Simulación en Java para el campo de Poynting de un circuito circular

Rubén Sánchez-Sánchez
Centro de Investigación en Ciencia Aplicada y Tecnología Avanzada,
Unidad Legaria, Instituto Politécnico Nacional.
Legaria 694. Col. Irrigación. Del. Miguel Hidalgo. CP 11500, México, D.F.

E-mail: rsanchezs@ipn.mx

(Recibido el 13 de Febrero de 2013; aceptado el 26 de Mayo de 2013)

Resumen

Las simulaciones de fenómenos físicos por computador ocupan hoy en día, un buen lugar dentro de las técnicas empleadas para apoyar a la enseñanza tradicional de la Física. En este documento, describo la manera de realizar una simulación en Java para el campo de Poynting alrededor de un circuito circular. A través de un análisis matemático podemos hallar el campo vectorial para el vector de Poynting. Una vez obtenido, podemos elaborar una forma gráfica de representarlo con una simulación para su fácil asimilación en el Nivel Superior de enseñanza. La ventaja de elaborar una simulación, es que el estudiante puede aprovechar la comodidad brindada por el computador para visualizar y entender el campo de Poynting de un circuito eléctrico circular sin necesidad de recurrir a un experimento real de laboratorio tradicional.

Palabras clave: Simulaciones por computadora, circuitos eléctricos, vector de Poynting.

Abstract

By now, the computer simulations of physical phenomena are occupying a good place in the techniques used to support traditional teaching of physics. In this chapter, I describe how to perform a simulation in Java for the Poynting field around a circular circuit. Through a mathematical analysis we can find the vector field for the Poynting vector. Once obtained, we can develop a graphical way to represent a simulation for easy assimilation in High School level. The great point of developing a simulation is that the student can take advantage of the convenience provided by the computer to visualize and understand the Poynting field of an electric circular circuit, without resorting to a traditional lab real experiment.

Keywords: Computer simulations, electric circuits, Poynting vector.

PACS: 01.40.-d, 01.50.H-, 41.20.-q **ISSN 1870-9095**

I. INTRODUCCIÓN

En este trabajo el autor describe la manera de elaborar un mapa de colores que está diseñado de acuerdo a las intensidades que adquiere el vector de Poynting para un circuito circular de corriente continua.

Este vector de Poynting tiene un interés teórico, en la disposición del flujo de energía para un circuito con esta geometría. La idea es hacer más amena una clase de Física al mostrarle al alumno, como se puede utilizar Java para visualizar la magnitud que adquiere este vector a lo largo del plano coordenado *xy* cuando el mismo circuito está colocado en ese plano y su centro coincide con el origen de coordenadas.Davis y Kaplan [1], muestran como deducir este vector de Poynting. Siguiendo su resultado podemos decir que la expresión del vector de Poynting para este caso viene dado por la expresión matemática (en el sistema MKS de unidades). La configuración del fenómeno físico se puede observar en la Figura 1.

$$\begin{aligned}\mathbf{S} &= \varepsilon_0 c^2 \mathbf{E}\times\mathbf{B} \\ &= \frac{VIR^4}{8r^6\ln(R/r_0)}\left[\left(1-3\frac{y^2}{r^2}-3\frac{z^2}{r^2}\right)\hat{\mathbf{i}}\right. \\ &\quad \left.+\left(\frac{3xy}{r^2}\right)\hat{\mathbf{j}}+\left(\frac{3xz}{r^2}\right)\hat{\mathbf{k}}\right].\end{aligned} \tag{1}$$

En primer lugar hay que instalar Java [2] en nuestro sistema operativo de nuestra preferencia. Luego, para facilitar aún más las cosas instalamos el paquete de software de Francisco Esquembre de la universidad de Murcia en España [3]. El paquete se llama *Easy Java Simulations* y está pensado para personas que no son profesionales del

lenguaje de propósito general Java. El paquete se baja como un archivo "zip" de su sitio de internet [4]. Se desempaca en algún directorio como en "Mis Documentos" (si estamos usando Windows). Y se revisa ahora una carpeta llamada EJS_(versión), donde la expresión entre paréntesis varía de acuerdo a la versión de EJS que estemos usando en el momento. Adentro de la carpeta EJS_(versión) localizamos el archivo "EjsConsole.jar" y le damos doble clic, para iniciar a correr EJS. Cuando se corre EJS por primare vez, pregunta los datos personales del (o los) autor(es) de la simulación que se va a crear. Después de proporcionar los datos pertinentes, EJS abre dos ventanas, la que generalmente abre en la esquina inferior izquierda se llama la "consola de EJS" y sirve (entre otras cosas) para ver donde están instaladas las librerías de Java, y regular otros parámetros sencillos (como por ejemplo ubicar el directorio de trabajo que usará EJS).

La segunda ventana se llama la "interfaz" del usuario (o identificada aquí como ventana principal o de edición de EJS). Es aquí donde se edita un archivo de trabajo para EJS, que tendrá la extensión correspondiente ".ejs". Este es el archivo principal de trabajo de EJS, y a partir de él se puede generar otro tipo de archivo con extensión "jar", que es autoejecutable. De manera que si nuestro archivo de trabajo se llama (nombre_archivo).ejs, entonces el archivo autoejecutable que genera EJS llevará por nombre (nombre_archivo).jar". Basta con hacer doble clic en él para hacer correr la simulación. (Fig. 9).

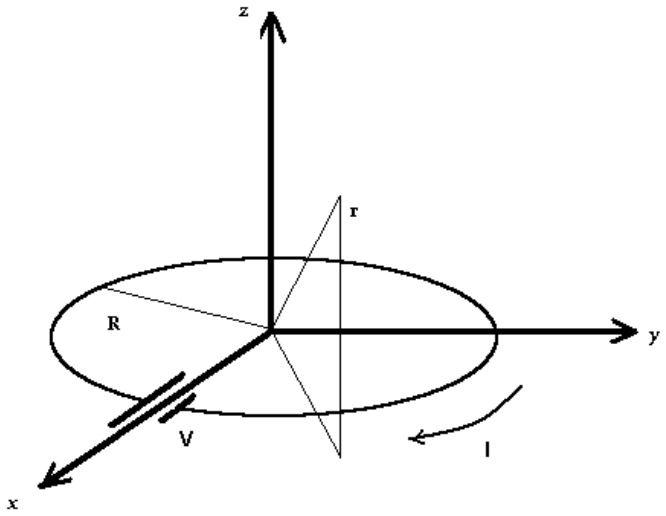

FIGURA 1. Circuito circular de corriente continúa, el punto de observación del vector de Poynting, está indicado con el vector de posición $\mathbf{r}$, con coordenadas (x,y,z). El circuito está alimentado por una pequeña pila situada en la intersección del circuito con el eje x.

II. ESCRIBIENDO LA SIMULACIÓN

Para empezar a escribir cualquier simulación Java en EJS, el autor de la misma debe de pasar por tres pasos o etapas básicas de edición. Estas etapas llevan por nombre en orden:

1. Descripción.
2. Modelo.
3. Vista.

Están identificados por tres radio–botones que se encuentran en la parte superior de la ventana de edición de EJS y que se pueden apreciar en la Figura 2.

En las siguientes secciones detallamos la manera de editar un Mapa de colores que representa, las intensidades que adquirirá el ventor de Poynting (ec. (1)), de un circuito de corriente continua de forma circular.

La parte de edición más sencilla, la constituye la Descripción de la simulación. En términos técnicos se necesitan editar en una hoja HTML, las características teóricas del fenómeno físico de la simulación. Aquí puede ir el material didáctico que el instructor de Física necesita detallarles a los alumnos. La hoja HTML hay que guardarla en un lugar seguro, al igual que el archivo "ejs", y las imágenes que se ocupen para construir la simulación. Ya que con estos archivos se puede reconstruir la simulación con EJS. Por esto, estos archivos fuente, son los archivos más importantes, ya que serían los archivos irremplazables para la reconstrucción de la simulación. Tanto el archivo "zip" de EJS como el archivo de instalación de Java, pueden posteriormente, reemplazarse al volver a bajarlos de internet. En las siguientes secciones vamos a describir el paso o fase de la construcción del Modelo, y posteriormente de la Vista de la simulación.

III. ESCRITURA DEL MODELO

En la parte superior de la ventana de interfaz del usuario, seleccionamos el radio-botón "Modelo", e inmediatamente

se abren otros seis subradio-botones, por debajo de los radio-botones originales, que aparecen de izquierda a derecha como "Variables", "Inicialización", "Evolución", "Relaciones fijas", "Propio" y "Elementos". Aquí sólo describo los más importantes para construir una simulación con las características básicas en ella. Empezamos por describir como editar la parte "Variables", en la siguiente sección.

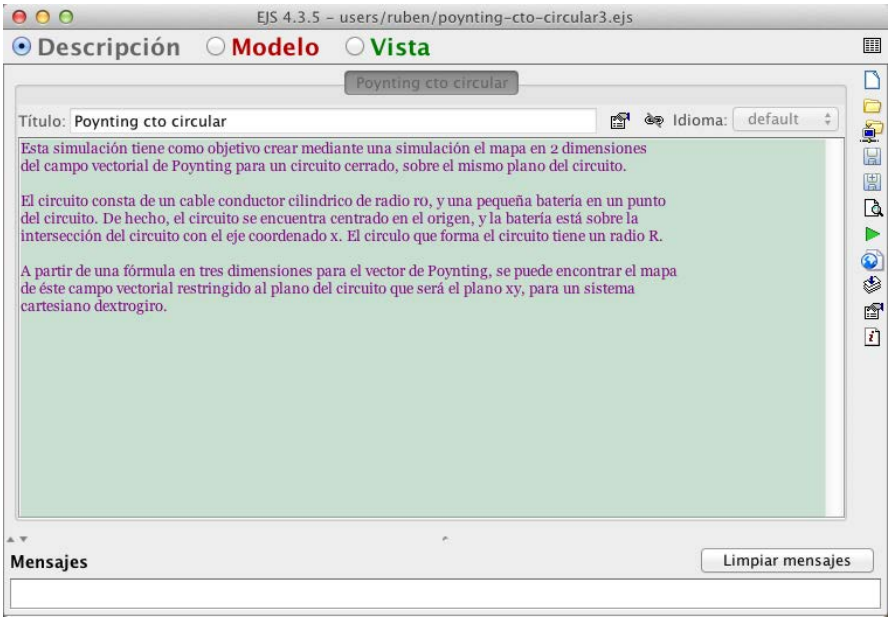

FIGURA 2. Descripción de la simulación para la magnitud del vector de Poynting, de un circuito circular de corriente continua.

IV. VARIABLES UTILIZADAS

Cuando seleccionamos el subradio-botón "Variables" construimos una página con una lista de todas las variables empleadas por la simulación. En este caso, estamos simulando, el mapa de intensidad del vector de Poynting para un circuito circular (Fig. 1), empleando colores. Creamos la lista de las siguientes variables:

Tabla Variables:

1. R, 200.0, doublé
2. r0, 1.0, doublé
3. I, 0.001, doublé
4. V, 300.0, doublé
5. m, 50, int
6. n, 2*m, int
7. dx, 4*R/n, double
8. dy, 4*R/n, doublé
9. dz, 4*R/n, doublé
10. x[i], i*dx-2*R, doublé, [n+1]
11. y[j], j*dy-2*R, double, [n+1]
12. z[k], k*dz-2*R, double, [n+1]
13. Sxy[i][j], , double, [n+1][n+1]
14. Syz[j][k], , double, [n+1][n+1]
15. Szx[k][i], , double, [n+1][n+1]
16. K, V * I * Math.pow(R,4) / (8 * Math.log(R/r0)), double

La lista de variables consta de varios renglones que se van llenando a manera de una tabla de hoja de cálculo, con cuatro columnas que son:

1. Nombre
2. Valor Inicial
3. Tipo
4. Dimensión

Por ejemplo, para la variable (constante) que denota el radio del circuito circular, la llamamos "R", le damos un valor inicial (y permanente) de "200.0", y en una variable que para Java es de tipo doublé (es decir, es un decimal de doble precisión), como lo podemos ver de la lista que nombramos "Tabla Variables". Otro ejemplo lo constituye una matriz $(n+1)\times(n+1)$ donde se almacenan los valores de la norma del vector de Poynting, para el plano coordenado xy. La línea 13 de la lista de variables llama Sxy[i][j] a este arreglo, no tiene un valor inicial (en la lista), el tipo de cada entrada es "doublé", y su dimensión es 2x2, pues el array que representa contiene n+1, elementos por cada dimensión, esto es lo que quiere decir la notación [n+1][n+1]. En la Figura 3, mostramos en forma gráfica como se puede llenar esta lista.

El significado que tiene cada variable, para el circuito circular de corriente continua, la escribimos en la siguiente lista:

1. R, radio exterior del circuito.
2. r0, radio interior del cable conductor.
3. I, magnitud de la corriente que circula.
4. V, voltaje de la fuente de corriente.

5. m, número de subdivisiones en que divido a una longitud horizontal de 2*R. Esta subdivisión sirve posteriormente para lograr la desratización de las coordenadas cartesianas.
6. n, número efectivo de subdivisiones para el eje cartesiano x, con longirud elegida de 4*R. también para los ejes y y z.
7. dx, se forma una malla tridimensional, de varios cubos, en el espacio donde se encuentra el circuito. La longitud de cada cubo en la dirección cartesiana x es “dx”.
8. dy, la malla del espacio real, tiene un arreglo de cubos con lado “dy” en la dirección vertical.
9. dz, la malla del espacio real del circuito tiene un arreglo de cubos, perfectamente ordenados en el sentido cartesiano, con lados “dz” en la dirección “z”.
10. x[i], cada vértice de la malla anterior tiene coordenadas cartesianas discretas “x[i]”, “y[j]”, “z[k]”.
11. y[j], coordenada “y” de un vértice de la malla.
12. z[k], coordenada “z” de un vértice de la malla.
13. Sxy[i][j], magnitud del vector de Poynting en el plano xy para el punto (x[i],y[j],0).
14. Syz[j][k], magnitud del vector de Poynting en el plano yz, para el punto (0,y[j],z[k]).
15. Szx[k][i], magnitud del vector de Poynting en el plano zx, para el punto (x[i],0,z[k]).
16. Constante que sirve para calcular la magnitud del vector de Poynting.

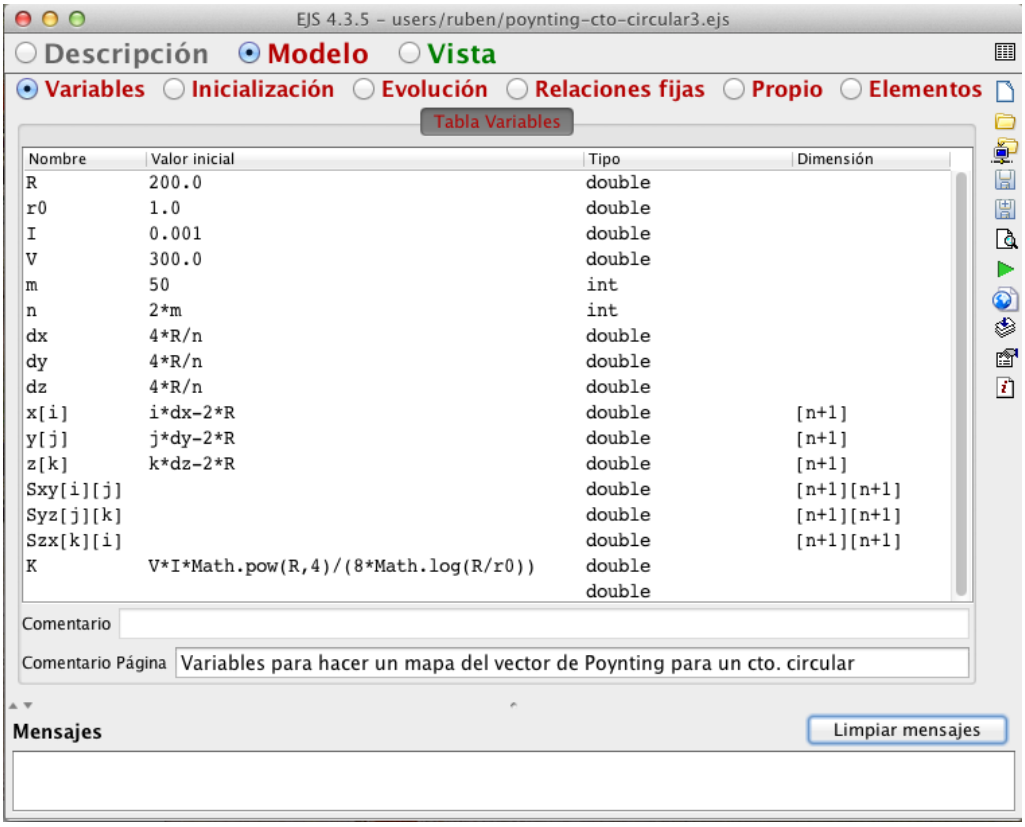

FIGURA 3. Lista de variables utilizadas.

V. INICIALIZACIÓN

Para editar esta hoja de la ventana principal de EJS, sólo necesitamos colocar el valor que tendrá el coeficiente constante *K*, que de hecho, ya la hemos inicializado en la sección “Variables”. En este ejemplo sólo requerimos anotar el valor de *K*, que está dado por, la sencilla expresión:

$$K = \frac{VIR^4}{8\ln(R/r_0)}. \qquad (2)$$

Esta constante aparece como coeficiente constante en la ecuación (1).

En la Figura 4, ilustramos como editar esta ventana escribiendo la expresión directamente en código Java.

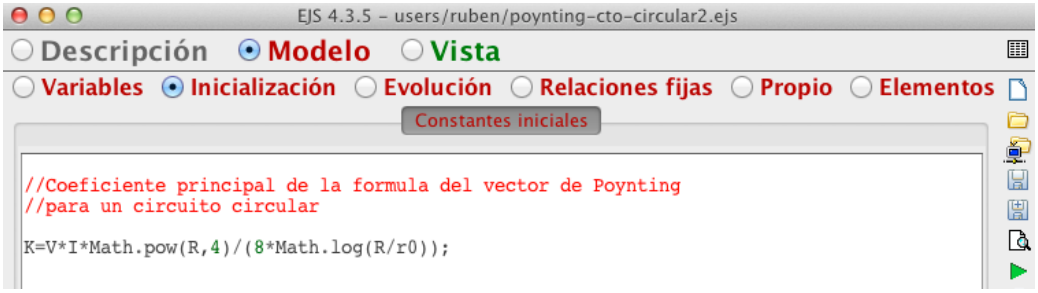

FIGURA 4. Coeficiente constante del vector de Poynting.

VI. EVOLUCIÓN

Para editar esta parte, utilizamos la magnitud que el vector de Poynting tiene sobre los planos coordenados xy, yz, zx. Utilizando para esto, el Teorema de Pitágoras aplicado a las componentes cartesianas del vector (1). Por ejemplo sobre el plano xy, se cumple $z[k]=0$. Y la magnitud que tendrá en vector de Poynting en el punto $(x[i], y[j], 0)$ del plano xy.

$$Sxy_{ij} = \ln\left(\frac{K}{r^6}\sqrt{|\,My\_Poynting2(i,j,k)\,|}\right). \qquad (3)$$

Como podemos observar de la ecuación (3), lo que en realidad estamos colocando en la simulación es el logaritmo de la magnitud del vector de Poynting.

También hemos escalado su magnitud original, con un logaritmo natural. La función $My\_Poynting2(i,j,k)$, toma el cuadrado de la magnitud de la parte principal del vector de Poynting, en este caso es:

$$My\_Poynting2(i,j,k) = \left(1-\frac{3(y_j{}^2+z_k{}^2)}{r^2}\right)^2 + \left(\frac{3x_i y_j}{r^2}\right)^2 + \left(\frac{3x_i z_k}{r^2}\right)^2. \qquad (4)$$

En este caso estamos sobre el plano xy, y debemos anular los valores de z_k en la fórmula (4). La manera de lograr esto, es sustituyendo $k=m$, en la fórmula (4) que representa el punto medio del eje z, esto es para $k=m$, la coordenada z está en el origen ($z=0$). Y la magnitud de $Sxy[i][j]$, va entonces a corresponder a la magnitud del vector de Poynting sobre el plano xy.

Para evaluar la magnitud del vector de Poynting sobre los planos coordenados yz, zx. Se utilizan un método similar. Resumiendo, tenemos las siguientes situaciones para cada plano coordenado.

1. $k=m=n/2, \Leftrightarrow z_k=0,$ relaciones que se satisfacen sobre el plano coordenado xy.
2. $i=m=n/2, \Leftrightarrow x_i=0,$ relaciones que se cumplen sobre el plano yz.
3. $j=m=n/2, \Leftrightarrow y_j=0,$ relaciones que se cumplen en el plano zx.

En la Figura 6, se puede observar cómo se emplean estos sencillos valores de los índices sobre los ciclos "for" externos de los ciclos, para obtener el logaritmo de la magnitud del vector de Poynting, sobre cada plano coordenado.

VII. PROPIO

Las funciones propias para esta simulación son varias expresiones de la distancia del punto de observación al origen, sobre los planos $xy, yz, zx,$ y en el espacio xyz. También damos la definición de la función $My\_Poynting2(i,j,k)$, de la ecuación (3). En la Figura 6 se puede observar como editar estas funciones. Como se puede observar, para editar una función en EJS, basta con seguir las reglas del lenguaje de programación Java, por lo que es recomendable que el autor de la simulación conozca un poco acerca de las reglas de sintaxis para dicho lenguaje de programación.

Después de la edición de las funciones propias de la simulación, podemos editar la parte correspondiente a la "Vista" de la simulación.

VIII. VISTA

Para construir la Vista de la simulación, necesitamos oprimir el radio-botón respectivo con la etiqueta "Vista". Para construir una Vista es necesario construir una estructura de datos en forma de árbol que va a constar de varios elementos gráficos, que serán los hijos de la raíz

común llamada “Vista de la simulación”. En la ventana de la Vista Se nos presentan dos regiones principales, la del lado izquierdo presenta un solo elemento gráfico que es la raíz de toda la construcción gráfica. En el lado derecho se presentan varias áreas y varias pestañas, que tienen varios elementos gráficos, que se usan para construir la Vista, a partir de la raíz del lado izquierdo.

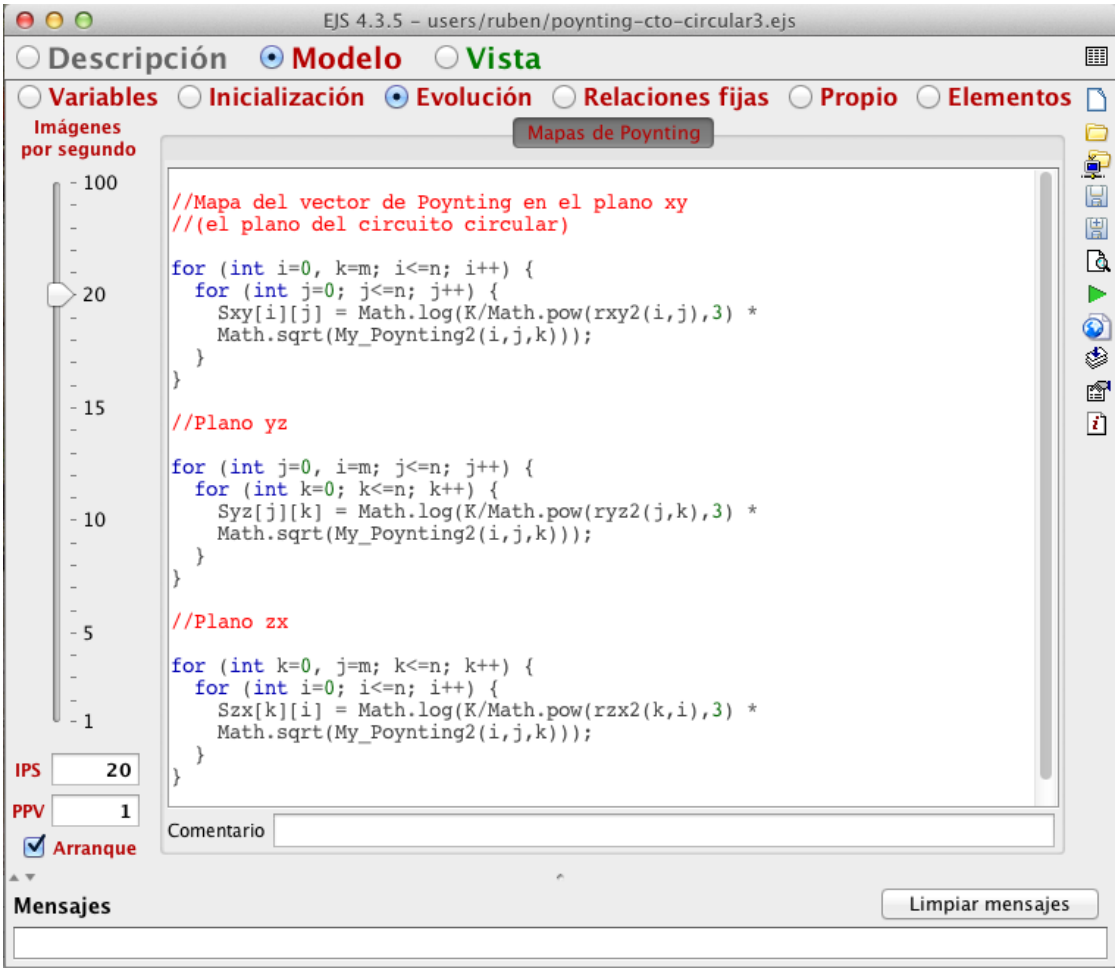

FIGURA 5. Edición de la evolución, donde se calcula la magnitud del vector de Poynting en los planos coordenados xy, yz, zx. La función My_Poynting2(i,j,k) denota el cuadrado de la magnitud de la parte principal del vector de Poynting. En la Figura 6 se define a esta función.

Entonces el procedimiento de construcción del árbol de la simulación, consiste en elegir un elemento del lado derecho y arrastrarlo hacia el lado izquierdo donde está la raíz principal. Los elementos gráficos se van así acumulando, forman la estructura de árbol. Un elemento gráfico puede funcionar como nodo o padre de otros, o bien fungir como hijos de otro elemento del árbol. Pero el árbol siempre tendrá un padre común a todos los elementos gráficos representado por la raíz del árbol. Como este procedimiento es repetitivo, hasta que se tienen todos los elementos gráficos, mostramos el árbol ya construido en la Figura 7.

Después, de la construcción del árbol como está mostrado en la Figura 7, sigue el paso de dotar de varias propiedades a cada uno de estos elementos. Para editar las propiedades de uno de éstos elementos, se hace clic derecho con el ratón en uno de ellos. Esto abre un menú emergente, con la opción “Propiedades”, eligiendo esta opción, se abre una ventana de propiedades del elemento respectivo. La ventana, contiene varios campos. Algunos de ellos se editan y otros se dejan en blanco para dejar el valor por default. Una manera de describir cómo llenar estos campos es enlistando las propiedades, junto con los valores que van en los campos. Como varios tipos de elementos se repiten para este ejemplo, es suficiente con enlistar las propiedades de varios elementos clave. A continuación describimos las propiedades de varios elementos.

En la Figura 8, se observa una ventana de nombre “Mapa_ Poynting_xy”, su listado de propiedades es el siguiente:

Propiedades de Mapa_Poynting_xy (Ventana):

1. Título: “Mapa_Poynting_xy”
2. Distribución: border
3. Visible: true

4. Posición: “30,157”
5. Tamaño: “300,300”

Otros elementos gráficos son los siguientes:

Propiedades de panelDibujo (PanelDibujo):

1. Autoscala X: true
2. Autoscala Y: true
3. Mínimo X: -2*R
4. Máximo X: 2*R
5. Mínimo Y: -2*R
6. Máximo Y: 2*R
7. Cuadrado: true

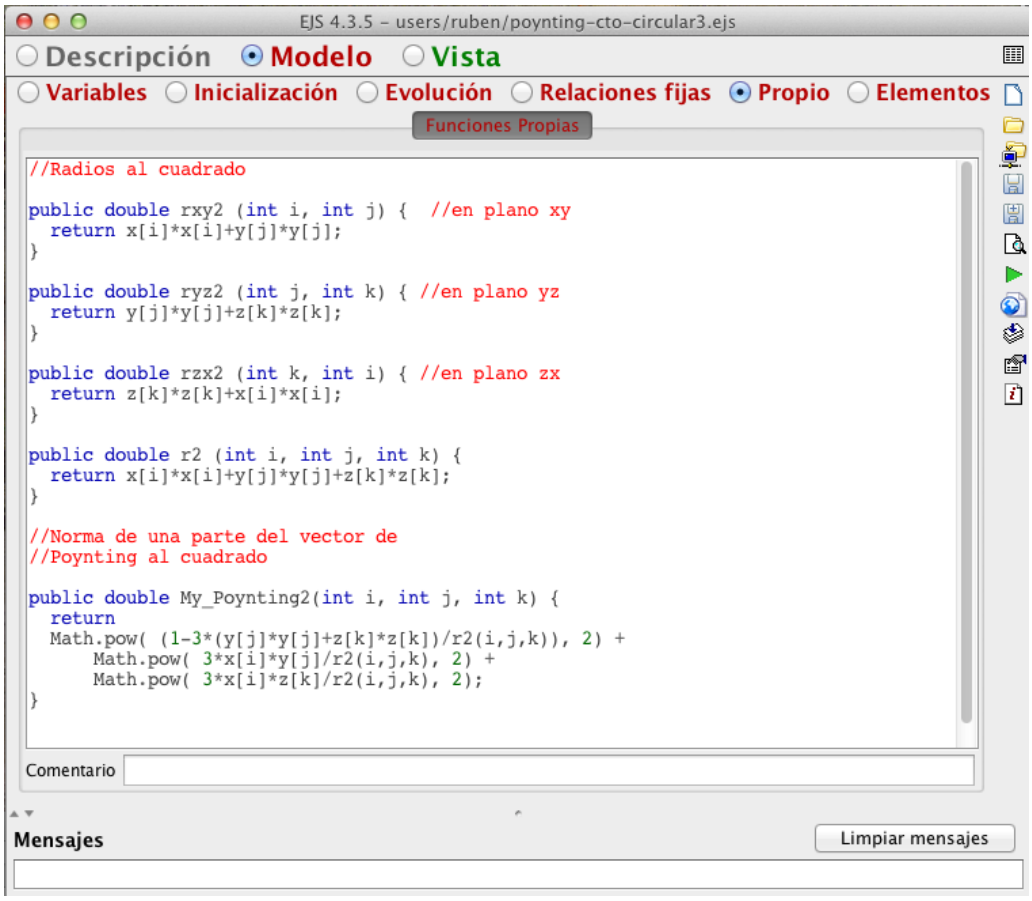

FIGURA 6. Aquí se muestran las funciones propias de la simulación.

Y tenemos las siguientes propiedades para el campo escalar de la magnitud del vector de Poynting en el plano coordenado *xy* :

Propiedades de campoPoynting_xy_2D (CampoEscalar2D):

1. Datos Z: Sxy
2. Autoscala Z: true
3. Mínimo Z: 0.0
4. Máximo Z: 1.0
5. Mínimo X: -2*R
6. Máximo X: 2*R
7. Mínimo Y: -2*R
8. Máximo Y: 2*R
9. Tipo de Gráfico: INTERPOLATED
10. Niveles: 400
11. Modo Color: SPECTRUM

Para hacer la curva de nivel “Poynting_xy”, se necesita llenar la ventana “propiedades” de este elemento gráfico como sigue:

Propiedades de Poynting_xy (CampoEscalar2D):

1. Datos Z: Sxy
2. Autoscala Z: true
3. Mínimo Z: 0.0
4. Máximo Z: 1.0
5. Mínimo X: -2*R
6. Máximo X: 2*R
7. Mínimo Y: -2*R
8. Máximo Y: 2*R
9. Tipo de Gráfico: SURFACE
10. Niveles: 400
11. Modo Color: REDBLUESHADE

Los demás elementos gráficos se editan de manera similar.

Una vez hecho esto, hemos prácticamente terminado de editar la simulación completa.

Después se construye una archivo autoejecutable de extensión "jar", para esto se oprime el botón que está en el margen derecho de la ventana principal de EJS. Véase por ejemplo, la Figura 7 para identificar este botón. El nombre del archivo como ya se había mencionado antes lleva el formato "(nombre_archivo).jar" si el nombre del archivo fuente es "(nombre_archivo).ejs".

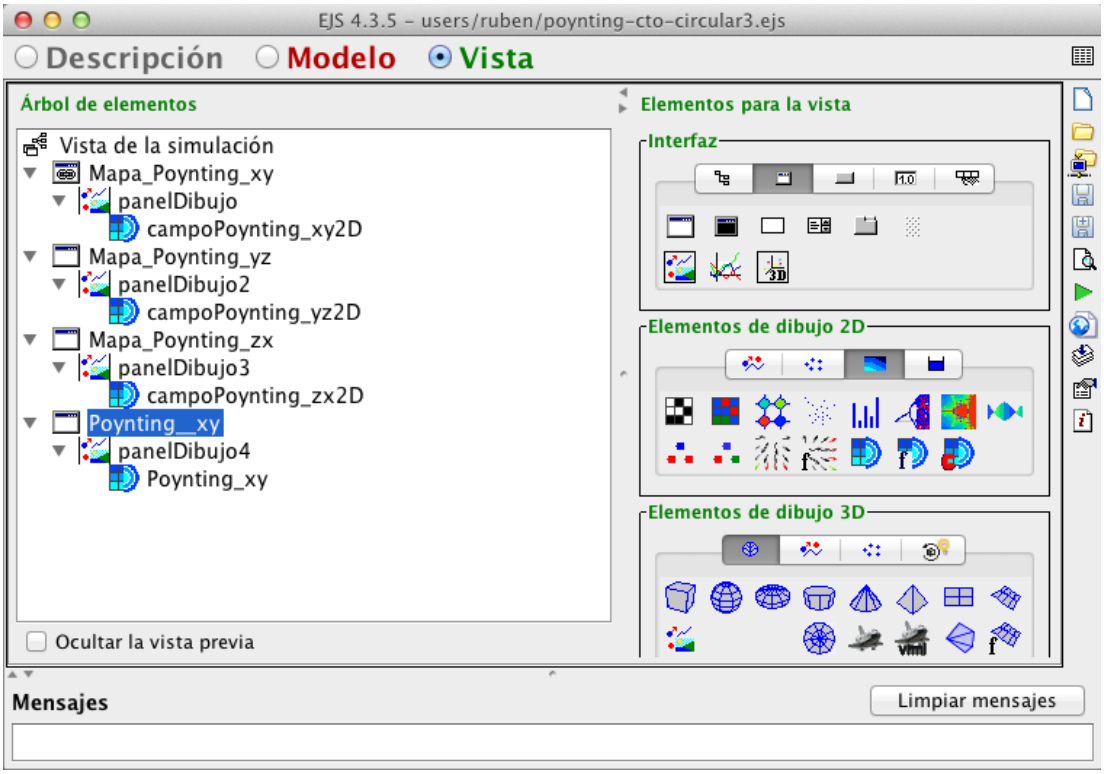

FIGURA 7. Árbol de elementos de la simulación.

FIGURA 8. Propiedades del elemento gráfico o ventana "Mapa_Poynting_xy", esta ventana contiene un elemento gráfico "PanelDibujo", que contiene a su vez un "CampoEscalar2D", donde se muestra en escala logarítmica la magnitud del vector de Poynting para el plano coordenado "xy", que es el plano donde se encuentra el circuito de la Figura 1.

IX. EXPOSICIÓN EN EL AULA DE CLASE

Para que el ejercicio este completo, se sugiere llevar la simulación al aula de clase, y apoyar la instrucción de los alumnos con la simulación. El tema del vector de Poynting está tomando un nuevo interés, para el análisis de cómo fluye la energía electromagnética para un circuito de forma circular, alimentado con corriente continua. El hecho de observar la magnitud del vector de Poynting, aclara la manera de cómo la densidad de energía se distribuye en el espacio. La distribución es simétrica, para todos los planos coordenados. La corrida de la simulación en un computador convencional se puede observar en la Figura 9.

Las simulaciones como la que hemos descrito, son un apoyo visual a la clase tradicional de Física, permitiendo al estudiante hacerse una idea más clara de los fenómenos físicos estudiados. Además algunos de los parámetros pueden ser modificados, por ejemplo para tener otras secciones de la distribución de energía. La densidad de energía decrece con la sexta potencia de la distancia que tiene el punto de observación del circuito r^6, según la aproximación descrita por la ecuación (1).

La simulación se hizo, para demostrar que la curvatura del alambre constructor contribuye a una pequeña corrección del vector de Poynting, para un circuito grande de radio *R*, y el radio del alambre es muy pequeño. Esto lo muestran Davis y Kaplan en su artículo [1].

La herramienta didáctica que representan las simulaciones para la clase de Física, es muy importante porque gracias a ellas la clase se vuelve más interactiva para los estudiantes. También permiten visualizar la magnitud de los campos, que se generan en el circuito, y que normalmente no se observarán en un laboratorio de Física tradicional. Los parámetros físicos se pueden cambiar con facilidad en la simulación de manera segura. Además se puede solicitar la distribución logarítmica de la magnitud del campo de Poynting sobre otros planos, que sean paralelos a los coordenados, realizando unos pocos ajustes en el código mostrado aquí.

Como podemos ver, las ventajas que ofrece la simulación hacen que la clase de Física tenga mayor interés. Por eso, no hay que menospreciar el potencial y el alcance que tienen las simulaciones en la educación. El ejemplo aquí mostrado nos sirve como una referencia de cómo el instructor de Física, puede utilizar Java y EJS para crear un elemento didáctico con relativa facilidad. No hay que olvidar los elementos lúdicos que puede tener una simulación. Estos elementos se pueden hallar normalmente en los juegos por computadora. De hecho, según Aldrich [5, 6] y Quinn [7] las simulaciones educacionales deben de ser atractivas a los estudiantes, para que puedan tener éxito. Este punto es importante, ya que entre más atractiva sea la clase de Física, los educandos aprenderán con mayor facilidad. En esta época de tecnología, software e internet, el implementar este tipo de herramienta debe ser más o menos sencillo para el instructor.

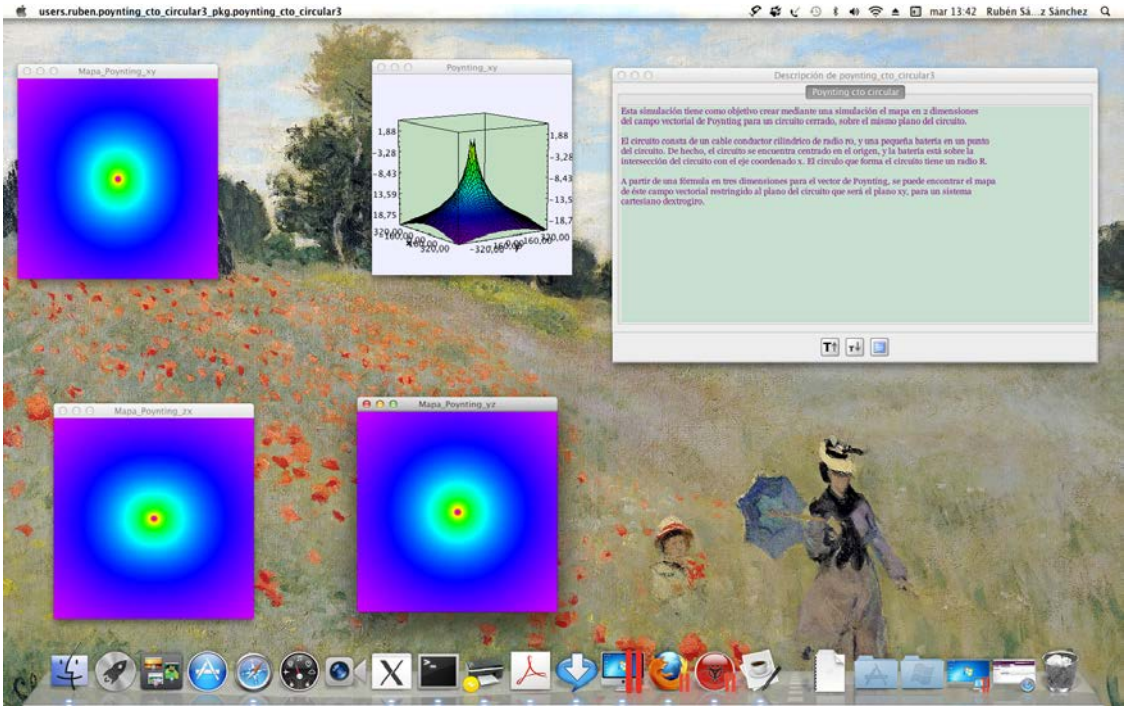

FIGURA 9. Corrida de la simulación, se observa que la densidad de energía electromagnética tiene una distribución simétrica para todos los planos coordenados. La primera (esquina de arriba a la izquierda) ventana corresponde al logaritmo natural de la magnitud del campo de Poynting, sobre el plano "*xy*". A su derecha está el mismo campo, representado mediante una curva de nivel. En el extremo derecho y arriba de esta pantalla esta la descripción de la simulación. Las ventanas de la parte inferior de la pantalla corresponden a las magnitudes logarítmicas del mismo vector de Poynting, pero en los planos "*yz*", "*zx*" respectivamente y en el orden de izquierda a derecha.

AGRADECIMIENTOS

El autor quiere dar su agradecimiento al apoyo otorgado por la COFAA del Instituto Politécnico Nacional (IPN), al proyecto SIP 20131706 de la Secretaría de Investigación y Posgrado del IPN y al Consejo Nacional de Ciencia y Tecnología (CONACyT) de México, D. F.

REFERENCIAS

[1] Davis, B. S., Kaplan, L., *Poynting vector flow in a circular circuit*, Am. J. Phys. **79**/11, 1155-1162 (2011).
[2] Oracle, <http://java.com/es/download>, consultado el 5 de Diciembre de 2011.
[3] Francisco, E., Easy Java Simulations (EJS), <http://fem.um.es/Ejs>, consultado el 5 de diciembre de 2011.
[4] Francisco, E., Easy Java Simulations (EJS), http://www.um.es/fem/EjsWiki/Main/Download, consultado el 5 de Diciembre de 2011.
[5] Aldrich, C., *Learning Online with Games, Simulations, and Virtual Worlds: Strategies for Online Instruction*, (Jossey-Bass, San Francisco, 2009).
[6] Aldrich, C., *Learning by Doing: A comprehensive Guide to Simulations, Computer Games, and Pedagogy in e-Learning and Other Educational Experiences*, (Pfeiffer, San Francisco, 2009).
[7] Quinn, C. N., *Engaging Learning: Designing e-Learning Simulations Games*, (Pfeiffer, San Francisco, 2005)
[8] Jackson, J. D., *Classical Electrodynamics*, (Wiley, 2da. Edición, USA, 1975).
[9] Francisco, E., *Creación de Simulaciones Interactivas en Java: Aplicación en la enseñanza de la Física*, (Pearson, Prentice-Hall, Madrid, 2005).
[10] Flanagan, D., *Java in a Nutshell: A Desktop Quick Reference*, (O'Reilly, 5ta. Edición, USA, 2005).
[11] Sierra, K. and Bates, B., *Head First Java*, (O'Reilly, 2da. Edición, USA, 2002).