NSTITUTO POLITÉCNICO NACIONAL

Unidad Profesional Interdisciplinaria de Biotecnología

**Unidad de Aprendizaje**: Métodos Numéricos

**Tarea No 2 - 3° parcial.**

*“Solución Numérica de EDO'S de Primer Orden con condiciones iniciales.”*

**Profesora:**

Marin Albino María del Carmen

**Alumnos:**

Escalante Villalba Alexa

Minajas Carbajal Francisco Javier

Mireles Pérez María Caridad

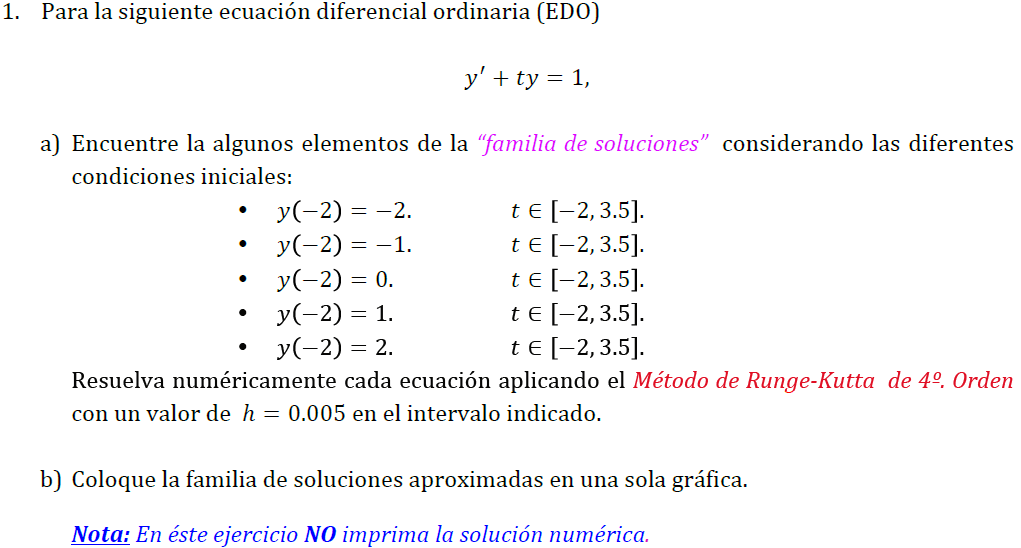
Salmerón Ramírez Amanda

**Grupo:** 4FV3

**Fecha de entrega:** 28/11/2017

Equipo 9

**Ciclo escolar:** 2018/1

clear all; close all; clc;

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

t(1)=-2; y(1)=-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

solucion\_rk4=[t',y']

plot(t,y, 'o', 'markerfacecolor', 'r');

hold on

%se irán graficando sobre la misma figura

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

t(1)=-2; y(1)=-1; %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

solucion\_rk4=[t',y'];

plot(t,y, 'o', 'markerfacecolor', 'b')

hold on

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

t(1)=-2; y(1)=0; %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

solucion\_rk4=[t',y']

plot(t,y, 'o', 'markerfacecolor', 'g');

hold on

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

t(1)=-2; y(1)=1; %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

solucion\_rk4=[t',y']

plot(t,y, 'o', 'markerfacecolor', 'y');

hold on

%se irán graficando sobre la misma figura

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

t(1)=-2; y(1)=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;

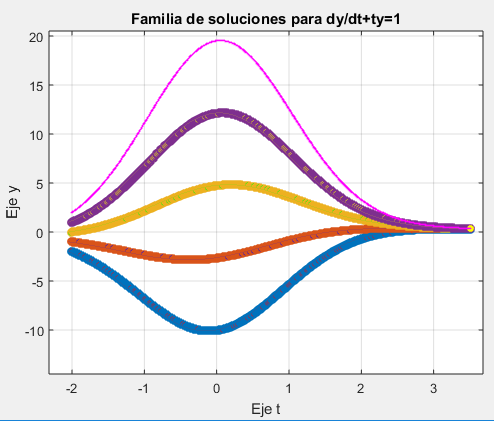
end

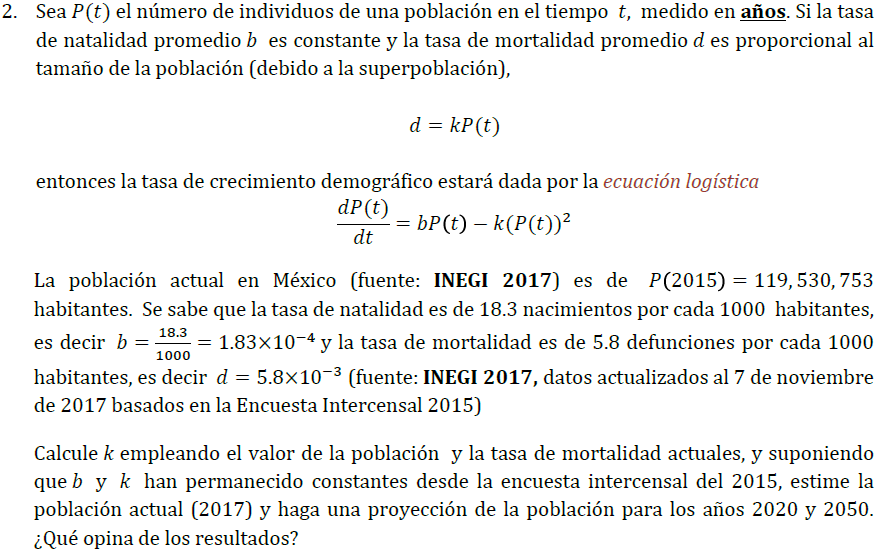
solucion\_rk4=[t',y'];

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





**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 2017, condicionamos nuestro bucle

%hasta dicho año

while t(i)<=2017

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 |
| 2017 | 111098589.3512 |

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)

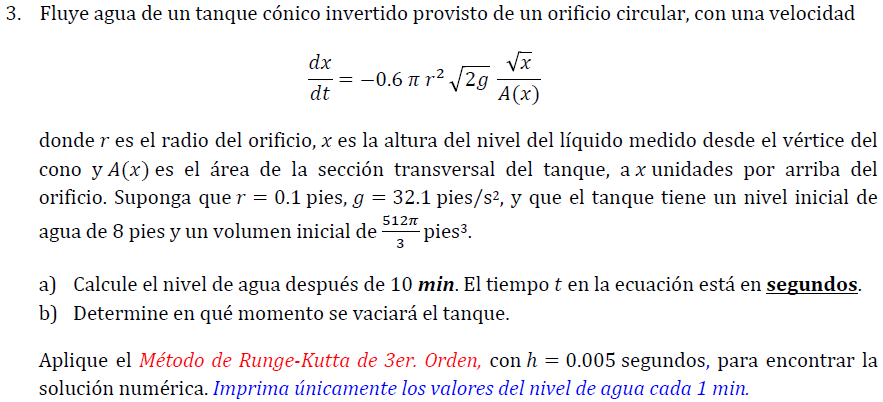
fprintf('%3.2f \t %1.4f \n', t(i), P(i))

end

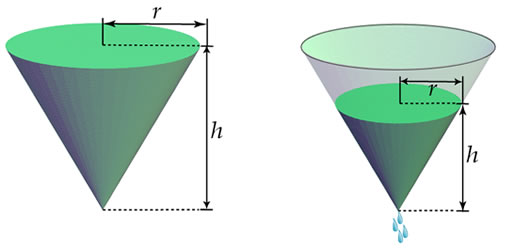
Los valores que imprime el programa son los siguientes:

|  |  |
| --- | --- |
| **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.



Sabemos que un tanque cónico es de la forma:



**Datos;**

Primeramente establecemos la relación de triángulos que existe:

Una vez que contemos con el volumen a los 10 min, utilizaremos la relación anterior y obtendremos los datos solicitados en el inciso a y b, que se analizarán a continuación con ayuda del programa elaborado en MATLAB.

Primeramente escribimos el modelo a analizar:

Como datos tenemos:

Por otro lado sabemos que:

El área entonces está dada por:

La sustituimos en nuestro modelo y queda:

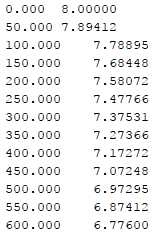
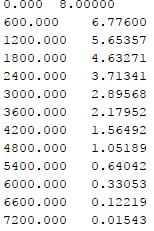
Para finalmente quedarnos:

Calculemos el área inicial:

Con ayuda del programa obtuvimos los siguientes datos:

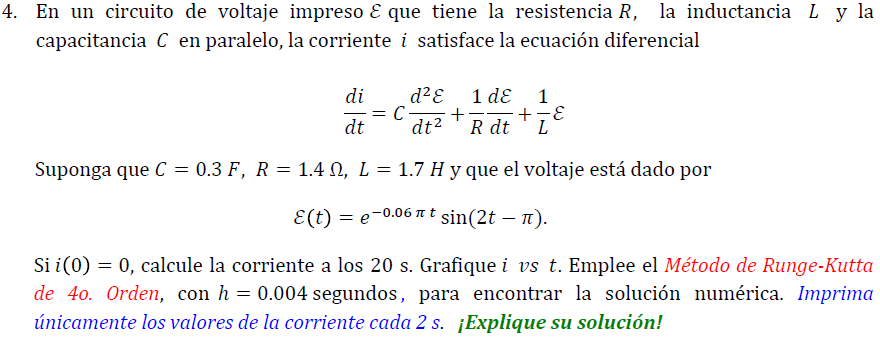
* Después de 10 min el nivel de agua está a una altura de:
* El tanque se vacía en el tiempo de:

**Pantallazos**:

**Codigo:**

1. %--------------------------------EULER---------------------------------%
2. clc;
3. clear all;
4. %funcion de dos variables
5. r=0.1;
6. g=32.1;
7. x=8;
8. A=64\*pi;
9. %A=64\*pi;
10. f=@(t,x)-0.6\*pi\*r^2\*sqrt(2\*g)\*(sqrt(x)/A);
11. %segundo ingrediente condicion inicial
12. %primer paso
13. t(1)=0;
14. x(1)=8;
15. h=0.005;
16. i=1;
17. %ciclo while para el calculo de los parametros
18. while t(i)<=600
19. t(i+1)=t(i)+h;
20. k1=h\*f(t(i),x(i));
21. k2=h\*f(t(i)+h/2,x(i)+k1/2);
22. k3=h\*f(t(i),x(i)-k1+2\*k2);
23. x(i+1)=x(i)+1/6\*(k1+4\*k2+k3);
24. i=i+1;
25. end
26. SOL\_K\_3=[t',x'];
27. plot(t,x,'g\*');
28. grid on
29. for i=1:50/h:length(t)
30. fprintf(' %3.3f\t%3.5f\n',t(i),x(i));
31. end



**Solución:**

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

clc;clear all;close all;

%Insertamos los valores conocidos y ponemos 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)');

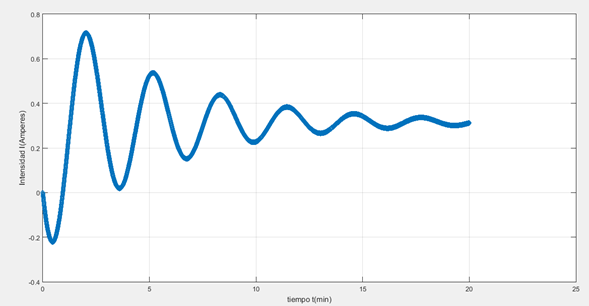
%Después imprimimos los valores cada dos segundos

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

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

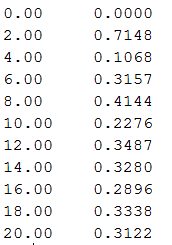
end

Y matlab nos arroja:



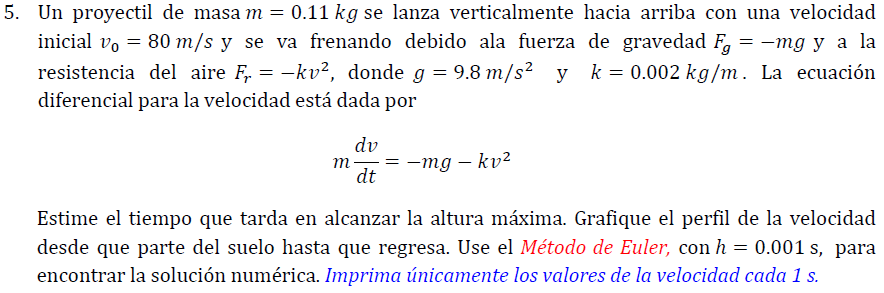
**SOLUCIÓN.**

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

****

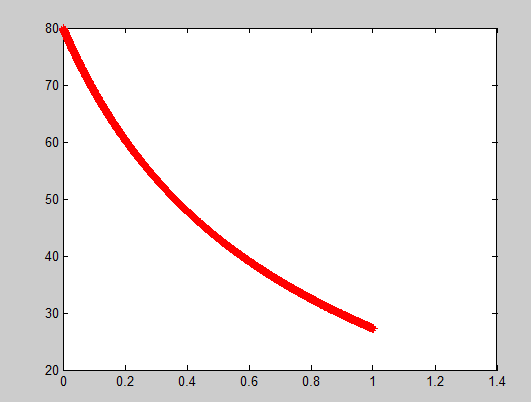
**I(20)= 0.3122A**

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

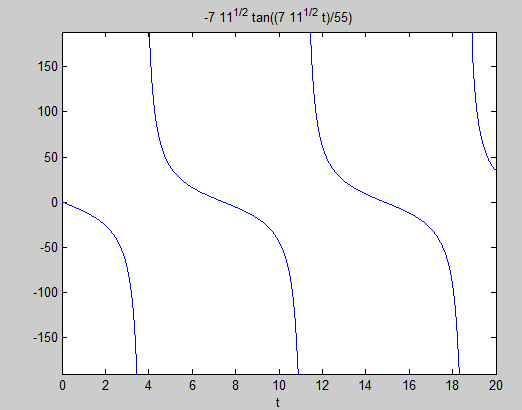


Teniendo a la ecuación, despejamos a dv/dt:

Una vez teniendo en cuenta esta parte se procede a obtener la gráfica, dando como resultado:



Solución exacta.



Código empleado:

1. clear all
2. clc
3. m=0.11;
4. g=9.8;
5. k=0.002;
6. f=@(t,V)-g-(k\*V^2)/m;%Variable dependediente y var independiente
7. %mi condicion inicial
8. t(1)=0;
9. V(1)=80;
10. %Tamaño de paso
11. h=0.001;
12. i=1; %el paso uno el valor inicial
13. while t(i)<=1
14. %poner el esquema de euler
15. t(i+1)=t(i)+h;
16. V(i+1)=V(i)+h\*f(t(i),V(i));
17. i=i+1;
18. end
19. %poner quien es mi soolución euler
20. sol\_Euler=[t' V']
21. figure(1)
22. plot(t,V,'r\*')
24. %solución exacta
25. clear V t
26. syms V(t)
27. %escrbir tal cual la ecuación diferencial
28. sol\_exacta=dsolve(diff(V)==-g-(k\*V^2)/m,V(0)==0)
29. figure(2)
30. ezplot(sol\_exacta,[0,20])