DESCARTES

matemáticas interactivas

Documentación de *Cuerpos En Movimiento*

Alejandro Radillo Díaz

José Luis Abreu León

Joel Espinosa Longi

10 de octubre de 2021

Índice general

1

Sobre la herramienta

1.1. ¿Qué es *Cuerpos en Movimiento*?

Cuerpos en Movimiento, abreviado *CenM*, es una escena digital interactiva creada en *DescartesJS*, que simula modelos mecánicos mediante el formalismo Euler-Lagrange de la mecánica analítica. Los archivos para su uso, así como algunos modelos prediseñados, se pueden descargar de https://descartes.matem.unam.mx/ejemplos/CuerposEnMovimiento.zip

Como cualquier escena interactiva de *DescartesJS*, puede correrse en el editor mismo de *DescartesJS*, o bien como un archivo *html* en un navegador.

Más allá de un objeto lúdico, *CenM* ha sido creado con el objeto de favorecer que profesores y alumnos, tanto de licenciatura como de preparatoria, tengan una herramienta más para profundizar en el conocimiento de la mecánica clásica y analítica.

Como se menciona en el apartado que trata sobre *la presente documentación*, tanto esta documentación como el programa al que documenta requieren dominio de algunos temas algo avanzados. En este sentido, está más dirigido a nivel licenciatura, pero puede llegar a ser usado también para explicar algunos conceptos básicos de mecánica a nivel preparatoria. Esto se puede consultar en el apartado sobre *posible usos de Cuerpos en Movimiento*.

Es importante también conocer algo sobre el programa en que fue creado este simulador mecánico. Ello se puede consultar en el apartado *sobre DescartesJS*.

1.2. Sobre *DescartesJS*

DescartesJS es un editor de escenas digitales interactivas libre, y que se puede descargar de https://descartes.matem.unam.mx. Las escenas generadas en dicho programa son archivos *html* que pueden visualizarse en un navegador. *Cuerpos en Movimiento* es una escena de este tipo.

Se sugiere al usuario que busque conocer más sobre el funcionamiento interno de *Cuerpos en Movimiento*, e inclusive aventurarse a modificar el código, adentrarse antes en *DescartesJS*. Para dicho propósito, puede consultar la documentación técnica de dicho editor en https://descartes.matem.unam.mx/doc/DescartesJS.

1.3. Cómo lanzar *Cuerpos en Movimiento*

El archivo *CuerposEnMovimiento.html* es el responsable de lanzar la herramienta. Es un archivo generado en *DescartesJS*. Aunque es un archivo de navegador, hay algunos detalles a considerar para poder lanzarlo correctamente.

Algo importante a tener en mente es que, debido a cambios recientes de privacidad de los navegadores, no se permite que un archivo de navegador tenga acceso a archivos locales del ordenador, a menos que esto ocurra dentro de un servidor. Y *Cuerpos en Movimiento* requiere de archivos asociados como se describe en el apartado sobre el capítulo 5.

Así, si se desea usar la herramienta dentro de un servidor, simplemente se hace referencia al archivo *html*. Por ejemplo, con el vínculo https://descartes.matem.unam.mx/ejemplos/CuerposEnMovimiento/CuerposEnMovimiento.html.

Sin embargo, si se intenta abrir el archivo de forma local en un ordenador, no será posible. Para usarlo localmente, es preciso abrirlo dentro de la herramienta *DescartesJS*. Posteriormente, si de desea, se puede subir a un servidor, con todos sus archivos asociados, y desde ahí será posible abrirlo en un navegador.

1.4. Sobre la presente documentación

La presente documentación está destinada principalmente a profesores y alumnos interesados en abordar aspectos de la mecánica analítica, y a generar propios modelos mecánicos. Para poder lograrlo, y para entender el funcionamiento del simulador, es preciso contar con un cierto dominio de la mecánica analítica.

Dos aspectos fundamentales son el formalismo de Euler-Lagrange y el método Runge-Kutta de orden 4 para integrar numéricamente, mismos que se tratan en el apartado sobre *el motor de Cuerpos en Movimiento*.

Esta documentación no es estática. *Cuerpos en Movimiento* está en constante cambio. Ya sea por mejoras o corrección de errores, dicho programa cambia constantemente. De tal suerte que la presente documentación también está sujeta a cambios. Por lo mismo se sugiere al usuario descargar de forma periódica nuevas versiones de la documentación del sitio https://descartes.matem.unam.mx/doc/CenM.

2

Usando *Cuerpos en Movimiento*

En este capítulo se trata el uso como tal de la herramienta *Cuerpos en Movimiento*. Por lo mismo, se enfoca en el uso de los controles disponibles.

En la Figura 2.1 se muestra la herramienta al abrir.

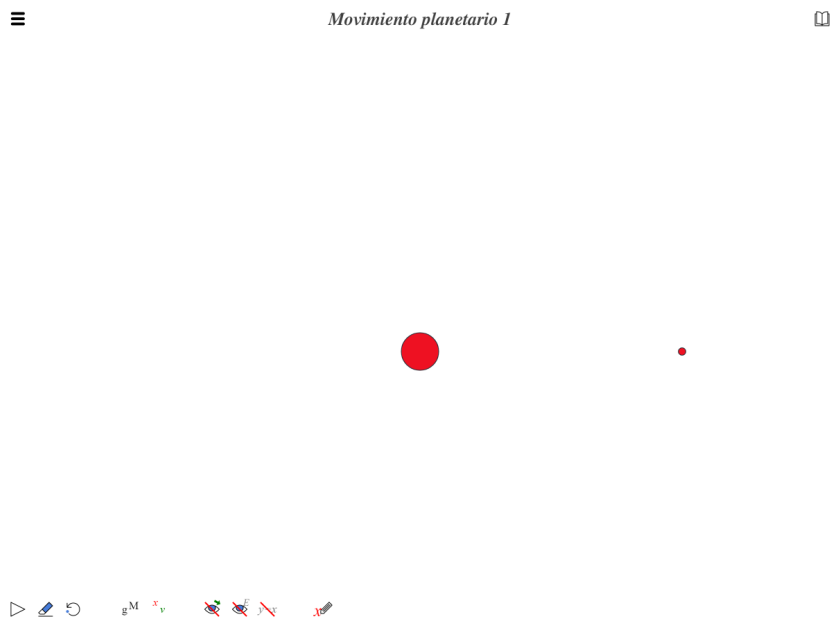


Figura 2.1: *Cuerpos en Movimiento* al abrir.

En el margen superior se muestra centrado el título del modelo en cuestión. Este margen también alberga un botón de menú en su extremo izquierdo, mediante el cual el usuario puede elegir un modelo de la lista de modelos disponible. Los archivos individuales de cada modelo se encuentran en la carpeta *models*.

En el margen inferior se encuentran varios botones con funciones que van de controles de animación a visualización y edición del sistema mecánico.

2.1. Los controles de *Cuerpos en Movimiento*

- ***Menú***. Este botón cubre la pantalla con una página que enlista los sistemas dinámicos disponibles, como se puede observar en la Figura 2.2.

Figura 2.2: El menú de sistemas dinámicos lanzado por el botón de *menú*.

Como se puede observar, este botón está también presente en el extremo superior izquierdo del menú en caso que se desee cerrar el mismo.
En el menú basta pulsar en algún modelo para cerrar el menú y cargar el modelo pulsado. Si se visita nuevamente el menú, el modelo visitado se encuentra remarcado por un recuadro amarillo.

- ***Documentación técnica***. Este botón lanza en un navegador la documentación técnica de la herramienta, que es el mismo documento que se está revisando, y cuya dirección en la red es https://descartes.matem.unam.mx/doc/CuerposEnMovimiento.

- / ***Animar / Pausar***. Este par de botones, que ocupan el primer lugar en el margen inferior, se usan para animar / pausar el sistema.
Cuando hay una animación en curso, los botones *Borrador*, *Reinicio*, *Introducción de parámetros*, *Introducción de condiciones*, y el *Modo de edición gráfica* se encuentran inactivos; y el botón *pausar* se encuentra visible.

- ***Borrador***. Algunos objetos del sistema mecánico se encuentran dispuestos para dejar un rastro de su trayectoria. Este botón se encarga de borrar cualquier rastro así dejado.

- ***Reinicio***. Transcurridos cambios debidos a una animación, este botón regresa a las últimas condiciones del sistema dictadas por el usuario antes de haber animado.

- gM ***Introducción de parámetros***. Este botón lanza un panel, como el mostrado en la Figura 2.3, donde se pueden visualizar los parámetros del sistema. Los parámetros típicamente involucran elementos como las masas, constantes tales como de resorte o de gravitación, e inclusive abreviaturas de cantidades.

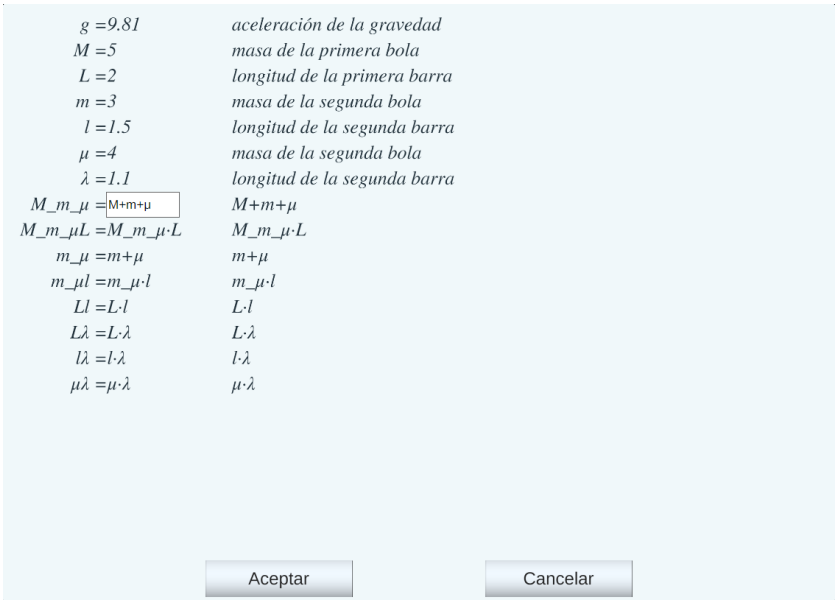


Figura 2.3: El panel de parámetros. Obsérvese el campo de texto de edición activo para uno de los parámetros.

Obsérvese que el pánel de introducción de parámetros incluye datos del sistema. Tanto el nombre de los parámetros, como sus valores y su descripción son tomados del archivo del modelo en cuestión.
Es posible modificar el valor de alguno de ellos pulsando en su valor actual. Con ello aparece un campo de texto, que se puede editar, pulsar *ENTER* y luego el botón *Aceptar* del panel, con lo que el campo queda actualizado.
Mientras se muestra este panel, todo botón ajeno al mismo queda oculto. Por lo mismo, es preciso cerrarlo antes de poder usar cualquiera de ellos.
Los parámetros involucrados, así como sus datos de valor y descripción, no se definen de origen en la herramienta como tal, sino en el *bloque PARAMETERS* del archivo de modelo.

- xv ***Introducción de condiciones***. Este botón lanza un panel, similar al de parámetros, pero en donde se pueden consultar y editar las condiciones del sistema. Por condiciones se entienden las coordenadas y velocidades del sistema. En la Figura 2.4 se muestra un ejemplo de dicho panel.
Al igual que para el panel de parámetros, es posible editar el valor de alguna variable pulsando en el mismo. Con ello se despliega el campo de texto correspondiente, que

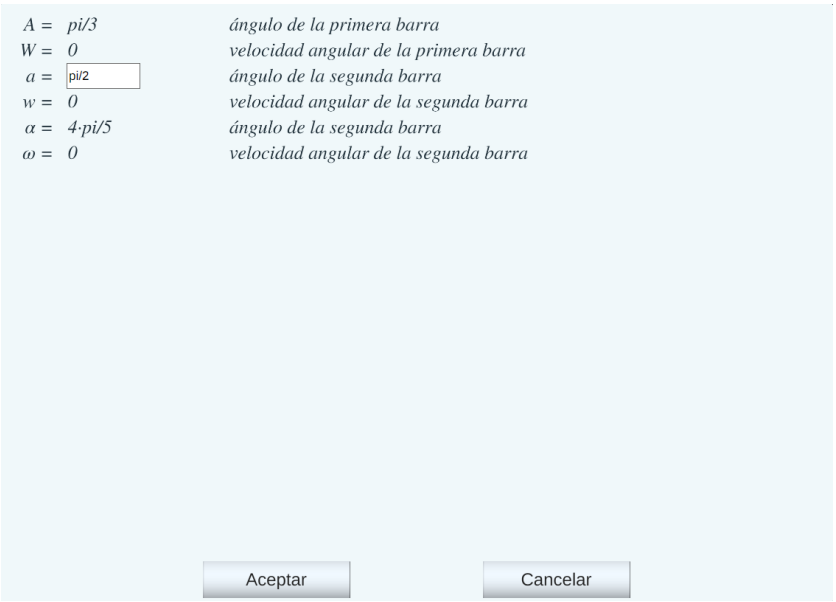


Figura 2.4: El panel de condiciones del sistema.

se usa igual que en el otro panel para cambiar algún valor.
También como el otro panel, éste cubre todos los botones. De tal suerte que hay que cerrarlo antes de poder echar mano de alguno de ellos.
Las condiciones involucradas, así como sus datos de valor y descripción, no se definen de origen en la herramienta como tal, sino en el *bloque CIs* del archivo de modelo.

- / ***Mostrar / Ocultar vectores de velocidades***. Cuando el ojo sin tache está visible en el botón, se trazan flechas correspondientes a las velocidades de los objetos. Cuando se pinta el ojo tachado, dichas flechas no se trazan.
Es útil en ocasiones trazar las velocidades para entender mejor el comportamiento de un sistema. Sin embargo, si llegan a estorbar, se puede ocultar con este botón.

- / ***Mostrar / ocultar panel de energías***. Cuando se muestra el ojo sin tachar, también se muestra un panel con información sobre las energías, como el de la Figura 2.5.

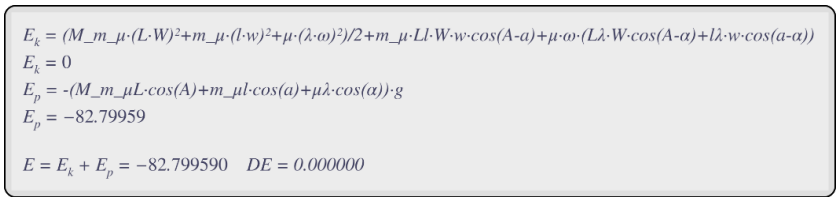
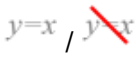


Figura 2.5: Ejemplo del panel con información sobre las energías.

Este panel muestra la **expresión** de la energía cinética (E_k) y potencial (E_p) del sistema. Por expresión nos referimos a la variables de las que depende y cómo se operan. Adicionalmente, se muestra el **valor numérico** de ambas energías. Y, finalmente, también muestra la suma de ambas ($E = E_k + E_p$). El valor *DE* corresponde al error máximo numérico cometido durante el proceso de integración. Esto es, la diferencia absoluta máxima respecto al valor de la energía total del sistema previo a la animación. Para todos los valores se muestra un máximo de seis cifras significativas.
Las expresiones de las energías se proporcionan en el *bloque de energías*, dentro del *archivo .dat del modelo*.

-  ***Mostrar / ocultar valores del sistema***. Si el ícono del botón aparece sin tache, se muestra un panel con los valores actuales de las variables del sistema, como en la Figura 2.6.

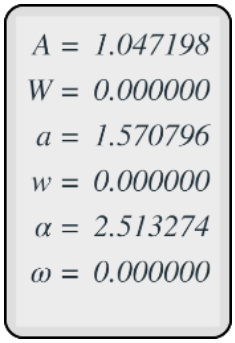


Figura 2.6: Ejemplo de una parte del panel con valores numériocs del sistema.

Estas variables son las mismas del *panel de introducción de condiciones*. Sólo que en este panel, los valores de dichas variables no son editables.
En ocasiones puede ser útil ver los valores numéricos de una variable en tiempo real durante una simulación. Es para ello que se utiliza este panel.

-  ***Modo de edición gráfica***. Este botón alterna también entre dos modos. Si el botón muestra la ***x***, es posible editar gráficamente las **coordenadas** de los objetos móviles. Si muestra ***v***, es posible editar gráficamente las velocidades.
Este botón se encuentra inactivo durante una animación, por lo que es preciso detenerla si está en curso y se desea alterar algo en el sistema.
En el modo de edición de coordenadas, es posible pulsar y arrastrar los objetos declarados como móviles. Es decir, aquellos cuya posición se encuentra definida por medio de variables en el *panel de condiciones del sistema*.
En el modo de edición de velocidades, éstas aparecen como flechas, cuyas puntas se pueden pulsar y arrastrar también.

2.2. Detalles a considerar sobre el uso de *Cuerpos en Movimiento*

A continuación se mencionan algunos puntos importantes sobre los controles recientemente descritos y el uso de la herramienta.

2.2.1. Sobre los distintos modos de edición de condiciones del sistema

Las condiciones del sistema se pueden editar de dos formas: mediante el botón de *introducción de condiciones*, o bien mediante el *modo de edición gráfica*, ya sea de coordenadas o velocidades. El modo de edición gráfica permite *mover manualmente* los objetos, lo cual es deseable cuando se quiere hacer un cambio rápido en el sistema. Estos cambios ahorran al usuario tener que, por ejemplo, descomponer mentalmente un vector en coordenadas. No obstante, de quererse hacer un cambio fino y preciso en algún valor, ello se logra mejor mediante el panel de introducción de condiciones. De cualquier forma, hay que tener presente que cuando se hace un cambio mediante cualquiera de las dos formas, ello se verá reflejado en la otra forma también.

Los valores de coordenadas, velocidades y energías mostrados en *Cuerpos en Movimiento* **no** cuentan con unidades, con el fin de dejar más libertad al usuario de manejar las de su elección. Así pues, cuando se imprime un valor, no debe sorprender que sólo aparece el número en cuestión.

2.2.2. Sobre las flechas de velocidades

Las flechas mostradas tanto en el modo de *mostrado de vectores de velocidades* como en el modo de *modo de edición gráfica* cuando se encuentra en el modo de velocidades, son de caracter representativo. Esto es, la longitud de una flecha representa una velocidad, y una crece proporcional a la otra. De tal forma que no debe tomarse la longitud en unidades dadas de una flecha directamente como el valor de una velocidad.

Por otra parte, las velocidades tan pequeñas como para ser consideradas despreciables no se muestran. Y, en el caso de edición gráfica de velocidades, si un objeto muestra un punto en lugar de una flecha, éste representa a una flecha de tamaño casi nulo. Por lo mismo, se puede pulsar y arrastrar para modificar gráficamente a la velocidad en cuestión.

En modelos que dependen de una barra, como los tipo pendulares, las velocidades de los extremos de la barra están constreñidas a velocidades angulares. Es decir, las flechas arrastrables asociadas a las velocidades están constreñidas a ser siempre perpendiculares a la barra misma, aún cuando las velocidades reales (aquellas mostradas en el modo *mostrar vectores de velocidades*) no respetan dicha constricción.

2.2.3. Parámetros auxiliares

Es posible introducir expresiones auxiliares como parte de los parámetros usando el botón *introducción de parámetros*. Ello se realiza en el archivo del modelo, por lo que se aborda más a fondo en el contenido de dicho archivo. Sin embargo, se hace la mención aquí pues se considera pertinente el usuario se percate de esta posibilidad.

Para ejemplificar la utilidad de esto, considere el péndulo triple, en el que la energía potencial depende de un término $(M+m+\mu)L$. En cada paso de integración, es necesario calcular primero esta triple suma y luego la multiplicación por L. Y este valor no cambiará, pues depende sólo de constantes. Así, resulta útil definir un parámetro $M\_m\_\mu L$ que albergue esta información. Funciona algo así como una variable del programa que no altera su valor dadas ciertas condiciones. Y ahorra operación al animar el sistema, lo que se traduce en un mejor desempeño al animar el sistema.

Además de poder introducir expresiones como la mencionada, también es posible usar una notación alternativa para operaciones y letras correspondientes a variables. Por ejemplo, es posible usar el punto sustituyendo al asterisco para el producto. O usar un superíndice 2 directamente sin usar la notación ^2 para un cuadrado. Es posible usar letras griegas como variables, como lo es μ en el ejemplo. Todo esto ayuda a que las expresiones de parámetros, valores del sistema y energías puedan ser más estéticas y legibles matemáticamente. Pero es importante mencionar que hay que usar los caracteres disponibles y específicos de *DescartesJS*. En este sentido, se recomienda al usuario que busque echar mano de esta funcionalidad, que revise los modelos incluidos en el archivo *zip* del contenido de la herramienta, y que de ellos extraiga los caracteres para sus modelos propios.

3

El motor de *Cuerpos en Movimiento*

En este apartado se trata el funcionamiento interno de cálculo de *Cuerpos en Movimiento*. Es decir, el *motor* del mismo.

Debido a la periodicidad de su uso en el presente apartado, recordamos al usuario que la notación de un punto sobre una variable corresponde a la derivada de la misma respecto del tiempo. Por ejemplo, $\dot{a} = \frac{d}{dt}a$.

3.1. Archivos de biblioteca

Dentro de los archivos de instalación, hay una carpeta *biblioteca*, la cual contiene archivos de texto correspondientes a bibliotecas generadas para *DescartesJS*. Se recuerda que las bibliotecas se accesan, para un archivo *html* correspondiente a una escena interactiva de *DescartesJS*, en el selector *Definiciones*. El apartado sobre *carpeta biblioteca* en el capítulo 5 contiene información más detallada de cada biblioteca.

3.2. El formalismo de Euler-Lagrange

Cuerpos en Movimiento echa mano del formalismo de *Euler-Lagrange* en la evolución temporal de los modelos dinámicos. La ecuación de Euler-Lagrange nos dice:

$$\frac{\partial L}{\partial q_i} - \frac{d}{dt}\frac{\partial L}{\partial \dot{q}_i} = 0 \tag{3.2.1}$$

En la ecuación 3.2.1, L es el lagrangiano $E_k - E_p$ (la diferencia de la energía cinética menos la potencial), q_i es la i-ésima coordenada generalizada y $\dot{q}_i$ es la velocidad generalizada (o bien, la derivada temporal) correspondiente a la i-ésima coordenada generalizada.

Cuando se requieren derivadas numéricas del lagrangiano respecto a una coordenada y/o velocidad generalizada, éstas se hacen centrales y tomando un intervalo de $2\epsilon = 2 \cdot 10^{-6}$. Por ejemplo, para la i-ésima coordenada:

$$\frac{\partial L}{\partial q_i} = \frac{L(q_i + \epsilon) - L(q_i - \epsilon)}{2\epsilon} \tag{3.2.2}$$

3.3. La integración numérica

Para la evolución del sistema en el tiempo, el usuario indica el paso de tiempo dt para cada modelo. Cada paso de la animación dista del anterior en dt. No obstante, el programa realiza, dentro de cada paso interno de la animación, N pasos de integración, donde N también es especificado para cada modelo físico. De esta forma, el programa no gasta recursos de cálculo pintando cada paso de la integración. En su lugar, hace una integración muy fina (con diferenciales de tiempo $\delta = \frac{dt}{N}$), pero sólo pinta cambios cada N pasos de la integración numérica.

Cada paso de la integración numérica del programa implementa un método de Runge-Kutta de orden 4 para varias variables, donde dichas variables son las coordenadas generalizadas y sus momentos correspondientes. Primero que nada, es necesario guardar el estado del sistema en el tiempo en donde se inicia el paso de la integración. Considerando que el sistema tiene n coordenadas generalizadas, los valores de las coordenadas y momentos generalizados se guardan en un vector X de $2n$ entradas. Las primeras n entradas se reservan para las n coordenadas generalizadas. Las últimas n entradas son para los n momentos generalizados. Por ejemplo, el valor de la i-ésima coordenada y del i-ésimo momento al inicio del paso de integración se guardan en X_i y X_{i+n}, respectivamente. Los valores de las coordenadas generalizadas en un determinado tiempo se guardan en un vector q de n entradas; mientras que los momentos se guardan en un vector p también de n entradas.

Es importante notar que, dado que al principio sólo se cuenta con los valores de coordenadas generalizadas (guardadas en el vector q ya mencionado) y velocidades generalizadas (guardadas en un vector qv de n entradas), los momentos generalizados habrán de calcularse a partir de las velocidades generalizadas. Para ello se echa mano de una matriz cuadrada T, compuesta de los coeficientes para convertir un vector de velocidades en uno de momentos. Es decir, el i-ésimo momento está dado por $p_i = T_{ij}\dot{q}_j$. Dicha matriz se define en *el bloque de la matriz T*, dentro del *el archivo .dat del modelo*.

Se extraen los valores de los coeficientes de k para cada coordenada y velocidad generalizada. Para un sistema con n coordenadas generalizadas, se tiene una lista de funciones como la siguiente:

$$\begin{aligned}
\dot{q}_1 &= f_1(t, q_1, \ldots, q_n, p_1, \ldots, p_n) \\
&\vdots \\
\dot{q}_n &= f_n(t, q_1, \ldots, q_n, p_1, \ldots, p_n) \\
\dot{p}_n &= f_{n+1}(t, q_1, \ldots, q_n, p_1, \ldots, p_n) \\
&\vdots \\
\dot{p}_n &= f_{2n}(t, q_1, \ldots, q_n, p_1, \ldots, p_n)
\end{aligned} \tag{3.3.1}$$

Es sobre dicha lista que se aplicará el método de Runge-Kutta, con el fin de obtener una evolución temporal para las q y las p. Así, para cada variable (ya sea coordenada o momento generalizado) se calculan sus 4 k:

$$\begin{aligned} k_1 &= f(t, \boldsymbol{y}) \\ k_2 &= f\left(t + \frac{h}{2}, \boldsymbol{y} + \frac{h}{2}k_1\right) \\ k_3 &= f\left(t + \frac{h}{2}, \boldsymbol{y} + \frac{h}{2}k_2\right) \\ k_4 &= f(t + h, \boldsymbol{y} + hk_3) \end{aligned} \tag{3.3.2}$$

En la ecuación 3.3.2, $\boldsymbol{y}$ es un vector que representa a todas las coordenadas y momentos generalizados explícitamente marcados en la ecuación 3.3.1.

Como para cada coordenada generalizada y velocidad generalizada se requieren 4 valores de k según el método de Runge-Kutta, en el programa éstos se guardan en un arreglo bidimensional K. La primera entrada de cada arreglo indica cuál de las cuatro k se está considerando. La segunda entrada del arreglo lleva el número de coordenada generalizada, o bien del momento generalziado en cuestión. Similar al vector X, si la segunda entrada de K está entre 1 y n, corresponde a una coordenada generalizada. Si es mayor que n, corresponde a un momento. Por ejemplo, $K_{1,3}$ lleva el valor de k_1 para la coordenada generalizada q_3, mientras que $K_{1,3+n}$ guarda el valor de k_1 del momento p_3, si el sistema tiene n coordenadas generalizadas.

En general, la "*pendiente*" de evolución temporal para una coordenada generalizada se obtiene sólo como la velocidad generalizada correspondiente. Por otra parte, aquella para la evolución temporal de un momento generalizado se obtiene como la derivada del lagrangiano respecto a la coordenada generalizada correspondiente a la velocidad en cuestión, según la ecuación 3.2.2. Ello se justifica considerando que, de la ecuación 3.2.1 tenemos que $\frac{d}{dt}p_i = \frac{\partial L}{\partial q_i}$.

Una vez obtenidos las K iniciales para cada coordenada y momento generalizados, se usan para calcular la evolución temporal para un diferencial del tiempo $\delta/2$. Por ejemplo, el valor de la i-ésima coordenada generalizada se obtiene sumando el valor guardado de la misma con el valor de la $K_{1,i}\frac{\delta}{2}$. Es decir, $q_i = X_i + K_{1,i}\frac{\delta}{2}$. El valor del i-ésimo momento generalizado se obtiene sumando su valor guardado con $K_{1,i+n}\frac{\delta}{2}$. Es decir, $p_i = X_{i+n} + K_{1,i+n}\frac{\delta}{2}$. Posteriormente, se obtienen las velocidades generalizadas a partir de los momentos generalizados, previamente invirtiendo la matriz de conversión T. Es decir, $\dot{q}_i = T^{-1}_{ij}p_j$. Se puede consultar un ejemplo concreto el uso de la matriz T para la conversión entre momentos y velocidades generalizads en el apartado sobre *un ejemplo concreto del uso de la matriz T*.

Nótese que se obtuvieron estimaciones intermedias de las coordenadas, momentos, y velocidades generalizadas (las q guardadas internamente en el vector q, los p en el vector p, y las $\dot{q}$ en el vector qv), así como el valor k_1 para coordenadas y momentos. Esto es, al

terminar este paso, las coordenadas y momentos q y p quedan con valores correspondientes a una evolución en el tiempo de $\frac{\delta}{2}$. Cada q y p queda con el valor tras hacer un paso simple de integración de tamaño $\frac{\delta}{2}$ usando la k_1 para cada variable. Es decir, quedan listas para ser usadas para el cálculo de la k_2 de cada variable.

Ya contando con las estimaciones intermedias usando k_1, se pueden usar para calcular k_2 para todas las coordenadas y velocidades, repitiendo el proceso, pero considerando que ahora las q y p no son las originales guardadas en el vector K, sino los valores tras hacer un paso de integración de tamaño $\frac{\delta}{2}$. Esto asegura que, al ahora calcular k_2, los argumentos de la función f correspondiente no son $t, q_1, \ldots, q_n, p_1, \ldots, p_n$, como en el cálculo original para k_1, sino $t+\frac{\delta}{2}, q_1+\frac{\delta}{2}k_1, \ldots, q_n+\frac{\delta}{2}k_1, p_1+\frac{\delta}{2}k_1, \ldots, p_n+\frac{\delta}{2}k_1$. Dado que es la k_2 de cada coordenada y momento la que se está calculando, los valores ahora se guardarán en $K_{2,i}$ para la k_2 de la i-ésima coordenada generalizada y en $K_{2,i+n}$ para la k_2 del momento generalizado correspondiente. Ya que se tienen los valores de k_2, nuevamente se estiman las q y las p tomando el valor original guardado (en K) y sumándole su k_2 correspondiente por un diferencial de tiempo nuevamente de $\frac{\delta}{2}$. Y se vuelven a recalcular las velocidades generalizadas para este paso a partir de sus momentos correspondientes usando T^{-1}.

Todo este proceso se repite otra vez para calcular las k_3 de todas las variables, una vez contando con la estimación de las q y las p obtenida a partir de k_2. La k_3 de la i-ésima coordenada y momento se guardan, respectivamente, en $K_{3,i}$ y $K_{3,i+n}$. Las q y p se calculan nuevamente, sólo que ahora se toma diferencial de tiempo de δ, ya que los valores de las q y p se usarán ahora para calcular k_4, el coeficiente para el final del intervalo.

En el último paso se calcula k_4 de la misma forma, sólo que ahora se consideró la estimación correspondiente a un paso de integración de tamaño δ.

Para cada variable (coordenada o momento generalizado) se calcula su evolución temporal a partir de sus cuatro k correspondientes con la fórmula de Runge-Kutta. Por ejemplo, para la i-ésima coordenada generalizada y su momento correspondiente se tendría:

$$\begin{aligned} q_{i+1} &= q_i + \frac{\delta}{6}\left(K_{1,i} + 2K_{2,i} + 2K_{3,i} + K_{4,i}\right) \\ p_{i+1} &= p_i + \frac{\delta}{6}\left(K_{1,i+n} + 2K_{2,i+1} + 2K_{3,i+n} + K_{4,i+n}\right) \end{aligned} \tag{3.3.3}$$

3.3.1. La matriz T: Un ejemplo concreto

A manera de ejemplo, considere la expresión de energía cinética del *péndulo doble*:

$$E_k = \frac{1}{2}\left((M+m)(L\dot{q_1})^2 + 2mLl\cos(q_1-q_2)\dot{q_1}\dot{q_2} + m(l\dot{q_2})^2\right) \tag{3.3.4}$$

En la ecuación 3.3.4, M es la masa conectada a la varilla principal, m es la de la conectada a la secundaria, L es la longitud de la varilla principal, l la de la secundaria, q_1

es la primera coordenada generalizada (el ángulo vertical de la varilla principal), q_2 es la segunda coordenada generalizada (el ángulo vertical de la varilla secundaria), $\dot{q}_1$ la velocidad angular de la varilla principal, y $\dot{q}_2$ la de la secundaria.

A partir de 3.2.1 tenemos que, para este ejemplo,

$$\begin{aligned} p_1 &= \frac{\partial E_k}{\partial \dot{q}_1} = (M+m)L^2\dot{q}_1 + 2mLl\cos(q_1-q_2)\dot{q}_2 \\ p_2 &= \frac{\partial E_k}{\partial \dot{q}_2} = 2mLl\cos(q_1-q_2)\dot{q}_1 + ml^2\dot{q}_2 \end{aligned} \tag{3.3.5}$$

Para este ejemplo, la matriz T sería simplemente:

$$T = \begin{pmatrix} (M+m)L^2 & mLl\cos(q_1-q_2) \\ mLl\cos(q_1-q_2) & ml^2 \end{pmatrix} \tag{3.3.6}$$

De tal forma que $p_1 = T_{11}p_1 + T_{12}p2$, y $p_2 = T_{21}p_1 + T_{22}p2$. Resumiendo usando la notación de suma de Einstein para subíndices, $p_i = T_{ij}\dot{q}_j$.

Nótese la simetría dentro del par de ecuaciones en 3.3.5, que se refleja en la simetría de la matriz 3.3.6.

Cuando se requiere obtener las velocidades a partir de los momentos, se invierte la matriz T, resultando en T^{-1}. El cálculo de la i-ésima velocidad generalizada sería entonces $\dot{q}_i = T_{ij}p_j$.

Es importante mencionar que los valores numéricos de la matriz T se guardan, internamente en el programa, en un arreglo bidimensional con nombre *M*. En el ejemplo anterior no echamos mano de su nombre real para evitar una posible confusión con la masa del primer péndulo. Igualmente, los valores de la matriz inversa de T se guardan, internamente en el programa, en un arreglo bidimensional con nombre *Minv*.

3.3.2. Un resumen enumerado

A continuación se presenta una lista que compendia, paso a paso, un resumen de los cálculos que ocurren en cada paso de integración para un sistema con n coordenadas generalizadas.:

1. Se calculan los valores de la matriz T (internamente al programa es la matriz M) para el tiempo en cuestión.
2. Se calculan los valores del vector p de momentos generalizados a partir de las velocidades generalizadas ($p_i = T_{ij}\dot{q}_j$).
   Recordamos que, internamente al programa, las coordenadas generalizadas se guardan en un vector q, las velocidades generalizadas $\dot{q}$ se guardan en un vector qv, y los momentos en un vector p.

3. Se guardan los valores de las coordenadas y momentos generalizados en el vector X. X_i guarda el valor de q_i y X_{i+n} guarda el valor de p_i.
4. Se calcula el valor de k_1 del método de Runge-Kutta para coordenadas generalizadas y momentos.

   - k_1 de la i-ésima coordenada es $\dot{q}_i$. Internamente, $K_{1,i} = q\,v_i$.
   - k_1 del i-ésimo momento es $K_{1,i+n} = \frac{\partial L}{\partial q_i}$, con L=T-V.

   Y se usan dichas K para evolucionar en el tiempo a las q y las p un intervalo $\delta/2$ respecto a los valores guardados al inicio del paso de integración:

$$\begin{aligned} q_i &= X_i + K_{1,i}\frac{\delta}{2} \\ p_i &= X_{i+n} + K_{1,i+n}\frac{\delta}{2} \end{aligned} \tag{3.3.7}$$

   Donde las q_i y las p_i quedan con un valor correspondiente a la evolución temporal.
5. Se calculan los valores de las velocidades generalizadas $\dot{q}$ (internamente, el vector qv): $\dot{q}_i = T^{-1}_{ij}\,p_j$.
6. Se calcula el k_2 del método de Runge-Kutta, de la misma forma que la mencionada en el paso 4, pero usando las q, p y $\dot{q}$ ya evolucionadas mediante un paso de $\delta/2$. Es decir, se obtiene, para cada coordenada y momento, un $k_2 = f(t + \frac{\delta}{2}, \boldsymbol{y} + \frac{\delta}{2}k_1)$, donde $\boldsymbol{y}$ representa un "*vector*" de coordenadas y momentos generalizados de los que depende el sistema.
   Nuevamente se evolucionan las q y las p como en la ecuación 3.3.7 respecto al paso de integración, otra vez usando un paso de tiempo de $\delta/2$, pero ahora usando los valores de q, p y $\dot{q}$ obtenidos evolucionando en el tiempo con k_2. Y se obtienen las velocidades a partir de los momentos como antes.
7. Se calcula el k_3, usando las q, p y $\dot{q}$ evolucionadas a partir de k_2.
   Se usa este nuevo k_3 para evolucionar las q y p respecto a los valores guardados en el vector X, ahora usando un paso δ.
   Y se obtienen las $\dot{q}$ a partir de los p.
8. Se calcular k_4 como se hizo antes, pero considerando los nuevos valores de q, p y $\dot{q}$.
9. Se usan los valores de las k, según la fórmula de Runge-Kutta, para estimar el valor de las q y las p para el siguiente paso, como en la ecuación 3.3.3.

4

Usos posibles de *Cuerpos en Movimiento*

En este apartado se analizan los usos potenciales del simulador *Cuerpos en Movimiento*, particularmente para cuestiones didácticas.

A continuación se enlistan algunos problemas didácticos en que se considera se puede aprovechar a *Cuerpos en Movimiento*:

- **Poca familiarización con el concepto de vector**.
  La mecánica requiere del concepto de vector para el manejo de diversas cantidades (la velocidad en particular). Se podrían incluir, por ejemplo, velocidades representadas con flechas de tamaño correspondiente a la velocidad y en la dirección del movimiento para que el alumno pudiera notar en qué puntos ésta es máxima o mínima, puntos de retorno, etc.

- **Poca familiarización con medidas angulares**.
  Principalmente en los sistemas que involucran objetos que rotan (por ejemplo, péndulos), tanto los ángulos, pero más las velocidades y aceleraciones angulares, son conceptos que a veces no son tan naturales para alumnos que recién toman materias de mecánica.

- **La conservación de la energía**.
  Esta propiedad se desprende de las leyes de Newton. Son, de hecho, parte fundamental dentro del formalismo de Lagrange y Hamilton que permiten simular un sistema mecánico mediante un algoritmo. En el ámbito mecánico se restringe a la suma de energía potencial y cinética, mismas que se podrían mostrar gráficamente conforme el sistema evoluciona.

- **El principio de la mínima acción**.
  Dentro del análisis variacional, uno de los comportamientos de la naturaleza es la tendencia a llevar cantidades a un extremo (normalmente un mínimo). En el caso de la acción, la tendencia es a minimizarla. De hecho, es posible obtener las leyes de Newton a partir de este principio.

- **La conservación del momento**.
  Una de las propiedades más importantes, aún independiente de si el ámbito de un sistema es o no relativista, es la conservación del momento. Resulta, así pues, un concepto muy importante con el cual conviene familiarizarse. Y es también indispensable en la deducción de las leyes de Newton.

El momento lineal total de un sistema en ausencia de fuerzas se conserva independientemente de lo complejo del sistema en cuestión. Por ejemplo, el modelo precargado Péndulo deslizante puede permitir un análisis e impresión del momento conforme el sistema evoluciona con el tiempo.
Adicional al momento lineal, también se puede aprovechar el interactivo para mostrar el momento angular, especialmente en modelos de mecánica celeste como el de Movimiento Planetario que se encuentra actualmente precargado en el interactivo. Además, también permitiría una familiarización con operaciones entre vectores como el producto cruz (dado que se puede calcular como $r \times p$).

- **Desarrollo de intuición mecánica**. Es decir, desarrollo de una intuición que permita responder a preguntas tales como:

  - ¿Qué comportamiento se espera para un péndulo doble si la masa inicial es mucho más grande que la segunda masa?
  - ¿Qué se espera si se llegara en un péndulo doble a una situación en que la masa más masiva se encuentra cercana a su punto inferior y con poca velocidad?
  - ¿Cómo se comportaría el modelo precargado Péndulo deslizante si se suelta el péndulo desde el reposo y la masa pendular es mucho mayor a la del bloque deslizante?
  - ¿En qué condiciones, en particular en el Péndulo doble y Péndulo triple, el cambio de la energía máxima aumenta drásticamente y por qué?
  - ¿Por qué todo el sistema en el ejemplo del Péndulo deslizante mantiene un desplazamiento relativamente homogéneo en una dirección (derecha o izquierda)? Expresado de otra forma, ¿qué condiciones deben cumplir la velocidad del bloque y la velocidad angular de la barra para que, por ejemplo, no se mueva el sistema en conjunto horizontalmente?
  - En el modelo planetario 1 en que el Sol se considera fijo, y dadas las otras condiciones iniciales, ¿qué velocidad tangencial se le debería dar al planeta para que su velocidad sea siempre constante?
  - Despreciando los errores de integración, ¿qué condición deben cumplir las condiciones iniciales para que cada uno de los objetos tracen curvas cerradas en el modelo planetario 2? Recordando que en este modelo, tanto el Sol como el planeta pueden moverse libremente.
  - Diseñe condiciones iniciales para el sistema de tres cuerpos en 2D de tal forma que el Sol robe el satélite del planeta.
  - El alumno puede practicar con las nociones sobre el tipo de energía en el modelo planetario que debe manejar para obtener órbitas cerradas o no.

  Este tipo de preguntas no necesariamente requieren cálculos para ser respondidas, pero sí requieren que el alumno internalice propiedades como conservación de energía en su razonamiento para su respuesta.

- **Análisis de sistemas con caos determinista**.
  A pesar de que los sistemas mecánicos aquí considerados son deterministas (se puede establecer su evolución hacia un tiempo dado si se proporcionan las condiciones iniciales), la sensibilidad a condiciones iniciales involucra el estudio del caos del sistema. Un mismo péndulo doble al que se le cambia ligeramente alguna de sus condiciones iniciales evolucionará muy distintamente al original debido a que es caótico. Podría eventualmente incluirse el exponente de Lyapunov con el fin de mostrar cómo y por qué es una medida del caos de un sistema.

- **Análisis del espacio fase**.
  La graficación del momento contra la posición de cada elemento del modelo permitiría conocer las condiciones únicas de un sistema dado en un tiempo determinado. Adicionalmente, permite una noción más clara de la razón por la cual, para un sistema dado, la curva en el espacio fase no puede cruzarse a sí misma, y se puede relacionar esto con la unicidad de la solución de las ecuaciones diferenciales asociadas.

5

Sistema de archivos de *Cuerpos en Movimiento*

En este apartado se da una descripción del sistema de archivos contenidos en el archivo *CuerposEnMovimiento.zip*, de donde el usuario puede descargar toda la herramienta.

En este capítulo y en el que aborda la *estructura de un archivo de modelo* se hace referencia a archivos de texto en general. Estos archivos **deben estar** en una codificación UTF-8, de preferencia *sin BOM*, para el correcto funcionamiento de *Cuerpos en Movimiento*. Para ello, se recomienda el uso de un editor de texto tal como Notepad++, cuya página es https://notepad-plus-plus.org/. La descarga de dicho editor es gratuita. Basta seleccionar la opcion *UTF-8 sin BOM* del menú *Codificación* de dicho editor para que cualquier guardado subsecuente se encuentre en la codificación adecuada.

También se recomienda habilitar la opción que hace que se muestren las extensiones de archivos. Algunos sistemas operativos por defecto ocultan las extensiones, lo que puede resultar problemático para el usuario que usa archivos generados directo por el mismo y no mediante aplicaciones de terceros.

5.1. Las carpetas

5.1.1. La carpeta *biblioteca*

Esta carpeta incluye las bibliotecas necesarias para que el programa funcione adecuadamente. Los archivos en la biblioteca son los siguientes:

- *Constructors*. Contiene las definiciones necesarias para generar nuevos objetos de los modelos mecánicos.
- *InvertMatrixM*. Contiene las definiciones necesarias para generar la inversa de una matriz.
- *NumMthds*. Contiene las definiciones necesarias para realizar la integral numérica del sistema dinámico mediante un método muy similar a *Runge-Kutta orden 4*.
- *ParamsCIs*. Contiene las definiciones necesarias para el guardado y acceso de los parámetros de los objetos mecánicos y las condiciones iniciales del sistema.
- *TranslVars*. Contiene las definiciones necesarias para traducir entre variables externas (coordenadas y velocidades nombradas por el usuario) e internas (el nombre

dado a cada una internamente en el programa, y que típicamente corresponde a coordenadas y velocidades generalizadas).

- *HandleModels*. Contiene las definiciones necesarias para el correcto cargado y manipulación de los distintos modelos dinámicos.

Estas bibliotecas son archivos de texto creados a partir de definiciones en el programa *DescartesJS*, por lo que es necesario un entendimiento del mismo para poder editarlos en caso que así se desee.

5.1.2. La carpeta *icons*

Esta carpeta contiene los íconos de los botones de la herramienta. Son imágenes tipo *svg*.

5.1.3. La carpeta *lib*

Esta carpeta contiene un archivo *descartes-min.js*. Este archivo, conocido como *intérprete de DescartesJS* es un archivo *JavaScript* necesario para que escenas generadas en *DescartesJS* sean correctamente interpretadas dentro de navegadores.

El intérprete de *DescartesJS* es un archivo que constantemente se modifica, según cambios en la funcionalidad de *DescartesJS*. La última versión del mismo se puede obtener directamente usando dicha herramienta. Para más información al respecto se puede consultar la documentación de *DescartesJS*, disponible en https://descartes.matem.unam.mx/doc/DescartesJS.

5.1.4. La carpeta *macros*

Esta carpeta contiene macros, o pequeñas instrucciones gráficas para facilitar el trazado de algunos objetos. Estos archivos son de tipo texto, y son generados también a partir de instrucciones de *DescartesJS*.

En particular, el trazado de resortes, que sufren deformaciones al evolucionar un modelo dinámico, está realizado mediante un macro, mismo que se encuentra dentro de esta carpeta.

Nuevamente, los macros de *DescartesJS* se pueden consultar en la documentación técnica de dicha herramienta.

5.1.5. La carpeta *models*

Esta carpeta contiene archivos de texto, de extensión *.dat*, cada uno de los cuales corresponde a un modelo de los mostrados en *Cuerpos en Movimiento*.

Estos archivos pueden cargarse o no según si aparecen en el archivo *models.xml*. Es decir, el usuario está en libertad de elegir cuáles de ellos se muestran o no.

5.1.6. El archivo *models.xml*

El último elemento a describir es este archivo, que se encuentra en la raíz del archivo *zip* de *Cuerpos en Movimiento*. Este archivo, un archivo de texto, contiene la información de los modelos que se mostrarán en la herramienta, de entre aquellos disponibles en la *carpeta models*. La Figura 5.1 contiene un ejemplo de este tipo de archivo.

```
<MODELS>
<MODEL>
<FILE>movimientoPlanetario1.dat</FILE>
<TITLE>Movimiento planetario 1</TITLE>
</MODEL>
<MODEL>
<FILE>movimientoPlanetario2.dat</FILE>
<TITLE>Movimiento planetario 2</TITLE>
</MODEL>
</MODELS>
```

Figura 5.1: Ejemplo de un archivo *models.xml* que sólo incluye 2 modelos.

El formato de este archivo es de tipo *xml*. Contiene una etiqueta principal `<MODELS>`, con su respectivo cierre `</MODELS>`, entre las cuales está todo el contenido del archivo. Cada modelo a incluir se flanquea entre una etiqueta de apertura `<MODEL>` y una de cierre `</MODEL>`. Como se aprecia en la Figura 5.1, en dicho ejemplo hay sólo dos modelos.

La información particular de cada modelo incluye una etiqueta `<FILE>...</FILE>`, dentro de la cual viene el nombre del archivo con extensión *.dat* del modelo (sólo el nombre de archivo del modelo). Al tratarse de un nombre de archivo, no es recomendable incluir caracteres fuera de lo común como tildes, etc. Posterior a dicha etiqueta, viene otra `<TITLE>...</TITLE>`, dentro de la cual viene el nombre del modelo a mostrar al usuario en general. Al tratarse de un título, es permitido usar caracteres tales como acentos, etc.

La figura de ejemplo contiene sólo dos modelos. Sin embargo, el usuario puede incluir tantos como modelos tenga. Si desea que uno de sus modelos, a pesar de estar presente en la carpeta *models*, no esté presente en la herramienta; basta excluirlo de la lista en el archivo *models.xml*.

6

Estructura de un archivo de modelo

Los archivos de modelo, aquellos que se encuentran en la *carpeta models*, son archivos de texto. Deben llevar una extensión *.dat*. Como se recomendó, es útil habilitar que se muestren las extensiones de archivos. Y, nuevamente, también se recomienda realizar la edición de estos archivos en un editor como Notepad++, usando la codificación *UTF-8 sin BOM* previo a guardar.

Este archivo, de los cuales el usuario puede consultar los ya existentes, viene en un formato basado en etiquetas tipo *xml*. La Figura 6.1 contiene un ejemplo de un archivo de modelo.

6.1. El bloque de la etiqueta *GENERAL*

El primer bloque de datos viene dentro de una etiqueta `<GENERAL>...</GENERAL>`. Es decir, hay una etiqueta de apertura `<GENERAL>`, luego una serie de datos, y al final la etiqueta de cierre `</GENERAL>`. Los datos dentro de esta etiqueta vienen con un signo igual, tras el cual trae algún valor numérico para la variable en cuestión. Los valores deben ser números en formato decimal. Es decir, no se permiten notaciones exponenciales. Los elementos de este bloque son:

- **escala**. El número asignado a esta variable es la escala del espacio en el que se muestra el modelo dinámico. La escala es el número de pixeles (px) que componen una unidad de medición. Por ejemplo, 54 indica que una unidad de medida está compuesta de 54 px.
- **Ox**. El número asociado corresponde al desplazamiento (en px) horizontal del espacio en el que se muestra el modelo. El origen del espacio se encuentra típicamente centrado (el caso en que **Ox** es cero). Para valores positivos, el espacio se moverá el número correspondiente de px a la derecha. Para valores negativos, a la izquierda.
- **Oy**. Similar a *Ox*, sólo que corresponde al desplazamiento *vertical* del espacio. Valores positivos corresponde a desplazamientos hacia abajo; valores negativos corresponden a desplazamientos hacia arriba.
- **dt**. Contiene el valor del paso de tiempo que transcurre entre un trazo gráfico del modelo y el siguiente. Es decir, el tiempo que pasa entre que se pinta el modelo una vez hasta que se pinta la siguiente vez. Corresponde a el parámetro *dt* descrito en el apartado sobre la *integración numérica*.

```
<GENERAL>
escala=54
Ox=0
Oy=-150
dt=0.03125
N=32
nQ=2
</GENERAL>

<PARAMETERS>
<P>g|9.81|aceleración de la gravedad</P>
<P>k|120|constante del resorte en la ley de Hooke</P>
<P>l|4|longitud de la barra</P>
<P>L|5|longitud del resorte sin estirar</P>
<P>m|4|masa de la bola atada a la barra</P>
<P>M|6|masa del bloque atado al resorte</P>
<P>mgl|m·g·l|abreviatura</P>
</PARAMETERS>

<CIs>
<CI>a|pi/8|ángulo de la barra</CI>
<CI>va|2|velocidad angular de la barra</CI>
<CI>x|4|abscisa del bloque</CI>
<CI>vx|0|velocidad horizontal del bloque</CI>
</CIs>

<TMATRIX>
<T>0,0|'m·l²'</T>
<T>0,1|'0'</T>
<T>1,0|'0'</T>
<T>1,1|'M'</T>
</TMATRIX>

<ENERGIES>
<Ek>(m·l²·va²+M·vx²)/2</Ek>
<Ep>-mgl·cos(a)+k·(L-sqrt((x-l·sen(a))²+(l-l·cos(a))²))²/2</Ep>
</ENERGIES>

<GRAPHOBJ>
<G>Bar|b0|0.1|'l'|'0'|'2'|'a'|'0'|'0'|'va'|1|-1</G>
<G>C|C1|0.4|Obj[b0,_xf]|Obj[b0,_yf]|'0'|'0'|0|b0|0</G>
<G>Blk|B1|1.6|1|'x'|'-2'|'0'|'vx'|'0'|'0'|1|-1</G>
<G>Spr|S0|5|0.2|Obj[C1,_x]|Obj[C1,_y]|Obj[B1,_x]|Obj[B1,_y]|1|B1</G>
<G>Bar|b1|0.1|'20'|'-10'|'-2.6'|'pi/2'|'0'|'0'|'0'|0|-1</G>
</GRAPHOBJ>
```

Figura 6.1: Ejemplo de un archivo de modelo.

- **N**. Contiene el valor del número de particiones regulares en que se divide el intervalo de tiempo *dt*. Como se menciona en el apartado sobre la *integración numérica*, entre cada trazo gráfico del sistema y el siguiente, se hace una subdivisión en N pedazos, sobre los que se hace realmente la integración numérica.
- **nQ**. Contiene el número de coordenadas del sistema dinámico. En el ejemplo de la Figura 6.1, que corresponde a un modelo con un bloque constreñido a un plano horizontal y un péndulo, las coordenadas generalizadas son la posición horizontal del bloque y el ángulo del péndulo. Es por ello que, para dicho ejemplo, $nQ = 2$.

6.2. El bloque de la etiqueta *PARAMETERS*

Este bloque está compuesto por una serie de renglones. Cada renglón tiene información flanqueada por las etiquetas `<P>...</P>`; y corresponde a un parámetro del sistema.

La información de cada renglón consiste de tres datos, separados por el caracter *pipe* | (la barra vertical). El primer dato es el identificador o nombre que se le da al parámetro. Como será una variable de sistema, no admite caracteres especiales como tildes. El segundo dato es el valor numérico del mismo, que puede ser un número como tal, o una expresión resultante de una operación. El tercer dato es una descripción del parámetro, que es la misma que aparece en el modelo de la herramienta como tal.

Por ejemplo, el primer parámetro en la Figura 6.1 está identificado mediante la variable *g*, tiene un valor de *9.81*, y su descripción es el texto *aceleración de la gravedad*. El último parámetro está identificado por una variable *mgl*, su valor es *m·g·l* (interpretado como el producto de *m*, *g*, y *l*, y su descripción es *abreviatura*, ya que es una variable de abreviatura que permite ahorrar estar calculando un mismo producto que se usará en repetidas ocasiones.

Los parámetros definidos en este bloque corresponden a aquellos mostrados en el panel al que se accede pulsando el botón de *introducción de parámetros*. Sin embargo, es en el archivo *.dat* de modelo donde se definen cuántos parámetros habrán, con qué identificadores y con qué descripciones. En la herramienta *Cuerpos en Movimiento* sólo se puede editar el valor del parámetro.

6.3. El bloque de la etiqueta *CIs*

Este bloque contiene las condiciones del sistema. Como las introducidas aquí son las que definen al sistema desde el inicio, realmente corresponden a *condiciones iniciales*. El bloque de información está contenido dentro de la etiqueta `<CIs>...</CIs>`.

Cada renglón corresponde a una condición distinta, y la información está flanqueada por las etiquetas `<CI>...</CI>`. La información es la misma que para los parámetros (identificador, valor, y descripción). Cada dato está separado también por el caracter |.

Por ejemplo, el primer renglón de este bloque en la Figura 6.1 indica que el identificador es una variable de nombre *a*, con valor inicial $\frac{\pi}{8}$, y la descripción a mostrar al usuario es *ángulo de la barra*.

Aquí cabe hacer un par de aclaraciones. Por un lado, puede notarse que el sistema acepta *pi* y lo interpreta como el valor de la constante π. Por lo mismo, es preciso no tratar de definir una variable justo con ese nombre. Por otra parte, es posible incluir, como valor de una condición, una operación, como también se podía con los parámetros.

Este bloque lleva la información que se muestra en el panel lanzado al pulsar el botón *introducción de condiciones*. No obstante, es en el archivo *.dat* de modelo donde se defi-

nen cuántas condiciones habrán, con qué identificadores y con qué descripciones. En la herramienta *Cuerpos en Movimiento* sólo se puede editar el valor de cada condición.

6.4. El bloque de la *matriz T*

La *matriz T* es aquella que se menciona en el apartado sobre *la integración numérica*, y sobre la cual se ofrece un ejemplo en el apartado *ejemplo concreto de la matriz T*.

La información de la matriz en el archivo *.dat* del modelo se encuentra dentro de las etiquetas `<TMATRIX>...</TMATRIX>`. Cada entrada de la matriz se encuentra dentro de una etiqueta `<T>...</T>`. Y hay dos datos por cada entrada. El primero es los índices de la matriz (separados por una coma). Posteriormente viene el caracter de separación |. Finalmente viene el valor o expresión de la entrada de la matriz. Los índices de la matriz se empiezan a contar desde cero.

En el ejemplo de la Figura 6.1, la primera entrada de la matriz está dada por el primer renglón dentro del bloque *TMATRIX*. El contenido de dicho renglón es `0,0|'m·l²'`. El *0,0* corresponde a los índices de la entrada. La expresión después del | corresponde a lo que va en dicha entrada de la matriz.

Los valores dados a la matriz, ya sean numéricos o algebraicos, son todos realmente expresiones a manera de texto. Por lo mismo, **es imperativo flanquearlos con comillas sencillas**. De lo contrario, la herramienta no sabrá como interpretar las entradas.

6.5. El bloque de las energías

Este bloque contiene las expresiones para la energía cinética y potencial del sistema. Se encuentra dentro de las etiquetas `<ENERGIES>...</ENERGIES>`.

Dentro de dicho bloque hay una etiqueta `<Ek>...</Ek>`, que en su interior contiene la expresión de la energía cinética. La otra etiqueta, `<Ep>...</Ep>`, contiene la expresión de la energía potencial.

Estas expresiones corresponden a las mostradas en el panel de energías, lanzado al oprimir el botón *mostrar / ocultar energías*.

6.6. El bloque de *objetos gráficos*

Los objetos gráficos que se muestran y animan en *Cuerpos en Movimiento* se definen en este bloque en el archivo *.dat*. Cada elemento gráfico es un renglón dentro de este bloque, y se encuentra flanqueado por las etiquetas `<G>...</G>`.

El número de datos por renglón depende del objeto gráfico en cuestión. No obstante, se sigue usando el mismo separador | entre datos del objeto. Por ejemplo, en la Figura 6.1 de ejemplo, los datos del primer renglón son:

```
Bar|b0|0.1|'1'|'0'|'2'|'a'|'0'|'0'|'va'|1|-1
```

El primer dato indica que el objeto es una barra. El segundo nos dice que el identificador asociado a la barra es *b0*. El resto de los datos dependen del objeto.

En ocasiones se hablará de si el objeto es *arrastrable* o no. Por arrastrable se entiende si el objeto podrá ser pulsado y arrastrado con el mouse. De ser arrastrable, el dato asociado debe valer 1; de lo contrario vale cero.

También se habla del *objeto padre*. Por objeto padre se entiende el objeto del cual depende el objeto abordado. Por ejemplo una barra pendular puede pender de un bloque, en cuyo caso el bloque es el objeto padre de la barra. Si un objeto depende de otro, en el dato de padre debe ir el identificador del padre. De lo contrario, va un valor -1.

Es posible definir que algunos objetos dejen rastros al moverse (algo así como un trazo de por dónde han pasado). Por ejemplo, los extremos de barras o los círculos o bolas. Hay un dato de dichos objetos asociado a si dejan rastro (en cuyo caso el dato vale 1) o no (en cuyo caso el dato vale 0).

A continuación se desglosa cada objeto y los datos que se especifican en su renglón de objetos gráficos:

- **Barra**. Un objeto gráfico es barra si su primer dato de información es *Bar*. El segundo dato corresponde al identificador asociado a la barra. El tercer dato corresponde a la anchura de la misma; el cuarto a la longitud; el quinto y el sexto a las coordenadas de abscisas y ordenadas de donde se encuentra sujeta la barra; el séptimo al ángulo inicial (en radianes) que forma la barra con la vertical; el octavo y noveno a las velocidades horizontal y vertical del punto de sujeción de la barra; el décimo a la velocidad angular (en *rad/s*) de la barra; el undécimo indica si el objeto es arrastrable o no; y el duodécimo es el identificador del objeto padre de la barra.
  Las barras se usan principalmente para representar las barras de péndulos.
  Los datos que son expresiones a manera de texto son del cuarto al décimo.
- **Bloque**. Un objeto gráfico es bloque si su primer dato de información es *Blk*. El segundo dato corresponde a su identificador. El tercer dato es la anchura; el cuarto dato es la altura; el quinto y sexto datos son la coordenada horizontal y vertical del centro del bloque; el séptimo es el ángulo que forma con la vertical, el octavo y noveno son las velocidades horizontal y vertical, el décimo es la velocidad angular del bloque, el undécimo indica si el bloque es arrastrable o no; y el duodécimo es el identificador del objeto padre del bloque.
  Los datos que son expresiones a manera de texto son del quinto al décimo.
- **Círculo o bola**. Un objeto gráfico es bola si su primer dato de información es *C*. El segundo dato corresponde a su identificador. El tercero es el radio de la bola; el cuarto y quinto son sus coordenadas horizontal y vertical; el sexto y séptimo son sus velocidades horizontal y vertical; el octavo indica si la bola es arrastrable; el noveno lleva el identificador del padre del objeto; y el décimo indica si lleva rastro o no.

Las bolas suelen usarse para representar extremos masivos de péndulo, o bien objetos puntuales libres como los usados en mecánica celeste.
Los datos que son expresiones a manera de texto son del cuarto al séptimo.

- **Resorte**. Un objeto es un resorte si su primer dato de información es *Spr* (del inglés *spring*). Su segundo dato es el identificador del resorte. El tercer y cuarto datos son la longitud y grosor del resorte; el quinto y el sexto son las coordenadas horizontal y vertical del primer punto de sujeción; el séptimo y octavo son las coordenadas horizontal y vertical del segundo punto de sujeción; y el noveno es el padre del objeto.
  El grosor del resorte rara vez se atiende, independientemente del valor que el usuario entre, debido a que los resortes quedan definidos por un macro. Igualmente, el usuario puede ignorar el último parámetro, debido a que los extremos del resorte quedan definidos por los puntos de sujeción.
  Los datos que son expresiones a manera de texto son del quinto al octavo.

Es importante tener en mente que los ángulos se miden respecto a la vertical y hacia abajo. Tal como el ángulo que hace un péndulo con la vertical. Y se miden en radianes.

Otro punto importante a tener en cuenta es que algunos de los datos de los objetos son expresiones a manera de texto. Así pues, si es un número que debe entrarse, dicho número debe ir flanqueado por comillas sencillas. No obstante, en algunos casos en lugar del número va una variable que contiene la expresión del dato. En el ejemplo de la Figura 6.1 puede verse que el cuarto y quinto datos del objeto círculo son `Obj[b0,_xf]` y `Obj[b0,_yf]`. Así, se está especificando a la herramienta que el círculo (extremo del péndulo), lleva como posiciones horizontal y vertical aquellas del extremo final de la barra con identificador *b0*. O bien, los datos quinto al octavo del objeto resorte son `Obj[C1,_x]`, `Obj[C1,_y]`, `Obj[B1,_x]` y `Obj[B1,_x]`. Con ello se está indicando que el resorte está atado, en su primer extremo, al círculo con identificador *C1* y, en el segundo extremo, al bloque con identificador *B1*. Dichos datos no van entre comillas sencillas, ya que corresponden a variables que representan un atributo del objeto que ya internamente lleva las comillas sencillas. En el apartado sobre *atributos de objetos* se describen cuáles son éstos.

6.7. Atributos de objetos

Internamente, cada objeto está representado por un *Obj[id,atributo]*. Por ejemplo, *Obj[B1,_x]* es la coordenada horizontal (es un atributo) de un objeto con identificador *B1*. Se incluye en esta documentación una lista de los atributos de cada objeto, con el fin de que se entienda cuando se usan en el bloque de objetos gráficos:

- *Para todo tipo de objetos*:
  - **_type**. En él se indica el tipo de objeto del que se trata. Admite los valores 'circulo', 'resorte', 'barra' o 'bloque'.

- **_draggable**. Se usa para indicar si un objeto es arrastrable o no. Lleva un 1 en caso de serlo, y un 0 de lo contrario.
- **_parent**. Se usa para indicar el identificador del objeto que fungirá como padre del objeto en cuestión.

- *Para círculos o bolas*:
  - **_r**. El radio de la bola.
  - **_rastro**. 1 si la bola ha de dejar rastro al moverse; 0 de lo contrario.
- *Para bloques, barras y resortes*:
  - **_w**. Anchura del objeto.
  - **_h**. Altura del objeto.
  - **_xf**. Posición horizontal del extremo final del objeto.
  - **_yf**. Posición vertical del extremo final del objeto.
  - **_a**. Ángulo del objeto respecto a la vertical.
  - **_va**. Velocidad angular del objeto.
  - **_vxf**. Velocidad horizontal del extremo final del objeto.
  - **_vyf**. Velocidad vertical del extremo final del objeto.
- *Para círculos, bloques y barras*:
  - **_x**. Posición horizontal del objeto, o de su punto de sujeción.
  - **_y**. Posición vertical del objeto, o de su punto de sujeción.
  - **_vx**. Velocidad horizontal del objeto, o de su punto de sujeción.
  - **_vy**. Velocidad vertical del objeto, o de su punto de sujeción.
- *Para círculos y bloques*:
  - **_vtot**. Velocidad neta del objeto (es decir, no respecto a otro objeto).
- **Para resortes**:
  - **_l**. La longitud del resorte.
  - **_k**. La constante de la ley de Hooke del resorte.

Se recomienda al usuario que desea construir sus propios modelos, duplicar y editar el duplicado de alguno de los modelos provistos, mientras sigue esta guía. Ello evitará errores principalmente de formato.