

# **EXTENSIÓN DEL ALGORITMO DE VERLET PARA EL CÁLCULO DE TRAYECTORIAS DE MOVILES EN PRESENCIA DE FUERZAS GRAVITACIONALES CON Y SIN FRICCIÓN**

## **1. Introducción**

### **1.1 Las ecuaciones diferenciales de 2º orden y la modelación matemática**

### **1.2 Métodos numéricos para la solución de ecuaciones diferenciales con valor inicial**

#### **1.2.1 Método de Euler**

#### **1.2.2 Método de Runge-Kutta**

#### **1.2.3 Aplicación comparativa de un problema simple**

## **2. El algoritmo de Verlet**

### **2.1 Deducción del Algoritmo clásico de Verlet y sus principales variantes**

### **2.2 Aplicación y evaluación del Algoritmo de Verlet para la simulación del movimiento planetario**

### **2.3 Navegación asistida por fuerzas gravitacionales**

### **2.4 Problema de 3 cuerpos**

## **3. Extensión del Algoritmo de Verlet para fuerzas que involucran términos de fricción (dependencia con la primera derivada)**

### **3.1 Desarrollo**

### **3.2 Generalización a dos o más dimensiones**

### **3.3 Aplicaciones del Algoritmo Extendido de Verlet**

#### **3.3.1 Movimiento de varios cuerpos con interacción electrostática en un medio resistivo.**

#### **3.3.2 Ingreso de un meteorito a la atmósfera terrestre.**

## **4. Conclusiones**

## 1. INTRODUCCIÓN

La Computación Científica (CC) ha tenido un gran impulso durante los últimos años que va claramente de la mano con el desarrollo de procesadores ultra rápidos y de tamaños cada vez más pequeños. De igual manera la capacidad de almacenaje de las computadoras parece que no tiene límite. Entendiendo la CC como una herramienta general para resolver problemas numéricos en distintas disciplinas con un enfoque científico, el auge e importancia de la Ciencia e Ingeniería de la Computación es más que natural. En particular en las áreas del conocimiento como la Física, Química, Biología e Ingeniería que intervienen en aplicaciones tecnológicas la CC da lugar a nuevas ramas del conocimiento como las Simulaciones en Medicina, Arquitectura, Física, Química Computacional entre otras. Desafortunadamente, en países desarrollados, la CC tiene aplicaciones hasta en los llamados Juegos de Guerra que se utilizan para optimizar estrategias de Guerra para aplicar en distintos posibles escenarios reales. Tal vez una de las aplicaciones más espectaculares de la CC sea la simulación en computadora de explosiones nucleares, para probar nuevas ideas, y así evitar hacer explosiones reales que contaminarían al planeta. En el caso de Juegos de Guerra muchos de los desarrollos e innovaciones en cuanto a la creación de nuevos algoritmos, técnicas numéricas, programas y hardware permanecen frecuentemente como secretos de estado que con el tiempo tal vez la comunidad científica y de Ingeniería y aun la misma Sociedad conozcan y disfruten.

La parte medular de ésta tesis tiene que ver con el desarrollo de nuevos algoritmos para la solución de ecuaciones de movimiento donde las fuerzas tienen términos que dependen linealmente y cuadráticamente de las velocidades, como es el caso de problemas donde hay fuerzas de fricción, los cuales, algunas veces pueden tratarse mediante algoritmos de optimización como el de Relajación en Energía. El algoritmo base y que se generaliza en la presente tesis es el algoritmo de Verlet usado en la técnica llamada de Dinámica Molecular que se aplica para estudiar el movimiento de muchas partículas sujetas a interacciones entre si y a campos externos. Ejemplos de

estos estudios son la formación y crecimiento de materiales, estabilidad y respuesta óptica de cristales líquidos, fatiga de materiales, por mencionar algunos. Nótese que los ejemplos mencionados tienen aplicaciones tecnológicas que provienen de la Ciencia de la Materia Condensada.

Otro aspecto relevante para la Ciencia e Ingeniería de la Computación es el concepto que se maneja de simulaciones en la presente tesis. Aparte del análisis numérico, propuesta de nuevos algoritmos y desarrollo de programas en ambiente WINDOWS, la tesis tiene en casi todos los ejemplos usados una visualización y una animación de la dinámica del fenómeno. Esto es, la simulación tiene como ingredientes el análisis numérico y animaciones de fenómeno a estudiar. Para los ejemplos presentados es como si se desarrollaran en un “Laboratorio Virtual” donde las condiciones y parámetros del problema se cambian a capricho del “experimentador”. Los programas así desarrollados son interactivos y pretenden que los puede manejar cualquier estudiante o investigador que quiera “experimentar” con los fenómenos físicos aquí planteados.

La presentación de la tesis está organizada de la siguiente manera: La extensión del Algoritmo de Verlet se presenta en el Cap. 3, mientras que en el Cap. 1 se hace una introducción sobre algoritmos clásicos para la solución de ecuaciones diferenciales de 1er y 2º orden con condiciones iniciales. En el Cap. 2 se desarrolla el algoritmo clásico de Verlet que se va a modificar. En el transcurso de las exposiciones de cada capítulo se hacen aplicaciones específicas para ilustrar o validar los algoritmos presentados. Se concluye en el Cap. 4. En los apéndices se encuentran copias de los programas de los ejemplos aquí desarrollados que a continuación se enlistan:

- Dinámica de cargas interactuando en un medio resistivo (1-D) mediante los métodos de Euler, Runge-Kutta, Scherer y Verlet Extendido
  - 1\_d\_Euler
  - 1\_d\_Runge

- 1\_d\_Scherer
  - 1\_d\_Verlet
- Cálculo de Órbitas Planetarias mediante el Algoritmo de Verlet
  - orbitas
- Navegación Asistida por Fuerzas Gravitacionales
  - S\_Solar
- Problema de Tres Cuerpos
  - 3\_cuerpos
- Dinámica de cargas interactuando en un medio resistivo (2-D) mediante el Algoritmo de Verlet Extendido y de Optimización (Relajación en Energía)
  - relax
- Ingreso de un Meteorito a la Atmósfera Terrestre
  - ingreso

## **1.1 LAS ECUACIONES DIFERENCIALES DE 2º ORDEN Y LA MODELACIÓN MATEMÁTICA**

La importancia del empleo de modelos para el estudio de algún sistema radica en la posibilidad de trabajar con una representación alternativa e indirecta de la realidad, es decir, tener la posibilidad de contar con algo que se comporte, parezca o reaccione como el sistema que deseamos estudiar, un ejemplo podría ser el uso de ratones en un laboratorio para probar algún medicamento, o la construcción de maniqués con sensores para el simulacro de un accidente automovilístico. En ambos casos, se busca modelar diferentes aspectos del cuerpo humano y cada modelo es adecuado para su aplicación, también es claro que un modelo será mejor en cuanto más fidedigna sea la información que se obtenga de él.

El concepto de modelo matemático tiene, quizá, un nivel de abstracción mayor que los ejemplos antes mencionados, ya que se refiere a una ecuación que contiene la mayor información posible acerca del sistema a estudiar basada en las leyes que rigen su comportamiento. De este modo, al analizar dicha ecuación mediante la solución de sus incógnitas, la variación de los posibles parámetros que contenga, la obtención de su representación gráfica, etcétera, obtendremos conocimiento acerca del sistema original al cual modelamos.

En el estudio de sistemas de partículas, el empleo de modelos matemáticos es indispensable, ya que uno de los objetivos de la física es la predicción de su comportamiento. Para ello, se cuentan con las reglas que gobiernan su evolución, es decir, los cambios que sufre cuando se modifica alguno de sus parámetros. Dichos cambios pueden ser representados a través del concepto matemático de la derivada, de modo que al plantear la relación que sostiene ésta con las reglas antes mencionadas, empleando una ecuación, se tendrá como resultado un modelo matemático.

La relación a partir de la cual es posible plantear las ecuaciones de movimiento es la 2ª ley de Newton, la cual es una ecuación diferencial:

$$F = m \frac{d^2 x}{dt^2}, \quad (1.1.1)$$

Donde  $F$  es una función que describe a la fuerza aplicada sobre un cuerpo de masa  $m$  que se encuentra en la posición  $x$ , la cual además puede depender de la velocidad del cuerpo, o de las características del medio en el cual se evoluciona el sistema.

Esta simple ecuación es la base para los modelos matemáticos que describen a los sistemas físicos en movimiento, sin embargo, su expresión puede complicarse tanto como sea necesario para representar a la fuerza que actúa sobre los cuerpos que intervengan en el análisis. Este hecho nos lleva, en la mayoría de los casos no triviales, a la formulación de problemas para los cuales es imposible obtener una expresión que prediga la trayectoria del móvil, es decir, que carecen de solución analítica, es en estos casos donde es indispensable el empleo de los métodos numéricos para dar una solución aproximada a un problema específico, definido por sus condiciones iniciales.

La aproximación numérica a la solución de estos problemas, llamados de *valor inicial*, consiste en obtener una tabulación de la posición del móvil correspondiente a una sucesión de valores del parámetro tiempo, es decir, a partir de una discretización del tiempo en intervalos de longitud  $h$ . Para ello se han desarrollado varios métodos para calcular la posición del móvil en un cierto tiempo a partir de las condiciones iniciales del problema. El desarrollo de estos métodos abre una amplia gama de aplicaciones dentro del área de la *Computación Científica* al involucrar el empleo de las computadoras para cálculos cada vez más complejos permitiendo así el desarrollo de programas que permitan al investigador interaccionar directamente con el modelo matemático de su objeto de estudio. Lo anterior requiere no sólo tener la solución numérica de algún problema, sino enriquecer el desarrollo de las herramientas de software para incrementar la posibilidad de tener simulaciones más realistas en lo que se refiere a velocidad de cálculo (simulaciones en tiempo real) e incluso a visualización, esto es, tener una representación en pantalla agradable, que contenga la

mayor información posible y que refleje fielmente los conceptos involucrados. Es por ello que es fundamental el desarrollo de nuevos métodos numéricos eficientes que aprovechen al máximo las ventajas del hardware que se desarrolla a cada momento.

El presente trabajo muestra algunos de los algoritmos usados para la solución numérica de ecuaciones diferenciales de segundo grado, así como aplicaciones representativas para la simulación de sistemas físicos. En particular, se muestra que el algoritmo clásico de Verlet, usado comúnmente en problemas de escala microscópica como es el caso de la dinámica molecular, es útil incluso para el cálculo hasta de órbitas planetarias, exhibiendo así el gran campo de aplicación del mismo.

Es imposible exagerar la importancia de este algoritmo dentro del área de la computación científica, propiamente en la física computacional, sin embargo, cuando la expresión que gobierna a la fuerza ( $m \, d^2x/dt^2$ ) contiene términos dependientes de la primera derivada, como ocurre al trabajar con sistemas en medios resistivos, el algoritmo presenta una dificultad. Aunque en la literatura se muestran resultados obtenidos mediante el Algoritmo de Verlet para cuando se manejan fuerzas dependientes de la velocidad, sin embargo, el detalle de la implementación del algoritmo no es expuesto en forma clara. A continuación se muestra el desarrollo erróneo en el cual puede caerse.

Consideremos un problema donde  $F = F(x, v)$ , con las condiciones iniciales  $x_0$  y  $v_0$ , entonces, el algoritmo de Verlet es:

$$\begin{aligned}x_{-1} &= x_0 - v_0 h + \frac{h^2}{2m} F(x_0, v_0) \\x_{n+1} &= 2x_n - x_{n-1} + \frac{h^2}{m} F(x_n, v_n) \\v_n &= \frac{1}{2h} [x_{n+1} - x_{n-1}]\end{aligned},$$

donde  $x_n = x(t+nh)$ ,  $v_n = v(t+nh)$ .

Entonces, desarrollando las primeras iteraciones:

$$\begin{aligned}
 x(-h) &= x_0 - v_0 h + \frac{h^2}{2m} F(x_0, v_0) \\
 x(0) &= x_0 \\
 x(h) &= 2x_0 - x(-h) + \frac{h^2}{2m} F(x_0, v_0) \\
 x(2h) &= 2x(h) - x(-h) + \frac{h^2}{m} F(x(h), v(h)) ,
 \end{aligned}$$

pero  $v(h)$  está dada por:

$$v(h) = \frac{1}{2h} [x(2h) - x_0],$$

de modo que el término  $x(2h)$  está indeterminado, por lo que sería necesario entonces usar el cálculo de la velocidad en el tiempo anterior, lo cual podría ocasionar inestabilidad en los resultados. Una forma alternativa presentada por C. Scherer<sup>1</sup>, consiste en calcular la velocidad por medio del método de Euler, es decir:

$$\begin{aligned}
 v_{n+1} &= v_n + hF(x_n, v_n) \\
 x_{n+1} &= 2x_n - x_{n-1} + \frac{h^2}{m} F(x_n, v_n) ,
 \end{aligned}$$

de este modo es posible usar la aproximación de la velocidad para el cálculo de la fuerza en la expresión original de Verlet. Este método se analiza en la sección correspondiente comparando su desempeño con el algoritmo alternativo que se propone en el presente trabajo donde se consideran expresiones para la fuerza que contienen términos dependientes de la posición y términos dependientes de la velocidad, de la forma:



$$\underline{F}(t) = \underline{F}^x(t) + \underline{F}^v(t),$$

con el cual se pretende no sólo enriquecer el campo de aplicación del algoritmo de Verlet, sino aumentar el alcance de la computación científica hacia el estudio de los sistemas dinámicos.

---

<sup>1</sup> <http://pcleon.if.ufrgs.br/~leon/metcomp/ode/node10.html>

## 1.2 MÉTODOS NUMÉRICOS PARA LA SOLUCIÓN DE ECUACIONES DIFERENCIALES CON VALOR INICIAL

Como se mencionó anteriormente, no todos los problemas de valor inicial pueden resolverse explícitamente, esto es, frecuentemente es imposible hallar una expresión matemática que represente la evolución del sistema con respecto al tiempo, de modo que es necesario disponer de métodos numéricos para obtener una aproximación a la solución. Para que esta aproximación sea útil en la mayoría de los casos es necesario que tenga una buena precisión, lo cual puede exigir un gran trabajo computacional, de modo que nuestra tarea consistirá en encontrar métodos con los cuales se obtenga la mayor la precisión con el menor número de operaciones posible. En este esfuerzo se han desarrollado varios métodos, con diferentes características de precisión y complejidad, de los cuales, mostramos algunos a continuación.

### 1.2.1 EL MÉTODO DE EULER

El método más sencillo para la solución de ecuaciones diferenciales es el llamado “Método de Euler” el cual se basa en la aproximación por diferencias de la derivada, de modo que tiene un error global relativamente alto, del orden de  $O(h)$ , su deducción es como sigue; si representamos  $y(t_n)$  como  $y_n$  se tiene<sup>2</sup>:

$$y' \approx \frac{y_{n+1} - y_n}{h} \quad (1.2.1)$$

donde  $h$  es el tamaño el paso, cuya elección dependerá de la tolerancia al error del problema particular que se ataque y que analizaremos más adelante, entonces:

$$y_{n+1} = y_n + hf(y_n, t_n).$$

---

<sup>2</sup> J.M. Mathews, K.D. Fink, *Métodos Numéricos con MatLab*, (Prentice Hall, Madrid, 2000) p. 470

De modo que teniendo la información del valor inicial  $y_0 = y(t_0)$  es posible desarrollar la tabulación para el intervalo deseado.

Dado que los problemas que nos interesa resolver (ecuaciones de movimiento) involucran a la segunda derivada, es necesario desarrollar el planteamiento para ecuaciones diferenciales de segundo orden mediante su descomposición en un sistema simultáneo de ecuaciones diferenciales de primer orden, de modo que:

$$x''(t) = f(x, x', t) \quad (1.2.2)$$

se puede escribir como:

$$\begin{aligned} x' &= v \\ v' &= f(x, v, t) \end{aligned} \quad (1.2.3)$$

y el esquema del método de Euler para ecuaciones de segundo orden quedaría como sigue:

$$\begin{aligned} v_{n+1} &= v_n + hf(x_n, v_n) \\ x_{n+1} &= x_n + hv_n \end{aligned} \quad (1.2.4)$$

Como se mostró anteriormente, este método se basa en la integración aproximada por diferencias finitas, no obstante, pueden obtenerse métodos más precisos y estables como el método de Runge-Kutta con el cual es posible obtener un planteamiento general, que, en principio, tendrá un error del orden  $O(h^n)$ .

### 1.2.2 MÉTODO DE RUNGE-KUTTA

Si tenemos la ecuación diferencial ordinaria:

$$y' = f(y, t) \quad (1.2.5)$$

podemos integrar en un intervalo  $[t_n, t_{n+1}]$  para obtener  $y_{n+1} = y(t_{n+1})$  a partir de  $y_n = y(t_n)$ :

$$y_{n+1} = y_n + \int_{t_n}^{t_{n+1}} f(y, t) dt \approx y_n + \frac{1}{2} h [f(y_n, t_n) + f(y_{n+1}, t_{n+1})] \quad (1.2.6)$$

notemos que  $y_{n+1}$  también aparece en el lado derecho de la ecuación como argumento de la función  $f$ , por lo que este segundo término será aproximado mediante el método de Euler, de modo que se tendrá<sup>3</sup>:

$$\begin{aligned} \tilde{y}_{n+1} &= y_n + hf(y_n, t_n) \\ y_{n+1} &= y_n + \frac{h}{2} [f(y_n, t_n) + f(\tilde{y}_{n+1}, t_{n+1})] \end{aligned} \quad (1.2.7)$$

Donde  $\tilde{y}_{n+1}$  es la aproximación a  $y_{n+1}$  por medio del método de Euler; así que podemos escribir lo anterior de la siguiente forma, obteniendo la forma general del método de Runge-Kutta de segundo orden:

$$\begin{aligned} k_1 &= hf(y_n, t_n) \\ k_2 &= hf(y_n + k_1, t_{n+1}) \\ y_{n+1} &= y_n + \frac{1}{2} [k_1 + k_2] \end{aligned} \quad (1.2.8)$$

---

<sup>3</sup> S. Nakamura, *Métodos Numéricos Aplicados con Software*, (Prentice Hall, México, 1992) p. 299

Del mismo modo en que se obtenía la solución numérica de ecuaciones diferenciales de segundo orden mediante un sistema de ecuaciones simultáneas de primer orden con el método de Euler, es posible escribir el método de Runge-Kutta para el sistema

$$\begin{aligned}x' &= v \\v' &= f(x, v, t)\end{aligned}\tag{1.2.9}$$

como:

$$\begin{aligned}k_1 &= hv(x_n, t_n) \\l_1 &= hf(x_n, v_n, t_n) \\k_2 &= h(v_n + l_1) \\l_2 &= hf(x_n + k_1, v_n + l_1, t_{n+1}) \\x_{n+1} &= x_n + \frac{1}{2}(k_1 + k_2) \\v_{n+1} &= v_n + \frac{1}{2}(l_1 + l_2)\end{aligned}\tag{1.2.10}$$

Ahora bien, con la fórmula de 1/3 de Simpson para la integral, se obtiene una mejor precisión dando lugar al método de Runge-Kutta de tercer orden, esto es, la integral

$$y_{n+1} = y_n + \int_{t_n}^{t_{n+1}} f(y, t) dt\tag{1.2.11}$$

puede expresarse numéricamente como:

$$y_{n+1} = y_n + \frac{h}{6} \left[ f(y_n, t_n) + 4f(\tilde{y}_{n+\frac{1}{2}}, t_{n+\frac{1}{2}}) + f(\tilde{y}_{n+1}, t_{n+1}) \right]\tag{1.2.12}$$

donde  $\tilde{y}_{n+\frac{1}{2}}$  y  $\tilde{y}_{n+1}$ , son estimaciones que se obtendrán nuevamente del método de Euler, de

modo que el esquema del método quedaría así:

$$\begin{aligned}
 k_1 &= hf(y_n, t_n) \\
 k_2 &= hf(y_n + \frac{1}{2}k_1, t_n + \frac{h}{2}) \\
 k_3 &= hf(y_n - k_1 + 2k_2, t_n + h) \\
 y_{n+1} &= y_n + \frac{1}{6}(k_1 + 4k_2 + k_3)
 \end{aligned}
 \tag{1.2.13}$$

y la extensión para resolver el sistema

$$\begin{aligned}
 x' &= v \\
 v' &= f(x, v, t)
 \end{aligned}
 \tag{1.2.14}$$

sería el siguiente:

$$\begin{aligned}
 k_1 &= hv_n \\
 l_1 &= hf(x_n, v_n, t_n) \\
 k_2 &= h(v_n + \frac{1}{2}l_1) \\
 l_2 &= hf(x_n + \frac{1}{2}k_1, v_n + \frac{1}{2}l_1, t_n + \frac{h}{2}) \\
 k_3 &= h(v_n - l_1 + 2l_2) \\
 l_3 &= hf(x_n - k_1 + 2k_2, v_n - l_1 + 2l_2, t_n + h)
 \end{aligned}
 \tag{1.2.15}$$

$$\begin{aligned}
 v_{n+1} &= v_n + \frac{1}{6}[l_1 + 4l_2 + l_3] \\
 x_{n+1} &= x_n + \frac{1}{6}[k_1 + 4k_2 + k_3]
 \end{aligned}
 \tag{1.2.16}$$

De manera análoga, se pueden ir agregando términos a la aproximación numérica de la integral y realizar el mismo proceso para encontrar métodos cada vez más precisos, incluso, existe el llamado *Método de Runge-Kutta de orden general*<sup>4</sup>, el cual tiene la siguiente expresión:

$$x_{n+1} = x_n + h \sum_{r=1}^R a_r k_r, \quad (1.2.17)$$

donde

$$k_1 = f(t_n, x_n),$$

y  $k_2, k_3, \dots, k_r, \dots, k_R$  están definidas por la relación recurrente:

$$k_r = f(t_n + hp_r, x_n + h \sum_{s=1}^{r-1} q_{r,s} k_s). \quad (1.2.18)$$

Ahora bien, es necesaria la manipulación algebraica para encontrar los valores de los coeficientes  $p$ ,  $q$  y  $a$ . A continuación, se desarrollará el Método de Runge-Kutta de cuarto orden para ejemplificar el desarrollo general.

De la ecuación (1.2.17) se tiene

$$x_{n+1} = x_n + h \{a_1 k_1 + a_2 k_2 + a_3 k_3 + a_4 k_4\}, \quad (1.2.19)$$

y de (1.2.18):

---

<sup>4</sup> S.C. Chapra, R.P. Canale, *Métodos Numéricos Para Ingenieros*, (Mc Graw Hill, México, 1989) p. 550

$$\begin{aligned}
k_1 &= f(t_n, x_n) \\
k_2 &= f\left(t_n + hp_2, x_n + h\{q_{2,1}k_1\}\right) \\
k_3 &= f\left(t_n + hp_3, x_n + h\{q_{3,1}k_1 + q_{3,2}k_2\}\right) \\
k_4 &= f\left(t_n + hp_4, x_n + h\{q_{4,1}k_1 + q_{4,2}k_2 + q_{4,3}k_3\}\right),
\end{aligned} \tag{1.2.20}$$

Obteniendo la expansión en serie de Taylor de orden  $N=4$  para  $k_2$ ,  $k_3$  y  $k_4$ , es posible sustituir estas aproximaciones en la ecuación **(1.2.19)** y obtener así una nueva expresión para  $x_{n+1}$ , la cual, al comparar términos con la expansión correspondiente al desarrollo en serie de la función de posición  $x$ , que puede verse como:

$$x(t + \Delta t) = x(t) + x'(t)\Delta t + x''(x, t)\frac{\Delta t^2}{2} + \dots \tag{1.2.21}$$

o bien:

$$x_{n+1} = x_n + f(x_n, t_n)h + f'(x_n, t_n)\frac{h^2}{2} + \dots \tag{1.2.22}$$

es posible identificar términos y llegar al conjunto de ecuaciones lineales siguiente:

$$\begin{aligned}
a_2 p_1 + a_3 p_2 + a_4 p_3 &= \frac{1}{2} \\
a_2 p_1^2 + a_3 p_2^2 + a_4 p_3^2 &= \frac{1}{3} \\
a_2 p_1^3 + a_3 p_2^3 + a_4 p_3^3 &= \frac{1}{4} \\
a_3 p_1 q_{3,2} + a_4 (p_1 q_{4,2} + p_2 q_{4,3}) &= \frac{1}{6} \\
a_3 p_1 p_2 q_{3,2} + a_4 p_3 (a_1 q_{4,2} + p_2 q_{4,3}) &= \frac{1}{8} \\
a_3 p_1^2 q_{3,2} + a_4 (p_1^2 q_{4,2} + p_2^2 q_{4,3}) &= \frac{1}{12} \\
a_4 p_1 q_{3,2} q_{4,3} &= \frac{1}{24}
\end{aligned} \tag{1.2.23}$$



y con las condiciones:

$$p_r = \sum_{s=1}^{r-1} q_{r,s} \quad \text{y} \quad \sum_{r=1}^R a_r = 1 \quad (1.2.24)$$

se tendrán 11 ecuaciones con 13 incógnitas, de modo que debemos incluir dos condiciones adicionales, por simplicidad se eligen:

$$p_1 = \frac{1}{2}, \quad \text{y} \quad q_{3,1} = 0.$$

Entonces, resolviendo el sistema se tiene que:

$$\begin{aligned} p_2 &= \frac{1}{2}, \quad p_3 = 1, \\ q_{2,1} &= \frac{1}{2}, \quad q_{3,2} = \frac{1}{2}, \quad q_{4,1} = 0, \quad q_{4,2} = 0, \quad q_{4,3} = 1, \\ a_1 &= \frac{1}{6}, \quad a_2 = \frac{1}{3}, \quad a_3 = \frac{1}{3}, \quad a_4 = \frac{1}{6}, \end{aligned} \quad (1.2.25)$$

de modo que el esquema final del método de Runge-Kutta de cuarto orden quedaría como sigue:

$$x_{n+1} = x_n + \frac{h}{6} (k_1 + 2k_2 + 2k_3 + k_4) \quad (1.2.26)$$

con:

$$\begin{aligned} k_1 &= f(t_n, x_n) \\ k_2 &= f\left(t_n + \frac{h}{2}, x_n + \frac{h}{2} k_1\right) \\ k_3 &= f\left(t_n + \frac{h}{2}, x_n + \frac{h}{2} k_2\right) \\ k_4 &= f(t_n + h, x_n + h k_3) \end{aligned}$$

Análogamente al caso anterior, es posible plantear un sistema de ecuaciones para resolver ecuaciones diferenciales de segundo orden y obtener un esquema similar al presentado en las ecuaciones (1.2.14)-(1.2.16).

Con este método se tiene un error global final de orden  $O(h^4)$ , y en principio, tomando  $n$  términos en la serie es posible obtener errores cada vez más pequeños ( $O(h^n)$ ), no obstante los métodos que resulten son también cada vez más complicados de programar e involucran más operaciones al aumentar significativamente el número de llamadas a la función, lo cual se ve reflejado fuertemente en el tiempo de cómputo requerido, lo cual nos lleva a tener un fuerte compromiso entre precisión y rapidez del cálculo, es aquí dónde la introducción del algoritmo de Verlet es importante, ya que se tiene una precisión equivalente a la del método de Runge-Kutta de tercer orden con la ventaja de que se emplean menos llamadas a la función y es muy sencillo de programar, lo cual lo hace ideal para la implementación de simulaciones.

### 1.2.3 APLICACIÓN COMPARATIVA EN UN PROBLEMA SIMPLE

Ahora, planteemos un problema sencillo para aplicar este resultado. Sean dos cargas eléctricas fijas  $q_1$  y  $q_2$  separadas una distancia  $l$ , y una carga libre  $q$  cuya distancia a la primera carga es  $x$ . Consideremos que  $\alpha$  es el coeficiente de fricción de  $q$  sobre la superficie. La ecuación que modela a las fuerzas que intervienen en su movimiento sería como sigue:

$$m \frac{d^2 x}{dt^2} = K \left[ \frac{qq_1}{r_1^2(x)} \hat{r}_1 + \frac{qq_2}{r_2^2(x)} \hat{r}_2 \right] - \alpha \frac{dx}{dt} - \beta \left( \frac{dx}{dt} \right)^2, \quad (1.2.27)$$

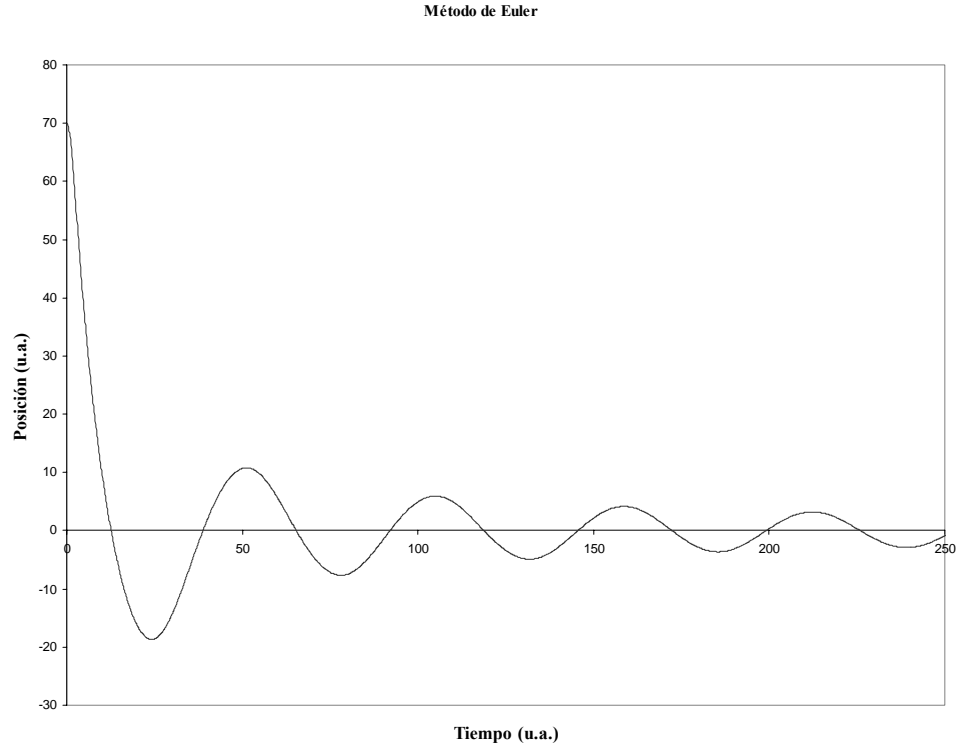
donde  $K$  es una constante con las unidades adecuadas,  $r_1, r_2$  son las distancias de  $q$  a  $q_1$  y  $q_2$  respectivamente,  $\alpha$  y  $\beta$  son los coeficientes de fricción lineal y cuadrática del móvil con la superficie. La expresión en un sistema de ecuaciones diferenciales es:

$$\begin{aligned} x' &= v \\ v' &= \frac{K}{m} \left[ \frac{qq_1}{r_1^2(x)} \hat{r}_1 + \frac{qq_2}{r_2^2(x)} \hat{r}_2 \right] - \frac{\alpha}{m} v - \frac{\beta}{m} v^2. \end{aligned} \quad (1.2.28)$$

El método de Euler estaría planteado de la siguiente manera:

$$\begin{aligned} v_{n+1} &= v_n + h \frac{K}{m} \left[ \frac{qq_1}{r_1^2(x_n)} \hat{r}_1 + \frac{qq_2}{r_2^2(x_n)} \hat{r}_2 \right] - \frac{\alpha}{m} v_n - \frac{\beta}{m} v_n^2, \\ x_{n+1} &= x_n + h v_n. \end{aligned} \quad (1.2.29)$$

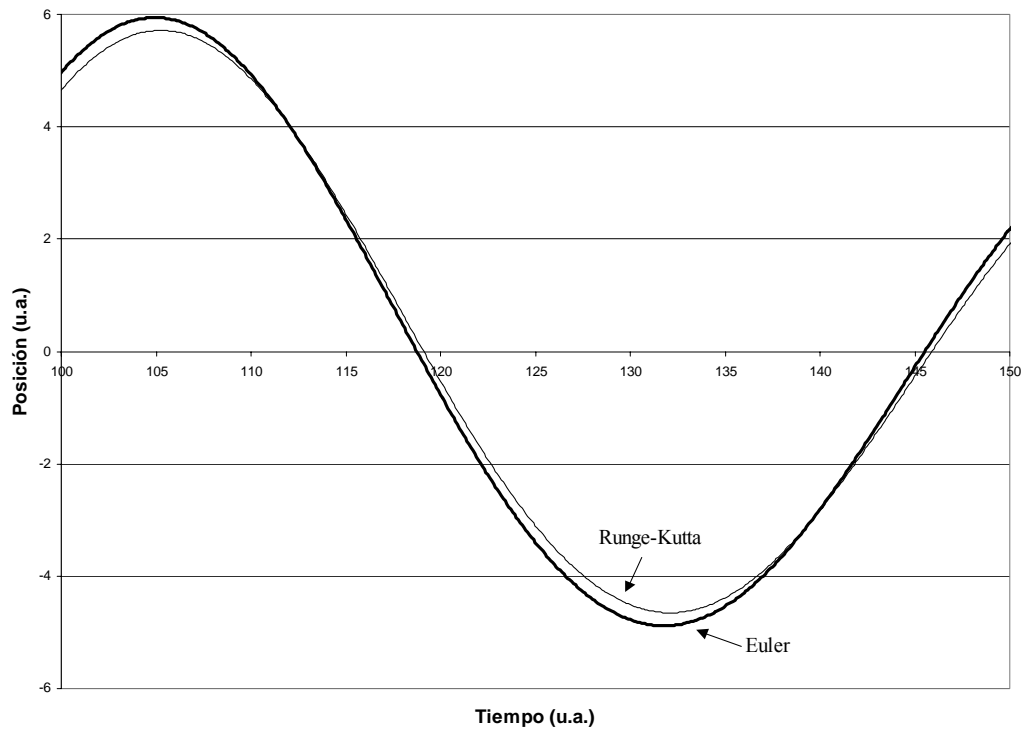
Es claro que en este caso la inicialización consiste simplemente en dar valores a  $x_0$  y  $v_0$ . Así, mediante un programa sencillo puede obtenerse la curva de la solución aproximada que aparece en la figura siguiente, donde hemos elegido  $q_1 = q_2$ , de modo que la carga móvil  $q$  tiende a estabilizarse en el origen.



Comparemos ahora este resultado con el obtenido al emplear el método de Runge-Kutta de segundo orden aplicando las ecuaciones (1.2.10). Obteniendo así:

$$\begin{aligned}
 k_1 &= hv_n \\
 l_1 &= h \frac{K}{m} \left[ \frac{qq_1}{r_1^2(x_n)} \hat{r}_1 + \frac{qq_2}{r_2^2(x_n)} \hat{r}_2 \right] - \frac{\alpha}{m} v_n - \frac{\beta}{m} v_n^2 \\
 k_2 &= h(v_n + l_1) \\
 l_2 &= h \frac{K}{m} \left[ \frac{qq_1}{r_1^2(x_n + k_1)} \hat{r}_1 + \frac{qq_2}{r_2^2(x_n + k_1)} \hat{r}_2 \right] - \frac{\alpha}{m} (v_n + l_1) - \frac{\beta}{m} (v_n + l_1)^2 \quad (1.2.30) \\
 v_{n+1} &= v_n + \frac{1}{2}(l_1 + l_2) \\
 x_{n+1} &= x_n + \frac{1}{2}(k_1 + k_2)
 \end{aligned}$$

Nuevamente, mediante la implementación del método con las ecuaciones anteriores, es posible obtener la curva solución, la cual es cualitativamente idéntica a la obtenida mediante el método de Euler, sin embargo, utilizando el mismo paso ( $h = 0.1$ ) los valores de los puntos difieren ligeramente (como se muestra en la figura siguiente) debido a la mayor precisión del método. Cabe mencionar que es posible mejorar la precisión del método de Euler mediante la elección adecuada del valor de  $h$ .



Para el esquema de Runge-Kutta de tercer orden, se tiene:

$$k_2 = h(v_n + \frac{1}{2}l_1)$$

$$l_2 = h \frac{K}{m} \left[ \frac{qq_1}{r_1^2(x_n + \frac{1}{2}k_1)} \hat{r}_1 + \frac{qq_2}{r_2^2(x_n + \frac{1}{2}k_1)} \hat{r}_2 \right] - \frac{\alpha}{m}(v_n + \frac{1}{2}l_1) - \frac{\beta}{m}(v_n + \frac{1}{2}l_1)^2$$

$$k_3 = h(v_n - l_1 + 2l_2)$$

$$l_3 = h \frac{K}{m} \left[ \frac{qq_1}{r_1^2(x_n - k_1 + 2k_2)} \hat{r}_1 + \frac{qq_2}{r_2^2(x_n - k_1 + 2k_2)} \hat{r}_2 \right] - \frac{\alpha}{m}(v_n - l_1 + 2l_2) - \frac{\beta}{m}(v_n - l_1 + 2l_2)^2$$

$$v_{n+1} = v_n + \frac{1}{6}(l_1 + 4l_2 + l_3)$$

$$x_{n+1} = x_n + \frac{1}{6}(k_1 + 4k_2 + k_3)$$

el cual una vez implementado, muestra que los valores en la tabulación difieren sólo en la tercera cifra decimal respecto al método anterior, de modo que para este ejemplo quizá la precisión del método de Runge-Kutta de segundo orden sea suficiente, no obstante, los métodos Runge-Kutta de ordenes superiores pueden ser útiles para aplicaciones más complejas.

## 2. EL ALGORITMO DE VERLET

### 2.1 DEDUCCIÓN DEL ALGORITMO CLÁSICO DE VERLET Y SUS PRINCIPALES VARIANTES

Este algoritmo para la solución de ecuaciones diferenciales de 2º orden ha tenido una historia larga y distinguida desde su aparición en el estudio de la dinámica de moléculas. Con este algoritmo se han obtenido resultados teóricos interesantes en el crecimiento de películas delgadas, fatiga de materiales, predicción de estructuras estables en moléculas, cristales líquidos y en un sin número de aplicaciones a sistemas microscópicos<sup>5</sup>. Cabe mencionar que, para atacar el problema de la solución de ecuaciones diferenciales de segundo orden con los métodos expuestos en las páginas anteriores, es necesario plantear un sistema de ecuaciones, lo cual aumenta considerablemente el número de operaciones a realizar en su implementación en un programa de cómputo, de modo que el algoritmo de Verlet, al trabajar directamente sobre la segunda derivada, ofrece ventajas en la mayoría de las aplicaciones donde interviene la mecánica clásica, sin embargo, como se verá más adelante, tiene limitaciones importantes en cuanto a la dependencia de la fuerza con la primera derivada, lo cual reduce de manera importante su aplicación en simulación de movimiento en medios resistivos.

El algoritmo de Verlet en su forma clásica se deduce a partir de la expansión en serie de Taylor de la función de posición  $x(t)$  en  $x(t + \Delta t)$ :

$$x(t + \Delta t) = x(t) + x'(t)\Delta t + x''(x, t)\frac{\Delta t^2}{2} + \dots \quad (2.1.1)$$

y en  $x(t - \Delta t)$ :

---

<sup>5</sup> L. Verlet, *Physical Review* **159**, 98 (1967)

$$x(t - \Delta t) = x(t) - x'(t)\Delta t + x''(x, t)\frac{\Delta t^2}{2} + \dots \quad (2.1.2)$$

sumando las Ecs. (2.1.1) y (2.1.2) obtenemos:

$$x(t + h) \approx 2x(t) - x(t - h) + h^2 x''(x, t), \quad (2.1.3)$$

donde hemos representado el incremento en tiempo  $\Delta t$  por el tamaño del paso  $h$ . O bien usando la notación anterior e introduciendo la ley de Newton  $F = m \frac{d^2 x}{dt^2}$ :

$$x_{n+1} = 2x_n - x_{n-1} + \frac{h^2}{m} F(x_n, t_n) \quad (2.1.4)$$

Para su inicialización, es necesario, además de la posición y velocidad iniciales, calcular el valor correspondiente a  $x(-h)$ , (o bien  $x_{-h}$ ) el cual se puede obtener del desarrollo en serie de Taylor "hacia atrás" haciendo  $t = 0$ :

$$x_{-1} = x_0 - v_0 h + \frac{h^2}{2m} F(x_0, t_0) \quad (2.1.5)$$

Con lo cual se tiene un método para la solución numérica de ecuaciones diferenciales de segundo grado, el cual tiene una forma sencilla y en sus aplicaciones a dinámica molecular (MD) ha resultado muy confiable pues ofrece gran estabilidad para la energía del sistema.

Una variante del algoritmo de Verlet es el llamado "Algoritmo *Leap-Frog*"<sup>6</sup>, la cual resulta de definir la velocidad como:

---

<sup>6</sup> R.W. Hockney, J.W. Eastwood, *Computer Simulation Using Particles* (McGraw Hill, New York, 1981)



$$v\left(t + \frac{h}{2}\right) \equiv \frac{x(t+h) - x(t)}{h}, \quad (2.1.6)$$

y

$$v\left(t - \frac{h}{2}\right) \equiv \frac{x(t) - x(t-h)}{h}, \quad (2.1.7)$$

entonces, tomando la diferencia de ambas se tiene:

$$v\left(t + \frac{h}{2}\right) - v\left(t - \frac{h}{2}\right) = \frac{2x(t) + x(t-h) + x(t+h)}{h}, \quad (2.1.8)$$

y el lado derecho de la ecuación, se puede comparar con la expresión del algoritmo clásico de Verlet, resultando:

$$v\left(t + \frac{h}{2}\right) - v\left(t - \frac{h}{2}\right) = x''(x, t)h, \quad (2.1.9)$$

entonces

$$v\left(t + \frac{h}{2}\right) = v\left(t - \frac{h}{2}\right) + a(x, t)h, \quad (2.1.10)$$

con este resultado y despejando  $x(t+h)$  de la primera definición se llega al sistema que describe a la variante *Leap-Frog*:

$$\begin{aligned} x(t+h) &= x(t) + hv\left(t + \frac{h}{2}\right), \\ v\left(t + \frac{h}{2}\right) &= v\left(t - \frac{h}{2}\right) + a(x, t)h. \end{aligned} \quad (2.1.11)$$

Esta variante tiene la ventaja de que permite conocer de manera explícita la velocidad aunque no en el mismo tiempo en que es calculada la posición, sin embargo, la velocidad al tiempo  $t$  se puede aproximar como:

$$v(t) = \frac{1}{2} \left[ v\left(t - \frac{h}{2}\right) + v\left(t + \frac{h}{2}\right) \right] \quad (2.1.12)$$

La siguiente variante es conocida como "Algoritmo de Verlet con velocidades explícitas", o simplemente "Algoritmo de Verlet de la velocidad"<sup>7</sup> el cual tiene la ventaja de permitir el cálculo de la posición y velocidad al tiempo  $t$ . Está dado de la siguiente forma:

$$\begin{aligned} x(t+h) &= x(t) + hv(t) + h^2 a(t) \\ v(t+h) &= v(t) + \frac{h}{2} [a(t) + a(t+h)] \end{aligned} \quad (2.1.13)$$

La primera expresión se obtiene de sustituir la expresión de la velocidad al tiempo  $t$ ,

$$v(t) = \frac{x(t) - x(t-h)}{h}, \quad (2.1.14)$$

en la expresión del Algoritmo clásico de Verlet. La segunda expresión (para la velocidad), se puede obtener de aplicar la regla extendida del trapecio para calcular

$$v(t+h) = v(t) + \int_t^{t+h} a(t) dt. \quad (2.1.15)$$

---

<sup>7</sup> W.C. Swope, H.C. Andersen, P.H. Berens, and K.R. Wilson, J. Chem. Phys. **76**, 637 (1982)

## 2.2 APLICACIÓN Y EVALUACIÓN DEL ALGORITMO DE VERLET PARA LA SIMULACIÓN DEL MOVIMIENTO PLANETARIO

Una aplicación interesante para el algoritmo de Verlet dadas las dimensiones involucradas, es el cálculo de trayectorias de móviles en presencia de fuerzas gravitacionales, ya que su empleo se ha enfocado generalmente a sistemas microscópicos. Intentemos pues, como primer ejercicio recuperar las órbitas planetarias del Sistema Solar.

Para agilizar los cálculos y facilitar el manejo de las cantidades involucradas, es necesario elegir las unidades adecuadas para el problema, en este caso, se manejarán unidades astronómicas para longitud (1 *ua* es el radio de la órbita terrestre), años para el tiempo, y masas solares (*mS*) como unidad de masa. Ahora bien, para el cálculo de la Constante Gravitacional *G* en las unidades elegidas, se emplea la Tercera Ley de Keppler<sup>8</sup>, la cual relaciona al periodo de un planeta con el radio de su órbita y con la masa del planeta atrayente:

$$T^2 = \frac{4\pi^2 a^3}{GM_0} \quad . \quad (2.2.1)$$

En este caso tomamos el radio de la órbita terrestre (1 *ua*) y el periodo de su órbita alrededor del sol (1 *año*), y la masa del Sol (1 *mS*), y despejamos *G*:

$$G = 4\pi^2 \left[ \frac{ua^3}{año^2 mS} \right] \quad . \quad (2.2.2)$$

El cual es un valor fácil de manejar, y dada su expresión es posible aprovechar al máximo la precisión de la máquina al realizar los cálculos.

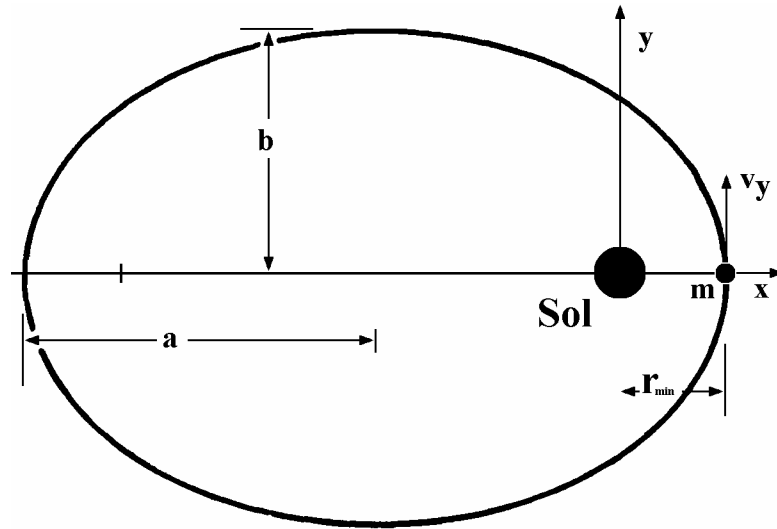
---

<sup>8</sup> M. Alonso, E.J. Finn, *Física, Vol. I, Mecánica*, (Addison-Wesley, México, 1986) p. 416

Ahora, debemos calcular las condiciones iniciales (velocidad y posición) para nuestra simulación en base a los datos que aparecen en los textos de enseñanza comunes, esto es periodo ( $T$ ), excentricidad ( $\epsilon$ ) y masa. Para facilitar el manejo algebraico definamos la cantidad  $\kappa$  como:

$$\kappa = GMm, \quad (2.2.3)$$

donde  $M$  y  $m$  son las masas de los planetas que intervienen en el sistema, de modo que  $M$  sería la masa del Sol y  $m$  la masa del planeta del cual deseamos obtener su trayectoria, de modo que en nuestras unidades  $\kappa = 4\pi^2 m$ .



Tenemos que la energía total del sistema está dada por<sup>9</sup>:

$$E = -\frac{GmM}{2a} = -\frac{k}{2a}, \quad (2.2.4)$$

y despejando  $a$  de la ecuación (2.2.1) y sustituyendo en la anterior se tiene:

$$E = -\frac{k}{2\left(\frac{T^2 k}{4\pi^2 m}\right)^{\frac{1}{3}}}, \quad (2.2.5)$$

y por lo tanto:

$$E = -\left(\frac{\pi^2 \kappa^2}{2T^2} m\right)^{\frac{1}{3}}, \quad (2.2.6)$$

con esto y la expresión de la excentricidad:

$$\varepsilon = \sqrt{1 + \frac{2EL^2}{m\kappa^2}} \quad (2.2.7)$$

obtenemos:

$$L = \sqrt{\frac{m\kappa^2(\varepsilon^2 - 1)}{2E}}, \quad (2.2.8)$$

y por otro lado, minimizando la expresión analítica de la trayectoria en coordenadas polares, se llega a lo siguiente:

$$r_{\min} = \frac{L^2}{(1 + \varepsilon)m\kappa}, \quad (2.2.9)$$

con lo cual, introduciendo en esta expresión las cantidades antes calculadas, obtenemos la posición inicial para nuestro problema. Por otro lado, expresando la energía total como la suma de las energías cinética y potencial:

---

<sup>9</sup> M. Alonso, E.J. Finn, *Física, Vol. I, Mecánica*, (Addison-Wesley, México, 1986) p. 427

$$E = \frac{1}{2}mv_y^2 - \frac{\kappa}{r_{\min}} \quad (2.2.10)$$

podemos obtener la velocidad inicial:

$$v_0^y = \sqrt{\frac{2}{m} \left( E + \frac{\kappa}{r_{\min}} \right)}. \quad (2.2.11)$$

Ahora bien, la fuerza central que interviene en el sistema está dada por:

$$\vec{F} = G \frac{mM}{r^2} \hat{r}, \quad (2.2.12)$$

donde  $\mathbf{r}$  es el vector que une ambas masas.

Entonces, dado que el sistema es bidimensional, el esquema del algoritmo de Verlet sería el siguiente:

$$\begin{aligned} x_{n+1} &= 2x_n - x_{n-1} + h^2 \frac{GM}{r^2} \cos(\theta) \\ y_{n+1} &= 2y_n - y_{n-1} + h^2 \frac{GM}{r^2} \text{sen}(\theta) \end{aligned} \quad (2.2.13)$$

donde  $\theta$  es el ángulo que forma  $\mathbf{r}$  con el eje horizontal. Para la inicialización del algoritmo se emplea la ecuación (2.1.5), de modo que:

$$\begin{aligned} x_{-1} &= x_0 - v_0^x h + \frac{h^2}{2m} F(x_0, y_0) \\ y_{-1} &= y_0 - v_0^y h + \frac{h^2}{2m} F(x_0, y_0) \end{aligned} \quad (2.2.14)$$

con  $v_0^x = 0$ ,  $y_0 = 0$  y  $x_0 = r_{\min}$ .

En este caso, es claro que el valor de  $h$  estará dado por el tamaño de intervalos en los cuales se divida el periodo de la órbita, lo cual será un factor importante para la precisión del cálculo, esto es, que al hacer divisiones más finas, podremos obtener mejores resultados, sin embargo, nuevamente, aparece el compromiso de tener una simulación ágil (en términos de velocidad de cálculo) o una simulación precisa, dado que la velocidad de procesamiento del equipo de cómputo será siempre una limitación. No obstante, en esta aplicación, el algoritmo de Verlet es sumamente versátil, ya que es posible variar el valor del paso  $h$  en un amplio rango de acuerdo con las características que se busquen en la aplicación particular.

Con los datos anteriores es posible llevar a cabo una implementación que presenta una animación del movimiento planetario y entrega una tabulación de posición contra tiempo. Paralelamente se realizó un planteamiento analítico del problema para evaluar el desempeño del método numérico:

Dado que al programar el método numérico obtenemos una posición para cada intervalo de tiempo  $nh$ , es necesario obtener la solución analítica del problema como una parametrización en el tiempo, lo cual presenta un problema interesante desde el punto de vista tanto físico como computacional, como puede verse en el siguiente desarrollo:

Partiendo de la condición de conservación del momento angular<sup>10</sup>:

$$\frac{d\theta}{dt} = \frac{L}{mr^2} \quad ; \quad \int_0^\theta mr^2 d\theta = L \int_0^t dt \quad , \quad (2.2.15)$$

---

<sup>10</sup> M. Alonso, E.J. Finn, *Física, Vol. I, Mecánica*, (Addison-Wesley, México, 1986) p. 428

en la cual es posible plantear  $r$  en función de  $\theta$  mediante la ecuación de una sección cónica en coordenadas polares:

$$r = \frac{\varepsilon d}{1 + \varepsilon \cos \theta}, \quad (2.2.16)$$

donde  $d$  es una constante dada por la distancia del foco a la directriz, y puede obtenerse mediante la siguiente relación:

$$d = r_{min} \left(1 + \frac{1}{\varepsilon}\right). \quad (2.2.17)$$

Sustituyendo la ecuación de la cónica en la ecuación (2.2.15) se tiene:

$$m\varepsilon^2 d^2 \int_0^\theta \frac{d\theta}{(1 + \varepsilon \cos \theta)^2} - Lt = 0, \quad (2.2.18)$$

y

$$m\varepsilon^2 d^2 \int_0^\theta \frac{d\theta}{(1 + \varepsilon \cos \theta)^2} = 2 \arctan \left\{ \frac{(\varepsilon - 1) \tan\left(\frac{\theta}{2}\right)}{\sqrt{1 - \varepsilon^2}} \right\} \left\{ \frac{m\varepsilon^2 d^2}{\sqrt{1 - \varepsilon^2} (\varepsilon^2 - 1)} + \frac{m\varepsilon^3 d^2 \sin \theta}{(\varepsilon^2 - 1)(1 + \varepsilon \cos \theta)} \right\} \quad (2.2.19)$$

entonces, de la ecuación (2.2.18) se tiene que:

$$2 \arctan \left\{ \frac{(\varepsilon - 1) \tan\left(\frac{\theta}{2}\right)}{\sqrt{1 - \varepsilon^2}} \right\} \left\{ \frac{m\varepsilon^2 d^2}{\sqrt{1 - \varepsilon^2} (\varepsilon^2 - 1)} + \frac{m\varepsilon^3 d^2 \sin \theta}{(\varepsilon^2 - 1)(1 + \varepsilon \cos \theta)} \right\} - Lt = 0. \quad (2.2.20)$$



Será necesario ahora encontrar el valor de  $\theta$  que satisfaga esta ecuación para cada tiempo  $t_n$ , con lo cual quedaría resuelto el problema al tener finalmente la parametrización  $\theta = \theta(t_n)$  y por lo tanto  $r = r(\theta(t_n))$  mediante la ecuación de la sección cónica.

Dado que la ecuación anterior carece de solución analítica, se empleó el método de Newton-Raphson para el cálculo de raíces, el cual produce una sucesión  $\{p_k\}$  que converge a la raíz  $p$  de la función  $f$  a partir de una aproximación inicial  $p_0$  mediante la siguiente relación iterativa<sup>11</sup>:

$$p_{n+1} = p_n - \frac{f(p_n)}{f'(p_n)}. \quad (2.2.21)$$

esto es:

$$\begin{aligned} \theta_{n+1} = \theta_n - 2 \arctan \left\{ \frac{(\varepsilon - 1) \tan\left(\frac{\theta_n}{2}\right)}{\sqrt{1 - \varepsilon^2}} \right\} & \frac{(1 + \varepsilon \cos \theta_n)^2}{\sqrt{1 - \varepsilon^2} (\varepsilon^2 - 1)} \\ & + \frac{\varepsilon(1 + \varepsilon \cos \theta_n) \sin \theta_n}{(\varepsilon^2 - 1)} - \frac{(1 + \varepsilon \cos \theta_n)^2}{m \varepsilon^2 d^2} L t \end{aligned} \quad (2.2.22)$$

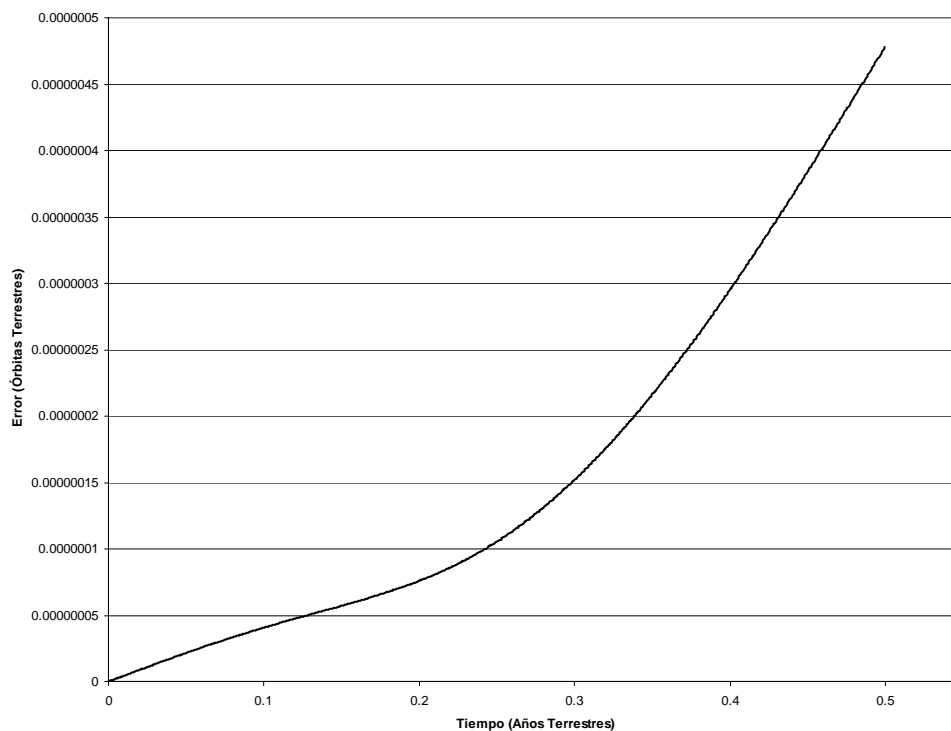
Con lo anterior se incluyó en el programa un módulo que realiza este cálculo, donde se especifica una cierta tolerancia para el valor de la raíz como criterio de paro en el desarrollo de la sucesión. Así pues, se dará como aproximación inicial el valor de  $\theta_n$  para calcular  $\theta_{n+1}$  teniendo en cuenta que  $\theta = 0$  en  $t = 0$ . Cabe mencionar que debido a que el proceso para calcular la raíz mediante el método de Newton-Raphson es iterativo, es decir, se calcula un término de la sucesión usando como dato al término anterior hasta que éste cumpla con

---

<sup>11</sup> J.M. Mathews, K.D. Fink, *Métodos Numéricos con MatLab*, (Prentice Hall, Madrid, 2000) p. 77

estar dentro de la tolerancia especificada, no es factible tener una idea clara acerca del número de operaciones necesarias para llegar a un resultado.

A continuación se muestra una gráfica donde aparece la distancia entre los puntos calculados mediante el algoritmo de Verlet y los calculados con el tratamiento analítico para el planeta Tierra.



La tendencia creciente del error en la gráfica anterior se debe a que en la ecuación trascendente (2.2.20) interviene la función  $\tan(\theta/2)$ , la cual crece indefinidamente para valores de  $\theta$  cercanos a  $\pi$ , por lo cual el método de Newton-Raphson se comporta de manera inestable. No obstante, el valor máximo de la gráfica, muestra una discrepancia del orden de  $5 \times 10^{-7}$  órbitas terrestres en el cálculo de la posición, lo cual equivale aproximadamente a 74.7 Km., representando esta cantidad un 0.6 % del diámetro ecuatorial de la tierra, de modo que puede considerarse un error aceptable.

Ahora veamos que tan útil puede ser al algoritmo de Verlet para predecir el tiempo en el que ocurre algún evento, como puede ser algún equinoccio o solsticio, para ello, es necesario hacer un cálculo analítico directo con los datos correspondientes a la Tierra para comparar posteriormente con el resultado que arroje el procedimiento numérico. Por ejemplo mediante la ecuación (2.2.20) podemos fácilmente obtener  $t$  para  $\theta = \pi/4$ , lo cual correspondería al equinoccio de Otoño y se reduce a:

$$t = \frac{1}{L} \left\{ 2 \arctan \left( \frac{\varepsilon - 1}{\sqrt{1 - \varepsilon^2}} \right) \right\} \frac{m \varepsilon^2 d^2 (1 + \varepsilon \sqrt{1 - \varepsilon^2})}{\sqrt{1 - \varepsilon^2} (\varepsilon^2 - 1)}, \quad (2.2.23)$$

y toma un valor de 0.2446839 años. Para obtener este dato empleando el algoritmo Verlet, simplemente se modificó el programa de modo que reporte el tiempo en el cual la coordenada X del móvil cambie de signo, resultando una diferencia con el cálculo anterior de  $1.1002 \times 10^{-6}$  años, lo cual equivale aproximadamente a 35 segundos y a 0.025 minutos de arco (un minuto de arco = 1/60 grados) empleando un valor para  $h$  de  $5 \times 10^{-6}$  años. Es posible mejorar este resultado disminuyendo el valor de  $h$ , sin embargo, esta precisión está comprometida con el tiempo de ejecución.

Con lo anterior se ha demostrado que el algoritmo de Verlet es útil en aplicaciones que involucran dimensiones astronómicas ya que los resultados que arroja son equivalentes a los obtenidos mediante un desarrollo analítico cuyo planteamiento es más complicado y su programación exige un tratamiento numérico más costoso en pasos de programa y llamadas a funciones, y por lo tanto en tiempo, que el propio algoritmo de Verlet.

### 2.3 NAVEGACIÓN ASISTIDA POR FUERZAS GRAVITACIONALES<sup>12</sup>

En la sección anterior se mostró que el algoritmo clásico de Verlet es útil para problemas que involucran dimensiones astronómicas, de modo que ahora tenemos la confianza de trabajar con él empleando un modelo más complicado para la fuerza que actúa sobre el móvil, como el llamado “Navegación asistida por fuerzas gravitacionales” y que consiste en lanzar un objeto al espacio de tal forma que se aproveche la interacción gravitacional con los cuerpos celestes para modificar su velocidad y trayectoria de acuerdo a una ruta planeada anteriormente, como se ha hecho con las sondas espaciales (Voyager, Pioner, etc.). El diseño de este tipo de lanzamientos requieren de un planteamiento sumamente complejo que involucra un nivel de detalle en el modelo matemático que escapa de los alcances del presente trabajo, sin embargo, es posible realizar una simulación simple del movimiento planetario en dos dimensiones considerando posiciones “reales” en el tiempo a partir de la fecha y hora del lanzamiento. A continuación se revisará el planteamiento del problema:

Emplearemos el sistema de unidades elegido para el ejemplo anterior, sin embargo, en este caso, la simulación de las órbitas planetarias que se presentó no es útil, ya que se requiere tener el dato de la posición de cada planeta del Sistema Solar para una fecha específica. De modo que será necesario basarse en el tipo de soluciones empleadas en los programas que calculan cartas celestes para fines educativos. Para nuestra simulación se empleó el algoritmo propuesto por Paul Schlyter<sup>13</sup>, el cual, dado que contempla algunas simplificaciones tiene una precisión de aproximadamente 2 minutos de arco, y las unidades que emplea son fácilmente adaptables a las que elegimos.

Con lo anterior, fue posible elaborar un programa que calculara las posiciones de los planetas para una fecha y hora específica y a partir de ésta actualiza la posición para los incrementos en el tiempo adecuados para aplicar el algoritmo de Verlet (del orden de  $10^{-4}$

---

<sup>12</sup> V. Barger, M. Olsson, *Classical Mechanics: a Modern Perspective*, (McGraw-Hill, New York, 1973) p. 131

<sup>13</sup> <http://hotel04.ausys.se/pausch/comp/ppcomp.html>

años). De este modo, con los datos requeridos, y aplicando el principio de superposición, es posible plantear la función que representa a la fuerza que actuaría sobre un objeto en el Sistema Solar debida a la atracción gravitacional de los cuerpos celestes:

$$\underline{F} = Gm \sum_i \frac{M_i}{r_i^2} \hat{r}_i, \quad (2.3.1)$$

donde  $m$  es la masa del móvil (para tener una idea del orden de magnitud, recordemos que el *Voyager II* tiene una masa de 722 Kg. aproximadamente)  $M_i$  es la masa del  $i$ -ésimo planeta y  $r_i$  es el vector que une a éste con el móvil.

De acuerdo con lo anterior, el esquema del algoritmo de Verlet sería como sigue:

$$\begin{aligned} x_{n+1} &= 2x_n - x_{n-1} + Gh^2 \sum_i \frac{M_i}{r_i^2} \cos(\theta_i) \\ y_{n+1} &= 2y_n - y_{n-1} + Gh^2 \sum_i \frac{M_i}{r_i^2} \sen(\theta_i) \end{aligned}, \quad (2.3.2)$$

donde  $\theta$  es el ángulo que forma  $r$  con el eje horizontal. Los datos de velocidad y posición inicial se dan en la inicialización del algoritmo mediante las siguientes relaciones:

$$\begin{aligned} x_{-1} &= x_0 - v_0^x h + \frac{h^2}{2m} F(x_0, y_0) \\ y_{-1} &= y_0 - v_0^y h + \frac{h^2}{2m} F(x_0, y_0) \end{aligned}. \quad (2.3.3)$$

En el caso que nos ocupa, la velocidad y posición inicial del Voyager estarán relacionados con la posición y velocidad de la tierra en el momento del lanzamiento, la idea es considerar que el proyectil es lanzado después de estar en órbita en la tierra a una distancia

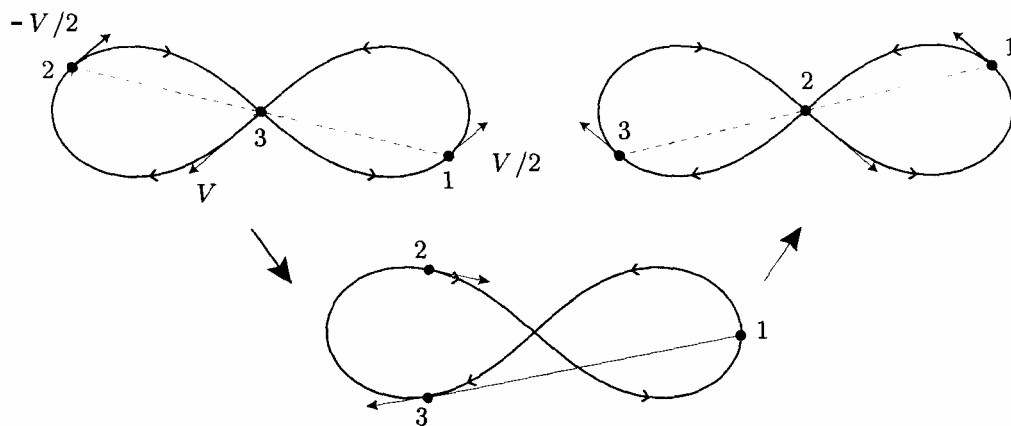
del orden de 30000 Km., y con una velocidad dada en dirección tangencial a la órbita de la tierra.

Así pues, se agregó al programa el desarrollo de Verlet así como algunas opciones en la visualización tales como “*Zoom in/out*” y la posibilidad de seguir (es decir, cambiar el sistema de referencia durante la simulación) a algún objeto del sistema, ya sea la propia sonda o a algún planeta.

## 2.4 PROBLEMA DE 3 CUERPOS

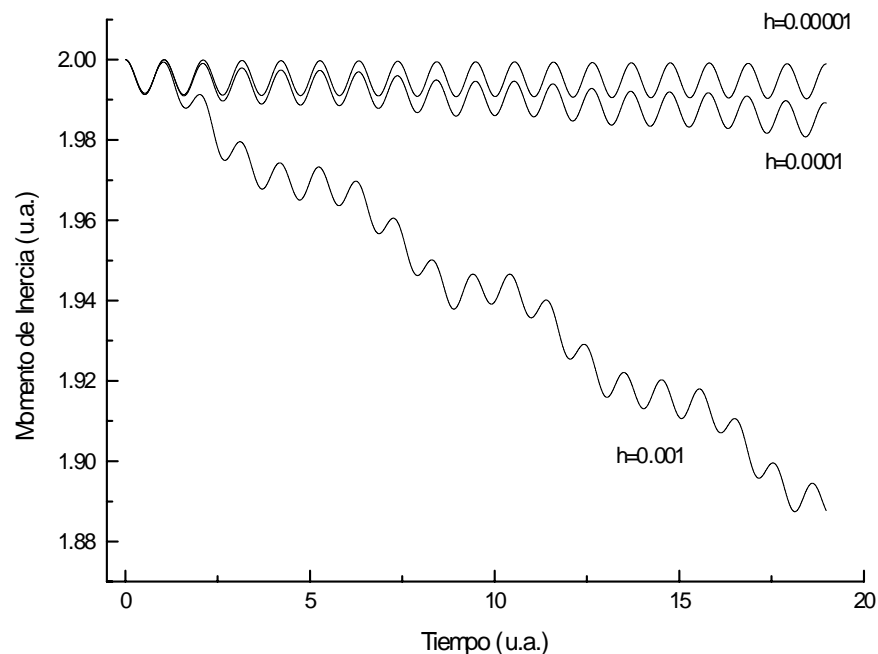
Un problema sumamente interesante de la mecánica clásica es el llamado “*Problema de  $n$  cuerpos*”, cuyo tratamiento mediante el algoritmo de Verlet fue introducido en el presente trabajo por sugerencia del Dr. Eduardo Piña, investigador de la Universidad Metropolitana, consiste en describir la trayectoria que seguirían un número arbitrario de partículas aisladas interaccionando gravitacionalmente entre sí. De forma analítica existen avances recientes en cuanto a las soluciones de sistemas de hasta 3 cuerpos, sin embargo, empleando algún método numérico eficiente, es posible generalizar la solución estando limitado el número de cuerpos solamente por la capacidad y velocidad del equipo de cómputo. A continuación, emplearemos el algoritmo de Verlet para verificar algunas soluciones analíticas.

En la referencia al pie<sup>14</sup> se presenta una solución particular al problema de 3 cuerpos demostrando la existencia de una órbita analítica estable con momento angular cero, momento de inercia constante y un patrón característico con algunas propiedades de simetría, en la cual las partículas siguen una trayectoria en forma de “*ocho*” como puede verse en la figura. Este resultado es particularmente interesante, ya que anteriormente, la única órbita calculada de este tipo era la conocida como “Equilibrio Relativo de Lagrange”, en la cual las partículas se mueven sobre un círculo formando un triángulo equilátero.



<sup>14</sup> A. Chenciner, and R. Montgomery, *Annals of Mathematics*, **152** (2000) 881

Esta aplicación puede ser útil para estimar un valor de  $h$  adecuado, ya que se demuestra analíticamente que el momento de inercia permanece constante durante la evolución del sistema, de modo que podemos seguir el valor de este parámetro durante la simulación implementada mediante el algoritmo de Verlet, empleando diferentes valores para el paso  $h$ , obteniendo el resultado que se muestra en la figura siguiente:



Podemos observar una clara dependencia del parámetro  $h$  con la estabilidad del momento de inercia, que si bien para valores relativamente altos de  $h$  (del orden de  $10^{-3}$ ) el error numérico acumulado hace colapsar la simulación en un tiempo corto, al disminuir en dos órdenes de magnitud ( $h = 10^{-5}$ ) se tiene una estabilidad aceptable durante la simulación, teniendo sólo pequeñas fluctuaciones periódicas de  $0.0005 \text{ u.a.}$  que representan alrededor del  $0.025 \%$  valor promedio del momento de inercia.



### 3. EXTENSIÓN DEL ALGORITMO DE VERLET PARA FUERZAS QUE INVOLUCRAN TÉRMINOS DE FRICCIÓN (DEPENDENCIA CON LA PRIMERA DERIVADA)

#### 3.1 DESARROLLO

Como se ha expuesto, el algoritmo de Verlet ha demostrado ser útil para diversas aplicaciones, además de ser eficiente dada sencillez en su implementación, al número de operaciones a realizar y en llamadas a la función que describe a la fuerza involucrada en el sistema que otros métodos usados comunmente, sin embargo, a diferencia de los métodos que se han expuesto en las páginas anteriores, la velocidad no aparece de manera explícita, y su expresión

$$v_n = \frac{x_{n+1} - x_{n-1}}{2h}, \quad (3.1.1)$$

involucra a la incógnita, por lo cual es necesario realizar el tratamiento algebraico adecuado.

A continuación se expondrá un desarrollo original mediante el cual se amplía enormemente el rango de aplicación del algoritmo de Verlet conservando sus cualidades originales. Esta extensión nos permite trabajar en problemas en los cuales intervienen fuerzas de fricción y donde podemos representar a la fuerza total como:

$$\underline{F}(t) = \underline{F}^x(t) + \underline{F}^v(t), \quad (3.1.2)$$

esto es, una componente que depende únicamente de la posición y otra que depende únicamente de la velocidad, es claro que en ambas existe una dependencia intrínseca con el tiempo. Como hemos visto en los ejemplos anteriores, la fuerza de fricción se escribe como:

$$\begin{aligned}
\underline{E}^v &= (-\alpha|v| - \beta|v|^2)\hat{v} \\
&= \left(-\alpha|v| - \beta|v|^2\right)\frac{v}{|v|} \\
&= -\alpha v - \beta v|v|
\end{aligned} \tag{3.1.3}$$

De modo que el algoritmo de Verlet (para una dimensión) se ve como:

$$\begin{aligned}
x_{n+1} &= 2x_n - x_{n-1} + \frac{F^x + F^v}{m} h^2 \\
&= 2x_n - x_{n-1} + h^2 \left( a_n^x - \frac{\alpha}{m} v - \frac{\beta}{m} v|v| \right)
\end{aligned} \tag{3.1.4}$$

Substituyendo la velocidad calculada anteriormente:

$$\begin{aligned}
x_{n+1} &= 2x_n - x_{n-1} + a_n^x h^2 - \frac{\alpha h^2}{m} \left( \frac{x_{n+1} - x_{n-1}}{2h} \right) \\
&\quad - \frac{\beta h^2}{m} \left( \frac{x_{n+1} - x_{n-1}}{2h} \left| \frac{x_{n+1} - x_{n-1}}{2h} \right| \right),
\end{aligned} \tag{3.1.5}$$

para tener una notación más compacta, definamos  $\bar{\beta}$  como:

$$\bar{\beta} = \begin{cases} \beta & \text{si } x_{n+1} - x_{n-1} > 0 \\ -\beta & \text{si } x_{n+1} - x_{n-1} < 0 \end{cases} \tag{3.1.6}$$

De este modo  $\bar{\beta}$  conserva la información del signo de la velocidad, entonces:

$$\begin{aligned}
x_{n+1} &= 2x_n - x_{n-1} + a_n^x h^2 - \frac{\alpha h^2}{m} \left( \frac{x_{n+1} - x_{n-1}}{2h} \right) - \frac{\bar{\beta} h^2}{m} \left( \frac{x_{n+1} - x_{n-1}}{2h} \right)^2 \\
&= 2x_n - x_{n-1} + a_n^x h^2 - \frac{\alpha h}{2m} x_{n+1} + \frac{\alpha h}{2m} x_{n-1} - \frac{\bar{\beta}}{4m} [x_{n+1}^2 + x_{n-1}^2 - 2x_{n+1}x_{n-1}] \\
&= 2x_n - x_{n-1} + a_n^x h^2 - \frac{\alpha h}{2m} x_{n+1} + \frac{\alpha h}{2m} x_{n-1} - \frac{\bar{\beta}}{4m} x_{n+1}^2 - x_{n-1}^2 + \frac{\bar{\beta}}{2m} x_{n+1}x_{n-1} .
\end{aligned} \tag{3.1.7}$$

Arreglando los términos obtenemos una ecuación de segundo orden para  $x_{n+1}$  :

$$x_{n+1}^2 \frac{\bar{\beta}}{4m} + x_{n+1} \left[ 1 + \frac{\alpha h}{2m} - \frac{\bar{\beta}}{2m} x_{n-1} \right] - 2x_n + x_{n-1} \left[ 1 - \frac{\alpha h}{2m} \right] + x_{n-1}^2 \frac{\bar{\beta}}{4m} - a_n^x h^2 = 0, \tag{3.1.8}$$

entonces:

$$x_{n+1} = \frac{-B \pm \sqrt{B^2 - 4AC}}{2A}, \tag{3.1.9}$$

con:

$$\begin{aligned}
A &= \frac{\bar{\beta}}{4m} \\
B &= 1 + \frac{\alpha h}{2m} - \frac{\bar{\beta}}{2m} x_{n-1} \\
C &= -2x_n + \left[ 1 - \frac{\alpha h}{2m} \right] x_{n-1} + \frac{\bar{\beta}}{4m} x_{n-1}^2 - a_n^x h^2
\end{aligned} \tag{3.1.10}$$

Ahora:

$$B^2 = \left( 1 + \frac{\alpha h}{2m} \right)^2 + \frac{\bar{\beta}^2}{4m^2} x_{n-1}^2 - \left( 1 + \frac{\alpha h}{2m} \right) \frac{\bar{\beta}}{m} x_{n-1} \tag{3.1.11}$$

$$4AC = \frac{\bar{\beta}}{m} \left\{ -2x_n + \left[ 1 + \frac{\alpha h}{2m} \right] x_{n-1} + \frac{\bar{\beta}}{4m} x_{n-1}^2 - a_n^x h^2 \right\} \quad (3.1.12)$$

y

$$B^2 - 4AC = \left( 1 + \frac{\alpha h}{2m} \right)^2 + \frac{\bar{\beta}}{m} [2x_n - 2x_{n-1} + a_n^x h^2]$$

entonces:

$$x_{n+1} = x_{n-1} - \left( 1 + \frac{\alpha h}{2m} \right) \frac{2m}{\bar{\beta}} \pm \frac{2m}{\bar{\beta}} \sqrt{\left( 1 + \frac{\alpha h}{2m} \right)^2 + \frac{\bar{\beta}}{m} [2x_n - 2x_{n-1} + a_n^x h^2]} \quad (3.1.13)$$

Ahora bien, tenemos que discernir cuál de los signos es el correcto, para ello, desarrollemos la raíz en serie para  $\bar{\beta} \ll 1$ :

$$\begin{aligned} & \sqrt{\left( 1 + \frac{\alpha h}{2m} \right)^2 + \frac{\bar{\beta}}{m} [2x_n - 2x_{n-1} + a_n^x h^2]} \\ &= \left( 1 + \frac{\alpha h}{2m} \right) \left\{ 1 + \frac{\bar{\beta}}{m \left( 1 + \frac{\alpha h}{2m} \right)^2} [2x_n - 2x_{n-1} + a_n^x h^2] \right\}^{\frac{1}{2}} \end{aligned} \quad (3.1.14)$$

y entonces podemos aplicar el desarrollo en serie:

$$(1+x)^n = 1 + nx + \frac{n(n-1)}{2!} x^2 + \dots \quad (3.1.15)$$

$$\begin{aligned}
\sqrt{\left(1 + \frac{\alpha h}{2m}\right)^2 + \frac{\bar{\beta}}{m} [2x_n - 2x_{n-1} + a_n^x h^2]} &= \left(1 + \frac{\alpha h}{2m}\right) \left[ 1 + \frac{1}{2} \frac{\bar{\beta}}{m \left(1 + \frac{\alpha h}{2m}\right)^2} [2x_n - 2x_{n-1} + a_n^x h^2] + \dots \right] \\
&= \left(1 + \frac{\alpha h}{2m}\right) + \frac{\bar{\beta}}{2m \left(1 + \frac{\alpha h}{2m}\right)} [2x_n - 2x_{n-1} + a_n^x h^2] + \dots
\end{aligned}
\tag{3.1.16}$$

Sustituyendo lo anterior en la expresión para  $x_{n+1}$  tenemos:

$$\begin{aligned}
x_{n+1} &\approx x_{n-1} - \left(1 + \frac{\alpha h}{2m}\right) \frac{2m}{\bar{\beta}} \pm \frac{2m}{\bar{\beta}} \left\{ \left(1 + \frac{\alpha h}{2m}\right) + \frac{\bar{\beta}}{2m \left(1 + \frac{\alpha h}{2m}\right)} [2x_n - 2x_{n-1} + a_n^x h^2] \right\} \\
&\approx x_{n-1} - \left(1 + \frac{\alpha h}{2m}\right) \frac{2m}{\bar{\beta}} \pm \left(1 + \frac{\alpha h}{2m}\right) \frac{2m}{\bar{\beta}} \pm \frac{1}{\left(1 + \frac{\alpha h}{2m}\right)} [2x_n - 2x_{n-1} + a_n^x h^2]
\end{aligned}
\tag{3.1.17}$$

Si se toma la solución positiva:

$$x_{n+1} = x_{n-1} + \frac{1}{\left(1 + \frac{\alpha h}{2m}\right)} [2x_n - 2x_{n-1} + a_n^x h^2] ,
\tag{3.1.18}$$

si  $\alpha \rightarrow 0$  se recupera la expresión original para el método de Verlet sin fricción:

$$x_{n+1} = x_{n-1} + 2x_n - 2x_{n-1} + a_n^x h^2 ,
\tag{3.1.19}$$

mientras que si tomamos la solución negativa, se llega a:

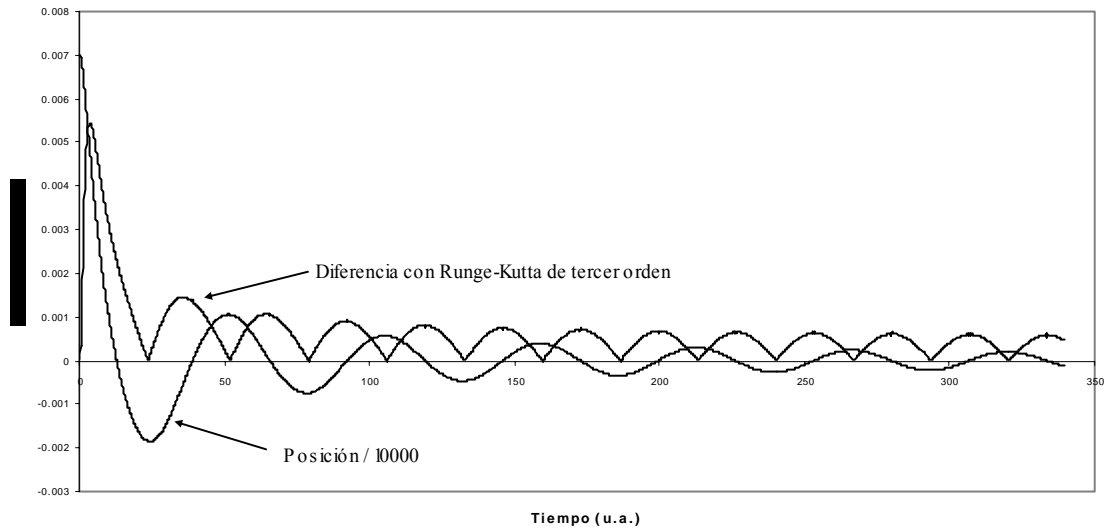
$$x_{n+1} = x_{n-1} - 2 \left( 1 + \frac{\alpha h}{2m} \right) \frac{2m}{\bar{\beta}} - \frac{2x_n - 2x_{n-1} + a_n^x h^2}{\left( 1 + \frac{\alpha h}{2m} \right)}, \quad (3.1.20)$$

con lo cual no se recupera el comportamiento original, por lo tanto tenemos que la expresión correcta es:

$$x_{n+1} = x_{n-1} - \frac{2m}{\bar{\beta}} \left\{ \left( 1 + \frac{\alpha h}{2m} \right) - \sqrt{\left( 1 + \frac{\alpha h}{2m} \right)^2 + \frac{\bar{\beta}}{m} [2x_n - 2x_{n-1} + a_n^x h^2]} \right\}. \quad (3.1.21)$$

Lo anterior es válido para  $\bar{\beta} \neq 0$ , sin embargo, si se requiere aplicar el modelo lineal para la fuerza de fricción  $\underline{F}^v = -\alpha \underline{v}$ , se aplica un planteamiento simplificado que se describe más adelante.

Con este nuevo algoritmo se atacó el mismo problema que se ha manejado hasta ahora, y como puede verse en la siguiente figura, la diferencia numérica con el resultado obtenido con el método de Runge-Kutta de tercer orden es mínima (del orden de  $6 \times 10^{-4}$  en promedio), comparada con las dimensiones del sistema, donde la separación entre las cargas es de 180 u.a. Nótese que esta diferencia tiene máximos relativos en los puntos donde la partícula pasa por el origen, esto es, donde su velocidad es máxima, y se van atenuando en la medida en que esta disminuye, asimismo, la diferencia se anula en los puntos donde la partícula cambia de dirección, es decir, donde su velocidad es cero.



Con este análisis podemos concluir que el algoritmo extendido de Verlet tiene al menos la precisión del método de Runge-Kutta de tercer orden, sin embargo, el número de operaciones necesarias para cada iteración, es mayor en éste último para un problema típico de mecánica clásica del tipo  $mf(\bar{x}, \bar{v}) = \frac{d^2 \bar{x}}{dt^2}$ , ya que como se describió anteriormente, se requiere el planteamiento de un sistema dos ecuaciones diferenciales:

$$\begin{aligned} x' &= v \\ v' &= f(x, v, t) \end{aligned} \quad (3.1.22)$$

y del esquema:

$$\begin{aligned}
k_1 &= hv_n \\
l_1 &= hf(x_n, v_n, t_n) \\
k_2 &= h(v_n + \frac{1}{2}l_1) \\
l_2 &= hf(x_n + \frac{1}{2}k_1, v_n + \frac{1}{2}l_1, t_n + \frac{h}{2}) \\
k_3 &= h(v_n - l_1 + 2l_2) \\
l_3 &= hf(x_n - k_1 + 2k_2, v_n - l_1 + 2l_2, t_n + h) \\
v_{n+1} &= v_n + \frac{1}{6}[l_1 + 4l_2 + l_3] \\
x_{n+1} &= x_n + \frac{1}{6}[k_1 + 4k_2 + k_3]
\end{aligned} \tag{3.1.23}$$

de modo que se tiene un total de tres llamadas a la función  $f$ , y un número considerable de operaciones básicas.

Otra solución para manejar fuerzas dependientes de la velocidad es la presentada por C. Scherer<sup>15</sup> consiste en calcular la velocidad mediante el método de Euler teniendo:

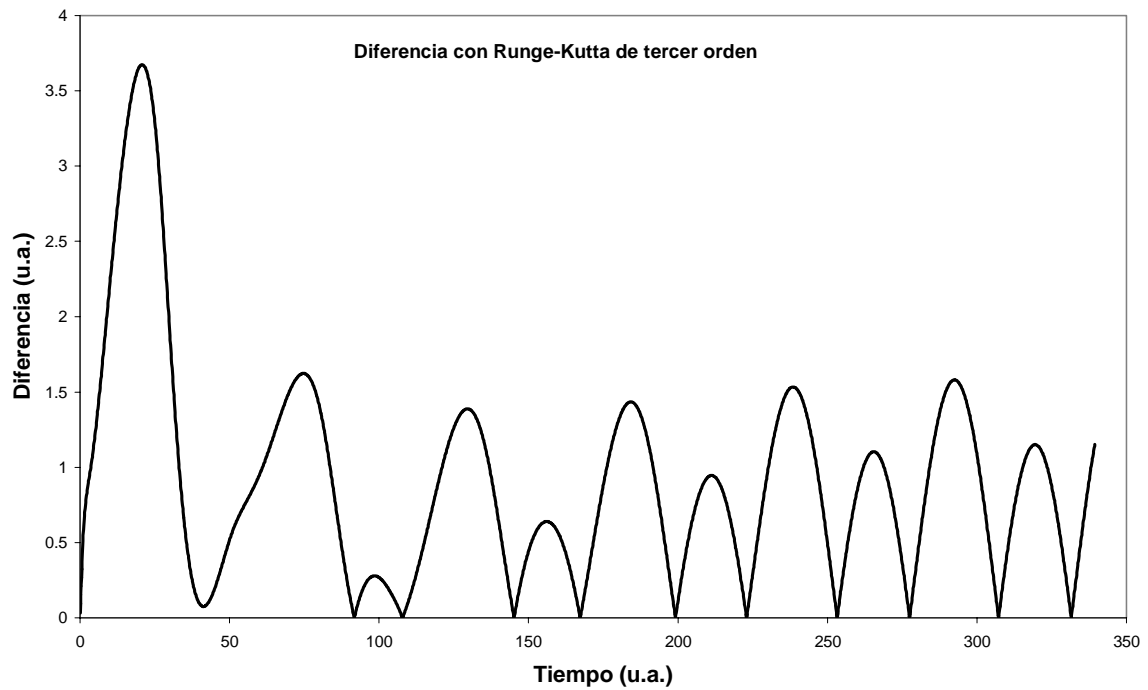
$$\begin{aligned}
v_{n+1} &= v_n + h \left[ a_n^x - \frac{\alpha}{m} v_n - \frac{\bar{\beta}}{m} v_n^2 \right] \\
x_{n+1} &= 2x_n - x_{n-1} + h^2 \left[ a_n^x - \frac{\alpha}{m} v_n - \frac{\bar{\beta}}{m} v_n^2 \right],
\end{aligned} \tag{3.1.24}$$

de modo que para el cálculo de la posición al tiempo  $n+1$ , sólo se necesita conocer la posición en los dos tiempos anteriores y la velocidad al tiempo anterior. Implementando este algoritmo para el problema anterior, y nuevamente calculando la diferencia de la posición en cada tiempo con lo obtenido del Método de Runge-Kutta de tercer orden, se obtuvo la gráfica siguiente, dónde pueden verse las fluctuaciones de la diferencia, las cuales son en promedio del orden de 0.9 u.a.

---

<sup>15</sup> <http://pcleon.if.ufrgs.br/~leon/metcomp/ode/node10.html>





De lo anterior se tiene que la extensión al algoritmo de Verlet propuesta en el presente trabajo es mucho más fiel a los resultados obtenidos con un método tan utilizado como es el de Runge-Kutta, respecto al expuesto en la Ref. 15, con la ventaja de que nuestro algoritmo sólo requiere una llamada a la función que describe a la fuerza involucrada en el sistema. Esto último es importante, ya que dependiendo de la aplicación pueden aparecer expresiones complicadas, y en algunos casos, su pueden no tener una solución analítica, de modo que en cada iteración se tenga que emplear algún método numérico, de modo que el número de llamadas a la función puede ser determinante para que la solución numérica de un problema sea o no factible.

### 3.2 GENERALIZACIÓN A DOS O MÁS DIMENSIONES

El planteamiento anterior demostró ser útil y eficiente para problemas que involucran una dimensión, sin embargo, la extensión a dos y tres dimensiones no es obvia, ya que en el análisis aparecen ecuaciones acopladas como se verá a continuación.

La velocidad en el tiempo  $t$  se calcula para cada componente de la siguiente manera:

$$\begin{aligned} v_x &= \frac{x(t+h) - x(t-h)}{2h} \\ v_y &= \frac{y(t+h) - y(t-h)}{2h} \\ &\vdots \end{aligned} \quad (3.2.1)$$

de modo que  $|\underline{v}| = v = \sqrt{v_x^2 + v_y^2 + v_z^2}$ .

Tenemos ahora que la fuerza en el sistema

$$\underline{F} = \underline{F}^r + \underline{F}^v, \quad (3.2.2)$$

donde  $\underline{F}^r$  se refiere al término de la fuerza dependiente del vector de posición  $\underline{r} = (x\hat{i} + y\hat{j} + z\hat{k})$ , y la fuerza de fricción  $\underline{F}^v = -\alpha\underline{v} - \beta\underline{v}^2$  se vería como:

$$\begin{aligned} \underline{F}^v &= (-\alpha - \beta v)(v_x\hat{i} + v_y\hat{j} + v_z\hat{k}) \\ &= [-\alpha v_x - \beta v_x v]\hat{i} + [-\alpha v_y - \beta v_y v]\hat{j} + [-\alpha v_z - \beta v_z v]\hat{k}, \end{aligned} \quad (3.2.3)$$

y retomando el desarrollo del algoritmo extendido de Verlet:

$$x(t+h) = 2x(t) - x(t-h) + \frac{F_x^r(t)}{m} h^2 + \frac{F_x^v(t)}{m} h^2, \quad (3.2.4)$$

esto es:

$$\begin{aligned} x(t+h) &= 2x(t) - x(t-h) + a_x^r(t)h^2 - \alpha \frac{v_x}{m} h^2 - \beta \frac{v_x v}{m} h^2 \\ &= 2x(t) - x(t-h) + a_x^r(t)h^2 + \frac{v_x}{m} h^2 \{-\alpha - \beta v\} \end{aligned} \quad (3.2.5)$$

Entonces, sustituyendo:

$$\begin{aligned} x(t+h) &= 2x(t) - x(t-h) + a_x^r(t)h^2 + \frac{h}{2m} [x(t+h) - x(t-h)] \\ &\quad \times \left\{ -\alpha - \beta \sqrt{[x(t+h) - x(t-h)]^2 + [y(t+h) - y(t-h)]^2 + [z(t+h) - z(t-h)]^2} \right\} \end{aligned} \quad (3.2.6)$$

y análogamente:

$$\begin{aligned} y(t+h) &= 2y(t) - y(t-h) + a_y^r(t)h^2 + \frac{h}{2m} [y(t+h) - y(t-h)] \\ &\quad \times \left\{ -\alpha - \beta \sqrt{[x(t+h) - x(t-h)]^2 + [y(t+h) - y(t-h)]^2 + [z(t+h) - z(t-h)]^2} \right\} \\ z(t+h) &= 2z(t) - z(t-h) + a_z^r(t)h^2 + \frac{h}{2m} [z(t+h) - z(t-h)] \\ &\quad \times \left\{ -\alpha - \beta \sqrt{[x(t+h) - x(t-h)]^2 + [y(t+h) - y(t-h)]^2 + [z(t+h) - z(t-h)]^2} \right\} \end{aligned} \quad (3.2.7)$$

Como puede verse, la solución de este sistema de ecuaciones involucra términos acoplados, de modo que se tendría que resolver una ecuación de cuarto orden para dos dimensiones y de sexto orden para tres dimensiones, lo cual hace que la complejidad del algoritmo sea inaceptable, sin embargo, es posible emplear el modelo para la fuerza de fricción

considerando solamente la parte lineal de esta, es decir  $\underline{F}^v = -\alpha \underline{v}$ . En este caso, el desarrollo es muy simple como se verá a continuación:

Ya que  $\underline{F}^v = -\alpha \underline{v}$ , entonces se puede dividir en componentes como sigue:

$$\begin{aligned} F_x^v &= -\alpha v_x \\ F_y^v &= -\alpha v_y \\ &\vdots \end{aligned} \quad (3.2.8)$$

Y se tenía de la Ec (3.2.4) que:

$$x_{n+1} = 2x_n - x_{n-1} + \frac{F_x^x}{m} h^2 + \frac{F_x^v}{m} h^2, \quad (3.2.9)$$

entonces, sustituyendo:

$$x(t+h) = 2x(t) - x(t-h) + a_x(t)h^2 - \frac{\alpha h}{2m} [x(t+h) - x(t-h)] \quad (3.2.10)$$

entonces:

$$2x(t) + x(t-h) \left[ \frac{\alpha h}{2m} - 1 \right] + a_x(t)h^2 + x(t+h) \left[ -\frac{\alpha h}{2m} - 1 \right] = 0, \quad (3.2.11)$$

y finalmente:

$$x(t+h) = \frac{2m}{\alpha h + 2m} \left[ 2x(t) + x(t-h) \left[ \frac{\alpha h}{2m} - 1 \right] + a_x(t)h^2 \right]. \quad (3.2.12)$$

Es claro que lo anterior se extiende a  $n$  dimensiones haciendo lo propio para cada entrada del vector de posición obteniendo así un conjunto de ecuaciones no acopladas:

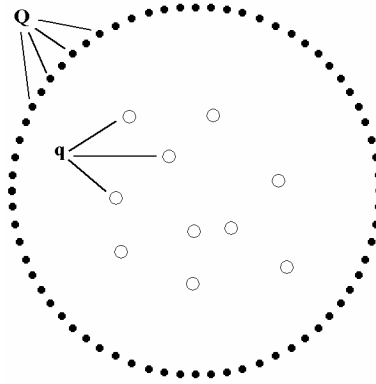
$$\begin{aligned}x(t+h) &= \frac{2m}{\alpha h + 2m} \left[ 2x(t) + x(t-h) \left[ \frac{\alpha h}{2m} - 1 \right] + a_x(t) h^2 \right] \\y(t+h) &= \frac{2m}{\alpha h + 2m} \left[ 2y(t) + y(t-h) \left[ \frac{\alpha h}{2m} - 1 \right] + a_y(t) h^2 \right] \\z(t+h) &= \frac{2m}{\alpha h + 2m} \left[ 2z(t) + z(t-h) \left[ \frac{\alpha h}{2m} - 1 \right] + a_z(t) h^2 \right].\end{aligned}$$

Dada la sencillez de este esquema, es posible introducir perturbaciones en la velocidad durante la simulación, teniendo así la posibilidad de enriquecer la simulación mediante la inclusión de modelos de motores de propulsión, corrientes de viento, etc. Lo anterior se logra simplemente retomando las condiciones del sistema al tiempo  $t$ , planteando con esta información y el efecto de la perturbación sobre la velocidad del móvil, nuevas condiciones para reinicializar el algoritmo sin perder la continuidad de la simulación.

### 3.3 APLICACIONES DEL ALGORITMO EXTENDIDO DE VERLET

#### 3.3.1 MOVIMIENTO DE VARIOS CUERPOS CON INTERACCIÓN ELÉCTROSTÁTICA EN UN MEDIO RESISTIVO.

Un problema interesante tratado anteriormente con un método alternativo propio por el Dr. Enrique Cabrera y el Dr. Héctor Riveros<sup>16</sup> del instituto de Física de la UNAM, consiste en un grupo de partículas cargadas eléctricamente con carga  $q$  que se mueven en un medio resistivo, y que están dentro de un círculo formado por un cierto número de partículas fijas cargadas con carga  $Q$  y considerando que  $q$  y  $Q$  tienen el mismo signo.



Entonces, tenemos que la componente de la fuerza dependiente de la posición es análoga al caso gravitacional y está dada por:

$$\underline{E}_i = K \left[ q^2 \sum_{j \neq i} \frac{1}{r_j^2} \hat{r}_j + qQ \sum_n \frac{1}{r_n^2} \hat{r}_n \right] , \quad (3.3.1)$$

---

<sup>16</sup> Simulación de nucleación analógica y digital, H.G. Riveros y Enrique Cabrera, Congreso de la Sociedad Mexicana de Cristalografía, del 24 al 28 de Noviembre de 1997, SLP, México.

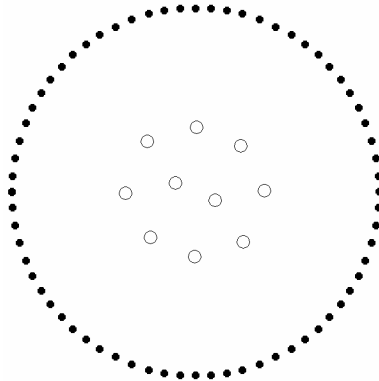
donde  $K$  es una constate de ajuste de unidades. La fuerza de fricción (dependiente de la velocidad) tendrá los componentes:

$$\begin{aligned} F_x^v &= -\alpha v_x \\ F_y^v &= -\alpha v_y \end{aligned} \quad (3.3.2)$$

De modo que el esquema del algoritmo de Verlet Extendido quedaría:

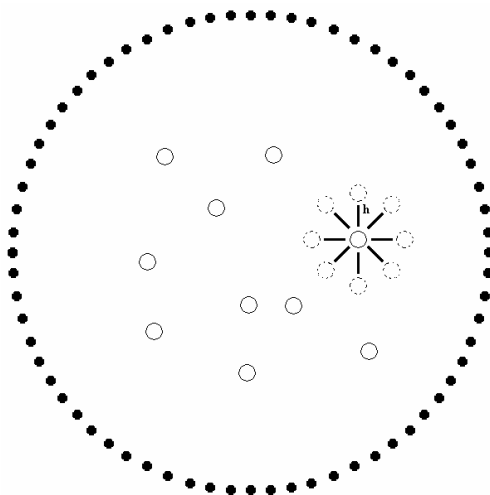
$$\begin{aligned} x(t+h) &= \frac{2m}{\alpha h + 2m} \left\{ 2x(t) + x(t-h) \left[ \frac{\alpha h}{2m} - 1 \right] + \frac{Kh^2}{m} \left[ q^2 \sum_{j \neq i} \frac{1}{r_j^2} \cos(\theta_j) + qQ \sum_n \frac{1}{r_n^2} \cos(\theta_n) \right] \right\} \\ y(t+h) &= \frac{2m}{\alpha h + 2m} \left\{ 2y(t) + y(t-h) \left[ \frac{\alpha h}{2m} - 1 \right] + \frac{Kh^2}{m} \left[ q^2 \sum_{j \neq i} \frac{1}{r_j^2} \text{sen}(\theta_j) + qQ \sum_n \frac{1}{r_n^2} \text{sen}(\theta_n) \right] \right\} \end{aligned} \quad (3.3.3)$$

Así pues, se implementó un programa empleando el esquema anterior, el cual realiza una animación que representa el movimiento de las partículas interiores desde su posición aleatoria inicial hacia su configuración de mínima energía. En la siguiente figura se muestra un ejemplo de la configuración del sistema estabilizado dada por la simulación.



Los resultados obtenidos mediante este programa fueron validados mediante un método alternativo que llamaremos *MER* (*Método de Relajación en Energía*) para encontrar las configuraciones de mínima energía, el cual se describe a continuación:

A cada partícula, de manera secuencial, se le calculan ocho “*posiciones de prueba*” equidistantes de la posición original, como se muestra en la figura. En cada una de estas posiciones se calcula la energía potencial del sistema, de este modo, es posible discernir en cual de ellas esta es menor, eligiéndola como la nueva posición de la partícula. Una vez hecho esto con cada una de las partículas del sistema, se almacena el valor de la energía potencial para la configuración obtenida para compararlo con su valor en la siguiente iteración, de este modo, cuando la diferencia sea menor que cierto umbral predefinido, podemos inferir que el sistema se encuentra en su configuración de mínima energía.



Los resultados de la comparación del método de Verlet extendido con el *MER* resultaron muy satisfactorios, ya que el valor final de la energía potencial no difiere sino hasta la cuarta o quinta cifra significativa (empleando unidades arbitrarias). Cabe mencionar que aunque ambos métodos sean consistentes en cuanto a la energía potencial del sistema una vez en reposo, las trayectorias que siguen las partículas para llegar a la configuración final difieren en gran medida, ya que en el caso del *MER* no se plantea una trayectoria física, sino que ésta está dada por la mecánica del algoritmo, sin embargo, usando el algoritmo

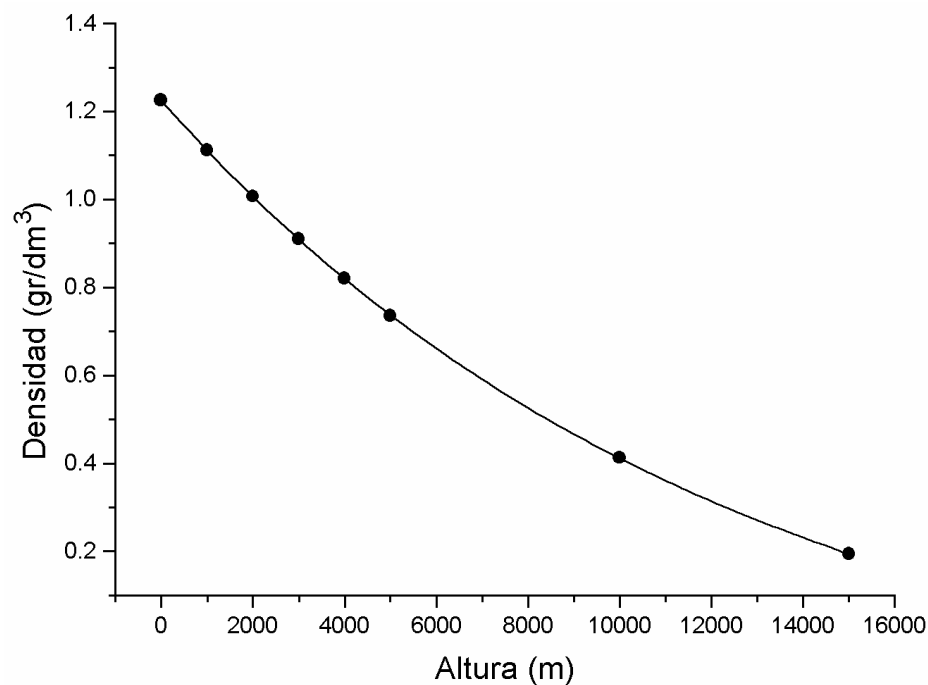


extendido de Verlet, las trayectorias son las que realmente seguirían las partículas de forma natural en un experimento realizado físicamente, ya que éstas forman la solución de la ecuación [diferencial] de movimiento.

En el programa se implementan simultáneamente ambos métodos, y en el caso del algoritmo de Verlet, es posible seleccionar algunos parámetros como el número de cargas y el coeficiente de fricción del medio.

### 3.3.2 INGRESO DE UN METEORITO A LA ATMÓSFERA TERRESTRE

Un problema interesante que puede tratarse mediante las herramientas que se han venido desarrollando en el presente trabajo es predecir la trayectoria de un meteorito al ingresar en la atmósfera terrestre. Lo anterior requiere plantear un modelo de la atmósfera para proponer una dependencia del coeficiente de fricción con la distancia a la superficie de la tierra (altura). En la figura siguiente, se muestra una serie de datos experimentales de la densidad del aire a diferentes alturas, así como un ajuste mediante una exponencial decreciente.



Por otro lado, sabemos que la fuerza de fricción en un fluido está dada como:

$$F_f = -K\eta v \quad (3.3.4)$$

y el coeficiente de fricción  $K$  depende de la forma del cuerpo, para una esfera, tenemos la relación conocida como la *ley de Stokes*:

$$K = 6\pi R, \quad (3.3.5)$$

donde  $R$  es el radio de la esfera. Ahora, el coeficiente de viscosidad  $\eta$  depende de la fuerza de fricción entre las diferentes capas del fluido que se muevan a diferentes velocidades, la cual depende a su vez de la temperatura. Es claro que en un fenómeno como el que se pretende simular la temperatura es algo difícil de tratar, debido a que entran en juego factores tales como la ionización del gas, la absorción de calor del meteorito, etc., de modo que en esta sección trabajaremos con un modelo simplificado, tomando un coeficiente de viscosidad proporcional únicamente a la densidad de la atmósfera, ya que el modelo que describe completamente a este fenómeno escapa de los propósitos del presente trabajo.

La expresión correspondiente al ajuste de la densidad de la atmósfera es la siguiente:

$$\rho_A = -0.24026 + 1.46634 \times \exp\left\{-\frac{y}{12337.90643}\right\}, \quad (3.3.6)$$

donde  $y$  está dada en metros. El coeficiente de viscosidad del aire a 40°C y a nivel del mar es 1.9 Poises, lo cual, en el sistema MKS equivale a 0.19 kg/(m·s), entonces, si consideramos que la viscosidad será directamente proporcional a la densidad, podemos obtener un modelo para la viscosidad  $\eta$  normalizando la expresión para  $\rho_A$  de modo que tome este último valor para distancias muy cercanas a la superficie, así la expresión para  $\eta$  sería

$$\eta = -0.0372331 + 0.227239 \times \exp\left\{-\frac{y}{12337.90643}\right\}, \quad (3.3.7)$$

entonces el coeficiente de fricción  $\alpha$  en las unidades propias del sistema MKS (*i.e.* kg/s) es:

$$\alpha(y) = 6\pi R \times \left[ -0.0372331 + 0.227239 \times \exp\left\{-\frac{y}{12337.90643}\right\} \right], \quad (3.3.8)$$

con  $R$  y  $y$  en metros.

Para aplicar lo anterior propongamos un problema en particular; recientemente, los astrónomos del Instituto Tecnológico de Massachusetts descubrieron un asteroide de 2.06 km de diámetro en una órbita tal que en su momento se consideró que tendría probabilidades de colisionar con la Tierra el 1 de abril de 2019. A diferencia de la mayoría de los asteroides, los cuales orbitan alrededor del Sol en el plano de los planetas, el 2002NT7 sigue una trayectoria que está inclinada 42 grados, de modo que pasa la mayor parte del tiempo arriba o abajo del resto del sistema solar, sin embargo, cada 2.3 años, el asteroide pasa a través del sistema solar, cerca de la órbita terrestre. A continuación implementaremos una simulación de este evento de este tipo empleando la expresión anterior para el coeficiente de fricción en el Algoritmo Extendido de Verlet en un sistema bidimensional.

La expresión para la fuerza, tomando en cuenta el Principio de Arquímedes, sería la siguiente:

$$\begin{aligned} F_y &= -g(m - \rho_A V) - \alpha(y)v_y \\ F_x &= -\alpha(y)v_x \end{aligned}, \quad (3.3.9)$$

donde  $y$  es la altura,  $v_x$ ,  $v_y$  son las componentes de la velocidad del móvil, y  $\alpha(y)$ . De acuerdo con lo anterior, el esquema del Algoritmo Extendido de Verlet sería el siguiente:

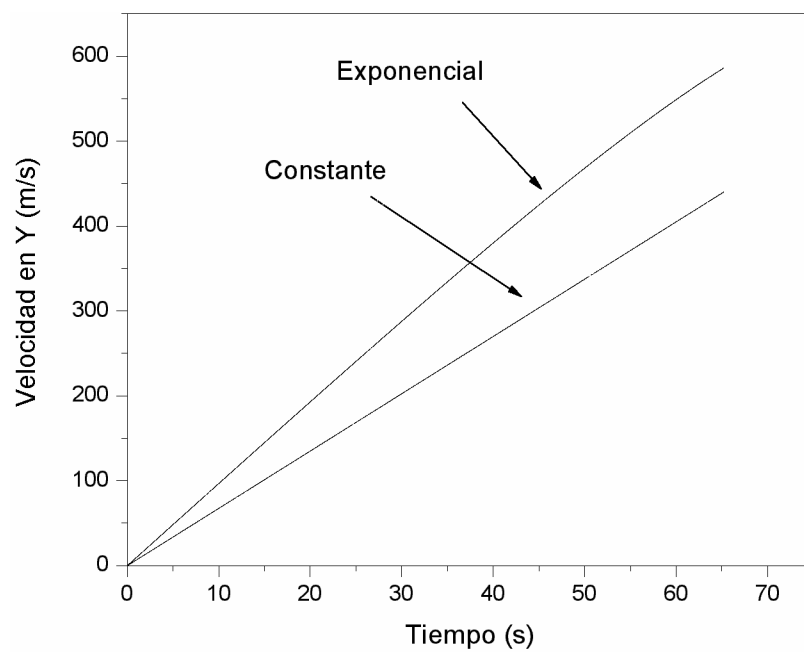
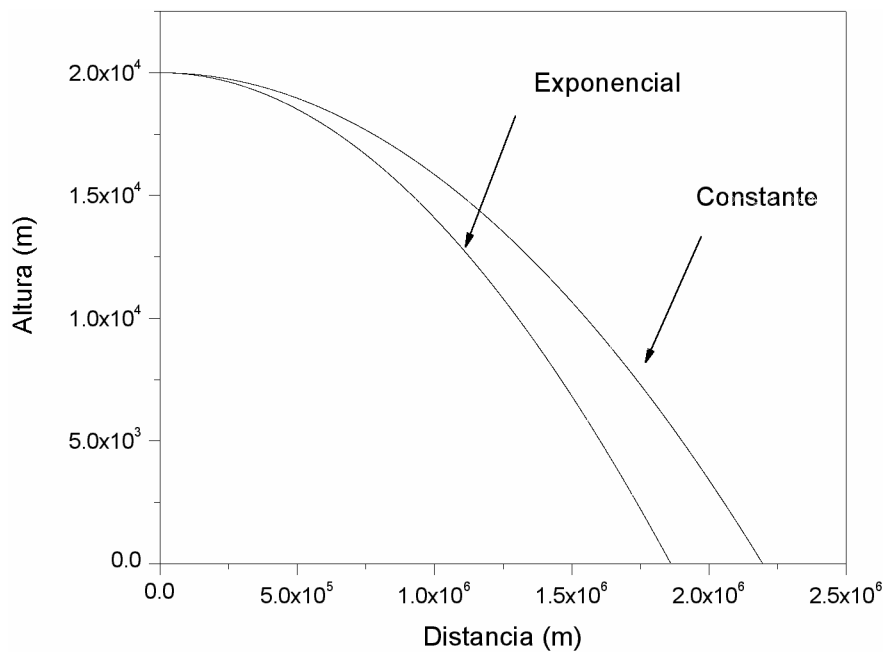
$$\begin{aligned}
x(t+h) &= \frac{2m}{\alpha(y)h + 2m} \left\{ 2x(t) + x(t-h) \left[ \frac{\alpha(y)h}{2m} - 1 \right] \right\} \\
y(t+h) &= \frac{2m}{\alpha(y)h + 2m} \left\{ 2y(t) + y(t-h) \left[ \frac{\alpha(y)h}{2m} - 1 \right] + h^2 \left[ -g \left( 1 - \frac{\rho_A V}{m} \right) \right] \right\} .
\end{aligned} \tag{3.3.10}$$

Así, pues, se implementó un programa, dónde es posible seleccionar el modelo para la densidad del aire, por tanto para la fricción, exponencial o constante, pudiendo así comparar las trayectorias y las velocidades en ambos casos, como se puede apreciar en las figuras siguientes, donde se consideró una velocidad inicial en  $x$  de 28.52 km/s. (102,672 km/h) una masa de  $1.2 \times 10^{13}$  kg. y un radio de 9000 m.

El programa considera un ingreso tangente a la superficie de la tierra, lo cual equivale a un tiro horizontal, no obstante el ángulo de ingreso puede ser estimado mediante un procedimiento análogo al expuesto en la sección 2.3, dónde la trayectoria del móvil es afectada por un cierto número de masas en movimiento y atraída gravitacionalmente por una en particular en un cierto ángulo de entrada, de este modo, puede sofisticarse el problema al considerar la rotación de la tierra, de modo que el lugar geométrico de ingreso a la atmósfera sea conocido, y a partir del instante en que se considere que se hizo contacto con la atmósfera, sea posible tomar los datos de ángulo y velocidad, y con estas condiciones inicializar el programa mencionado dando continuidad a la simulación. Así pues, sería posible predecir el lugar de colisión sobre la superficie del planeta.

Como en algunas de las aplicaciones anteriores, fue posible comprobar el desempeño del programa mediante el análisis de un problema conocido que arroje un resultado calculable analíticamente, en este caso, la velocidad terminal en un medio con densidad constante, la cual, de acuerdo con el modelo depende tanto de la masa del móvil como con sus dimensiones, de la siguiente manera:

$$v_t = g \frac{m - m_f}{K\eta}, \tag{3.3.11}$$



donde  $m$  corresponde a la masa del móvil,  $m_f$  a la masa del gas desplazado,  $g$  a la aceleración de la gravedad,  $K\eta$  será calculado de acuerdo a las ecuaciones (3.3.5) y (3.3.7) considerando una densidad atmosférica constante. De acuerdo con lo anterior, se calculó la velocidad terminal de un objeto esférico en caída libre de 1m de radio y con una masa de 30 Kg. Dando como resultado un valor de 68.176123  $m/s$ . Implementando la simulación mediante el algoritmo de Verlet extendido, bajo las mismas condiciones del cálculo, se llega a una velocidad terminal de 68.10636  $m/s$ ., teniendo una diferencia porcentual del 0.01% aproximadamente.

### 3.4 VALIDACIÓN DEL ALGORITMO EXTENDIDO DE VERLET MEDIANTE SU APLICACIÓN EN UN PROBLEMA CON SOLUCIÓN ANALÍTICA

Consideremos ahora el problema del oscilador armónico amortiguado, el cual, tiene en su ecuación de movimiento un término dependiente de la velocidad en forma lineal:

$$\frac{d^2x}{dt^2} = -kx - b \frac{dx}{dt}, \quad (3.4.1)$$

donde se consideró masa unitaria. Esta ecuación diferencial tiene solución analítica, la cual está dada por<sup>17</sup>:

$$x(t) = e^{-\frac{bt}{2}} a \cos \left( t \sqrt{k - \frac{b^2}{4}} + \alpha \right), \quad (3.4.2)$$

donde

$$a = \sqrt{\frac{\omega_0 x_0^2}{\omega_0^2 - 1}}, \quad \tan \alpha = -\frac{1}{\sqrt{\omega_0^2 - \frac{b^2}{4}}}.$$

Con las expresiones anteriores, se calculó la posición del cuerpo para una sucesión de valores de  $t$ , asimismo, se plantearon las soluciones numéricas empleando el Método de Runge-Kutta de 3er Orden, el Método de Verlet – Scherer (Ec. 3.1.24), y la Extensión al Algoritmo de Verlet.

El esquema para el Método de Runge-Kutta de 3er orden para este problema en particular es el siguiente:

---

<sup>17</sup> V. Barger, M. Olsson, *Classical Mechanics: a Modern Perspective*, (McGraw-Hill, New York, 1973) p. 17



$$\begin{aligned}
k_1 &= h v_n \\
l_1 &= h [-k x_n - b v_n] \\
k_2 &= h \left[ v_n + \frac{1}{2} l_1 \right] \\
l_2 &= h \left[ -k \left( x_n + \frac{1}{2} k_1 \right) - b \left( v_n + \frac{1}{2} l_1 \right) \right] \\
k_3 &= h [v_n - l_1 + 2l_2] \\
l_3 &= h [-k (x_n - k_1 + 2k_2) - b (v_n - l_1 + 2l_2)]
\end{aligned} \tag{3.4.3}$$

$$\begin{aligned}
v_{n+1} &= v_n + \frac{1}{6} [l_1 + 4l_2 + l_3] \\
x_{n+1} &= x_n + \frac{1}{6} [k_1 + 4k_2 + k_3],
\end{aligned}$$

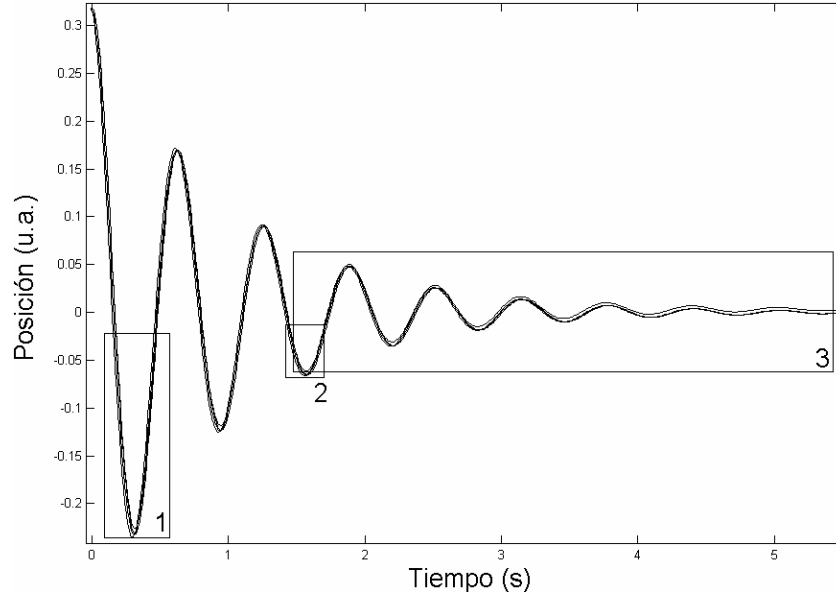
El esquema para el método de Verlet-Scherer tiene una forma más sencilla:

$$\begin{aligned}
v_n &= v_{n-1} + h [-k x - b v_{n-1}] \\
x_{n+1} &= 2x_n - x_{n-1} + h^2 [-k x - b v_n],
\end{aligned} \tag{3.4.4}$$

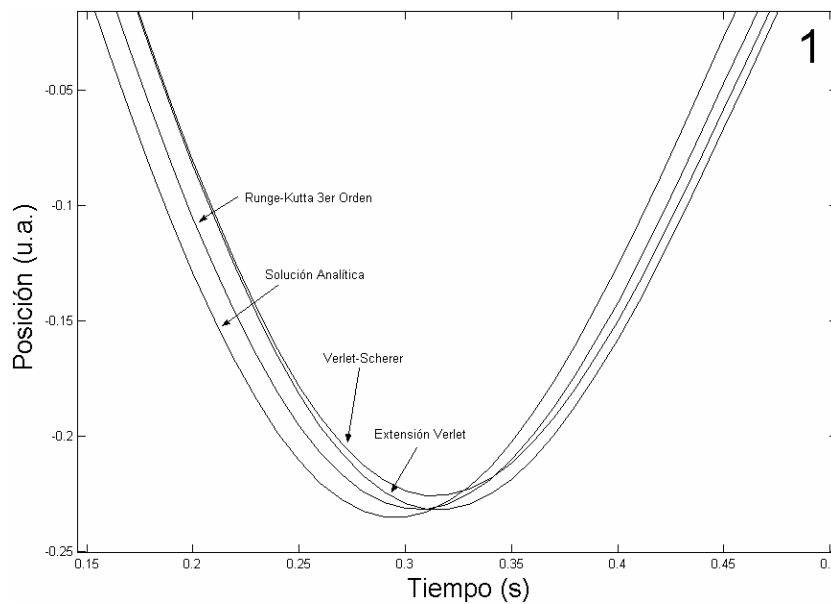
y finalmente, la Extensión al Algoritmo de Verlet:

$$x_{n+1} = \frac{2}{bh+2} \left[ 2x_n + x_{n-1} \left( \frac{bh}{2} - 1 \right) - kx_n h^2 \right]. \tag{3.4.5}$$

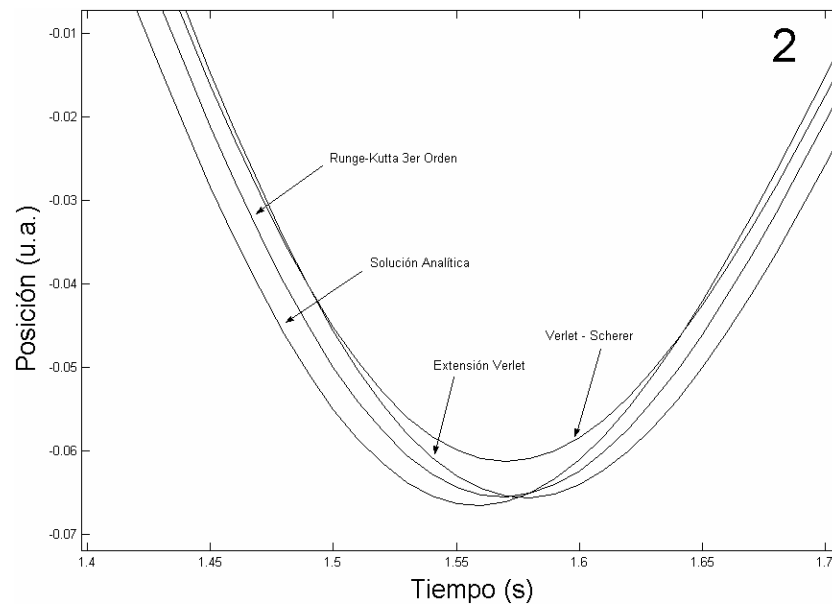
Implementando cada uno de los esquemas anteriores, usando los mismos parámetros en cada uno, incluso para la solución analítica, se generó la siguiente gráfica comparativa, donde cada recuadro numerado corresponde a una ampliación mostrada posteriormente:



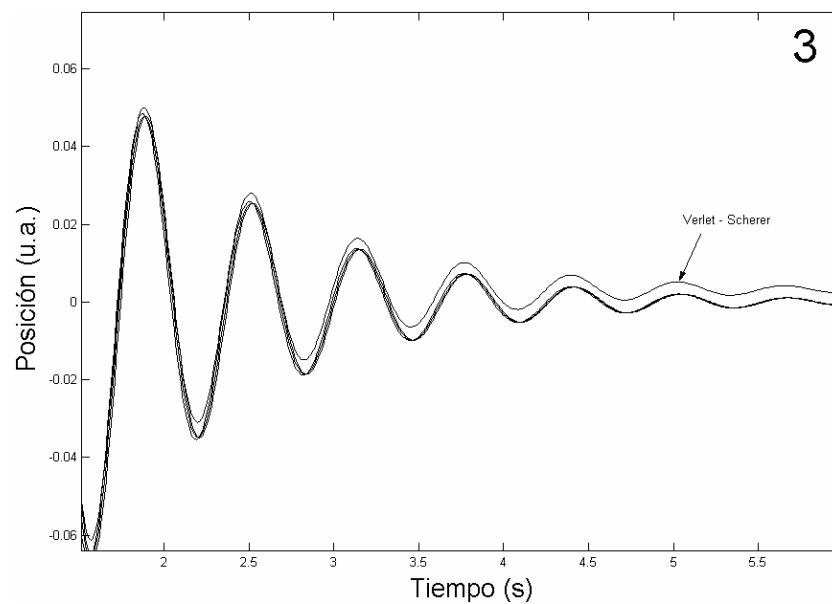
Como se aprecia en figura anterior, todas las soluciones tienen el mismo comportamiento cualitativo, sin embargo, como puede verse en el recuadro 1, existen diferencias notables. Observamos que la solución con Verlet–Scherer y con el Algoritmo Extendido, están prácticamente empalmadas al inicio de la simulación, dando valores ligeramente mayores a la solución analítica y aun a Runge-Kutta.



En el recuadro 2, se observa que la solución con Verlet-Scherer se aparta dramáticamente de las demás líneas, coincidiendo éstas en un punto.



En el recuadro 3 puede verse la evolución del sistema al acercarse al equilibrio, podemos notar que la diferencia de la solución con Verlet-Scherer difiere cada vez más, debido, quizá, a la acumulación del error en el cálculo de la velocidad al aplicar el método de Euler.



Es claro, de las ecuaciones (3.4.3)-(3.4.5), que la Extensión al Algoritmo de Verlet tiene ventajas en cuanto a simplicidad algebraica y computacional, y su desempeño es muy aceptable al compararlo con un método tan aceptado como es el de Runge-Kutta.

#### 4. CONCLUSIONES

Se encontró que el Algoritmo Clásico de Verlet es útil para aplicaciones que involucran dimensiones desde microscópicas hasta astronómicas, las cuales, hasta nuestro conocimiento, no han sido reportadas. El método clásico de Verlet es tan preciso como el Método de Runge-Kutta de 3<sup>er</sup> orden además de que emplea una tercera parte de las llamadas a la función.

Se desarrolló un nuevo algoritmo para la solución de ecuaciones diferenciales de segundo orden con dependencia con la primera derivada lineal y cuadráticamente, el cual resultó ser equivalente en su desempeño a los métodos más utilizados, con la ventaja de ser más eficiente en cuanto a su simplicidad algebraica, además de requerir menos localidades de memoria simultáneas en su implementación. También se evaluó una corrección al algoritmo de Verlet para introducir fuerzas dependientes de la velocidad, propuesta por C. Scherer [Ref. 15], cuyo desempeño no fue satisfactorio tanto en las posiciones como en las velocidades comparado con la solución exacta, de Runge-Kutta y la corrección aquí propuesta.

Pensamos que la Extensión al Algoritmo de Verlet presentada en el presente trabajo puede enriquecer el área de la Computación Científica, así como ampliar las posibilidades de aplicación del análisis numérico dentro de la Física Computacional, con las ventajas de simplicidad y confiabilidad del Algoritmo Clásico de Verlet, siendo lo suficientemente versátil como para aceptar modelos más completos de las fuerzas que rigen los fenómenos a estudiar pudiendo ser considerablemente complejos, de modo que la utilidad del nuevo algoritmo estará limitada únicamente por el nivel de confiabilidad y complejidad de dicho modelo.

En resumen:

- El algoritmo modificado preserva la esencia del algoritmo clásico de Verlet.
- Las llamadas a memoria son esencialmente las mismas.

- La confiabilidad en la estabilidad de la Energía se preserva a diferencia de los otros métodos reportados.
- La sencillez de la programación es la misma.
- Otras alternativas al Algoritmo de Verlet par cuando se usan fuerzas dependientes de la velocidad no son tan confiables.
- La aplicación del Algoritmo Clásico y del modificado se puede hacer para dimensiones del tamaño de moléculas hasta dimensiones planetarias.
- Las animaciones presentadas son pequeños experimentos muy versátiles en un Laboratorio Virtual que permiten “observar” y hacer “mediciones” directamente de fenómenos físicos difíciles de implementar en un laboratorio real.

La CC como parte de la Ciencia e Ingeniería de la Computación es una herramienta que va a permitir, aún más, el gran desarrollo de las otras Ciencias y de la Tecnología de los países que quieran y sepan aprovecharla.