UNIVERSIDAD DE BUENOS AIRES
Facultad de Ingeniería

Departamento de Computación

TRABAJO PRÁCTICO Nº 2

Análisis Numérico I (75.12)

- Curso de Verano 2013 –

Docentes:

Navarro, Fabián

Vicario, Federico

Alumnos:

APELLIDO, Nombres	N° PADRÓN
Rinaldi, Lautaro Ezequiel	

Calificación: Firma:

Observaciones:

ENUNCIADO

ANALISIS NUMERICO I - FIUBA

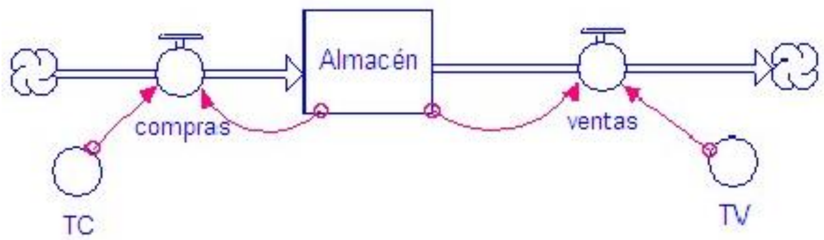
75.12 – 95.04 - Curso de verano

TRABAJO PRÁCTICO Nº 2

Introducción:

La dinámica de sistemas es una técnica que permite analizar y modelar el comportamiento temporal de entornos complejos. Fue originalmente desarrollada a principios de la década de 1960 por Jay Forrester del MIT Sloan School of Management (Massachusetts Institute of Technology).

A continuación se presenta un modelo reducido del almacén de una empresa comercial. El mismo aumenta o disminuye debido a las compras o envíos respectivamente realizados.



Donde TC y TV se definen como Tasa de Compras y Tasa de Ventas respectivamente.

En el presente trabajo no se exige comprender en detalle cómo realizar el modelo ni la simbología utilizada, sino la resolución y posterior análisis de la ecuación diferencial asociada al mismo. Dicha ecuación se presenta a continuación

$$\frac{dx}{dt} = TCx - TVx$$

La misma representa la tasa de crecimiento o disminución (según el caso) del producto en el almacén. Cabe destacar que la empresa presenta como política que tanto las ventas como los envíos dependen de la mercadería en stock (productos almacenados).

El objetivo del presente trabajo es poder determinar cómo varía el stock del almacén de la empresa a lo largo de 12 meses.

Se pide:

1- Suponiendo TC=10% (0,1), TV=20% (0,2), Stock inicial=100 unidades
   a. Resolver la ecuación diferencial en forma analítica.
   b. Resolver la ecuación diferencial mediante MatLab/Octave utilizando los siguientes métodos:
      i. Euler
      ii. Runge Kutta 2
      iii. Runge Kutta 4
   c. Investigar y resolver la ecuación diferencial mediante ode23 y ode45 (métodos predefinidos de MatLab/Octave). Indicar el número de pasos en cada caso y determinar la diferencia de dichos métodos respecto a los del punto b (i, ii, iii).

Para cada caso realizar una introducción teórica.

Suponer un paso de 0,1 y un tiempo final de 1,2.

Graficar los resultados obtenidos.

2- Realizar una introducción teórica acerca de la estabilidad en EDO. Encontrar la combinación de TC y TV que permita obtener la mejor estabilidad (Factor de Amplificación – FA – menor) para cada uno de los métodos numéricos anteriormente utilizados.

   Suponer un paso de 0,1.

   TC debe tener como valor mínimo 5% (0,05) y máximo 0,85 (85%), variando en 5% (0,05).

   TV debe tener como valor mínimo 10% (0,1) y máximo 0,9 (90%), variando en 5% (0,05).

   TC debe ser distinto de TV en el análisis.

   ¿Se puede afirmar que alguno de los métodos resulta ser estable? Justificar la respuesta.

   Representar las variaciones de TC, TV y FA en un mismo gráfico.

3- Utilizando los resultados del punto anterior, se pide resolver nuevamente la ecuación diferencial mediante cada uno de los métodos anteriormente utilizados. Suponer un paso de 0,1 y un tiempo final de 1,2. Stock inicial=100 unidades.

4- En el caso que la política de stocks cambiase y la TV no dependiera del stock, indique y justifique si cambiarían sus respuestas anteriores.

5- Comparar los resultados obtenidos y obtener conclusiones.

<u>Consideraciones sobre el Trabajo Práctico:</u>

1) Deben incluirse los códigos fuente utilizados en MatLab u Octave para el desarrollo del trabajo.
2) Los resultados de las corridas de máquina, junto con los gráficos realizados deben encontrarse impresos para su correcta evaluación.

INTRODUCCIÓN TEÓRICA

Las ecuaciones en las cuales la función desconocida, escalar o vectorial, se encuentra bajo el signo de derivada o de diferencial, se llaman ecuaciones diferenciales.

Si en una ecuación diferencial, las funciones desconocidas, escalares o vectoriales, son funciones de una sola variable, la ecuación diferencial es llamada *"ordinaria"* ; Si en cambio, la función desconocida es una función de dos o más variables independientes, la ecuación diferencial se llama *"ecuación en derivadas parciales"*.

Se denomina *"orden de la ecuación diferencial"* al orden de la derivada (o de la diferencial) máxima de la función desconocida que figura en la ecuación.

El *"grado"* de una ecuación diferencial ordinaria no trascendente, es el grado algebraico de la derivada que determina el orden de la ecuación diferencial, habiendo eliminado de la misma cualquier potencia fraccionaria.

Se llama *"solución"* de la ecuación diferencial, a una función que al ser sustituida en la misma, la convierte en una identidad.

El proceso de determinación de las soluciones de una ecuación diferencial, se llama *"integración"* de la misma. Por ejemplo, un método de integración es el *"Método de Euler"* , el cual consiste en sustituir la curva integral buscada, solución de la ecuación diferencial de primer orden, por una quebrada constituida por segmentos lineales, cada uno de los cuales es tangente a una curva integral de la ecuación diferencial.

La mayoría de las ecuaciones diferenciales de m-ésimo orden que aparecen en la práctica pueden transformarse, introduciendo funciones auxiliares, a un sistema de m ecuaciones diferenciales de primer orden.

Toda expresión de la forma

$$u_0 = y_0 , \quad u_{k+1} = u_k + h \cdot \phi(x_k, u_k, u_{k+1}; h; f)$$

constituye un *"Método de Paso Simple"* para aproximar numéricamente a y(x).

Si en la función ϕ aparece explícitamente u_{k+1}, decimos que el método es *"implícito"*; En caso contrario, decimos que es *"explícito"*.

A continuación, describimos brevemente los métodos de paso simple que vamos a utilizar en el desarrollo del presente práctico:

Método de Euler

Toma $u_0 = x(t_0)$;

Una vez que aproximó $x(t_n)$ por u_n, avanza sobre la tangente a la solución de la ecuación diferencial que pasa por el punto (t_n, u_n):

$$u_{n+1} = u_n + h \cdot f(t_n, u_n)$$

Este método es explícito y de orden 1.

Método de Euler Modificado o Runge-Kutta de orden 2

Toma $u_0 = x(t_0)$;

Una vez que aproximó $x(t_n)$ por u_n, avanza sobre una recta cuya pendiente es el promedio de las pendientes de las rectas tangentes a las soluciones que pasan por los puntos

$$(t_n, u_n),\ (t_{n+1}, u_n + h \cdot f(t_n, u_n)):$$

Siendo el método:

$$q_1 = h \cdot f(t_n, u_n)$$

$$q_2 = h \cdot f(t_{n+1}, u_n + q_1)$$

$$u_{n+1} = u_n + \frac{q_1 + q_2}{2}$$

Método de Runge-Kutta de orden 4

Toma $u_0 = x(t_0)$;

Una vez que aproximamos $x(t_n)$ por u_n, hacemos:

$$q_1 = h \cdot f(t_n, u_n)$$

$$q_2 = h \cdot f(t_{n+1/2}, u_n + \frac{q_1}{2})$$

$$q_3 = h \cdot f(t_{n+1/2}, u_n + \frac{q_2}{2})$$

$$q_4 = h \cdot f(t_{n+1}, u_n + q_3)$$

La solución avanza sobre una recta cuya pendiente es el promedio pesado de las pendientes de las rectas tangentes a las soluciones que pasan por los puntos:

$$(t_n, u_n),\ (t_{n+1/2}, u_n + \frac{q_1}{2}),\ (t_{n+1/2}, u_n + \frac{q_2}{2}),\ (t_{n+1}, u_n + q_3):$$

Siendo el método:

$$u_{n+1} = u_n + \frac{q_1 + 2 \cdot q_2 + 2 \cdot q_3 + q_4}{6}$$

Estabilidad en EDO

Podemos decir que un método es estable si dados cambios o perturbaciones pequeñas en los datos iniciales producen cambios igualmente pequeñas en las aproximaciones posteriores. Es decir, un método estable es aquel cuyos resultados se basan continuamente en los datos iniciales.

Matemáticamente hablando, diremos que un método es estable si y sólo sí existe una función de orden h, O(h), tal que:

$$\varepsilon_{k+1} = (1 + O(h)) \cdot \varepsilon_k \quad \text{para todo k}$$

siendo (1 + O(h)) el factor de amplificación.

Podemos concluir que al resolver un dado problema numérico, las propagaciones de los errores las maneja la función O(h) que aparece en la definición de estabilidad a través del **Factor de Amplificación:** (1 + O(h)). Si la función O(h) es negativa, para ciertos valores de h podremos hacer que el factor de amplificación tenga módulo menor o igual que 1, y tendremos una condición óptima de trabajo.

Si la función O(h) es positiva, tendremos que trabajar con valores de h que la hagan muy pequeña, y no podremos avanzar mucho en la solución sin que los errores propagados se hagan notar.

DESARROLLO

Resolución analítica de la ecuación diferencial

En primer lugar, dada la sencillez de la ecuación diferencial, vamos a resolverla analíticamente por ser de variables separables. De este modo, podremos conocer los valores reales de la solución y compararlos con los resultados obtenidos numéricamente.

$$\frac{dx}{dt} = TCx - TVx$$
$$\frac{dx}{dt} = (TC - TV) \cdot x$$
$$\frac{dx}{x} = (TC - TV) \cdot dt$$
$$\int \frac{dx}{x} = \int (TC - TV) \cdot dt$$
$$\int \frac{dx}{x} = (TC - TV) \cdot \int dt$$
$$\ln(x) = (TC - TV) \cdot t + C$$
$$x = k \cdot e^{(TC-TV)\cdot t}$$

Utilizando las condiciones iniciales $x(0) = 100$ obtenemos que $k = 100$

La solución analítica de la ecuación diferencial es $x(t) = 100 \cdot e^{(TC-TV)\cdot t}$

Obs:

Recordamos que en dicha ecuación diferencial TC es la tasa de compra y TV es la tasa de venta; *x* representa el stock, y t el tiempo que transcurrió en meses desde el instante inicial (correspondiéndose 0,1 a un mes, 0,6 para seis meses, y de la misma forma con los restantes meses).

Resolución numérica de la ecuación diferencial

A continuación, se particularizan los métodos descriptos en la introducción teórica a nuestro problema en particular

$$\frac{dx}{dt} = TCx - TVx$$

Método de Euler:

$$u_0 = 100$$

$$u_{n+1} = u_n + h \cdot (TC - TV) \cdot u_n$$

Método Runge Kutta de orden 2

$$u_0 = 100$$

$$q_1 = h \cdot (TC - TV) \cdot u_n$$

$$q_2 = h \cdot (TC - TV) \cdot (u_n + q_1)$$

$$u_{n+1} = u_n + \frac{q_1 + q_2}{2}$$

Método Runge Kutta de orden 4

$$u_0 = 100$$

$$q_1 = h \cdot (TC - TV) \cdot u_n$$

$$q_2 = h \cdot (TC - TV) \cdot (u_n + \frac{q_1}{2})$$

$$q_3 = h \cdot (TC - TV) \cdot (u_n + \frac{q_2}{2})$$

$$q_4 = h \cdot (TC - TV) \cdot (u_n + q_3)$$

$$u_{n+1} = u_n + \frac{q_1 + 2 \cdot q_2 + 2 \cdot q_3 + q_4}{6}$$

Factores de Amplificación

A continuación, detallamos las igualdades que nos permiten conocer los factores de amplificación en los métodos numéricos que vamos a utilizar para resolver la ecuación diferencial del presente trabajo práctico.

Método de Euler:

$$\varepsilon_{n+1} = [1 + h \cdot (TC - TV)] \cdot \varepsilon_n$$

Método Runge Kutta de orden 2

$$\varepsilon_{n+1} = [1 + \frac{a}{2} \cdot (2 + a)] \cdot \varepsilon_k$$

siendo

$$a = h \cdot (TC - TV)$$

Método Runge Kutta de orden 4

$$\varepsilon_{n+1} = [1 + \frac{1 + 2b + 2c + d}{6} \cdot a] \cdot \varepsilon_k$$

siendo

$$a = h \cdot (TC - TV)$$

$$b = 1 + \frac{a}{2}$$

$$c = 1 + \frac{a}{2} \cdot b$$

$$d = 1 + a \cdot c$$

CÓDIGOS FUENTE

```
function [t, u] = Euler(paso, tiempoinicial, tiempofinal, TC, TV, stockinicial)
% Euler(0.1, 0, 1.2, 0.1, 0.2, 100)

    t= zeros(fix((tiempofinal-tiempoinicial)/paso),1);
    u= zeros(fix((tiempofinal-tiempoinicial)/paso),1);
    u(1)= stockinicial;
    t(1)= tiempoinicial;

    contador= 1;

    for x= (tiempoinicial+paso) : paso : tiempofinal
        t(contador+1)= x;
        u(contador+1)= u(contador) + (paso * (TC-TV) * u(contador));
        contador= contador + 1;
    end

return
```

```
function [t, U]= RK2(paso, tiempoinicial, tiempofinal, TC, TV, stockinicial)
% RK2(0.1, 0, 1.2, 0.1, 0.2, 100)

    t= zeros(fix((tiempofinal-tiempoinicial)/paso),1);
    U= zeros(fix((tiempofinal-tiempoinicial)/paso),1);
    U(1)= stockinicial;
    t(1)= tiempoinicial;

    contador= 1;

    for x= (tiempoinicial+paso) : paso : tiempofinal
        t(contador+1)= x;
        q1= paso * (TC-TV) * U(contador);
        q2= paso * (TC-TV) * (U(contador) + q1);
        U(contador+1)= U(contador) + ((q1+q2)/2);
        contador= contador + 1;
    end

return
```

```
function [t, U] = RK4(paso, tiempoinicial, tiempofinal, TC, TV, stockinicial)
% RK4(0.1, 0, 1.2, 0.1, 0.2, 100)

    t= zeros(fix((tiempofinal-tiempoinicial)/paso),1);
    U= zeros(fix((tiempofinal-tiempoinicial)/paso),1);
    U(1)= stockinicial;
    t(1)= tiempoinicial;

    contador= 1;

    for x= (tiempoinicial+paso) : paso : tiempofinal
        t(contador+1)= x;
        q1= paso * (TC-TV) * U(contador);
        q2= paso * (TC-TV) * (U(contador) + q1/2);
        q3= paso * (TC-TV) * (U(contador) + q2/2);
        q4= paso * (TC-TV) * (U(contador) + q3);

        U(contador+1)= U(contador) + ((q1 + 2*q2 + 2*q3 + q4)/6);
        contador= contador + 1;
    end

return
```

```
function [FAmin, TCmin, TVmin, matriz] = minimoEuler(paso)

FAmin= abs(1 + paso * (0.05 - 0.1));
TCmin= 0.05;
TVmin= 0.1;

matriz = ones(17, 17);

for x= 1:17
    for y= 1:17
           FA= abs(1 + paso * (x*0.05 - (y*0.05+0.05)));
           matriz( x, y)= FA;
           if FA < FAmin
                FAmin= FA;
                TCmin= x*0.05;
                TVmin= y*0.05+0.05;
           end
    end
end

return
```

```
function [FAmin, TCmin, TVmin, matriz] = minimoRK2(paso)

A= paso * (0.05 - 0.1);

FAmin= abs( 1 + A/2 * (2 + A));
TCmin= 0.05;
TVmin= 0.1;

matriz = ones(17, 17);

for x= 1:17
    for y= 1:17
           A= paso * (0.05*x - (0.05*y+0.05));
           FA= abs(1 + A/2 * (2 + A));
           matriz( x, y)= FA;
           if FA < FAmin
                FAmin= FA;
                TCmin= x*0.05;
                TVmin= y*0.05+0.05;
           end
    end
end

return
```

```
function [FAmin, TCmin, TVmin, matriz] = minimoRK4(paso)

A= paso * (0.05 - 0.1);
B= 1+ A/2;
C= 1 + A/2 * B;
D= 1 + A * C;


FAmin= abs( 1 + (A + 2*A*B + 2*A*C + A*D)/6 );
TCmin= 0.05;
TVmin= 0.1;

matriz = ones(17, 17);

for x= 1:17
    for y= 1:17
            A= paso * (0.05*x - (0.05*y+0.05));
            B= 1+ A/2;
            C= 1 + A/2 * B;
            D= 1 + A * C;
            FA= abs( 1 + (A + 2*A*B + 2*A*C + A*D)/6 );
            matriz( x, y)= FA;
           if FA < FAmin
                FAmin= FA;
                TCmin= x*0.05;
                TVmin= y*0.05+0.05;
           end
    end
end

return
```

```
function dy = Funcion(t, x)

dy= -0.1 * x ;

return
```

```
function dy = Funcion(t, x)

dy= -0.85 * x ;

return
```

```
function [t, u] = analitica(paso, tiempoinicial, tiempofinal, TC, TV)

t= zeros(fix((tiempofinal-tiempoinicial)/paso), 1);

u= zeros(fix((tiempofinal-tiempoinicial)/paso), 1);

for x= tiempoinicial:paso:tiempofinal
    t(fix(x/paso)+1)= x;
    u(fix(x/paso)+1)= 100 * exp((TC-TV)*x);
end

return
```

IMPRESIÓN DE LOS RESULTADOS

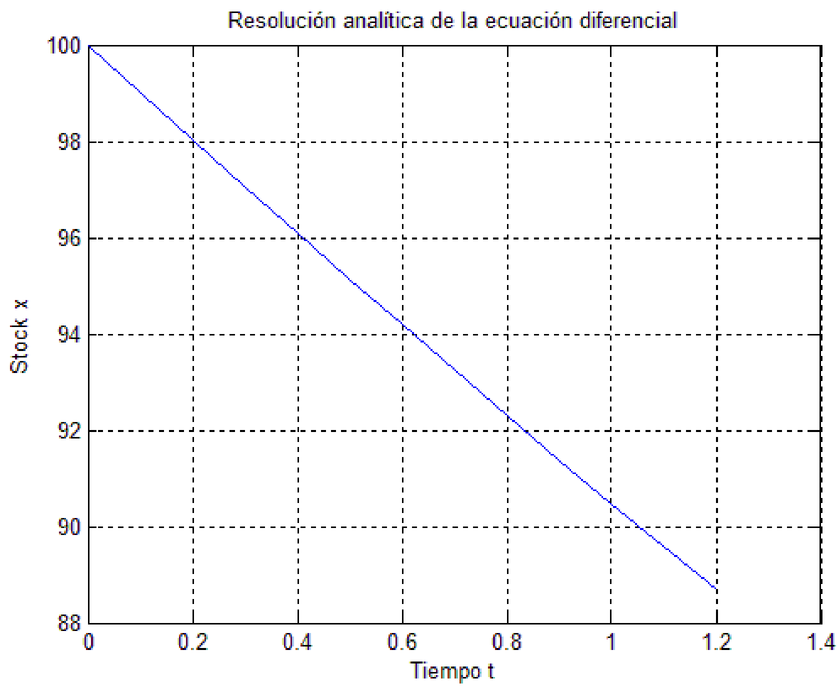
Resolución de la ecuación analítica:

Para invocar a la función que evalúa la función que es solución de la ecuación diferencial (obtenida en forma analítica), y por ende nos otorga los valores verdaderos, debemos escribir en la línea de comandos:

[t, u] =analitica(0.1, 0, 1.2, 0.1, 0.2)

t	x
0	100.0000000000000
0.1000	99.0049833749168
0.2000	98.0198673306755
0.3000	97.0445533548508
0.4000	96.0789439152323
0.5000	95.1229424500714
0.6000	94.1764533584249
0.7000	93.2393819905948
0.8000	92.3116346386636
0.9000	91.3931185271228
1.0000	90.4837418035960
1.1000	89.5834135296528
1.2000	88.6920436717157

Gráfico:



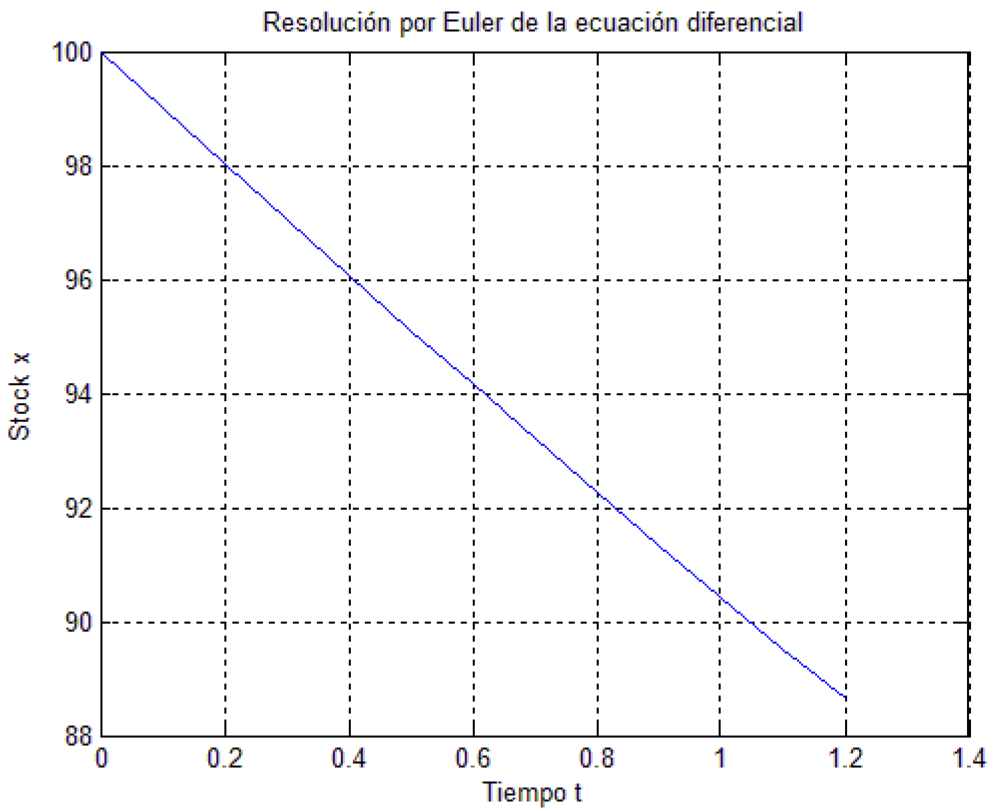
Método de Euler

Para obtener la solución numérica de la ecuación diferencial, debemos ejecutar la rutina

[t, u]= Euler(0.1, 0, 1.2, 0.1, 0.2, 100)

t	x
0	100.0000000000000
0.1000	99.0000000000000
0.2000	98.0100000000000
0.3000	97.0299000000000
0.4000	96.0596010000000
0.5000	95.0990049900000
0.6000	94.1480149401000
0.7000	93.2065347906990
0.8000	92.2744694427920
0.9000	91.3517247483641
1.0000	90.4382075008805
1.1000	89.5338254258716
1.2000	88.6384871716129

Gráfico:



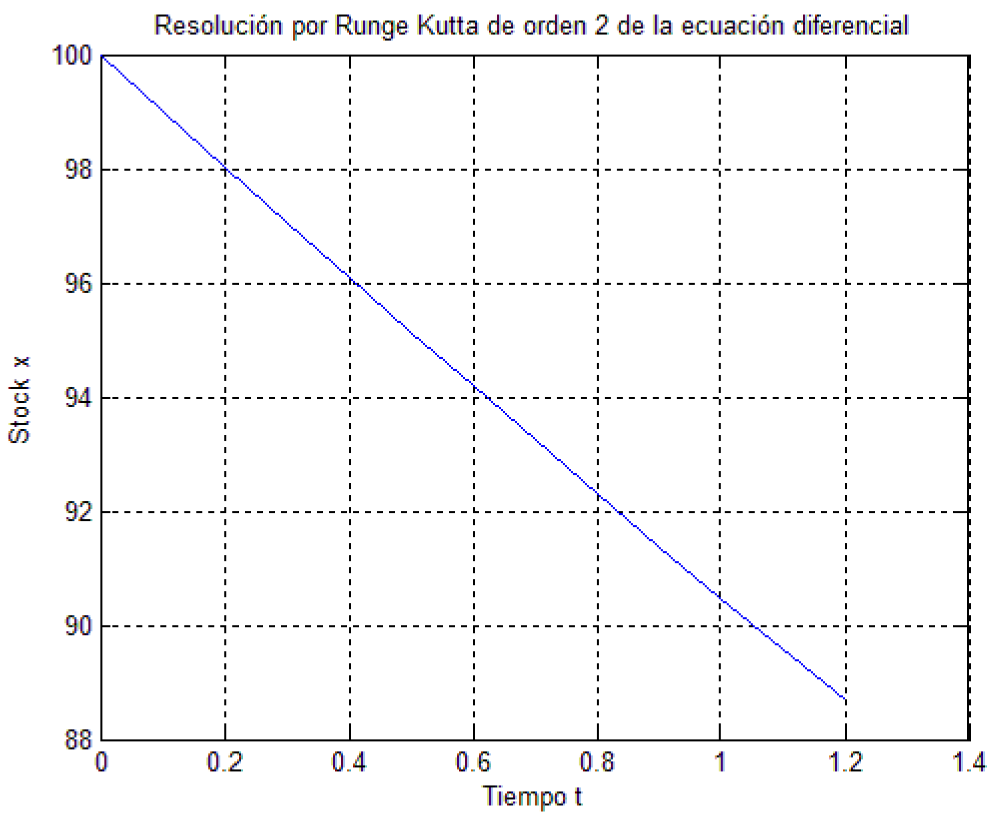
Método Runga Kutta de orden 2

Para obtener la solución numérica de la ecuación diferencial, debemos ejecutar la rutina

[t, U] = RK2(0.1, 0, 1.2, 0.1, 0.2, 100)

t	x
0	100.0000000000000
0.1000	99.0050000000000
0.2000	98.0199002500000
0.3000	97.0446022425125
0.4000	96.0790084501995
0.5000	95.1230223161200
0.6000	94.1765482440746
0.7000	93.2394915890461
0.8000	92.3117586477351
0.9000	91.3932566491901
1.0000	90.4838937455307
1.1000	89.5835790027626
1.2000	88.6922223916851

Gráfico:



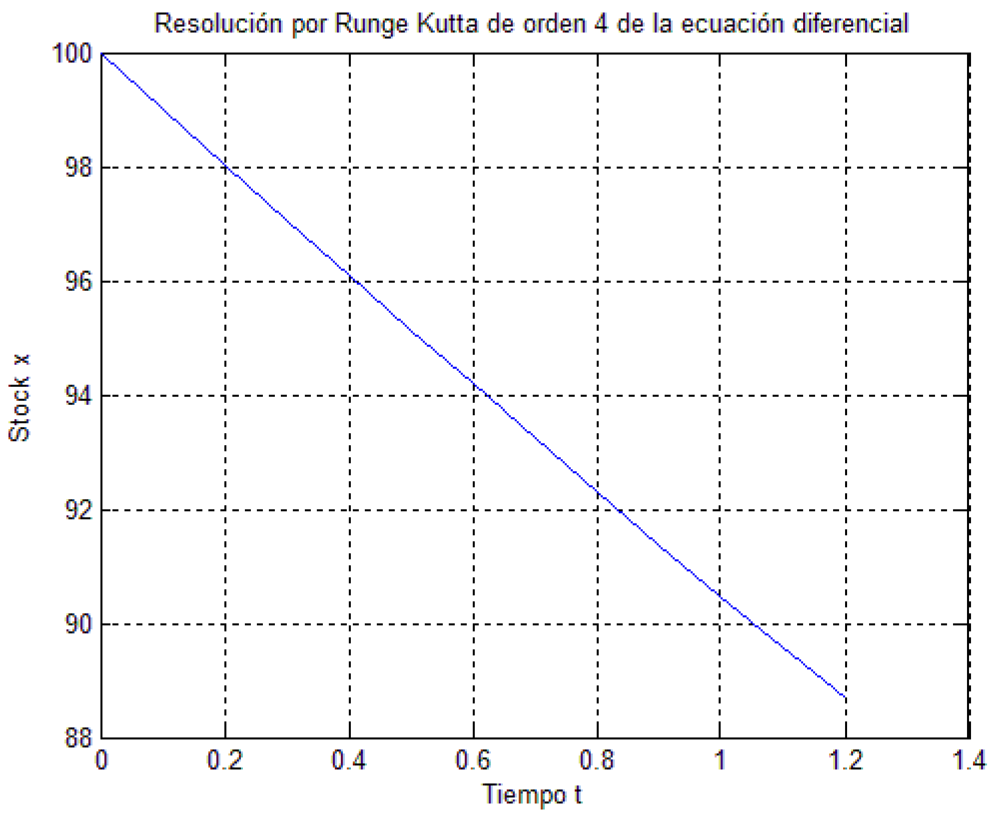
Método Runga Kutta de orden 4

Para obtener la solución numérica de la ecuación diferencial, debemos ejecutar la rutina

[t, U]= RK4(0.1, 0, 1.2, 0.1, 0.2, 100)

t	x
0	100.0000000000000
0.1000	99.0049833750000
0.2000	98.0198673308403
0.3000	97.0445533550954
0.4000	96.0789439155552
0.5000	95.1229424504710
0.6000	94.1764533588997
0.7000	93.2393819911433
0.8000	92.3116346392841
0.9000	91.3931185278140
1.0000	90.4837418043563
1.1000	89.5834135304809
1.2000	88.6920436726101

Gráfico:



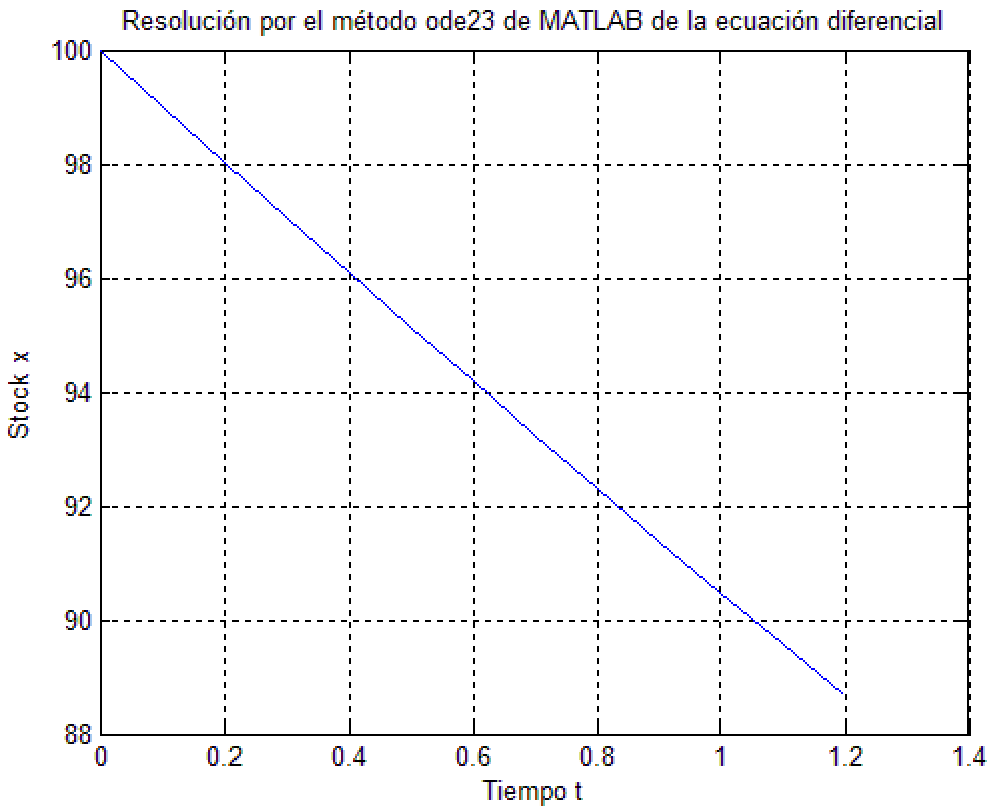
Método predefinido en MATLAB **ode23**

Para obtener la solución numérica de la ecuación diferencial, debemos ejecutar la rutina

[x, y]= ode23(@Funcion , [0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0, 1.1, 1.2], 100)

t	x
0	100.0000000000000
0.1000	99.0049833333333
0.2000	98.0198672087341
0.3000	97.0445531626808
0.4000	96.0789436602825
0.5000	95.1229421357140
0.6000	94.1764529826433
0.7000	93.2393815447526
0.8000	92.3116341211897
0.9000	91.3931179475163
1.0000	90.4837411688321
1.1000	89.5834128428680
1.2000	88.6920429380813

Gráfico:



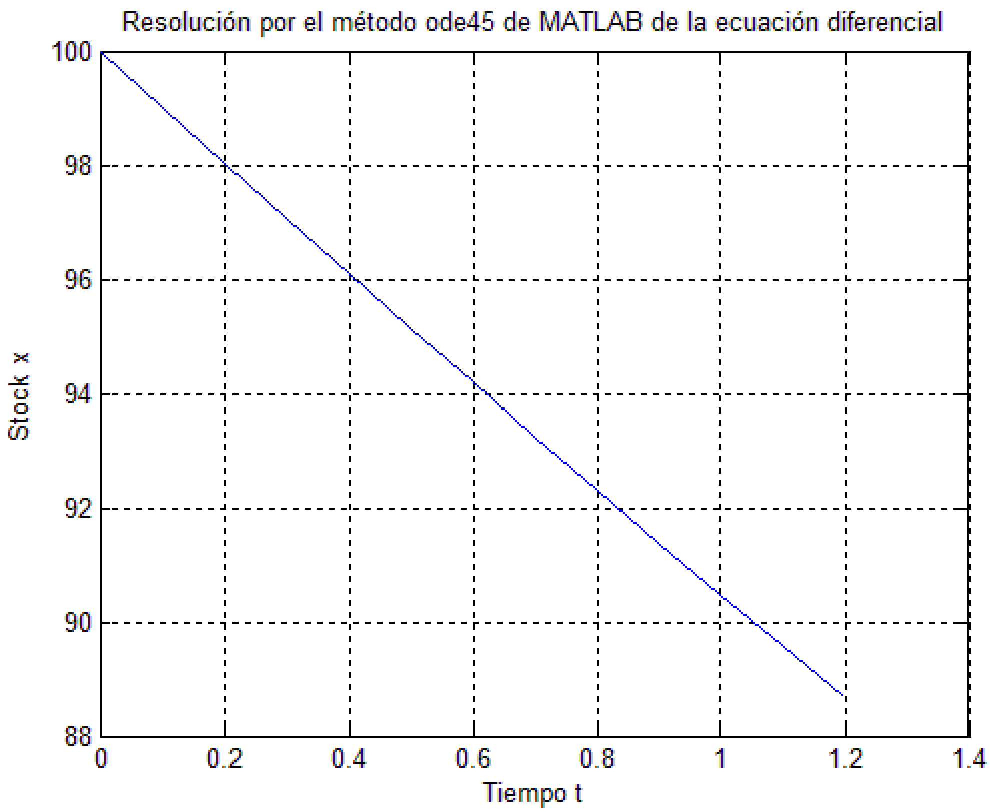
<u>Método predefinido en MATLAB **ode45**</u>

Para obtener la solución numérica de la ecuación diferencial, debemos ejecutar la rutina

[x, y]= ode45(@Funcion , [0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0, 1.1, 1.2], 100)

t	x
0	100.0000000000000
0.1000	99.0049833749168
0.2000	98.0198673306729
0.3000	97.0445533548425
0.4000	96.0789439152199
0.5000	95.1229424500602
0.6000	94.1764533584202
0.7000	93.2393819905953
0.8000	92.3116346386615
0.9000	91.3931185271154
1.0000	90.4837418035846
1.1000	89.5834135296426
1.2000	88.6920436717165

<u>Gráfico</u>:



Diferencia entre los métodos **ode23** y **ode45** de MATLAB contra los métodos que nosotros programamos

La diferencia fundamental entre los métodos que hemos programado y los de MATLAB, es que en estos últimos, el paso de integración se determina dinámicamente, en función del comportamiento de la ecuación y con el objetivo de que el error estimado esté dentro de cierta tolerancia, mientras que en los métodos que hemos programado nosotros, el paso es fijo.

Es por eso, que las funciones ode23 y ode45, para poder mostrarnos los valores en los puntos que nosotros le pedimos, internamente está interpolando los puntos que obtuvo como solución de la ecuación diferencial.

Análisis de Errores

Método de Euler

Graficamos el error verdadero para Euler (solución analítica menos solución por Euler)

>> format long

>> [a1, b1] =analitica(0.1, 0, 1.2, 0.1, 0.2)

>> [a2, b2] = Euler(0.1, 0, 1.2, 0.1, 0.2, 100)

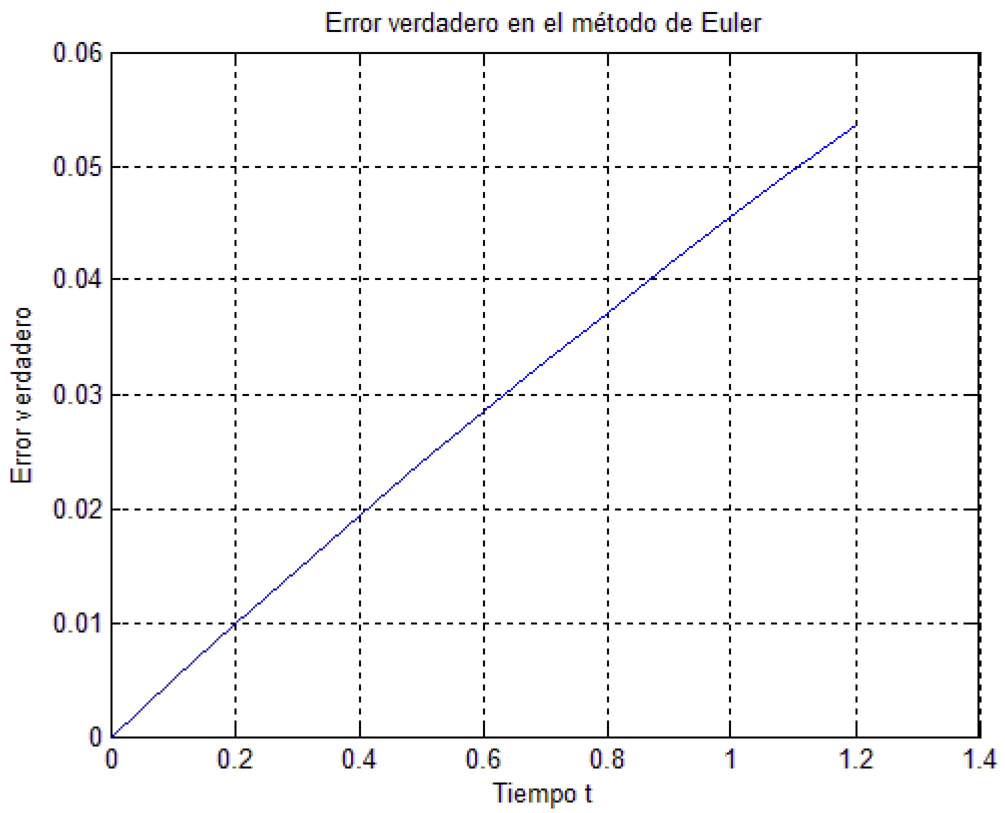
>> plot(a1, b1-b2)

>> grid on

>> xlabel('Tiempo t')

>> ylabel('Error verdadero')

>> title('Error verdadero en el método de Euler)



Observación:

Notar que el error verdadero es del orden de 10^{-2}. Como el error es positivo, podemos decir que el método de Euler subestima la solución verdadera.

Método Runga Kutta de orden 2

Graficamos el error verdadero para Runge Kutta 2 (solución analítica menos solución por RK2)

>> format long

>> [a1, b1] = analitica(0.1, 0, 1.2, 0.1, 0.2)

>> [a3, b3] = RK2(0.1, 0, 1.2, 0.1, 0.2, 100)

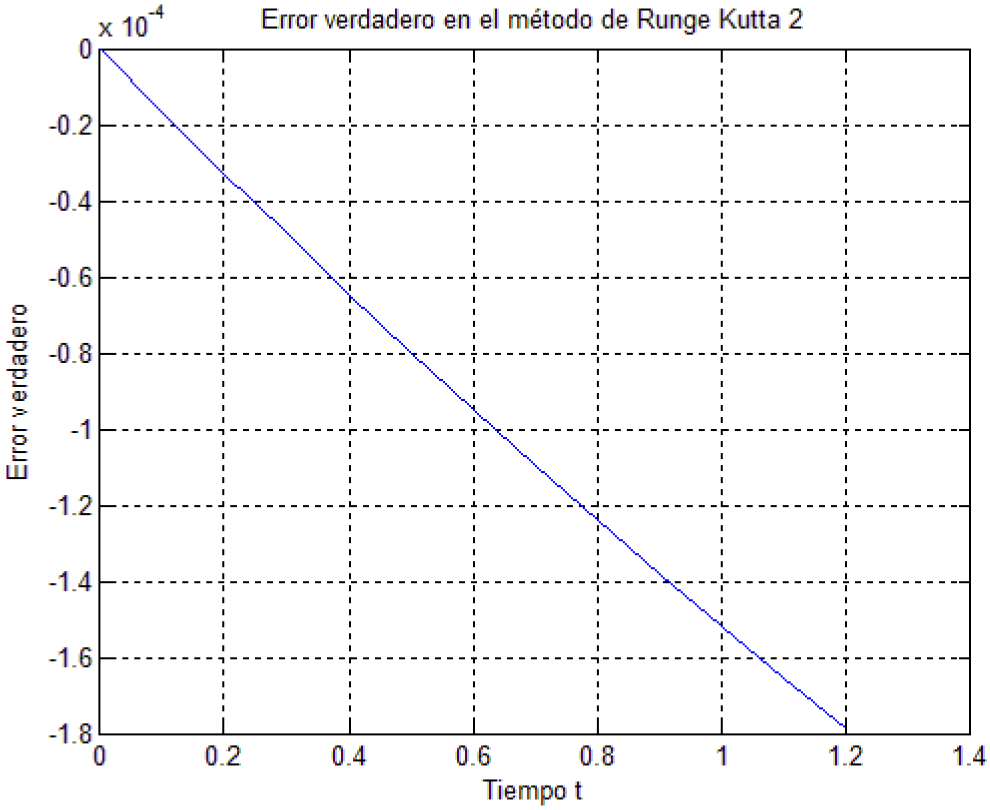
>> plot(a1, b1-b3)

>> grid on

>> xlabel('Tiempo t')

>> ylabel('Error verdadero')

>> title('Error verdadero en el método de Runge Kutta 2')



Observación:

Notar que el error verdadero es del orden de 10^{-4}. Como el error es negativo, podemos decir que el método de Euler sobreestima la solución verdadera.

Método Runga Kutta de orden 4

Graficamos el error verdadero para Runge Kutta 4 (solución analítica menos solución por RK4)

>> format long

>> [a1, b1] = analitica(0.1, 0, 1.2, 0.1, 0.2)

>> [a3, b3] = RK4(0.1, 0, 1.2, 0.1, 0.2, 100)

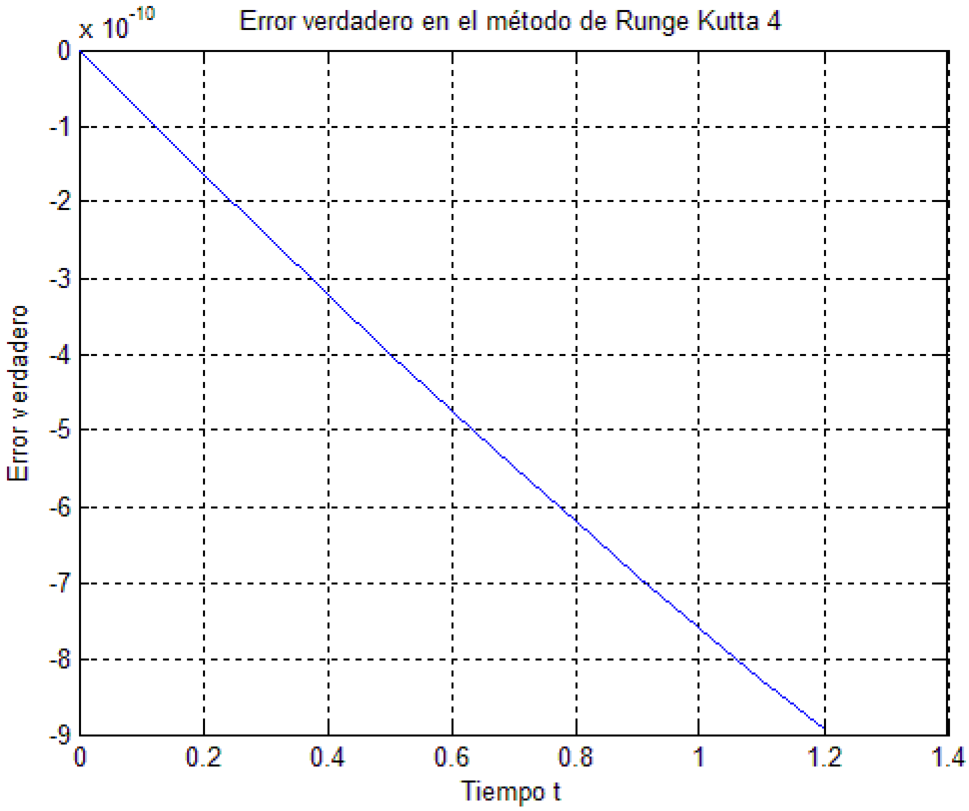
>> plot(a1, b1-b4)

>> grid on

>> xlabel('Tiempo t')

>> ylabel('Error verdadero')

>> title('Error verdadero en el método de Runge Kutta 4')



Observación:

Notar que el error verdadero es del orden de 10^{-10}. Como el error es negativo, podemos decir que el método de Euler sobreestima la solución verdadera.

Método predefinido en MATLAB **ode23**

Graficamos el error verdadero para ode23 (solución analítica menos solución por ode23)

>> format long

>> [a1, b1] = analitica(0.1, 0, 1.2, 0.1, 0.2)

>> [a5, b5]= ode23(@Funcion , [0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0, 1.1, 1.2], 100)

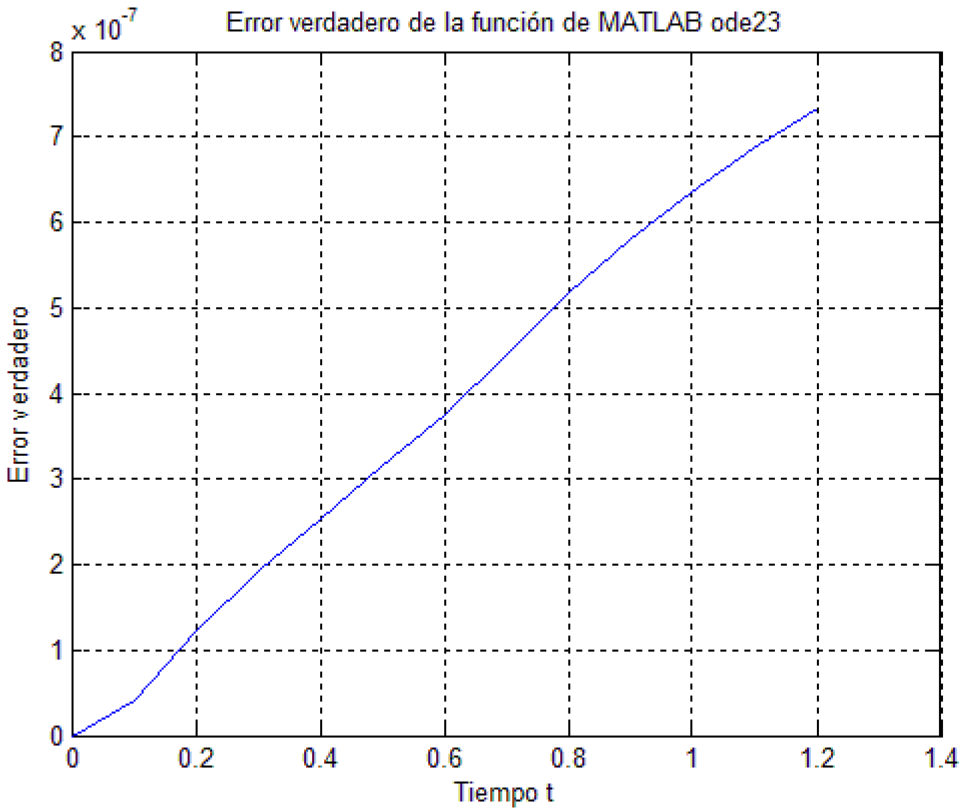
>> plot(a1, b1-b5)

>> grid on

>> xlabel('Tiempo t')

>> ylabel('Error verdadero')

>> title('Error verdadero de la función de MATLAB ode23')



Observación:

Notar que el error verdadero es del orden de 10^{-7}.

Método predefinido en MATLAB **ode45**

Graficamos el error verdadero para ode45 (solución analítica menos solución por ode45)

>> format long

>> [a1, b1] = analitica(0.1, 0, 1.2, 0.1, 0.2)

>> [a6, b6]= ode45(@Funcion , [0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0, 1.1, 1.2], 100)

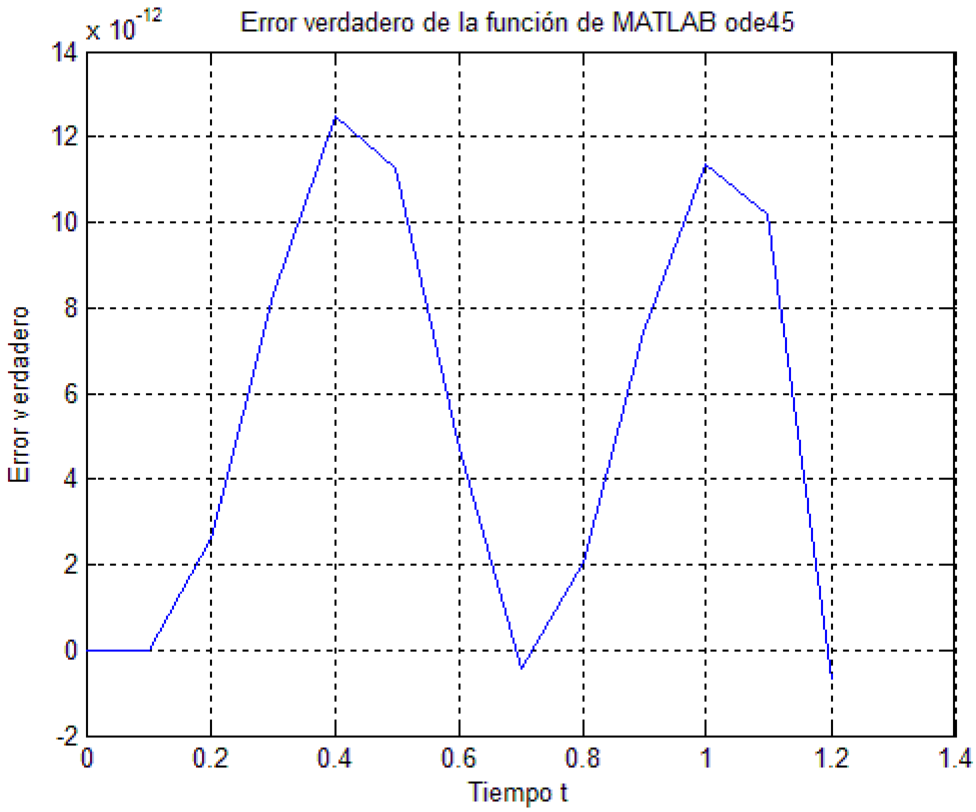
>> plot(a1, b1-b6)

>> grid on

>> xlabel('Tiempo t')

>> ylabel('Error verdadero')

>> title('Error verdadero de la función de MATLAB ode45')



Observación:

Notar que el error verdadero es del orden de 10^{-12}.

FACTORES DE AMPLIFICACIÓN MÍNIMOS

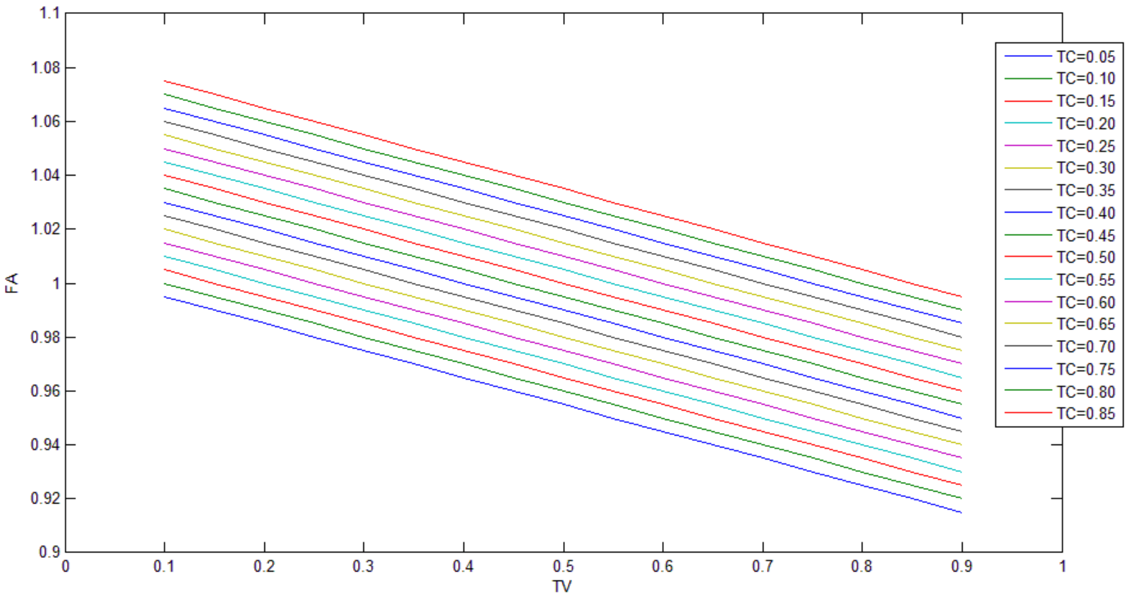
A continuación, se detallan los Factores de Amplificación mínimos para cada método y los valores de TC y TV que lo minimizan:

Método de Euler

Para obtener los valores de TC y TV que hacen mínimo el factor de amplificación, debemos ejecutar la rutina

> *[FAmin, TCmin, TVmin, matriz] = minimoEuler(0.1)*

El Factor de amplificación mínimo es 0.9150, y está asociado a TC = 0.0500 y TV = 0.9000.

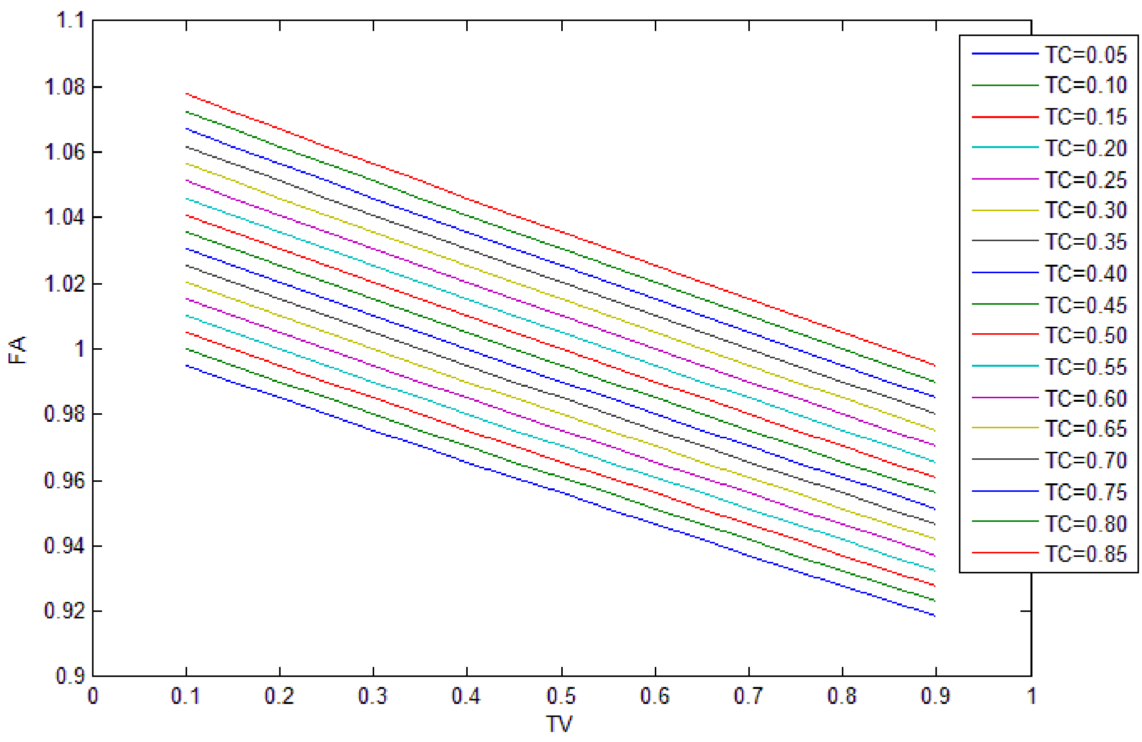


Método Runga Kutta de orden 2

Para obtener los valores de TC y TV que hacen mínimo el factor de amplificación, debemos ejecutar la rutina

[FAmin, TCmin, TVmin, matriz] = minimoRK2(0.1)

El Factor de amplificación mínimo es 0.9186, y está asociado a TC = 0.0500 y TV = 0.9000.

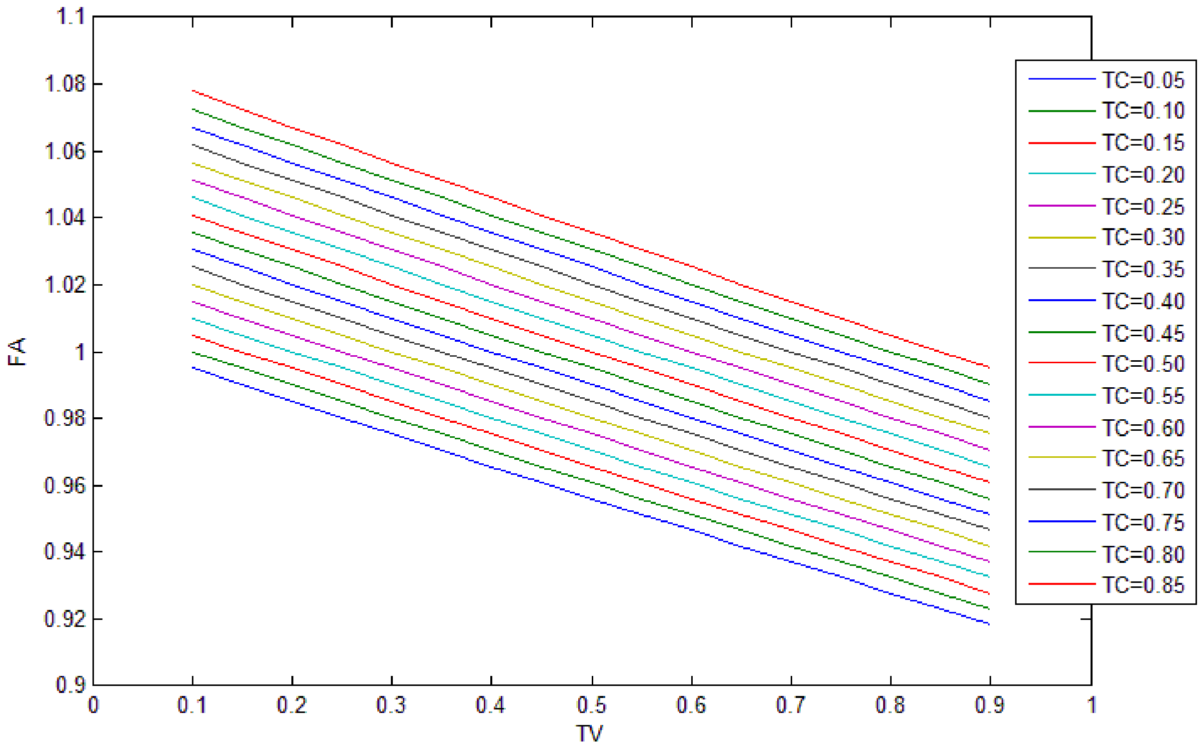


Método Runga Kutta de orden 4

Para obtener los valores de TC y TV que hacen mínimo el factor de amplificación, debemos ejecutar la rutina

[FAmin, TCmin, TVmin, matriz] = minimoRK4(0.1)

El Factor de amplificación mínimo es 0.9185, y está asociado a TC = 0.0500 y TV = 0.9000.



Observaciones:

En los gráficos anteriores, puede notarse que cuando aumento TC (y también cuando disminuyo TV) aumenta el Factor de Amplificación.

También es de destacar que los 3 factores de amplificación tienen módulo menor que 1, motivo por el cual se puede afirmar que los 3 métodos son estables.

RESOLUCIÓN DE LA ECUACIÓN DIFERENCIAL CON FA MÍNIMO

Volvemos a resolver la ecuación diferencial con cada método utilizando los valores de TC y TV que hacen mínimo el factor de Amplificación

Euler

Para obtener la solución numérica de la ecuación diferencial, debemos ejecutar la rutina

[t, u]= Euler(0.1, 0, 1.2, 0.05, 0.9, 100)

t	x
0	100.0000000000000
0.1000	91.5000000000000
0.2000	83.7225000000000
0.3000	76.6060875000000
0.4000	70.0945700625000
0.5000	64.1365316071875
0.6000	58.6849264205766
0.7000	53.6967076748275
0.8000	49.1324875224672
0.9000	44.9562260830575
1.0000	41.1349468659976
1.1000	37.6384763823878
1.2000	34.4392058898848

Runga Kutta de orden 2

Para obtener la solución numérica de la ecuación diferencial, debemos ejecutar la rutina

[t, U] = RK2(0.1, 0, 1.2, 0.05, 0.9, 100)

t	x
0	100.0000000000000
0.1000	91.8612500000000
0.2000	84.3848925156250
0.3000	77.5170170760096
0.4000	71.2081008487358
0.5000	65.4126515409094
0.6000	60.0888793636236
0.7000	55.1983956944167
0.8000	50.7059362648373
0.9000	46.5791068770829
1.0000	42.7881498161243
1.1000	39.3057292729645
1.2000	36.1067342317611

Runga Kutta de orden 4

Para obtener la solución numérica de la ecuación diferencial, debemos ejecutar la rutina

[t, U] = RK4(0.1, 0, 1.2, 0.05, 0.9, 100)

t	x
0	100.0000000000000
0.1000	91.8512320859375
0.2000	84.3664883570475
0.3000	77.4916590235872
0.4000	71.1770435769984
0.5000	65.3769914878177
0.6000	60.0495721822790
0.7000	55.1562719117577
0.8000	50.6617153236193
0.9000	46.5334097206145
1.0000	42.7415101599818
1.1000	39.2586036940795
1.2000	36.0595111927474

Método predefinido en MATLAB **ode23**

Para obtener la solución numérica de la ecuación diferencial, debemos ejecutar la rutina

[x, y]= ode23(@FuncionFAmin , [0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0, 1.1, 1.2], 100)

t	x
0	100.0000000000000
0.1000	91.8510574979709
0.2000	84.3659277279991
0.3000	77.4908133112730
0.4000	71.1759909389062
0.5000	65.3757824069987
0.6000	60.0482320281939
0.7000	55.1548042049056
0.8000	50.6601268635999
0.9000	46.5317520808791
1.0000	42.7398219602669
1.1000	39.2569076135044
1.2000	36.0578397150207

Método predefinido en MATLAB **ode45**

Para obtener la solución numérica de la ecuación diferencial, debemos ejecutar la rutina

[x, y]= ode45(@FuncionFAmin , [0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0, 1.1, 1.2], 100)

t	x
0	100.0000000000000
0.1000	91.8512284512515
0.2000	84.3664815663582
0.3000	77.4916494900943
0.4000	71.1770318652097
0.5000	65.3769782046215
0.6000	60.0495578225413
0.7000	55.1562566958593
0.8000	50.6616992746071
0.9000	46.5333930000689
1.0000	42.7414930273954
1.1000	39.2585864408716
1.2000	36.0594941421796

Resolución de la ecuación analítica

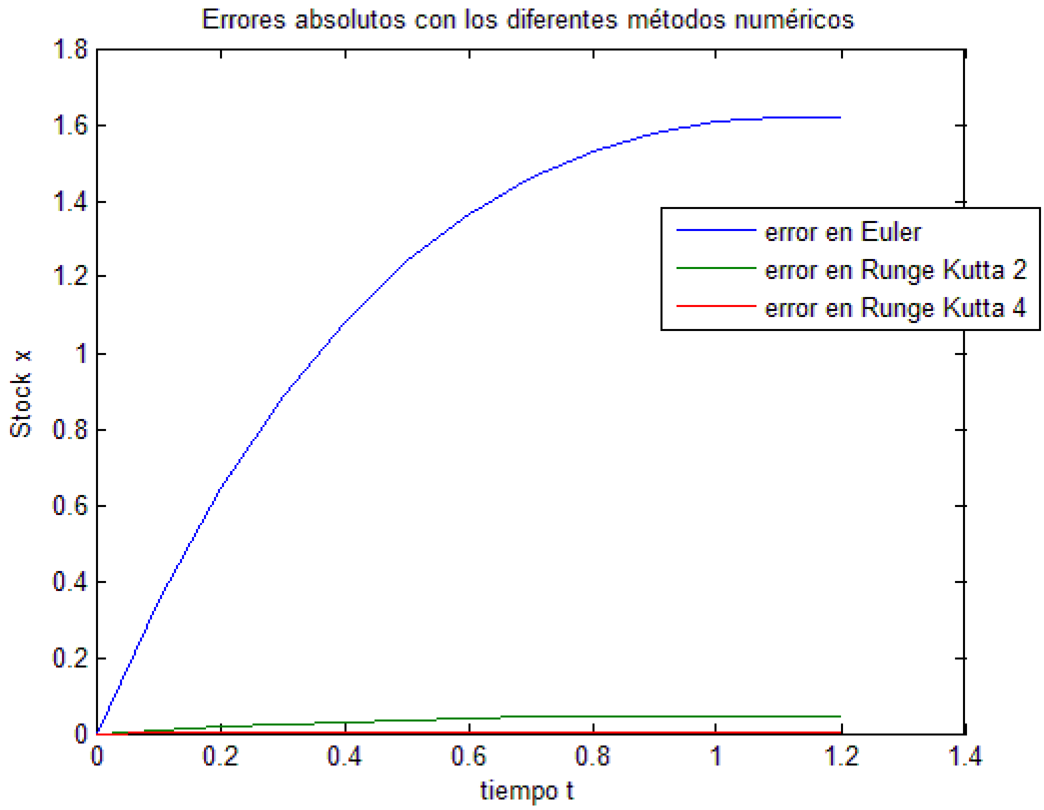
Para invocar a la función que evalúa la función que es solución de la ecuación diferencial (obtenida en forma analítica), y por ende nos otorga los valores verdaderos, debemos escribir en la línea de comandos:

analitica(0.1, 0, 1.2, 0.05, 0.9)

t	x
0	100.0000000000000
0.1000	91.8512284401457
0.2000	84.3664816596384
0.3000	77.4916497961081
0.4000	71.1770322762610
0.5000	65.3769785129847
0.6000	60.0495578812266
0.7000	55.1562565867830
0.8000	50.6616992365590
0.9000	46.5333930974313
1.0000	42.7414931948727
1.1000	39.2585865531518
1.2000	36.0594940173078

Gráfico: Comparativa de Errores con los métodos numéricos

A continuación se muestra un gráfico que nos permite apreciar la diferencia entre los errores absolutos en los distintos métodos, al resolver la ecuación con los valores de TC y TV que hacen mínimo el Factor de Amplificación



Observación:

Puede notarse que la mejoría más significativa se consigue al pasar del método de Euler (que es de orden 1) al método de Runge Kutta de orden 2.

CONCLUSIONES

Con este trabajo práctico logramos apreciar:

Que los métodos de Eurler, Runge-Kutta de orden 2, y Runge Kutta de orden 4 son relativamente fáciles de implementar, y en este caso fueron muy efectivos.

La gran precisión con la que se pueden resolver ecuaciones diferenciales mediante métodos numéricos. Es notorio ver, que aunque se utilice un paso h = 0.1, el resultado final difiere muy poco del valor verdadero.

Que al aumentar el orden del método obtenemos ventajas muy significativas en el resultado (pasamos de un Error de 10^{-1} en Euler a 10^{-7} en Runge Kutta de orden 2)

Que en el caso particular de la interpretación de la ecuación diferencial del presente trabajo práctico, como el valor de **x** representa una cantidad en stock de un producto, que es un valor entero, no requerimos tener muchos decimales significativos en el resultado final, sino que basta con que la parte entera podamos redondearla bien, ya que los decimales no tendrían ninguna interpretación física coherente de acuerdo al enunciado dado.

Que el método de Euler nos subestimó el valor de la solución mientras que los métodos de Runge Kutta lo sobreestimaron.

Que al volver a resolver la ecuación diferencial con los valores de TC y TV que hacen mínimo el Factor de Amplificación, obtuvimos más error. Esto se debe a que estamos cambiando el problema matemático y por ende también el problema numérico: La función decrece más abruptamente, y por ende la derivada tiene un valor más grande en módulo: eso hace que el error en cada paso sea considerable (puede verse que en el último paso de Euler, el error vale aproximadamente 1,6).

Que si la función tiene un comportamiento de crecimiento o decrecimiento abrupto, es aconsejable elegir un método de mayor orden para resolverla numéricamente, ya que por más que tengamos un Fa mínimo, si tenemos que avanzar varios pasos en la resolución de la ecuación diferencial, los errores empiezan a ser significativos.