

Simulación de órbitas post-newtonianas usando EJS para una clase de Física



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 3 de Febrero de 2013, aceptado el 15 de Junio de 2013)

Resumen

En estos días la teoría de la relatividad general ha tomado una posición importante dentro del mundo de las leyes de la física, y es también una parte esencial de la enseñanza de la física. En este artículo describimos, las posibles órbitas que una partícula puede seguir alrededor de un clásico agujero negro de Schwarzschild. Esto es importante en el campo de la educación, debido a que los estudiantes pueden adquirir algunos conocimientos básicos que se encuentran dentro de la teoría de la relatividad especial de Einstein. Además de este conocimiento teórico básico, los estudiantes adquieren experiencia en la gestión del paquete de software EJS (Easy Java Simulations), para crear las simulaciones físicas en el aula. La ventaja de EJS, es que es muy fácil de usar, y con él, se pueden crear varias simulaciones que el mismo estudiante apreciará por su valor educativo.

Palabras clave: Educación en Física, relatividad general, agujero negro de Schwarzschild.

Abstract

In these days, the general theory of relativity has taken an important position within the world of physics laws, and is also an essential part of physics education. In this article we describe, the possible orbits that a particle can follow around a classical Schwarzschild's black hole. This is important in the field of education, because students may acquire some basic knowledge that is within the general theory of relativity of Einstein. In addition to this basic theoretical knowledge, students gain experience in the management of EJS (Easy Java Simulations) software package, to create physical simulations in the classroom. The advantage of EJS is that is easy to use, and with it, you can create multiple simulations that the same student will appreciate for its educational value.

Keywords: Physics Education, general relativity, Schwarzschild's black hole.

PACS: 01.40.-d, 01.50.-i, 04.20.-q

ISSN 1870-9095

I. INTRODUCCIÓN

En este trabajo nosotros mostramos como se puede realizar una simulación de un objeto que orbita alrededor de un agujero negro. Utilizamos para ello, la solución de Schwarzschild y las ecuaciones de las geodésicas respectivas a esta solución. La solución de Schwarzschild fue históricamente hallada por Karl Schwarzschild un poco después de que Albert Einstein publicará las ecuaciones definitivas para su teoría de la Relatividad General. Pero un docente de Física podría preguntarse: ¿Qué impacto puede tener un ejercicio así, en un aula de clases donde se da la clase Física? Bueno, nuestro objetivo es mostrar primero un sencillo modelo matemático que no lleve mucho tiempo exponer en una clase de Física. Como se puede constatar de varias referencias [1, 2, 3], las simulaciones siempre son importantes como apoyo didáctico a una clase de Física. Pues tienen el efecto positivo de mejorar el interés que los estudiantes muestran por su clase normal, además de que sirve de apoyo didáctico al instructor de Física. Una

herramienta así, es de gran valor, si se tiene en cuenta la sencillez de cómo se logran estas simulaciones, y sirven como un primer ejercicio, tanto para el software utilizado, como para la comprensión de las ecuaciones de la relatividad general. El propósito es sencillo, pues con esta herramienta es posible apoyar al estudiante para que investigue más detalles, una vez que vea un pequeño logro en esta teoría. La simulación puede apoyar mejor la comprensión de las ecuaciones involucradas en el fenómeno. Y con este apoyo de infraestructura teórica, se mejoran las habilidades cognitivas de un estudiante de Física. Además de servir de ejemplo de simulación de un fenómeno físico normal. La simulación puede emplearse como un pequeño laboratorio virtual, donde las dudas e incógnitas que se tiene alrededor de cómo orbitan los planetas, puede quedar satisfactoriamente resuelta. En este trabajo utilizamos el software de utilización libre llamado Easy Java Simulations (de siglas "EJS"), software elaborado por Francisco Esquembre [4, 5], de la Universidad de Murcia en España. La razón de proceder así, y de elegir este

software educativo, es simple, ya que la ventaja que tiene sobre los paquetes de software comerciales, es que su instalación no requiere de licencia especial. De esta forma el software contribuye bien a la economía tanto de las escuelas como de sus estudiantes, al emplear código gratuito, que posteriormente el estudiante después de egresar de su escuela, puede posteriormente acceder sin costo, y recrear varias simulaciones hasta en su mismo hogar. Para la preparación de la utilización de EJS, es necesario, que antes, la computadora donde se vaya a emplear, tenga instalado de preferencia el llamado JDK (Java Development Kit) del lenguaje de programación Java. Ya que éste viene completo, y no sólo, es para correr programas que estén escritos en Java, sino que también permite la escritura de nuevos programas en Java que posteriormente se pueden correr. Esta es la razón principal de nuestra recomendación de instalar el SDK completo, en lugar de instalar solamente el JRE (denominado así, por sus siglas en inglés “Java Runtime Environment”), que permite sólo la corrida de programas en Java ya compilados.

Siendo Java un lenguaje de propósito general, su elección como lenguaje de trasfondo está bien justificada. Además de que Java también es de uso gratuito, y maneja tanto gráficos como aplicaciones para navegador, también conocidos como applets.

En este trabajo, empezamos a describir las sencillas ecuaciones que vamos a emplear en la simulación, ya que es básico tener primero bien en claro, sobre que trasfondo estamos escribiendo la simulación, antes de proceder a explicar cómo se utiliza EJS en la creación de un primer prototipo de simulación, para un objeto que orbite alrededor de una solución de Schwarzschild para gravedad.

II. DESCRIPCIÓN MATEMÁTICA DE LA MÉTRICA DE SCHWARZSCHILD

Para un cuerpo que órbita alrededor de un agujero negro de Schwarzschild [6, 7, 8, 9] es necesario antes que nada escribir la forma matemática de la métrica de Schwarzschild como sigue:

$$ds^2 = \left(1 - \frac{2GM}{c^2 r}\right) c^2 dt^2 - \frac{dr^2}{\left(1 - \frac{2GM}{c^2 r}\right)} - r^2 d\Omega^2. \quad (1)$$

En donde:

$$d\Omega^2 = r^2 \text{sen}(\theta)^2 d\theta^2 + r^2 d\phi^2. \quad (2)$$

Es el elemento de ángulo sólido. Y también

$$ds = c d\tau. \quad (3)$$

Es el elemento de línea de la métrica en términos de la velocidad de la luz en el vacío y del tiempo propio τ , como lo mediría un observador en caída libre. La métrica de Schwarzschild nos provee del espacio-tiempo para nuestra simulación, en él, vamos a colocar una partícula de prueba

para estudiar su movimiento. Si el movimiento es a lo largo de un plano, podemos sin pérdida de generalidad suponer que durante todo el movimiento, θ el ángulo polar permanece a un valor constante, que por conveniencia propia escogemos como

$$\theta = \frac{\pi}{2}. \quad (4)$$

Ahora dividimos a ambos miembros de la métrica (1), por el tiempo propio y fijamos este valor de θ , para obtener la expresión:

$$c^2 = \left(1 - \frac{2MG}{c^2 r}\right) c^2 \left(\frac{dt}{d\tau}\right)^2 - \frac{1}{\left(1 - \frac{2MG}{c^2 r}\right)} \left(\frac{dr}{d\tau}\right)^2 - r^2 \left(\frac{d\phi}{d\tau}\right)^2. \quad (5)$$

Hasta aquí, podemos llegar sin un tratamiento formal. Para encontrar a otras dos constantes de movimiento, es necesario emplear el formalismo matemático de Lagrange o de Hamilton. Los cuales son equivalentes. Usaremos aquí, el comúnmente empleado formalismo de Lagrange. Aquí se hace notar que las órbitas seguidas por el objeto de prueba, son geodésicas en Relatividad General. Por lo tanto, se toma la variación del elemento de línea, y éste se iguala a cero. Así

$$0 = \delta s = \delta \int ds. \quad (6)$$

O bien

$$\delta \int L d\tau = \delta \int \left(g_{\mu\nu} \frac{dx^\mu}{d\tau} \frac{dx^\nu}{d\tau} \right)^{1/2} d\tau = 0. \quad (7)$$

Usemos la notación

$$x^{\mu'} = \frac{dx^\mu}{d\tau}. \quad (8)$$

Para la derivada de $x^{\mu'}$ con respecto al tiempo propio. Luego las ecuaciones de Euler-Lagrange se pueden escribir

$$\frac{d}{d\tau} \left(\frac{\partial L}{\partial x^{\mu'}} \right) - \frac{\partial L}{\partial x^\mu} = 0, \quad (9)$$

Lo cual para este caso, y $x^0 = ct$,

$$\frac{d}{d\tau} \left(r^2 \frac{d\phi}{d\tau} \right) = 0. \quad (10)$$

Además tenemos

$$\frac{d}{d\tau} \left(\left[1 - \frac{2GM}{c^2 r} \right] \frac{dt}{d\tau} \right) = 0. \quad (11)$$

Para $x^3 = \phi$. De manera que tenemos otras dos constantes de movimiento. Así:

$$r^2 \left(\frac{d\phi}{d\tau} \right) = C_1. \quad (12)$$

Y también

$$\left(1 - \frac{2GM}{c^2 r} \right) \frac{dt}{d\tau} = C_2. \quad (13)$$

La constante C_1 se le identifica como el momento angular de la partícula de prueba por unidad de masa [10].

$$C_1 = \frac{l}{m} = \frac{p_\phi}{m} = \frac{1}{m} \frac{\partial L}{\partial \dot{\phi}}. \quad (14)$$

En cuanto a la segunda constante, se le puede identificar como un múltiplo de la energía mecánica, ya que

$$\begin{aligned} E = -p_t &= -\frac{1}{c} \frac{\partial L}{\partial \dot{t}} = mc^2 \left(1 - \frac{2GM}{c^2 r} \right) \frac{dt}{d\tau}, \\ &= \gamma mc^2 \left(1 - \frac{2GM}{c^2 r} \right). \end{aligned} \quad (15)$$

Entonces

$$C_2 = \frac{E}{mc^2}. \quad (16)$$

De esta forma tenemos las expresiones de derivada siguientes

$$\phi' = \frac{d\phi}{d\tau} = \frac{L}{mr^2}. \quad (17)$$

$$t' = \frac{dt}{d\tau} = \frac{E}{mc^2 \left(1 - \frac{2GM}{c^2 r} \right)}. \quad (18)$$

De donde es posible obtener la expresión

$$\frac{d\phi}{dt} = \frac{\phi'}{t'} = \left(1 - \frac{2GM}{c^2 r} \right) \frac{L}{r^2} \cdot \frac{c^2}{E}. \quad (19)$$

Al sustituir, las expresiones (17), (18), (19) en el elemento de línea (5) obtenemos la expresión siguiente

$$\begin{aligned} c^2 &= \frac{1}{1 - \frac{2GM}{c^2 r}} \frac{E^2}{m^2 c^2} \\ &\quad - \frac{1}{1 - \frac{2GM}{c^2 r}} \left(\frac{dr}{d\tau} \right)^2 - \frac{L^2}{m^2 r^2}. \end{aligned} \quad (20)$$

Y de aquí obtenemos la expresión para la razón de cambio de r con respecto al tiempo propio

$$r' = \left(\frac{E}{mc} \right)^2 - \left(1 - \frac{2GM}{c^2 r} \right) \left(c^2 + \frac{L^2}{m^2 c^2} \right). \quad (21)$$

Combinando este resultado con la ecuación (17), obtenemos, también la ecuación diferencial para la órbita de la partícula de prueba

$$\begin{aligned} \left(\frac{dr}{d\phi} \right)^2 &= \left(\frac{r'}{\phi'} \right)^2 \\ &= \frac{r^4}{(Lc/E)^2} - \left(1 - \frac{2GM}{c^2 r} \right) \times \\ &\quad \times \phi r^2 \left(1 + \frac{r^2}{(L/mc)^2} \right). \end{aligned} \quad (22)$$

III. COMO PREPARAR LA ANIMACIÓN CON EJS Y JAVA

Antes de preparar la simulación del fenómeno, tenemos que bajar Easy Java Simulations (EJS) de su sitio web que es <http://www.um.es/fem/EjsWiki/Main/Download> que es un archivo zip y cuyo nombre sigue el siguiente formato EJS_#.#.######.zip. Donde los signos # se reemplazan con números según la versión de EJS que haya disponible en la web. (Un ejemplo de cuando escribí esta simulación fue EJS_4.3.7_120920.zip). Una vez que bajamos este archivo zip, procedemos a desempaquetarlo en algún directorio, donde instalemos el paquete. Una vez desempaquetado contaremos con una carpeta cuyo nombre sigue un formato similar a EJS_#.#.#. Una vez que ingresamos en dicha carpeta, corremos por primera vez EJS, dando doble click a un archivo jar llamado EjsConsole.jar. El programa pide datos personales la primera vez que lo corremos para registrar al autor de los trabajos de simulación, que se realicen en esta instalación de EJS. Una vez proporcionados los datos iniciales, el programa consta de dos ventanas principales Una de ellas es la "Consola de EJS#.#.#", donde se configuran las opciones de Java, y otra ventana principal denominada "EJS_#.#.#". Es, en esta última ventana donde se edita la simulación, y donde elaboramos las opciones que ésta va a tener. Abajo del título de la ventana principal, se encuentran tres radio botones para diseñar tres etapas principales de toda simulación en EJS:

- Descripción

- Modelo
- Vista

En la “descripción” editamos el material general de la simulación, es decir, aquí va escrito una breve descripción del fenómeno que informa al usuario sobre lo que es la simulación, y sus características principales. Esto es, aquí va el material teórico de apoyo para un estudiante que quiera estudiar brevemente, las ecuaciones y la Física que describan al fenómeno. El resultado final de escribir la “descripción” es una página descriptiva en formato HTML, que siempre se carga, a la hora de “correr” o ejecutar la simulación. Por lo que esta parte, se edita al justo del autor y no requiere de una información técnica mayor. Pues básicamente consta de un editor de texto para HTML, y pueden cargarse aquí algunas figuras o ilustraciones clave del fenómeno, que facilitan la descripción del mismo a un alumno. También pueden cargarse aquí, las instrucciones de las actividades que un estudiante de Física, puede llevar a cabo, si es que éste va a ver la simulación como parte de una práctica virtual guiada.

En la parte del “modelo”, se cargan las ecuaciones y los parámetros responsables de la ejecución de la simulación. Se puede decir que esta parte, es el corazón mismo de la simulación, pues gracias a ella, podemos ingresar las ecuaciones diferenciales que rigen la evolución de la simulación, y en otros casos, podemos editar manualmente las ecuaciones solución de las mismas, que aparecen como producto de aplicar algún método numérico a las ecuaciones diferenciales originales. Claro está, esta parte de la edición representa gran parte del trabajo matemático que hay detrás de cualquier simulación y debe de ser tratada con cuidado. Si el comportamiento de nuestra simulación no es el esperado, podemos revisar esta parte con cuidado y checar que las ecuaciones físico-matemáticas que describen al modelo, estén bien escritas. También debemos checar nuestros parámetros, las tolerancias numéricas que vamos a emplear, así como ciertas ecuaciones adicionales (llamadas ecuaciones de “ligadura”), que sean necesarias para la correcta ejecución de la simulación. Cabe mencionar que también es aquí, donde un estudiante puede variar los parámetros de la simulación, modificando lo que en teoría se conocen como “las condiciones iniciales del problema” o “las condiciones de frontera” (según el caso). Entonces es aquí, donde los estudiantes pueden explotar la simulación, “experimentando” con varios parámetros y cambiando así, las “condiciones del problema”, de manera que la evolución de la simulación cambiará de acuerdo a éstos parámetros. Por lo que recomendamos que el instructor prepare una práctica guiada, donde se le sugiera al alumno “modificar” o “cambiar” estos parámetros, y comparar los resultados y el comportamiento observado, con sus “premisas” iniciales que tenía del mismo, antes de ejecutar la simulación. Una vez, que observe el comportamiento, podría “modificar” su hipótesis inicial. Claro que aquí entra en juego una metodología educativa, que el profesor puede proponer de acuerdo a las necesidades de aprendizaje de sus estudiantes. La última parte que hay que editar de la simulación, corresponde al último radio botón de la ventana principal de edición de EJS, denominada “vista”. Aquí se concluyen las

ecuaciones de la simulación incorporando los elementos gráficos que tiene el fenómeno y “ligando” a ellos, las variables (estáticas o dinámicas) del fenómeno estudiado con la simulación. Esta parte de la edición, es importante, porque es aquí donde finalmente se edita la parte responsable de la animación de la simulación, y sus partes evolucionarán de acuerdo a cómo evolucionan las variables de la parte del “modelo”. De hecho, es aquí, donde el estudiante observa el comportamiento general de la simulación, así que posiblemente sea la parte más atractiva de la misma, ya que los elementos gráficos de la simulación pueden participar de forma activa o pasiva (pero descriptiva), de acuerdo al fenómeno o a la característica física que se quiera aproximar mediante la simulación virtual.

En la siguiente sección describo brevemente como se pueden editar las partes del “modelo” y de la “vista” de una simulación para imitar el movimiento planetario, utilizando para ello, las ecuaciones de la relatividad general, en el caso de que el objeto “atractor”, sea un agujero negro de Schwarzschild, aunque el mismo campo gravitatorio exterior del “Sol”, puede aproximarse bastante bien con el modelo de Schwarzschild. Aquí vamos a usar una región circular de color azul, para indicar la frontera con el horizonte de eventos del agujero negro. Si la partícula de prueba se acerca demasiado al círculo central azul, entonces la simulación parará en ese punto, ya que al cruzar el horizonte de eventos toda la Física cambia, y sólo nos interesa el movimiento afuera del horizonte de eventos. El horizonte de eventos está situado a una distancia del centro de coordenadas (donde se encuentra la singularidad espacio-temporal), indicada por el radio de Schwarzschild:

$$r_s = \frac{2GM}{c^2} = 2M_0. \quad (23)$$

En donde se ha usado $M_0 = GM/c^2$, para abreviar la ecuación, para el Sol este parámetro toma el siguiente valor $M_0 = 1.4766 \times 10^5 \text{ cm}$, y $M = 1.989 \times 10^{33} \text{ g}$, con $G/c^2 = 7.4237 \times 10^{-29} \text{ cm/g}$.

La cantidad

$$M_0 = 1.4766 \times 10^5 \text{ cm},$$

de la ecuación (23) se agrega en forma convencional para abreviación de las ecuaciones. De hecho, esta cantidad se usa como ejemplo de las llamadas coordenadas geométricas, podemos consultar el libro de Robert M. Wald [11], para una referencia más amplia acerca del uso de las cantidades geométricas en relatividad general, y de sus factores de conversión.

IV. DESCRIPCION

Esta viene a ser la parte técnicamente más sencilla de editar de una simulación, pues básicamente consiste en colocar la

información necesaria que describe teóricamente a la simulación, se puede usar de sección informativa y básicamente consiste en editar texto en formato HTML. El documento así creado, va a complementar las partes más

técnicas de la simulación. En la figura No. 1 mostramos una posible edición de esta parte, donde hemos omitido los diagramas para simplificar la muestra.

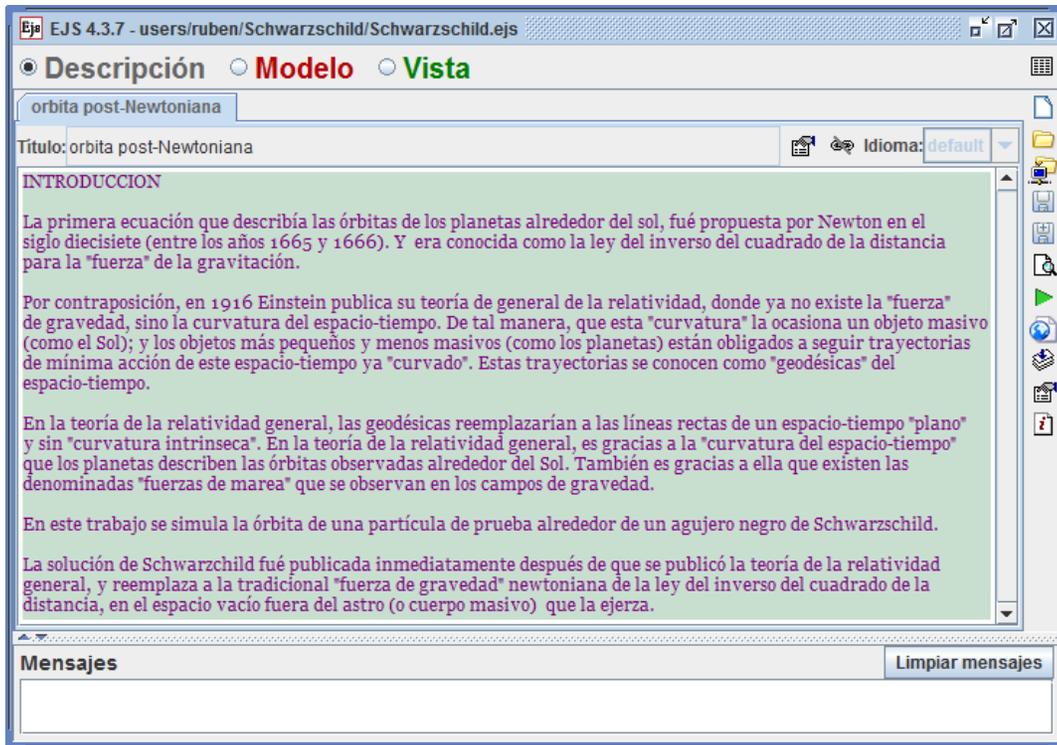


FIGURA 1. Aquí se muestra un ejemplo de una descripción para la simulación. Con el objeto de simplificar las cosas se ha elaborado una descripción sin ilustraciones. En la práctica se deberán añadir si la clase necesita una descripción más gráfica.

V. MODELO

Una vez que elegimos el radio botón “modelo”, aparecen nuevos subradio-botones con los siguientes títulos

- Variables
- Inicialización
- Evolución
- Relaciones fijas
- Propio
- Elementos

Algunas de estas subsecciones se editan y otras no, dependiendo de la simulación y de las necesidades del que vaya a programar la simulación en EJS. Aquí sólo mostramos un ejemplo de cómo pueden editarse estos campos, y dependerá de cada caso particular y si la simulación lleva o no, determinadas características y la forma en cómo se dispongan éstas características.

La siguiente subsección muestra brevemente cómo es posible editar una de éstas subsecciones en EJS.

A. Variables

A continuación describimos, como podemos organizar y anotar cuales van a ser nuestras variables en esta simulación

de un cuerpo orbitando alrededor de un hoyo negro de Schwarzschild. Algunas de éstas variables pueden, convenientemente tener valores iniciales antes de ejecutar la animación de la simulación, como hacemos notar en los siguientes párrafos.

Se le da un click al campo principal que dice “Pulse para crear una página de variables” y aparece un cuadro de diálogo que dice “Introduzca el nuevo nombre”, aquí elegimos como nombre de la primera página “parametros constantes”, y se abre una tabla con cuatro columnas denominadas “Nombre”, “Valor inicial”, “Tipo” y “Dimensión”. Es en estas columnas donde introducimos los siguientes datos, que describirán algunas de nuestras variables. Básicamente en el “Modelo” se describen las relaciones matemáticas que se presentan entre estas variables. Es tarea de la Física el describir el comportamiento de los fenómenos físicos a través de éstos modelos matemáticos, y las relaciones que se dan entre las variables. Por este motivo la parte del “modelo” de la simulación, es donde se concentra la teoría. Y básicamente no puede modificarse sin perjudicar los resultados de la parte de la animación. Por eso, es que se necesita estudiar cada fenómeno Físico en forma matemática y con especial

cuidado, antes de elaborar esta importante fase de la simulación.

TABLA I. Aquí se muestran algunas de las variables que van a ser utilizadas por la simulación. Es importante hacer una identificación con las variables Físicas del fenómeno.

| Modelo | | | |
|--|---------------|--------|-----------|
| Variables | | | |
| Parámetros constantes [nombre de página] | | | |
| Nombre | Valor inicial | Tipo | Dimensión |
| M | 1.4766 | double | |
| m | 0.05 | double | |

| | | | |
|-------|-------------|--------|--|
| v0 | 0.4 | double | |
| v | v0 | double | |
| C | 1 | double | |
| G | 1 | double | |
| gamma | | double | |
| M0 | $G*M/(c*c)$ | double | |
| rs | $2*M0$ | double | |
| E | | double | |
| L | | double | |

Donde, cuando se escriben dos comas, con un espacio en medio, se está indicando un campo vacío. En la figura 2, se muestra esta página de variables con los datos ya mencionados anteriormente.

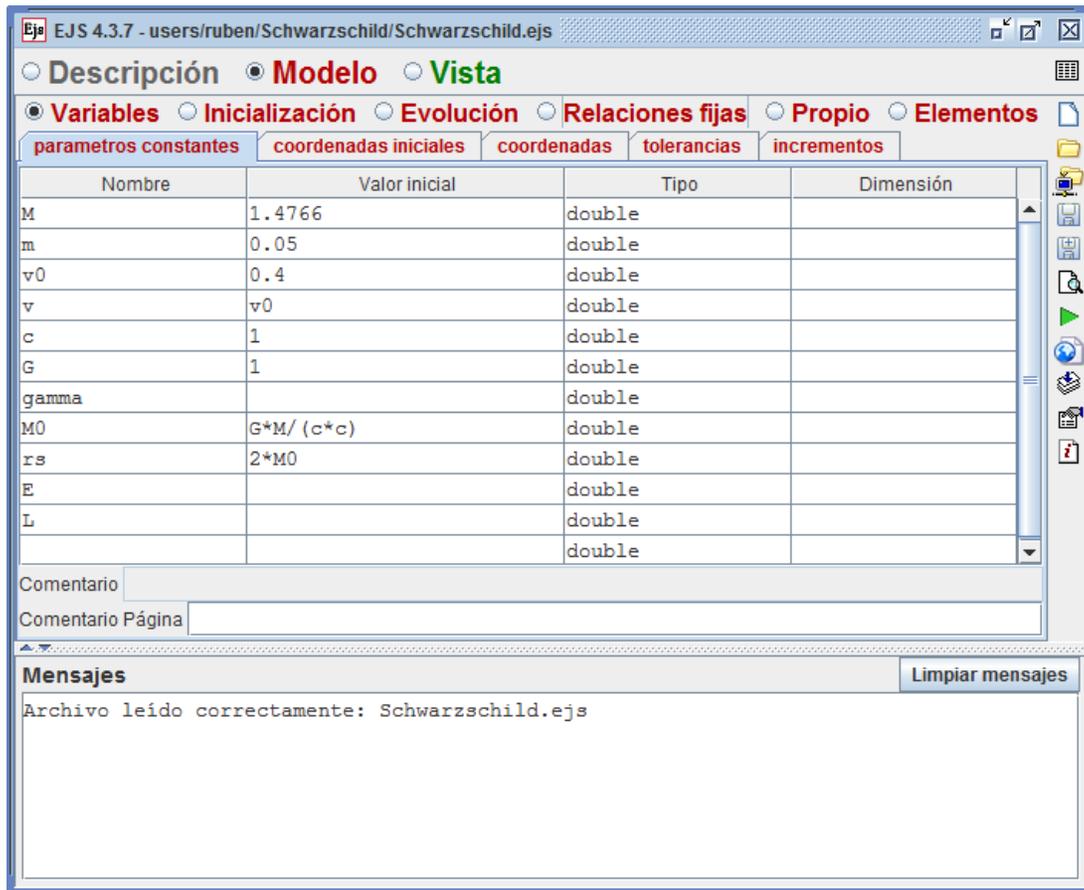


FIGURA 2. Se muestra como en la subsección “Variables” se editan varios parámetros de la simulación, indicando su “nombre”, “valor inicial” y su “tipo”. Las otras páginas denominadas “coordenadas iniciales”, “coordenadas”, “tolerancias” e “incrementos”, se van llenando en forma parecida con los datos que indica el texto.

Las variables antes escritas tienen un significado Físico. Este significado es importante, ya que nos ubica, que

cantidad física estamos representando con dicha variable. A continuación listamos el significado de estas variables:

- M: Masa del agujero negro.
- m: Masa del cuerpo de prueba.

- v_0 : velocidad inicial del cuerpo de prueba.
- v : velocidad instantánea del cuerpo de prueba.
- c : velocidad de la luz.
- G : constante de gravitación universal.
- γ : factor relativista del mismo nombre.
- M_0 : Masa del agujero negro en unidades naturales.
- r_s : radio de Schwarzschild.
- E : Energía total del cuerpo de prueba.
- L : Momento angular del cuerpo de prueba.
- r_0 : Distancia inicial del cuerpo de prueba.
- ϕ_0 : ángulo polar inicial del cuerpo de prueba.
- x_0 : abscisa inicial del cuerpo de prueba.
- y_0 : ordenada inicial del cuerpo de prueba.
- anguloCascara : ángulo local inicial de la velocidad inicial del cuerpo de prueba.
- r : coordenada radial.
- ϕ : coordenada angular.
- $rPrima$: derivada de r respecto al tiempo propio.
- \phiPrima : derivada de ϕ respecto al tiempo propio.
- animar : variable booleana que controla, cuando se corre la animación.
- vel : variable booleana que controla la apertura de la ventana secundaria donde se muestran las velocidades polares instantáneas del cuerpo de prueba.
- t : tiempo transcurrido para un observador situado a lo lejos del hoyo negro.
- x : abscisa del cuerpo de prueba.
- y : ordenada del cuerpo de prueba.
- tolerancia : valor interno mínimo de tolerancia para los cálculos de movimiento.
- ϵ : umbral numérico, para medir que tanto se acerca el cuerpo de prueba al horizonte de eventos.
- dt : incremento numérico de tiempo, para la animación.
- moduloArrastre : modulo local del vector de velocidad inicial.
- faseArrastre : ángulo local de la velocidad inicial del cuerpo de prueba.
- $\text{vectorDeArrastre}_x$: componente x de la velocidad inicial de arrastre del cuerpo de prueba.
- $\text{vectorDeArrastre}_y$: componente y de la velocidad inicial de arrastre del cuerpo de prueba.
- ac_r : aceleración radial inicial del cuerpo de prueba.
- ac_ϕ : aceleración angular inicial del cuerpo de prueba.

Para crear una nueva página de variables con el puntero del ratón nos vamos a donde está la pestaña ya creada “parámetros constantes”, y con un click derecho aparece un menú contextual, donde seleccionamos la opción “Añadir una página”. Una vez creada la página “coordenadas iniciales” la editamos de manera similar a la anterior, pero con los siguientes campos:

TABLA II. Aquí mostramos otro tipo de variables, a los cuales se les identifica como “coordenadas iniciales”. Ya que capturan los valores de las condiciones iniciales del problema.

| Modelo | | | |
|--|---------------------------------|--------|-----------|
| Variables | | | |
| Coordenadas iniciales [nombre de página] | | | |
| Nombre | Valor inicial | Tipo | Dimensión |
| r_0 | 17 | double | |
| ϕ_0 | 0.6 | double | |
| x_0 | $r_0 * \text{Math.cos}(\phi_0)$ | double | |
| y_0 | $r_0 * \text{Math.sin}(\phi_0)$ | double | |
| anguloCascara | ϕ_0 | double | |

Después creamos otras tres páginas con los nombres “coordenadas”, “tolerancias” e “incrementos”, cada una de éstas páginas se llenan de acuerdo a los siguientes tres esquemas:

TABLA III. Este tipo de variables representan en su mayoría las coordenadas instantáneas del cuerpo de prueba. La variable booleana “animar” controla cuando hay animación. La variable booleana “vel” controla la apertura de la ventana secundaria.

| Modelo | | | |
|--------------------------------|---------------|---------|-----------|
| Variables | | | |
| Coordenadas [nombre de página] | | | |
| Nombre | Valor inicial | Tipo | Dimensión |
| R | r_0 | double | |
| $rPrima$ | | double | |
| ϕ | ϕ_0 | double | |
| \phiPrima | | double | |
| animar | | boolean | |
| vel | | boolean | |
| t | 0 | double | |
| x | x_0 | double | |
| y | y_0 | double | |

TABLA IV. La siguiente lista de variables, son las tolerancias numéricas que se manejan durante la simulación, si ciertos valores críticos se llegan a alcanzar entonces el comportamiento de la simulación cambia. Por ejemplo si el cuerpo se llega a acercar demasiado al horizonte de eventos, entonces la animación se detiene.

| Modelo | | | |
|--------------------------------|---------------|--------|-----------|
| Variables | | | |
| tolerancias [nombre de página] | | | |
| Nombre | Valor inicial | Tipo | Dimensión |
| tolerancia | 1.0e-6 | double | |
| ϵ | 0.5e-4 | double | |

TABLA V. Aquí se muestra el incremento en el tiempo para cada cuadro de la animación de la simulación. Asimismo se muestran varios parámetros que se manejan de manera local de acuerdo al principio de equivalencia de Einstein, como el modulo y fase de arrastre y el vector de arrastre.

| Modelo | | | |
|--------------------------------|----------------|--------|-----------|
| Variables | | | |
| incrementos [nombre de página] | | | |
| Nombre | Valor inicial | Tipo | Dimensión |
| Dt | 10.0 | double | |
| moduloArrastre | 2.5 | double | |
| faseArrastre | Phi0+Math.PI/2 | double | |
| vectorDeArrastre_x | | double | |
| vectorDeArrastre_y | | double | |

Y digamos que hasta aquí, hemos logrado anotar y agrupar en categorías a todas las variables que va a ocupar la simulación. Otras formas de organizarlas son también válidas y este sencillo ejemplo que hemos escrito, puede servir de ejemplo.

B. Inicialización

Hemos descrito a todo el conjunto de variables que necesitamos para construir esta simulación. Ahora pasamos a describir la parte del “Modelo”, que tiene que ver con la inicialización de estos parámetros. Para ello, es necesario escribir una breve página de código Java, en una subsección del “Modelo” llamada “Inicialización”. Entonces, procediendo en forma similar a como lo hicimos anteriormente para las variables de la simulación, disponemos entonces de una nueva página que hemos llamado en este ejemplo “valores iniciales”. En ella, escribimos el siguiente código en Java, que será ejecutado al inicio de la simulación y cada vez que se llame a la función interna de EJS conocida con el nombre de `_initialize()`. Como ilustración, parte de este código se puede observar en la figura 2. Para no perder ninguna parte de él, escribimos éste código a continuación:

TABLA VI. Forma de escribir código Java en EJS en forma de página, en este caso se muestra el código para la página que hemos titulado “valores iniciales”.

| Modelo |
|--|
| Inicialización |
| valores iniciales [Nombre de página] |
| <pre>t=0; animar=true; r=r0; phi=phi0; v=v0; x=x0=r0*Math.cos(phi0); y=y0=r0*Math.sin(phi0);</pre> |

```
//Masa en unidades geométricas
M0=G*M/(c*c);
//Radio de Schwarzschild
rs=2*M0;
gamma=1/Math.sqrt(1-(v*v)/(c*c));
//Energía
E=gamma*m*c*c*(1-rs/r0);
//Momento angular
L=r0*gamma*v*Math.sin(anguloCascara);
vectorDeArrastre_x=
moduloArrastre*Math.cos(faseArrastre);
vectorDeArrastre_y=
moduloArrastre*Math.sin(faseArrastre);
double radical=E*E/(m*m)-(1-rs/r0)
*(c*c+L*L/(m*m*r0*r0));

if (radical>0) {
//dr/dtau
rPrima=m*Math.sqrt(radical)*(1-rs/r0)/E;
} else rPrima=0;
//dphi/dtau
phiPrima=L*m*(1-rs/r0)/(E*r0*r0);
if((anguloCascara>3*Math.PI/2)||
(anguloCascara<Math.PI/2)) {
rPrima=-rPrima;
}
}
```

Explicaremos un poco éste código. Primero inicializamos el tiempo y los valores de las coordenadas rectangulares y polares, además de la velocidad inicial, que queda expresada como una fracción de la velocidad de la luz. Por lo que su rango queda expresado por las desigualdades siguientes

$$0 \leq v \leq 1 = c. \tag{24}$$

Anotamos también la masa en unidades geométricas, y escribimos cual es el radio de Schwarzschild según la teoría del agujero negro de Schwarzschild. Inicializamos el factor de la relatividad especial “gamma” como sigue:

$$\gamma = \frac{1}{\sqrt{1-\beta^2}}, \quad \beta = \frac{v}{c}. \tag{25}$$

Donde, hemos tomado en cuenta el “Principio de Equivalencia” de Einstein que asegura, que en una vecindad lo suficientemente pequeña de un evento (En este caso el “evento” está dado por la posición y tiempo inicial del cuerpo en órbita) en el espacio-tiempo, se cumplen las leyes de la relatividad especial. Y es por eso que a esta vecindad topológica la describimos con un sistema coordenado local o de “cáscara”, como puede consultarse en la teoría del hoyo negro de Schwarzschild. De acuerdo a ésta teoría escribimos la energía y el momento angular del cuerpo que está en órbita. También calculamos un “vector de arrastre”, que nos dará los parámetros cinéticos iniciales del cuerpo de prueba. Calculamos, asimismo sus velocidades radial y angular de acuerdo a la teoría, y establecemos la velocidad radial de acuerdo al cuadrante que ocupe el ángulo local o de “cáscara”.

Otra de las variables que se inicializan es la variable booleana “animar”. Ella se usa para controlar cuando la simulación va a hacer la animación del cuerpo orbitando el agujero negro y cuando la detiene. Cuando tiene el valor “true”, la animación está en curso, cuando tiene el valor

“false” la detenemos. Esto generalmente ocurre cuando el cuerpo de prueba alcanza una distancia radial igual que el radio de Schwarzschild.

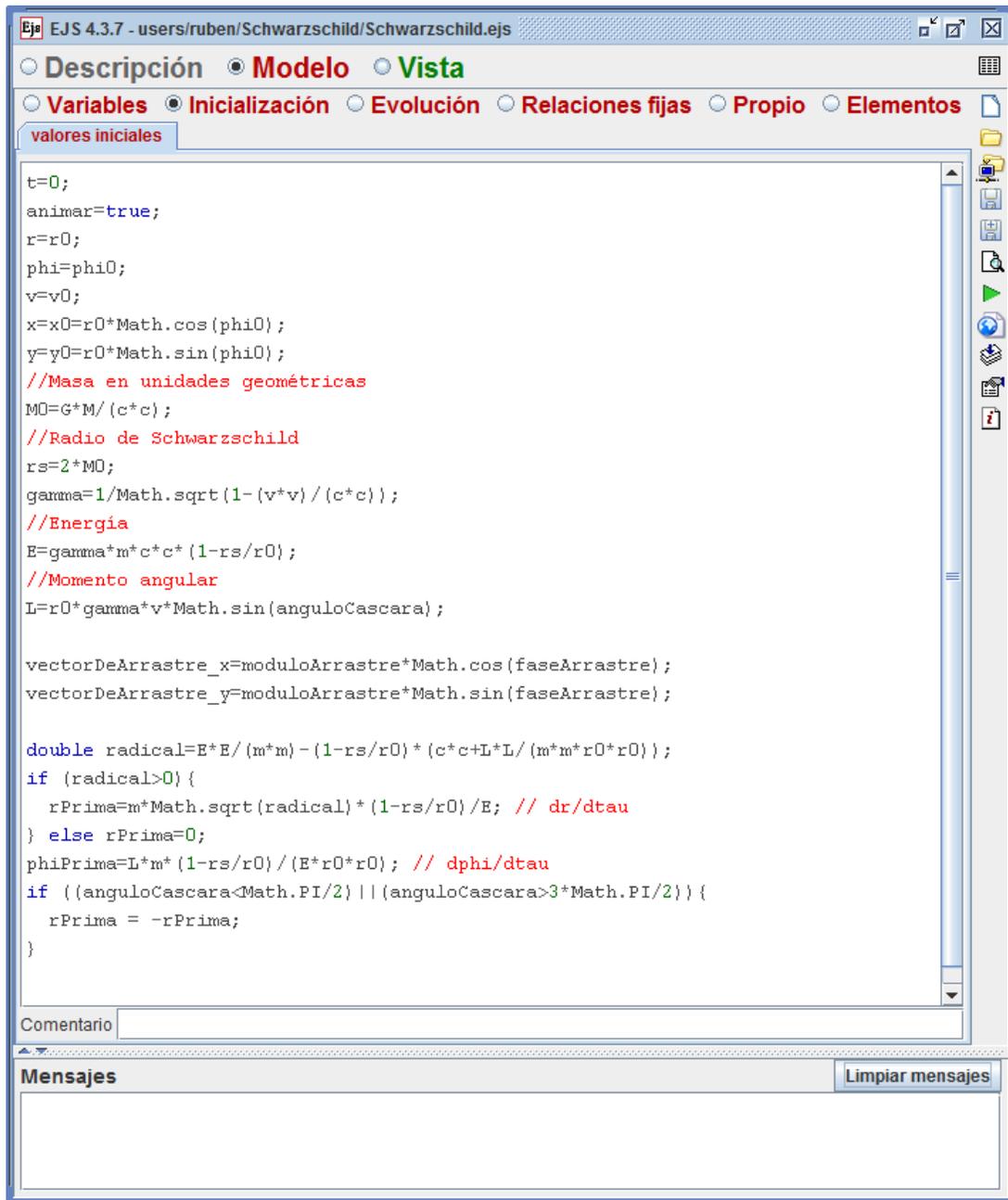


FIGURA 3. Aquí mostramos el código de inicialización de la simulación. Este código se ejecuta cada vez que se invoca a la función estándar de EJS llamada “_initialize()”. Debemos de tomar la teoría de un marco local o de cáscara en cuenta, durante la escritura del código.

C. Evolución

En esta parte, es necesario introducir una página con ecuaciones diferenciales, la figura 3, ilustra como editar esta parte de la simulación, que es donde se carga la dinámica del movimiento del cuerpo de prueba alrededor del hoyo

negro de Schwarzschild. También podemos escribir éstas ecuaciones diferenciales como lo muestra la tabla 2, que está con código en Java (dentro de la ventana de edición de EJS para esta etapa).

Easy Java Simulations emplea métodos numéricos para aproximar las soluciones a las ecuaciones diferenciales.

Rubén Sánchez-Sánchez

Existen diferentes métodos numéricos que se pueden escoger utilizando una lista desplegable que se encuentra en la esquina inferior izquierda de la ventana de “Evolución”. En este caso hemos escogido el método de Cash-Karp para ecuaciones diferenciales ordinarias (EDO).

TABLA VII. Forma de elaborar una página (“Evolución”), que contiene las ecuaciones diferenciales de movimiento para un objeto que se mueve bajo la influencia de un hoyo negro de Schwarzschild.

```
double phiPrima2 = Math.pow(phiPrima,2);
double rPrima2 = Math.pow(rPrima,2);
double ac_r, ac_phi;

ac_r = -(4*M3-4*M2*r+(4*M0*r4-4*M2*r3-r5)
*phiPrima2 + r2*M0*(1-3*phiPrima2))
/(r4*(1-rs/r));

ac_phi = -2*(1-3*M0/r)*phiPrima*
rPrima/(r*(1-rs/r));
```

De acuerdo a las fórmulas para las aceleraciones de un cuerpo que orbita un agujero negro de Schwarzschild:

$$\frac{d^2 r}{d\tau^2} = -(4M_0^3 - 4M_0^2 r + (4M_0 r^4 - r^5) \left(\frac{d\phi}{d\tau}\right)^2 + r^2 M_0 \times \left(1 - 3\left(\frac{d\phi}{d\tau}\right)^2\right) (r^4 (1 - r_s / r))^{-1}, \quad (26)$$

$$\frac{d^2 \phi}{d\tau^2} = -2 \left(1 - 3 \frac{M_0}{r}\right) \left(\frac{d\phi}{d\tau}\right) \cdot \left(\frac{dr}{d\tau}\right) \cdot \left(r \left(1 - \frac{r_s}{r}\right)\right)^{-1}. \quad (27)$$

A continuación se pulsa el botón “Eventos” situado a la izquierda de la ventana “Evolución” y se abre una nueva ventana. Aquí controlamos la animación cuando una circunstancia especial acontece. Por ejemplo, en nuestro caso no queremos continuar con la simulación si nuestro cuerpo de prueba se aproxima sensiblemente al horizonte de eventos del agujero negro, y preferimos detener la animación, aprovechando el valor booleano de la variable “animar”. En dicho caso cambiamos su valor a “false”, lo cual detiene la “evolución” de nuestro cuerpo de prueba y la animación se detiene. Con el objeto de hacer la edición de tal evento clara, aquí escribimos en forma de tabla las entradas a editar pertinentes. Además en la tabla IX ilustramos mejor el punto, mostrando el código Java que hay que introducir para editar esta ventana.

TABLA IX. En el caso de que el cuerpo de prueba se acerque demasiado al horizonte de eventos (indicado en el diagrama de la animación por un círculo de color azul), entonces el valor booleano de “animar” cambia a false y la animación se detiene. Ya que sólo nos interesa observar el movimiento del cuerpo fuera del horizonte de eventos.

| Eventos para EDO Evolución | | |
|--|-------------------|------------------------|
| Horizonte de eventos | | |
| Tipo Evento de estado | Interacciones 100 | Método SECANTE |
| Valor cero | | Tol: tolerancia |
| <pre>double err=Double.MAX_VALUE; ef(r-rs-epsilon<err){ err=r-rs-epsilon; }</pre> | | |

| Modelo | |
|--|------------------------|
| Evolución | |
| Evolución del cuerpo de prueba [nombre de la página] | |
| Var. Indep. t | Incremento: dt |
| Estado | Derivada |
| $\frac{dr}{dt} =$ | animar?rPrima:0 |
| $\frac{drPrima}{dt} =$ | animar?ac_r:0 |
| $\frac{d\phi}{dt} =$ | animar?phiPrima:0 |
| $\frac{d\phiPrima}{dt} =$ | animar?ac_phi:0 |
| Método: Cash_Karp5(4) | Tol: tolerancia |

En la tabla notamos que la primer derivada de la coordenada polar r es igual a la expresión “rPrima” hallada en la inicialización, cuando “animar” tiene el valor booleano de “true”, lo cual corresponde a una situación donde el objeto orbita fuera del horizonte de eventos. Cuando el objeto muy está cerca del horizonte de eventos “animar” adquiere el valor “false” y la animación se detiene.

Fijamos un método numérico para resolver las ecuaciones diferenciales (método de Cash-Karp 5 (4)) en esta ventana. Y podemos escribir las expresiones de la aceleración radial y angular de acuerdo a la teoría. Para éste objetivo presionamos un botón situado en la esquina superior derecha de la página llamado “Prelim” (a la derecha de “Incremento”) y donde colocamos el siguiente código descrito en la tabla VIII. Estas ventanas se ocupan para describir la aceleración que tendrá el cuerpo de prueba (de acuerdo a la teoría) antes de ejecutar la evolución. El código es importante, pues para describir una evolución es necesario escribir las dos primeras derivadas de las coordenadas polares.

TABLA VIII. Aquí se muestran las componentes polares de la aceleración sufrida por el cuerpo de prueba.

| Prelim |
|--|
| <pre>double r2 = Math.pow(r,2); double r3 = r2*r; double r4 = Math.pow(r2,2); double M2 = Math.pow(M0,2); double M3 = M2*M0;</pre> |

| |
|--|
| return err; |
| Acción <input checked="" type="checkbox"/> Terminar <u>p</u> aso en el evento |
| animar=false; |
| Comentario <input type="text"/> |
| Acción efecto Zenón |
| Comprobar eventos a pasos no mayores de <input type="text"/> |

Simulación de órbitas post-newtonianas usando EJS para una clase de Física hoja de edición que titulamos aquí “Relaciones matemáticas fijas” con el objeto de llevar a cabo en forma integrada este cambio de coordenadas, las ecuaciones matemáticas son muy conocidas, y las mostramos en seguida:

$$x = r \cos(\theta), \quad y = r \sin(\theta). \quad (28)$$

En la siguiente subetapa de edición de la simulación escribiremos las relaciones matemáticas fijas. Es decir, aquellas relaciones que se mantienen válidas durante toda la simulación, En EJS hay un subradio botón dedicado a esta tarea llamado “Relaciones fijas”, y lo trataremos de describir en la siguiente subsección.

D. Relaciones fijas

Las relaciones fijas para este tipo de simulación son simplemente el cambio de coordenadas de polares a rectangulares. Esto es necesario ya que EJS, maneja por default las coordenadas rectangulares cartesianas para ejecutar la animación. De manera que hay que crear otra

Esto mismo lo ilustramos en la tabla no. #, y adjuntamos también una imagen de cómo se vería la ventana de edición del programa cuando editamos esta parte de la simulación, en la figura 4.

TABLA X. Las relaciones que se mantiene ciertas durante toda la simulación son las de cambio de coordenadas.

| |
|--|
| Modelo |
| Relaciones fijas |
| Relaciones matemáticas fijas |
| // Cambio de coordenadas x=r*Math.cos(phi); y=r*Math.sin(phi); |

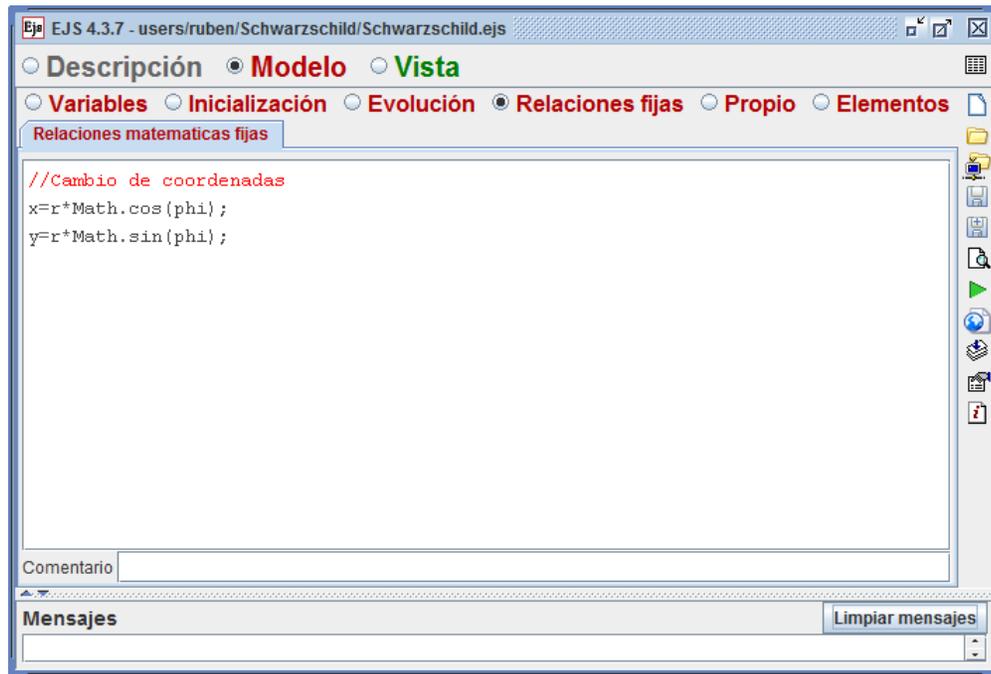


FIGURA 4. Las relaciones matemáticas que se mantienen ciertas durante toda la simulación. En este caso se hace el cambio de coordenadas de polares a rectangulares para que la animación se ejecute de manera adecuada.

En la siguiente subetapa se manejarán los códigos Java de las funciones definidas por el autor de la simulación. Estas funciones son necesarias para el correcto funcionamiento de otras partes del código del “Modelo”. Las funciones propias se añadirán a la larga lista de funciones ya predeterminadas de EJS. Por ejemplo, una de estas funciones predeterminadas es `_initialize()`, cuya función en EJS es correr el código de inicialización de la simulación. Aquí por ejemplo, `_initialize()`

correría el código anotado en la tabla VI.

E. Propio

Esta parte de la simulación, guarda las funciones propias definidas por el desarrollador, que se ocupan en las demás partes de la simulación, como en la *inicialización* o en la *evolución* de la misma. En este caso, preparamos tres hojas

Rubén Sánchez-Sánchez

de código con nombre “Arrastre”, “Vector” y “mapeoAngular”. El código Java para cada una de estas páginas las mostramos en las siguientes tablas 11, 12 y 13. Es importante mencionar que este código de escribió en hojas de edición separadas para la misma subetapa de EJS llamada “Propio”. Es una buena costumbre de programación hacer este tipo de divisiones ya que ayuda a organizar mejor el código y en caso de algún cambio en el código ayuda a localizar mejor el punto donde deberá ir la actualización de código.

TABLA XI. Aquí se muestra la función propia “arrastrar” que funciona si el usuario “arrastra” al cuerpo de prueba al principio de la simulación, antes de que empiece la animación del movimiento del cuerpo de prueba.

| Modelo |
|--|
| Propio |
| Arrastre [nombre de página] |
| <pre>public void arrastrar () { t = 0; phi0 = Math.atan2(y0, x0); //si el radio es muy pequeño aumentarlo if(r0 < 2*M0+2*epsilon) r0 = 2*M0+2*epsilon; x0 = r0*Math.cos(phi0); y0 = r0*Math.sin(phi0); _initialize(); }</pre> |

TABLA XII. Mostramos una función propia cuyo objetivo es la inicialización del vector de arrastre, del cuerpo de prueba. Este vector determinará de manera local las coordenadas de la velocidad inicial del cuerpo de prueba.

| Modelo |
|---|
| Propio |
| Vector [nombre de página] |
| <pre>public void vectorDeArrastre () { double anguloImagen=Math.atan2 (vectorDeArrastre_y, vectorDeArrastre_x); vectorDeArrastre_x=Math.cos(anguloImagen); vectorDeArrastre_y=Math.sin(anguloImagen); anguloCascara= mapeoAngular(r0,phi0,anguloImagen); _initialize(); }</pre> |

TABLA XIII. El “mapeo angular” es una función propia que se utiliza para fijar al vector de arrastre.

| Modelo |
|--|
| Propio |
| mapeoAngular [nombre de página] |
| <pre>public double mapeoAngular (double r, double phi, double anguloImagen) { // dr/dt double rPunto=Math.cos(anguloImagen-phi);</pre> |

```
double proyeccionX=rPunto/Math.sqrt(1-2*M0/r);
// dphi/dt
double phiPunto=Math.sin(anguloImagen+phi);
double angulo=Math.atan2(phiPunto,-proyeccionX);
if ( angulo<0 )
    angulo=angulo+2*math.PI;
if ( angulo >= 2*math.PI )
    angulo =angulo-2*Math.PI;

return angulo;
}
```

Con esto, hemos concluido la etapa de escribir el “Modelo” de la simulación, en la siguiente sección describimos una parte que tiene que ver con la animación gráfica de la simulación y que depende de que el modelo matemático de esta sección este bien escrita. Esta etapa del desarrollo de la simulación se conoce como “Vista” en EJS. Dependiendo del número de “elementos gráficos” que utilice, es la extensión de código que hay que editar. Ya que cada elemento, posee una lista de “propiedades” que hay que precisar para poder construir esta etapa de la simulación.

VI. VISTA

Una vez, concluida la parte principal de la simulación el “Modelo”, donde se incorporan todas las ecuaciones matemáticas que rigen el comportamiento general de la simulación, resta sólo editar la parte correspondiente a la “Vista”. En la “Vista”, se agregan los elementos gráficos de una animación que pretende seguir el comportamiento del fenómeno Físico, la animación o “Vista” es la parte visual de toda simulación. Ahí podemos observar cómo evoluciona el fenómeno Físico que queramos estudiar y se pueden presentar gráficas auxiliares para mostrar un análisis visual de las cantidades matemáticas involucradas.

Para acceder a la fase de la “Vista” en EJS, es necesario seleccionar el radio-botón del mismo nombre, que se encuentra en la esquina superior derecha de la ventana principal de edición de EJS. Una vez que seleccionamos este botón hallamos que una ventana divide en tres secciones, la sección de abajo corresponde a los “Mensajes”, que genera el programa en Java que genera EJS, cada vez que se carga o descarga un archivo, o cuando se corre la simulación. Esta parte es importante, porque aquí es donde el programa se puede “debugear”. Es decir, aquí podemos ver los mensajes de error generados por nuestra simulación, cuando ésta se encuentra en desarrollo, y necesitamos corregirla para que finalmente se pueda ejecutar sin errores de código.

La parte superior de esta ventana se divide, a su vez, en otras dos secciones principales. Una está a la izquierda y se llama “Árbol de elementos”, y la otra se encuentra a la derecha, y tiene el nombre de “Elementos para la vista”.

La forma de “editar” esta parte de la simulación en EJS, es arrastrando a los elementos que se encuentran del lado derecho hacia el panel de la izquierda donde se encuentra

Simulación de órbitas post-newtonianas usando EJS para una clase de Física

propiedades. Por ejemplo en la figura 8 hay un ejemplo de edición de estas propiedades. En este caso, estamos editando las propiedades del “horizonte de eventos de la simulación, que estará situado en el centro del mapa donde se ejecuta la animación del cuerpo de prueba moviéndose de acuerdo a las ecuaciones de Einstein para la solución de Schwarzschild.

una “raíz” de la vista llamada “Vista de la Simulación”. Y poco a poco se va construyendo un árbol, a partir de estos elementos, que constituye la “Vista” de la simulación. En las figuras 5, 6 y 7 mostramos el árbol por partes.

Una vez construido el árbol de elementos de vista, se procede a editar cada uno de estos elementos para otorgarles ciertas propiedades. El comportamiento de la parte de la animación dependerá de cómo se editen estas

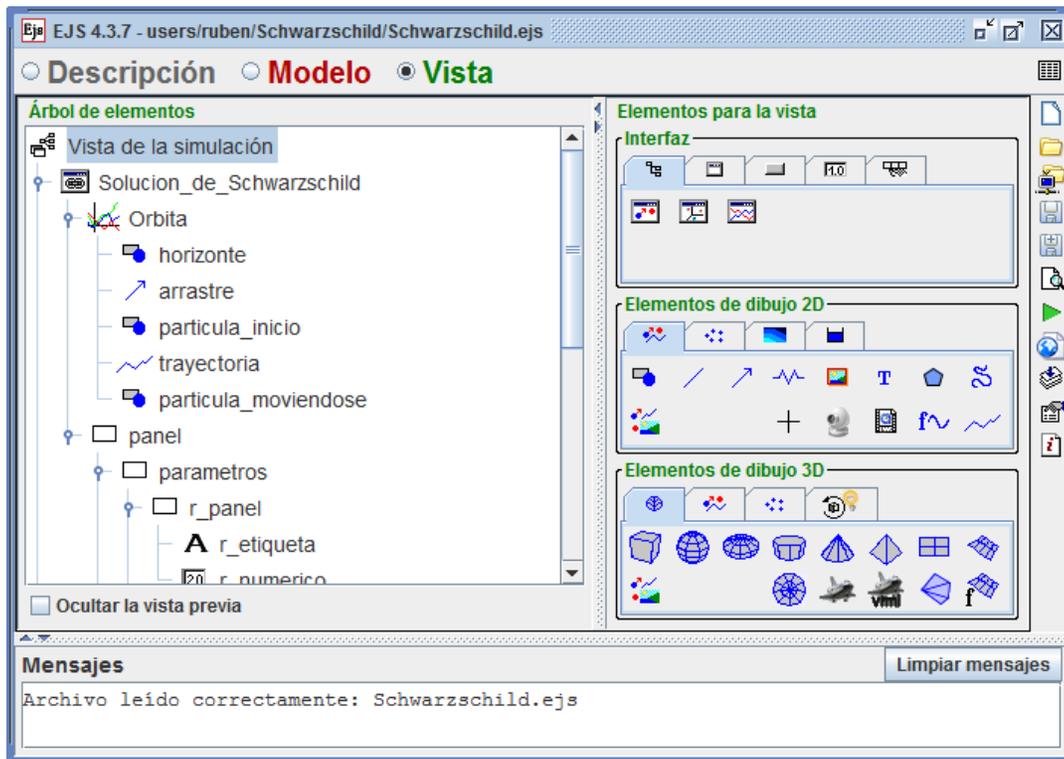


FIGURA 5. Primera parte ya construida de la “Vista de la simulación”, se han arrastrado elementos del panel derecho de la ventana de edición de EJS hacia la izquierda para ir construyendo una estructura de datos en forma de árbol con diferentes tipos de nodos. Para editar cada nodo o elemento de la vista hay que señalarlo y dar un “click” derecho del ratón.

En la figura 8 vemos el tamaño del horizonte de eventos, su tamaño según la teoría tiene un radio igual al radio de Schwarzschild de $2M_0$, según la teoría y la relación (23).

También se puede observar que al hoyo negro lo ubicamos en el origen de coordenadas. Otra característica es que es de forma circular, aunque se selecciona ELLIPSE los tamaños en el eje x y y son los mismos, el color que elegimos en este sencillo ejemplo fue el azul.

Así como este ejemplo de las propiedades gráficas del horizonte de eventos, nos muestra como editar los campos. Así existen muchos otros elementos que hay que editar abriendo un cuadro de propiedades, característico al tipo de elemento dado.

En esta sección, en lugar de mostrar los diagramas de todos estos elementos, optamos por mostrar tablas con los campos de las características que son editados, para cada ventana propiedades de cada elemento. Los campos vacíos, de cada cuadro de propiedades, no se

muestran para simplificar la descripción del llenado de todos estos cuadros. Dependiendo de lo que el diseñador quiera mostrar en la simulación y del diseño gráfico que elija será el número de elementos que haya que editar.

La parte de la “Vista” de la simulación, puede ser muy entretenida de editar. En primer lugar, hay que notar que en su construcción utilizamos dos ventanas: una ventana principal llamada “Solucion_de_Schwarzschild” y otra ventana de carácter secundario, llamada “velocidades”.

La ventana principal de la simulación es la primera, y siempre se abre cuando la simulación corre. En cuanto a la segunda ventana, ella se abrirá al seleccionar el valor booleano “vel” como “true”. Cuando “vel” toma un valor de “false”, esta ventana se cierra. La forma gráfica de controlar el valor de verdad de “vel” es a través de un elemento gráfico llamado “selector”, que se muestra en la simulación como un cuadro, que a veces tiene una paloma (para indicar que la propiedad ha sido seleccionada) y a veces no. Esto

depende de si la ventana secundaria está abierta o no, respectivamente. En el lenguaje de programación Java a esto

se le llama una “función de acción”, que en EJS es más fácil de manipular a través de este elemento “selector”.

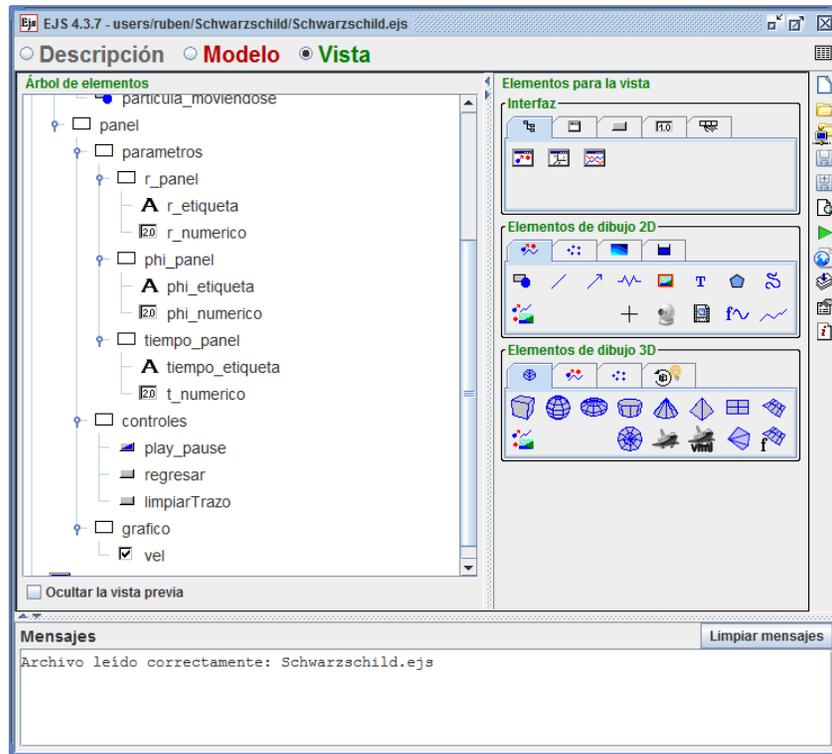


FIGURA 6. Segunda parte del árbol de la vista ya construida. Aquí se pueden apreciar varios paneles para la disposición de diferentes campos visuales, donde se muestran las coordenadas polares del cuerpo de prueba, y como van cambiando durante la evolución de la simulación.

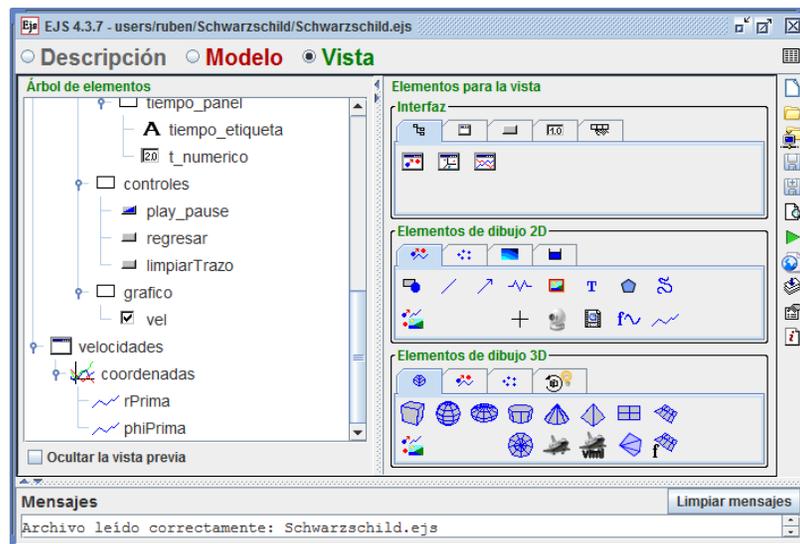


FIGURA 7. Tercera parte de la vista ya construida. Desde luego, este es un ejemplo y se pueden añadir varios elementos más.

Simulación de órbitas post-newtonianas usando EJS para una clase de Física

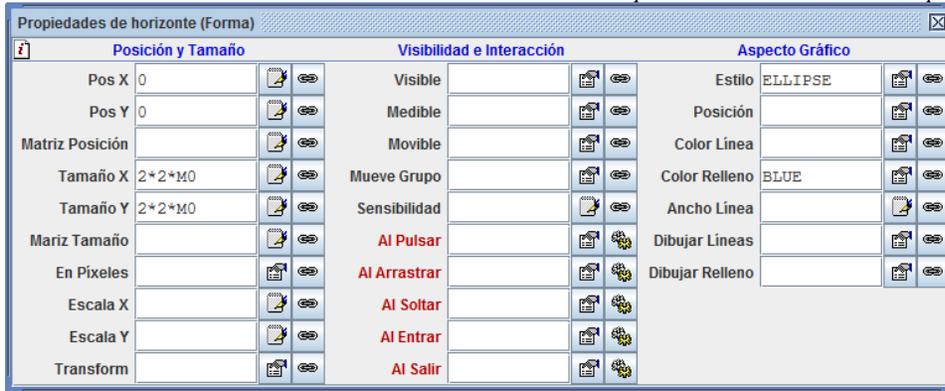


FIGURA 8. La ventana “propiedades” del elemento “horizonte” se activa haciendo un “click” derecho con el ratón sobre dicho elemento gráfico. Aquí se puede apreciar que algunos campos los hemos editado, y no es necesario llenar todos os campos. Ya que varios de ellos toman valores por “default”.

Las propiedades de la ventana principal se muestran enlistadas en la tabla XIV, y las propiedades de la ventana secundaria se encuentran en la tabla XXXIV.

Las propiedades gráficas del cuadro de selección o “selector” que abre a la ventana secundaria, se encuentran enlistadas en la tabla XXXIII.

TABLA XIV. Aquí mostramos las propiedades que tiene la ventana principal de la simulación llamada “Solucion_de_Schwarzschild”. Sólo se muestran los campos editados ya que existen campos que por tener valores de “default” no requieren de ser editados. Aquí mostramos el título de la ventana, su distribución respecto a otros elementos. Su valor de visibilidad y su posición y tamaño.

| Vista | |
|--|-----------------------------|
| Árbol de elementos | |
| Propiedades de Solucion_de_Schwarzschild (Ventana) | |
| Principal | |
| Título | “Solución de Schwarzschild” |
| Distribución | Border |
| Visible | True |
| Posición y Tamaño | |
| Posición | “22,263” |
| Tamaño | “533,554” |

En la tabla XV se muestran las propiedades del gráfico principal donde se ilustra el movimiento de un cuerpo de prueba alrededor de un agujero negro de Schwarzschild. Dentro de este “panel con ejes”, se incorporan nuevos elementos gráficos como lo son el horizonte de eventos del agujero negro de Schwarzschild, mostrado en la tabla 16, el cuerpo de prueba mostrado en las tablas 18 y 20, junto con su velocidad inicial de la tabla 17, también se muestra como elemento “hijo” de este panel a la traza que va dejando el cuerpo de prueba mientras se mueve, la tabla 19, resume sus propiedades. Los elementos gráficos que contienen a otros llamados “hijos” se llaman convenientemente nodos “padres” de los segundos. Esta nomenclatura es usual en la teoría de las “estructuras de datos”, siempre que se aprende a programar en algún lenguaje, estas estructuras son muy útiles para aprender el respectivo lenguaje de programación.

Cabe mencionar que si bien, esta lista de tablas de propiedades de elementos gráficos, no es exhaustiva, si contiene los elementos que de alguna forma son significativos para la correcta edición de la “Vista”. Porque algunos “paneles” omitidos son similares a otros ya dados, así también como algunas “etiquetas”.

TABLA XV. Aquí se muestran las propiedades del elemento gráfico “Orbita” que básicamente es un panel con ejes coordenados. Elegimos un tipo de malla cartesiana, como se puede notar en la propiedad “Tipo de Ejes”. El campo de visión es de -20 a 20 tanto en la coordenada x como en la y. Lo que le da un aspecto cuadrado. Así el campo “Cuadrado” debe de colocarse como “true”.

| Vista | |
|--|------------|
| Árbol de elementos | |
| Propiedades de Orbita (Panel con ejes) | |
| Escala | |
| Autoscala X | true |
| Autoscala Y | true |
| Mínimo X | -20 |
| Máximo X | 20 |
| Mínimo Y | -20 |
| Máximo Y | 20 |
| Cuadrado | true |
| Decoración | |
| Título | “Orbita” |
| Tipo Ejes | CARTESIAN1 |

TABLA XVI. Aquí se muestran en forma de tabla las propiedades para el horizonte de eventos de la simulación. Estas mismas propiedades se muestran en la figura 8.

| Vista | |
|--------------------------|--------|
| Árbol de elementos | |
| Propiedades de horizonte | |
| Posición y tamaño | |
| Pos X | 0 |
| Pos Y | 0 |
| Tamaño X | 2*2*M0 |
| Tamaño Y | 2*2*M0 |
| Aspecto Gráfico | |

| | |
|----------------------|---------|
| Estilo | ELLIPSE |
| Color Relleno | BLUE |

TABLA XVII. Aquí colocamos una flecha sobre el cuerpo de prueba que representa su velocidad inicial.

| Vista | |
|----------------------------------|--------------------|
| Árbol de elementos | |
| Propiedades de arrastre (Flecha) | |
| Posición y Tamaño | |
| Pos X | x0 |
| Pos Y | y0 |
| Tamaño X | vectorDeArrastre_x |
| Tamaño Y | vectorDeArrastre_y |
| Escala X | 1 |
| Escala Y | 1 |
| Visibilidad e Interacción | |
| Visible | t==0 |
| Medible | True |
| Movible | True |
| Dimensionable | t==0 |
| Mueve Grupo | True |
| Dim Grupo | True |
| Sensibilidad | 50 |
| Al Arrastrar | vectorDeArrastre() |
| Al Soltar | _initialize() |
| Al Entrar | _initialize() |
| Al Salir | _initialize() |
| Aspecto Gráfico | |
| Estilo | ARROW |
| Color Línea | DARKGRAY |
| Ancho Línea | 2.8 |

TABLA XVIII. Aquí tenemos al cuerpo de prueba en su posición inicial, con un tamaño especificado por las propiedades “Tamaño X” y “Tamaño Y”. Como se puede ver esta imagen del cuerpo de prueba es visible solo al principio en “t==0”.

| Vista | |
|---|-------------------|
| Árbol de elementos | |
| Propiedades de partícula_inicio (Forma) | |
| Posición y Tamaño | |
| Pos X | x0 |
| Pos Y | y0 |
| Tamaño X | 15 |
| Tamaño Y | 15 |
| En Píxeles | True |
| Visibilidad e interacción | |
| Visible | t==0 |
| Medible | True |
| Movible | True |
| Mueve Grupo | True |
| Sensibilidad | 50 |
| Al Arrastrar | arrastrar() |
| Al Soltar | _view.ClearData() |
| Al Salir | _initialize() |
| Aspecto Gráfico | |
| Estilo | WHEEL |

| | |
|----------------------|------|
| Color Relleno | PINK |
|----------------------|------|

TABLA XIX. Conforme el cuerpo de prueba se mueve va dejando una traza de su trayectoria alrededor del agujero negro de Schwarzschild. Sus propiedades se pueden editar fácilmente en EJS. Simplemente se anotan los campos significativos. Por ejemplo la posición de la partícula en un determinado tiempo t se fija con sus coordenadas cartesianas x y y , que son los valores de las propiedades “Entrada X” y “Entrada Y” de manera respectiva. Trazamos un número máximo de puntos de 50000, porque así se nos permite ver en detalle la evolución de la trayectoria del cuerpo de prueba conforme pasa el tiempo.

| Vista | |
|------------------------------------|-------------|
| Árbol de elementos | |
| Propiedades de trayectoria (Traza) | |
| Entrada | |
| Entrada X | x |
| Entrada Y | y |
| Aspecto Gráfico | |
| Max Puntos | 50000 |
| No Repetir | true |
| Conectar | true |
| Color Línea | 192,128,192 |
| Grosor | 1 |

TABLA XX. Aquí tenemos al cuerpo de prueba en un instante determinado de tiempo t . Esto es un complemento a la partícula de prueba en el estado inicial de la simulación, que se describió en la tabla XVIII.

| Vista | |
|---|-------------|
| Árbol de elementos | |
| Propiedades de partícula_moviendose (Forma) | |
| Posición y Tamaño | |
| Pos X | x |
| Pos Y | y |
| Tamaño X | 8 |
| Tamaño Y | 8 |
| En Píxeles | true |
| Visibilidad e Interacción | |
| Movible | true |
| Mueve Grupo | true |
| Sensibilidad | 50 |
| Aspecto Gráfico | |
| Estilo | ELLIPSE |
| Color Línea | BLACK |
| Color Relleno | 128,128,255 |

En la tabla XXI, se puede apreciar cómo se editan las propiedades de un panel situado en la primera ventana de la simulación. Dicho panel sirve a su vez, de contenedor de otros elementos gráficos como lo son los campos que muestran la posición del cuerpo de prueba conforme evoluciona la simulación, dando sus coordenadas polares, para cada instante de tiempo t . También aparecen tres botones que controlan el comportamiento de la simulación.

Hay un botón de inicio/pausa (play/pause), un botón de regreso (reset), y un botón que borra la traza del cuerpo de prueba, para empezar a volver a trazar su rastro desde la posición actual de evolución. Todos los paneles contenedores, los campos de muestra, y los botones de control se muestran editados en sus propiedades de las tablas XXI a la XXXIII.

TABLA XXI. Aquí se muestran las propiedades de un sencillo panel que servirá de contenedor para otros elementos gráficos como, campos numéricos y botones de control de la simulación.

| Vista | |
|------------------------------|-----------------|
| Árbol de elementos | |
| Propiedades de panel (Panel) | |
| Principal | |
| Distribución | FLOW:center,0,0 |
| Tamaño | "533,50" |
| Bordes | |
| Tipo Borde | RAISED_ETCHED |

TABLA XXII. Otro panel contenedor para los parámetros visibles de la simulación los cuales son la posición polar del cuerpo de prueba r , ϕ y el tiempo transcurrido t .

| Vista | |
|-----------------------------------|-----------------|
| Árbol de elementos | |
| Propiedades de parámetros (Panel) | |
| Principal | |
| Distribución | FLOW:center,0,0 |
| Bordes | |
| Tipo Borde | LOWERED_ETCHED |

TABLA XXIII. Un sencillo panel contenedor para otros elementos gráficos que muestran el valor de la coordenada polar r en forma instantánea para un tiempo dado t .

| Vista | |
|------------------------------------|---------------|
| Árbol de elementos | |
| Propiedades de r_{panel} (Panel) | |
| Principal | |
| Distribución | FLOW:left,0,0 |

TABLA XXIV. Aquí se muestra la etiqueta para una de las coordenadas polares r , del cuerpo de prueba.

| Vista | |
|--|---------|
| Árbol de elementos | |
| Propiedades de $r_{etiqueta}$ (Etiqueta) | |
| Principal | |
| Texto | " r = " |
| Alineación | RIGHT |

TABLA XXV. Aquí mostramos las propiedades del campo numérico para la coordenada polar r del cuerpo de prueba.

| Vista | |
|---|---|
| Árbol de elementos | |
| Propiedades de $r_{numerico}$ (CampoNumerico) | |
| Principal | |
| Variable | r |
| Inicial Value | r0 |
| Formato | "0.0#" |
| Editable | t==0 |
| Acción | x0=r0*Math.cos(phi0); y0=r0*Math.sin(phi0); arrastrar() |
| Aspecto Gráfico | |
| Columnas | 6 |
| Visible | true |
| Tamaño | "40,24" |
| Ayuda | "Coordenada radial calculada de la circunferencia reducida" |

TABLA XXVI. Mostramos las propiedades del campo numérico correspondiente a la coordenada polar ϕ del cuerpo de prueba.

| Vista | |
|--|---|
| Árbol de elementos | |
| Propiedades de $\phi_{numerico}$ (CampoNumerico) | |
| Principal | |
| Variable | phi |
| Inicial Value | phi0 |
| Formato | "0.0#" |
| Editable | t==0 |
| Acción | x0=r0*Math.cos(phi0); y0=r0*Math.sin(phi0); arrastrar() |
| Aspecto Gráfico | |
| Columnas | 6 |
| Visible | true |
| Tamaño | "40,24" |
| Ayuda | "Ángulo azimutal" |

TABLA XXVII. Aquí se enumeran las propiedades del campo numérico destinado a mostrar el tiempo transcurrido desde que se inicio la animación del cuerpo de prueba moviéndose alrededor de un agujero negro de Schwarzschild.

| Vista | |
|---|---------------------|
| Árbol de elementos | |
| Propiedades de $t_{numerico}$ (CampoNumerico) | |
| Principal | |
| Variable | T |
| Inicial Value | 0.0 |
| Formato | "0.0#" |
| Editable | False |
| Aspecto Gráfico | |
| Columnas | 10 |
| Visible | True |
| Tamaño | "40,24" |
| Ayuda | "tiempo a lo lejos" |

TABLA XXVIII. Aquí mostramos las propiedades de un sencillo panel llamado “controles”, y que alberga a varios botones de control de la animación.

| Vista | |
|----------------------------------|----------------|
| Árbol de elementos | |
| Propiedades de controles (Panel) | |
| Principal | |
| Distribución | FLOW:right,0,0 |

TABLA XXIX. Mostramos las propiedades del primer botón de control de animación llamado “play_pause”. Este es un botón de dos estados que corresponden al estado de “play” cuando corre la simulación, y el estado de “pause” cuando la simulación se interrumpe momentáneamente.

| Vista | |
|---|--|
| Árbol de elementos | |
| Propiedades de play_pause (BotonDosEstados) | |
| Principal | |
| Variable | _isPaused |
| Aspecto Gráfico | |
| Ayuda | “Comienza o para la simulación” |
| Propiedades Sí | |
| Imagen Sí | “/org/opensourcephysics/resources/controls/images/play.gif” |
| Acción Si | _play() |
| Color Sí | 200,220,208 |
| Propiedades No | |
| Imagen No | “/org/opensourcephysics/resources/controls/images/pause.gif” |
| Action No | _pause() |
| Color No | RED |

TABLA XXX. Este botón regresa toda la animación a su estado inicial mediante la función incorporada de EJS llamada _reset(). Al pulsar el botón se llama desde luego a esta función de “reseteado” de la animación.

| Vista | |
|---------------------------------|--|
| Árbol de elementos | |
| Propiedades de regresar (Boton) | |
| Principal | |
| Imagen | “/org/opensourcephysics/resources/controls/images/reset.gif” |
| Acción | reset(); t=0 |
| Aspecto Gráfico | |
| Ayuda | “Regresa la simulación” |

TABLA XXXI. Este botón de control es especial, ya que gracias a él, se puede borrar la traza, que hasta el momento de su pulsación, ha dejado el cuerpo de prueba al moverse alrededor del agujero negro.

| Vista | |
|-------------------------------------|--|
| Árbol de elementos | |
| Propiedades de limpiarTrazo (Boton) | |
| Principal | |
| Imagen | “/org/opensourcephysics/resources/controls/images/erase.gif” |
| Acción | _view.clearData() |
| Aspecto Gráfico | |
| Ayuda | “borra la trayectoria” |

TABLA XXXII. Propiedades de un simple panel que alberga a un botón de selección que a su vez controla la apertura o el cierre de la ventana secundaria de la simulación. En esta ventana secundaria se muestran en forma gráfica las componentes polares de la velocidad instantánea del cuerpo de prueba.

| Vista | |
|--------------------------------|----------------|
| Árbol de elementos | |
| Propiedades de grafico (Panel) | |
| Principal | |
| Distribución | FLOW:right,0,0 |

TABLA XXXIII. Aquí mostramos las propiedades de un cuadro de selección que controla la apertura o cierre de la ventana secundaria de la animación.

| Vista | |
|-------------------------------|-----------------------|
| Árbol de elementos | |
| Propiedades de vel (Selector) | |
| Principal | |
| Variable | vel |
| Seleccionada | true |
| Texto | “velocidades polares” |

En la tabla XXXIV se muestran las propiedades de la ventana secundaria de la simulación. Esta ventana es un elemento auxiliar gráfico, porque en ella se muestran dos gráficos, que corresponden a la velocidad radial y angular del cuerpo de prueba, y esto se muestra conforme va pasando el tiempo, y la simulación evoluciona. Cada tabla viene comentada. Por lo que no debería ser un problema, realizar la edición de las propiedades de los elementos gráficos respectivos.

TABLA XXXIV. Aquí mostramos las propiedades de la ventana secundaria de la animación. En ella se muestran las componentes de la velocidad instantánea que va adquiriendo el cuerpo de prueba conforme éste se mueve alrededor del hoyo negro de Schwarzschild. Como se puede notar su apertura se controla mediante el valor de la variable booleana “vel”.

| Vista | |
|--------------------------------------|--|
| Árbol de elementos | |
| Propiedades de velocidades (Ventana) | |
| Principal | |

| | |
|--------------------------|-----------------------|
| Título | “velocidades polares” |
| Distribución | Border |
| Visible | Vel |
| Posición y tamaño | |
| Posición | “318,109” |
| Tamaño | “400,400” |

| | |
|--------------------|-----------|
| Conectar | true |
| Color Línea | 64,128,64 |

TABLA XXXV. En la ventana secundaria de la simulación encontramos a un “panel con ejes”, donde se grafican los valores de las componentes polares de la velocidad instantánea que adquiere el cuerpo de prueba.

| | |
|--|---------------------------------|
| Vista | |
| Árbol de elementos | |
| Propiedades de coordenadas (PanelConEjes) | |
| Escalas | |
| Autoscala X | True |
| Autoscala Y | True |
| Decoración | |
| Título X | “tiempo ‘t’” |
| Título Y | “velocidades polares” |
| Pos Eje X | 0 |
| Título Y | “dr/dt (azul), dphi/dt (verde)” |
| Pos Eje Y | -0.5 |

TABLA XXXVI. En la gráfica mostrada en la ventana secundaria, existen dos gráficas. Aquí mostramos la traza de la primera de ellas que corresponde a la velocidad radial instantánea del cuerpo de prueba.

| | |
|--------------------------------------|--------|
| Vista | |
| Árbol de elementos | |
| Propiedades de rPrima (Traza) | |
| Entrada | |
| Entrada X | T |
| Entrada Y | rPrima |
| Aspecto Gráfico | |
| Max Puntos | 200 |
| No Repetir | True |
| Conectar | True |
| Color Línea | BLUE |

TABLA XXXVII. La gráfica que corresponde a la velocidad angular instantánea del cuerpo de prueba, debe de tener su lista de propiedades. Estas se muestran en esta tabla.

| | |
|--|----------|
| Vista | |
| Árbol de elementos | |
| Propiedades de phiPrima (Traza) | |
| Entrada | |
| Entrada X | T |
| Entrada Y | phiPrima |
| Aspecto Gráfico | |
| Max Puntos | 200 |
| No Repetir | true |

Recordar que varios de los valores aquí mostrados, pueden ser modificados de acuerdo a las características gráficas del monitor donde se ejecute a simulación. Esto es, los valores de los tamaños de los elementos gráficos deben de ser ajustados de acuerdo a las necesidades de cada monitor.

También cabe mencionar que varios de los valores de las variables dinámicas de la simulación, pueden cambiar, para modificar las condiciones iniciales del problema. De hecho, en una secuencia didáctica donde se esté empleando esta simulación (o una parecida a ella), es recomendable dejar que los mismos estudiantes sean los que las cambien. La finalidad de cambiar estos parámetros es de motivos de enseñanza, ya que así se logra hacer que el estudiante pueda observar diferentes comportamientos del fenómeno de movimiento del cuerpo de prueba. Así le será más fácil entender la teoría que yace detrás del fenómeno, y valorará el trabajo de hacer una simulación. También valorará la importancia que tiene la teoría General de la Relatividad para describir el movimiento planetario, como lo fue históricamente desde que la teoría fue publicada por Albert Einstein y la solución de agujero negro estacionario por Schwarzschild.

Una vez concluidos todos los cambios mostrados en esta sección, hemos prácticamente finalizado y acabado de crear la simulación de un cuerpo de prueba moviéndose alrededor de un agujero negro de Schwarzschild, con la ayuda de EJS. Resta mencionar, como se ve en marcha la simulación, y como empaquetarla en un archivo “jar” auto ejecutable de Java, para su uso en el aula de clases.

VII. CORRIDA DE LA SIMULACIÓN

Para correr la simulación recién escrita se necesita tan sólo oprimir un botón que se encuentra en el flanco derecho de la ventana principal de edición de EJS. Este es el botón de “play” ►. En cuanto se oprime el botón aparecen las dos ventanas de la simulación listas para ser utilizadas. La simulación puede empezar a correr cuando se oprime el botón de “play” de la ventana principal (la ventana donde se encuentra el horizonte de eventos del agujero negro mostrada con un círculo azul, y el cuerpo de prueba mostrado con otro círculo más pequeño). Las variables o parámetros que gobiernan en este caso, la evolución de la simulación, son los parámetros físicos, como la masa del agujero negro, la masa del cuerpo de prueba y la velocidad inicial que tenga este cuerpo. Además de su posición inicial. Se recomienda que durante una exposición en clase, el instructor vaya variando estos parámetros para que el estudiante aprecie los cambios producidos en el movimiento del cuerpo de prueba.

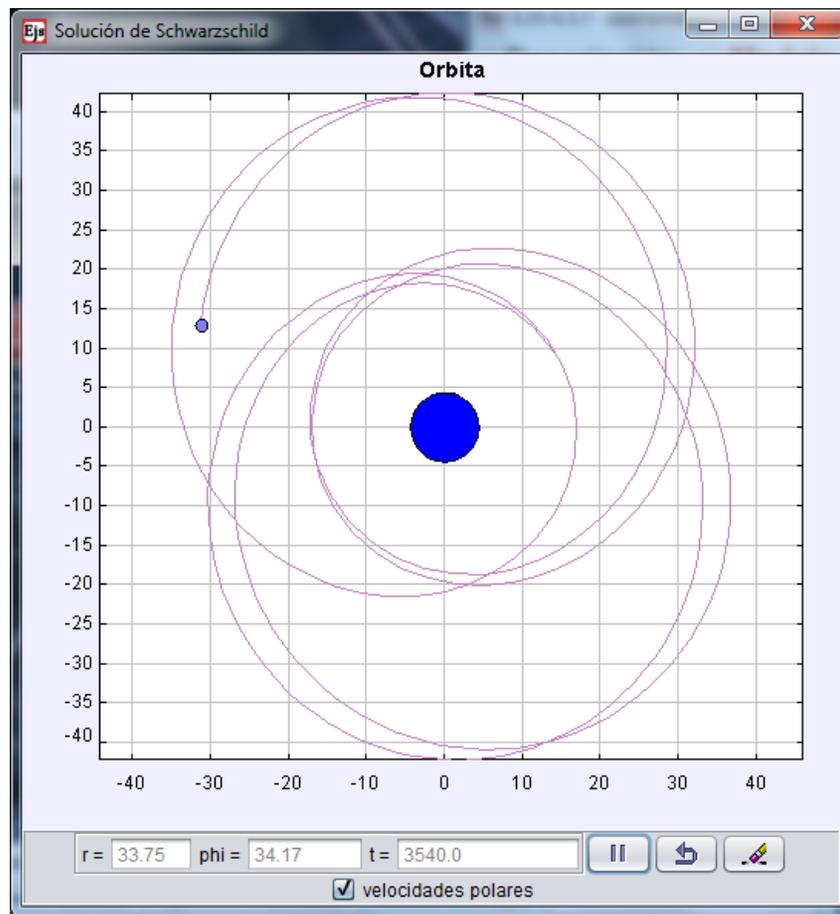


FIGURA 9. Aquí mostramos una corrida de la simulación con los valores de los parámetros dinámicos mostrados en el escrito. Esta es la ventana principal de la simulación. Se muestra en la parte superior una gráfica en coordenadas cartesianas de nombre “Orbita”. Aquí aparece el cuerpo de prueba moviéndose alrededor del horizonte de eventos mostrado como un círculo azul en el origen de coordenadas. La trayectoria se va trazando con una línea morada. En la parte de abajo se muestran los paneles contenedores con los valores r , ϕ , que son las coordenadas polares del cuerpo de prueba. También se muestra el tiempo transcurrido. A la derecha de estos “displays”, se encuentran botones de control. El primer botón (de izquierda a derecha) se usa para iniciar y pausar la animación, el segundo para reiniciarla y el tercero para borrar la traza de la trayectoria. En la parte de abajo hay un cuadro de selección activado, para mostrar la ventana secundaria.

VIII. EMPAQUETADO DE LA SIMULACIÓN

El empaquetar la simulación recién editada con EJS, no es difícil tampoco, basta con oprimir el botón de “comprimir” que se encuentra también en el costado derecho de la ventana principal de edición de EJS.

Una vez creado el archivo “jar” autoejecutable, éste se podrá correr al señalarlo y hacer doble click izquierdo con el ratón sobre de él. La forma estándar de almacenar el archivo “jar” de EJS indica que éste se encuentra en el subdirectorio “export” que está dentro del directorio EJS_#.#.#., que se produjo cuando se descomprimió el archivo “zip” de Easy Java Simulations (EJS) bajado de internet.

IX. UNA POSIBLE SECUENCIA DIDÁCTICA PARA EL SALÓN DE CLASES

Sokoloff y Thornton [12], son famosos por sus métodos de Aprendizaje Activo de la Física. En lo que nos concierne a

nosotros, podemos emplear su método de ocho pasos didácticos basados en lo que ellos llaman el ciclo PODS (predicción, observación, descripción y síntesis).

De esta forma, y siguiendo los pasos del Aprendizaje Activo de la Física, que tanto éxito ha reportado en ciertos lugares de Estados Unidos. Podemos sugerir una posible secuencia didáctica de acuerdo a este esquema y a las recomendaciones anteriores, para el manejo del código dentro del salón de clases.

1. El instructor describe el fenómeno del movimiento del cuerpo de prueba alrededor del hoyo negro de Schwarzschild, lo hace en el salón, sin ejecutar la simulación, ya previamente preparada.
2. Se le pide a los estudiantes que registren de manera personal sus predicciones en una hoja proporcionada para tal efecto, donde el estudiante apunta su nombre (los estudiantes se aseguran de que esta hoja no se califique, aunque para estimular su participación, se les recompensa con la asistencia y las actividades que lleven en las ILD, o Interactive Lecture Demonstration).

3. Los estudiantes se agrupan en pequeños grupos de dos o tres integrantes y discuten sus predicciones, y llegan a un acuerdo común.
4. El instructor elige las predicciones más comunes de toda la clase.
5. Los estudiantes registran sus resultados en una hoja de predicción común.
6. El instructor hace la demostración, del fenómeno del movimiento del cuerpo de prueba alrededor del agujero negro de Schwarzschild empleando la simulación numérica.
7. Algunos de los estudiantes, ahora discuten los resultados de la demostración con la simulación. Y llenan otra hoja similar a la de predicción, y ésta se pueda guardar para un futuro estudio.
8. Ahora los estudiantes discuten el fenómeno físico estudiado pero haciendo variar las condiciones iniciales de movimiento. Cambiando los parámetros dinámicos de la simulación.

Esta es la famosa secuencia didáctica de ocho pasos de Sokoloff y Thornton [10], pero adecuada al contexto del fenómeno de movimiento de un cuerpo de prueba alrededor de un hoyo negro de Schwarzschild. La simulación es un instrumento técnico conveniente para llevar a cabo esta secuencia didáctica de Aprendizaje Activo de la Física y es bastante flexible con ella una vez, que hemos editado la simulación. Ya que gracias a que hemos guardado ciertos valores iniciales de las variables dinámicas, éstas se pueden reeditar para modificarlas y así obtener varios ejemplos de movimiento del cuerpo de prueba alrededor del hoyo negro de Schwarzschild.

Se pueden hacer los tradicionales test de prueba preliminar (pretest) y los de prueba posterior al ejercicio (postest) para calcular la ganancia en el aprendizaje mediante el factor de ganancia de Hake. Esto demuestra la flexibilidad de la utilización de simulaciones en el aula de clase. Como el Aprendizaje Activo de la Física, ha tenido buenos resultados en diversos sitios de la Unión Americana. Esperamos que tenga buenos resultados en Latinoamérica, ajustando el método al contexto de la Física que se quiera estudiar. El cual es el objetivo del presente trabajo.

Esta forma de aprendizaje ha sido probada en otros contextos, basta señalar, por ejemplo, los trabajos de investigación educativa en el área de la Física Educativa de José Orozco [13] y de Lino Velázquez [14], donde se han utilizado las técnicas del Aprendizaje Activo. En estos trabajos se ha aplicado la secuencia didáctica de Sokoloff y Thornton pero no en Estados Unidos, sino en la ciudad de México, Distrito Federal., con resultados positivos para la técnica del Aprendizaje Activo de la Física.

X. CONCLUSIONES

Podemos decir, que el campo de las simulaciones representa un campo muy prometedor para el futuro del aprendizaje de la Física en las aulas de clase. En este trabajo, hemos mostrado

las bondades de una edición de una simulación, y cómo puede ser empleada en el aula de clases por el instructor del curso de Física. La forma en cómo estos recursos son y pueden ser aprovechados por los estudiantes, nos asegura una forma interesante y entretenida (o lúdica) de aprender y enseñar Física. En este ejemplo se muestra cómo puede ser llevada a cabo una actividad agradable para conseguir el conocimiento que yace detrás tanto de uno de los aspectos más importantes de la Teoría de la Relatividad General de Einstein, como del modelo de agujero negro de Schwarzschild. La clave para entender el movimiento planetario de nuestro sistema solar, viene a ser dada precisamente por el modelo matemático descrito por la solución de Schwarzschild. Tampoco debemos de menospreciar una buena técnica de enseñanza en la Física, por eso hemos tratado de abordar este punto con un posible esquema de una secuencia didáctica basada en el modelo de Aprendizaje Activo de la Física según Sokoloff y Thornton. Esperamos que el trabajo sea de utilidad para la enseñanza creativa y lúdica de este tema tan apasionante de la Física dentro de las aulas reales de clases.

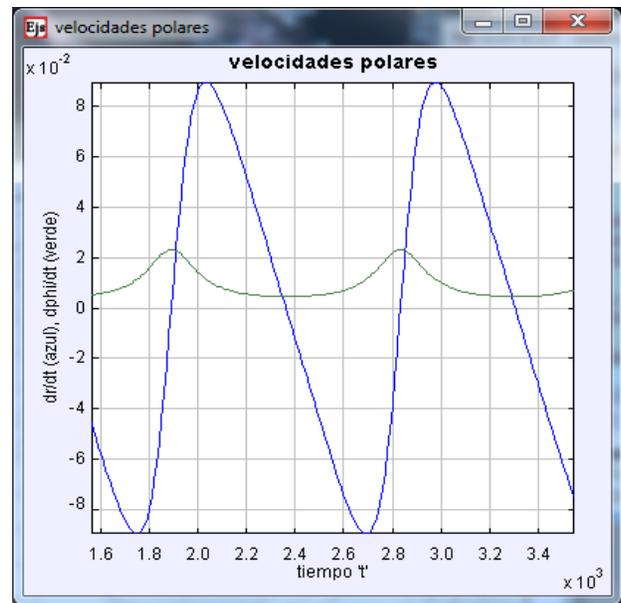


FIGURA 10. Esta es la ventana secundaria de la simulación, donde se muestran dos diagramas superpuestos para la velocidad radial instantánea y la velocidad angular instantánea del cuerpo de prueba. El eje horizontal se usa para mostrar el tiempo transcurrido. Esta ventana aparece cuando el cuadro de selección “velocidades polares” de la primer ventana está activa. Aparece inicialmente abierta, ya que en la tabla 33 se editó la propiedad “Seleccionada” como “true” (verdadera).

AGRADECIMIENTOS

El autor quiere expresar su agradecimiento al Consejo Nacional de Ciencia y Tecnología (México), por el apoyo otorgado en la realización de este trabajo. Este trabajo fue

REFERENCIAS

- [1] Gray, C. G., Poisson, E., When action is not least for orbits in general relativity, *American Journal of Physics* **79**, 43-56 (2011).
- [2] Gray, C. G., Poisson, E., *Einstein Cannon*, <<http://www.compadre.org/OSP/items/detail.cfm?ID=10456>>, consultada el 31 de octubre de 2012.
- [3] Gray, C. G., Poisson, E., *Particle Worldline*, <<http://www.compadre.org/osp/items/detail.cfm?ID=10457>>, consultada el 31 de octubre de 2012.
- [4] Esquembre, F., *Easy Java Simulations (EJS)*, <<http://fem.um.es/Ejs>>, consultado el 31 de octubre de 2012.
- [5] Esquembre, F., *Creación de Simulaciones Interactivas en Java, Aplicación a la Enseñanza de la Física*, (Pearson, Prentice Hall, 1ra. Edición, 2005).
- [6] Glendenning, N. K., *Special and General Relativity with Applications to White Dwarfs, Neutron Stars and Black Holes*, (Springer Science + Bussines Media, LLC, Astronomy & Astrophysics Library, 1ra. Edición, 2007).
- [7] Susskind, L., Lindesay, J., *An introduction to Black Holes, Information and the String Theory Revolution. The Holographic Universe*, (Worl Scientific Publishing Co. Pte. Ltd., 1ra. Edición, 2005).
- [8] Camenzind, M., *Compact Objects in Astrophysics, White Dwarfs, Neutron Stars and Black Holes*, (Springer Verlag Berlin Heidelberg, Astronomy & Astrophysics Library, 1ra. Edición, 2007).
- [9] Weinberg, S., *Gravitation and Cosmology, Principles and Applications of the General Theory of Relativity*, (John Wiley & Sons, Inc., 1ra. Edición, 1972).
- [10] Goldstein, H., *Classical Mechanics*, (Addison-Wesley Publishing Company, 2da. Edición, 1980), p. 72, Ecs. 3-7.
- [11] Wald, R. M., *General Relativity*, (University of Chicago Press, 1ra. Edición, 1984).
- [12] Sokoloff, D. R., Thornton, R. K., *Interactive Lecture Demonstration, Active Learning in Introductory Physics*, (Wiley & Sons, Inc., 1ra. Edición, USA, 2006).
- [13] Orozco-Martínez, J., *El teléfono celular como recurso didáctico en el álgebra vectorial para la Física en el Nivel Medio Superior*, Tesis de Maestría en Ciencias en Física Educativa, Instituto Politécnico Nacional, (2012).
- [14] Velázquez-Arteaga, L. J., *Aprendizaje Activo para las Leyes de Newton a Nivel Medio Superior*, Tesis de Maestría en Ciencias en Física Educativa, Instituto Politécnico Nacional, (2012).