

**“INSTITUTO POLITÉCNICO NACIONAL”**

**“UNIDAD PROFESIONAL INTERDISCIPLINARIA DE BIOTECNOLOGÍA”**

**TAREA VII-SOLUCIÓN NUMÉRICA DE EDO DE PRIMER ORDEN CON CONDICIONES INICIALES**

**PROGRAMA ACADÉMICO:** INGENIERÍA BIOMÉDICA

**UNIDAD DE APRENDIZAJE:** MÉTODOS NUMÉRICOS

**PROFESORA.** CARMEN MARÍN ALBINO

**PROFESOR.** JORGE LUIS ROSAS MENDOZA

**INTEGRANTES/ALUMNO:**

-ABAD JUÁREZ OSCAR ALFREDO

-BARRÓN RODRÍGUEZ DAFNE VICTORIA

-DORANTES CAÑAS ISAAC ALÍ

-HERNÁNDEZ HERNÁNDEZ MARISOL

-MORENO HERRERA ALEJANDRO

-NORIEGA GASPAR EDGAR LEONARDO

**GRUPO:** 4MM1 **EQUIPO:** 8

***MEXICO, D.F A 06 DE DICIEMBRE DE 2016***

**Problema 1.** Para la siguiente ecuación diferencial ordinaria (EDO)

1. Encuentre la solución analítica (puede usar el comando dsolve).
2. Resuélvalas numéricamente aplicando los *Métodos de Runge-Kutta del Primer Orden (Método de Euler) al Cuarto Orden* empleando el valor de en el intervalo indicado.
3. Coloque la solución exacta y las soluciones aproximadas en una sola gráfica. Anexe una ampliación de la gráfica en el que se aprecien las diferentes aproximaciones numéricas.

*Nota: En éste ejercicio NO imprima la solución numérica.*

Tenemos la siguiente ecuación diferencial

Despejamos de la forma

Obteniendo:

B)

1.- Primero resolvemos por el método de Euler

clc;clear all;close all

format long

%empezamos por el método de euler

%SIEMPRE HAY QUE DESPEJAR A LA DERIVADA

%primero necesito la función

f=inline('t^2\*exp(-t)-(2\*x)/2','t','x')

%2do, necesito el paso

h=0.005;

%por ultimo ncesito la 'semilla'

t(1)=0;x(1)=exp(1)/sqrt(2);

i=1;%este es el paso 1

while x(i)>=0.001

t(i+1)=t(i)+h;

x(i+1)=x(i)+h\*f(t(i),x(i));

i=i+1;

end

solucion\_euler=[t',x'];

plot(t,x,'o');t=1:2;grid on

Ahora resolvemos por el método de Rounge-Koutta de 4to orden

clear x t

f=inline('t^2\*exp(-t)-(2\*x)/2','t','x')

%2do, necesito el paso

h=0.005;

%por ultimo ncesito la 'semilla'

t(1)=0;x(1)=exp(1)/sqrt(2);

i=1;%este es el paso 1

while x(i)>=0.001

t(i+1)=t(i)+h;

k1=h\*f(t(i),x(i));

k2=h\*f(t(i)+h/3,x(i)+k1/3);

k3=h\*f(t(i)+2\*h/3,x(i)+k1/3+k2/3);

k4=h\*f(t(i+1),x(i)+k1-k2+k3);

x(i+1)=x(i)+1/8\*(k1+3\*k2+3\*k3+k4);

i=i+1;

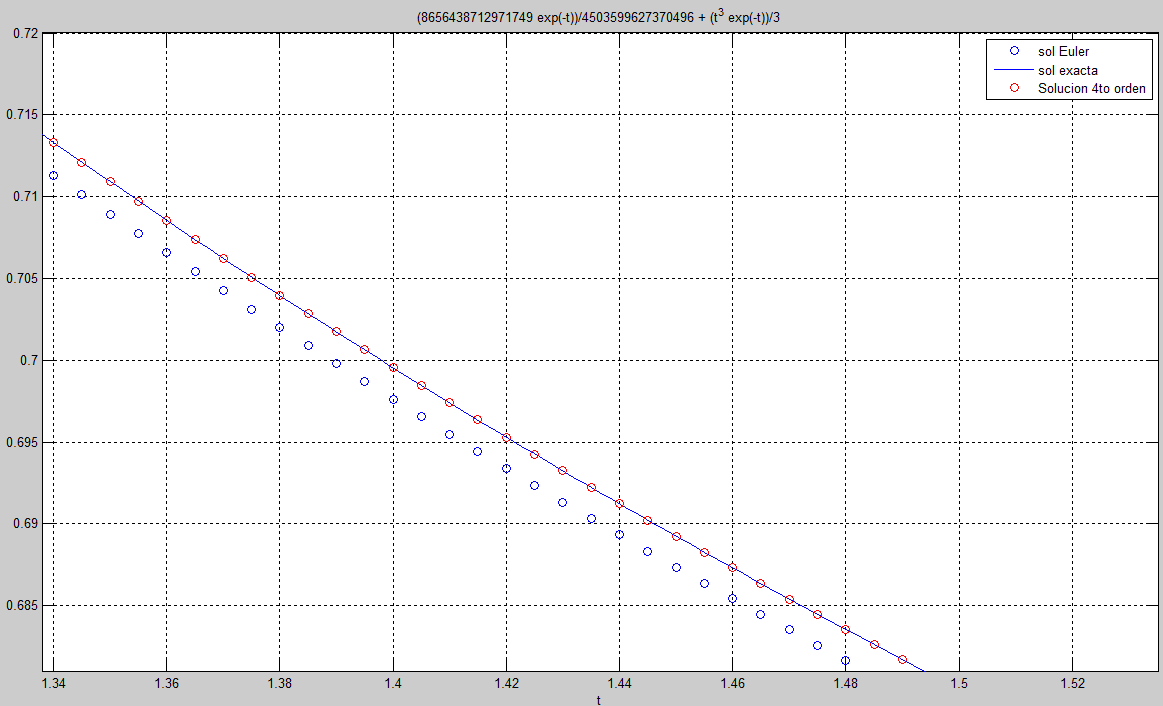
end

solucion\_rk2=[t',x'];

plot(t,x,'or')

legend('sol Euler','sol exacta','Solucion 4to orden')

Las gráficas con las soluciones numéricas y la analíticas son:



Como podemos observar la solución por el método RK4 resulto ser más exacto a la solución analítica resuelta con comandos de MatLab.

**Problema 2.** Para la siguiente ecuación diferencial ordinaria (EDO) encuentre algunos elementos de la *“familia de soluciones”,* considerando las diferentes condiciones iniciales:

.

.

.

.

.

**Solución.** Se puede obtener la familia de soluciones para la ecuación diferencial ordinaria atendiendo al hecho de que es una de las formas de analizar numéricamente a la ecuación, soluble por el método de RK4, programado como se muestra a continuación:

f=inline('1-t\*y','t','y') %la función depende de dos variables

t(1)=-2; y(1)=-2 **ó -1 ó 0 ó 1 ó 2**; %son las condiciones iniciales

i=1;

h=0.005;

while t(i)>=-2 && t(i)<=3.5 %establece el dominio de la función

t(i+1)=t(i)+h;

k1=h\*f(t(i),y(i));

k2=h\*f(t(i)+h/3,y(i)+k1/3);

k3=h\*f(t(i)+2\*h/3,y(i)+k1/3+k2/3);

k4=h\*f(t(i+1),y(i)+k1-k2+k3);

y(i+1)=y(i)+1/8\*(k1+3\*k2+3\*k3+k4);

i=i+1;

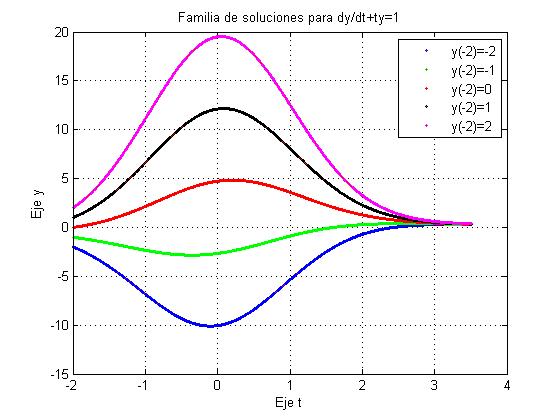
end

plot(t,y,'mo','markersize',1,'markerfacecolor','r'),grid on,xlabel('Eje t'),ylabel('Eje y'),title('Familia de soluciones para dy/dt+ty=1')

%colocar un *hold on* luego del primer corrimiento

%se irán graficando sobre la misma figura

La familia de soluciones graficada en una misma figura se muestra a continuación:



**Problema 3.** Sea el número de individuos de una población en el tiempo , medido en **años**. Si la tasa de natalidad promedio es constante y la tasa de mortalidad promedio es proporcional al tamaño de la población (debido a la superpoblación),

entonces la tasa de crecimiento demográfico estará dada por la *ecuación logística*

La población actual en México (fuente: **INEGI 2016**) es de habitantes. Se sabe que la tasa de natalidad es de nacimientos por cada habitantes, es decir y la tasa de mortalidad es de 5.7 defunciones por cada 1000 habitantes, es decir (fuente: **INEGI 2016,** datos actualizados al 1 de diciembre de 2016 basados en el censo del 2010).

Calcule empleando el valor de la población y la tasa de mortalidad actuales, y suponiendo que y han permanecido constantes desde el censo del 2010, estime la población actual (2016) y haga una proyección de la población para los años 2020 y 2050. ¿Qué opina de los resultados?

Aplique el *Método de Euler,* con , para encontrar la solución numérica. *Imprima únicamente los valores de la población cada 5 años*.

**Resolución**

Para el caso de estimación de la población en año 2016, se tiene el siguiente código:

clc; clear all; close all;

format long;

%La tasa de mortalidad según la INEGI en 2015 fue de

d=0.0057;

%El tamaño de paso para los cálculos será cada 0.00025 años

h=0.00025;

%El primer dato se guarda en el primer espacio del array

P(1)=112336538;

%Según la formula dada de la tasa de mortalidad, se despeja k y estonces

%que se obtiene su valor

k=d/P(1)

%Finalmente, para obtener la cantidad de Población dentro de algunos años,

%se ingresa la función proporcionada

f=inline('0.00019\*P-5.074039223106555e-011\*P^2', 't', 'P');

%Se inicializa nuestro contador en 1

i=1;

%Referimos que nuestro primer tiempo dado es del año 2015

t(1)=2015;

%Al querer obtener la población en el año 2016, condicionamos nuestro bucle

%hasta dicho año

while t(i)<=2016

t(i+1)=t(i)+h;

P(i+1)=P(i)+h\*f(t(1),P(1));

i=i+1;

end

%Se imprimen la cantidad de habitantes en México en el año 2015 y 2016

%indicado por nuestro rango, donde 1 indica por cada año

for i=1:1/h:length(t)

fprintf('%3.2f \t %1.4f \n', t(i), P(i))

end

Los valores que imprime el programa son los siguientes:

|  |  |
| --- | --- |
| Año | Población |
| 2015 | 112336538.0000 |
| 2016 | 111717563.6756 |

Para el caso de estimación de la población en el año 2020 cada 5 años, tan sólo se modifica la condición en nuestro bucle while y el intervalo de proyección para nuestro ciclo for, obteniendo el siguiente código:

while t(i)<=2020

t(i+1)=t(i)+h;

P(i+1)=P(i)+h\*f(t(1),P(1));

i=i+1;

end

for i=1:5/h:length(t)

fprintf('%3.2f \t %1.4f \n', t(i), P(i))

end

Los valores que imprime el programa son los siguientes:

|  |  |
| --- | --- |
| Año | Población |
| 2015 | 112336538.0000 |
| 2020 | 109241666.3780 |

Y finalmente, para el caso de estimación de la población hasta el año 2050 cada 5 años, se modifica la condición en nuestro bucle while y el intervalo de proyección para nuestro ciclo for, obteniendo el siguiente código:

while t(i)<=2050

t(i+1)=t(i)+h;

P(i+1)=P(i)+h\*f(t(1),P(1));

i=i+1;

end

for i=1:5/h:length(t)

fprintf('%3.2f \t %1.4f \n', t(i), P(i))

end

Los valores que imprime el programa son los siguientes:

|  |  |
| --- | --- |
| Año | Población |
| 2015 | 112336538.0000 |
| 2020 | 109241666.3780 |
| 2025 | 106146794.7559 |
| 2030 | 103051923.1339 |
| 2035 | 99957051.5119 |
| 2040 | 96862179.8899 |
| 2045 | 93767308.2678 |
| 2050 | 90672436.6458 |

La proyección de los habitantes al paso de los años indica un decremento, en base a los datos de la INEGI, la tasa de natalidad es más de tres veces mayor que la tasa de mortalidad, por lo que se esperaría un incremento poblacional, sin embargo, en base a la ecuación diferencial, se obtiene un decremento y se atribuye a que la constante de defunción poblacional es relativamente baja. Sí en 2015 hubo mayores nacimientos que muertes, entonces es de esperar que, acorde a la estimación de vida mayor a 60 años, en el año 2050 los habitantes que nacieron en 2015 aún vivan.

**Problema 4.** Un tanque cilíndrico de 5 m de diámetro y 11 m de largo aislado con asbesto se carga con un líquido que está a 220 °C y el cual se deja reposar durante 5 días. A partir de los datos del diseño del tanque, las propiedades térmicas y físicas del líquido, y el valor de la temperatura ambiente, se encuentra la siguiente ecuación que relaciona la temperatura del líquido (en °C) con el tiempo en horas.

¿Cuál es la temperatura final del líquido? Grafique el comportamiento de la temperatura con respecto al tiempo. Aplique el *Método de Runge-Kutta de 2o. Orden,* con , para encontrar la solución numérica.

**Solución.** Se sabe que pasarán 120 horas desde que se dejó reposando al tanque cilíndrico, además, la ecuación diferencial ordinaria es soluble muy fácilmente ya que se puede obtener que la derivada de la temperatura respecto del tiempo se tendrá una función de dos variables, f(t,T), según el orden de las variables independiente y dependiente, luego: **,** con el programa ya desarrollado se tendrá, por el método RK2:

clear all; close all; clc

f=inline('0.615+0.175\*cos(pi\*t/12)-0.0114\*T','t','T')

t(1)=0; T(1)=220; %se establecen las condiciones iniciales

i=1;

h=0.005;

while t(i)<=120 %se determina el tiempo de reposo del tanque

t(i+1)=t(i)+h;

k1=h\*f(t(i),T(i));

k2=h\*f(t(i+1),T(i)+k1);

T(i+1)=T(i)+1/2\*(k1+k2);

i=i+1;

end

for j=1:120/h:length(T) %se delimita el rango de "toma de lectura"

fprintf('%3.2f \t %1.4f \n',t(j),T(j)) %se imprimen los resultados

end

Los resultados que se muestran a continuación fueron los devueltos por el programa cada 24 horas, por lo que se esperan seis resultados, efectivamente:

|  |  |
| --- | --- |
| **Tiempo (horas)** | **Temperatura (°C)** |
| 0.00 | 220.0000 |
| 24.00 | 180.2600 |
| 48.00 | 150.0323 |
| 72.00 | 127.0400 |
| 96.00 | 109.5512 |
| 120.00 | 96.2486 |

La serie de resultados es lógica con la física del problema, ya que se espera que el líquido disminuya su temperatura conforme avanza el tiempo, la gráfica que puede demostrar que el comportamiento es consistente, se puede realizar simplemente si:

plot(t,T,'ro','markerfacecolor','g','markersize',1),grid on, xlabel('Tiempo (horas)'),ylabel('Temperatura (°C)'), title('Temperatura de líquido en tanque cilíndrico')

Igualmente puede graficarse la solución particular de la ecuación diferencial, obteniendo el valor de la solución exacta por el siguiente código y su posterior gráfica:

hold on

syms T(t)

sE=dsolve(diff(T)==0.615+0.175\*cos(pi\*t/12)-0.0114\*T,T(0)==220);

ezplot(sE,[-10:140])

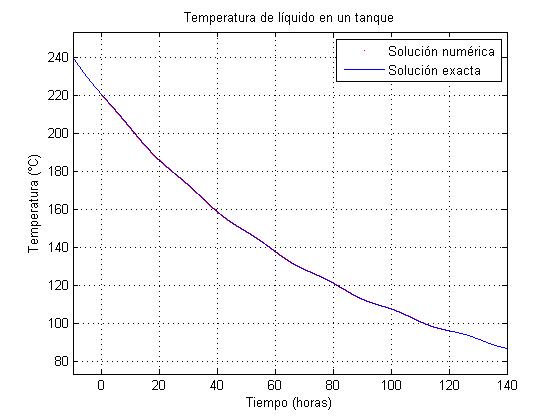
legend('Solución numérica','Solución exacta') syms T(t)

sE=dsolve(diff(T)==0.615+0.175\*cos(pi\*t/12)-0.0114\*T,T(0)==220);

ezplot(sE,[-10:140])

legend('Solución numérica','Solución exacta')

Se tiene, pues, una gráfica con una similitud muy grande con la solución exacta de la ecuación diferencial, es necesario hacer un gran acercamiento a la gráfica para notas las diferencias:



**Problema 6:** En un circuito de voltaje impreso que tiene la resistencia , la inductancia y la capacitancia en paralelo, la corriente satisface la ecuación diferencial

Suponga que y que el voltaje está dado por

Si , calcule la corriente a los 20 s. Grafique . Emplee el *Método de Runge-Kutta de 4o. Orden*, con , para encontrar la solución numérica. *Imprima únicamente los valores de la corriente cada 2 s*. ***¡Explique su solución!***

**Solución:**

**Primero hice un script para calcular la función mediante los siguientes códigos:**

clc;clear all;close all;

%Insertamos los valores conocidos y potemos simbólica la t para poder

%Derivarla

C=0.3;R=1.4;L=1.7;

syms t

E=exp(-0.06\*pi\*t)\*sin(2\*t-pi)

Primera\_Derivada=diff(E,1,t)

Segunda\_Derivada=diff(E,2,t)

%y aquí obtenemos la función que después ingresaremos a otro script

f=C\*Segunda\_Derivada+1/R\*Primera\_Derivada+1/L\*E

**Y MATLAB nos arroja:**

f=(52\*sin(2\*t)\*exp(-(3\*pi\*t)/50))/85 - (10\*cos(2\*t)\*exp(-(3\*pi\*t)/50))/7 + (9\*pi\*cos(2\*t)\*exp(-(3\*pi\*t)/50))/125 + (3\*pi\*sin(2\*t)\*exp(-(3\*pi\*t)/50))/70 - (27\*pi^2\*sin(2\*t)\*exp(-(3\*pi\*t)/50))/25000

**Esta ya es nuestra función y procederemos a meterla en el siguiente script para calcular la corriente a los 20 segundos, graficar y encontrar la solución numérica solicitada**

clc;clear all;close all;

f=inline('(52\*sin(2\*t)\*exp(-(3\*pi\*t)/50))/85 - (10\*cos(2\*t)\*exp(-(3\*pi\*t)/50))/7 + (9\*pi\*cos(2\*t)\*exp(-(3\*pi\*t)/50))/125 + (3\*pi\*sin(2\*t)\*exp(-(3\*pi\*t)/50))/70 - (27\*pi^2\*sin(2\*t)\*exp(-(3\*pi\*t)/50))/25000','t','I')

pause

%Ponemos el h de 0.004

h=0.004;

t(1)=0;I(1)=0;i=1;

%En nuestra condición tenemos que mientras el tiempo sea menor a 20 el ciclo continuará y cuando llegue a 20 termina y tenemos nuestra solución

while t(i)<20

t(i+1)=t(i)+h

k1=h\*f(t(i),I(i));

k2=h\*f(t(i)+h/3,I(i)+k1/3);

k3=h\*f(t(i)+2\*h/3,I(i)+k1/3+k2/3);

k4=h\*f(t(i)+1,I(i)+k1-k2+k3);

I(i+1)=I(i)+1/8\*(k1+3\*k2+3\*k3+k4);

i=i+1;

end

Sol\_Euler=[t' I']

%Graficamos nuestra solución

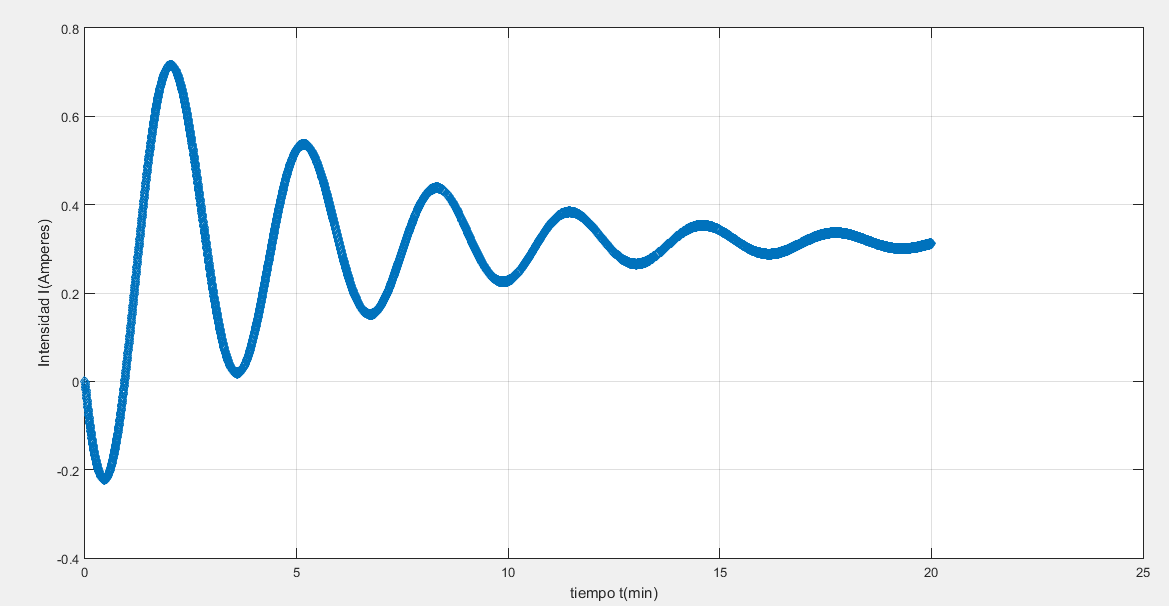
plot(t,I,'d');grid on;xlabel('tiempo t(min)');ylabel('Intensidad I(Amperes)');

%Después imprimimos los valores cada dos segundos

for i=1:2/h:length(t)

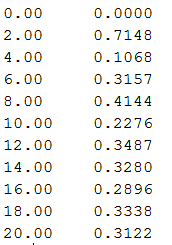
fprintf('%3.2f \t %1.4f \n',t(i),I(i))

end

**Y MATLAB nos arroja: **

**SOLUCIÓN:**

**-En intervalos de 2 segundos hasta llegar al resultado**

****

**Y normalmente en un circuito este tipo de comportamiento es una respuesta amortiguada.**

**Problema 7.** Un proyectil de masa se lanza verticalmente hacia arriba con una velocidad inicial y se va frenando debido ala fuerza de gravedad y a la resistencia del aire donde y . La ecuación diferencial para la velocidad está dada por

Estime el tiempo que tarda en alcanzar la altura máxima. Grafique el perfil de la velocidad desde que parte del suelo hasta que regresa. Use el *Método de Euler,* con para encontrar la solución numérica. *Imprima únicamente los valores de la velocidad cada 1 s.*

Despejamos de la forma

Obteniendo:

Sustituyendo valores.

%problema del proyectil

clc;clear all;close all

format long

%empezamos por el método de euler

%SIEMPRE HAY QUE DESPEJAR A LA DERIVADA

%primero necesito la función

f=inline('-9.8-.002\*v^2/0.11','t','v')

%2do, necesito el paso

h=0.001;

%por ultimo ncesito la 'semilla'

t(1)=0;v(1)=80;

i=1;%este es el paso 1

while v(i)>=0.00001

t(i+1)=t(i)+h;

v(i+1)=v(i)+h\*f(t(i),v(i));

i=i+1;

end

solucion\_euler=[t',v'];

plot(t,v,'or');grid on;

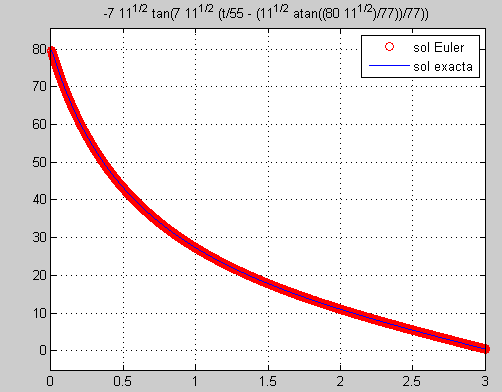
syms v(t)

sol\_ex=dsolve(diff(v)==-9.8-.002\*v^2/0.11,v(0)==80)

hold on

ezplot(sol\_ex,[0,3.])

legend('sol Euler','sol exacta')



Impresión cada 1 segundos

for i=1:1/h:length(t)

fprintf('%3.2f \t%1.4f\n',t(i),v(i))

end

La solución para cada segundo es:

**Tiempo(segundos) Altura (metros)**

0.00 80.0000

1.00 27.2899

2.00 11.0327

3.00 0.4984

**Problema 8:** Un circuito eléctrico consta de un capacitor de capacitancia constante faradios, que está en serie con un resistor de resistencia constante ohms. Se aplica un voltaje en el tiempo . Cuando el resistor se calienta, la resistencia se transorma en una función de la corriente :

donde , y la ecuación diferencial de se convierte en

Calcule suponiendo que Aplique el *Método de Runge-Kutta de 2o. Orden,* con Grafique la solución para y muestre la solución cada 1 s.

**Solución:**

Sabiendo que la derivada de 110sin(t) es 110cos(t) y sustituyendo valores , despejando di/dt obtenemos la siguiente función=

Por comodidad la intensidad del problema será I para no confundirla con la del ciclo.

clc;clear all;close all;

f=inline('(110\*cos(t)-I)/(2.1\*1.1+2\*0.9\*1.1\*I)','t','I')

pause

h=0.0001;

t(1)=0;I(1)=0;i=1;

while t(i)<10

t(i+1)=t(i)+h;

k1=h\*f(t(i),I(i));

k2=h\*f(t(i+1),I(i)+k1);

I(i+1)=I(i)+1/2\*(k1+k2);

i=i+1;

end

Sol\_Euler=[t' I']

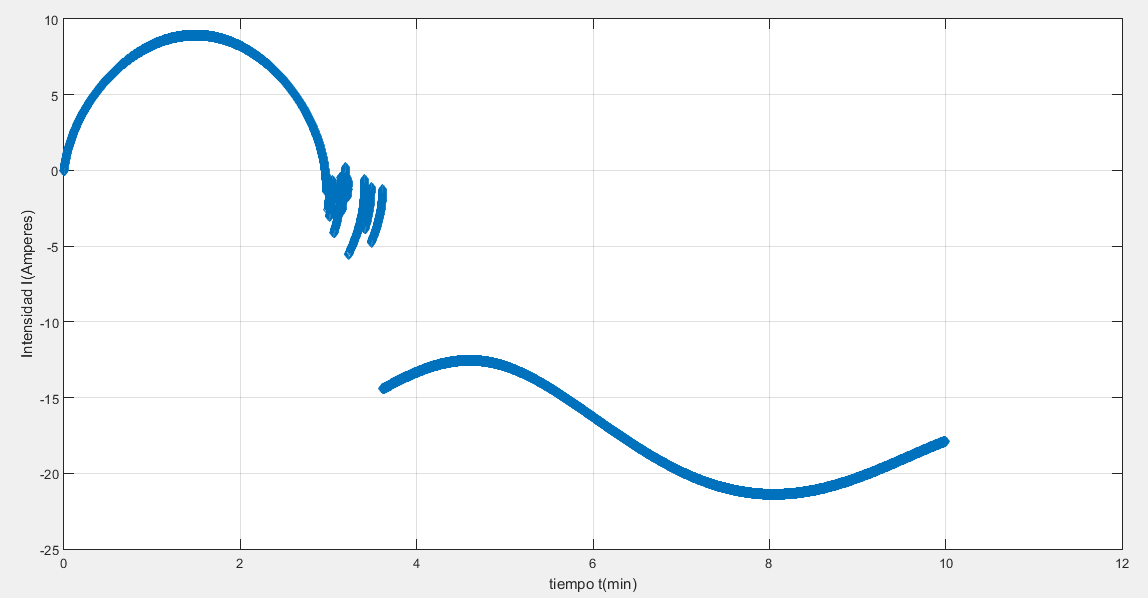
plot(t,I,'d');grid on;xlabel('tiempo t(min)');ylabel('Intensidad I(Amperes)');

for i=1:1/h:length(t)

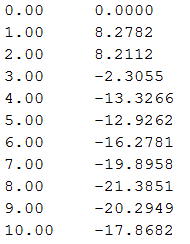
fprintf('%3.2f \t %1.4f \n',t(i),I(i))

end

**Gráfica arrojada por MATLAB:**



**SOLUCIÓN A CADA SEGUNDO DE TIEMPO:**



**El comportamiento del circuito es debido a la relación que existe la intensidad y la resistencia debido a que tienen una relación directamente proporcional, no como la ley de OHM convencional que es una relación inversamente proporcional. , Modelo de aproximación válido para tiempos menores a 3 segundos t<3 segundos**