nuestro programa funcionaba dentro de lo esperado.

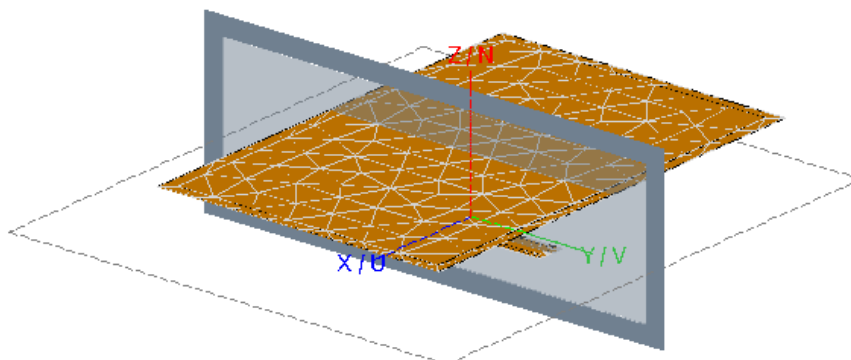
Para poder llevarlo a cabo, se detallará a continuación, el proceso seguido para obtener una agrupación objetivo de 8x8 y 16x16 elementos, a partir del mismo archivo correspondiente a la matriz de admitancias 4x4, pero en este caso, empleando como antena aislada, una antena de tipo parche.

### 4.3. Análisis de resultados: Antena tipo parche

Para empezar, debemos entender porque, de entre todas las opciones de antenas que podríamos elegir, se ha utilizado la antena de tipo parche para comprobar el funcionamiento del programa.

Este tipo de antenas (conocidas también por el nombre de antenas de tipo *microstrip*, por basar

su funcionamiento en dicha tecnología), son uno de los tipos de antena más utilizados dentro del ámbito de las telecomunicaciones, del pasado y actuales, ya que debido a su reducido tamaño, se utilizan para un gran número de aplicaciones dentro de campos como la aeronáutica, para el desarrollo de tecnología satelital, aplicaciones militares y comunicaciones móviles e inalámbricas, entre otras.



**Figura 4.10: Aspecto de una antena de tipo parche generada desde *Feko***

Otra de las características más importantes que explican la gran importancia de este tipo de antenas, es el rango de frecuencias en el que puede llegar a ser empleado, alcanzando frecuencias considerablemente elevadas, dentro de la banda de las microondas e incluso, de las ondas milimétricas.

Por todas las aplicaciones de este tipo de antena, hemos decidido comprobar el funcionamiento, siguiendo el mismo proceso que se siguió con el dipolo.

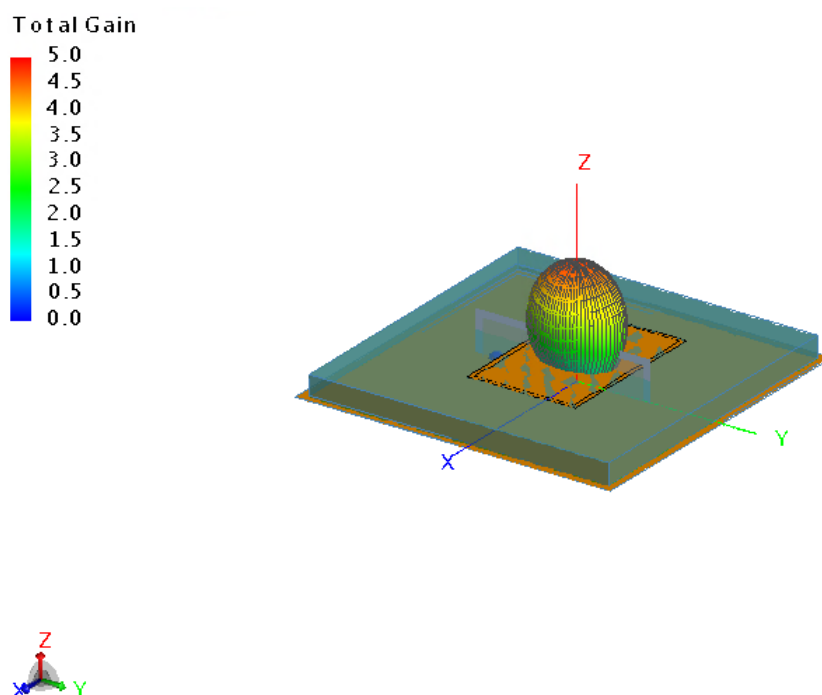
En primer lugar, se obtendrá el diagrama de radiación de la antena de tipo parche, para después, generar la matriz de admitancias correspondiente a una agrupación de 4x4 elementos utilizando este tipo de antena, finalizando el proceso, demostrando su funcionamiento, comprobando los resultados obtenidos con nuestro programa, y los obtenidos automáticamente con la herramienta generadora de agrupaciones de *Feko*.

En este caso, el diagrama de radiación estará formado por dos cortes distintos, dos correspondientes al campo eléctrico de la antena (y que representan el eje YZ) y otros dos que hacen referencia al campo magnético (representando el eje XZ). En todo caso, el diagrama de radiación completo, de nuestra antena de tipo parche tendrá el aspecto de la figura 4.11.

Donde puede observarse el lóbulo principal de la representación del diagrama de radiación que, posteriormente nos ayudará a obtener el *array* extrapolado.

Pero, para ello, es necesario obtener otro de los elementos principales de nuestro proyecto, la matriz de admitancias. Como ya se realizó anteriormente, la matriz de admitancias será la correspondiente a una agrupación de 4x4 elementos utilizando este tipo de antena.

Siguiendo los mismos pasos que se siguieron en el capítulo 3, en el apartado referente a la matriz de admitancias, podemos obtener esta matriz para poder importarla en nuestro programa de *Matlab* y así continuar con el proceso.



**Figura 4.11: Diagrama de radiación de la antena de tipo parche**

Es el momento de comprobar el funcionamiento de nuestro entorno desarrollado con *Appdesigner*, en este caso, utilizando como elemento radiante, una antena de tipo parche.

Como es costumbre, comenzamos el proceso importando el archivo correspondiente a la matriz de admitancias de una agrupación de 4x4 elementos, utilizando una antena de tipo parche para generarlo. El resto de datos, quedan por defecto y podremos observar una previsualización de la matriz, así como el módulo y la fase de las tensiones efectivas, en la figura 4.12 (exactamente igual a lo realizado hasta ahora).

La diferencia con los procesos anteriores la encontramos en esta segunda pestaña que vamos a comentar (4.13), donde, en vez de importar como archivo correspondiente a la antena aislada un dipolo, importaremos el archivo correspondiente al diagrama de radiación de la antena de tipo parche, y que puede visualizarse en la figura 4.11.

Continuamos el proceso, pasando a la tercera pestaña de nuestro programa, en la cual, seremos capaces de obtener la agrupación extrapolada, donde, en este caso, y debido a la cantidad de recursos computacionales que las agrupaciones que emplean este tipo de antenas, supone, vamos a analizar únicamente el caso correspondiente a una agrupación objetivo de 8x8 elementos, teniendo la representación de esta ventana en la figura 4.14.

Una vez visualizadas tanto la matriz mapeada, como el módulo y la fase de las amplitudes de los pesos de las corrientes, es el momento de observar la representación final del diagrama de radiación de una agrupación objetivo de 8x8 elementos, empleando una matriz de admitancias correspondiente a una agrupación de 4x4 elementos y empleando como antena aislada, una antena de tipo parche (4.15)

Por último, debemos comparar los resultados obtenidos mediante nuestro programa, y super-

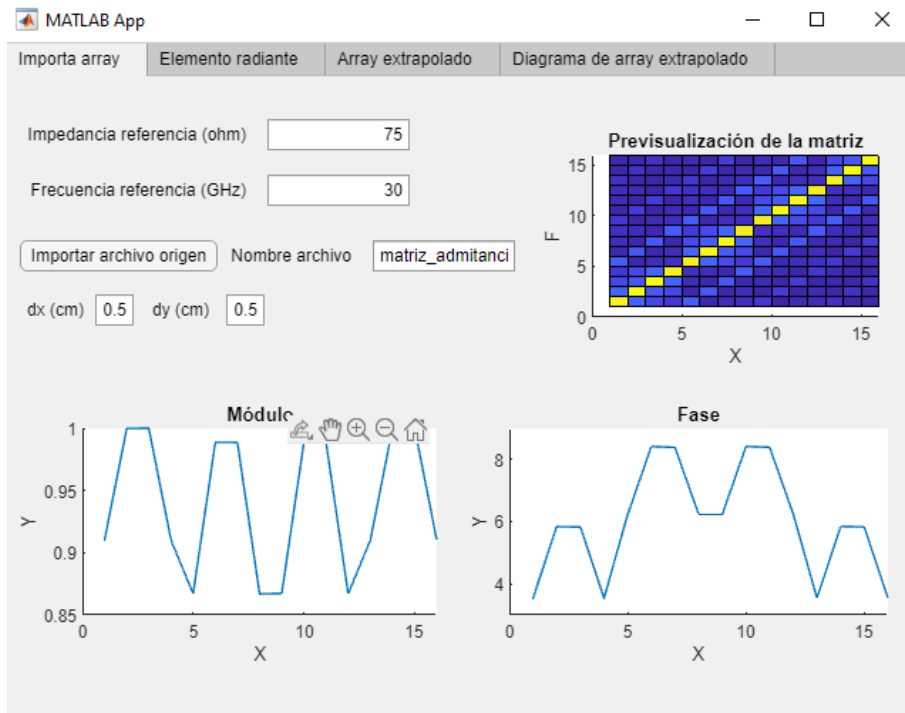


Figura 4.12: Primera pestaña del programa para una antena tipo Parche

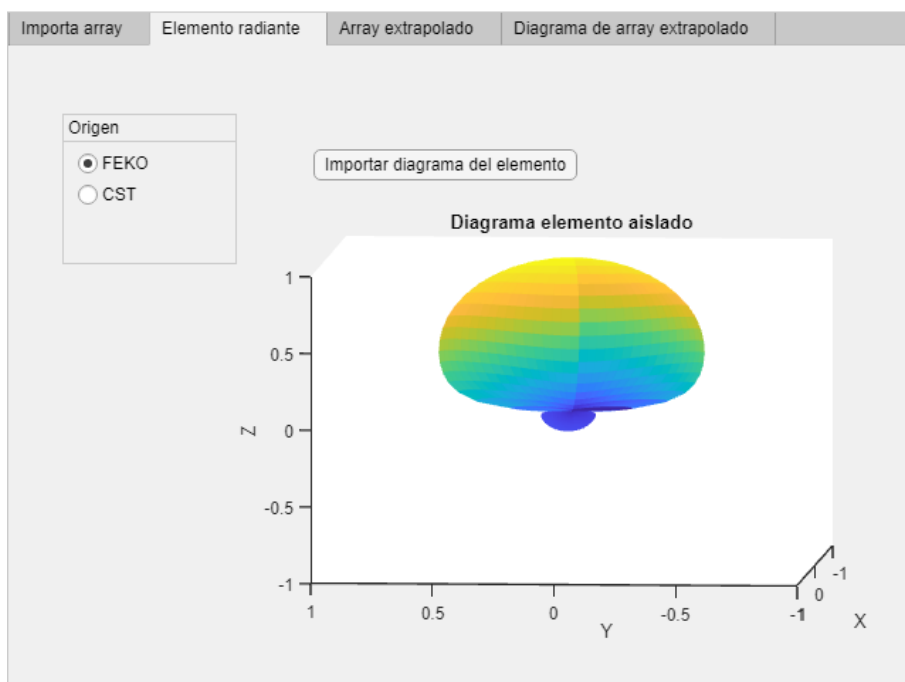
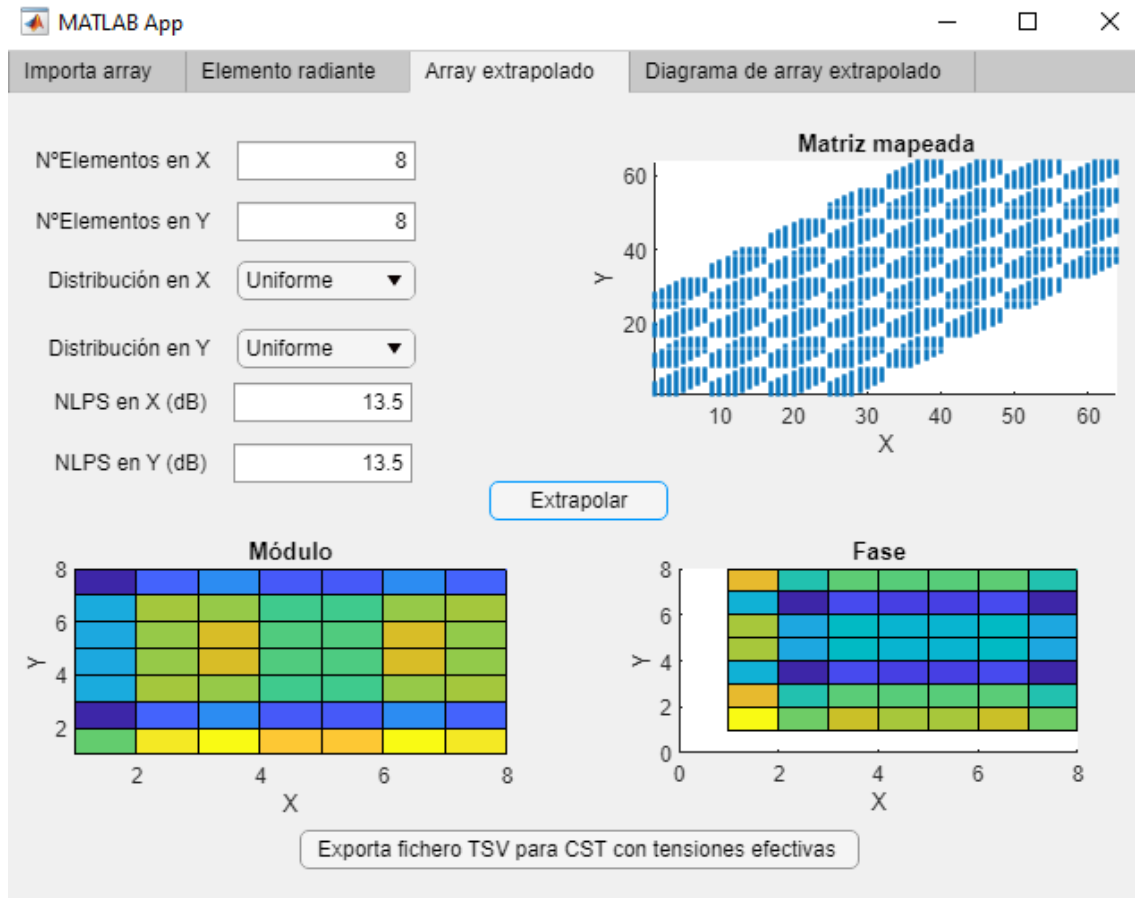


Figura 4.13: Diagrama de radiación de la antena de tipo parche en la segunda pestaña del programa

ponerlos a los resultados que *Feko*, obtiene automáticamente mediante su herramienta generadora de agrupaciones, con el resultado visible en la figura 4.16.



**Figura 4.14: Array extrapolado 8x8 utilizando una antena tipo parche**

Si observamos los resultados del corte de la imagen anterior, correspondiente a un corte con  $\phi$  en  $0^\circ$ , observamos como, al igual que ocurría con los dipolos, la respuesta entre ambos resultados es prácticamente idéntica.

Es de gran importancia entender el mérito de este resultado tan similar, ya que, al contrario de los casos correspondientes a los dipolos como antena aislada, al utilizar antenas de tipo parche, la dificultad aumenta de forma considerable, algo que se ha podido comprobar a la hora de compilar los resultados de la agrupación con *Feko*, donde se ha hecho uso de un gran coste computacional.

Observamos la aparición del lóbulo principal (en ambos extremos de las imágenes), con una coincidencia casi perfecta, difiriendo ligeramente a medida que nos vamos acercando a los lóbulos secundarios.

Por otra parte comentar que el resultado no es tan exacto como el obtenido con los dipolos, debido al comportamiento no ideal de estas antenas, que simulan un caso real y debido a la gran cantidad de incógnitas que debía calcular el programa

Para finalizar, revisaremos esta comparación con un corte correspondiente a los  $90^\circ$  en  $\phi$ , donde podremos observar si en este caso la respuesta es tan aproximada como la anterior. El resultado lo podemos apreciar en la figura 4.17

Podemos comprobar de nuevo, la aparición de un lóbulo principal perfectamente coincidente,

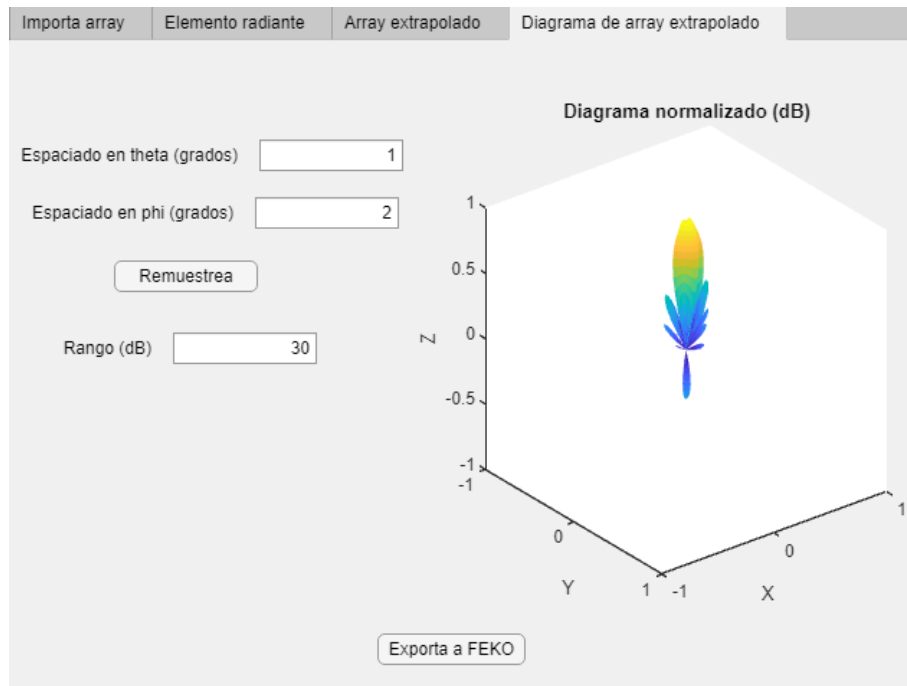


Figura 4.15: Diagrama de radiación final de la agrupación objetivo

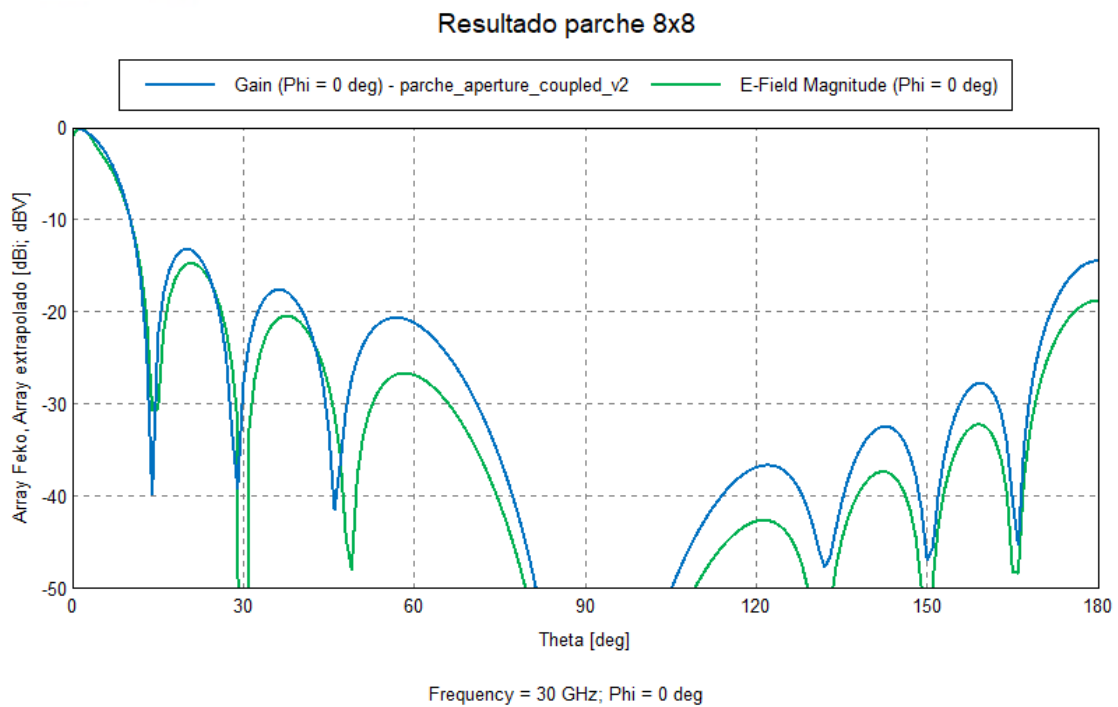
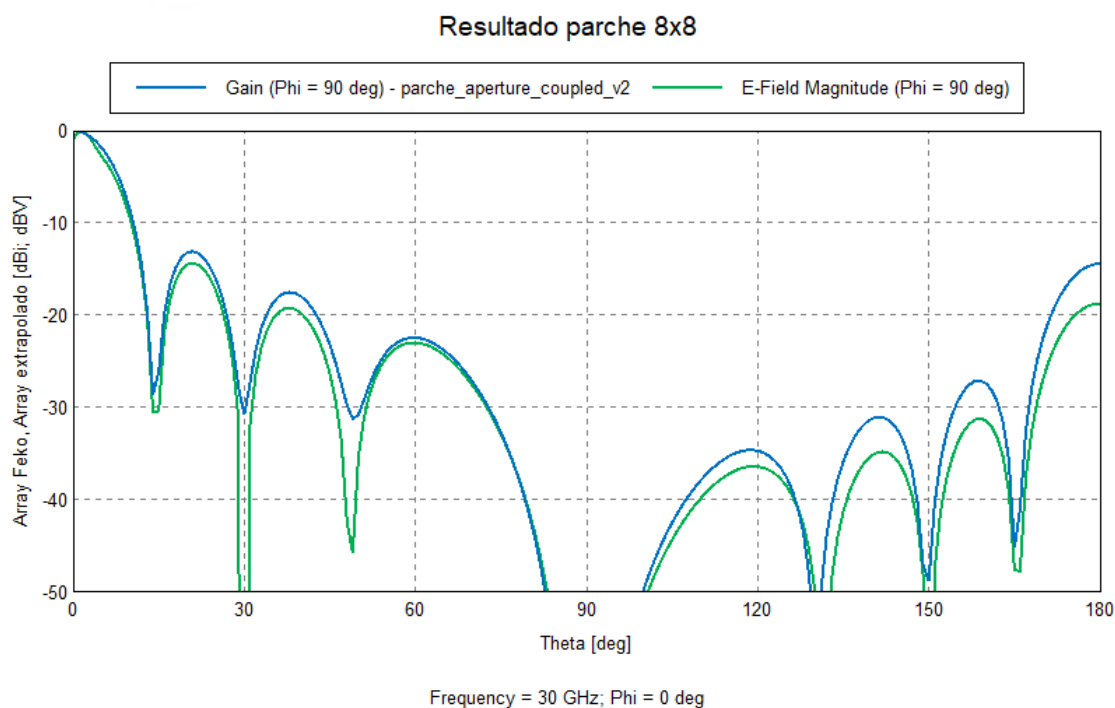


Figura 4.16: Comparación de resultados antena parche 0°

con leves diferencias a medida que nos alejamos de este, pero con una respuesta sorprendentemente aproximada.



**Figura 4.17: Comparación de resultados antena parche 90°**

Con todo lo comentado anteriormente, queda demostrado que el proceso de obtención del diagrama de radiación y de las características de la agrupación objetivo a partir de la matriz de admitancias y del diagrama de radiación de una antena aislada, puede realizarse con cualquier tipo de antena, como se ha demostrado utilizando un dipolo y una antena de tipo parche. En el caso del parche no hemos hecho la prueba para un *array* 16x16 ya que el coste computacional del procedimiento directo con *Feko* excedía las prestaciones del ordenador empleado. Para entender la importancia del desarrollo de este programa, si comparamos el coste computacional entre el resultado obtenido con nuestro programa y el obtenido con *Feko*, para el caso de los dipolos y teniendo en cuenta la agrupación de 8x8 elementos (ya que esta es la común en ambos tipos de antenas), nuestro programa obtiene los resultados prácticamente al acto, mientras que, en el caso de *Feko*, los resultados tardan en obtenerse, un total de 0.001 horas (según especifica el archivo del programa), con un coste computacional de 15.429 MByte.

Con este resultado, no podemos determinar que nuestro programa resulte una ventaja tan grande con respecto a *Feko*, pero la diferencia se aprecia en el caso de la antena de tipo parche, ya que, considerando el mismo caso (agrupación de 8x8 elementos), nuestro programa vuelve a obtener los resultados prácticamente al acto, mientras que en *Feko*, se requieren de 0.923 horas (prácticamente una hora de simulación) y un coste computacional de 6.939 GByte, mostrando de este modo, la ventaja que supone el uso del programa desarrollado en *Appdesigner*.

Con todo ello, finaliza este capítulo de la memoria, cerrando de este modo, los principales análisis de resultados, así como los procedimientos para obtenerlos. En el próximo capítulo, se detallarán las principales conclusiones y se comentarán tanto las modificaciones que podrían hacer que el proyecto mejorase, como las líneas futuras que, tanto la idea de este proyecto, como su desarrollo, podrían aparecer próximamente.



## Capítulo 5

# Conclusiones

Llegamos en este último capítulo de la memoria, a las conclusiones fundamentales que se han podido obtener después de haber realizado las diferentes explicaciones de los cálculos y procedimientos teóricos, después de haber realizado el posterior desarrollo de un entorno de simulación que nos permitiese automatizar dichos cálculos, y después de haber obtenido y analizado los diferentes resultados obtenidos por ese entorno.

Además de las conclusiones que comentaremos a lo largo de este capítulo, también se detallarán diferentes modificaciones que podrían realizarse en este proyecto, así como las líneas futuras relacionadas con este y con el objetivo principal que se buscaba en un principio.

La motivación principal de este proyecto era solucionar el problema que se presenta cuando tratamos de analizar agrupaciones de antenas que poseen un gran número de elementos, debido al coste computacional que eso conlleva.

Nuestra idea, era solucionar este problema para que, el número de elementos de un *array*, no fuese el problema, y que las diferentes características relacionadas con este, como puede ser, el diagrama de radiación, fueran accesibles sin necesidad de utilizar una computadora con grandes recursos.

Si nos fijamos en este objetivo principal, tal y como se ha demostrado en el Capítulo 4: Resultados, hemos satisfecho esa necesidad. A partir de una matriz de admitancias dato (cuyo proceso de obtención también ha sido comentado en la memoria), y del diagrama de radiación de una antena aislada (con su proceso de obtención también explicado), hemos conseguido desarrollar un programa que nos ha proporcionado esa agrupación objetivo, extrapolada, que, como se ha demostrado posteriormente, es prácticamente coincidente con la generada automáticamente con un entorno de simulación comercial como es *Feko*.

Pero este no ha sido el único objetivo alcanzado a lo largo del proyecto, ya que, hay ciertos aspectos que también se han conseguido abordar de manera clara y concisa, entre los cuales se encuentran:

- *Appdesigner*: Gracias a este proyecto, hemos podido conocer en profundidad una de las herramientas más útiles de las que dispone el software *Matlab*, pero no solo hemos conocido como funciona su interfaz para crear GUIs (entornos de desarrollo), con gráficas, botones, desplegables... Sino que, también hemos podido entender de que es capaz el software, al

haber permitido el desarrollo de un entorno que ha obtenido resultados tan aproximados, de una manera intuitiva y amena para el usuario.

- Entender los principales principios teóricos de las agrupaciones de antenas: Gracias al segundo capítulo de esta memoria, otro de los aspectos que se ha podido abordar, es la explicación de los diferentes procesos y conceptos teóricos, necesarios para poder entender las características fundamentales de las agrupaciones de antenas, permitiendo, a un lector sin formación en la materia, entender mediante fórmulas e imágenes, la importancia de los diferentes procedimientos teóricos utilizados.
- Fundamentar las principales conclusiones con varios casos específicos: Ya que, para poder estar seguros de que el programa funciona perfectamente, se han supuesto varios casos dentro de las pruebas realizadas. Se ha aumentado el número de elementos, se ha utilizado el programa con dos tipos distintos de antenas aisladas...
- Coste computacional: Ya que, se ha demostrado, como el uso del programa supone un menor coste computacional dedicado a resolver las distintas simulaciones, como en el caso de la antena de tipo parche.

## 5.1. Aplicaciones

Si bien el programa desarrollado es algo sencillo y primitivo, permite obtener las respuestas esperadas y como hemos podido comprobar, son respuestas muy ajustadas a aquellas que se pueden obtener con un software comercial.

La intención de este programa, podría aplicarse a diferentes campos dentro de las telecomunicaciones, pues, las agrupaciones de antenas, son la principal técnica de desarrollo de antenas inteligentes, que permiten mejorar cualquier sistema de comunicaciones inalámbricas.

Si esos sistemas utilizasen agrupaciones con un número de elementos, cada vez mayor, el funcionamiento de estas antenas inteligentes sería uno de los más potentes jamás vistos, ya que, la principal ventaja que proporcionarían estos *arrays* con un mayor número de elementos en sus ejes, sería la de poseer una mayor directividad y, por tanto, supondría diferentes ventajas como el aumento de la zona de alcance de la señal de las antenas (denominado comúnmente radio de cobertura), permitiría reducir las interferencias y efectos como la propagación multicamino ya que, la potencia de emisión de estas agrupaciones sería bastante más considerable, entre otras muchas ventajas que harían que el uso de agrupaciones de antenas fuera más que justificable.

Uno de los principales sectores en los que encontramos el uso de agrupaciones de antenas, es en los sistemas radar y las comunicaciones por satélite. La importancia que podrían llegar a tener estas agrupaciones de antenas que permitan tener un mayor número de elementos a su disposición, podría resultar en mayor precisión de la detección o mejorar el ancho de banda de las comunicaciones.

## 5.2. Mejoras futuras

Entre las posibles líneas futuras de trabajo, es decir, aspectos que podrían añadirse o modificarse en el proyecto para que este sea mucho más completo, encontramos, por ejemplo, que el

programa solo está pensado para funcionar para una frecuencia. Con lo cual, uno de los principales puntos para poder mejorar el programa, sería desarrollarlo para varias frecuencias.

Además, otra posible mejora que podría incluirse en nuestro programa y que ayudaría a apreciar de mejor forma las diferencias con respecto a los programas comerciales, sería implementar una mejora a la hora de visualizar los resultados, ya que, si nos fijamos en algunas de las tablas obtenidas con el programa desarrollado, la información no se aprecia como gustaría, como en el caso del módulo y la fase de los pesos de las amplitudes de las corrientes en la tercera ventana.



# Bibliografía

- [1] Ahmed A.Kishk. “Prediction of large array characteristics from small array parameters”. En: *Department of Electrical Engineering and Center for Applied Electromagnetic Systems Research (CAESR), University of Mississippi, USA* 8 (2021), págs. 84-87. DOI: 10.1109/LAWP.2008.2011147.
- [2] Abdelmoniem T. Hassan y Ahmed A. Kishk. “Efficient Procedure to Design Large Finite Array and Its Feeding Network With Examples of ME-Dipole Array and Microstrip Ridge Gap Waveguide Feed”. En: *IEEE TRANSACTIONS ON ANTENNAS AND PROPAGATION* 68.6 (2020), págs. 4560-4570.
- [3] Alejandro Valero Nogueira, Héctor Esteban González y Miguel Ferrando Bataller. *Fundamentos de teoría de antenas*, pág. 121.
- [4] MATHWORKS. *User Guide Appdesigner*. Natick, Massachusetts: The MathWorks Inc., 2021.