

## CAPÍTULO 4. DESARROLLO DEL SOFTWARE

La primera parte consistió en la búsqueda de textos relacionados con el tema y entender la forma de propagación de las Ondas Planas Uniformes, con sus distintas variantes, para poder graficarlas. Ya que se tenía esto, se comenzó por identificar cuales eran las ecuaciones que se iban a utilizar para cada caso. Se utilizaron las ecuaciones en el dominio del tiempo, ya que desde un principio de la tesis éste era el principal objetivo. Ahora se necesitaba saber el alcance del proyecto, y éste es el siguiente, se van a graficar para los medios sin pérdidas las distintas polarizaciones: transversal eléctrico y transversal magnético, lineal con pendiente de  $45^\circ$  y con pendiente distinta de  $45^\circ$ , circular de mano izquierda y derecha, elíptica de mano derecha e izquierda. Así como también se graficará la incidencia normal, en medios con pérdidas y en medios sin pérdidas, por último, las ecuaciones en el dominio del tiempo también para la incidencia oblicua, en medios sin pérdidas y conductores perfectos, en sus dos polarizaciones paralela y perpendicular. Ya que el alcance del proyecto hace que el software sea demasiado grande este se dividió en 3 partes, el primer programa sólo es para graficar las polarizaciones en los medios sin pérdidas, el segundo es sólo para graficar la incidencia normal en medios con pérdidas y sin pérdidas, y por último el

tercer programa es para graficar la incidencia oblicua en medios sin pérdidas y conductores perfectos, en sus dos polarizaciones paralela y perpendicular.

#### **4.1 Selección del Software.**

Se tenían varias alternativas para hacer el diseño del software, pero después de hacer una búsqueda y un análisis de todas las herramientas que se podrían utilizar, se llegó a la conclusión de que matlab es una herramienta muy completa, ya que no sólo podemos hacer cálculos matemáticos, también combina una herramienta muy poderosa para graficar. Matlab también nos da muchas posibilidades de visualización y movimientos de gráficas, gracias a su barra de herramientas y por esto se utilizó el programa Matlab 6.5 release 13.

#### **4.2 Selección de las variables de entrada.**

Se hizo una selección de las variables de entrada, éstas se iban a necesitar para poder hacer las gráficas, pero también se requería que no fueran demasiadas variables de entrada ya que se buscaba un programa fácil de utilizar y conveniente para el usuario.

Ahora hablaremos del programa que grafica las distintas polarizaciones, en éste se necesitaban las siguientes variables que se muestran en la tabla 4.1, en el lado izquierdo aparece qué significa cada una de las variables y en el lado derecho el nombre que se les asignó dentro del programa de matlab.

Datos de Entrada	Asignación de Variables
Amplitud de campo eléctrico	Em
Amplitud de campo eléctrico en y	Emy
Frecuencia	Fr
Distancia	D
Permitividad del medio 1	er1
Permeabilidad del medio 1	ur1
Conductividad del medio 1	con1

Tabla 4.1. Asignación de variables para el programa de polarizaciones.

Ahora vamos con el segundo programa y las variables de entrada seleccionadas para éste fueron las siguientes:

Datos de Entrada	Asignación de Variables
Amplitud de campo eléctrico	Em
Frecuencia	Fr
Distancia del medio 1	D1
Distancia del medio 2	D
Permitividad del medio 1	er1
Permeabilidad del medio 1	ur1
Conductividad del medio 1	con1
Permitividad del medio 2	er2
Permeabilidad del medio 2	ur2
Conductividad del medio 2	con2

Tabla 4.2. Asignación de variables para el programa de incidencia normal.

Por último para el programa de incidencia oblicua se utilizaron las siguientes variables:

Datos de Entrada	Asignación de Variables
Amplitud de campo eléctrico	Em
Frecuencia	Fr
Distancia del medio 1	D1
Distancia del medio 2	D
Ángulo de Incidencia	Aincd
Permitividad del medio 1	er1
Permeabilidad del medio 1	ur1
Conductividad del medio 1	con1
Permitividad del medio 2	er2
Permeabilidad del medio 2	ur2
Conductividad del medio 2	con2

Tabla 4.3. Asignación de variables para el programa de incidencia oblicua.

#### 4.3 Datos de salida.

También se hizo una selección de los datos de salida. Estos son los cálculos que se realizan automáticamente en matlab.

Datos de Salida	Asignación de Variables
Constante de fase	Bet
Constante de atenuación	Alf
Impedancia intrínseca	Eta
Ángulo de la impedancia intrínseca	Teeta
Velocidad de fase	V
Longitud de onda	Lam
Densidad de potencia promedio	Sav

Tabla 4.4. Datos de salida para el programa de polarizaciones.

Ahora vamos con el segundo programa y sus datos de salida, o datos que serán calculados.

<b>Datos de Salida</b>	<b>Asignación de Variables</b>
Constante de fase del medio 1	bet1
Constante de atenuación del medio 1	alf1
Constante de fase del medio 2	bet2
Constante de atenuación del medio 2	alf2
Impedancia intrínseca del medio 1	eta1
Ángulo de impedancia intrínseca del medio 1	eta1an
Impedancia intrínseca del medio 2	eta2
Ángulo de impedancia intrínseca del medio 2	eta2an
Coefficiente de reflexión	Crefle
Ángulo del coeficiente de reflexión	Creflean
Coefficiente de transmisión	Ctrans
Ángulo del coeficiente de transmisión	Ctransan
Densidad de potencia promedio incidente	Densiin
Densidad de potencia promedio transmitida	Densitr

Tabla 4.5. Datos de salida para los programas de incidencia normal y oblicua.

Ya que son los mismos datos de salida, en los dos programas, en la tabla 4.2 se ven las variables asignadas utilizadas para los dos. Obviamente los cálculos que se hacen en los programas son distintos, ya que las ecuaciones para hacer los cálculos en la incidencia normal son totalmente diferentes, a las de incidencia oblicua.

#### 4.4 Desarrollo de las gráficas.

Después de tener las variables de entrada que se iban a utilizar, se buscó la manera de hacer las gráficas, para esto se necesitaba saber que era exactamente lo que se quería ver, entonces se llegó a la conclusión con la ayuda de mi asesor de tesis y mis sinodales, de que se requería ver las ondas propagándose con movimiento y también con vectores.

Primero se comenzó graficando la forma de onda más sencilla, esta es la polarización transversal eléctrica y fue la primera que se graficó, el campo eléctrico en el eje  $x$  y el campo magnético en el eje  $y$ , y estos propagándose en  $z$ .

Aquí en matlab existe una orientación dada por el mismo programa, de los ejes, como se muestra en la figura 4.1:

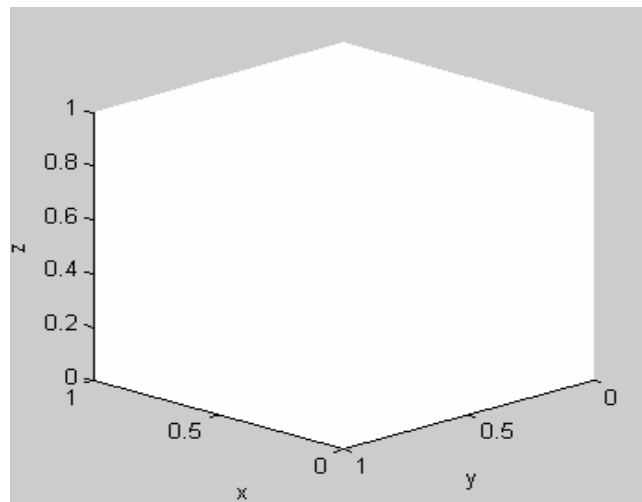


Figura 4.1. Orientación de los ejes determinada por matlab.

Se muestra un área para graficar vacía, para mostrar como tiene predeterminado sus ejes matlab. Pero aquí se necesitaba que la onda se propagara en el eje x para matlab, que en realidad es el eje z para nosotros, así que se hizo un cambio de ejes, entonces en el programa el eje x es el eje z y viceversa.

El comando que se utilizó para graficar en matlab fue el “plot3”, éste te permite hacer gráficas en “3-d”, de una manera sencilla.

Cada una de las gráficas fue hecha por separado, esto quiere decir que dentro de matlab se hicieron varios programas, en los cuales se hacían las distintas gráficas.

Al principio se intentó agregarle el tiempo de evaluación a las variables de entrada, pero esto resultó ser demasiado impráctico, ya que al momento de hacer la gráfica, a veces con el tiempo que se le capturaba y la frecuencia hacían que las gráficas se movieran de una manera muy lenta, y otras veces demasiado rápido como para poder apreciar bien el movimiento de las mismas, así que por eso se decidió quitar esa variable.

Para darle el movimiento a las gráficas se hace un incremento en una variable conocida como t, que es el tiempo y éste no se pide ya que lo que se requiere en el programa es que el movimiento se aprecie de una manera constante y no de una manera aleatoria, y esto hizo mas fácil el manejo del programa, por eso esta parte se define dentro del programa como una constante.

#### 4.5 Visualización por medio de la ventana de comandos de matlab.

Al principio los datos de entrada se capturaban como se muestra en la figura 4.2:

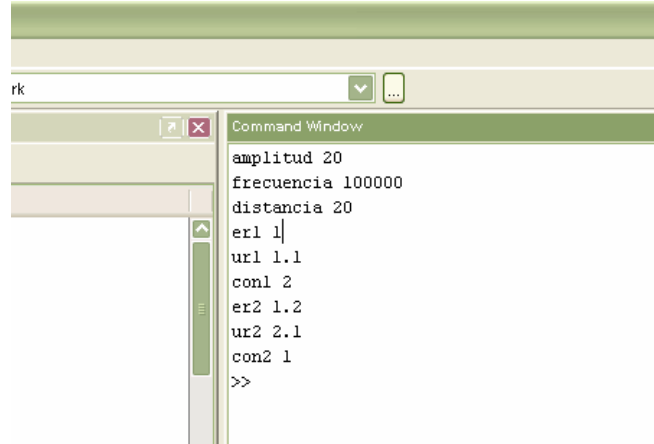


Figura 4.2. Entrada de datos desde la ventana de comandos de matlab.

Así era la entrada de datos, para todos los programas que se realizaron en matlab, esto es demasiado complejo ya que para cada gráfica que se hiciera, se necesitaría volver a capturar los mismos datos, una y otra vez.

Y la salida era de la misma manera por medio de la ventana de comandos del matlab.

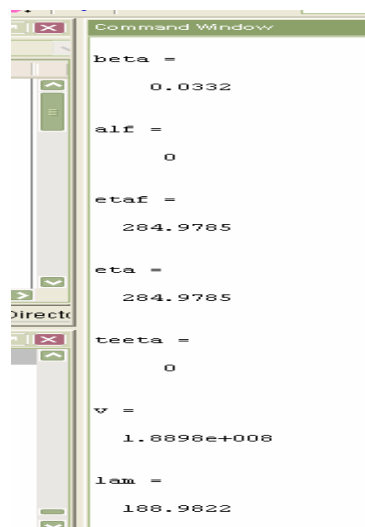


Figura 4.3. Datos de salida en la ventana de comandos de matlab.



Como se muestra en la figura 4.3, era en un principio la salida de datos del programa, esto es demasiado impráctico, para el usuario, ya que tiene que moverse dentro de la ventana de comandos de matlab, para poder ver los datos de salida, e incluso necesitaría saber el nombre de las variables que se utilizaron dentro de el programa para ver qué significa cada una de las salidas, en algunos casos.

#### 4.6 Creando la interfaz grafica del usuario.

Anteriormente se mostró cómo se manejaba el programa, era de una manera un poco rústica e incluso tediosa para el usuario, por lo que se buscó la manera de mejorar el programa, éste ya hacia las gráficas, pero la captura y salida de datos era demasiado larga, para el usuario.

Así es que se recurrió a una herramienta de matlab conocida como la “GUI”, que significa interfaz grafica de usuario, ésta hace que el usuario, por medio de una ventana pueda hacer la captura de datos y también se pueda ver la salida de los datos en la misma ventana.

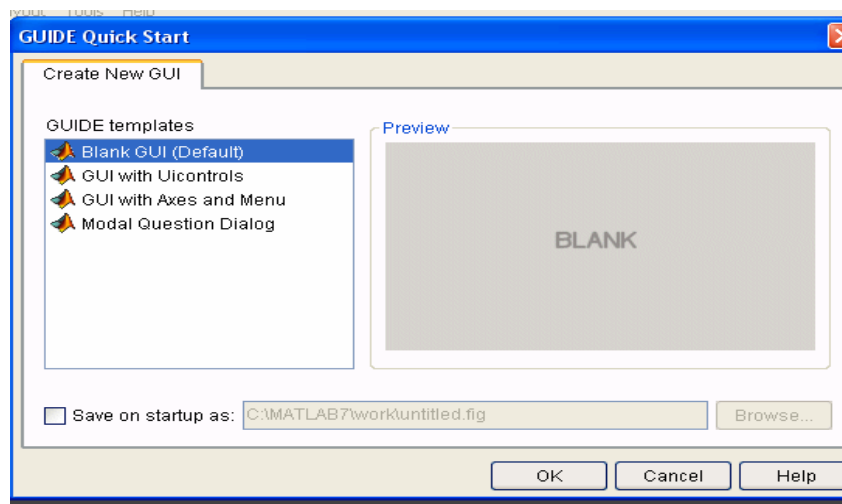


Figura 4.4. Se abre la ventana para crear una nueva interfaz gráfica de usuario.

Se entró a una “blank GUI” como se puede apreciar en la figura 4.4, para poder hacer una nueva interfaz y como ya se contaba con el nombre de las variables, tanto de entrada, como las de salida se podía hacer la interfaz.

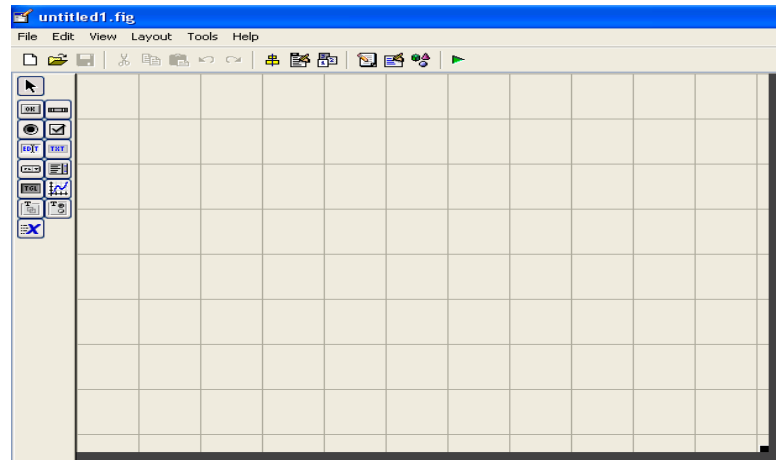


Figura 4.5. Área de trabajo para crear la interfaz gráfica de usuario.

Se comenzó por colocar todas las variables como se muestra en la figura 4.6, que se fueron agregando las etiquetas en los campos, para poder identificar, los distintos campos y su relación con las variables.

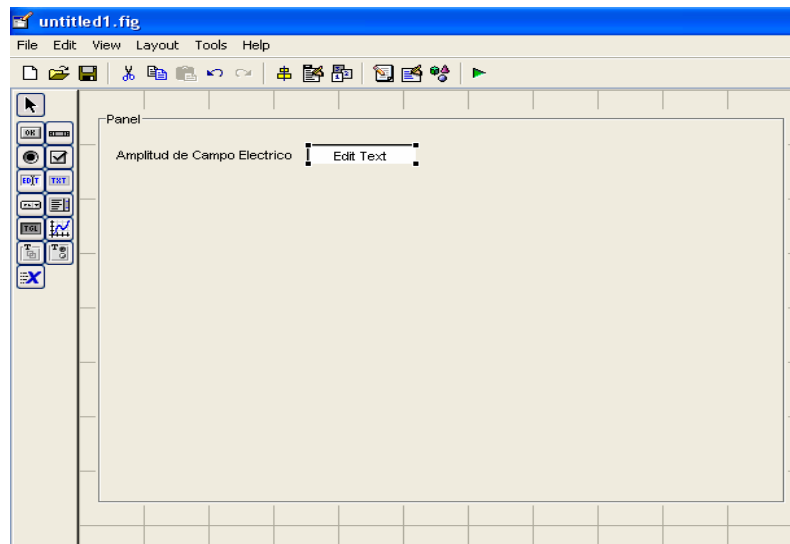


Figura 4.6. Diseñando la interfaz gráfica de usuario y acomodando los campos.

En la figura 4.6 se puede ver como se tienen que poner uno a uno los campos y su respectivo texto.

En la figura 4.7 se muestra la ventana terminada de uno de los tres programas que se hicieron, se siguieron los mismos pasos para crear los tres programas.



Figura 4.7. Ventana terminada del programa polarización.

Los campos de datos de salida están bloqueados y esto se hizo con la finalidad de que no se puedan alterar los valores de salida y sólo se muestren los que fueron calculados dentro de matlab.

Proceso de la elaboración de la ventana: Se tenía que planear como se quería la interfaz, y también se tuvo que escoger el color, de las letras, los campos, los botones, e incluso de las letras del menú emergente, todo esto se hace por medio de una selección de colores, dentro de las propiedad que se le pueden modificar a las “GUIs”.

Para pasar de un campo a otro, se aprieta la tecla de enter en la computadora, o la otra opción que se tiene es por medio del mouse, se le da click donde se quiera que poner el siguiente dato.

Otra característica general de todos los programas es que cuentan con una protección en caso de que se les quiera escribir letras en lugar de números.



Figura 4.8. Ventana de error en caso de querer introducir letras.

En la figura 4.8 se puede ver la ventana de error y ésta sale justo después de que se intentó introducir letras en lugar de números, en este caso se hizo este ejemplo en el campo de amplitud de campo eléctrico, este mensaje sale justo después de presionar la tecla de enter, o cuando se pasa al siguiente campo, y hace el borrado automático de las letras y como se puede ver limpia de nuevo este campo, ya que en este caso lo vuelve a poner en cero. Se hizo esta validación en los 3 programas..



Figura 4.9. Botones de reinicio y calculo.

En la figura 4.9 se muestran dos botones, el primero hace que se borren los valores que se tienen en los campos y pone todos los campos en cero.

El segundo botón que dice calcular, genera los cálculos internamente para cada uno de los programas y nos despliega en los datos de salida los valores obtenidos, tomando en cuenta los datos que se le dieron de entrada, para los distintos programas.

En la figura 4.10 se puede ver el programa de incidencia oblicua, una de las condiciones que se deben de cumplir para que este programa funcione de manera correcta es que la permeabilidad del medio 1 y la permitividad del medio 1, deben ser mayores que las del medio 2 respectivamente, y se necesita respetar esta condición para que el cálculo del ángulo de transmisión nos de un número real.



Figura 4.10. Ventana de error en caso de que la permitividad del medio 2 sea mayor que la permitividad del medio 1.

Lo mismo sucede en caso de querer que la permeabilidad 2 sea mayor que la 1, en el caso del ejemplo se intentó que la permitividad 1 tuviera un valor de 2 y que la permitividad 2 se le tuviera un valor de 3, después de dar enter, como se ve en la figura 4.10, el campo de la permitividad 2 se vuelve 0 y despliega el mensaje de error.



Figura 4.11. Ventana de error para el ángulo de incidencia.

Esta ventana se hizo sólo para el programa de incidencia oblicua, y saldrá en caso de querer poner un ángulo de incidencia menor que 0 o mayor igual que 90 grados, estos casos, no se deben dar ya que no se encuentran considerados dentro de esta tesis. Cuando exista un error se debe dar click a la ventana de error para que se cierre.

Después de hacer la interfaz gráfica para cada uno de los programas, lo que se hizo fue relacionar los programas y juntarlos todos dentro de uno solo por medio de un “case”, (función de matlab), esto gracias a la herramienta de menú emergente.

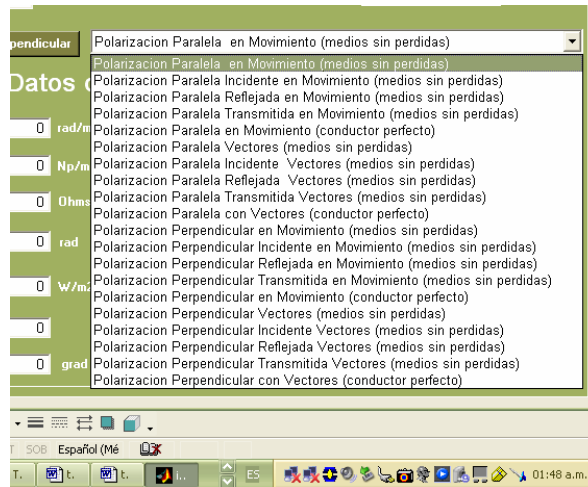


Figura 4.12. Menú emergente del programa de incidencia oblicua.

Cada uno de los elementos que se muestran en el menú emergente, están relacionados con los programas que se hicieron para cada una de las distintas gráficas.

Las ventanas terminadas para cada uno de los programas:



Figura 4.13. Ventana del programa de polarización.



Figura 4.14. Ventana del programa de incidencia normal.

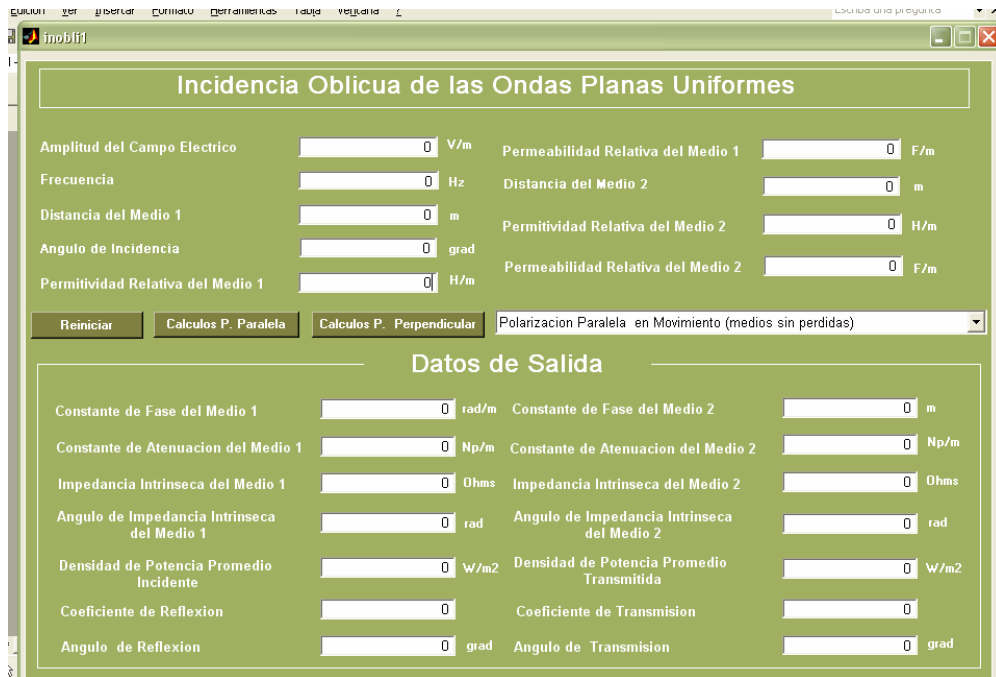


Figura 4.15. Ventana del programa de incidencia oblicua.