Simulación de la ecuación de calor en coordenadas polares

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

E-mail: rsanchezs@ipn.mx

(Recibido el 5 de Octubre de 2011; aceptado el 20 de Noviembre de 2011)

Resumen

En este documento se analiza la forma en cómo se puede implementar una simulación de la ecuación de calor para una placa conductora circular utilizando coordenadas polares. Tiene el objetivo de servir como material didáctico en una clase de Física a Nivel Superior donde se vea el tema de la difusión de calor en recintos circulares. Se ha observado que el proceso de enseñanza aprendizaje mejora con la utilización de material de apoyo como lo es el software de simulación, donde el alumno puede fácilmente ver cómo evoluciona el fenómeno y además puede modificar las condiciones iniciales del problema. Hecho, que beneficia la asimilación de los conocimientos físicos involucrados en el fenómeno.

Palabras clave: Difusión del calor, coordenadas polares, enseñanza de la Termodinámica.

Abstract

This paper discusses how to implement a simulation of the heat equation for a circular conducting plate using polar coordinates. It aims to serve as teaching material in a physics class at a High School level where you see the issue of heat diffusion in circular enclosures. It has been observed that the teaching-learning process improves with the use of support material such as simulation software, where the student can easily see how the phenomenon evolves and she can modify the initial conditions. Indeed, it benefits the assimilation of physical skills involved in the phenomenon.

Keywords: Heat diffusion, polar coordinates, teaching of Thermodynamics.

PACS: 44.05.+e, 44.20.+b, 01.40.-d, 01.40.E- **ISSN 1870-9095**

I. INTRODUCCIÓN

En estos tiempos el desarrollo de la investigación en educación ha tenido una buena perspectiva en el empleo de simulaciones para hacer más interactiva las clases [1, 2, 3, 4, 5]. En el presente trabajo, mostramos como construir una simulación de la difusión de calor sobre un disco circular con una distribución inicial de temperatura. El calor fluye de acuerdo a la ecuación diferencial del calor en coordenadas polares. Éste ejemplo trabaja en la clase de Física con la finalidad de simplificar la exposición del profesor. El lenguaje de programación nativo para escribir este código es Java [6]. Este lenguaje tiene la propiedad de ser independiente de la plataforma que se use para la simulación y es muy versátil para escribir programas de propósito general. Con el objeto de facilitar la programación de esta simulación nosotros usamos el paquete de software *Easy Java Simulations* [7, 8], el cual se escribió con el fin de auxiliar a usuarios inexpertos en Java a construir simulaciones de fenómenos físicos. Nosotros trataremos de ser lo suficientemente claros como podamos, para mostrar los pasos fundamentales en la programación de esta interesante simulación. Las condiciones de frontera dadas aquí, pueden fácilmente ser cambiadas de acuerdo a las necesidades de una clase de Física donde se estudie el fenómeno de la difusión de calor. Esperamos que éste ejemplo sea de gran utilidad en la comprensión de cómo utilizar y emplear Easy Java Simulations (EJS), como un auxiliar educativo. Revisaremos cómo preparar con ayuda de éste software, una clase interactiva de Física, que sea del interés de los estudiantes.

II. CONFIGURACIÓN PRELIMINAR

Antes de que empecemos con la programación del fenómeno físico de la difusión de calor en un recinto circular, nosotros necesitamos alguna configuración preliminar de nuestro computador. Empezamos este paso, al bajar Java, de su sitio oficial [6]. Después en Windows, sólo ejecutamos el archivo que bajamos de internet y así instalamos Java. Éste procedimiento varía de acuerdo a la plataforma y recomendamos seguir las instrucciones que el propio sitio web de Java nos diga, según sea nuestro caso.

Después de que instalamos Java, procedemos con la instalación del propio paquete EJS. Éste paso se logra al visitar la página web de EJS [7], y bajar el archivo "zip" correspondiente. Aconsejamos bajar la versión más actual, ya que el sitio constantemente agrega actualizaciones del paquete, y las características de EJS van mejorándose y actualizándose con el tiempo. Entonces bajamos de la red de internet un archivo comprimido "zip" de nombre EJS_(versión)_(construcción).zip, donde (versión) y (construcción) se refieren a una serie de números que varían de acuerdo a la versión y la construcción del paquete EJS que estemos bajando. A continuación desempaquetamos el archivo en algún subdirectorio de nuestro gusto. Por ejemplo en Windows se puede usar la carpeta de "Mis documentos". Aquí aparecerá una nueva subcarpeta de nombre EJS_(versión). Donde hemos usado la notación (versión) para designar al número de versión de EJS que instalemos. Por lo tanto deberemos de reemplazar el paréntesis por los números correspondientes de la versión de EJS que estemos utilizando. Dentro de folder EJS_(versión) podemos buscar un archivo autoejecutable de nombre *EjsConsole.jar*. Basta con darle doble clic al icono  para que EJS pueda empezar a correr.

Cuando EJS corre por primera vez, nos pregunta unos datos personales sencillos para guardarlos en un registro. Esta información sirve para futura referencia de los autores de la simulación.

En una corrida normal, EJS muestra dos ventanas, una de las cuales es la "consola" y la otra es la "interfaz de usuario". En la consola nosotros podemos ajustar nuestro folder personal de trabajo llamado "*workspace*", que es donde se guardan tanto el archivo fuente "ejs" de trabajo, como el archivo "empaquetado" de extensión "jar" que es autoejecutable. También es aquí, donde se pueden observar los mensajes que EJS, muestra a sus usuarios, durante el desarrollo de una simulación. La ventana llamada "interfaz de usuario", que podemos llamar aquí, la *ventana principal de EJS,* quizá sea la ventana más importante, para el usuario que por primera vez trata de preparar una simulación con ayuda de EJS. Es aquí donde se editan y preparan las simulaciones.

Nuestro ejemplo, describe mediante un mapa de colores (que en realidad es un campo 2D para EJS), la distribución de temperaturas en un disco circular plano, que es conductor de calor. Éste disco tiene una distribución de temperaturas inicial para cada uno de sus puntos, dada según la Ec. (7), en coordenadas polares. Además se supone que el disco se encuentra inmerso en un depósito de calor a temperatura constante de 1.0. Vale la pena mencionar en este punto, que la simulación se prepara de tal forma que el mapa de temperaturas está normalizado. De esta forma todos los valores de temperatura que se observan varían en el rango que va desde 0.0 a 1.0. Las unidades de temperatura son entonces simbólicas y pueden escogerse dentro del sistema MKS, o el sistema inglés de unidades. En la siguiente sección empezaremos por describir cuales son las fases o etapas principales de edición para esta simulación. Aunque cada simulación es única en características y forma de edición, esperamos que el lector, encuentre útil éste material, como apoyo a los pasos esenciales que se deben de seguir para elaborar una simulación.

III. TRES PASOS BÁSICOS

Para crear una simulación utilizando EJS, nosotros necesitamos primero entender que el proceso de escritura de un programa bajo EJS se realiza en tres pasos principales, los cuales se llaman respectivamente

1. Descripción.
2. Modelo.
3. Vista.

Cada uno de éstos pasos está asignado a un radio botón del mismo nombre que se encuentra en la parte superior de la interfaz de usuario o ventana principal de EJS. Cuando seleccionamos el radio botón Descripción y damos un clic en la interfaz se nos pregunta por un nombre para la página en formato HTML, que acompañará a la simulación. Aquí se pueden anotar los elementos teóricos de la simulación. Por lo tanto, la edición de ésta página es relativamente sencilla, y sirve como material auxiliar para la clase. (véase la Fig. 1).

Cuando elegimos el radio botón *Modelo,* estamos tratando de editar las ecuaciones diferenciales fundamentales del fenómeno físico que estamos estudiando, así como las variables que intervienen en ellas. Algunas simulaciones, requieren de la definición de ciertas funciones propias, que utilizaremos para describir la evolución del fenómeno mediante ecuaciones diferenciales. Además en ciertos casos, también se llegan a ocupar relaciones matemáticas que permanecen sin alteración durante la simulación y de cuya satisfacción depende la correcta evolución de la simulación.

Finalmente cuando seleccionamos el radio-botón *Vista,* nos encontramos con los elementos visuales de la simulación que dependen de la evolución de las variables del Modelo. Es en ésta parte donde se editan las características necesarias para que un usuario pueda poner a correr la simulación, la pueda detener, o pueda dar un paso de su desarrollo en el tiempo.

En las siguientes secciones detallamos como realizar cada uno de éstos pasos para la simulación de la difusión de una distribución inicial de calor sobre un recinto conductor circular.

IV. LA DESCRIPCIÓN

Este es el primer paso o la primera etapa de la construcción de una simulación en EJS. Simplemente se selecciona el primer radio botón de la ventana "interfaz de usuario" de EJS, y se escribe un nombre para la página HTML. A continuación, se edita ésta página, con la información relevante acerca del fenómeno físico que se representa en la simulación. Aquí puede ir todo el material de lectura que el instructor juzgue como importante, para la comprensión teórica del problema a tratar en la simulación. El material

didáctico que va en esta sección, variará de acuerdo a las necesidades específicas de la clase. En la Fig. 1, por simplicidad de exposición hemos optado por mostrar una descripción mínima (el nombre del fenómeno simulado), pero obviamente, se pueden agregar aquí todas los detalles y características del fenómeno estudiado.

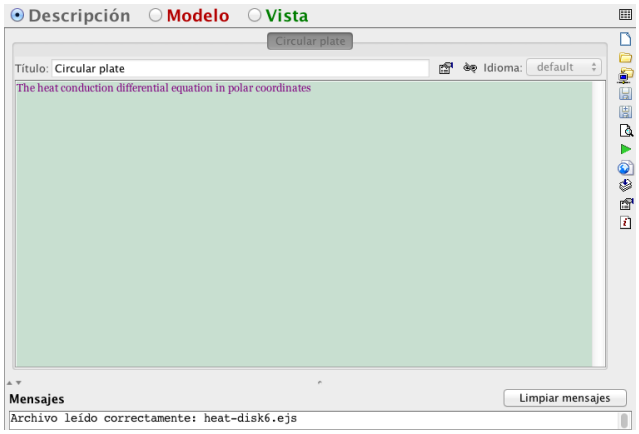

FIGURA 1. Se muestra una hoja HTML que constituye la primera parte de la construcción de toda simulación EJS y que forma el paso conocido como la *Descripción* de la simulación.

V. EL MODELO

La segunda etapa de toda construcción de simulación en EJS se llama el "Modelo". Una vez que seleccionamos el segundo radio-botón de la parte superior de la ventana principal o interfaz de usuario de EJS (véase Fig. 1), nos encontramos con varios subradio-botones que describen paso a paso como se va construyendo el modelo matemático que yace en el fondo de toda simulación.

Todo modelo consta a su vez de seis pasos o etapas de edición que están nombradas de acuerdo al subradio-botón respectivo en la interfaz (véase la Fig. 4). Estos subradio-botones son:

1. Variables
2. Inicialización
3. Evolución
4. Relaciones Fijas
5. Propio
6. Elementos

En este trabajo, se necesitan emplear sólo los primeros 5 pasos, ya que el último tiene que ver con la preparación de una simulación más elaborada, que tiene mayor manejo de las herramientas de internet, y que perfeccionan y retocan más el trabajo final. Pero para una clase sencilla de Termodinámica, basta con revisar los primeros cinco pasos. De hecho, para construir el modelo, es esencial la comprensión matemática de las ecuaciones diferenciales que gobiernan la evolución de éste fenómeno físico de difusión del calor. Para implementarla en éste tipo de ejemplo, se necesita un análisis profundo de los métodos numéricos que implementarán en forma aproximada las razones de cambio del estado del sistema físico. Esto es, necesitamos analizar numéricamente a las derivadas de la ecuación del calor en coordenadas polares. Además también se necesita derivar una condición numérica de *estabilidad.* Ya que los intervalos de tiempo *dt*, que se usan para hacer evolucionar la distribución de calor sobre el disco, de un estado previo al siguiente no pueden ser arbitrarios. Si no, la simulación colapsa, por el error numérico acumulado luego de varios pasos de evolución. Para aproximar las derivadas en forma numérica recomendamos revisar el libro de Mathews [9]. Para encontrar el intervalo óptimo de tiempo *dt*, recomendamos el trabajo de J. Crank [10]. Luego de un cálculo sencillo es posible hallar que *dt* debe de satisfacer la siguiente desigualdad:

$$dt \le \frac{dr^2\, d\theta^2}{\beta(2 + d\theta^2)}. \tag{1}$$

Donde definimos a la constante beta β de difusión como la que aparece en la ecuación de diferencial parcial del calor

$$\frac{\partial u}{\partial t} = \beta\left(\frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial u}{\partial r}\right) + \frac{1}{r^2}\frac{\partial^2 u}{\partial \theta^2}\right). \tag{2}$$

Otras ecuaciones que tomamos en cuenta, para la evolución del sistema son las de aproximación numérica de las derivadas para coordenadas polares. Una de ellas es la de la parte radial de la Ec. (2):

$$\begin{aligned}\frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial u_{ij}}{\partial r}\right) &= \frac{1}{2i\left(dr\right)^2} * \\ &\left[(2i+1)u_{i+1\,j} - 4iu_{ij} + (2i-1)u_{i-1\,j}\right], \quad i \neq 0, m, \\ &\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad j \neq 0, n, \\ &= \frac{4}{\left(dr\right)^2}(u_{1j} - u_{0j}), \quad i = 0, \\ &= \frac{1}{2i(dr)^2} * \\ &\left[(2i+1)h(t) - 4iu_{ij} + (2i-1)u_{i-1\,j}\right], \; i = m.\end{aligned} \tag{3}$$

En donde el índice *i* de la variable dependiente *u* está ligado con la variable radial *r*, de manera que está asociado a la coordenada discreta $r = r_i = r[i] = i\,dr$, y el índice j de *u* está ligado con la variable angular θ, de esta forma este índice está asociado con la variable discreta $\theta = \theta_j = \theta[j] = j\,d\theta$. En cuanto a la otra variable independiente, que es el tiempo *t*, se maneja de otra forma, tenemos las funciones *u* y *u1*, de tal forma que $u = u_{ij} = u[i][j]$ representa la temperatura del disco para

las coordenadas $r[i]$ y $\theta[j]$ a un tiempo *t*, dado. Para un tiempo posterior *t+dt*, tenemos que la distribución de temperatura para el mismo punto es $u1_{ij} = u1[i][j]$. También para simplificar las condiciones de frontera en r, hemos tomado como función de frontera a $h(t) = 1.0$, cuando r toma el valor límite $r[m] = R$. Ésta condición de frontera se interpreta físicamente, como que el disco conductor de calor tiene una fuente de calor externa que permanece a temperatura constante 1.0. De manera que conforme pasa el tiempo este calor se difunde en el disco poco a poco.

Además hay que notar que por comodidad hemos hecho la aproximación de las diferenciales de las variables independientes correspondientes r, θ, *t*, a sus incrementos respectivos. Así tenemos que implícitamente hemos hecho las siguientes identificaciones

$$\Delta r = dr, \quad \Delta\theta = d\theta, \quad \Delta t = dt. \tag{4}$$

La segunda derivada parcial angular con respecto a θ de la Ec. (2), la aproximamos como sigue:

$$\begin{aligned} \frac{1}{r^2}\frac{\partial^2 u_{ij}}{\partial\theta^2} &= \frac{1}{i^2(dr)^2(d\theta)^2} * \\ &\left[u_{i\,j-1} - 2u_{ij} + u_{i\,j+1}\right], \quad j \neq 0, n, \quad i \neq 0, m, \\ &= \frac{1}{i^2(dr)^2(d\theta)^2} * \\ &\left[u_{in} - 2u_{ij} + u_{i\,j+1}\right], \quad j = 0, \\ &= \frac{1}{i^2(dr)^2(d\theta)^2} * \\ &\left[u_{i\,j-1} - 2u_{ij} + u_{i\,j+1}\right], \quad j = n. \end{aligned} \tag{5}$$

En esta ecuación se han tomado los casos de condición de frontera angular cuando la medida angular inicia de cero $\theta = 0,$ (que corresponde a $j = 0,$ esto es, en forma discreta $\theta[0] = 0$) y cuando da exactamente una vuelta completa en el disco $\theta = 2\pi,$ (que corresponde a $j = n,$ esto es, en forma discreta tenemos $\theta[n] = 2\pi$).

El lado izquierdo de la Ec. (2) denota la derivada parcial con respecto al tiempo de la distribución de temperaturas *u*. Se aproxima fácilmente utilizando las funciones *u* y *u1*, que son las distribuciones en los tiempos *t* y *t+dt*, respectivamente. Por lo tanto el primer miembro de la Ec. (2), se aproxima como:

$$\frac{\partial u}{\partial t} = \frac{u1_{ij} - u_{ij}}{\Delta t} = \frac{u1_{ij} - u_{ij}}{dt}. \tag{6}$$

Otras consideraciones del modelo matemático, son la distribución inicial de temperatura, la cual simbolizamos con la variable dependiente $z_{ij} = z[i][j].$ Sus valores describen una distribución inicial de temperaturas dadas por la siguiente ecuación

$$z_{ij} = e^{-0.0002*r^2}\,\sin(\theta)^2. \tag{7}$$

Esta distribución inicial de temperaturas puede ser cambiada, posteriormente. Y la hemos elegido aquí, sólo como un ejemplo.

VI. LAS VARIABLES

Las variables que intervienen tanto en la ecuación diferencial parcial de calor, como en la simulación de la difusión, aparecen en la Fig. 2. Algunas de estas variables son:

1. t (el tiempo).
2. dt el intervalo de tiempo, entre cada paso de la simulación.
3. R el radio del recinto circular.
4. β la constante de difusión de calor.
5. m el radio lo dividimos en este número de intervalos.
6. n un círculo de 2π radianes lo dividimos en este número de intervalos.
7. $dr = \Delta r$ Paso de R.
8. $d\theta$ Paso angular.
9. $r_i = r[i]$ Coordenada *i*-ésima radial.
10. $\theta_j = \theta[j]$ Coordenada *j*-ésima angular.
11. u_{ij} temperatura en el punto polar *ij* al tiempo *t*.
12. z_{ij} distribución inicial de temperaturas.
13. $u1_{ij}$ temperatura para el punto polar (r_i, θ_j) a un tiempo posterior *t+dt*.
14. a longitud horizontal del cuadrado circunscrito al disco circular, correspondiente a la primera ventana de la simulación (Fig. 2) donde se muestra el recinto circular.
15. $b = a$ longitud vertical o altura del cuadrado circunscrito al disco circular.
16. $dx = \Delta x$ intervalo horizontal en coordenadas cartesianas.
17. $dy = \Delta y$ intervalo vertical en coordenadas cartesianas.
18. U_{pq} temperatura para el punto cartesiano (X_p, Y_q), sobre el recinto circular.
19. $\phi = phi$. Fase inicial para el cambio de coordenadas cartesianas a polares.

20. $x_{ij} = x[i][j]$ Abscisa rectangular cartesiana correspondiente al punto polar (r_i, θ_j).
21. $y_{ij} = y[i][j]$ Ordenada rectangular cartesiana correspondiente al punto polar (r_i, θ_j).
22. $Erre[p][q] = \sqrt{X[p]^2 + Y[q]^2}$ Coordenada polar radial, sobre el recinto circular de la simulación (véase la Fig. 2).
23. $Theta[p][q] = \arctan\left(\frac{Y[q]}{X[p]}\right) - phi$ Coordenada polar angular, correspondiente al recinto circular mostrado en la segunda ventana de la simulación (véase la Fig. 2).
24. r_index_change función de índice polar radial correspondiente a un par de índices rectangulares.
25. $theta\_index\_change$ función de índice polar angular correspondiente a un par de índices rectangulares, previamente calculados.
26. $epsilon\_r = dr/2$, mitad de separación entre dos valores de r seguidos (p. ej. $r[i]$ y $r[i+1]$).
27. $epsilon\_theta = dtheta/2 = d\theta/2$, mitad de separación entre dos valores de θ seguidos (p. ej. $\theta[j]$ y $\theta[j+1]$).

Quizá un aspecto notorio de todas estas variables, es que se usaron para hacer un cambio de variables de coordenadas polares a rectangulares. La simulación consta de dos ventanas. En la primera ventana, vamos a mostrar al recinto circular o disco conductor de calor tal como es (véase la Fig. 2). Las coordenadas horizontales son las $X[p]$ y las verticales $Y[q]$, y corresponden a un sistema cartesiano o rectangular. Además de estas coordenadas existen otro tipo de variables para localizar un punto sobre el plano de esta primer ventana de la simulación que corresponden a las coordenadas circulares o polares de la Fig. 2, éstas variables las identificamos con las letras $Erre[p][q]$ y $Theta[p][q]$.

En la segunda ventana (véase la Fig. 3), representamos al mismo disco, pero aquí la coordenada radial ($r[i]$) tiene una dirección y carácter cartesiano horizontal, es decir, las abscisas corresponden a la coordenada radial del disco de la primera ventana. En cuanto a las ordenadas de esta segunda ventana ($\theta[j]$), ellas corresponden a la coordenada angular del disco mostrado en la primera ventana (Fig. 2). Esto funciona así porque se supone que existe un mapeo matemático que va de la segunda ventana a la primera, y corresponde al cambio de coordenadas de polares a cartesianas. El cambio de coordenadas, propiamente dicho se maneja con el conjunto de ecuaciones siguientes

$$\begin{aligned} x &= r\cos(\theta + \phi), \\ y &= r\sin(\theta + \phi). \end{aligned} \tag{8}$$

Donde la variable auxiliar ϕ representa una "fase inicial" que puede ser escogida a conveniencia. De esta forma la distribución de temperaturas se "rota" por un ángulo inicial arbitrario. Obviamente en la segunda ventana de la simulación (Fig. 3), se muestra la misma distribución de temperaturas que en la Fig. 2, pero bajo el mapeo inverso.

De manera, que podemos decir que la distribución mostrada en la Fig. 2, sufrió una transformación continua matemática, descrita por la transformación dada por el conjunto de Ecs. (8).

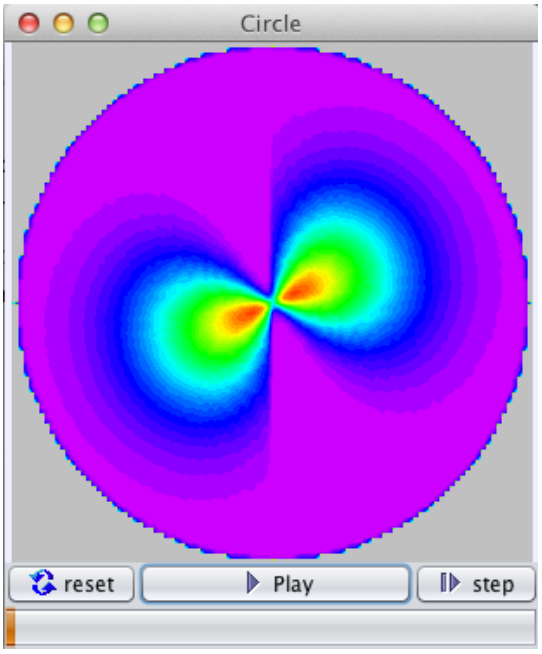

FIGURA 2. Primera ventana de la simulación "Circle" (recinto circular de la simulación).

En cuanto a la larga lista de variables que participan en esta simulación la colocamos en la Fig. 4, donde mostramos como llenar los campos de cada renglón para el subradio-botón "Variables" de la interfaz de usuario de EJS. Para clarificar cada una de estas variables, las enlistamos a continuación:

Variables:

1. t, 0.0, double
2. dt, 0.044, double
3. R, 150.0, double
4. beta, 1.0, double
5. m, 100, int
6. n, 100, int
7. dr, R/m, double
8. dtheta, 2*Math.PI/n, double
9. epsilon_r, dr/2, double
10. épsilon_theta, dtheta/2, double
11. r[i], i*dr, double, [m+1]
12. theta[j], j*dtheta, double, [n+1]
13. u, , double, [m+1][n+1]
14. z, , double, [m+1][n+1][1]
15. u1, ,double, [m+1][n+1]
16. r1, 1/(2*dr*dr), double
17. r2, 1/(dr*dr*dtheta*dtheta), double
18. M, 100, int
19. N, 100, ,int
20. a, 2*R, double
21. b, 2*R, double
22. dx, a/M, double
23. dy, b/N, double
24. U, , double, [M+1][N+1]
25. X[p], p*dx-R, double, [M+1]
26. Y[q], q*dy-R, double, [N+1]
27. phi, 2.0, double
28. x[i][j], r[i]*Math.cos(theta[j]+phi), double, [m+1][n+1]
29. y[i][j], r[i]*Math.sin(theta[j]+phi), double, [m+1][n+1]
30. Erre[p][q], Math.sqrt(X[p]*X[p]+Y[q]*Y[q]), double, [M+1][N+1]
31. Theta[p][q], Math.atan(Y[q]/X[p])-phi, double, [M+1][N+1]
32. r_index_change, 0, int, [M+1][N+1]
33. theta_index_change, 0, int, [M+1][N+1]

En esta lista, cada renglón separa cada campo de la ventana tipo hoja de cálculo (véase la Fig. 4) mediante comas. El primer campo de cada renglón representa el "nombre" de cada variable, el segundo campo representa su "valor inicial", el tercer campo denota su "tipo", y el último campo se emplea para los arreglos de Java, y denota su "dimensión", (para identificar todos estos nombres entre comillas, véanse las columnas mostradas en la Fig. 4).

En el último campo, cuando las variables son arreglos de una dimensión se muestra un número entre paréntesis cuadrados en la forma [n+1], el "n+1" es el número de elementos en este arreglo de una dimensión. Cuando el último campo tiene la forma [m+1][n+1], entonces la variable denota un arreglo Java de dos dimensiones. Un arreglo de dos dimensiones puede ser visto como un arreglo de arreglos de una dimensión, cada entrada del arreglo contiene un elemento de un tipo especificado. El "m+1" es el número de arreglos unidimensionales que tiene la variable, y el "n+1" es el número de elementos (en este caso de tipo doublé) que tiene cada arreglo unidimensional.

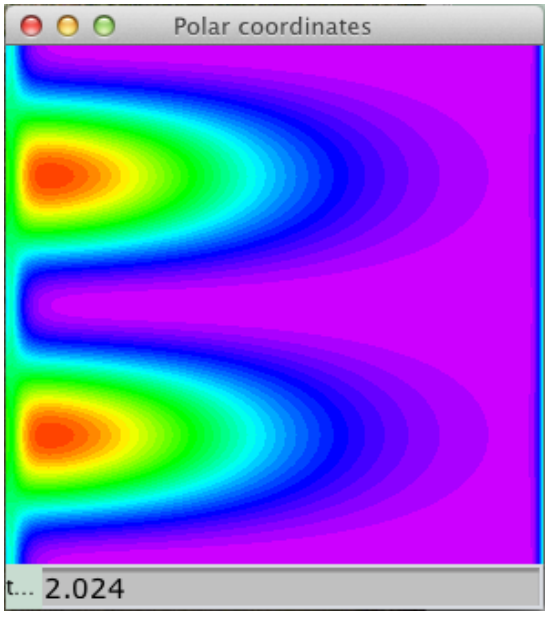

FIGURA 3. Segunda ventana de la simulación "Polar coordinates" (recinto rectangular de la simulación).

Por ejemplo, la variable no. 15 de la lista: "u1" representa un arreglo bidimensional, que contiene $(m+1)\times(n+1)$ elementos que son números de doble precisión en Java. En este caso, estos números representan un arreglo rectangular

de temperaturas según la ecuación diferencial de calor (2) para un instante de tiempo $t + dt$, posterior al instante t.

Esta variable la vamos a utilizar posteriormente cuando editamos la fase correspondiente a la "Evolución" de la simulación.

VII. LA INICIALIZACIÓN

En esta etapa del "Modelo" consta de cuatro etapas o pasos. En el primer paso colocamos la condición de estabilidad de la simulación, si ésta no se cumple, el programa despliega un mensaje de precaución o advertencia en una ventana. La condición de estabilidad ya la hemos mostrado en la inecuación (2). En la segunda etapa, inicializamos el tiempo a cero ($t = 0.0$). En la tercera etapa, cargamos los valores de temperatura inicial z_{ij}, de acuerdo a la Ec. (7), de manera que la función "$f = f(r_i, \theta_j) = f(r[i], \theta[j])$" que aparece en la declaración

$$z[i][j][0] = f(r[i], theta[j]). \quad (9)$$

Es una de las funciones "propias" de la simulación (éstas funciones aparecen en la Fig. 8).

Por último, en la cuarta etapa, corregimos fases de $Theta[p][q]$, que sean negativas. Todas estas acciones de inicialización la mostramos en la Fig. 5, para una fácil referencia del lector.

VIII. LA EVOLUCIÓN

Esta es la parte principal de la simulación, pues contiene en sí la ecuación diferencial de calor en coordenadas polares. Y también consta de varias etapas.

Si t es cero se carga en u la distribución inicial de temperaturas guardadas en z. Si t es diferente de cero, entonces se calcula la temperatura u de cada punto del recinto circular, de acuerdo a la ecuación de calor.

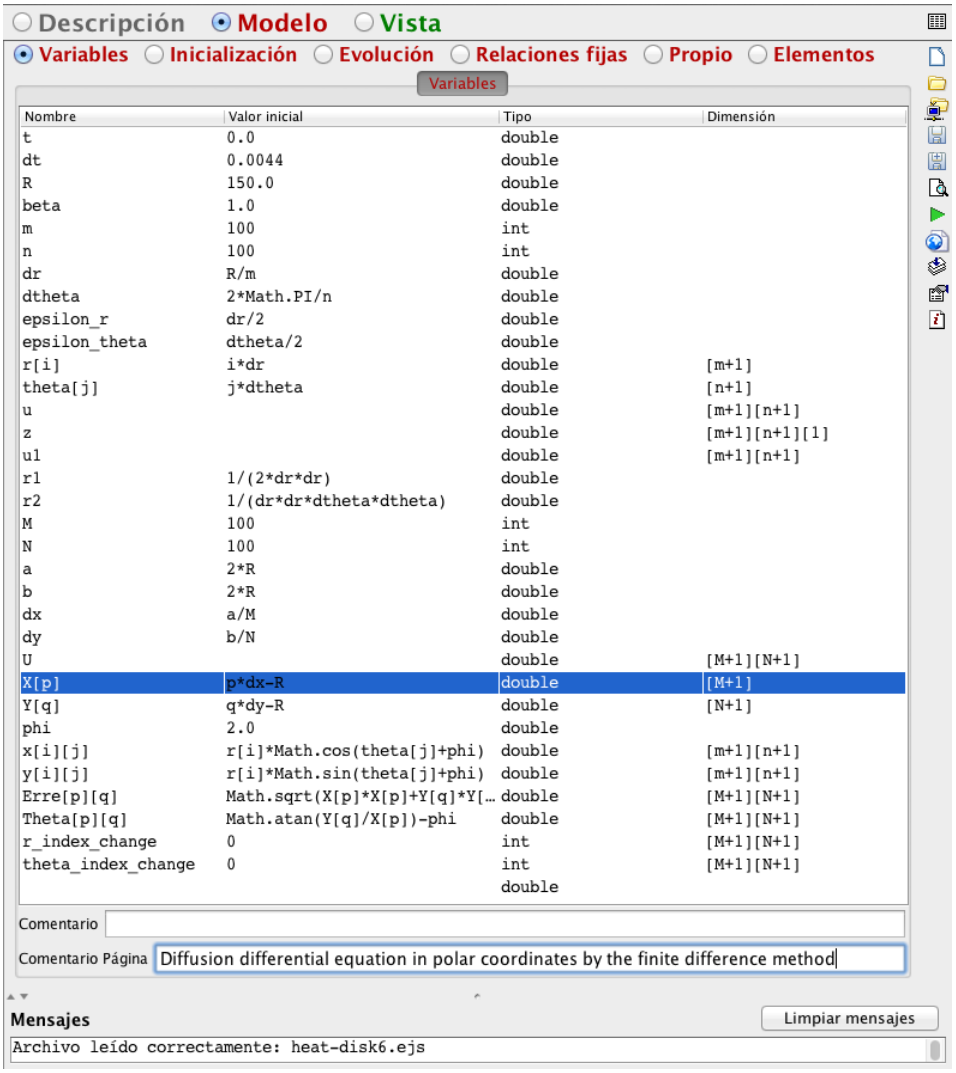

FIGURA 4. Lista de las variables empleadas en esta simulación, de la ecuación de calor en coordenadas polares.

Por último se hace avanzar el tiempo, y se actualizan los valores de las temperaturas *u*. La edición de la evolución la mostramos en la Fig. 6.

Vale la pena mencionar que aquí se emplea una función propia llamada "derivada2", con tres argumentos. Esta función se emplea para encontrar la segunda derivada de la distribución de temperaturas "u", de acuerdo al segundo miembro de la ecuación de calor (2), siguiendo las aproximaciones numéricas dadas por las relaciones (3) y (5). También utilizamos la aproximación numérica dada en la Ec. (6) para la primera derivada parcial de "u" con respecto al tiempo "t".

Al juntar todos estos elementos matemáticos y trabajarlos, encontramos que la Ec. (2) se transforma en la siguiente expresión (véase la Fig. 6):

$$\frac{u1[i][j]-u[i][j]}{dt} = \beta * derivada2(i, j, u). \qquad (10)$$

Donde la función "derivada2" adquiere la siguiente forma (última expresión de la función "derivada2" en la Fig. 8):

$$derivada2(i, j, u) = \frac{1}{2i(dr)^2}\left[(2i+1)u_{i+1\,j} - 4iu_{ij} + (2i-1)u_{i-1\,j}\right] + \frac{1}{i^2(dr)^2(d\theta)^2}\left[u_{i\,j-1} - 2u_{ij} + u_{i\,j+1}\right]. \qquad (11)$$

Para el caso en que los índices i y j satisfagan las desigualdades siguientes:

$$i \neq 0, m; \quad j \neq 0, n. \qquad (12)$$

Esta relación (11) se satisface para la mayoría de los puntos de la placa circular que no se encuentran sobre las fronteras de la placa (o en fronteras matemáticas), tanto para la coordenada radial

$$r_i = r[i] \neq 0, R \quad \Leftrightarrow \quad i \neq 0, m. \qquad (13)$$

Como para la coordenada angular

$$\theta_j = \theta[j] \neq 0, 2\pi \quad \Leftrightarrow \quad j \neq 0, n. \qquad (14)$$

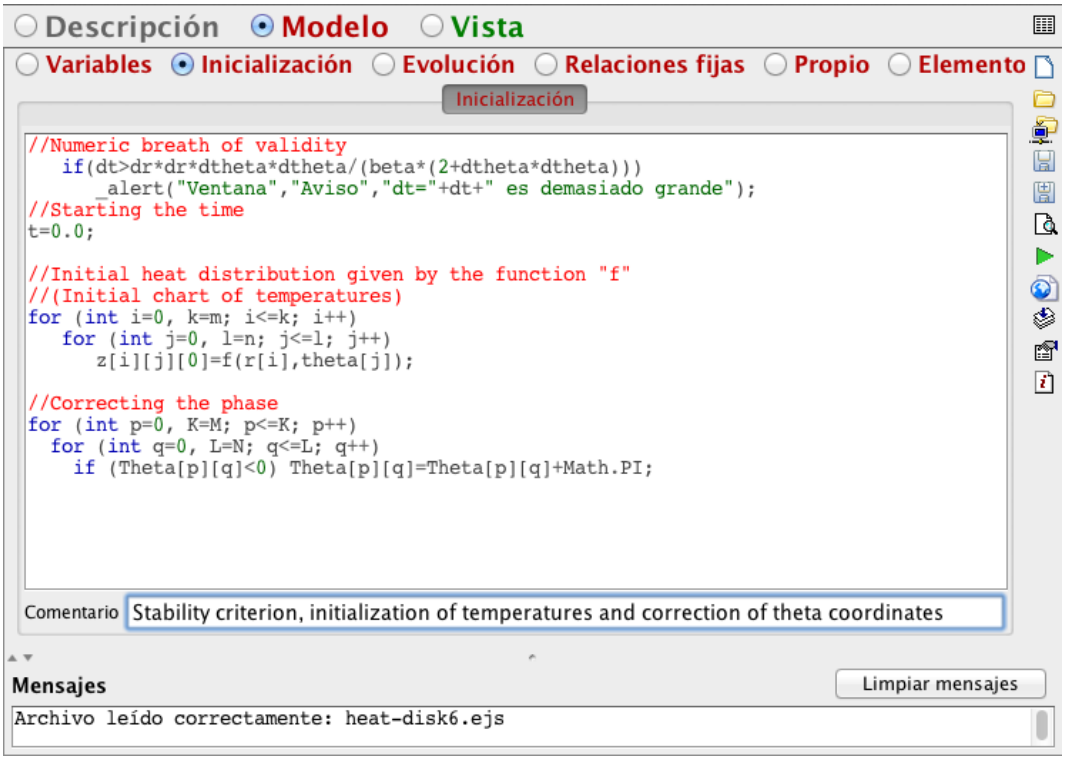

FIGURA 5. Inicialización de la distribución de temperaturas del disco conductor de calor.

Existen otras identidades para la función "derivada2", que se cumplen cuando se calcula la temperatura "u1"para un punto que se encuentre sobre la frontera radial $r[i] = 0, R \quad \Leftrightarrow \quad i = 0, m.$ O bien sobre la frontera angular $\theta[j] = 0, 2\pi \quad \Leftrightarrow \quad j = 0, m.$ Estas fórmulas se muestran ya

editadas en la Fig. 8, después del comentario en Java que versa: "//2nd derivative function in polar coordinates".
Y son cuatro expresiones matemáticas que corresponden a los casos en que "$i=0$", "$i=m$", "$j=0$" y "$j=m$", respectivamente. La formula (11), que acabamos de ver correspondería a la última expresión enlistada en este código en Java.

IX. LAS RELACIONES FIJAS

En esta parte de la construcción de la simulación, se da el cambio de coordenadas rectangulares a polares. Se emplean intrínsecamente las relaciones:

$$Erre[p][q]=\sqrt{\left(X[p]\right)^2+\left(Y[q]\right)^2},$$
$$Theta[p][q]=\arctan\left(\frac{Y[q]}{X[p]}\right)-phi. \qquad (15)$$

Que se dan en la sección de las variables. Aquí vale la pena identificar a las variables.

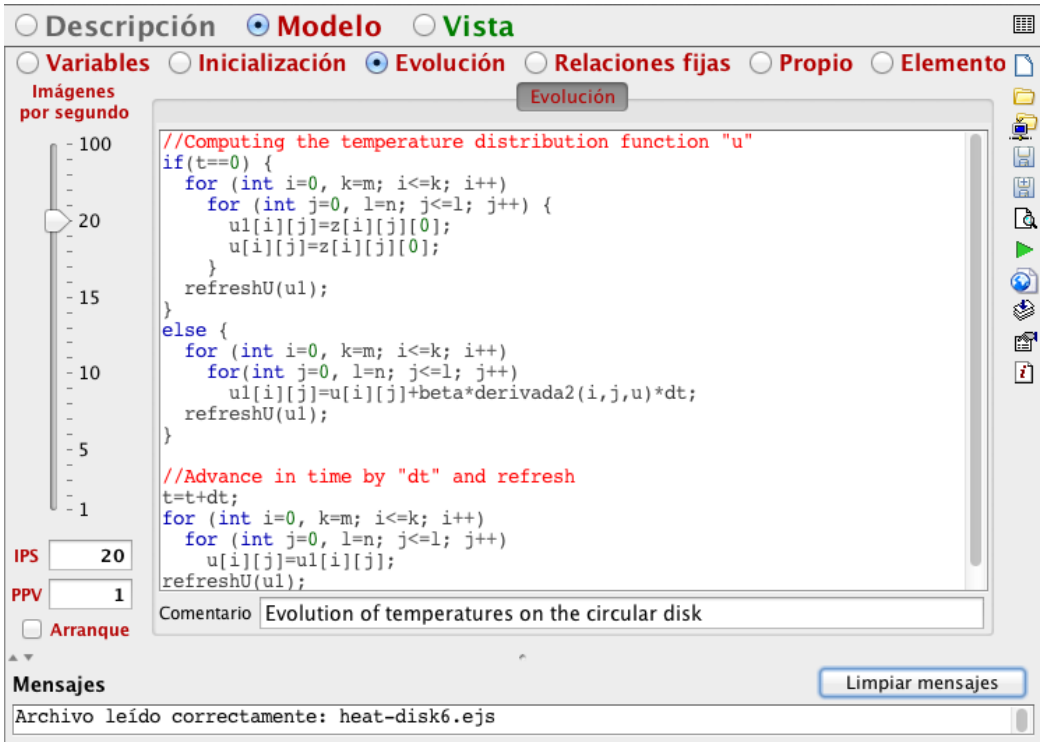

FIGURA 6. Edición de la evolución de la distribución de calor para una placa circular.

Cuando la simulación está corriendo se abren dos ventanas. En la primera, mostrada en la Fig. 2 y llamada "Circle", se hallan dos ejes coordenados rectangulares que son $X[p]$ y $Y[q]$. Sobre este recinto rectangular se efectúa el cambio de coordenadas de rectangulares a polares. Las coordenadas polares de este recinto de la ventana "Circle" se calculan como $Erre[p][q]$ y $Theta[p][q]$. Hay que mencionar que en esta ventana se manejan los índices p y q como convención, para identificar las coordenadas, manejadas aquí.

En la segunda ventana que se abre cuando la simulación corre (llamada "Polar coordinates", véase la Fig. 3), se muestra también una zona rectangular con distribuciones de temperaturas. Aquí el eje horizontal se identifica con la variable $r[i]$ y el eje rectangular vertical se identifica con $\theta[j]$. En esta ventana los índices manejados por convención, son i y j.

La finalidad de las relaciones fijas en esta simulación es lograr un cambio de coordenadas, efectuando para ello, un "cambio de índices". De esta forma, existe un mapeo de los índices i y j, a los índices p y q, dado en forma natural por el cambio de coordenadas. Para esto, se comparan los valores de $Erre[p][q]$ y $Theta[p][q]$ de la primera ventana "Circle", con los valores respectivos $r[i]$ y $\theta[j]$ de la segunda ventana "Polar coordinates". Los valores respectivos no deben de exceder en valor absoluto a ciertos números precalculados llamados respectivamente "*epsilon_r*" y "*epsilon_theta*". De tal forma que se cumplen las desigualdades

$$\left|Erre[p][q] - r[i]\right| < epsilon\_r, \tag{16}$$

y

$$\left|Theta[p][q] - \theta[j]\right| = \left|Theta[p][q] - theta[j]\right| < epsilon\_theta \tag{17}$$

Estos valores de "epsilon" se toman como la mitad de los pasos dr y $dtheta = d\theta$, de la ventana "Polar coordinates".

$$epsilon\_r = dr/2, \quad epsilon\_theta = dtheta/2. \tag{18}$$

Cuando la desigualdad (16), se cumple para ciertos valores p, q, i, entonces formamos una función de índice en p y q llamado *r_index_change* y aquí grabamos el valor de i, que corresponde a los índices p y q, para la coordenada r.

$$r\_index\_change[p][q] = i. \tag{19}$$

Y similarmente procedemos igual con la coordenada $theta = \theta$:

$$theta\_index\_change[p][q] = j. \tag{20}$$

Estos cambios de índice se emplean, posteriormente para "mapear" el recinto rectangular de temperaturas $u_{ij} = u[i][j]$ con las coordenadas polares $r[i]$, $\theta[j]$ de la segunda ventana de la simulación "Polar coordinates", hacia las temperaturas $U_{pq} = U[p][q]$ del recinto de la primera ventana "Circular". La manera explícita de lograr este mapeo es mediante la asignación

$$U[p][q]=u[r\_index\_change[p][q]][theta\_index\_change[p][q]]. \tag{21}$$

Que se pone en la sección "Propio" del "Modelo" de EJS, y que se usa en la etapa de la "Evolución".
Las "Relaciones fijas" se muestran en la Fig. 7.

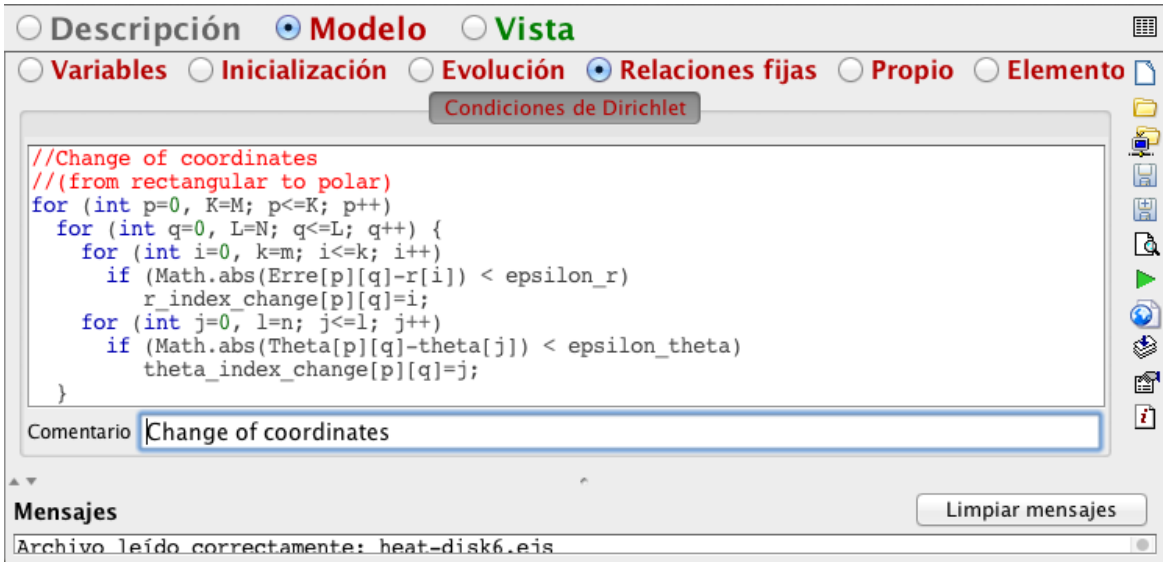

FIGURA 7. Las Relaciones fijas, que emplea la simulación para hacer el cambio de coordenadas.

X. FUNCIONES PROPIAS

La parte final, del "Modelo" de esta simulación consiste en elegir el subradio-botón "Propio", y en esta ventana anotar todas las funciones auxiliares que se utilizaron en el desarrollo del Modelo. Por ejemplo aquí se definen las derivadas, de la ecuación diferencial de calor en coordenadas polares, la función que da la distribución inicial de temperatura del recinto circular, la condición de frontera de Dirichlet (para mantener el borde del disco a temperatura 1.0), y la función de mapeo de temperaturas $u_{ij} = u[i][j]$ a temperaturas $U_{pq} = U[p][q]$, mostradas en la Ec. (21).

La Fig. 8 muestra como editar esta ventana del Modelo.

La ventana está editada con código Java. Primero notamos los comentarios "//Initial distribution function of temperature", y se muestra entonces la función de distribución inicial de temperaturas para la placa circular escrita en la Ec. (7). Luego se encuentra una condición de frontera tipo Dirichlet, para el borde del disco, hemos elegido la temperatura máxima del rango (1.0), para el borde. De esta forma, conforme pasa el tiempo, el disco se va calentando poco a poco hasta alcanzar esta temperatura

límite. Además durante las primeras etapas de su evolución, el calor de la distribución inicial, se va dispersando a través de todo el disco. Es interesante observar como algunas regiones del disco se van enfriando, y otras se van calentando. Se espera que el despliegue de colores de este ejemplo sea atractivo para los estudiantes.

Después sigue la edición de la segunda derivada parcial en dos dimensiones y en coordenadas polares para cada región circular. Esta derivada (salvo por un múltiplo de β) corresponde al miembro del lado derecho de la ecuación de calor (2). Las regiones que definen las fronteras de la región circular y las propias coordenadas polares, por un lado, y las regiones internas del disco, por otro lado, tienen una expresión diferente para está derivada. Las expresiones están separadas mediante varias clausulas "if", en el código.

Finalmente, se muestra la manera de actualizar la distribución de temperaturas "U" (de la Fig. 2) para el disco conductor de calor, utilizando la distribución de calor "u" (de la Fig. 3).

También se aproxima el cambio de coordenadas de rectangulares a polares, con un cambio de índices. Éste cambio de índices se logra mediante las dos funciones enteras de índice llamadas $r\_index\_change[p][q]$ y $theta\_index\_change[p][q]$ ya definidas por las condiciones mostradas por el código de la Fig. 7, y por las Ecs. (16), (17), (18), (19) y (20) de este escrito.

XI. LA VISTA

La "Vista" de una simulación se elige al seleccionar el último radio botón de la parte superior de la ventana de la interfaz de EJS. Consiste en crear un árbol de elementos gráficos a partir de un icono de la sección izquierda de esta ventana llamada "Vista de la simulación", que viene a ser la raíz del árbol. En la sección derecha se agrupan varios subpaneles con diferentes elementos gráficos, como son ventanas, ejes coordenados, trazas, y campos escalares bidimensionales, entre otros.

Para mayor claridad, en la Fig. 9 mostramos el árbol ya construido, con los elementos adecuados a esta simulación.

Como se puede observar, aquí se encuentran las dos ventanas de la simulación llamadas "Circle" y "Polar coordinates" mostradas como dos iconos que son los elementos gráficos "hijos" de la raíz genérica "Vista de la simulación". El primer icono, denota una ventana principal para EJS, y el segundo icono denota una ventana normal. La diferencia entre la ventana principal y las ventanas normales, radica en que sólo existe una ventana principal en cada simulación de EJS, pero puede haber cero, una o varias ventanas normales dentro de la misma simulación. Otra diferencia consiste en que si el usuario cierra la ventana principal, entonces toda la simulación se interrumpe. Pero si el usuario cierra una ventana normal o secundaria, entonces sólo esa ventana se cierra, y el resto de las ventanas de la simulación permanecen abiertas.

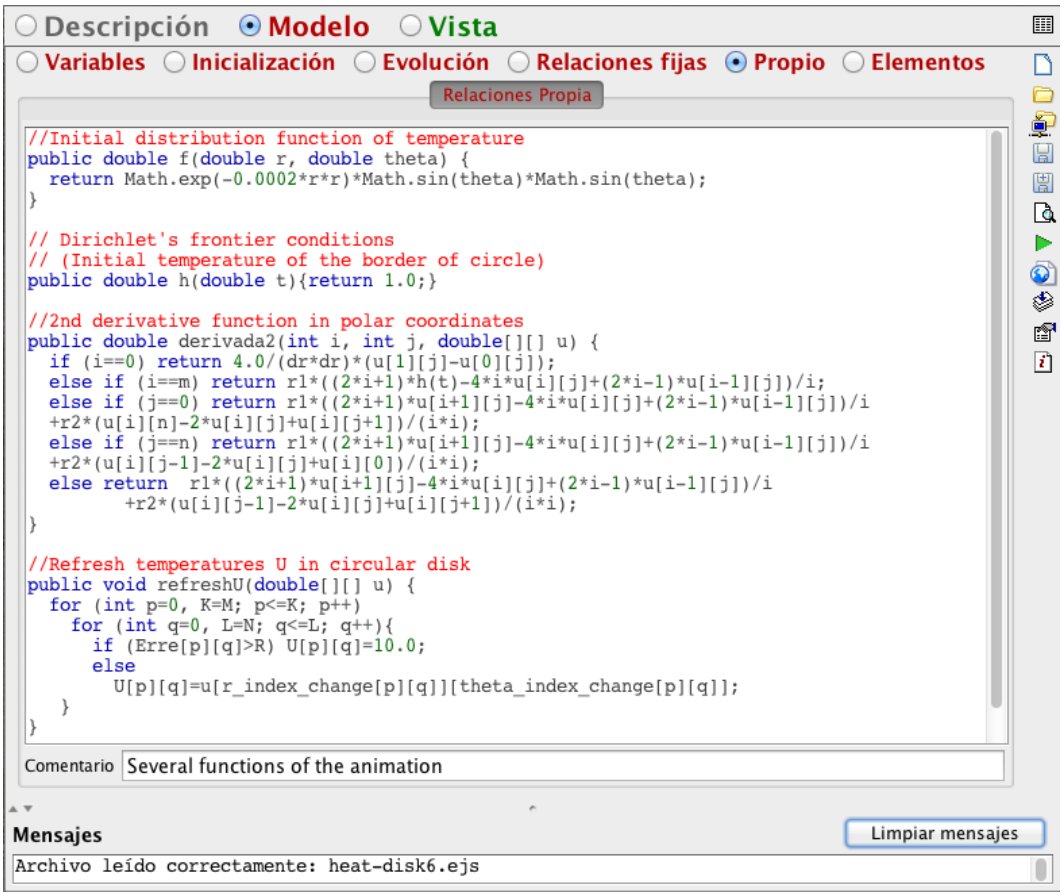

FIGURA 8. Las Funciones Propias, que emplea la simulación.

Francisco Esquembre [8], detalla estas características de las ventanas en su libro *Creación de Simulaciones Interactivas en Java: Aplicación a la Enseñanza de la Física*. Que es una guía completa para empezar a utilizar y editar simulaciones en Java usando la herramienta de software EJS [7], que él mismo ha diseñado para este fin. Nótense en la Fig. 9 a dos elementos gráficos fundamentales para la simulación llamados "CircularHeat" y "RectangularHeat" que son campos escalares 2D y que se encuentran dentro de contenedores 2D, dentro de sus respectivas ventanas. Para editar las propiedades de estos elementos y ligarlos con las variables del Modelo, es necesario hacer clic derecho en cada elemento, y seleccionar del menú contextual que emerge la opción "Propiedades". Una vez hecho esto se muestra una ventana de las "propiedades de ese elemento gráfico en particular.

Es tarea de nosotros editar los campos de estas ventanas de propiedades, para lograr el correcto funcionamiento de la simulación. Para ilustrar esto, mostramos las ventanas de "propiedades" para estos dos campos y sus respectivos contenedores 2D, en las Figs. 10, 11, 12, 13.

En la Fig. 10 se muestran las propiedades del contenedor 2D para el campo escalar "CircularHeat". Éstos elementos gráficos se encuentran dentro de la primer ventana de la simulación en ejecución, y llamada por nosotros "Circle". La ventana "Circle", se mostró en la Fig. 2. Y que representa al disco circular conductor con su distribución de temperaturas $U_{pq} = U[p][q]$, a un tiempo *t*.

La ventana de propiedades para este elemento gráfico la podemos describir enlistando sólo los campos que vamos rellenando, para este ejemplo. Así, tenemos las siguientes propiedades para "CircularDashBoard":

Propiedades de CircularDashboard (PanelDibujo):

1. Autoscala X: false
2. Autoscala Y: false
3. Mínimo X: 0
4. Máximo X: 2*R
5. Mínimo Y: 0
6. Máximo Y: 2*R
7. Cuadrado: true

Los demás campos no rellenos, adquieren valores por default.

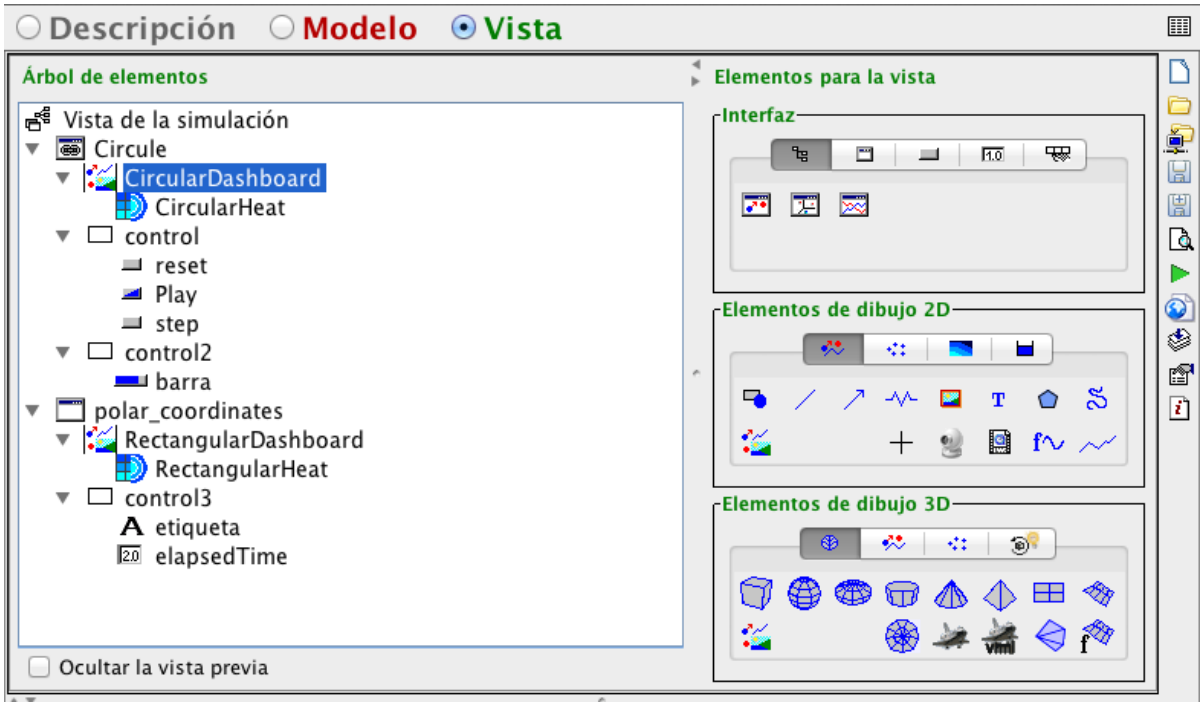

FIGURA 9. El árbol de elementos gráficos para la simulación.

En la Fig. 11 mostramos la ventana de propiedades para el campo escalar 2D de temperaturas $U_{pq} = U[p][q]$. Su lista de propiedades será entonces:

Propiedades de CircularHeat (CampoEscalar2D):

1. Datos Z: "U"
2. Autoscala Z: false
3. Mínimo Z: 0.0
4. Máximo Z: 1.0
5. Mínimo X: 0
6. Máximo X: 2*R
7. Mínimo Y: 0
8. Máximo Y: 2*R
9. Tipo de Gráfico: INTERPOLATED
10. Niveles: 37

Aquí notamos que estos valores quedan representados mediante colores para el campo escalar ligado con la variable de doble precisión "U" y son lo que es llamado por EJS "Datos Z". El rango de valores para Z, representa la escala de temperaturas que tiene el disco circular, y esto determina como se van distribuyendo los colores en el disco. El número de colores se controla con el valor entero introducido en el campo "Niveles". En cuanto a los rangos para X y Y, se les dan valores correspondientes a las coordenadas rectangulares del disco circular mostrado en la Fig. 2.

En la Fig. 12, a su vez mostramos el contenedor 2D, que se encuentra en la segunda ventana de la simulación, que hemos nombrado "Polar coordinates", y que se mostró en la Fig. 3. Estas propiedades son importantes pues delimitan al rango de valores para las variables $r[i]$ y $\theta[j]$. El rango de la variable discreta $r[i]$ es de 0.0 a R; mientras que el rango para $\theta[j]$ es de 0.0 a 2π.

Las propiedades que se editaron para el elemento gráfico "RectangularDashboard" (véase la Fig. 9 y la 12) son:

Propiedades de RectangularDashboard (PanelDibujo):

1. Autoscala X: false
2. Autoscala Y: false
3. Mínimo X: 0.0
4. Máximo X: R
5. Mínimo Y: 0.0
6. Máximo Y: 2*3.1416

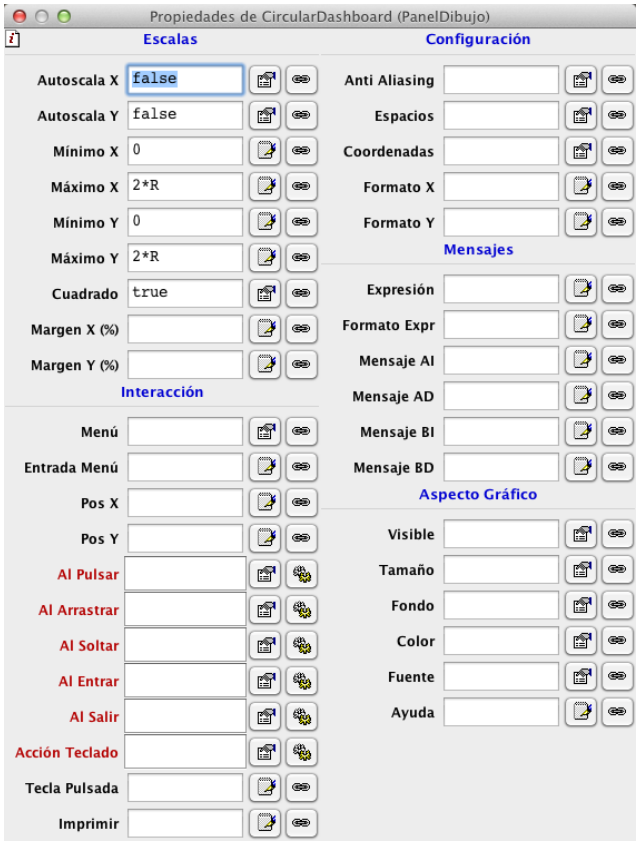

FIGURA 10. Propiedades del contenedor gráfico 2D llamado "CircularDashboard", para el campo escalar 2D "CircularHeat".

En la Fig. 13, se muestran las propiedades para el campo escalar 2D de temperaturas $u_{ij} = u[i][j]$, correspondientes a las coordenadas polares $r[i]$ y $\theta[j]$. Esta distribución se muestra en la segunda ventana de la simulación llamada "Polar coordinates", de la Fig. 3. El enlistado de propiedades que se editaron para el elemento gráfico respectivo llamado en este ejemplo "RectangularHeat".

Propiedades de RectangularHeat (CampoEscalar2D):

1. Datos Z: "u"
2. Autoscala Z: false
3. Mínimo Z: 0.0
4. Máximo Z: 1.0
5. Mínimo X: 0.0
6. Máximo X: R
7. Mínimo Y: 0.0
8. Máximo Y: 2*3.1416
9. Tipo de Gráfico: INTERPOLATED
10. Niveles: 37

Hasta aquí, hemos descrito las propiedades de cuatro de los elementos gráficos, mostrados en el árbol de elementos de la Fig. 9. Otros elementos gráficos se encargan de controlar la simulación, y de mostrar el tiempo transcurrido. En seguida, se muestran los enlistados de varios elementos clave de la Fig. 9.

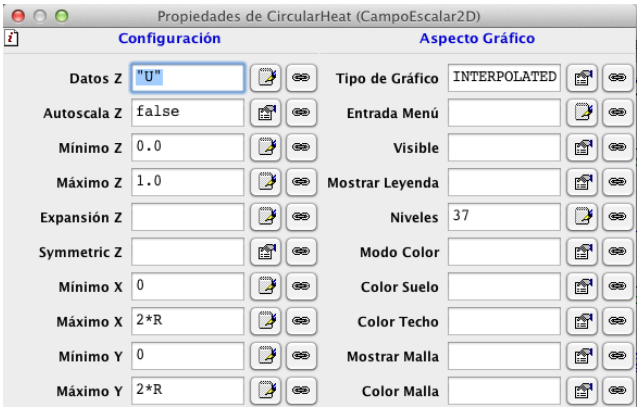

FIGURA 11. Propiedades del campo escalar 2D de temperaturas *U*, llamado "CircularHeat" y mostrado en la primer ventana de la simulación de la Fig. 2.

Por ejemplo, para crear un contenedor gráfico que puede tener a otros elementos gráficos, se utilizan los llamados "paneles". En la Fig. 9, se nota que la presentación de la "Vista" para este ejemplo consta de tres paneles denominados "control", "control2" y "control3". Para ejemplificar como se pueden editar las propiedades de estos paneles, enlistamos las propiedades editadas del panel "control":

Propiedades de control (Panel):

1. Distribución: border

La única propiedad que necesita especificarse sería la relacionada a la distribución que tendrán los elementos gráficos contenidos en este panel. La distribución "border" (la configuración por default), puede contener a tres elementos en forma horizontal. Las posiciones de los elementos "reset", "Play" y "step" serán sucesivamente "Izquierda", "Centro" y "Derecha" respectivamente y existen otras dos posiciones no ocupadas: "Arriba" y "Abajo". Dependiendo del aspecto gráfico final, hay que elegir una distribución específica. Si por ejemplo, necesitáramos más de tres elementos dispuestos en un mismo renglón horizontal, entonces necesitaríamos elegir otra distribución para el panel. Por ejemplo se podría escoger la distribución de "Caja horizontal", que aparece en una ventana auxiliar, cuando el usuario oprime el botón de propiedades pre-construidas de EJS, que se encuentra a la derecha del campo "Distribución", de la ventana de propiedades para este "Panel".

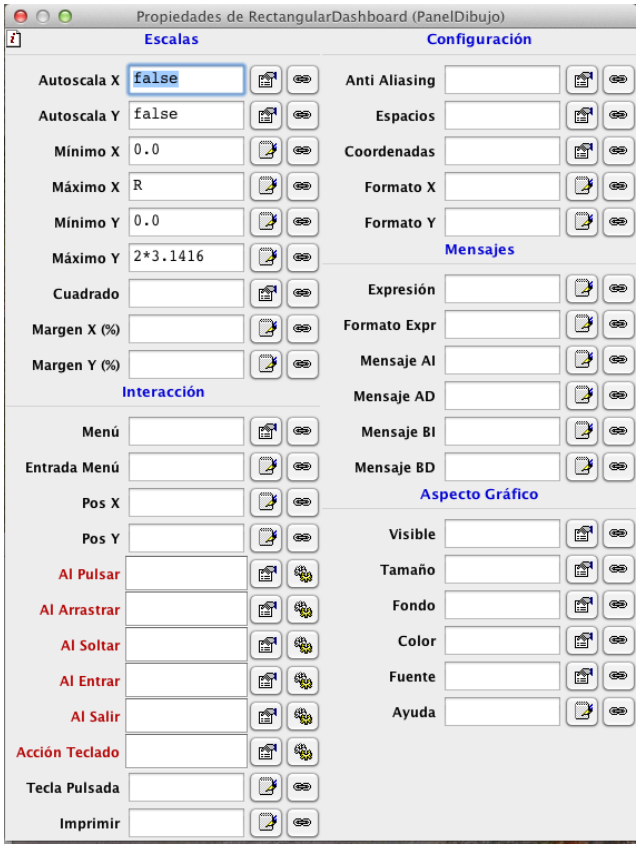

FIGURA 12. Propiedades del contenedor gráfico 2D llamado "RectangularDashboard", para el campo escalar 2D de nombre "RectangularHeat". Se muestran el rango de valores de las coordenadas polares para *r*, en los campos Mínimo X y Máximo X, y para *theta* = θ en los campos Mínimo Y y Máximo Y.

Dentro del panel "control" existen otros tres elementos que controlan la corrida de la simulación. A continuación enlistamos sus propiedades, para ilustrar como se editan los campos respectivos para cada uno de estos elementos.
El enlistado para el botón "reset" es el siguiente:

Propiedades de reset (Boton):

1. Texto: "reset"
2. Imagen: "/org/opensourcephysics/resources/controls/images/reset2.gif"
3. Acción: _reset()

La función predefinida de EJS "_reset()" ocasiona que la simulación regrese a su estado inicial, una vez que el usuario, pulsa sobre el botón "reset". La imagen, para el botón se obtiene de una ventana auxiliar, que aparece al oprimir el botón propiedades situado a la derecha del campo "Imagen".

Para el botón de dos estados "Play" tenemos la siguiente lista de propiedades:

Propiedades de Play (BotonDosEstados)

1. Variable: _isPaused
2. Texto Sí: "Play"
3. Imagen Si: "/org/opensourcephysics/resources/controls/images/play.gif"
4. Acción Si: _play()
5. Texto No: "Pause"
6. Imagen No: "/org/opensourcephysics/resources/controls/images/pause.gif"
7. Acción No: _pause()

Para el botón "step" tenemos la siguiente lista de propiedades:

Propiedades de step (Boton):

1. Texto: "step"
2. Imagen: "/org/opensourcephysics/resources/controls/images/stepforward.gif"
3. Acción: _step()

Este botón causa que la simulación avance un paso (correspondiente al intervalo de tiempo *dt*), en su evolución.

FIGURA 13. Propiedades del campo escalar 2D de temperaturas *u*, llamado "RectangularHeat" y mostrado en la segunda ventana de la simulación de la Fig. 3.

En la ventana "Circle", dentro del panel "control2", implementamos una barra de avance, sus propiedades las enlistamos en seguida:

Propiedades de barra:

1. Variable: t
2. Mínimo: 0.0
3. Máximo: 100.0
4. Tamaño: "200.25"
5. Fondo: LIGHTGRAY
6. Color: 200,220,208

Otro detalle gráfico, que sirve para mostrar el valor del tiempo transcurrido en forma numérica, involucra a otros dos elementos gráficos del árbol principal. En la segunda ventana "Polar coordinates", dentro del panel "control3", tenemos los dos elementos "etiqueta" y "elapsedTime".

Las propiedades del elemento "etiqueta" sólo son para mostrar un letrero que anuncia que en el siguiente campo numérico se despliega el valor del tiempo transcurrido. Aquí tenemos su enlistado:

Propiedades de etiqueta (Etiqueta):

1. Texto: " t ="
2. Tamaño: "20,25"
3. Fondo: 200,220,208
4. Color: BLACK

Y las propiedades editadas del campo numérico "elapsedTime" son las siguientes:

Propiedades de elapsedTime (CampoNumerico):

1. Variable: t
2. Editable: true
3. Visible: true
4. Tamaño: “280,25”
5. Fondo: LIGHTGRAY
6. Color: BLACK
7. Fuente: MS Reference Sans Serif, PLAIN, 15

Hemos descrito, todos los elementos gráficos, del árbol principal de la “Vista”, para poder terminar con la edición de la simulación. En la siguiente sección se describe brevemente como lograr un archivo autoejecutable, que puede mostrarse a una audiencia de estudiantes. El primer impacto que tengan ellos, puede ser determinante en el desarrollo posterior de la clase de Física.

XII. CORRIDA Y EMPAQUETADO

Cuando finalizamos de editar la simulación la podemos correr para probarla, oprimiendo el botón de “play” o “Ejecutar la simulación” ▶ que se encuentra en el menú del borde derecho de la ventana principal o de interfaz de EJS. Ya se han mostrado las figuras que muestran a esta simulación para el disco conductor de calor en las Figs. 2 y 3. La Fig. 2 muestra el recinto circular y la distribución de temperaturas manejada con colores. La Fig. 3, muestra esta misma distribución pero para las coordenadas polares r y θ, antes de hacer el cambio de variable.

Para elaborar el empaquetado de un archivo autoejecutable “ejs” de Java, sólo hace falta oprimir el botón “Empaquetar simulación actual” del menú del borde derecho de la misma ventana anterior.

XIII. IMPACTO EDUCATIVO

Hablar sobre el impacto que tienen las simulaciones de eventos físicos dentro del área educativa, es hablar de un mundo muy extenso y rico en posibilidades. Según Clark Aldrich [2], existen cuatro géneros básicos de simulación. Estos son:

1. Historias Ramificadas (Branching Stories).
2. Hojas de Trabajo Interactivas (Interactive Spreadsheets).
3. Modelos basados en Juegos (Game-Based Models)
4. Productos Virtuales y Laboratorios Virtuales (Virtual Products and Virtual Labs).

Nuestro tipo de simulación entraría dentro de la categoría de Productos Virtuales y Laboratorios Virtuales.

Como ya sabemos, hemos tratado de imitar el comportamiento real que tendría el flujo de calor en un disco conductor, que está sobre un depósito externo de calor, y que tiene una distribución inicial de temperaturas (expresada gráficamente en la Fig. 2, para el tiempo $t = 2.024\,seg$). Éste tipo de simulaciones, pretenden auxiliar el aprendizaje de conceptos básicos, dentro de la física de los objetos involucrados en la simulación. Aunque no son propiamente “Modelos basados en Juegos”, pretendemos que el estudiante encuentre interesante el tema y se divierta, modificando las condiciones de frontera del problema.

En este caso, pretendemos sugerir una serie de pasos recomendados para el instructor de la clase de Física. El posible escenario didáctico podría ser como sigue: Primero, el instructor da una breve introducción a la ecuación de calor en coordenadas polares.

Después de explicar qué es la ecuación de calor y cuál es su uso, el instructor puede mostrar la simulación a sus estudiantes. En seguida, mediante su supervisión les sugiere que ellos formen equipos (que cuenten con al menos una computadora). A continuación, cada equipo modifica las condiciones de distribución inicial de temperaturas del disco, o en su caso, pueden modificar sus condiciones de frontera. De esta manera, el alumno puede ejercer su creatividad utilizando la simulación como material didáctico, y disfrutará más de la clase de Física.

El objeto de todo juego educativo, es precisamente utilizar la diversión mediante el juego para ayudar al estudiante a aprender. Aunque la simulación está más orientada, a construir un ambiente de Laboratorio Virtual, no hay que menospreciar su potencial lúdico. De hecho, la idea de aprender divirtiéndose, está vinculada con el aprendizaje conducido por la interactividad del alumno.

Aldrich [2], comenta que para tener una buena experiencia educacional, hay que reunir tres elementos esenciales en nuestra herramienta didáctica. Estos elementos esenciales los identifica de la siguiente forma:

1. Simulaciones
2. Juegos
3. Pedagogía

El investigador comenta que nosotros frecuentemente confundimos las partes de la Simulación y el Juego, porque las simulaciones tienen muchos de los elementos que hallamos en los juegos. Sin embargo, es mejor separarlos, para tener un esquema más claro.

En cuanto a la Pedagogía, se espera que las simulaciones estén enfocadas hacia el aspecto productivo de los alumnos, y que promuevan su aprendizaje. De esta manera, se logra una mejor eficiencia en el proceso educativo.

Hay que comentar que el proceso de enseñanza aprendizaje en el área de la Física (que es el área que nos interesa, en este trabajo), tendrá que ser constantemente apoyado por los esfuerzos del docente. El docente, tendrá que aportar nuevo material interactivo, como el que ofrecimos en este artículo. Esto, agregado al material didáctico convencional, deberá de formar un apoyo firme en la enseñanza de la Física.

Desde luego, que la parte fundamental dentro de este proceso de aprendizaje, la tiene el alumno, que al momento de interaccionar con las simulaciones, puede mejorar sus conocimientos en el área de estudio. Las simulaciones de los fenómenos físicos, pueden prepararlo para las situaciones que tiene que resolver en los Laboratorios de Física reales, mejorando de esta manera su rendimiento. Y este hecho, a su vez, le forjará unos conocimientos básicos más sólidos para sus demás materias. Creando un individuo

más productivo y más capacitado, para las situaciones reales que deberá enfrentar, cuando ejerza los conocimientos de su carrera, ya como un profesionista.

De hecho, el mundo de las simulaciones, tiene un panorama más amplio, y constituye una rama de estudio con muchos alcances en más áreas del conocimiento. Aquí hemos sólo tocado, un aspecto muy sencillo del tema. Muchos investigadores intentamos dar un aspecto claro, de cómo estas herramientas pueden ser aplicadas en beneficio de la educación. Y así como la misma tecnología, tiene avances constantes que nos maravillan en cada momento, de la misma forma, la aplicación de las simulaciones virtuales en el área educativa, tiene que ir evolucionando positivamente, para mejorar sus perspectivas futuras.

XIV. CONCLUSIONES

Una vez obtenida la simulación para un disco conductor de calor, contamos con una herramienta valiosa para hacer la clase de Física más interesante. Se espera que la simulación apoye al profesor cuando éste vea este tema de la Termodinámica. Posteriormente se espera que el instructor promueva entre los alumnos mayor participación y entusiasmo por la clase.

Easy Java Simulations permite la edición de varios de los parámetros que controlan la simulación, como son, por ejemplo, las condiciones iniciales para el problema, y la distribución inicial de temperaturas, entre otros. Los estudiantes pueden mostrar su creatividad mediante una serie de experimentos virtuales guiados por el instructor. Y como vemos, el empleo de simulaciones en Java de los fenómenos físicos promueve a que la clase de Física sea una actividad más interactiva para los alumnos y el instructor, mejorando y facilitando la asimilación de los conceptos del fenómeno estudiado.

AGRADECIMIENTOS

El autor quiere dar su agradecimiento al apoyo otorgado por la COFAA del Instituto Politécnico Nacional, al proyecto SIP-20113327 de la Secretaría de Investigación y Posgrado del IPN y al Consejo Nacional de Ciencia y Tecnología (CONACyT) de México, D.F.

REFERENCIAS

[1] Aldrich, C., *Learning Online with Games, Simulations, and Virtual Worlds: Strategies for Online Instruction*, (Jossey-Bass, San Francisco, 2009).
[2] Aldrich, C., *Learning by Doing: A Comprehensive Guide to Simulations, Computer Games, and Pedagogy in e-Learning and Other Educational Experiences*, (Pfeiffer, San Francisco, 2005).
[3] Quinn, C. N., *Engaging Learning: Designing e-Learning Simulation Games*, (Pfeiffer, San Francisco, 2005).
[4] Alger, C. F., *Harold Guetzkow: A Scholar's Scholar*, Simulation & Gaming **42**, 295-300 (2011).
[5] Takatalo, J., Häkkinen, J., Kaistinen, J. and Nyman, G., *User Experience in Digital Games: Differences Between Laboratory and Home*, Simulation & Gaming **42**, 656-673 (2011).
[6] Oracle, <http://java.com/es/download>, consultado el 25 de Septiembre de 2011.
[7] Francisco, E., *Easy Java Simulations (EJS)*, <http://fem.um.es/Ejs>, consultado el 25 de Septiembre de 2011.
[8] Francisco, E., *Creación de Simulaciones Interactivas en Java: Aplicación a la Enseñanza de la Física*, (Pearson, Prentice-Hall, Madrid, 2005).
[9] Mathews, J. H. and Fink, K. D., *Métodos Numéricos con MATLAB*, (Prentice-Hall, Madrid, 2000).
[10] Crank, J., *The Mathematics of Diffusion*, (Clarendon Press, Oxford University Press, 1975).
[11] Zill, D. G., *Ecuaciones Diferenciales con Aplicaciones de Modelado*, 6a Ed. (International Thomson Publishing, México, 1997).
[12] Flanagan, D., *Java in a Nutshell: A Desktop Quick Reference*, 5a Ed. (O'Reilly, USA, 2005).
[13] Sierra, K. and Bates, B., *Head First Java*, 2a Ed. (O'Reilly, USA, 2005).
[14] Loy, M., Eckstein, R., Wood, D., Elliott, J. and Cole, B., *Java Swing*, 2a Ed. (O'Reilly, USA, 2002).