

# Métodos numéricos en el problema de valores iniciales en Ecuaciones Diferenciales Ordinarias

Yago Pego Martínez ([yago.pegomartinez@alumnos.upm.es](mailto:yago.pegomartinez@alumnos.upm.es))

Evaristo de Vega Galindo ([evaristo.devega.galindo@alumnos.upm.es](mailto:evaristo.devega.galindo@alumnos.upm.es))

## MARCO TEÓRICO

En matemáticas, una ecuación diferencial ordinaria, comúnmente abreviada a EDO, es la ecuación diferencial que relaciona una función desconocida de una sola variable independiente con una o más de sus derivadas respecto a dicha variable.

Sirva como ejemplo el posiblemente más conocido, la segunda ley de Newton:

$$m * \frac{d^2 u(t)}{dt^2} = F\left(t, u(t), \frac{du(t)}{dt}\right)$$

En general, una ecuación diferencial lineal de orden  $s$  se puede formular como:

$$a_s(t)y^{(s)} + a_{s-1}(t)y^{(s-1)} + \dots + a_1(t)y' + a_0(t)y = g(t),$$

donde  $a_i$  es una función dependiente de  $t$ .

Para el caso anterior, la ley de Newton, se observa que es una ecuación diferencial de segundo orden, pues su “mayor” derivada es, de hecho, la segunda.

En general, si no se especifican ciertas condiciones iniciales que deba satisfacer la solución, entonces no existirá una única solución. Concretamente, para obtener una función única en una ecuación diferencial de orden  $s$ , serán necesarias  $s$  condiciones de contorno. En este caso, estaremos intentando resolver un problema de Cauchy.

Plantear la resolución de estos problemas puede resultar sencillo analíticamente, pues toda EDO de orden  $s$  se puede transformar, por un cambio de variable, en un sistema de  $s$  ecuaciones.

Sea la ecuación:

$$x'' + \alpha * x' + \beta * x = f(t),$$

proponemos el siguiente sistema:

$$U(t) = \begin{pmatrix} u_1(t) \\ u_2(t) \end{pmatrix} = \begin{pmatrix} x(t) \\ x'(t) \end{pmatrix}$$

y, derivando, obtenemos la expresión:

$$\frac{dU(t)}{dt} = \begin{pmatrix} u_1'(t) \\ u_2'(t) \end{pmatrix} = \begin{pmatrix} x'(t) \\ x''(t) \end{pmatrix} = \begin{pmatrix} u_2(t) \\ f(t) - \alpha * u_2(t) - \beta * u_1(t) \end{pmatrix} .$$

Con vistas a obtener las soluciones al problema planteado, nos remitimos ahora a operaciones computacionales. Presentamos aquí tres métodos distintos que nos han permitido trabajar de forma adecuada con un problema de movimiento armónico simple, que enunciaremos próximamente.

## MÉTODOS NUMÉRICOS

- Método de Euler

Llamado así en honor al matemático suizo Leonard Euler, el método homónimo es un procedimiento de integración numérica para resolver ecuaciones diferenciales ordinarias a partir de un valor inicial dado.

Existen dos formas de diferentes de plantear el problema:

- El método explícito, en que, en orden ascendente o natural, se puede obtener el valor de la función original en sucesivos puntos. Habiendo dividido un intervalo  $[a, b]$  en partes equiespaciadas, los resultados se obtendrán a partir de la siguiente ecuación:

$$u_{N+1} = U_N + \Delta t * F_N(U_N, t) ,$$

- donde  $U_N$  es un dato conocido, ya sea condición de contorno (primer paso) u obtenido en el paso anterior;
- $\Delta t$  es el incremento utilizado en el problema;
- $F_N$  es la función derivada del problema aplicada con las condiciones referidas.

```
subroutine Euler_Explicito(dt, t, U, F)
  real(8), intent(in) :: dt
  real(8), intent(in) :: t
  real(8), dimension(:), intent(inout) :: U

  interface
    function F(U, t)
      real(8), dimension(:), intent(in) :: U
      real(8), intent(in) :: t
      real(8), dimension(size(U)) :: F
    end function
  end interface
  real(8), dimension(size(U)) :: U_0

  U_0 = U

  U = U_0 + dt * F(U_0, t)

end subroutine Euler_Explicito
```

He aquí la subrutina empleada.

El tiempo y el incremento son variables de entrada (input) y la U es “transformable”.

La variable F se tratará desde un módulo externo. Por eso se utiliza la función *interface*.

- El método implícito posibilita calcular el valor de la función en puntos sucesivos de manera descendente. Es el método de Euler “al revés”. La dificultad del programa aumenta considerablemente, pues será necesarios el cálculo de raíces en los sucesivos pasos (método de Newton-Raphson), entre otras complicaciones. Faltos de tiempo, nos limitaremos a los restantes métodos.

(El código del método implícito ha sido particularizado para nuestro problema físico.  
Para acceder a éste, diríjase a la página 9)

### • Método de Adams-Bashforth

El método de Adams-Bashforth es un método lineal multipaso para la resolución numérica de EDO's. Se dice que es multipaso, a diferencia del método de Euler, porque, en cada paso, se refiere a más de un punto y derivada previos (a dos, concretamente).

Adams-Bashforth es un método explícito que sigue la siguiente fórmula:

$$U_{N+2} = U_{N+1} + \frac{\Delta t}{2} * (3 * F_{N+1} - F_N)$$

De la fórmula se deduce que será necesario conocer las derivadas de los puntos anteriores. Consecuentemente, el primer paso no será posible sin acudir a otro método. En los pasos restantes, se aplicará la fórmula sin problema.

Los *dummy arguments* utilizados no difieren del método anterior.

Alguna novedad respecto al método anterior es la introducción de la función *save\**, para guardar en cada paso el valor de  $F_N$ .

\*Es importante que esta variable (“F\_00” en el programa) sea de dimensión asignable, pues, en el caso contrario, sería un *automatic object* y no se le podría atribuir la función *save*.

```
subroutine Adams_Bashforth(dt, t, U, F)
  real(8), intent(in) :: dt
  real(8), intent(in) :: t
  real(8), dimension(:), intent(inout) :: U

  interface
    function F(U, t)
      real(8), dimension(:), intent(in) :: U
      real(8), intent(in) :: t
      real(8), dimension(size(U)) :: F
    end function
  end interface
  real(8), dimension(size(U)) :: U_0, F_0
  real(8), save, allocatable, dimension(:) :: F_00

  U_0 = U
  F_0 = F(U_0, t)

  if (t == dt) then
    ! Primer paso con Euler
    U = U_0 + dt * F_0
    allocate(F_00(size(U)))
  else
    ! Pasos restantes con Adams-Bashforth
    U = U_0 + (dt/2) * (3*F_0 - F_00)
  end if

  F_00 = F_0

end subroutine Adams_Bashforth
```

- **Método de Runge-Kutta**

En análisis numérico, los métodos de Runge-Kutta son un conjunto de procedimientos iterativos, implícitos y explícitos, para la resolución numérica de ecuaciones diferenciales. Fueron desarrollados a principios del siglo pasado por los matemáticos alemanes Carl Runge y Martin W. Kutta.

La expresión utilizada para el método de Runge-Kutta dependerá especialmente del orden de la ecuación. En el caso que nos ocupa, esta es de segundo grado y nos permitirá calcular función y derivada en diversos puntos:

$$U_{N+1} = U_N + \frac{\Delta t}{2}(k_1 + k_2)$$

donde  $k_1 = F(U_N, t_N)$

$k_2 = F(U_N + \Delta t * k_1, t_N + \Delta t)$

Se dice que el procedimiento de Runge-Kutta es un método por etapas, sean éstas  $t_N$  y  $t_N + \Delta t$ .

```
subroutine Runge_Kutta(dt, t, U, F)
  real(8), intent(in) :: dt
  real(8), intent(in) :: t
  real(8), dimension(:), intent(inout) :: U

  interface
    function F(U,t)
      real(8), dimension(:), intent(in) :: U
      real(8), intent(in) :: t
      real(8), dimension(size(U)) :: F
    end function
  end interface
  real(8), dimension(size(U)) :: U_0
  real(8), dimension(size(U)) :: K1, K2

  U_0 = U
  K1 = F(U_0, t)
  K2 = F(U_0 + dt*K1, t + dt)

  U = U_0 + (dt/2) * (K1 + K2)

end subroutine Runge_Kutta
```

## **NUESTRO PROBLEMA FÍSICO**

En el presente hito, hemos planteado un problema de movimiento armónico simple (MAS, de ahora en adelante) sometido a las exigencias de la teoría relativista. Conociendo la posición y la velocidad iniciales del cuerpo oscilante, podremos determinar estos datos en distintos instantes del tiempo.

Antes de profundizar en el caso relativista, conviene hacer un inciso y recordar las bases del MAS. Según la mecánica clásica, el MAS es un

movimiento periódico, y vibratorio en ausencia de fricción, producido por la acción de una fuerza recuperadora que es directamente proporcional a la posición, y que queda descrito en función del tiempo por una función senoidal (seno o coseno). Una de las soluciones posibles a la segunda ley de Newton es la siguiente:

$$x(t) = A * \cos(\omega t + \phi_0)$$

- $x(t)$  es la posición;
- $A$  es la máxima amplitud;
- $\omega$  es la velocidad angular;
- $t$  es el tiempo;
- $\phi_0$  es la fase inicial (estado de oscilación en  $t=0$ ).

Obtener las expresiones para la velocidad y la aceleración resulta, pues, sencillo, derivando respecto al tiempo la expresión anterior:

$$\begin{aligned} x'(t) &= -A * \omega * \sin(\omega t + \phi_0) \\ x''(t) &= -A * \omega^2 * \cos(\omega t + \phi_0) \end{aligned}$$

En mecánica relativista, el análogo del movimiento armónico simple es un movimiento en el que la fuerza es proporcional a la elongación, pero, debido a las peculiaridades de la teoría de la relatividad, el movimiento resultante es sólo cuasiarmónico.

El problema del oscilador en mecánica relativista es que no admite una solución analítica simple, pues la ecuación del movimiento implica integrar la siguiente ecuación:

$$\frac{d^2x}{dt^2} \left[ 1 - \left( \frac{1}{c} * \frac{dx}{dt} \right)^2 \right]^{-3/2} = -\frac{k}{m} * x$$

- $c$  es la velocidad de la luz;
- $k$  es la constante elástica del muelle;
- $m$  es la masa del cuerpo oscilante.

Sin embargo, puede darse una solución aproximada con las condiciones de contorno  $x(0) = 0$  y  $x'(0) = v_0$ , aproximada por:

$$x(t) = \frac{c}{\omega_2(B)} * \arcsin \left[ \frac{v_0}{c} * \sin(\omega_2 t) \right]$$

donde:

$$B = \frac{v_0}{\sqrt{c^2 - v_0^2}}$$

y:

$$\omega_2(B) = \omega \sqrt{\frac{(256 + 312B^2 + 83B^4)\sqrt{4 + 3B^2}}{512 + 1008B^2 + 620B^4 + 114B^6}}$$

En el problema presentado, se introduce por teclado el instante del tiempo en que se desea conocer la posición del cuerpo en MAS, al igual que el “incremento temporal” ( $\Delta t$ ) empleado en el problema. La velocidad inicial  $v_0$  (desde el programa principal), la masa del objeto y la constante elástica del resorte (desde el módulo-función) son otras variables que podremos modificar a nuestro antojo para conocer cómo éstas afectan al movimiento.

Retornando al comienzo del presente documento, se decía que toda EDO había de reducirse a un sistema de ecuaciones de grado igual al orden de la ecuación. Así lo hemos hecho en nuestro caso:

$$U(t) = \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} = \begin{pmatrix} x(t) \\ x'(t) \end{pmatrix}$$

y luego:

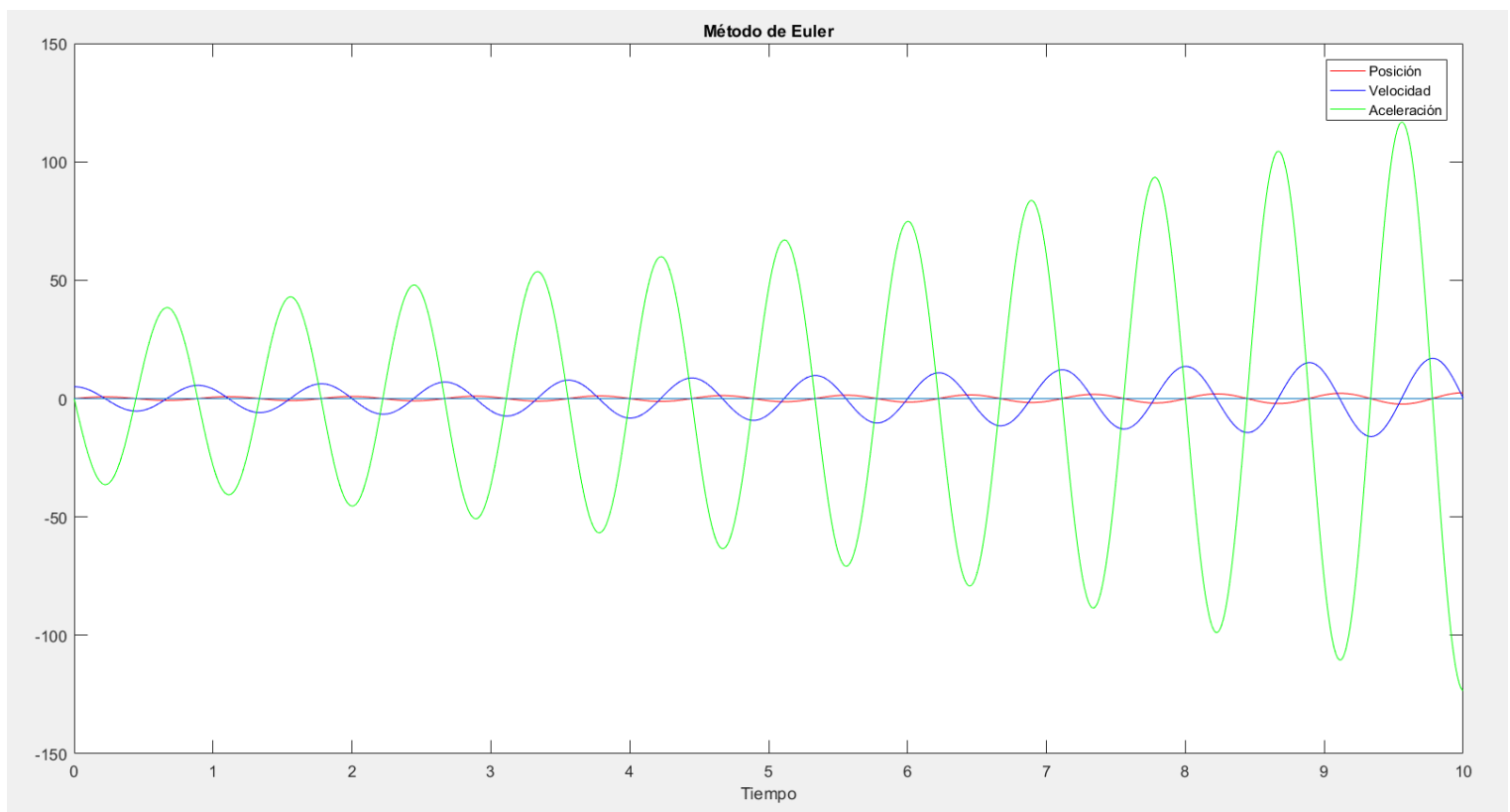
$$F(U, t) = \frac{dU(t)}{dt} = \begin{pmatrix} u'_1 \\ u'_2 \end{pmatrix} = \begin{pmatrix} x'(t) \\ x''(t) \end{pmatrix} = \begin{pmatrix} u_2 \\ -\frac{k}{m} * u_1 * \left(1 - \left(\frac{u_2}{c}\right)^2\right)^{\frac{3}{2}} \end{pmatrix}.$$

donde  $x''(t)$  se ha despejado de la ecuación diferencial original.

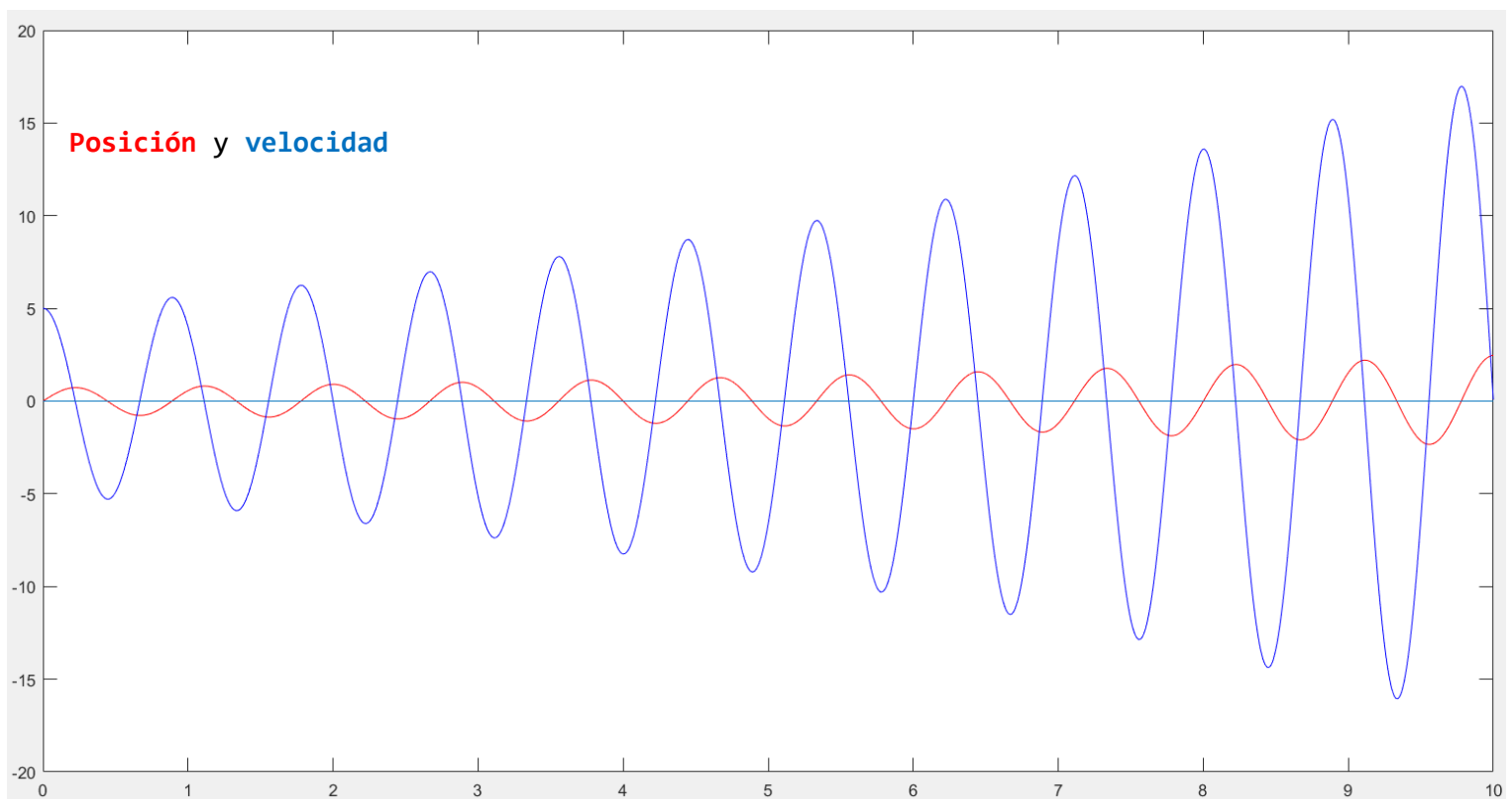
'Tiempo'	'Posicion'	'Velocidad'	'Aceleracion'
0.0000000000000000	0.0000000000000000	5.0000000000000000	0.0000000000000000
1.4999999999999999E-002	7.499999999999997E-002	4.9718749999999998	-3.7499999999999987
2.9999999999999999E-002	0.14915624999999999	4.8876582031249995	-7.4578124999999966
4.4999999999999998E-002	0.22163211914062497	4.7482979382324215	-11.081605957031245
5.9999999999999998E-002	0.29160990754394528	4.5553646729743953	-14.580495377197259
7.499999999999997E-002	0.35830007190862651	4.3110333160309553	-17.915003595431319
8.999999999999997E-002	0.42095013374460483	4.0180586996968115	-21.047506687230236
0.10500000000000000	0.47885316973774361	3.6797445192025635	-23.942658486887176
0.12000000000000000	0.53135578844600728	3.2999060789787413	-26.567789422300358
0.13500000000000001	0.57786550332067965	2.8828272659499805	-28.893275166033977
0.14999999999999999	0.61785741885375056	2.4332122350885022	-30.892870942687519
0.16499999999999998	0.65088015439902580	1.9561323521258165	-32.544007719951288
0.17999999999999999	0.67656093881241852	1.4569689918458395	-33.828046940620929
0.19500000000000001	0.69460981840928626	0.94135283715739271	-34.730490920464312
0.20999999999999999	0.70482293073809488	0.41510036364141772	-35.241146536904743
0.22499999999999998	0.70708480720731437	-0.11585177395763636	-35.354240360365722
0.23999999999999999	0.70136967855740873	-0.64551371313461048	-35.068483927870439
0.25500000000000000	0.68774176841850410	-1.1679099574162848	-34.387088420925203
0.27000000000000002	0.66635457160990574	-1.6771467902196961	-33.317728580495285
0.28499999999999998	0.63744912529130460	-2.1674787682321397	-31.872456264565230
0.29999999999999999	0.60135129243805896	-2.6333735441293125	-30.067564621902942

Intervalo [0, 0.3]  
 $\Delta t = 0.015 \text{ s}$   
 $m = 3 \text{ kg}$   
 $k = 150 \text{ N/m}$   
 $x_0 = 0 \text{ m}$   
 $v_0 = 5 \text{ m/s}$   
 $c = 3 \cdot 10^8 \text{ m/s}$

Tabla de resultados por el método Runge-Kutta.  
**Obsérvese** la aparición simultánea de la posición y aceleración máximas (en valor absoluto) en el mismo instante en que la velocidad se anula (condición básica del MAS).



Gráfica que relaciona el tiempo transcurrido con la posición, la velocidad y la aceleración de un cuerpo oscilante. Merece especial atención, como se escribía anteriormente, la oposición entre **velocidad** y **posición** y **aceleración**, pasando por el punto de equilibrio ( $x = 0$ ) y máximo relativo en el mismo instante.





El método Euler implícito, o *backwards Euler method* en inglés, recordemos, era el procedimiento de paso simple que permitía resolver EDOs según la fórmula:

$$U_{N+1} = U_N + \Delta t * F(U_{N+1}, t_{N+1}) ,$$

donde los únicos datos conocidos son  $U_N$  y  $\Delta t$ .

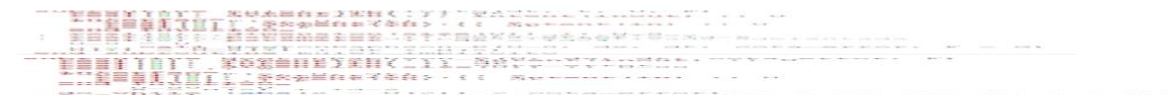
La resolución de este problema pasa por encontrar la posición y velocidad -  $U_{N+1} = \begin{pmatrix} x(t_{N+1}) \\ x'(t_{N+1}) \end{pmatrix}$  - que satisfagan en cada paso las ecuaciones resultantes. Para el problema que nos ocupa, tenemos:

$$\begin{pmatrix} U_1 \\ U_2 \end{pmatrix}^{(N+1)} = \begin{pmatrix} U_1 \\ U_2 \end{pmatrix}^{(N)} + \Delta t * \begin{pmatrix} U_2 \\ -\frac{k}{m} * U_1 * \left(1 - \left(\frac{U_2}{c}\right)^2\right)^{\frac{3}{2}} \end{pmatrix}^{(N+1)}$$

Despejando en la parte superior del sistema  $U_1^{(N+1)} = U_1^{(N)} + \Delta t * U_2^{(N+1)}$  y sustituyendo en la parte inferior, obtenemos una única ecuación de incógnita  $U_2^{(N+1)}$ :

$$U_2^{(N)} - \Delta t * \frac{k}{m} * \left( U_1^{(N)} + \Delta t * U_2^{(N+1)} \right) * \left( 1 - \left( \frac{U_2^{(N+1)}}{c} \right)^2 \right)^{\frac{3}{2}} - U_2^{(N+1)} = 0$$

Acudiendo al método de Newton-Raphson, obtenemos el valor de  $U_2^{(N+1)}$  que verifica la ecuación anterior. De forma trivial, hallamos también  $U_1^{(N+1)}$ . Se ha logrado ya conocer los valores de la posición y velocidad en el instante  $t_0 + \Delta t$ . La repetición del proceso permite conocer el resto de los resultados.

  
Enlace a código del método de Euler implícito (pulsar cuadro en Word)

## CONCLUSIONES

Dejando de lado las conclusiones físicas, es decir, las relaciones triviales que se pueden establecer entre las magnitudes del problema (variación masa-posición, amplitud-constante elástica, etc.), nos centraremos en aquellas conclusiones de tipo computacional:

- El movimiento oscilatorio observable en las gráficas anteriores nos confirma que el movimiento sí es armónico simple.

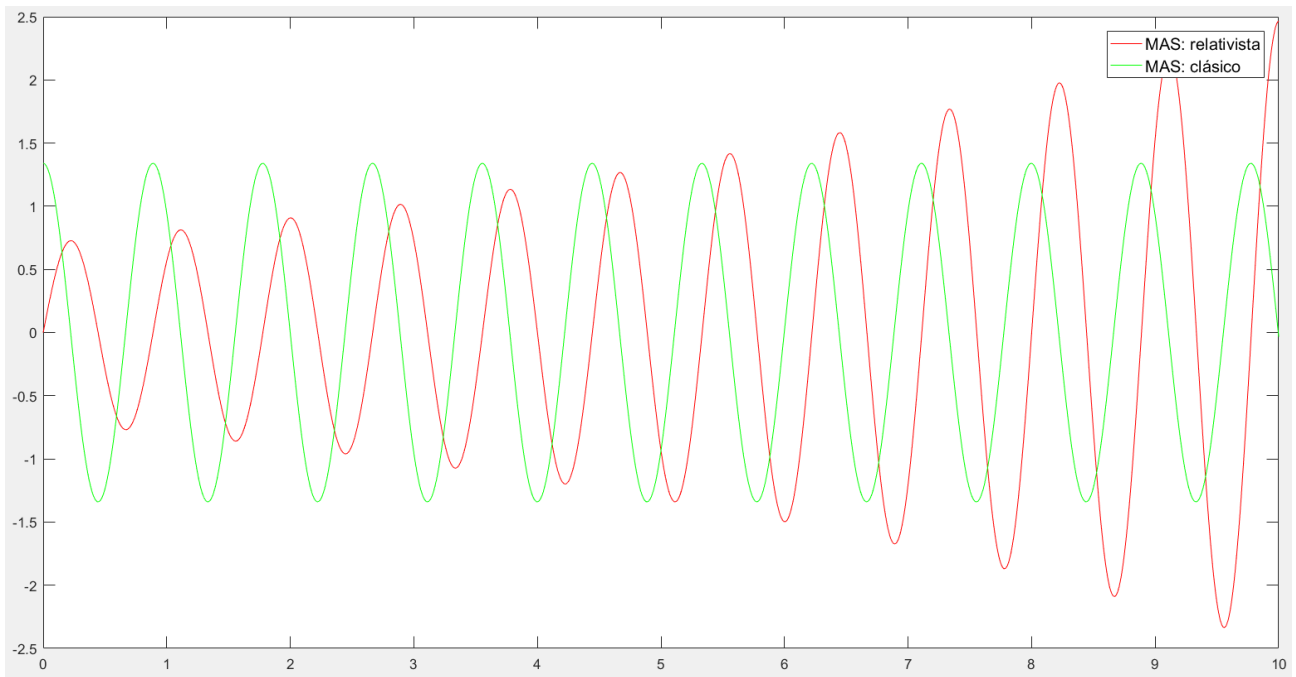
- Aunque a muy altas precisiones computacionales (como en el caso de las gráficas propuestas) los tres métodos funcionaban de una manera correcta, cuando probamos estos para incrementos temporales mayores, observamos que existía una gran diferencia entre los tres procedimientos:

Se presentan los métodos ordenados de menor a mayor facilidad para dispararse la posición en un instante dado (con  $\Delta t \approx 0.5$ ):

*Método de Runge-Kutta < Método de Adams-Bashforth < Método de Euler*

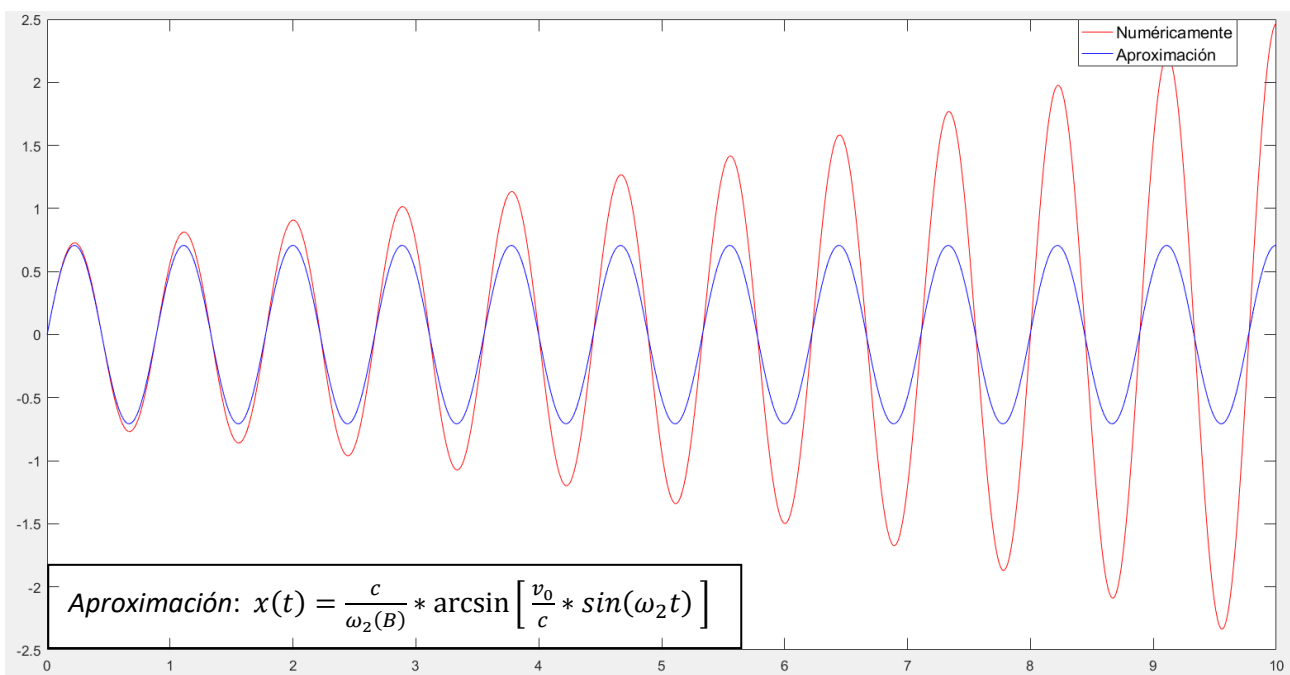
Se podría (¿se podría?) concluir de lo anterior que la precisión de los métodos multietapa es mayor que la de los métodos multipaso, cuya precisión es, a su vez, mayor que la del método de Euler (de un solo paso).

- Que en las gráficas vaya aumentando con el tiempo la amplitud del movimiento y, consecuentemente, la longitud de onda se debe simplemente a la naturaleza relativista del movimiento propuesto. He aquí una comparación del MAS entre movimientos regidos por la mecánica relativista y clásica:



- A decir verdad, no es algo que hayamos descubierto nosotros mismos, pero, buscando información, encontramos que el método de Euler implícito, a pesar de su complejidad computacional, es un método siempre estable y, frente al explícito, podría suponer una ventaja ante problemas no lineales.

- Los procedimientos iterativos multietapa, sea el caso del método Runge-Kutta, presentan un error total del orden del número de etapas que “acoge”. En nuestro ejemplo, el error es  $O(h^2)$ .



- En la tabla anterior, se representan simultáneamente la función del MAS obtenido numéricamente y por la aproximación al problema. Se observa que, en los primeros instantes, ambas siguen una trayectoria muy parecida. Se deja al lector la interpretación de por qué, con el paso del tiempo, las ondas presentan una forma distinta.

#### **\*COMENTARIOS**

Para realizar las gráficas anteriores, se ha introducido un  $\Delta t = 0.005$  en el intervalo  $[0,10]$ . Además, por si fuera de interés, los datos base del problema han sido los mismos para todos los casos presentados:

$$\begin{aligned}m &= 3 \text{ kg} \\k &= 150 \text{ N/m} \\x_0 &= 0 \text{ m} && \text{--punto de equilibrio--} \\v_0 &= 5 \text{ m/s} \\c &= 300\,000\,000 \text{ m/s}\end{aligned}$$

Dada la suficiente precisión que se ha podido lograr gracias a programas matemáticos, no se ha considerado necesario comparar gráficamente los resultados de los cuatro métodos. Las gráficas han sido realizadas a partir de uno de los métodos indistintamente.

A su vez, se ha evitado poner tablas de datos en el documento, pues, a nuestro parecer, resultan poco atractivos y no aportan nada al entendimiento del problema que se ha decidido plantear.

**Yago Pego Martínez**  
**Evaristo de Vega Galindo**