

# EJERCICIOS

## 1.Método de Euler

Realizar los siguientes pasos para hacer uso de código Matlab:

1. Insertar el código en New script
2. Dar click en la opción “Correr”
3. Se procede a digitar los valores en negrita depues de cada enter
4. En el comando command Window insertar la EDO:  **$2*x-3*y+1$**
5. Enter
6.  **$x_0=1$**
7. Enter
8.  **$y_0=5$**
9. Enter
10.  **$x_n=1.2$**
11. Enter
12.  **$h=0.05$**
13. Enter
14. value to aproximate= **$1.2$**

Automáticamente aparece la respuesta a dicha EDO así como su gráfica

A continuación se muestra como deben ir los datos

```
Enter function in form of dy/dx=f(x,y):2*x-3*y+1
Enter x0=1
Enter y0=5
Enter upper limit of interval xn=1.2
Enter width (equal space) h=0.05
Enter value to aproximate=1.2
```

```
-----
      xi          yi
-----
1.000000      5.000000
1.050000      4.400000
1.100000      3.895000
1.150000      3.470750
```

```
Solución real:
(2*x)/3 + (38*exp(-3*x)*exp(3))/9 + 1/9
```

```
La solución aproximada evaluada en x=1.200000 es: 3.115137
```

```
La solución real evaluada en x=1.200000 es: 3.228316
```

```
Error absoluto: 0.113178>>
```

```

%SOLUCION CON MATLAB
% Euler method using MATLAB

clear all;
close all;
clc;

syms x y
f(x,y)=input('Enter function in form of dy/dx=f(x,y):');

%Valores iniciales
i=1;
x0(i)=input('Enter x0=');
y0(i)=input('Enter y0=');

xn=input('Enter upper limit of interval xn='); %Alcance de la grafica de
aproximacion
h=input('Enter width (equal space) h='); %Incremento
n=(xn-x0)/h; %Numero de iteraciones

xapro=input('Enter value to approximate=');%Numero a aproximar

fprintf('-----\n')
fprintf('      xi          yi          \n');
fprintf('-----\n')

for i=1:(n+1)

    y1(i)=y0(i)+h*subs(subs(f,x,x0(i)),y,y0(i));
    fprintf('%f      %f      \n',x0(i),y0(i))
    y0(i+1)=y1(i);
    x0(i+1)=x0(i)+h;
end

plot(x0,y0,'*-');grid on;

%Solucion real
syms y(x)
dy = diff(y(x),x);
Y(x) = dsolve(dy== f(x,y), y(x0(1))==y0(1)); %Sol. real de la ED
fprintf('\n')
fprintf('Soluci?n real:\n')
disp(Y(x))
hold on;
fplot(Y(x),'+-')
title('Aproximacion vs Solucion anal?tica')
legend('aprox.','sol. real')

%Error absoluto
Er = Y(xapro) - y0(round((xapro-x0(1))/h+1));

```

```

%Mostrando resultados
fprintf('La soluci?n aproximada evaluada en x=%f es: %f',xapro ,y0(round((xapro-
x0(1))/h+1)))
%fprintf('La soluci?n aproximada evaluada en x=%f es: %f',xapro ,y0(i))
fprintf('\n\n')
fprintf('La soluci?n real evaluada en x=%f es: %f',xapro,Y(xapro))
fprintf('\n\n')
fprintf('Error absoluto: %f', Er)

```

## 1.Método de Euler Mejorado

Realizar los siguientes pasos para hacer uso de código Matlab:

1. Insertar el código en New script
2. Dar click en la opción “Correr”
3. Se procede a digitar los valores en negrita depues de cada enter
4. En el comando command Window insertar la EDO:  **$2 \cdot x \cdot y$**
5. Enter
6.  $x_0=1$
7. Enter
8.  $y_0=1$
9. Enter
10.  $x_n=1.5$
11. Enter
12.  $h=0.1$
13. Enter
14. value to aproximate=**1.5**

Automáticamente aparece la respuesta a dicha EDO así como su gráfica  
 continuación se muestra como deben ir los datos

```

Command Window
Enter function in form of dy/dx=f(x,y):2*x*y
Enter x0=1
Enter y0=1
Enter upper limit of interval xn=1.5
Enter width (equal space) h=0.1
Enter value to aproximate=1.5
-----
xi          yi
-----
1.000000    1.000000
1.100000    1.232000
1.200000    1.547885
1.300000    1.983150
1.400000    2.590787
1.500000    3.450929

Solución real:
exp(x^2)*exp(-1)

La solución aproximada evaluada en x=1.500000 es: 3.450929

La solución real evaluada en x=1.500000 es: 3.490343

fx Error absoluto: 0.039414>>

```

% Numerical Method  
 % Improved Euler method using MATLAB

%%Nota1: La n tiene que dar un numero entero. Mientras la h sea mas pequeña  
 %%es mas probable que la n sea entera.

%%Nota2: Si se va a resolver una ecuacin que no tiene solucion analitica el  
 %%program va a lanzar error, se tendra que borrar la seccion de la solucion  
 real y lo que  
 %%tenga asociado.

```

clear all;
close all;
clc;

```

```

syms x y
f(x,y)=input('Enter function in form of dy/dx=f(x,y):');

```

```

%Valores iniciales
i=1;
x0(i)=input('Enter x0=');
y0(i)=input('Enter y0=');

```

```

xn=input('Enter upper limit of interval xn='); %Alcance de la grafica de
aproximacion
h=input('Enter width (equal space) h='); %Incremento
n=(xn-x0)/h; %Numero de iteraciones

```

```

xapro=input('Enter value to aproximate=');%Numero a aproximar

```

```

fprintf('-----\n')
fprintf('      xi          yi          \n');
fprintf('-----\n')

```

```

for i=1:(n+1)

    y1(i)=y0(i)+h*subs(subs(f,x,x0(i)),y,y0(i));
    x0(i+1)=x0(i)+h;

```

```

        y2(i)=y0(i)+h*(subs(subs(f,x,x0(i)),y,y0(i)) +
subs(subs(f,x,x0(i+1)),y,y1(i)))/2;
        fprintf('%f      %f      \n',x0(i),y0(i))
        y0(i+1)=y2(i);

end

plot(x0,y0,'*-');grid on;

%Solucion real
syms y(x)
dy = diff(y(x),x);
Y(x) = dsolve(dy== f(x,y), y(x0(1))==y0(1)); %Sol. real de la ED
fprintf('\n')
fprintf('Solución real:\n')
disp(Y(x))
hold on;
fplot(Y(x),'+-',[-1 2])
title('Aproximacion vs Solucion analítica')
legend('aprox.','sol. real')

%Error absoluto
Er = Y(xapro) - y0(round((xapro-x0(1))/h+1));

%Mostrando resultados
fprintf('La solución aproximada evaluada en x=%f es: %f',xapro ,y0(round((xapro-
x0(1))/h+1)))
fprintf('\n\n')
fprintf('La solución real evaluada en x=%f es: %f',xapro,Y(xapro))
fprintf('\n\n')
fprintf('Error absoluto: %f', Er)

```