

Simulación en tres dimensiones del problema de N cuerpos

M^a Carmen Peña and Jesús F. Palacián

Departamento de Matemática e Informática

Universidad Pública de Navarra. E – 3 006 Pamplona

Resumen

We present a set of routines developed with MATLAB to simulate the motion of N bodies subject to Newtonian interaction in the three-dimensional space. The software allows to choose among several kinds of numerical integrators, say various Runge-Kutta methods and linear multistep algorithms, some of them of symplectic character. Initial conditions are introduced either in Cartesian coordinates or in orbital elements. The program detects collision events. Simulation is accomplished in real-time using the common visualization packages of MATLAB. We can change the point of view and project the motion in the three coordinate planes. The integrals of the motion are checked at each step of the integration. Several examples of the solar system, choreographies of the N body problem and other peculiar motions of planar and spatial problems are shown. This work is part of the Master Thesis Project of M. C. Peña.

1. Introducción

Desde la época de los griegos se intenta explicar el movimiento de los cuerpos sobre la bóveda celeste. En el Sistema Solar, debido a que casi toda la masa está concentrada en el Sol, el estudio de la trayectoria de un planeta se aproxima mediante los resultados del problema de dos cuerpos (dicho planeta y el Sol), ya que las ecuaciones que generan son completamente integrables. Sin embargo, para periodos extendidos de tiempo, los restantes planetas influyen sobre el movimiento del cuerpo considerado, y por lo tanto, su trayectoria será ligeramente diferente en tamaño y orientación espacial, a medida que transcurre el tiempo. El problema es que cuando se consideran configuraciones de más de tres cuerpos las ecuaciones diferenciales que se generan no tienen una solución analítica, por lo que debe usarse integradores matemáticos para generar aproximaciones más o menos fiables, dependiendo del error cometido. El objetivo del proyecto es la elaboración de un programa en el que dados N cuerpos sujetos a atracción gravitatoria (siendo

$N > 1$), de masas y radios arbitrarios, con unas condiciones iniciales de posición y velocidad cualesquiera en el espacio de tres dimensiones, el programa aproxime numéricamente, con el menor error posible, las ecuaciones del movimiento de los N cuerpos en un cierto intervalo prefijado, y además visualice sus trayectorias en tiempo real en un espacio tridimensional. Para ello se han implementado un conjunto de rutinas con la versión 6.5 de Matlab (Matrix Laboratory) bajo un sistema operativo de Windows.

2. Ecuaciones del Problema de N Cuerpos

El primer paso a realizar es buscar la ecuación que modela el movimiento de N cuerpos sometidos a interacción gravitatoria mutua. A partir de la ley de gravitación universal de Newton [2], que nos permite conocer la fuerza ejercida entre dos cuerpos, obtenemos la ecuación de movimiento para el cuerpo i -ésimo en un sistema de referencia inercial:

$$m_i \ddot{\mathbf{q}}_i = \sum_{j=1}^N \frac{G m_i m_j (\mathbf{q}_j - \mathbf{q}_i)}{\|\mathbf{q}_i - \mathbf{q}_j\|^3}, \quad (i \neq j), \quad \ddot{\mathbf{q}}_i = \frac{d^2 \mathbf{q}_i}{dt^2},$$

siendo G la constante de gravitación universal, de valor $G = 6,672 \cdot 10^{-11} \frac{Nm^2}{Kg^2}$ y donde la partícula i -ésima tiene por vector de posición \mathbf{q}_i y masa $m_i > 0$.

El problema de N cuerpos se define mediante un sistema de ecuaciones diferenciales ordinarias. Como las posiciones y velocidades iniciales de los cuerpos son datos, el problema corresponde a la categoría de los problemas de valor inicial para ecuaciones diferenciales ordinarias. Además, si nos ceñimos a que los cuerpos se mueven exclusivamente por el efecto de la gravitación del resto de cuerpos, el problema conserva la energía total del mismo, y puede plantearse como un problema de tipo hamiltoniano [12].

El problema de valor inicial para ecuaciones diferencial ordinarias es de dimensión $3N$, ya que para cada cuerpo es necesario encontrar las tres componentes de la aceleración. Pero cuando dicha ecuación diferencial se reduce a primer orden es de dimensión $6N$, ya que no sólo las aceleraciones se desconocen, sino también las velocidades.

Cuando son dos los cuerpos que se atraen mutuamente, el sistema es completamente integrable y puede resolverse de forma exacta, ya que el movimiento de un cuerpo, en torno a otro, está perfectamente definido mediante seis elementos orbitales constantes [1]:

- Semieje mayor (a)
- excentricidad de la órbita (e)
- inclinación respecto del plano de la eclíptica (i)
- ángulo del nodo (Ω)

- argumento del periastro (ω)
- instante de paso por el periastro (T)

Cuando el número de cuerpos presentes en el sistema es mayor que dos sólo se conocen diez integrales de movimiento independientes: el momento lineal (\mathbf{L}), el centro de masas (\mathbf{C}), el momento angular (\mathbf{O}) y el hamiltoniano (H), o energía total del sistema de partículas, que es la suma de la energía cinética y la energía potencial ($V = -U$):

$$H = \frac{1}{2} \mathbf{p}^T M^{-1} \mathbf{p} - U = \sum_{i=1}^N \frac{\|\mathbf{p}_i\|^2}{2m_i} - U,$$

donde $M = \text{diag}(m_1, m_1, m_1, \dots, m_N, m_N, m_N)$ es el vector de masas y $\mathbf{p} = (\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_N) \in \mathfrak{R}^{3N}$

Como el problema es de dimensión $6N$ y sólo se conocen diez integrales primeras de movimiento no es posible resolverlo analíticamente. Actualmente existen dos filosofías para resolver las ecuaciones que genera el problema de N cuerpos: métodos analíticos, que descansan en la teoría de las perturbaciones, y la integración numérica [10], que es una solución más inmediata, por lo que va a ser la solución empleada en este proyecto.

3. Integradores Numéricos: Resolución de las Ecuaciones

Los métodos numéricos no producen una aproximación continua a la solución del problema inicial, sino que generan unas aproximaciones a la solución, en algunos puntos específicos y, a menudo, igualmente espaciados.

Para resolver las ecuaciones diferenciales hemos usado dos tipos de integradores numéricos:

- Por un lado los que proporciona Matlab, que son métodos generales.
- Y además unos integradores específicos para movimientos planetarios, que son simplécticos y simétricos.

Matlab es capaz de calcular la evolución en el tiempo de sistemas de ecuaciones diferenciales ordinarias de primer orden. Sin embargo, el problema de N cuerpos es un sistema de ecuaciones diferenciales de orden dos, por lo que es necesario transformar esa ecuación en un sistema de ecuaciones diferenciales de primer orden. Para su solución incorpora siete integradores: los tres primeros son métodos no rígidos y los cuatro siguientes se utilizan ante una ecuación diferencial rígida.

- El **integrador ode45** es un integrador de un paso basado en la fórmula explícita Runge-Kutta-Fehlberg de orden 4 ó 5. Presenta una gran rapidez de cálculo, aunque esto implica un mayor error en la aproximación.

- El **integrador ode23** es un integrador de un paso basado en el método explícito Runge-Kutta de orden 2 ó 3. La ecuación de este método proporciona un resultado con una precisión de tercer orden y una estimación del error de segundo orden.
- El **integrador ode113** es un integrador Predictor-Corrector basado en la fórmula de Adam-Bashforth-Moulton. Es un método multipaso de orden variable (desde orden uno hasta orden doce).
- El **integrador ode15s** es un método multipaso de orden variable basado en fórmulas de diferenciación numérica. Ocasionalmente utiliza fórmulas de diferenciación regresiva, también conocidas como método Gear. El orden de este método puede variar entre uno y cinco.
- El **integrador ode23s** es un integrador de un solo paso, basado en la fórmula modificada de Rosenbrock de órdenes 3 y 2.
- El **integrador ode23t** es la implementación de la regla del trapecio.
- El **integrador ode23tb** es la implementación de una fórmula implícita Runge-Kutta con una primera etapa que usa la regla trapezoidal y una segunda etapa que es una fórmula de diferenciación regresiva de orden 2.

Todos los integradores de Matlab tienen control automático del error: el usuario fija el error que está dispuesto a admitir en la solución y Matlab ajusta el paso de la integración, para conseguir ese error, es decir, el integrador, en cada paso, estima el error local en la componente i -ésima de la solución. Este error debe ser menor o igual que el error aceptable, el cual es función de la tolerancia relativa, RelTol, y de la tolerancia absoluta, AbsTol, ambas especificadas por el usuario. Si el error cometido es mayor que el error aceptable, el integrador reducirá el tamaño del paso, volverá a calcular la aproximación en ese instante y comprobará de nuevo la condición del error. Así sucesivamente hasta que se encuentre el tamaño de paso adecuado, por lo que los integradores que usa Matlab son de paso variable. Sin embargo, estos integradores permiten al usuario obtener una solución equiespaciada y para ello interpolan la solución obtenida.

Por otro lado, disponemos además de los integradores simplécticos de E. Hairer [6, 7], enfocados principalmente a la resolución de sistemas hamiltonianos, donde se busca que la energía del sistema y los momentos lineales y angulares se mantengan prácticamente constantes. Estos integradores trabajan directamente con el problema de valor inicial formulado como una ecuación de segundo orden.

Hairer proporciona tres tipos de integradores [5]:

- El **integrador gni_irk2** usa un método implícito Runge-Kutta de orden 4, 8 ó 12.

- El **integrador gni_lmm2** es un método lineal multipaso para ecuaciones diferenciales ordinarias de segundo orden. Permite tres tipos de aproximaciones diferentes, todas ellas de orden ocho, simplécticas y simétricas.
- El **integrador gni_comp** basado en el método de Störmer/Verlet de orden 2, 4, 6, 8 ó 10.

En el caso de los integradores de Hairer el tamaño de paso es constante y es elegido por el usuario. Asimismo el orden de la aproximación y el tipo de método de integración son escogidos por el usuario.

Los errores que se introducen en la solución numérica del problema de valor inicial provienen del error de truncamiento y del error de redondeo. El error de truncamiento es función de la ecuación diferencial y del método numérico. Si todos los cálculos pudiesen realizarse con una precisión infinita, el error de truncamiento sería el único presente. El error de redondeo depende del hardware del ordenador y del programa usado, y es debido a la precisión no infinita de todo ordenador y al número máximo de dígitos que permite el programa (Matlab sólo es capaz de hacer cálculos con doble precisión (quince decimales)). Este error es de menor importancia que el error de truncamiento, excepto cuando se desea alcanzar una gran exactitud en los cálculos o cuando se quiere integrar durante un largo periodo de tiempo.

4. Diseño del Software

El objetivo de este proyecto es la simulación en $3D$ del problema de N cuerpos. Consta de dos partes bien diferenciadas: la integración del problema y la visualización de la solución.

Para generar las trayectorias seguidas por los N cuerpos es necesario conocer las condiciones iniciales de masas, radios, posiciones y velocidades de cada uno de los cuerpos, y a través de la integración de las ecuaciones que genera el movimiento de N cuerpos, se aproximan numéricamente las soluciones en una serie de instantes de tiempo. Es decir, como salida del integrador vamos a tener un vector con los instantes de tiempo, donde se han generado las soluciones, y una matriz con las posiciones y velocidades de los N cuerpos, en cada uno de esos instantes. Como la sintaxis de los dos tipos diferentes de integradores no es igual, ha sido necesario generar dos códigos independientes para la aproximación de las trayectorias.

En las visualizaciones espaciales de la solución del problema se ha hecho uso de una propiedad interna de los integradores, que permite controlar la salida y mostrarla gráficamente en cada instante de integración, en lugar de tener que esperar a que el integrador finalice su aproximación durante el intervalo de tiempo asignado.

Finalmente, para que el programa resultase más sencillo al usuario final, se ha creado un entorno basado en ventanas gráficas, que facilita la utilización del software, sin tener que hacer uso de la línea de comandos (como se puede ver en las figuras 1, 2 y 4). Para ello se ha usado la herramienta GUIDE que incorpora Matlab en sus últimas versiones, la cual permite crear, de una forma muy cómoda y sencilla, interfaces gráficas de usuario (GUIs), que pueden estar provistos de una serie de controles, botones, menús desplegables, etc, que permiten al usuario interactuar con la ventana.

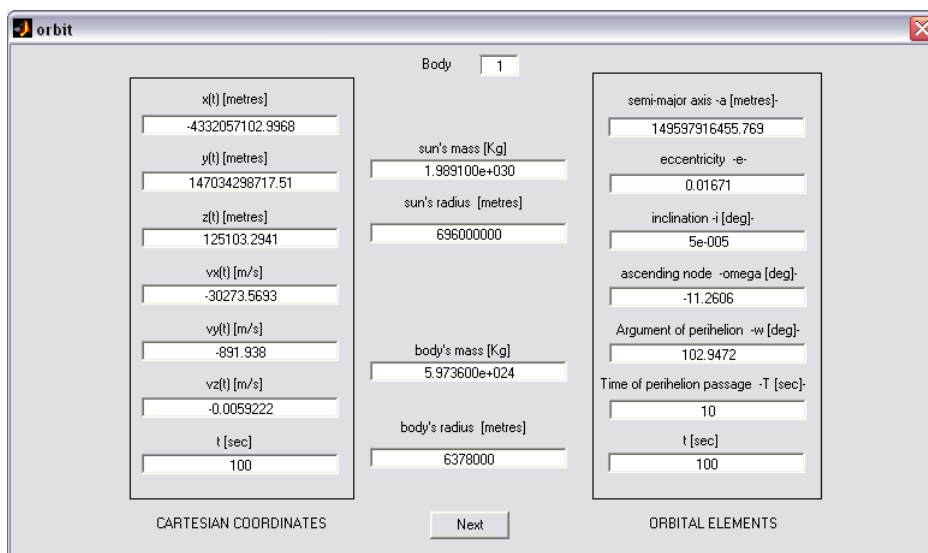


Figura 1.— Ventana del software para introducir las condiciones iniciales de cada cuerpo y sus equivalentes elementos orbitales.

5. Características Generales

El software que se ha creado es una aplicación sencilla, realizada por completo en Matlab. La simplicidad de manejo se ha conseguido mediante el uso de sucesivos interfaces gráficos, muy intuitivos, a través de los cuales se controlan todos los aspectos de la aplicación.

El programa es capaz de detectar todas las colisiones que se pueden producir entre dos o más cuerpos, siempre que los radios de todos ellos sean mayores que cero, y las resuelve conservando la cantidad de movimiento.

Antes de comenzar la simulación permite:

- Elegir entre varios integradores numéricos y modificar todas sus características internas.
- Los valores iniciales de los N cuerpos que se van a simular, pueden ser introducidos o bien en coordenadas cartesianas o en los elementos orbitales para cualquier tipo de

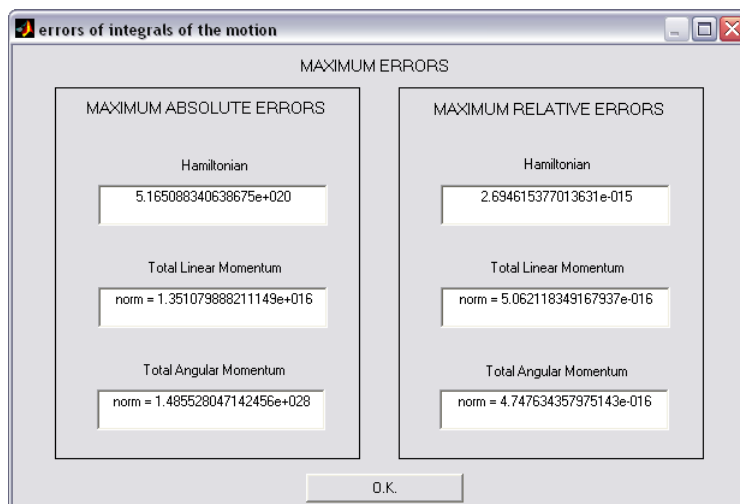


Figura 2.— Máximos errores relativos y absolutos cometidos en las integrales de movimiento, en una simulación de los planetas exteriores del Sistema Solar, donde se ha elegido uno de los integradores de Hairer.

movimiento (elíptico, hiperbólico o parabólico), coordenadas utilizadas en Mecánica Celeste, gracias a su mayor significado geométrico y astronómico (véase figura 1).

- Decidir en qué unidades se introducen todos los datos de los cuerpos y elegir el valor de la constante de gravitación universal, en dichas unidades. De esta forma permite visualizar, además de los movimientos generales, diferentes coreografías, donde las condiciones iniciales y el valor de la constante de gravitación universal están normalizados.

Durante la simulación, véase por ejemplo la figura 4, se puede:

- calcular las integrales de movimiento y se visualizan, en cada paso de integración.
- interactuar con la aplicación modificando el punto de vista, a través de los numerosos controles que se han creado en la pantalla principal.

Cuando la simulación finaliza, el software

- muestra las trayectorias en un espacio tridimensional y sus proyecciones en los tres planos principales del movimiento.
- exhibe los errores máximos cometidos en la conservación de las integrales de movimiento, respecto a los obtenidos con las condiciones iniciales, son mostrados al usuario (figura 2), y dependiendo de esa información puede elegir, para aproximar la solución, el integrador que comete el menor error en conservar dichas constantes.
- permite cargar configuraciones guardadas anteriormente.

- almacena las posiciones y velocidades, aproximadas en cada paso de integración.
- puede imprimir las trayectorias, seguidas por los cuerpos y guardarlas en un fichero gráfico, usando diferentes formatos.

A partir de las integrales de movimiento calculadas en cada paso de integración se puede realizar un análisis de los diferentes tipos de integradores implementados en el programa, para determinar cuál es el más exacto en la aproximación de las órbitas. Por ejemplo, para una simulación de los planetas interiores del Sistema Solar, cuyas condiciones iniciales se han obtenido de la base de datos de la NASA, y se muestran en [14], se obtiene la gráfica de la figura 3 donde se ha realizado una comparativa entre el método compuesto basado en el esquema Störmer/Verlet de orden 10 (que es uno de los integradores simplécticos) con el integrador predictor-corrector, ode113 de Matlab, que está basado en la fórmula de Adams-Bashforth-Moulton. Ambos métodos han realizado la mejor aproximación, dentro de cada grupo de integradores, de las órbitas de los planetas interiores del Sistema Solar. La simulación se realizó durante un periodo de tiempo de 38000000 segundos, es decir, aproximadamente, el tiempo que le cuesta a la Tierra dar una vuelta alrededor del Sol. Las tolerancias relativas y absolutas del integrador de Matlab se han fijado a 10^{-18} , y el tamaño de paso del método de Hairer tiene un valor de 90000 segundos. En la figura se muestran los errores cometidos por los integradores en la conservación del hamiltoniano, y se observa que el método compuesto de Hairer es el que comete el menor error. Además esos errores son del orden de 10^{-15} , por tanto, son debidos al redondeo del software más que a los errores de truncamiento del integrador. Estas mismas comparativas se pueden realizar con todos los integradores en función de la conservación de las tres integrales de movimiento (hamiltoniano, momento lineal y momento angular).

Como regla general para una primera aproximación son muy útiles los métodos de Matlab, porque varían el tamaño de paso internamente, y no lo tiene que elegir el usuario; además para tolerancias pequeñas el tiempo de CPU necesario suele ser muy bajo. Para aproximaciones más exactas será necesario utilizar el método Runge-Kutta, o el método compuesto de Hairer, con el tamaño de paso adecuado. Además en el caso de que se produzcan colisiones, como el tamaño de paso es modificado por los integradores de Matlab, son fácilmente detectables (mientras que en los integradores simplécticos es necesario realizar varias pruebas variando el tamaño de paso).

6. Manual de Uso

El funcionamiento del programa es muy sencillo. Actualmente sólo puede usarse como una aplicación integrada dentro de Matlab. Para conocer detalladamente la manera de utilizarlo, recomendamos descargar su manual de uso de la siguiente dirección de internet,

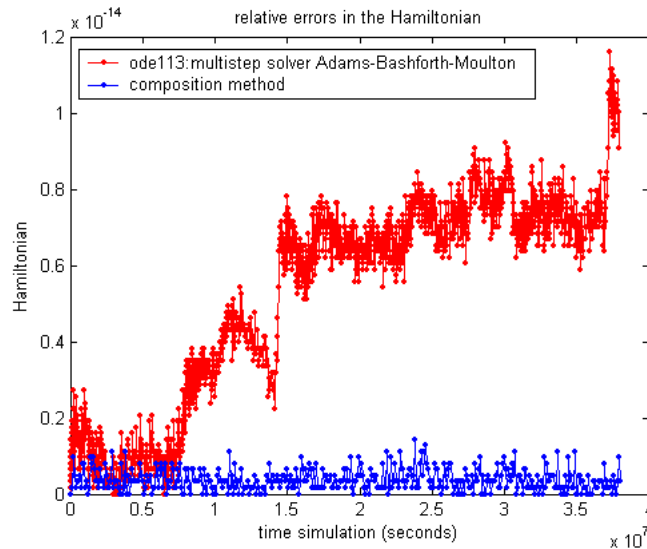


Figura 3.— Errores cometidos en la conservación del hamiltoniano por el integrador ode113 de Matlab y el método 1033 del integrador compuesto de Hairer en una simulación de las trayectorias seguidas por los planetas interiores del Sistema Solar.

donde además se encuentra el software de simulación para las dos últimas versiones de Matlab (6.5 y 7.0):

http://es.geocities.com/nemesis_mamen/descargas.html

7. Ejemplos

El propósito de este apartado es mostrar el resultado final de la simulación de varios ejemplos que se han simulado, comprobando el correcto funcionamiento del programa.

En la figura 5 se muestran dos configuraciones reales del Sistema Solar, en una se observan las órbitas seguidas por los planetas interiores y en la otra las de los planetas exteriores, donde las condiciones iniciales se obtuvieron a través de los elementos orbitales que figuran en la base de datos de la NASA, y en coordenadas cartesianas se pueden encontrar en [14].

Recientemente Chenciner y Montgomery [4] descubrieron la existencia de una solución plana del problema de tres cuerpos, con masas iguales, donde las tres masas siguen la misma trayectoria en forma de ocho, revolucionando a la comunidad de los sistemas dinámicos. Utilizando como condiciones iniciales los datos de [9] y mediante haciendo uso del integrador ode113, con unos valores de tolerancia de 10^{-6} y un valor de $G = 1 \frac{Nm^2}{Kg^2}$, las trayectorias seguidas por los tres cuerpos se pueden ver en la gráfica de la derecha de la figura 6.

Simó [16] descubrió otras soluciones para el caso plano de los tres cuerpos (hasta 345 distintas) para el caso de N cuerpos ($3 < N < 799$) dándoles el nombre de coreografías

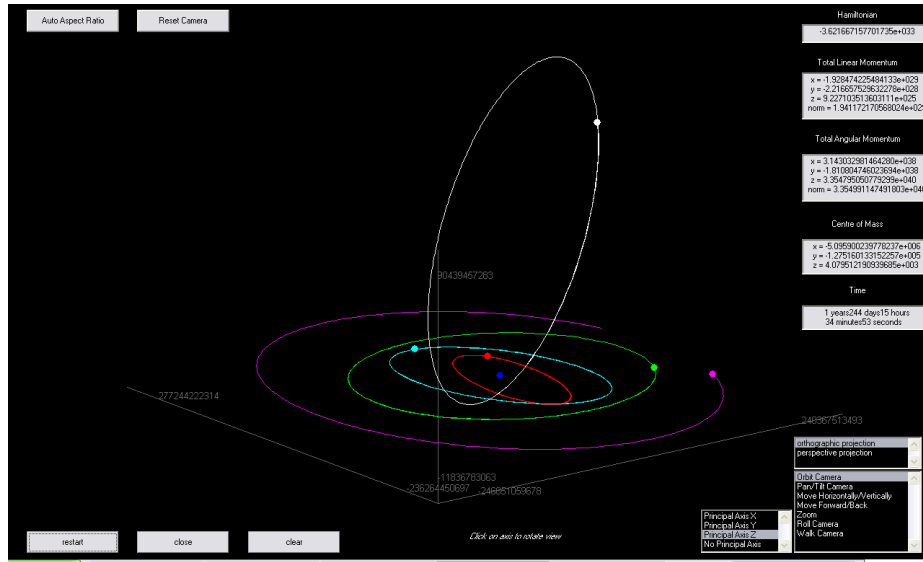


Figura 4.— Ventana gráfica donde se simula el movimiento de N cuerpos en 3D, en este caso la simulación corresponde a los planetas interiores del Sistema Solar más un cometa de periodo corto: Phaeton.

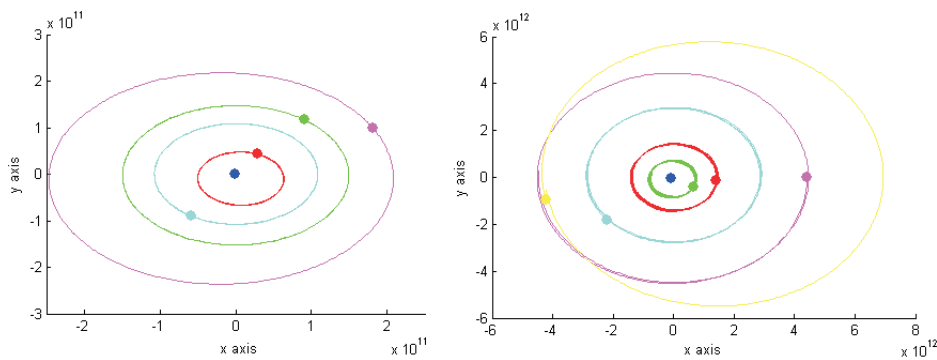


Figura 5.— Proyección en el plano $0xy$ de las trayectorias de los planetas interiores (izquierda) y exteriores (derecha) del Sistema Solar.

porque los cuerpos parece que bailan a lo largo de su trayectoria. En la figura 6 se pueden observar dos ejemplos, una configuración del ocho con cuatro cuerpos cuyas condiciones iniciales se han obtenido en [9] y una coreografía de seis cuerpos con condiciones iniciales extraídas de [8]. La simulación se realizó con el integrador compuesto de orden 10 con un paso de $h = 0,01$ segundos.

Muñoz-Almaraz aborda en su tesis doctoral [13] el problema de la continuación numérica y posibles bifurcaciones de la solución en forma de ocho del problema de tres cuerpos calculando varias familias de órbitas periódicas. Utilizando las condiciones iniciales que él ha obtenido para un caso tridimensional, se obtienen las trayectorias que se muestran en la figura 7, mediante una simulación de 7 segundos donde se ha utilizado el integrador compuesto de Hairer con un tamaño de paso de 0.007 segundos.

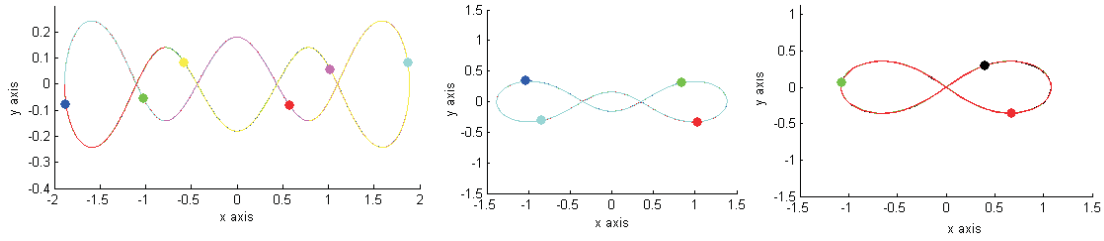


Figura 6.— Trayectorias con forma de ocho seguidas por seis, cuatro y tres cuerpos.

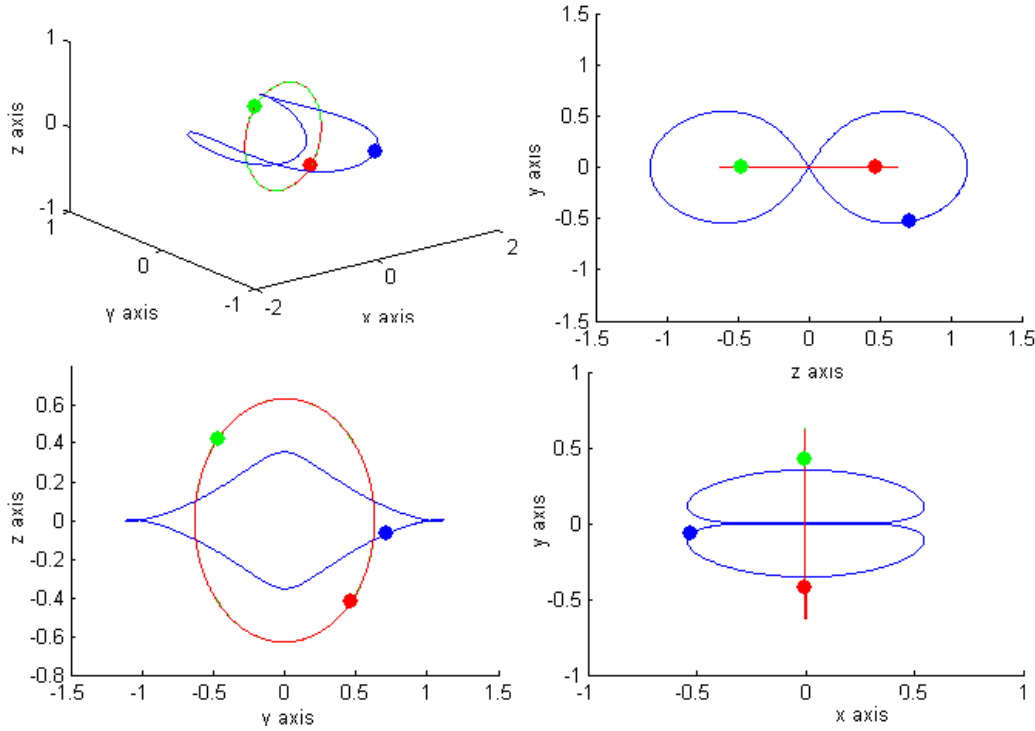


Figura 7.— Bifurcación de la solución en forma de ocho de F.J. Muñoz-Almaraz para el caso tridimensional y su proyección en los tres planos del movimiento.

Numerosas soluciones no triviales de trayectorias planas seguidas por tres cuerpos fueron encontrada por R. Broucke [3]. Para obtener las condiciones iniciales necesarias en la simulación se realizó un cambio de coordenadas baricéntricas a heliocéntricas. De ese modo la solución encontrada concuerda con la encontrada por Broucke como se observa en la figura 8, la cual fue obtenida utilizando el integrador ode113 con unos valores de tolerancias de 10^{-6} .

8. Conclusiones

El objetivo inicial de este proyecto era la elaboración de un programa informático, usando el lenguaje de programación Matlab, en el que dados N cuerpos (siendo $N > 1$), de masas arbitrarias, con unas condiciones iniciales de posición y velocidad cualesquiera

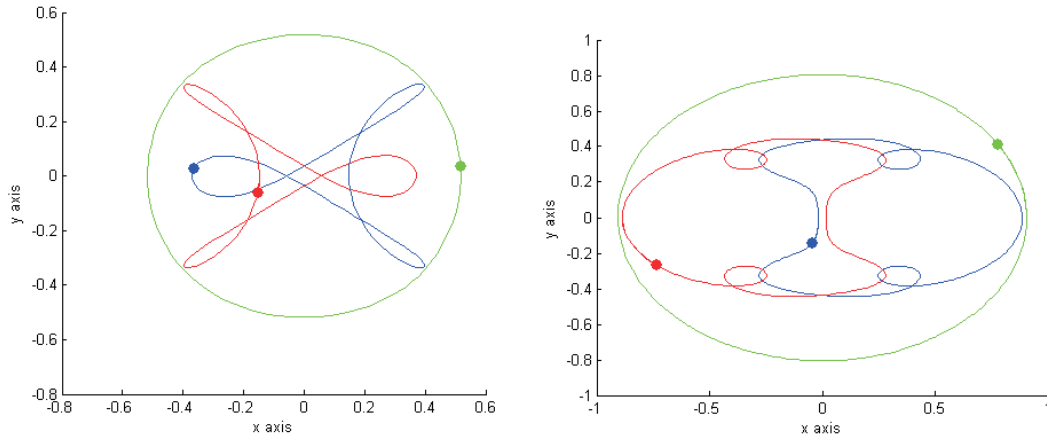


Figura 8.— Proyección en el plano $0xy$ de la solución periódica y plana, número 4 (izquierda) y número 122 (derecha) de R. Broucke [3].

en el espacio de tres dimensiones, se aproximasen numéricamente, con el menor error posible, las ecuaciones del movimiento de los N cuerpos, en un cierto intervalo prefijado, y además se visualizasen las trayectorias en tiempo real, en un espacio tridimensional.

El software creado es, bajo nuestro punto de vista, un software de propósito general y cumple, con creces, las expectativas iniciales. Además esta aplicación va a poder ser utilizada tanto por estudiantes como por investigadores en Mecánica Celeste.

Tiene un mejor funcionamiento que el software del que partió la idea de este proyecto: el programa de libre distribución de nombre Gravitation Ltd [15], construido por Jeff Rommereide, y que en la actualidad está en la versión 5.0, cuyo principal problema es, además de la imposibilidad de realizar simulaciones en $3D$, que sólo está disponible para Mac OS.

Durante la programación del código, necesario para el correcto funcionamiento de la aplicación, se descubrió un software en $3D$ de nombre Newton's Aquarium [11], con un entorno muy atractivo. Sin embargo, el movimiento interactivo presenta una gran deficiencia, el programa es incapaz de simular ciertas coreografías de N cuerpos y sólo permite usar el integrador Runge-Kutta explícito de cuarto orden. Todas estas deficiencias, junto con el hecho de que no es un software gratuito, hacen que nuestro simulador siga siendo la mejor opción actualmente, para tratar en $3D$ problemas reales, correspondientes a la dinámica del Sistema Solar o de cualquier otro sistema planetario o galáctico.

Referencias

- [1] Abad, A., Docobo J. A. y Elipe A.: 2002, *Curso de Astronomía*, Prensas Universitarias de Zaragoza, Zaragoza.

- [2] Boccaletti, D. y Pucacco, G.:1996, *Theory of Orbits. Volume 1: Integrable Systems and Non-Perturbative Methods*, Springer-Verlag, Berlín.
- [3] Broucke, R. y Boggs, D.: 1975, ‘Periodic orbits in the planar general three-body problem’, *Celest. Mech.* **11**, 13-38.
- [4] Chenciner, A. y Montgomery, R.: 2000, ‘A remarkable periodic solution of the three-body problem in the case of equal masses’, *Ann. Math.* **152**, 881-901
- [5] Hairer, E., Lubich, C. y Wanner G.: 2002, *Geometric Numerical Integration. Structure-Preserving Algorithms for ODE*, Ed Springer-Verlag, Berlín.
- [6] Hairer, E. y Hairer, M.: 2003, ‘GniCodes-Matlab programs for geometric numerical integration’, *Frontiers in Numerical Analysis (Durham 2002)*, Springer-Verlag, Berlin.
- [7] Hairer, E. y Lubich, C.: 2004, ‘Symmetric multistep methods over long time’, *Numer. Math.* **97**, 699-723.
- [8] Kapela, T. y Zgliczynski, P.: 2003, ‘The existence of simple choreographies for N -body problem. A computer assisted proof’, *Nonlinearity* **16**, 1899-1918.
- [9] Kotovych, O. y Bowman, J. C.: 2001, ‘An exactly conservative integrator for the n -body problem’, *J. Phys.* **35**, 7849-63
- [10] Lambert, J. D.: 1991, *Numerical Methods for Ordinary Differential Systems: The Initial Value Problem*, Ed John Wiley, Chichester.
- [11] Living Graphs: Programa Newton’s Aquarium. <http://www.livinggraphs.com/enu/products/na>
- [12] Meyer, K. R. :1999, *Periodic Solutions of the N -Body Problem*, Ed Springer, Berlín.
- [13] Muñoz-Almaraz, F. J.: 2003, ‘Continuación y Bifurcaciones de órbitas Periódicas en Sistemas Hamiltonianos con Simetría’, Tesis doctoral, Universidad de Sevilla, Sevilla.
- [14] Peña Moreno, M. C.: 2005, ‘Construcción de un simulador del Sistema Solar mediante la integración del problema de N cuerpos’, Proyecto Fin de Carrera, Universidad Pública de Navarra, Pamplona.
- [15] Rommereide, J.: Programa Gravitation Ltd. <http://www.pure-mac.com/astronomy.html>
- [16] Simó, C. Animaciones de distintas coreografías del problema de n cuerpos, descubiertas por C. Simó: <http://www.maia.ub.es/dsg/nbody.html>.

