

ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE CIENCIAS

**ANÁLISIS NUMÉRICO DE LA ESTABILIDAD ORBITAL
PLANETARIA EN FUNCIÓN DE LA VELOCIDAD DE
PROPAGACIÓN DE LA INTERACCIÓN GRAVITATORIA
NEWTONIANA**

**TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL TÍTULO DE
FÍSICO**

PROYECTOS DE INVESTIGACIÓN

GARY LEONEL GUAMÁN TOASA
gary.guaman91@gmail.com

Director: RAMON XULVI-BRUNET, PH.D.
ramon.xulvi@epn.edu.ec

QUITO, OCTUBRE 2023

DECLARACIÓN

Yo GARY LEONEL GUAMÁN TOASA, declaro bajo juramento que el trabajo aquí escrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y que he consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración cedo mis derechos de propiedad intelectual, correspondientes a este trabajo, a la Escuela Politécnica Nacional, según lo establecido por la Ley de Propiedad Intelectual, por su reglamento y por la normatividad institucional vigente.

Gary Leonel Guamán Toasa

CERTIFICACIÓN

Certifico que el presente trabajo fue desarrollado por GARY LEONEL GUAMÁN TOASA, bajo mi supervisión.

Ramon Xulvi-Brunet, Ph.D.
Director del Proyecto

AGRADECIMIENTOS

A todos ustedes, mis queridos abuelos, padres, hermanos, amigos y profesores, les agradezco de corazón por su incondicional apoyo, confianza y creencia en mí. Su presencia en mi vida ha sido fundamental y estoy eternamente agradecido por su influencia positiva en mi camino hacia el logro de mis metas.

Índice general

Resumen	VIII
Abstract	IX
1 Introducción	1
2 Marco Teórico	4
2.1 Algoritmo de velocidades de Verlet	4
2.1.1 Ventajas del Algoritmo de velocidades de Verlet	4
2.2 Deducción de Laplace sobre la velocidad finita de la gravedad	6
3 Incorporando velocidad finita en el algoritmo.	9
3.1 Descripción del método utilizado para la implementación del algoritmo de Velocidad de Verlet con velocidad finita de la gravedad.	9
3.1.1 Cálculo de δt	10
3.2 Funciones de error para evaluar la exactitud del nuevo algoritmo	12
3.3 Modificación del algoritmo	14
3.4 Simplecticidad del nuevo algoritmo	16
4 Resultados	17
4.1 Determinación del valor adecuado Δt (paso temporal del algoritmo) para cada sistema.	17
4.1.1 Cálculo de Δt para el sistema Tierra-Luna	18
4.1.2 Cálculo de Δt para el sistema Sol-Tierra-Luna	18
4.1.3 Cálculo de Δt para el sistema Solar (9 cuerpos)	19
4.2 Simulación el sistema Tierra-Luna	19
4.3 Simulación el sistema de 3 cuerpos Sol-Tierra-Luna	22
4.4 Simulación del sistema solar (9 cuerpos)	26
5 Conclusiones	33

6 Anexo	35
6.1 Anexo 1: Demostración de la simplecticidad del algoritmo de Velocidades de Verlet.	35
6.2 Anexo 2: Límite de VPG debido al determinante de la ecuación de δt	39
6.3 Anexo 3: Cálculo de δt con un mayor orden de precisión.	40
6.4 Anexo 4: Simplecticidad del nuevo algoritmo.	41
6.5 Anexo 5: Código Fortran90.	47
Bibliografía	52

Índice de figuras

2.0 Deducción de Laplace	6
3.1 Interacción de dos partículas con VGP finita.	9
4.1 Órbita Lunar al rededor de la Tierra desde el año 0 hasta 1,5 años. . .	20
4.2 Órbita Lunar desde el año 9998.5 hasta el año 1000.	20
4.3 Distancia Tierra-Luna desde el año 0 hasta el año 1.	21
4.4 Distancia Tierra-Luna desde el año 9999 hasta el año 1000.	21
4.5 Orbita Tierra-Luna desde año 0 hasta el año 5.	23
4.6 Orbita Tierra-Luna desde año 9995 hasta el año 1000.	23
4.7 Orbita Sol-Tierra desde el año 0 hasta el año 5.	24
4.8 Orbita Sol-Tierra desde el año 9995 hasta el año 1000.	24
4.9 Distancia Sol-Tierra desde año 0 hasta el año 5.	25
4.10 Diferencia entre las distancias de las orbitas Sol-Tierra para diferentes VPG, desde año 0 hasta el año 5.	25
4.11 Distancia Sol-Tierra desde el año 9995 hasta el año 1000.	26
4.12 Órbita Mercurio-Sol desde el año 990 hasta el año 1000.	27
4.13 Distancia Mercurio-Sol desde el año 990 hasta el año 1000.	28
4.14 Diferencia entre las distancias de las orbitas Sol-Mercurio para diferentes VPG.	28
4.15 Orbita de Jupiter desde el año 0 hasta el año 1000.	29
4.16 Distancia Jupiter-Sol desde el año 800 hasta el año 1000.	29

4.17	Diferencia entre las distancias de las orbitas Sol-Jupiter para diferentes VPG.	30
4.18	Orbita de Neptuno desde el año 0 hasta el año 1000.	30
4.19	Distancia Neptuno-Sol desde el año 800 hasta el año 1000.	31
4.20	Diferencia entre las distancias de las orbitas Sol-Neptuno para diferentes VPG.	31

Índice de cuadros

4.1	Valores de ϵ y ϵ' para el sistema T-L	18
4.2	Valores de ϵ y ϵ' para el sistema S-T-L	18
4.3	Valores de ϵ y ϵ' para el sistema Solar	19
4.4	Condiciones iniciales para el sistema Tierra-Luna	19
4.5	Condiciones iniciales sistema Sol-Tierra-Luna	22
4.6	Condiciones iniciales Sistema Solar	27

Resumen

En este trabajo se desarrolla un algoritmo computacional de dinámica molecular de n cuerpos que incorpora la propagación de la interacción a velocidad finita, utilizando como base el algoritmo de Velocity Verlet y las ecuaciones de la dinámica newtoniana. La necesidad de abordar este fenómeno radica en comprender y analizar sistemas complejos que presentan retrasos temporales o espaciales en su interacciones. En este trabajo, por su relevancia histórica y filosófica en la física, elegimos como ejemplo de interacción la gravedad newtoniana.

El algoritmo desarrollado es utilizado para simular sistemas de 2, 3 y 9 cuerpos. Estas simulaciones demuestran la capacidad del algoritmo para analizar qué se podría esperar de un sistema de n cuerpos con interacción gravitatoria Newtoniana que se propaga a velocidad finita.

Abstract

In this work, a computational algorithm for n-body molecular dynamics is developed, incorporating the propagation of interaction at finite velocity. The algorithm is based on the Velocity Verlet algorithm and the equations of Newtonian dynamics.

The motivation behind addressing this phenomenon lies in the need to comprehend and analyze complex systems that exhibit temporal or spatial delays in their interactions. For the purposes of this study, the example of Newtonian gravity is chosen as the interaction to investigate, given its historical and philosophical relevance in physics.

The developed algorithm is employed to simulate systems consisting of 2, 3, and 9 bodies. These simulations demonstrate the algorithm's capability to analyze the behavior of an n-body system with Newtonian gravitational interaction that propagates at finite velocity.

Capítulo 1

Introducción

En física existen varios fenómenos que presentan ciertas características de retraso ya sea temporal o espacial descritos por ecuaciones diferenciales con retardo (delay differential equations, DDE), por ejemplo:

- En el contexto del electromagnetismo, las ecuaciones diferenciales retardadas pueden ser utilizadas para describir sistemas en los que la respuesta de un campo electromagnético depende de eventos pasados. Por ejemplo, en el estudio de circuitos eléctricos con elementos retardados, como líneas de transmisión o cables, las señales electromagnéticas experimentan retardos en su propagación. [9].
- Mecánica celeste: Uno de los problemas que surgen en la mecánica celeste moderna es la necesidad de una integración numérica precisa de las ecuaciones de movimiento dinámico de la Luna. La acción de las fuerzas de marea se modela con un retardo temporal, por lo que el movimiento de la Luna se describe mediante una ecuación diferencial con retardo. [1]
- Dinámica de poblaciones: En ecología, las DDE se utilizan para modelar la dinámica de poblaciones que están sujetas a factores como la competencia, la predación, la migración y la estacionalidad. El retraso en la respuesta de una población a un cambio en el medio ambiente es un ejemplo de fenómeno que se puede modelar mediante una DDE. [13] [14]
- Óptica: En óptica, las ecuaciones diferenciales con retardos temporales surgen con frecuencia en el estudio de la dinámica no lineal de los láseres con retroalimentación óptica. [6]

La solución de ecuaciones diferenciales con retardo (DDE) representa un desafío, ya que muchas de ellas no tienen solución analítica. Por lo tanto, es de gran interés utilizar métodos numéricos para obtener soluciones aproximadas.

Como muestra del interés en estos fenómenos hacemos referencia al artículo del 2022 "DelayDiffEq: Generating Delay Differential Equation Solvers via Recursive Embedding of Ordinary Differential Equation Solvers"[16]. Este artículo aborda un tema importante en las ecuaciones diferenciales con retardos (DDEs). Destaca el desarrollo de DelayDiffEq, un paquete de software en Julia que resuelve eficientemente DDEs con retardos dependientes del estado. Su diseño aprovecha las funcionalidades de paquetes OrdinaryDiffEq, acelerando su desarrollo. Es un ejemplo de cómo un diseño de software puede mejorar la eficiencia en la computación científica.

Otra muestra del interés en estos fenómenos es la de 2019 "Numerical algorithms for time-delayed Newtonian gravity"(Algoritmos numéricos para la gravedad newtoniana con retardos temporales) [5]. En este artículo se explora la idea de que la precisión de la órbita de Mercurio puede ser explicada por el efecto de una velocidad finita de la gravedad, utilizando un análisis numérico moderno. Dicho estudio implementa el algoritmo del Runge-Kutta que permite simular la interacción gravitatoria newtoniana, considerando la velocidad finita de la gravedad igual a la velocidad de la luz, y evaluar su precisión y estabilidad.

La conclusión principal del proyecto citado es que usando solo una gravedad Newtoniana con retraso temporal no se puede predecir correctamente la precesión anómala del perihelio de Mercurio. Esto no se debe a un problema numérico, sino más bien una propiedad del sistema simulado. Los métodos numéricos utilizados en el proyecto también podrían tener aplicaciones en otros problemas como el cálculo de potenciales de Liénard-Wiechert en electrodinámica [5].

El estudio de la velocidad de propagación de la gravedad fue en su día un tema importante en la física teórica. La ley de gravitación universal de Newton, que establece la fuerza gravitatoria entre dos objetos en función de sus masas y la distancia entre ellos, asume que la gravedad actúa instantáneamente. Sin embargo, esta ley no puede explicar algunos fenómenos observados, como la precesión de la órbita de Mercurio. Por lo tanto, algunos científicos comenzaron a considerar la posibilidad de que la gravedad tenga una velocidad finita de propagación, y en 1805, Laplace propuso por primera vez un cálculo para esta velocidad. Aunque su explicación de que la gravedad se propaga a través de un medio material invisible llamado éter

resultó ser incorrecta, abrió la puerta a nuevas investigaciones y desarrollos en la física. [7]

En 1898, Gerber calculó el valor correcto del avance del perihelio de Mercurio y, basándose en esta fórmula, calculó que la velocidad de propagación de la gravedad era de 305,000 km/s. Este fue un importante avance en la comprensión de la naturaleza de la gravedad y su velocidad de propagación, cálculo que también fue incorrecto. [4]

En 1915, Albert Einstein propuso la teoría de la relatividad general, que incluía la idea de que la gravedad se propagaba a través del espacio a una velocidad finita. Esta teoría explicaba de manera más precisa la forma en que la gravedad afecta a la curvatura del espacio-tiempo y cómo los cuerpos en movimiento influyen en la propagación de la gravedad.

Como ejemplo de interacción a velocidad finita, consideramos la interacción gravitatoria Newtoniana a velocidad de propagación finita, siguiendo la idea propuesta en su día por Laplace.

El presente trabajo tiene como objetivo desarrollar un primer algoritmo de dinámica molecular para n cuerpos bajo interacción gravitatoria que se propaga a velocidad finita.

Es importante resaltar dos factores. En primer lugar, hasta la fecha no existe un algoritmo estándar para abordar fenómenos con características de retraso, y nuestra propuesta se presenta como una idea más para enfocar el problema. Por tanto, el algoritmo propuesto se considera un primer acercamiento que requerirá mejoras adicionales en futuras investigaciones. En segundo lugar, mencionamos que esta tesis no profundiza teóricamente en una teoría con velocidad finita de la gravedad.

Capítulo 2

Marco Teórico

2.1. Algoritmo de velocidades de Verlet

En esta tesis se desarrolla un algoritmo basado en el algoritmo de velocidades de Verlet. Se realiza a continuación una breve descripción de este método numérico.

Las ecuaciones que describen el movimiento de un sistema mecánico en términos de su posición y velocidad son las ecuaciones de Newton:

$$\frac{d^2x}{dt^2} = a(x)$$

donde x es la posición de la partícula, t es el tiempo, y $a(x)$ es la aceleración de la partícula en función de su posición. Y los pasos del algoritmo son [15] [2]:

$$\begin{aligned}x_{n+1} &= x_n + \Delta t v_n + \frac{1}{2} \Delta t^2 a(x_n) \\v_{n+1} &= v_n + \frac{1}{2} \Delta t [a(x_n) + a(x_{n+1})]\end{aligned}\tag{2.1}$$

Estas fórmulas se utilizan repetidamente para un número determinado de pasos temporales para obtener una solución completa para el sistema mecánico.

2.1.1. Ventajas del Algoritmo de velocidades de Verlet

Se utiliza este algoritmo como base debido a que la aproximación de velocidades de Verlet es simple, eficiente y cuenta con un segundo orden de precisión.

A nivel global, este algoritmo es de segundo orden, lo que significa que su error global es proporcional al cuadrado del tamaño del paso de integración. Esto implica

que al disminuir a la mitad el tamaño del paso de integración, el error global se reduce aproximadamente a la cuarta parte. Por lo tanto, ofrece una buena precisión en la evolución del sistema a lo largo del tiempo [12][15].

Sin embargo, a nivel local, es decir, en cada paso de la integración, el método es de cuarto orden. Esto significa que su error local de truncamiento es proporcional al cuarto poder del tamaño del paso de integración. En otras palabras, en cada paso de integración se desprecian términos de las expansiones de Taylor que corresponden a un cuarto orden en magnitud. Este mayor orden de precisión permite que el algoritmo capture con mayor exactitud los detalles locales de la evolución del sistema en comparación al algoritmo de Euler que es de primer orden global y segundo local[12]. Es importante destacar que estas propiedades teóricas del algoritmo de velocidades de Verlet son una ventaja significativa en términos de estabilidad en la simulación de sistemas físicos.

Además, el algoritmo es invariante al retroceso temporal y en cuanto al momento angular y la energía se conservan por mucho más tiempo que algoritmos no simplécticos, lo que lo convierte en una excelente elección para la simulación de sistemas físicos en la mecánica clásica. Los algoritmos simplécticos ofrecen una ventaja importante en la simulación de sistemas físicos, ya que garantizan la estabilidad global. Aunque existen aproximaciones no simplécticas más precisas, estas pueden desviarse significativamente de su valor inicial en tiempos prolongados [12].

Si bien la integración de las ecuaciones de Newton en un tiempo finito puede ser representada como una transformación de coordenadas canónica que preserva las ecuaciones de Hamilton, es altamente deseable que una aproximación numérica también sea una transformación canónica. Esta aproximación es conocida como una aproximación simpléctica. [17]

Un algoritmo simpléctico es una transformación canónica en el espacio de fases, lo que implica que cumple con el teorema de Liouville. Este teorema establece la invarianza del volumen del espacio de fases bajo una transformación canónica, lo que evita situaciones en las que las coordenadas (y, por lo tanto, la energía) aumenten sin límite ya que esto podría alterar el volumen del espacio de fases [3] [17].

La prueba de que dicha transformación es canónica se reduce a demostrar que el determinante de su Jacobiano es igual a la unidad [8] [17]:

$$\text{Det } J = 1 \tag{2.2}$$

Una deducción completa de la simplecticidad del algoritmo de Velocidades de

Verlet se presenta en el anexo 1.

2.2. Deducción de Laplace sobre la velocidad finita de la gravedad

Laplace presentó uno de los primeros intentos para poner un límite inferior a la velocidad de propagación de la gravedad en 1805 mediante su tratado titulado 'A Treatise in Celestial Mechanics', deducción presentada a continuación [11]

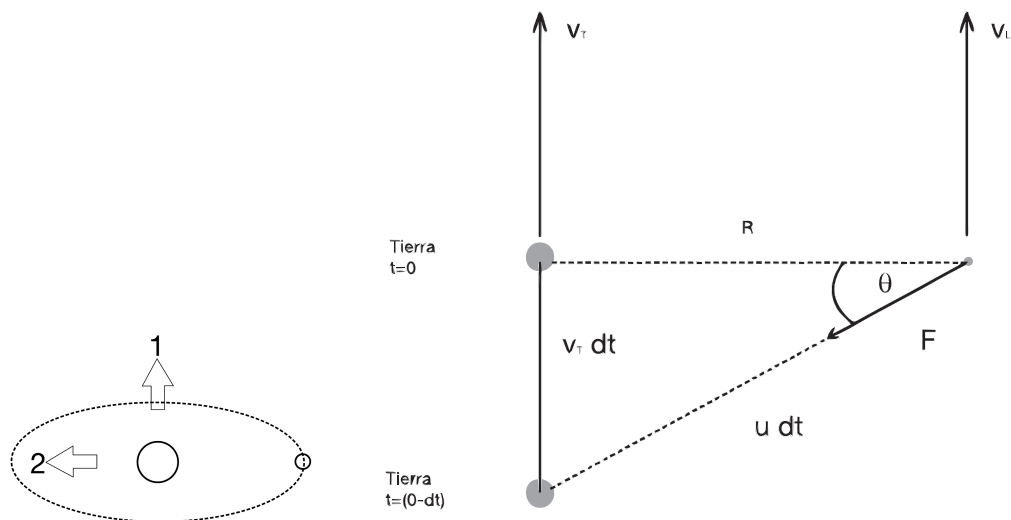


Figura 2.0: Deducción de Laplace. Interacción entre la Tierra y la Luna con una Gravedad Newtoniana con velocidad de propagación finita. El punto gris mas grande representa a la tierra y el gris pequeño de la derecha es la Luna.

Sea el sistema Tierra-Luna que se mueve en la dirección 1 donde:

$$v_T = 3 \times 10^4 m/s = \text{Velocidad orbital de la tierra al rededor del Sol}$$

$$v_L = \text{Velocidad de la Luna}$$

$$R = \text{Distancia entre la Tierra y la Luna}$$

$$u = \text{Velocidad de la gravedad}$$

Se asume que la fuerza sobre la luna apunta a una posición anterior de la tierra $r(t - dt)$, formando una componente de fuerza con un ángulo θ que frena el movimiento de la luna.

$$\begin{aligned}\text{sen } \theta &= \frac{v_T dt}{u dt} = \frac{v_T}{u} \\ \text{sen } \theta &= \frac{F_\theta}{F}\end{aligned}$$

$$F_\theta = -\frac{GMm}{R^2 + (v_T dt)^2} \frac{v_T}{u} \quad (2.3)$$

Si $dt \approx 0$

$$F_\theta \approx -\frac{GMm}{R^2} \frac{v_T}{u} \quad (2.4)$$

El trabajo hecho por esta fuerza sobre la luna es:

$$T = \vec{F} \cdot \Delta \vec{r}_m$$

diferenciando con respecto al tiempo tenemos el cambio de energía de la luna :

$$\frac{dE}{dt} = F_\theta \frac{dr_m}{dt} = F_\theta v_m \quad (2.5)$$

Siendo la energía del sistema:

$$E = \frac{-GMm}{2R} \Rightarrow \frac{dE}{dt} = \frac{GMm}{2r^2} \frac{dR}{dt} \quad (2.6)$$

Uniando las dos ecuaciones anteriores tenemos:

$$\begin{aligned}F_\theta v_m &= \frac{Gmm}{2r^2} \frac{dR}{dt} \\ -\frac{GMm}{R^2} \frac{v_T}{u} v_m &= \frac{GMm}{2r^2} \frac{dR}{dt}\end{aligned} \quad (2.7)$$

Al igualar la fuerza gravitacional con la centrípeta se obtiene:

$$v_m \approx \sqrt{\frac{GM}{R}}$$

Despejando $\frac{dR}{dt}$ de la ecuación (1.5):

$$\frac{dR}{dt} = -\frac{2v_m V_E}{u} = -2\sqrt{\frac{GM}{R}} \frac{V_E}{u}$$

Integrando esta ecuación obtenemos:

$$R^{3/2} = R_0^{3/2} - 3\sqrt{GM}\frac{V_E}{u}t = R_0^{3/2} - 3v_0\sqrt{R_0}\frac{v_T}{u}t = R_0^{3/2} \left(1 - \frac{6\pi v_T t}{u T_0}\right)$$

Donde T_0 es el periodo orbital de la luna.

El efecto de una velocidad finita de la gravedad es que la luna caerá sobre la superficie terrestre en un tiempo igual a:

$$\Delta t = \frac{c T_0}{6\pi v_T} \frac{u}{c} \approx \frac{3 \times 10^8 \text{ m/s} \cdot 2,5 \times 10^6 \text{ s}}{20 \cdot 3 \times 10^4 \text{ m/s}} \frac{u}{c} \approx 10^9 \frac{u}{c} \text{ s} \approx 300 \frac{u}{c} \text{ años}$$

$$\Delta t = 300 \frac{u}{c} \text{ años} \quad (2.8)$$

Esta es la ecuación que desarrolló Laplace. Como podemos observar, si suponemos que la luna tiene una edad de 3×10^9 años, entonces la relación entre la velocidad de propagación y la velocidad de la luz es la siguiente:

$$\frac{u}{c} \approx 10^7$$

Capítulo 3

Incorporando velocidad finita en el algoritmo.

3.1. Descripción del método utilizado para la implementación del algoritmo de Velocidad de Verlet con velocidad finita de la gravedad.

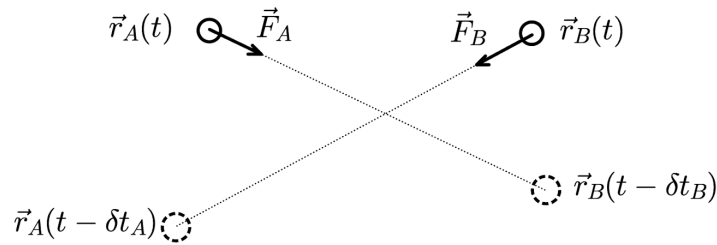


Figura 3.1: Interacción de dos partículas con VGP finita.

En primer lugar, se considera dos partículas A y B. El efecto de una velocidad de propagación finita de la gravedad (VGP) es equivalente a que la partícula A en la posición $\vec{r}_A(t)$ interactúe con la partícula B en la posición retardada $\vec{r}_B(t - \delta t_B)$. Donde:

$$\delta t_B = \frac{|\vec{r}_A(t) - \vec{r}_B(t - \delta t_B)|}{VGP} \quad (3.1)$$

$$\delta t_A = \frac{|\vec{r}_B(t) - \vec{r}_A(t - \delta t_A)|}{VGP} \quad (3.2)$$

δt representa el tiempo que tarda la interacción gravitatoria en alcanzar la otra

partícula.

Las ecuaciones de la fuerza de interacción son:

$$\vec{F}_A(t) = m_A \vec{a}_A(t) = -Gm_A m_B \frac{(\vec{r}_A(t) - \vec{r}_B(t - \delta t_B))}{|\vec{r}_A(t) - \vec{r}_B(t - \delta t_B)|^3} \quad (3.3)$$

$$\vec{F}_B(t) = m_B \vec{a}_B(t) = -Gm_B m_A \frac{(\vec{r}_B(t) - \vec{r}_A(t - \delta t_A))}{|\vec{r}_B(t) - \vec{r}_A(t - \delta t_A)|^3} \quad (3.4)$$

Generalizando esta deducción para el caso de n cuerpos, el algoritmo evalúa la interacción entre la partícula A y las demás partículas, representando esta interacción como una suma de las interacciones individuales:

$$\vec{F}_i(t) = m_i \vec{a}_i(t) = - \sum_j^N Gm_i m_j \frac{(\vec{r}_i(t) - \vec{r}_j(t - \delta t_j))}{|\vec{r}_i(t) - \vec{r}_j(t - \delta t_j)|^3} \quad (3.5)$$

Donde $1 \leq i, j \leq N$ representan los índices de las partículas del sistema.

3.1.1. Cálculo de δt

Usando una aproximación en series de Taylor de primer orden se escribe:

$$\vec{r}_A(t - \delta t_A) = \vec{r}_A(t) - \vec{v}_A(t) \delta t_A \quad (3.6)$$

Remplazamos en la ecuación 3.2 y elevamos al cuadrado.

$$[\vec{r}_B(t) - \vec{r}_A(t) + \vec{v}_A(t)(\delta t_A)] [\vec{r}_B(t) - \vec{r}_A(t) + \vec{v}_A(t)(\delta t_A)] = VGP^2 \delta t_A^2 \quad (3.7)$$

Sea:

$$\vec{r}_B(t) - \vec{r}_A(t) = \vec{r}(t) \quad (3.8)$$

$$|\vec{r}(t)|^2 + 2 [\vec{r}(t) \cdot \vec{v}_A(t)] \delta t_A + |\vec{v}_A(t)|^2 \delta t_A^2 = VGP^2 \delta t_A^2 \quad (3.9)$$

Cuya solución es:

$$\delta t_A = \frac{-2 [\vec{r}(t) \cdot \vec{v}_A(t)] \pm \sqrt{[2 [\vec{r}(t) \cdot \vec{v}_A(t)]]^2 - 4 \cdot [|\vec{v}_A|^2 - VGP^2] \cdot |\vec{r}(t)|^2}}{2 \cdot [|\vec{v}_A|^2 - VGP^2]} \quad (3.10)$$

Escogemos el valor con + para garantizar el valor positivo:

$$\delta t_A = \frac{\frac{[\vec{r}(t) \cdot \vec{v}_A(t)]}{VGP^2} + \sqrt{\frac{[\vec{r}(t) \cdot \vec{v}_A(t)]^2}{VGP^4} + \frac{[VGP^2 - |\vec{v}_A|^2]}{VGP^4} \cdot |\vec{r}(t)|^2}}{1 - \frac{|\vec{v}_A|^2}{VGP^2}} \quad (3.11)$$

De igual forma se procede para encontrar δt_B .

Al examinar detenidamente la siguiente ecuación, podemos observar que para valores donde $|\vec{v}_A(t)| \gg VGP$, la dependencia del tamaño del paso de integración δt_A es aproximadamente proporcional a $\delta t_A = \frac{|\vec{r}(t)|}{VGP}$.

Dado que en el algoritmo utilizamos las siguientes series de Taylor, que dependen del valor de Δt , para calcular la posición:

$$\vec{r}_n(t + \Delta t) = \vec{r}_n(t) + \vec{v}_n(t)(\Delta t) + \frac{1}{2}\vec{a}_n(t)(\Delta t)^2 + \sigma(\Delta t) \quad (3.12)$$

Donde $\sigma(\Delta t)$ representa la suma de todos los términos de mayor orden que se desprecian en la aproximación (error de aproximación).

Es importante asegurarnos de que el valor de δt utilizado en la aproximación para calcular la posición con un tiempo de retardo:

$$\vec{r}_A(t - \delta t_A) = \vec{r}_A(t) - \vec{v}_A(t)\delta t_A \quad (3.13)$$

sea menor o igual al paso temporal Δt :

$$\delta t \leq \Delta t \quad (3.14)$$

Esto asegura que el error global mantenga el mismo orden que el algoritmo de velocidades de Verlet en términos de precisión global.

Gracias a esta deducción, es importante tener en cuenta que cada sistema tiene una velocidad mínima de propagación que puede ser simulada. Si la velocidad de propagación es menor que esta velocidad mínima, el valor de δt superará el valor de

Δt , lo que significa que la aproximación de la posición con retardo ya no sería coherente. Por lo tanto, es crucial considerar este límite y asegurarse de que la velocidad de propagación sea compatible con el paso de integración elegido para mantener la coherencia de las aproximaciones en la simulación.

El valor del determinante de la ecuación (3.11) establece un límite para la velocidad de propagación. De forma simplificada, este límite establece que si dos cuerpos se alejan en línea recta, y la velocidad de propagación es menor que la velocidad de los cuerpos, el valor de δt será imaginario. Esto implica que en dicho escenario los cuerpos nunca interactuarán, lo cual no es especialmente relevante para una simulación. Para obtener más detalles sobre este límite, consultar el Anexo 2.

Eventualmente, en futuras investigaciones, será necesario mejorar la aproximación del valor de δt . En el Anexo 3 se presenta una deducción de δt con un orden mayor en la ecuación (3.5).

3.2. Funciones de error para evaluar la exactitud del nuevo algoritmo

La precisión en la simulación de sistemas planetarios es fundamental para obtener resultados confiables en la predicción del movimiento de los cuerpos celestes.

A continuación se definen dos funciones que permiten evaluar los términos de mayor orden en las expansiones de Taylor, para asegurarnos de que podemos despreciarlos debido a su pequeño valor. De esta forma, garantizamos que el código desarrollado en esta tesis obtendrá resultados válidos. Las funciones que se expanden en series de Taylor son:

$$\vec{r}_n(t + \Delta t) = \vec{r}_n(t) + \vec{v}_n(t)(\Delta t) + \frac{1}{2}\vec{a}_n(t)(\Delta t)^2 + \frac{1}{6}\dot{\vec{a}}_n(t)(\Delta t)^3 \quad (3.15)$$

$$\vec{r}_n(t - \delta t) = \vec{r}_n(t) - \vec{v}_n(t)(\delta t) + \frac{1}{2}\vec{a}_n(t)(\delta t)^2 \quad (3.16)$$

Donde Δt es el paso temporal del algoritmo y δt es el tiempo que tarda en viajar la interacción. A continuación se define las funciones ϵ y ϵ' .

Función ϵ

$$\epsilon = \frac{|\vec{a}(t + \Delta t) - \vec{a}(t)|}{|\vec{a}(t)|} \quad (3.17)$$

Se multiplica por $1 = \frac{\Delta t}{\Delta t}$

$$\epsilon = \frac{|\vec{a}(t + \Delta t) - \vec{a}(t)|}{\Delta t} \frac{\Delta t}{|\vec{a}(t)|} \quad (3.18)$$

Se aproxima el primer término de la derecha de la ecuación anterior como la derivada de la aceleración, lo cual solo es válido si el valor de Δt es suficientemente pequeño.

$$\epsilon = \frac{d|\vec{a}(t)|}{dt} \frac{\Delta t}{|\vec{a}(t)|} \quad (3.19)$$

Se despeja el valor $\frac{d|\vec{a}(t)|}{dt}$

$$\frac{d|\vec{a}(t)|}{dt} = \frac{\epsilon |\vec{a}(t)|}{\Delta t} = \dot{a}(t) \quad (3.20)$$

Se reemplaza en la expansión de Taylor (3.11):

$$\vec{r}_n(t + \Delta t) = \vec{r}_n(t) + \vec{v}_n(t)(\Delta t) + \frac{1}{2}\vec{a}_n(t)(\Delta t)^2 + \frac{1}{2}\vec{a}_n(t)(\Delta t)^2 \frac{\epsilon}{3} \quad (3.21)$$

La ecuación anterior muestra que el cuarto término en cada coordenada es $\frac{\epsilon}{3}$ veces el tercer término. Por ejemplo, si el valor de ϵ es 10^{-3} , el cuarto término es 3000 veces más pequeño que el tercer término. Esto ilustra la diferencia de magnitud entre ambos términos.

Función ϵ'

$$\epsilon' = \frac{|\vec{v}(t - \delta t) - \vec{v}(t)|}{|\vec{v}(t)|} \quad (3.22)$$

Se multiplica por $1 = \frac{\delta t}{\delta t}$

$$\epsilon' = \frac{|\vec{v}(t - \delta t) - \vec{v}(t)|}{\delta t} \frac{\delta t}{|\vec{v}(t)|} \quad (3.23)$$

Se aproxima el primer termino de la derecha de la ecuación anterior como la

derivada de la velocidad:

$$\epsilon' = \frac{d|\vec{v}(t)|}{dt} \frac{\delta t}{|\vec{v}(t)|} \quad (3.24)$$

$$|\vec{a}_n(t)| = \frac{\epsilon' |\vec{v}(t)|}{\delta t} \quad (3.25)$$

Se reemplaza en la expansión de Taylor (3.12):

$$\vec{r}_n(t - \delta t) = \vec{r}_n(t) - \vec{v}_n(t)(\delta t) + \frac{\epsilon'}{2} \vec{v}_n(t) \delta t \quad (3.26)$$

La ecuación anterior muestra que el tercer término es $\frac{\epsilon'}{2}$ veces el tercer término. Por ejemplo, si el valor de ϵ es 10^{-3} , el tercer término es 2000 veces más pequeño que el segundo término.

Se implementa el cálculo de estas dos funciones, ϵ y ϵ' , en cada paso de integración. Si en algún momento se supera un valor predefinido, como por ejemplo $\epsilon_{\max} = 10^{-3}$, el algoritmo se detendrá y será necesario repetir la simulación con un valor de Δt más pequeño o una velocidad de propagación más grande.

3.3. Modificación del algoritmo

Con la descripción de la interacción a velocidad finita de la sección anterior, procedemos a realizar modificaciones en el algoritmo de velocidades de Verlet en dos partes críticas:

- Calculamos el valor δt utilizando las ecuaciones (3.10) y (3.15) y las aceleraciones de las partículas utilizando la ecuación (3.5).
- Introducimos las condiciones de coherencia: $\delta t \leq \Delta t$ y $\epsilon \leq \epsilon_{\max}$.

Para una mejor explicación, se representa el código de forma didáctica con el menor número de pasos a continuación:

```
a(0)= Ecuacion 3.5 "PRIMER CALCULO DE ACELERACION"
do step=1,T
  r = r + dt*v + 0.5*dt**2*a
  do i = 1,N-1
    do j = i+1,N
```

$$\left. \begin{aligned} \delta t_{ij}, \delta t_{ji} &= \text{Ecuación 3.10} \\ r_{ij}(t - \delta t_{ij}) &= r_{ij} - v_{ij} * \delta t_{ij} \\ r_{ji}(t - \delta t_{ji}) &= r_{ji} - v_{ji} * \delta t_{ji} \\ a_{ij}, a_{ji} &= \text{Ecuación 3.5} \end{aligned} \right\} \text{Interacción modificada}$$

enddo

enddo

$$\left. \begin{aligned} \delta t &\leq \Delta t \\ \epsilon, \epsilon' &\leq \epsilon_{max} \end{aligned} \right\} \text{Condiciones de coherencia}$$

$$v = v + 0.5 * dt * a$$

enddo

Donde Δt (también denotado como dt) corresponde al paso de integración, δt es el tiempo de retardo, T es el número total de pasos a simular, N es el número de partículas, e i y j son los índices que representan las partículas.

En el caso de una velocidad de propagación muy baja en comparación con la velocidad de las partículas, es posible que las partículas sigan moviéndose en un movimiento rectilíneo uniforme de manera indefinida, sin que la interacción gravitacional las alcance.

Por otro lado, si la velocidad de propagación de la interacción gravitacional es grande en comparación con la velocidad de las partículas, estas pueden estar inicialmente en un movimiento rectilíneo uniforme durante un breve período de tiempo y luego comenzar a interactuar entre sí. En este caso, la interacción gravitacional alcanzaría a las partículas lo suficientemente rápido como para afectar su trayectoria.

Sin embargo, si se considera que las partículas ya interactuaron en una historia previa desconocida, los resultados obtenidos serán diferentes. En el caso del algoritmo original, donde la interacción se considera instantánea este comportamiento no ocurre.

En esta simulación, se realiza un cálculo inicial de la aceleración modificada al inicio del código para permitir la interacción de las partículas desde el principio de la simulación.

3.4. Simplecticidad del nuevo algoritmo

Se realizó un estudio sobre la simplicidad del algoritmo, y demostró que el algoritmo no es simpléctico al considerarlo como una transformación de coordenadas. Al calcular su jacobiano, se encontró que no es igual a 1. Esto indica que el algoritmo no cumple con la condición necesaria para ser una transformación canónica. Dicha deducción establece que:

$$\det[J_x] = \frac{1 - \frac{\Delta t}{2} \frac{\partial A}{\partial v_0}}{1 + \frac{\Delta t}{2} \frac{\partial A}{\partial v_1}} \quad (3.27)$$

$$\det[J] = \det[J_x] \cdot \det[J_y] \cdot \det[J_z] \neq 1 \quad (3.28)$$

Donde $\det[J_x]$ es el Jacobiano de la transformación en el eje x, cabe aclarar que la transformación tiene la misma forma en todos los ejes.

Se deduce que la transformación no es canónica, por ende el algoritmo no es simpléctico. Se considera importante esta deducción, ya que nos permite anticipar que la introducción de una velocidad finita en la propagación de la interacción resultará en una disminución de la estabilidad del algoritmo por defecto. La deducción completa se encuentra en el Anexo 4 de este documento.

Capítulo 4

Resultados

Se utilizó unidades astronómicas para medir las distancias, años para medir el tiempo y unidades de masa terrestre para medir la masa. Estas unidades son útiles en el estudio de sistemas planetarios. La constante gravitacional tiene un valor aproximado de $CG = 0,0001184$ en estas unidades. En cuanto a la precisión se utiliza 36 cifras decimales.

En todas las simulaciones se considera el centro de masas como sistema de referencia.

4.1. Determinación del valor adecuado Δt (paso temporal del algoritmo) para cada sistema.

Para calcular dicho valor Δt tenemos dos condiciones:

- Condición 1: ($\delta t \leq \Delta t$)

Cada sistema en particular presenta una velocidad de propagación mínima VPG_{min} , por debajo de la cual no se obtienen soluciones coherentes. En este límite, se cumple que $\delta t = \Delta t$. Las velocidades que podemos simular deben ser mayores a VPG_{min} que en primera instancia se puede aproximar con la siguiente formula:

$$VPG_{min} = \frac{D}{\Delta t} \quad (4.1)$$

Donde D es la distancia más grande del sistema.

Debido a esta dependencia del inverso del paso temporal, cuando se busca simular una *VP*G más pequeña, es necesario aumentar el valor de Δt , lo que afecta los valores de ϵ y ϵ' .

- Condición 2: ($\epsilon, \epsilon' \leq \epsilon_{max}$)

Los valores máximos de ϵ y ϵ' deben ser lo suficientemente pequeños como para garantizar que la aproximación sea válida.

4.1.1. Cálculo de Δt para el sistema Tierra-Luna

Aplicando las condiciones de coherencia generamos la siguiente tabla:

VGP	Δt	ϵ_{max}	ϵ'_{max}
$1 \times 10^8 c$	0,00001	$1,0301587 \times 10^{-3}$	$9,381948 \times 10^{-4}$
$1c$	0,00001	$1,0301584 \times 10^{-3}$	$9,381944 \times 10^{-4}$
$0,06c$	0,00001	$1,0301537 \times 10^{-3}$	$9,381886 \times 10^{-4}$
$0,008c$	0,0001	$1,03 \times 10^{-2}$	$7,154 \times 10^{-2}$

Cuadro 4.1: Valores de ϵ y ϵ' para el sistema T-L

En base a este resultado, se consideró que el valor adecuado para Δt es 0.00001 para el sistema Tierra-Luna. Los valores de ϵ son suficientemente pequeños para no comprometer la coherencia del algoritmo y se cumple la condición 1: $\Delta t \geq \delta t$.

Es importante tener en cuenta que, en caso de que la condición 1 no se cumpla para velocidades de propagación más bajas, se debe incrementar el valor de Δt de tal manera que $\delta t \leq \Delta t$.

Los valores de ϵ no son idénticos para un mismo Δt , pero su variación con respecto a la velocidad de propagación es pequeña. La diferencia entre los valores de ϵ y ϵ' es del orden de 10^{-9} y 10^{-7} como se puede observar en la tabla. Los valores de ϵ son más sensibles al cambio de Δt .

4.1.2. Cálculo de Δt para el sistema Sol-Tierra-Luna

Valores máximos de ϵ y ϵ' para el sistema Solar:

VGP	Δt	ϵ_{max}	ϵ'_{max}
$1 \times 10^8 c$	0,00001	$5,638002 \times 10^{-3}$	$1,079734 \times 10^{-4}$
$5c$	0,00001	$5,562850 \times 10^{-3}$	$1,0796375 \times 10^{-4}$
$0,5c$	0,0001	$5,47386 \times 10^{-2}$	$1,0775 \times 10^{-3}$

Cuadro 4.2: Valores de ϵ y ϵ' para el sistema S-T-L

Se utiliza un valor de $\Delta t = 0,00001$ debido a que los valores de las funciones ϵ son lo suficientemente pequeños para garantizar una buena aproximación.

4.1.3. Cálculo de Δt para el sistema Solar (9 cuerpos)

Valores máximos de ϵ y ϵ' para el sistema Sol-Tierra-Luna para $\Delta t = 0,00001$

VGP	Δt	ϵ_{max}	ϵ'_{max}
$1 \times 10^8 c$	0,00001	$3,991934 \times 10^{-4}$	$3,34302445 \times 10^{-4}$
90c	0,00001	$3,991934084 \times 10^{-4}$	$3,343024476 \times 10^{-4}$
9c	0,0001	$3,9927958 \times 10^{-3}$	$3,34343607 \times 10^{-3}$
0,9c	0,001	$3,99359744 \times 10^{-2}$	$3,34264999 \times 10^{-2}$

Cuadro 4.3: Valores de ϵ y ϵ' para el sistema Solar

Se utilizó un valor de $\Delta t = 0,0001$ porque considero que los valores de ϵ son lo suficientemente pequeños para garantizar una buena aproximación.

4.2. Simulación el sistema Tierra-Luna

Se realiza una simulación de 1000 años con un paso temporal de 10^{-5} años, lo que conlleva a 10^8 pasos de integración. Masa de la Tierra $M_t = 1$, masa de la Luna $m_l = 0,123$. Datos:

- $nt = 1000$ años
- $\Delta t = 0,00001$ años
- Valores máximos para $\epsilon_{max} = 1,0301587 \times 10^{-3}$ y $\epsilon'_{max} = 9,381948 \times 10^{-4}$.
- Condiciones iniciales:

Planeta/Satélite	Posición inicial (UA)	Velocidad inicial (UA/año)
Tierra	$(-2,91 \times 10^{-6}; -3,26 \times 10^{-5}; -1,98 \times 10^{-6})$	$(2,49 \times 10^{-3}; -2,2 \times 10^{-6}; -1,46 \times 10^{-4})$
Luna	$(2,37 \times 10^{-4}; 2,65 \times 10^{-3}; 1,61 \times 10^{-4})$	$(-0,2; 1,796 \times 10^{-4}; 1,19 \times 10^{-2})$

Cuadro 4.4: Condiciones iniciales para el sistema Tierra-Luna

Con el objetivo de ilustrar adecuadamente la simulación, se ha dividido la representación gráfica de la órbita en dos partes: una que muestra los primeros 1.5 años de simulación y otra que representa los últimos 1.5 años.

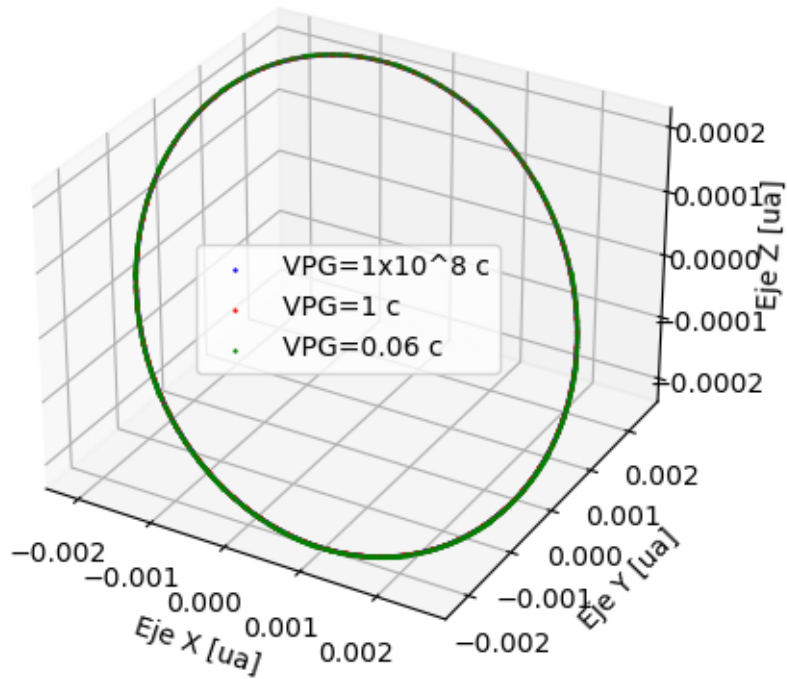


Figura 4.1: Órbita Lunar al rededor de la Tierra desde el año 0 hasta 1,5 años. Se puede observar que las orbitas están aparentemente superpuestas, que nos indica que su variación es muy pequeña.

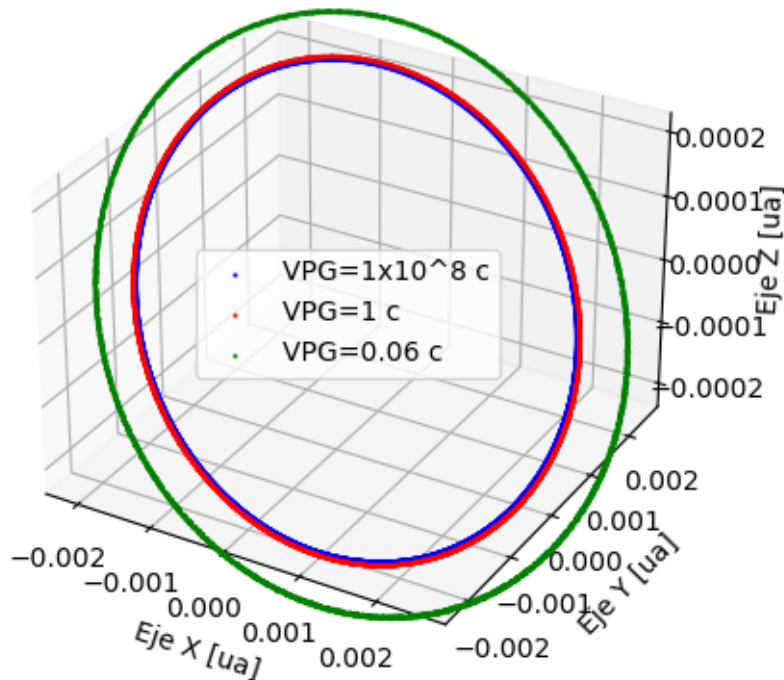


Figura 4.2: Órbita Lunar desde el año 9998.5 hasta el año 1000. Después de 1000 años, las órbitas se encuentran separadas, incluso la órbita con una velocidad de propagación 1c representada en color azul, aunque su variación sea apenas perceptible. La órbita con un radio más grande corresponde a la VPG más pequeña de 0,06c en color verde.

A continuación se grafica la distancia entre la Tierra y la Luna:

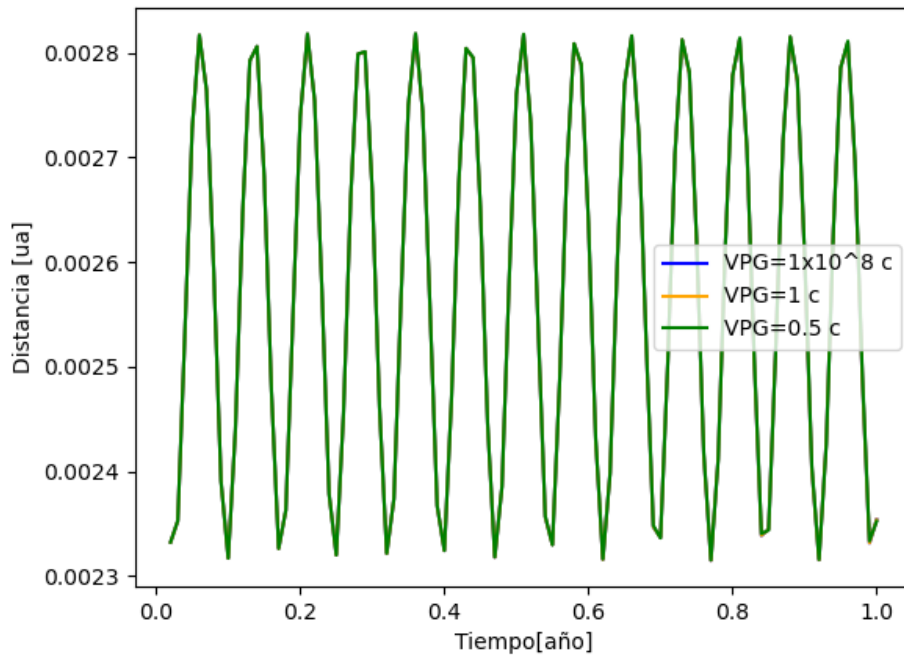


Figura 4.3: Distancia Tierra-Luna desde el año 0 hasta el año 1. De igual forma se observa que al inicio las órbitas están aparentemente superpuestas por lo cual no hay una diferencia apreciable en la distancia. Como dato los picos representan el perihelio y el afelio ya que es una órbita elíptica.

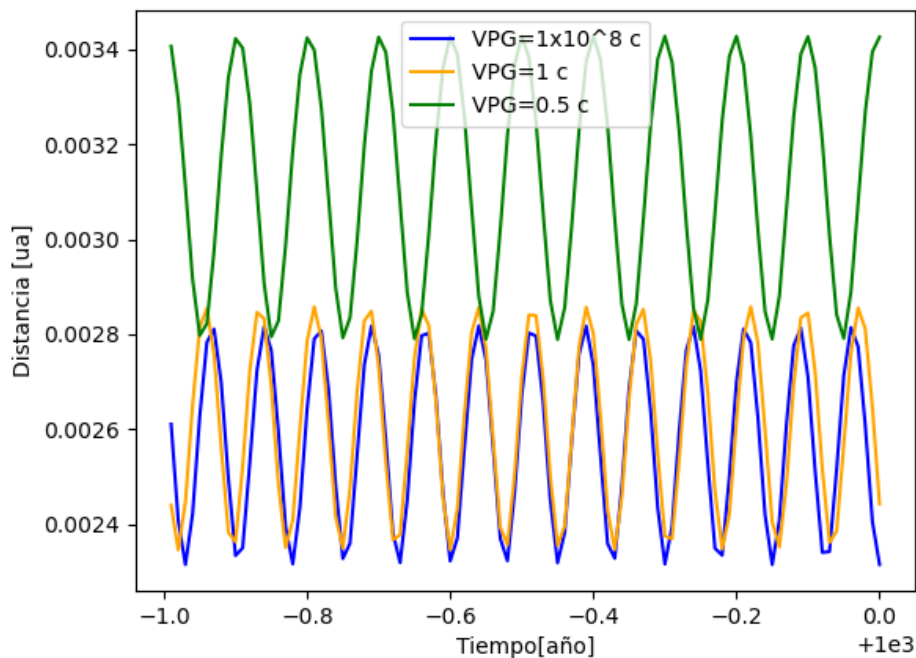


Figura 4.4: Distancia Tierra-Luna desde el año 9999 hasta el año 1000. Luego de 1000 años se observa una variación de la distancia de aproximadamente un $0,0006[ua]$ para $VPG = 0,5c$ en relación a $VPG = 10^8c$. Este comportamiento se observa de manera consistente a lo largo de toda la simulación, donde la Tierra y la Luna se alejan más a lo largo del tiempo.

Durante un período inicial, todas las órbitas para diferentes velocidades de propagación aparentemente se superponen indicando que su variación es mínima. Sin embargo, a lo largo de 1000 años, se puede notar que las órbitas con una velocidad de propagación de $VPG = 0,06c$ están más alejadas de la Tierra. Como era de esperar, para velocidades de propagación altas el sistema tiende a seguir una elipse propia del sistema con velocidad de propagación infinita. Además, se puede observar que el efecto de tener una velocidad finita en el sistema Tierra-Luna es que la distancia de la órbita aumenta con el tiempo.

Velocidades de propagación inferiores a $0,06c$, con un $\Delta t = 0,00001$ no generan resultados válidos en las simulaciones del sistema Tierra-Luna, ya que no cumplen la condición $\delta t \leq \Delta t$.

4.3. Simulación el sistema de 3 cuerpos Sol-Tierra-Luna

Se realiza una simulación de 1000 años con un paso temporal de 10^{-5} años, lo que conlleva a 10^8 pasos de integración. Masa del Sol $M_t = 333000$, masa de la Tierra $M_t = 1$, masa de la Luna $m_l = 0,123$. Datos:

- $nt = 1000$ años
- $\Delta t = 0,00001$ años
- Valores máximos para $\epsilon_{\max} = 5,638002 \times 10^{-3}$ y $\epsilon'_{\max} = 1,079734 \times 10^{-4}$.
- Condiciones iniciales:

Planeta/Satélite	Posición inicial (ua)	Velocidad inicial (ua/año)
Sol	$(-1,4 \times 10^{-6}; -2,6 \times 10^{-6}; 5,2 \times 10^{-12})$	$(1,7 \times 10^{-5}; -9,1 \times 10^{-6}; 3,4 \times 10^{-11})$
Tierra	$(0,482; 0,862; 5,8 \times 10^{-7})$	$(-5,61; 2,9; -0,00017)$
Luna	$(0,483; 0,864; -0,00019)$	$(-5,82; 3,07; 0,013)$

Cuadro 4.5: Condiciones iniciales sistema Sol-Tierra-Luna

Los valores de épsilon aseguran que los términos sean al menos 1500 veces más pequeños en la primera aproximación con ϵ y 3000 veces más pequeños en el caso de ϵ' aproximadamente.

Con el objetivo de ilustrar nuevamente de forma adecuada la simulación, se ha dividido la representación gráfica de la órbita en dos partes: una que muestra los primeros 5 años de simulación y otra que representa los últimos 5 años.

En primer lugar se grafica la órbita de la Luna al rededor de la Tierra:

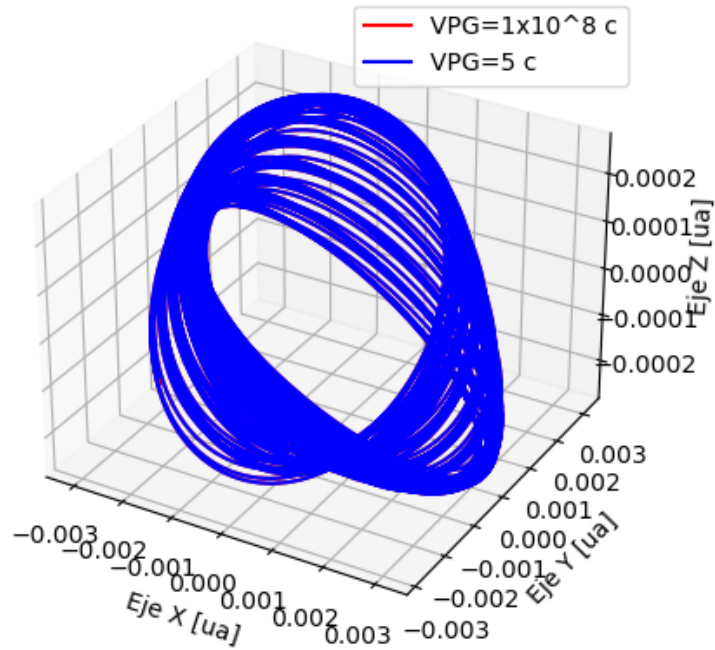


Figura 4.5: Órbita Tierra-Luna desde año 0 hasta el año 5. Las órbitas de color rojo no se observan porque las azules están superpuestas, lo que indica que su variación es muy pequeña.

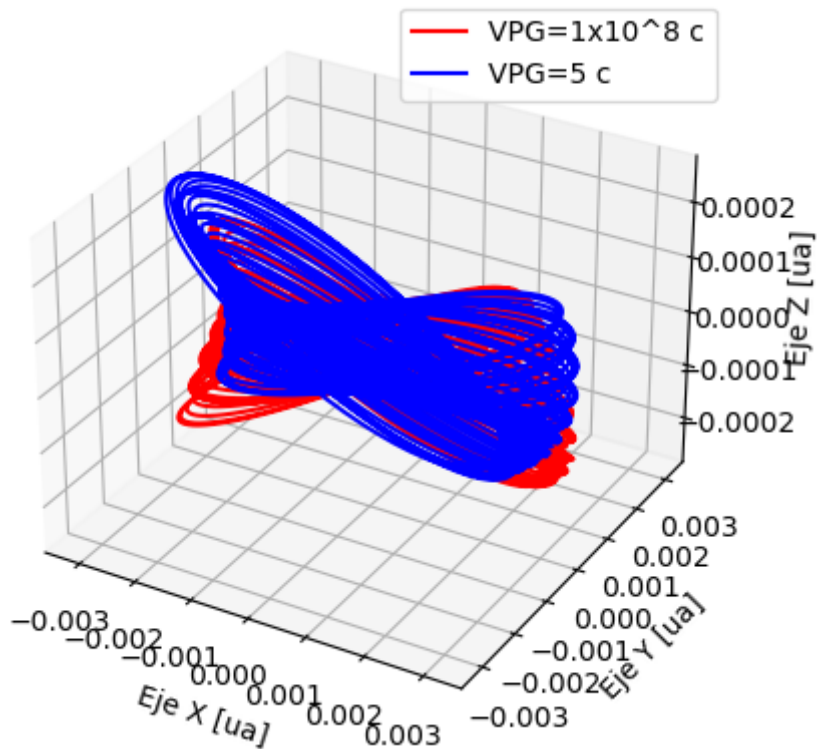


Figura 4.6: Órbita Tierra-Luna desde año 9995 hasta el año 1000. Se puede observar la diferencia entre las órbitas para diferentes VPG, ya que las órbitas no están superpuestas.

A continuación la órbita de la Tierra al rededor del Sol

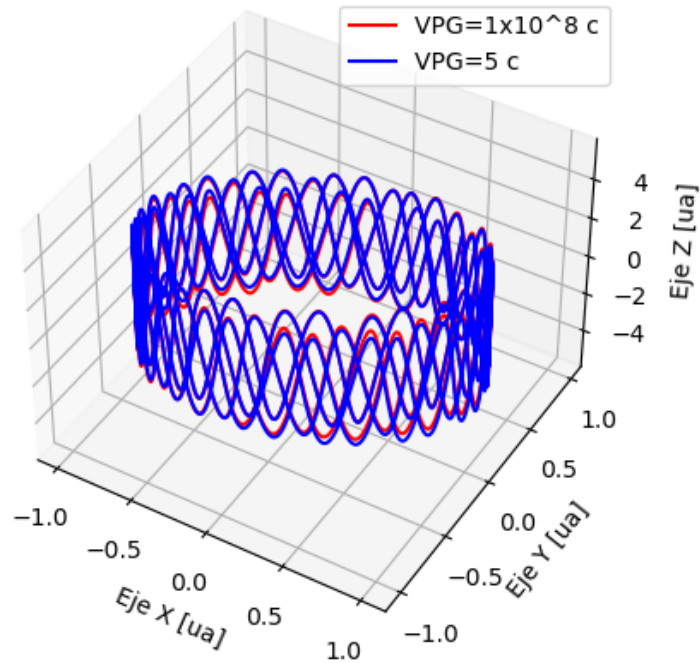


Figura 4.7: Órbita Sol-Tierra desde el año 0 hasta el año 5. Se utiliza una escala de $10^7 : 1$ del eje z, para una mejor visualización de las variaciones de la órbita.

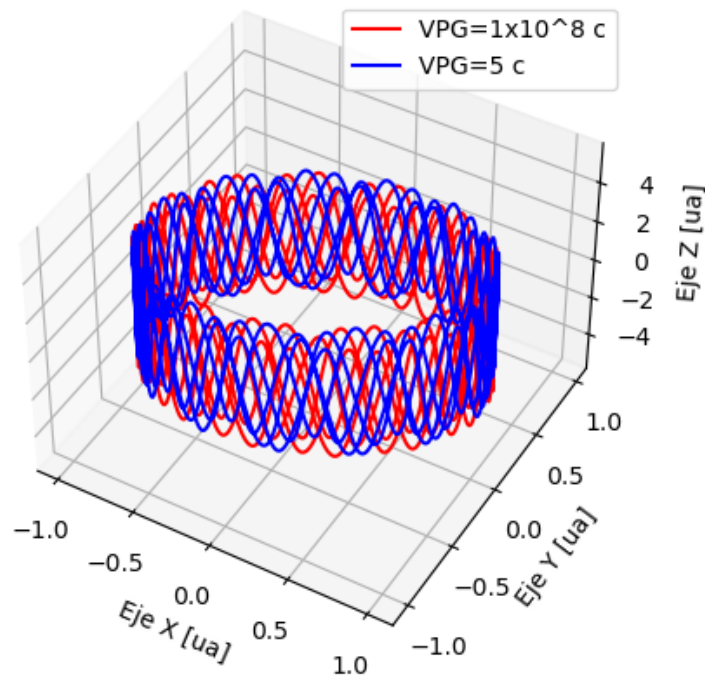


Figura 4.8: Órbita Sol-Tierra desde el año 9995 hasta el año 1000. Se debe multiplicar el valor del eje z por 10^{-7} para obtener el valor real. Luego de 1000 años se puede observar una notable diferencia entre las órbitas para diferentes velocidades de propagación.

A continuación las distancias entre el Sol y la Tierra.

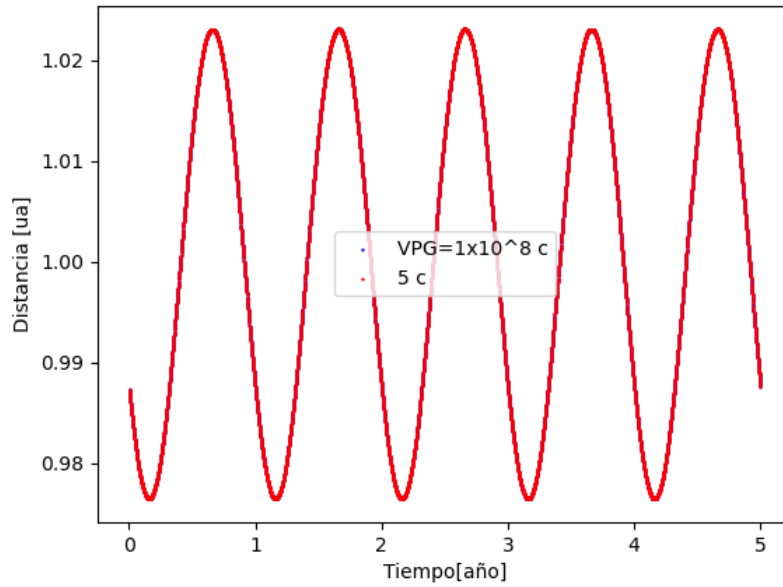


Figura 4.9: Distancia Sol-Tierra desde año 0 hasta el año 5. La diferencia entre estas dos distancias es inapreciable. Sin embargo, como se muestra a continuación, es del orden de $10^{-5}[ua]$.

Para apreciar la variación entre las distancia se restan los valores de la gráfica anterior distancia entre el Sol y la Tierra. D1 corresponde al valor con $VPG = 10^8c$, D2 valor de la distancia con $VPG = 5c$.

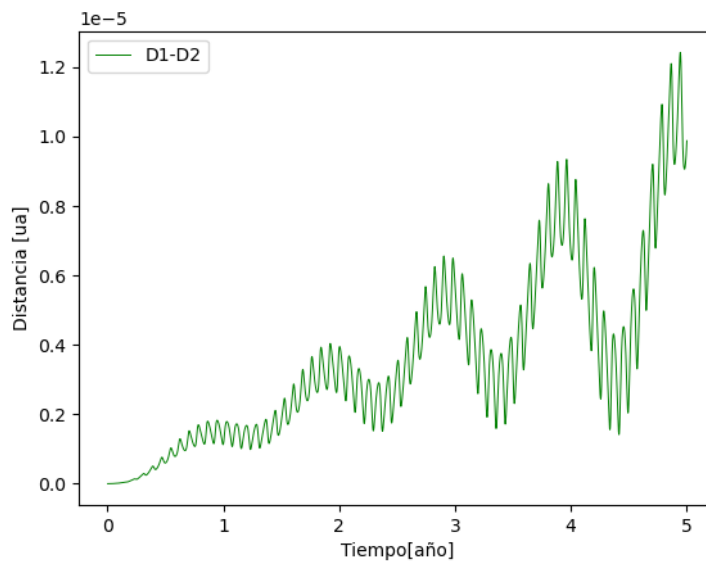


Figura 4.10: Diferencia entre las distancias de las orbitas Sol-Tierra para diferentes VPG, desde año 0 hasta el año 5. D1 corresponde a la órbita con $VPG = 5c$, D2 corresponde a la órbita con $VPG = 10^8c$. Para poder observar la variación, se ha graficado la diferencia entre las distancias durante los 5 primeros años. Dicha diferencia es del orden de $10^{-5}[ua]$.

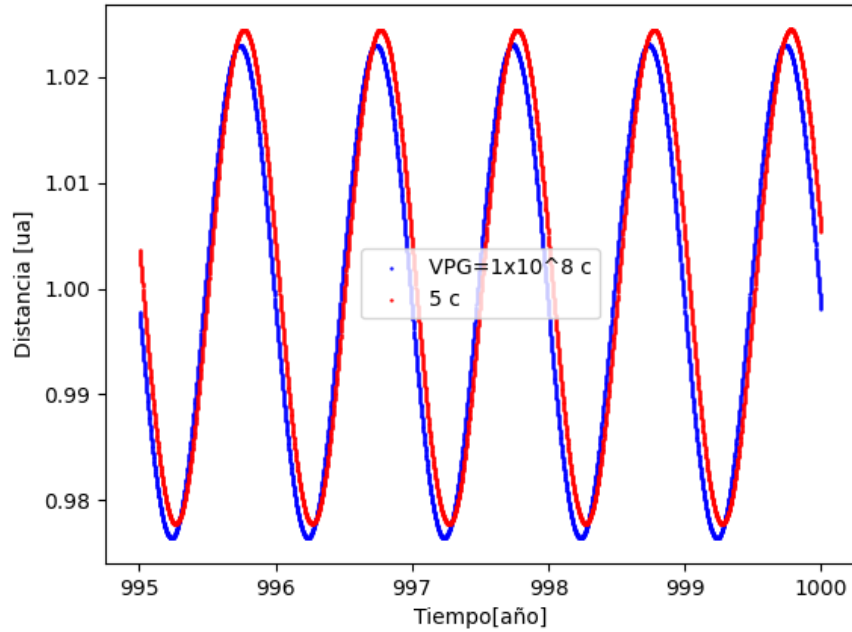


Figura 4.11: Distancia Sol-Tierra desde el año 1995 hasta el año 2000. Al final de la simulación (1000 años) se observa un aumento del radio orbital del orden de $10^{-3}[ua]$

Se observa que en la gráfica inicial de 5 años, todas las órbitas para diferentes velocidades de propagación aparentemente se superponen, lo cual se puede observar como un único color rojo en la gráfica (4.9). Sin embargo, es importante tener en cuenta que se espera una variación minúscula que no es apreciable al graficar las órbitas. No obstante, podemos observar esta diferencia en la gráfica (4.10), la cual muestra que el radio orbital de la Tierra aumenta a lo largo de los primeros 5 años en un orden de $10^{-5}[ua]$.

La gráfica (4.11) nos indica que luego de 1000 años existe una diferencia aproximadamente de un $0,03[ua]$ en el aumento del radio orbital para una $VPG = 5c$.

Además, es importante destacar que velocidades de propagación inferiores a $5c$ con un $\Delta t = 0,00001$ no generan resultados válidos en las simulaciones del sistema Sol-Tierra-Luna.

4.4. Simulación del sistema solar (9 cuerpos)

Se realiza una simulación de 1000 años con un paso temporal de 10^{-4} años.

- $nt = 1000$ años
- $\Delta t = 0,0001$ años

- Valores máximos para $\epsilon_{\max} = 3,992795857 \times 10^{-3}$ y $\epsilon'_{\max} = 3,34343607 \times 10^{-3}$.
- Condiciones iniciales:

Planeta/Satélite	Posición inicial (ua)	Velocidad inicial (ua/año)
Sol	$(-9,04 \times 10^{-3}, 7,2 \times 10^{-6}, 2,1 \times 10^{-4})$	$(3,8 \times 10^{-4}, -3,3 \times 10^{-3}, 1,8 \times 10^{-5})$
Mercurio	$(-0,021, 0,308, 0,026)$	$(-12,3, -0,61, 1,08)$
Venus	$(0,613, -0,391, -0,041)$	$(3,84, 6,25, -0,14)$
Tierra	$(-0,247, 0,952, -0,00015)$	$(-6,16, -1,65, -0,00014)$
Luna	$(-0,004, 1,57, 0,033)$	$(-4,92, 0,40, 0,13)$
Marte	$(4,82, 1,10, -0,11)$	$(-0,65, 2,82, 0,003)$
Júpiter	$(8,19, -5,44, -0,23)$	$(1,01, 1,70, -0,07)$
Saturno	$(13,28, 14,51, -0,12)$	$(-1,07, 0,91, 0,02)$
Urano	$(29,78, -2,76, -0,63)$	$(0,09, 1,16, -0,03)$

Cuadro 4.6: Condiciones iniciales Sistema Solar

Mercurio

En primer lugar órbita de Mercurio al final de la simulación (10 años finales):

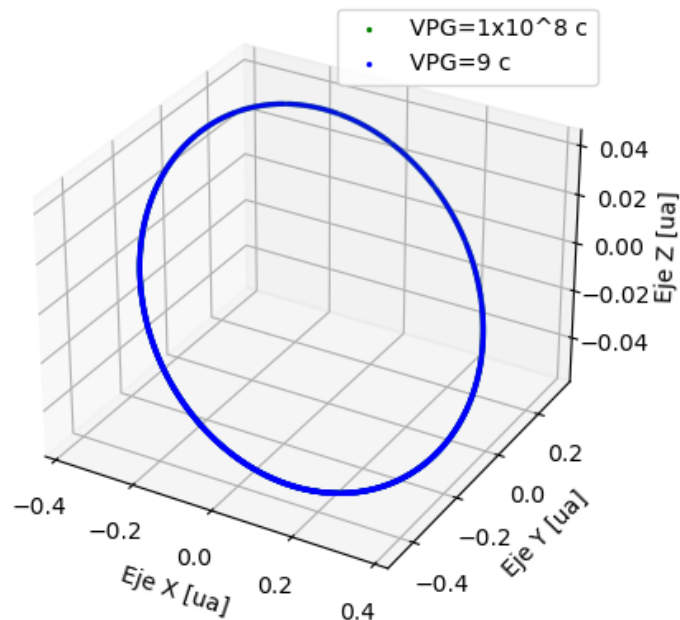


Figura 4.12: Órbita Mercurio-Sol desde el año 990 hasta el año 1000. Órbitas superpuestas indican que el cambio es minúsculo.

A continuación la grafica de la distancia entre Sol y Mercurio a lo largo de los últimos 10 años:

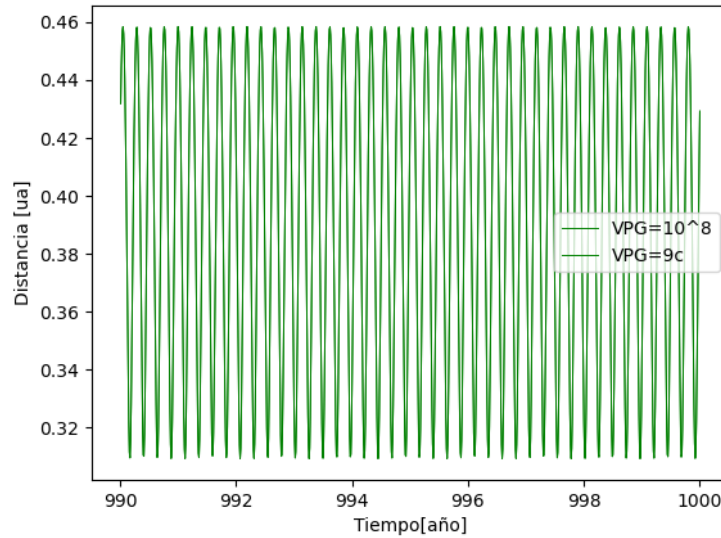


Figura 4.13: Distancia Mercurio-Sol desde el año 990 hasta el año 1000. Se puede apreciar aproximadamente las 40 vueltas al rededor del Sol.

Para apreciar la pequeña variación entre las distancia se restan los valores de la gráfica anterior distancia entre el Sol y Mercurio. D1 corresponde al valor con $VPG = 10^8 c$, D2 valor de la distancia con $VPG = 9c$.

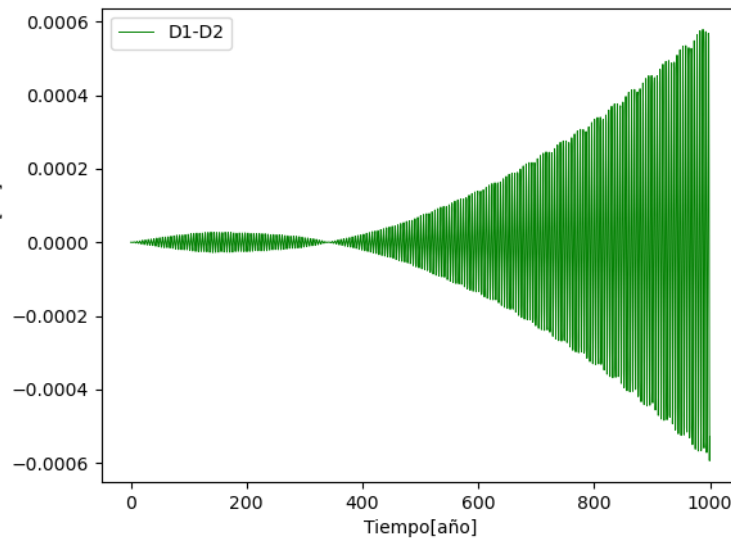


Figura 4.14: Diferencia entre las distancias de las orbitas Sol-Mercurio para diferentes VPG. D1 corresponde a la orbita con $VPG = 9c$, D2 corresponde a la órbita con $VPG = 10^8 c$. Se puede apreciar que dicha diferencia esta en el orden de $6 \times 10^{-4} [ua]$.

Jupiter

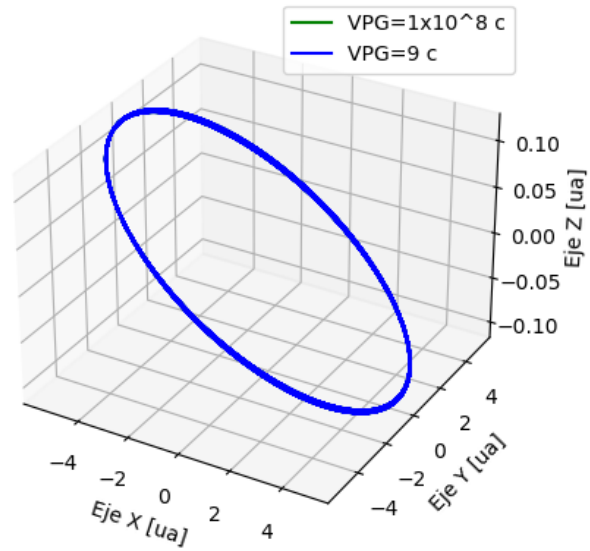


Figura 4.15: Orbits of Jupiter from year 0 to 1000. Approximately 80 revolutions around the sun. No significant change is observed for $VPG = 9c$.

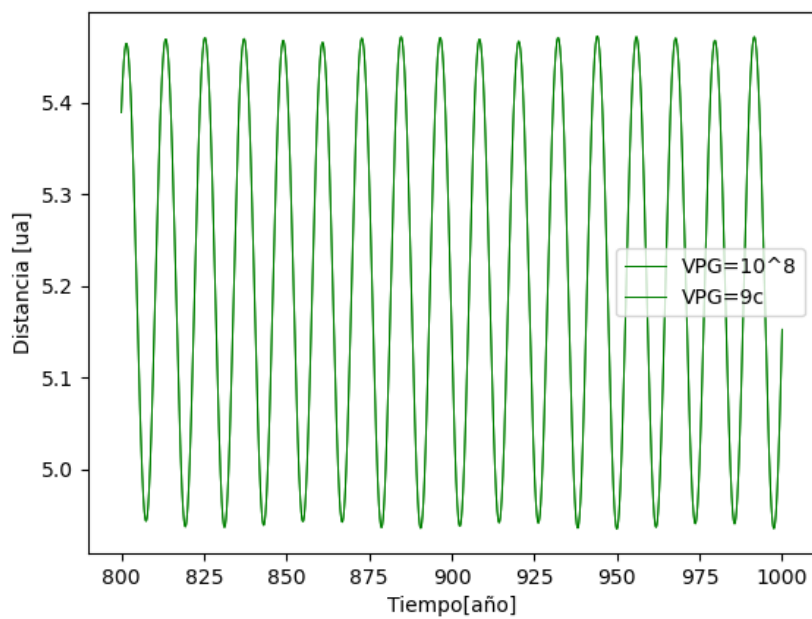


Figura 4.16: Distance Jupiter-Sun from year 800 to 1000. As we can observe, the variations are imperceptible in relation to the orbital radius, but this does not mean there is no variation.

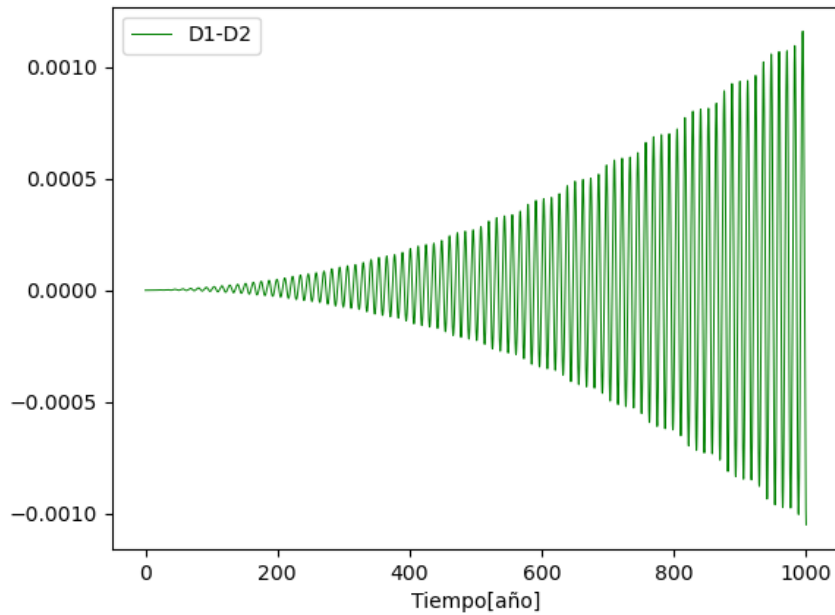


Figura 4.17: Diferencia entre las distancias de las orbitas Sol-Jupiter para diferentes VPG. Se puede apreciar que la diferencia esta en el orden de $10^{-3}[ua]$. Aproximadamente un orden de magnitud superior al caso de Mercurio.

Neptuno

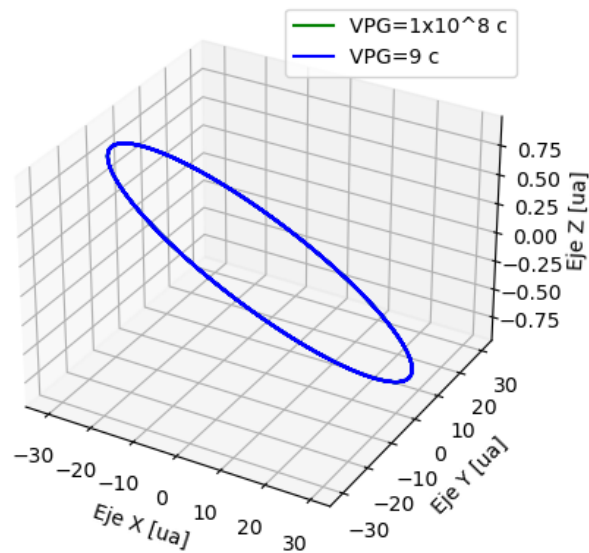


Figura 4.18: Orbita de Neptuno desde el año 0 hasta el año 1000. En el caso de Neptuno se ha graficado las orbitas completas durante el periodo de 1000 años y no se encuentra diferencia a simple vista.

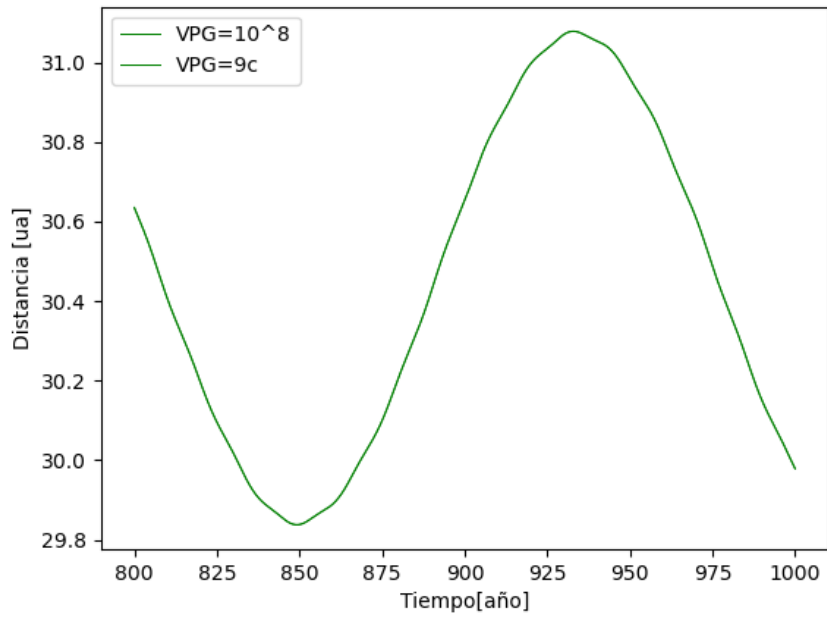


Figura 4.19: Distancia Neptuno-Sol desde el año 800 hasta el año 1000. La falta de una variación apreciable se debe al período relativamente corto de simulación en este caso, que abarca únicamente 1000 años.

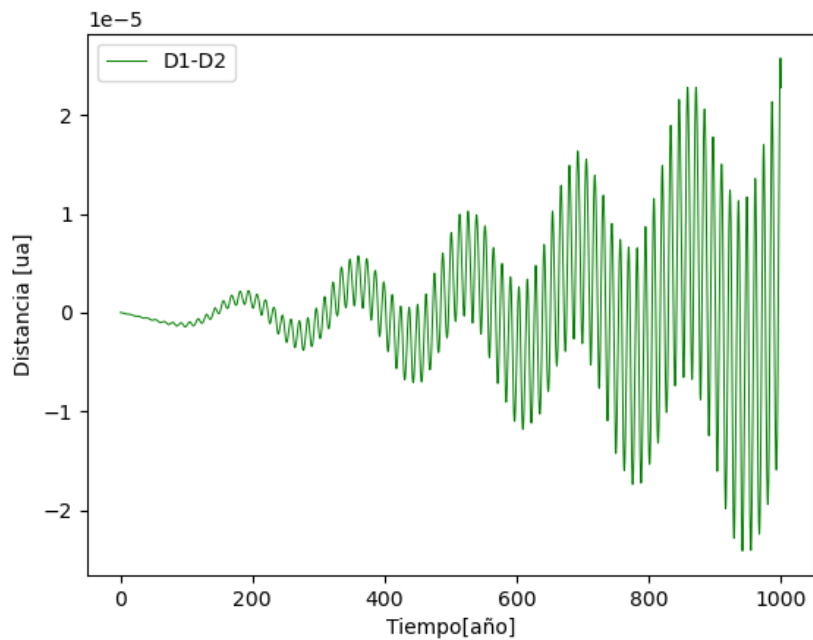


Figura 4.20: Diferencia entre las distancias de las órbitas Sol-Neptuno para diferentes VPG. Se puede apreciar que la diferencia está en el orden de $2 \times 10^{-5} [ua]$ dos órdenes de magnitud más pequeño que el caso de Júpiter.

En primer lugar, debemos notar que solo se han representado tres planetas; no obstante, el comportamiento de las órbitas y las distancias sigue el mismo patrón que los tres ejemplos. Es decir, presentan una variación minúscula, pero sus órbitas se alejan paulatinamente. El cambio en las órbitas de todos los planetas del sistema solar durante un período de 1000 años, con una velocidad de propagación de 9 veces la velocidad de la luz ($VPG = 9c$) y un $\Delta t = 0,00001$, induce una variación en la distancia orbital. Esta variación se muestra en las gráficas de Mercurio (4.15), Júpiter (4.18) y Neptuno (4.21), y tiene magnitudes aproximadas de 6×10^{-4} ua, 10^{-3} ua y 2×10^{-5} ua, respectivamente, pero no podemos confirmar efectivamente que el radio orbital siempre aumente; para ello, se requieren simulaciones más extensas.

En el caso del sistema solar y un $\Delta t = 0,0001$, no se obtienen resultados válidos para velocidades de propagación por debajo de $9c$.

Capítulo 5

Conclusiones

En resumen, hemos logrado el objetivo de desarrollar un programa que permite simular velocidades de propagación finita suficientemente altas. Sin embargo, es importante tener en cuenta que existen condiciones estrictas que deben cumplirse para obtener resultados correctos. Estas condiciones son: $\delta t \leq \Delta t$ y $\epsilon, \epsilon' \leq \epsilon_{max}$. Si estas condiciones no se cumplen, los resultados serán incorrectos.

Los resultados muestran que en el sistema Tierra-Luna, la distancia promedio entre la Luna y la Tierra aumenta de manera proporcional al tiempo a lo largo de un período de 1000 años. En el caso del sistema Sol-Tierra-Luna, se observa un aumento aproximado de 0,03 [ua] en el radio orbital de la órbita Sol-Tierra a lo largo de 1000 años. Es importante notar que, aunque las órbitas son teóricamente elípticas, en la práctica son casi circulares y es por eso a que hablo de un radio orbital. En la órbita Sol-Tierra, con velocidad ($VPG = 5c$), se aleja mucho más rápido que para ($VPG = 10^8 c$).

Para órbitas más amplias como el en el caso del sistema solar, se necesitan simulaciones más a largo plazo para apreciar cambios significativos, pero en principio las orbitas se alejan unas de otras, sin embargo para confirmar si el radio orbital aumenta se necesitan simulaciones más largas.

En resumen, a medida que el tiempo avanza, las órbitas se vuelven más grandes, y esta variación es más pronunciada para velocidades de propagación más pequeñas.

En cuanto al nuevo algoritmo, se ha demostrado no es simpléctico. Pero podemos afirmar que tiene el mismo orden de error que el de Velocidades de Verlet, ya que la condición $\delta t \leq \Delta t$ limita el error en las aproximaciones, asegurando que ambos algoritmos mantengan el mismo orden de error global.

Como recomendaciones, es necesario integrar la energía y el momento en el algoritmo y realizar un estudio teórico para confirmar los resultados. Además, se deben incluir pequeñas correcciones en el algoritmo para evitar el cálculo de epsilon cuando la velocidad de las partículas es igual a cero, ya que esto resultaría en valores infinitos, y considerar la posibilidad de un paso de integración adaptable.

Capítulo 6

Anexo

6.1. Anexo 1: Demostración de la simplecticidad del algoritmo de Velocidades de Verlet.

Teorema de Liouville: Sea la transformación $(\mathbf{q}, \mathbf{p}, t) \rightarrow (\mathbf{Q}, \mathbf{P}, t)$

El teorema de Liouville establece que el volumen en el espacio de fases es invariante[8].

$$\int \dots \int dq_1 \dots dq_s dp_1 \dots dp_s = \int \dots \int dQ_1 \dots dQ_s dP_1 \dots dP_s \quad (6.1)$$

Mediante el uso del Jacobiano J de la transformación, podemos escribir la ecuación anterior como sigue:

$$\int \dots \int dQ_1 \dots dQ_s dP_1 \dots dP_s = \int \dots \int \text{Det } J dq_1 \dots dq_s dp_1 \dots dp_s \quad (6.2)$$

Donde

$$J = \frac{\partial (Q_1, \dots, Q_s, P_1, \dots, P_s)}{\partial (q_1, \dots, q_s, p_1, \dots, p_s)} \quad (6.3)$$

Es el Jacobiano de la transformación. La prueba de que dicha transformación es canónica se reduce a demostrar que [8]:

$$\text{Det } J = 1 \quad (6.4)$$

Se escribe las ecuaciones del algoritmo de velocidades de Verlet de la siguiente forma por conveniencia:

$$\begin{aligned}
v_{n+1/2} &= v_n + \frac{\Delta t}{2} a(x_n) \\
x_{n+1} &= x_n + \Delta t \cdot v_{n+1/2} \\
v_{n+1} &= v_{n+1/2} + \frac{\Delta t}{2} a(x_{n+1})
\end{aligned} \tag{6.5}$$

Se observa que esta transformación es para una sola de las coordenadas, la cual es igual en las 3 coordenadas.

Partimos de sus ecuaciones en (2.7) y las evaluamos en dos puntos: (x_0, v_0) y $(x_0 + \delta x_0, v_0 + \delta v_0)$

Ecuación 1:

$$v_{n+1/2} = v_n + \frac{\Delta t}{2} a(x_n) \tag{6.6}$$

Función (2.7) evaluada en el punto (x_0, v_0) :

$$v_{1/2} = v_0 + \frac{\Delta t}{2} a(x_0) \tag{6.7}$$

Función (2.7) evaluada en el punto $(x_0 + \delta x_0, v_0 + \delta v_0)$

$$v_{1/2} + \delta v_{1/2} = v_0 + \delta v_0 + \frac{\Delta t}{2} a(x_0 + \delta x_0) \tag{6.8}$$

Restamos (2.8) - (2.9):

$$-\delta v_{1/2} = -\delta v_0 + \frac{\Delta t}{2} [a(x_0) - a(x_0 + \delta x_0)]$$

$$\delta v_{1/2} = \delta v_0 + \frac{\Delta t}{2} [a(x_0 + \delta x_0) - a(x_0)] \frac{\delta x_0}{\delta x_0}$$

El segundo término de la parte derecha puede ser aproximado como una derivada y se obtiene:

$$\delta v_{1/2} = \delta v_0 + \frac{\Delta t}{2} \frac{\delta a(x_0)}{\delta x} \delta x_0$$

Podemos escribir de forma matricial:

$$\begin{pmatrix} \delta x_0 \\ \delta v_{1/2} \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ \frac{\Delta t}{2} \frac{\delta a(x_0)}{\delta x} & 1 \end{pmatrix} \begin{pmatrix} \delta x_0 \\ \delta v_0 \end{pmatrix} \tag{6.9}$$

Ecuación 2:

$$x_{n+1} = x_n + \Delta t \cdot v_{n+1/2} \quad (6.10)$$

Función (2.11) evaluada en el punto (x_0, v_0) :

$$x_1 = x_0 + \Delta t \cdot v_{1/2} \quad (6.11)$$

Función (2.11) evaluada en el punto $(x_0 + \delta x_0, v_0 + \delta v_0)$

$$x_1 + \delta x_1 = x_0 + \delta x_0 + \Delta t \cdot v_{1/2} + \Delta t \cdot \delta v_{1/2} \quad (6.12)$$

Restamos (2.13) - (2.12):

$$\delta x_1 = \delta x_0 + \Delta t \cdot \delta v_{1/2}$$

Podemos escribir de forma matricial:

$$\begin{pmatrix} \delta x_1 \\ \delta v_{1/2} \end{pmatrix} = \begin{pmatrix} 1 & \Delta t \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \delta x_0 \\ \delta v_{1/2} \end{pmatrix} \quad (6.13)$$

Ecuación 3:

$$v_{n+1} = v_{n+1/2} + \frac{\Delta t}{2} a(x_{n+1}) \quad (6.14)$$

Función (2.15) evaluada en el punto (x_0, v_0) :

$$v_1 = v_{1/2} + \frac{\Delta t}{2} a(x_1) \quad (6.15)$$

Función (2.15) evaluada en el punto $(x_0 + \delta x_0, v_0 + \delta v_0)$

$$v_1 + \delta v_1 = v_{1/2} + \delta v_{1/2} + \frac{\Delta t}{2} a(x_1 + \delta x_1) \quad (6.16)$$

Restamos (2.17) - (2.18):

$$\delta v_1 = \delta v_{1/2} + \frac{\Delta t}{2} \frac{\delta a(x_1)}{\delta x} \delta x_1$$

Podemos escribir de forma matricial:

$$\begin{pmatrix} \delta x_1 \\ \delta v_1 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ \frac{\Delta t}{2} \frac{\delta a(x_0)}{\delta x} & 1 \end{pmatrix} \begin{pmatrix} \delta x_1 \\ \delta v_{1/2} \end{pmatrix} \quad (6.17)$$

Remplazamos 2.8 en 2.9 y en 2.10, obtenemos:

$$\begin{pmatrix} \delta x_1 \\ \delta v_1 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ \frac{\Delta t}{2} \frac{\delta a(x_1)}{\delta x} & 1 \end{pmatrix} \begin{pmatrix} 1 & \Delta t \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ \frac{\Delta t}{2} \frac{\delta a(x_0)}{\delta x} & 1 \end{pmatrix} \begin{pmatrix} \delta x_0 \\ \delta v_0 \end{pmatrix} \quad (6.18)$$

$$\begin{pmatrix} \delta x_1 \\ \delta v_1 \end{pmatrix} = A \cdot B \cdot C \begin{pmatrix} \delta x_0 \\ \delta v_0 \end{pmatrix} \quad (6.19)$$

Al calcular su determinante :

$$\det[J_x] = \det[A] \cdot \det[B] \cdot \det[C] \quad (6.20)$$

$$\det[J_x] = 1 \times 1 \times 1 \quad (6.21)$$

$$\det[J_x] = 1 \quad (6.22)$$

Para demostrar el Teorema de Liouville y establecer las ecuaciones (2.3) y (2.5), es necesario considerar todas las coordenadas:

Se procede a reescribir la ecuación (2.19) mediante la multiplicación de sus matrices de la siguiente manera:

$$\begin{pmatrix} \delta x_1 \\ \delta v_1 \end{pmatrix} = \begin{pmatrix} 1 & \Delta t \\ \frac{\Delta t}{2} \frac{\delta a(x_1)}{\delta x} & 1 + \frac{\Delta t^2}{2} \frac{\delta a(x_1)}{\delta x} \end{pmatrix} \begin{pmatrix} 1 & 0 \\ \frac{\Delta t}{2} \frac{\delta a(x_0)}{\delta x} & 1 \end{pmatrix} \begin{pmatrix} \delta x_0 \\ \delta v_0 \end{pmatrix} \quad (6.23)$$

$$\begin{pmatrix} \delta x_1 \\ \delta v_1 \end{pmatrix} = \begin{pmatrix} 1 + \frac{\Delta t^2}{2} \frac{\delta a(x_0)}{\delta x} & \Delta t \\ \frac{\Delta t}{2} \frac{\delta a(x_1)}{\delta x} + \left(\frac{\Delta t}{2} \frac{\delta a(x_0)}{\delta x} \right) \left(1 + \frac{\Delta t^2}{2} \frac{\delta a(x_1)}{\delta x} \right) & 1 + \frac{\Delta t^2}{2} \frac{\delta a(x_1)}{\delta x} \end{pmatrix} \begin{pmatrix} \delta x_0 \\ \delta v_0 \end{pmatrix} \quad (6.24)$$

$$\begin{pmatrix} \delta x_1 \\ \delta v_1 \end{pmatrix} = \begin{pmatrix} Ax & Bx \\ Cx & Dx \end{pmatrix} \begin{pmatrix} \delta x_0 \\ \delta v_0 \end{pmatrix} \quad (6.25)$$

Donde:

$$Ax \times Dx - Bx \times Cx = 1 \quad (6.26)$$

Recordemos que la transformación es igual en las tres coordenadas, por lo tanto, puede escribirse de la siguiente manera:

$$\begin{pmatrix} \delta x_1 \\ \delta v_{x1} \\ \delta y_1 \\ \delta v_{y1} \\ \delta z_1 \\ \delta v_{z1} \end{pmatrix} = \begin{pmatrix} Ax & Bx & 0 & 0 & 0 & 0 \\ Cx & Dx & 0 & 0 & 0 & 0 \\ 0 & 0 & Ay & By & 0 & 0 \\ 0 & 0 & Cy & Dy & 0 & 0 \\ 0 & 0 & 0 & 0 & Az & Bz \\ 0 & 0 & 0 & 0 & Cz & Dz \end{pmatrix} \begin{pmatrix} \delta x_0 \\ \delta v_{x0} \\ \delta y_0 \\ \delta v_{y0} \\ \delta z_0 \\ \delta v_{z0} \end{pmatrix} \quad (6.27)$$

Se puede demostrar que el determinante de esta matriz es igual a la multiplicación del determinante de las matrices diagonales de tamaño 2×2 , ya que los demás términos de la matriz son cero. Obteniendo:

$$DetJ = [Ax \times Dx - Bx \times Cx][Ay \times Dy - By \times Cy][Az \times Dz - Bz \times Cz] = 1 \times 1 \times 1 \quad (6.28)$$

$$DetJ = 1 \quad (6.29)$$

En conclusión, la afirmación de que el algoritmo de velocidades de Verlet, cumple con la conservación del área en el espacio de fases es correcta. Esto significa que el algoritmo tiene una estabilidad global y sus coordenadas y energía no crecerán sin límites, lo que lleva a una conservación de la energía más controlada que algoritmos no simplécticos.

6.2. Anexo 2: Límite de VPG debido al determinante de la ecuación de δt

Analicemos el determinante de la ecuación (3.9):

$$[2 [\vec{r}(t) \cdot \vec{v}_A(t)]]^2 - 4 \cdot [|\vec{v}_A|^2 - VGP^2] \cdot |\vec{r}(t)|^2 \geq 0 \quad (6.30)$$

$$[\vec{r}'(t) \cdot \vec{v}_A(t)]^2 \geq [|\vec{v}_A(t)|^2 - VGP^2] \cdot |\vec{r}'(t)|^2 \quad (6.31)$$

$$\frac{[\vec{r}'(t) \cdot \vec{v}_A(t)]^2}{|\vec{r}'(t)|^2} - |\vec{v}_A(t)|^2 \geq -VGP^2 \quad (6.32)$$

$$|\vec{v}_A(t)|^2 - \frac{[\vec{r}'(t) \cdot \vec{v}_A(t)]^2}{|\vec{r}'(t)|^2} \leq VGP^2 \quad (6.33)$$

Para comprender mejor esta restricción, supongamos que la velocidad y la posición son perpendiculares lo que conduce a la siguiente expresión:

$$|\vec{v}_A(t)|^2 \leq VGP^2 \quad (6.34)$$

Lo que significa que existe una velocidad mínima que podemos simular y es igual a la velocidad a la que se mueve el cuerpo A.

Ahora supongamos que la velocidad y la posición son paralelas lo que conduce a la siguiente expresión:

$$0 \leq VGP^2 \quad (6.35)$$

Lo que significa que basta que la velocidad de propagación sea diferente de cero para obtener valores δt reales.

6.3. Anexo 3: Cálculo de δt con un mayor orden de precisión.

Deducción de δt usando una aproximación en series de Taylor de tercer orden.

$$\vec{r}_2(t - \delta t) = \vec{r}_2(t) + \vec{v}_2(t)(-\delta t) + \frac{1}{2}\vec{a}_2(t)(-\delta t)^2 \quad (6.36)$$

Remplazamos en la ecuación 3.3 y elevamos al cuadrado.

$$\begin{aligned}
& \left[\vec{r}_1(t) - \vec{r}_2(t) - \vec{v}_2(t)(-\delta t) - \frac{1}{2}\vec{a}_2(t)(-\delta t)^2 \right] \\
& \left[\vec{r}_1(t) - \vec{r}_2(t) - \vec{v}_2(t)(-\delta t) - \frac{1}{2}\vec{a}_2(t)(-\delta t)^2 \right] \\
& = VGP^2\delta t^2
\end{aligned} \tag{6.37}$$

Luego de multiplicar y reagrupar tenemos:

$$\begin{aligned}
& |\vec{r}_1(t) - \vec{r}_2(t)|^2 - 2(\vec{r}_1(t) - \vec{r}_2(t)) \cdot \vec{v}_2(t)(-\delta t) \\
& + \left[|\vec{v}_2(t)|^2 - (\vec{r}_1(t) - \vec{r}_2(t)) \cdot \vec{a}_2(t) \right] (-\delta t)^2 \\
& + \vec{v}_2(t) \cdot \vec{a}_2(t)(-\delta t)^3 + \frac{1}{4} |\vec{a}_2(t)|^2 (-\delta t)^4 \\
& = VGP^2\delta t^2
\end{aligned} \tag{6.38}$$

Donde podemos despejar δt :

$$\begin{aligned}
A &= \frac{1}{4} |\vec{a}_2(t)|^2 \\
B &= \vec{v}_2(t) \cdot \vec{a}_2(t) \\
C &= \left[|\vec{v}_2(t)|^2 - (\vec{r}_1(t) - \vec{r}_2(t)) \cdot \vec{a}_2(t) \right] \\
D &= -2(\vec{r}_1(t) - \vec{r}_2(t)) \cdot \vec{v}_2(t) \\
E &= |\vec{r}_1(t) - \vec{r}_2(t)|^2
\end{aligned} \tag{6.39}$$

$$E + D(-\delta t) + C(-\delta t)^2 + B(-\delta t)^3 + A(-\delta t)^4 = VGP^2\delta t^2 \tag{6.40}$$

6.4. Anexo 4: Simplecticidad del nuevo algoritmo.

Iniciemos con una descripción de las interacciones.

En el caso del algoritmo original la aceleración es:

$$a(x_n) \quad \text{donde, } x_n = x_{A,n} - x_{B,n} \tag{6.41}$$

x_n es la distancia entre las dos partículas. En el caso del nuevo algoritmo la aceleración es:

$$a(\bar{x}_{A,n}) \quad \text{donde, } |\bar{x}_{A,n}| = |x_{A,n}(t) - [x_{B,n}(t - \delta t)]| = \delta t_B VPG \quad (6.42)$$

$$a(\bar{x}_{B,n}) \quad \text{donde, } |\bar{x}_{B,n}| = |x_{B,n}(t) - [x_{A,n}(t - \delta t)]| = \delta t_A VPG \quad (6.43)$$

En general aunque δt_A y δt_B pueden estar muy cercanas en valor, no son iguales.

De acuerdo con la ecuación (3.10), el tiempo de retraso δt_A depende de las coordenadas (x_n, v_A) . Esto implica que la aceleración $a(\bar{x}_{B,n})$ de la partícula B es una función de dichas coordenadas (x_n, v_A) . De manera similar para $a(\bar{x}_{A,n})$.

$$\begin{aligned} a(\bar{x}_{B,n}) &= a(x_n, v_A) \\ a(\bar{x}_{A,n}) &= a(x_n, v_B) \end{aligned} \quad (6.44)$$

Por lo tanto las ecuaciones del nuevo algoritmo son:

Partícula A:

$$\begin{aligned} v_{A,n+1/2} &= v_{A,n} + \frac{1}{2} a(x_n, v_{B,n}) \Delta t \\ x_{A,n+1} &= x_{A,n} + \Delta t \cdot v_{A,n+1/2} \\ v_{A,n+1} &= v_{A,n+1/2} + \frac{1}{2} a(x_{n+1}, v_{B,n+1}) \Delta t \end{aligned} \quad (6.45)$$

Partícula B:

$$\begin{aligned} v_{B,n+1/2} &= v_{B,n} + \frac{1}{2} a(x_n, v_{A,n}) \Delta t \\ x_{B,n+1} &= x_{B,n} + \Delta t \cdot v_{B,n+1/2} \\ v_{B,n+1} &= v_{B,n+1/2} + \frac{1}{2} a(x_{n+1}, v_{A,n+1}) \Delta t \end{aligned} \quad (6.46)$$

Evaluamos todas las funciones en los puntos: (x_0, v_0) y $(x_0 + \delta x_0, v_0 + \delta v_0)$.

Ecuación 1:

$$v_{A,n+1/2} = v_{A,n} + \frac{1}{2} \Delta t \cdot a(x_n, v_{B,n}) \quad (6.47)$$

Evaluada en $(x_{A,0}, v_0)$:

$$v_{A,1/2} = v_{A,0} + \frac{1}{2} \Delta t \cdot a(x_0, v_{B,0}) \quad (6.48)$$

Evaluada en $(x_0 + \delta x_0, v_0 + \delta v_0)$.

$$v_{A,1/2} + \delta v_{A,1/2} = v_{A,0} + \delta v_{A,0} + \frac{\Delta t}{2} a(x_0 + \delta x_0, v_{B,0} + \delta v_{B,0}) \quad (6.49)$$

Se resta las dos ecuaciones anteriores:

$$\delta v_{A,1/2} = \delta v_{A,0} + \frac{\Delta t}{2} [a(x_0 + \delta x_0, v_{B,0} + \delta v_{B,0}) - a(x_0, v_{B,0})] \quad (6.50)$$

Si los valores de δx_0 y $\delta v_{B,0}$ son muy pequeños, podemos aproximar la función a utilizando derivadas parciales de la siguiente manera:

$$\delta v_{A,1/2} = \delta v_{A,0} + \frac{\Delta t}{2} \left[\frac{\partial a(x_0, v_{B,0})}{\partial x_0} \delta x_0 + \frac{\partial a(x_0, v_{B,0})}{\partial v_{B,0}} \delta v_{B,0} \right] \quad (6.51)$$

Al proceder de igual forma para la primera ecuación del algoritmo de la partícula B, se obtiene:

$$\delta v_{B,1/2} = \delta v_{B,0} + \frac{\Delta t}{2} \left[\frac{\partial a(x_0, v_{A,0})}{\partial x_0} \delta x_0 + \frac{\partial a(x_0, v_{A,0})}{\partial v_{A,0}} \delta v_{A,0} \right] \quad (6.52)$$

Se resta las dos ecuaciones anteriores tomando en cuenta que: $v_n = v_{A,n} - v_{B,n}$.

$$\delta v_{1/2} = \delta v_0 + \frac{\Delta t}{2} \left[\frac{\partial a(x_0, v_{B,0})}{\partial x_0} \delta x_0 + \frac{\partial a(x_0, v_{B,0})}{\partial v_{B,0}} \delta v_{B,0} - \frac{\partial a(x_0, v_{A,0})}{\partial x_0} \delta x_0 - \frac{\partial a(x_0, v_{A,0})}{\partial v_{A,0}} \delta v_{A,0} \right] \quad (6.53)$$

Se realiza un cambio de variable:

$$A = a(x_0, v_{A,0}) - a(x_0, v_{B,0}) \quad (6.54)$$

Entonces se puede escribir:

$$\frac{\partial a(x_0, v_{A,0})}{\partial v_{A,0}} = \frac{\partial A}{\partial v_{A,0}} = \frac{\partial A}{\partial v_0} \frac{\partial v_0}{\partial v_{A,0}} = \frac{\partial A}{\partial v_0} (1) = \frac{\partial A}{\partial v_0} \quad (6.55)$$

$$\frac{\partial a(x_0, v_{B,0})}{\partial v_{B,0}} = -\frac{\partial A}{\partial v_{B,0}} = -\frac{\partial A}{\partial v_0} \frac{\partial v_0}{\partial v_{B,0}} = -\frac{\partial A}{\partial v_0} (-1) = \frac{\partial A}{\partial v_0} \quad (6.56)$$

Con estos cambios la ecuación (4.9) toma la forma:

$$\delta v_{1/2} = \delta v_0 + \frac{\Delta t}{2} \left[-\frac{\partial A}{\partial x_0} \delta x_0 + \frac{\partial A}{\partial v_0} \delta v_{B,0} - \frac{\partial A}{\partial v_0} \delta v_{A,0} \right] \quad (6.57)$$

$$\delta v_{1/2} = \delta v_0 \left[1 - \frac{\Delta t}{2} \frac{\partial A}{\partial v_0} \right] - \frac{\Delta t}{2} \frac{\partial A}{\partial x_0} \delta x_0 \quad (6.58)$$

Podemos escribir de forma matricial:

$$\begin{pmatrix} \delta x_0 \\ \delta v_{1/2} \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ -\frac{\Delta t}{2} \frac{\partial A}{\partial x_0} & \left[1 - \frac{\Delta t}{2} \frac{\partial A}{\partial v_0} \right] \end{pmatrix} \begin{pmatrix} \delta x_0 \\ \delta v_0 \end{pmatrix} \quad (6.59)$$

Ecuación 2:

$$x_{A,n+1} = x_{A,n} + \Delta t \cdot v_{n+1/2} \quad (6.60)$$

Evaluada en (x_0, v_0) :

$$x_{A,1} = x_{A,0} + \Delta t \cdot v_{1/2} \quad (6.61)$$

Evaluada en $(x_0 + \delta x_0, v_0 + \delta v_0)$:

$$x_{A,1} + \delta x_{A,1} = x_{A,0} + \delta x_{A,0} + \Delta t \cdot v_{A,1/2} + \Delta t \cdot \delta v_{A,1/2} \quad (6.62)$$

Al restar las dos ecuaciones anteriores:

$$\delta x_{A,1} = \delta x_{A,0} + \Delta t \cdot \delta v_{A,1/2} \quad (6.63)$$

Luego de realizar el mismo proceso para la partícula B:

$$\delta x_{B,1} = \delta x_{B,0} + \Delta t \cdot \delta v_{B,1/2} \quad (6.64)$$

Al restar las dos ecuaciones anteriores, tomando en cuenta que $v_n = v_{A,n} - v_{B,n}$, se obtiene:

$$\delta x_1 = \delta x_0 + \Delta t \cdot \delta v_{1/2} \quad (6.65)$$

Podemos escribir de forma matricial:

$$\begin{pmatrix} \delta x_1 \\ \delta v_{1/2} \end{pmatrix} = \begin{pmatrix} 1 & \Delta t \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \delta x_0 \\ \delta v_{1/2} \end{pmatrix} \quad (6.66)$$

Ecuación 3:

$$v_{A,n+1} = v_{A,n+1/2} + \frac{\Delta t}{2} a(x_{n+1}, v_{B,n+1}) \quad (6.67)$$

Evaluada en el punto (x_0, v_0) :

$$v_{A,1} = v_{A,1/2} + \frac{\Delta t}{2} a(x_1, v_{B,1}) \quad (6.68)$$

Evaluada en el punto $(x_0 + \delta x_0, v_0 + \delta v_0)$

$$v_{A,1} + \delta v_{A,1} = v_{A,1/2} + \delta v_{A,1/2} + \frac{\Delta t}{2} a(x_1 + \delta x_1, v_{B,1} + \delta v_{B,1}) \quad (6.69)$$

Se resta las dos ecuaciones anteriores:

$$\delta v_{A,1} = \delta v_{A,1/2} + \frac{\Delta t}{2} [a(x_1 + \delta x_1, v_{B,1} + \delta v_{B,1}) - a(x_1, v_{B,1})] \quad (6.70)$$

Si los valores de δx_0 y $\delta v_{B,0}$ son muy pequeños, podemos aproximar la función a utilizando derivadas parciales de la siguiente manera:

$$\delta v_{A,1} = \delta v_{A,1/2} + \frac{\Delta t}{2} \left[\frac{\partial a(x_1, v_{B,1})}{\partial x_1} \delta x_1 + \frac{\partial a(x_1, v_{B,1})}{\partial v_{B,1}} \delta v_{B,1} \right] \quad (6.71)$$

Al proceder de igual forma para la primera ecuación del algoritmo de la partícula B, se obtiene:

$$\delta v_{B,1} = \delta v_{B,1/2} + \frac{\Delta t}{2} \left[\frac{\partial a(x_1, v_{A,1})}{\partial x_1} \delta x_1 + \frac{\partial a(x_1, v_{A,1})}{\partial v_{A,1}} \delta v_{A,1} \right] \quad (6.72)$$

Tomando en cuenta que: $v_n = v_{A,n} - v_{B,n}$.

Además, realizamos un cambio de variable:

$$A = a(x_0, v_{A,0}) - a(x_0, v_{B,0}) \quad (6.73)$$

Entonces se puede escribir:

$$\frac{\partial a(x_1, v_{A,1})}{\partial v_{A,1}} = \frac{\partial A}{\partial v_{A,1}} = \frac{\partial A}{\partial v_1} \frac{\partial v_1}{\partial v_{A,1}} = \frac{\partial A}{\partial v_1} (1) = \frac{\partial A}{\partial v_1} \quad (6.74)$$

$$\frac{\partial a(x_1, v_{B,1})}{\partial v_{B,1}} = -\frac{\partial A}{\partial v_{B,1}} = -\frac{\partial A}{\partial v_1} \frac{\partial v_1}{\partial v_{B,1}} = -\frac{\partial A}{\partial v_1} (-1) = \frac{\partial A}{\partial v_1} \quad (6.75)$$

Con estos cambios al restar las ecuaciones (4.27)-(4.28) se obtiene:

$$\delta v_1 = \delta v_{1/2} + \frac{\Delta t}{2} \left[-\frac{\partial A}{\partial x_1} \delta x_1 + \frac{\partial A}{\partial v_1} [\delta v_{B,1} - \delta v_{A,1}] \right] \quad (6.76)$$

$$\delta v_1 = \delta v_{1/2} + \frac{\Delta t}{2} \left[-\frac{\partial A}{\partial x_1} \delta x_1 + \frac{\partial A}{\partial v_1} [-\delta v_1] \right] \quad (6.77)$$

$$\delta v_1 + \frac{\Delta t}{2} \frac{\partial A}{\partial v_1} \delta v_1 = \delta v_{1/2} - \frac{\Delta t}{2} \frac{\partial A}{\partial x_1} \delta x_1 \quad (6.78)$$

$$\delta v_1 = \frac{1}{1 + \frac{\Delta t}{2} \frac{\partial A}{\partial v_1}} \delta v_{1/2} - \frac{\frac{\Delta t}{2} \frac{\partial A}{\partial x_1}}{1 + \frac{\Delta t}{2} \frac{\partial A}{\partial v_1}} \delta x_1 \quad (6.79)$$

Podemos escribir de forma matricial:

$$\begin{pmatrix} \delta x_1 \\ \delta v_1 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ -\frac{\frac{\Delta t}{2} \frac{\partial A}{\partial x_1}}{1 + \frac{\Delta t}{2} \frac{\partial A}{\partial v_1}} & \frac{1}{1 + \frac{\Delta t}{2} \frac{\partial A}{\partial v_1}} \end{pmatrix} \begin{pmatrix} \delta x_1 \\ \delta v_{1/2} \end{pmatrix} \quad (6.80)$$

Ordenamos todas las transformaciones en forma de multiplicación de matrices:

$$\begin{pmatrix} \delta x_1 \\ \delta v_1 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ -\frac{\Delta t}{2} \frac{\partial A}{\partial x_0} & \left[1 - \frac{\Delta t}{2} \frac{\partial A}{\partial v_0} \right] \end{pmatrix} \begin{pmatrix} 1 & h \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ -\frac{\frac{\Delta t}{2} \frac{\partial A}{\partial x_1}}{1 + \frac{\Delta t}{2} \frac{\partial A}{\partial v_1}} & \frac{1}{1 + \frac{\Delta t}{2} \frac{\partial A}{\partial v_1}} \end{pmatrix} \begin{pmatrix} \delta x_{A,0} \\ \delta v_0 \end{pmatrix} \quad (6.81)$$

Al calcular el determinante :

$$\begin{pmatrix} \delta x_1 \\ \delta v_1 \end{pmatrix} = A \cdot B \cdot C \begin{pmatrix} \delta x_0 \\ \delta v_0 \end{pmatrix} \quad (6.82)$$

Al calcular su determinante :

$$\det[J_x] = \det[A] \cdot \det[B] \cdot \det[C] \quad (6.83)$$

$$\det[J_x] = \frac{1 - \frac{\Delta t}{2} \frac{\partial A}{\partial v_0}}{1 + \frac{\Delta t}{2} \frac{\partial A}{\partial v_1}} \quad (6.84)$$

$$\det[J_x] \neq 1 \quad (6.85)$$

Se deduce que la transformación NO es canónica, por ende el algoritmo NO es simpléctico.

6.5. Anexo 5: Código Fortan90.

```

module parametros
integer , parameter :: ikind=selected_real_kind(32,307)
integer , parameter :: DIM=3
real(kind=ikind) , parameter :: CG=0.0001184462068986038_ikind !unidades astronomicas-years
real(kind=ikind) , parameter :: dt=0.0001_ikind !iteration step
integer , parameter :: N=9 !numero de particulas
real(kind=ikind) , parameter :: ep=0.1 !valor funcion epsilon
real(kind=ikind) , parameter :: epp=0.1 !valor funcion epsilon prima
integer(kind=ikind) :: step , nt
character*80 :: datin
integer :: dat , i , j , l , nn , aux
real(kind=ikind) :: eka , epa , eta , ret1 , ret2 , vp , Rsqij , phij , dphij , Rsqji , phji , dphji , rtc , Rsq
real :: start , finish
real(kind=ikind) , dimension(3,N) :: r , v , a , rret , rret1 , a0 , v0 !! posicion , velocidad , aceleracion
real(kind=ikind) , dimension(N,3) :: p , k
real(kind=ikind) , dimension(DIM) :: Rij , Rji , real_vel
real(kind=ikind) , dimension(N) :: dist , m
real(kind=ikind) , dimension(2,N) :: acc1
real(kind=ikind) , dimension(3,N) :: orb
real(kind=ikind) :: coef(0:2)
real(kind=ikind) :: mcm
real(kind=ikind) , dimension(3) :: rcm , vcm
real(kind=ikind) :: ep1max , ep2max
end module parametros

PROGRAM ver
use parametros
IMPLICIT NONE

call cpu_time(start)

nt=1000000_ikind ! ##numero de pasos termpora/

read(* , '(a)') datin
dat = len_trim(datin)
r=0.0_ikind
v=0.0_ikind
a=0.0_ikind
m(1)=333000_ikind
m(2)=0.0553_ikind
m(3)=0.815_ikind
m(4)=1_ikind
m(5)=0.1074_ikind
m(6)=317.8_ikind
m(7)=95.2_ikind
m(8)=14.54_ikind

```



```

m(9)=17.15_ikind
! m(2)=1.0_ikind
! m(3)=0.0123_ikind

!m=5000_ikind

!vp=100000000._ikind
!vp=1000000000000000.000000000001_ikind
!vp=10000_ikind
vp=9_ikind
!vp=0.5_ikind
!vp=0.06_ikind
!vp=0.008_ikind

write(*,*) "tiempo a simular:", nt*dt ,"year"
vp=vp*63198_ikind ! transforma de unidades de c a unidades de (ua/year)

##### cambio de coordenadas al CENTRO DE MASA #####!!!!!!!
open(unit=1, file=datin(1:dat), status='unknown')
do l=1,N
  read (1,*) r(:,l),v(:,l),a(:,l)
end do

do l=1,N
  mcm=mcm+m(l)
end do
do l=1,N
  rcm=rcm+m(l)*r(:,l)/mcm
  vcm=vcm+m(l)*v(:,l)/mcm
end do

do l=1,N
  r(:,l)=r(:,l)-rcm
  v(:,l)=v(:,l)-vcm
end do

write(*,*) r,v

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

! write(*,*) rcm,vcm,"cm"
! write(*,*) r,v,"antes"
coef=0_ikind
a=0_ikind
aux=0
step=1_ikind
dist=0_ikind
!open(file='tiempo.dat', unit=211, status='unknown')
open(file='epsimax.dat', unit=21, status='unknown')
open(file='ener.dat', unit=77, status='unknown')
open(file='dist.dat', unit=13, status='unknown')
open(file='orb.dat', unit=7, status='unknown')
open(file='dat.dat', unit=1, status='unknown')

```

!!!!!!!!!!!!!! INICIO UN PRIMER CALCULO DE LA ACELACION INICIAL !111!!!!!!!!!!!!

```

do i = 1,N-1
  do j = i+1,N
    coef(2)= dot_product(v(:,j),v(:,j))-vp**2
    coef(1)= 2*dot_product(r(:,i)-r(:,j),v(:,j))
    coef(0)= dot_product(r(:,i)-r(:,j),r(:,i)-r(:,j))
    ret1 = (-coef(1)-sqrt(coef(1)**2 - 4*(coef(2)*coef(0))))/(2*coef(2))
    coef(2)= dot_product(v(:,i),v(:,i))-vp**2
    coef(1)= 2*dot_product(r(:,j)-r(:,i),v(:,i))
    coef(0)= dot_product(r(:,j)-r(:,i),r(:,j)-r(:,i))
    ret2 = (-coef(1)-sqrt(coef(1)**2 - 4*(coef(2)*coef(0))))/(2*coef(2))
    rret(:,i) = r(:,i) - (ret2)*v(:,i)
    rret(:,j) = r(:,j) - (ret1)*v(:,j)
    Rij = r(:,i) - rret(:,j)
    Rji = r(:,j) - rret(:,i)
    Rsqij= dot_product(Rij ,Rij)
    Rsqji= dot_product(Rji ,Rji)
    phij = -(m(j)*CG*m(i))/sqrt(Rsqij)
    dphij = -(m(j)*CG)/((sqrt(Rsqij))**3)
    phji = -(m(j)*CG*m(i))/sqrt(Rsqji)
    dphji = -(CG*m(i))/((sqrt(Rsqji))**3)
    p(i,1) = p(i,1) + 0.5_ikind*phij
    p(j,1) = p(j,1) + 0.5_ikind*phji
    a(:,i) = a(:,i) + dphij*Rij
    a(:,j) = a(:,j) + dphji*Rji
  enddo
enddo

```

!!!!!!!!!!!!!! INICIO DEL ALGORITMO VV!!!!!!!!!!!!

```

do step=1,nt
  rret1=r
  v0=v
  a0=a

  !v(:, :) = v(:, :) + 0.5_ikind*dt*a(:, :)
  !r(:, :) = r(:, :) + dt*v(:, :)
  r(:, :) = r(:, :) + dt*v(:, :) + 0.5d0*(dt**2)*a(:, :)
  a(:, :)=0_ikind
  eka=0_ikind
  epa=0_ikind
  p=0_ikind
  do i = 1,N-1
    do j = i+1,N

      coef(2)= dot_product(v(:,j),v(:,j))-vp**2
      coef(1)= 2*dot_product(r(:,i)-r(:,j),v(:,j))
      coef(0)= dot_product(r(:,i)-r(:,j),r(:,i)-r(:,j))
      ret1 = (-coef(1)-sqrt(coef(1)**2 - 4*(coef(2)*coef(0))))/(2*coef(2))
      coef(2)= dot_product(v(:,i),v(:,i))-vp**2
      coef(1)= 2*dot_product(r(:,j)-r(:,i),v(:,i))
      coef(0)= dot_product(r(:,j)-r(:,i),r(:,j)-r(:,i))
      ret2 = (-coef(1)-sqrt(coef(1)**2 - 4*(coef(2)*coef(0))))/(2*coef(2))

```

```

if (ret1.gt.dt) then
  write(*,*) "delta t mayor Delta t", step
  goto 100
end if

if (isnan(ret1)) then
  write(*,*) "No hay valores reales para (delta t) al paso:", step
  goto 100
end if
if (isnan(ret2)) then
  write(*,*) "No hay valores reales para (delta t) al paso:", step
  goto 100
end if
rret(:,i) = r(:,i)-ret2*v(:,i)
rret(:,j) = r(:,j)-ret1*v(:,j)

Rij = r(:,i) - rret(:,j)
Rji = r(:,j) - rret(:,i)
Rsquij= dot_product(Rij ,Rij)
Rsquji= dot_product(Rji ,Rji)
phij = -(m(j)*CG*m(i))/sqrt(Rsquij)
dphij = -(m(j)*CG)/((sqrt(Rsquij))**3)
phji = -(m(j)*CG*m(i))/sqrt(Rsquji)
dphji = -(CG*m(i))/((sqrt(Rsquji))**3)
p(i,1) = p(i,1) + 0.5_ikind*phij
p(j,1) = p(j,1) + 0.5_ikind*phji
a(:,i) = a(:,i) + dphij*Rij
a(:,j) = a(:,j) + dphji*Rji
enddo
enddo
!v(:, :) = v(:, :) + 0.5_ikind*dt*(a(:, :))
v(:, :) = v(:, :) + 0.5d0*dt*(a(:, :)+a0(:, :))
aux=aux+1

!write(*,*) v
if(step.gt.3)then
do i=1,N
acc1(1,i)=sqrt(dot_product((a(:,i)-a0(:,i)),(a(:,i)-a0(:,i)))) /
sqrt(dot_product(a0(:,i),a0(:,i)))
ep1max = max(acc1(1,i),ep1max)
if(ep1max.gt.ep)then
write(*,*) "Simulacion no Coherente, limite de Epsilon
superado al paso:" , step , "Particula n0:" , i
goto 100
endif
acc1(2,i)=sqrt(dot_product((v(:,i)-v0(:,i)),(v(:,i)-v0(:,i)))) /
sqrt(dot_product(v0(:,i),v0(:,i)))
ep2max = max(acc1(2,i),ep2max)

if(ep2max.gt.epp)then
write(*,*) "Simulacion no Coherente, limite de EpsilonPrima
superado al paso:" , step , "Particula n0:" , i
goto 100
endif
enddo
enddo

```

```

endif

if (aux.eq.100) then
  write (*,*) 100*step/nt,"%",ret1,ep1max,ep2max
  !!!!! aqui se cambia el numero datos que se imprimen
  do i=1,N
    real_vel = v(:,i)
    eka = eka+0.5_ikind*m(i)*dot_product(real_vel,real_vel)
    epa=epa+p(i,1)
  enddo

  do nn=1,N
    write(1,*) r(1,nn),r(2,nn),r(3,nn)
  end do
  write(1,*) ' '
  write(1,*) ' '

  eta = eka+epa
  do i=1,N
    dist(i)=sqrt(dot_product(r(:,i)-r(:,1), r(:,i)-r(:,1)))
    orb(:,i)= r(:,i)-r(:,1)
  enddo

  dist(1)=sqrt(dot_product(r(:,2)-r(:,1), r(:,2)-r(:,1)))
  orb(:,1)= r(:,2)-r(:,1)
  dist(2)=sqrt(dot_product(r(:,3)-r(:,2), r(:,3)-r(:,2)))
  orb(:,2)= r(:,3)-r(:,2)

  write(7,*) orb
  write(13,*) step*dt,dist
  write(77,*) step*dt,eta
  aux=0
endif
enddo
100 write(21,*) ep1max,ep2max
close(1)
close(13)
close(77)
close(21)
close(7)
call cpu_time(finish)
write(*,*) finish-start

END PROGRAM ver

```

Bibliografía

- [1] Aksim, D. y Pavlov, D. On the extension of adams–bashforth–moulton methods for numerical integration of delay differential equations and application to the moon’s orbit. *Mathematics in Computer Science*, 14(1):103–109, jan 2020.
- [2] Frenkel, D. y Smit, B. *Understanding Molecular Simulation: From Algorithms to Applications*. Academic Press, 2nd edición, 2002.
- [3] Gallego, G. *Introducción a la geometría simpléctica y los sistemas integrables*, 2018.
- [4] Gerber, P. La propagación espacial y temporal de la gravedad. *Revista de Matemáticas y Física*, 43:93–104, 1898.
- [5] Hilding, A. y Weilow, E. Numerical algorithms for time-delayed newtonian gravity. <https://www.diva-portal.org/smash/get/diva2:1335372/FULLTEXT01.pdf>, 2019.
- [6] Joglekar, Y. N., Vemuri, G., y Wilkey, A. LAMBERT FUNCTION METHODS TO STUDY LASER DYNAMICS WITH TIME-DELAYED FEEDBACK. *Acta Polytechnica*, 57(6):399, dec 2017.
- [7] Laplace, P.-S. *A Treatise in Celestial Mechanics*, volume IV of Book X, Chapter VII. Chelsea, New York, translated edition edición, 1805.
- [8] L.D. Landau, E. L. *Mechanics*. Butterworth Heinemann, 3rd ed. reverté, barcelona edición, 1991.
- [9] Luca, J. D. Neutral delay and a generalization of electrodynamics, 2013.
- [10] Malcolm, M. A., Reid, J. T., y Blair, J. R. S. *Fortran90 for Scientists and Engineers*. Addison-Wesley, 1996.
- [11] McDonald, K. T. Laplace and the speed of gravity. Joseph Henry Laboratories, Princeton University, Princeton, NJ 08544, 2018. Retrieved March 15, 2023, from <http://kirkmcd.princeton.edu/examples/laplace.pdf>.
- [12] Pang, T. *Introduction to Computational Physics*. Cambridge University Press, 2nd edición, 2006.
- [13] Rey Suárez, S. Estructuras simplécticas. *Revista de Matemática: Teoría y Aplicaciones*, 13(2):89–108, 2006.
- [14] Roja, J. An exact solution of delay-differential equations in association models. *REVISTA MEXICANA DE FISICA* 57, 57(2):117–124, 2011.
- [15] Rojas, J. F., Martínez, R., y Morales, M. A. Mecánica 3d: python y el algoritmo de verlet. *Revista Mexicana de Física E*, 60(1):51–65, 2014.
- [16] Widmann, D. y Rackauckas, C. Delaydiffeq: Generating delay differential equation solvers via recursive embedding of ordinary differential equation solvers. *arXiv preprint arXiv:2208.12879*, 2022.
- [17] Young, P. The leapfrog method and other “symplectic” algorithms for integrating newton’s laws of motion. <https://young.physics.ucsc.edu/115/leapfrog.pdf>, 2014.