

Métodos Numéricos Aplicados a Ecuaciones Diferenciales

2.1.1. Método de Euler Hacia Adelante

Pártase del más simple tipo de ecuación diferencial ordinaria, que la de tipo lineal de primer orden, el clásico *Problema de Cauchy* de la forma:

$$\wp \equiv \begin{cases} y'(t) = \frac{dy(t)}{dt} = t + y \\ y(t_0 = 0) = 1 \end{cases}$$

Se desea resolver el problema \wp para determinar el valor de la función $y(t)$, para $t = 1$. Emplee un $h = 0.2$.

Resolución.

Se procede a determinar el número de subintervalos equi-espaciados $t \in [0,1]$:

$$n = \frac{t_n - t_0}{h} = \frac{1 - 0}{0.2} = 5 \quad n = 5$$

Se procede a aplicar el método de Euler.

$$\begin{cases} Y_{j+1} = Y_j + hF_j \\ Y_0 = y_0 \end{cases}$$

Para $j = 0$, se tiene: $Y_0 = 1, t_0 = 0$.

$$Y_1 = Y_0 + hF_0 \quad Y_1 = Y_0 + h(t_0 + Y_0)$$

$$Y_1 = 1 + 0.2(0 + 1) \quad Y_1 = 1.2$$

Para $j = 1$, se tiene: $Y_1 = 1.2, t_1 = t_0 + h, t_1 = 0.2$

$$Y_2 = Y_1 + hF_1 \quad Y_2 = Y_1 + h(t_1 + Y_1)$$

$$Y_2 = 1.2 + 0.2(0.2 + 1.2) \quad Y_2 = 1.48$$

Para $j = 2$, se tiene: $Y_2 = 1.48, t_2 = t_1 + h, t_2 = 0.4$

$$Y_3 = Y_2 + hF_2 \quad Y_3 = Y_2 + h(t_2 + Y_2)$$

$$Y_3 = 1.48 + 0.2(0.4 + 1.48) \quad Y_3 = 1.856$$

Para $j = 3$, se tiene: $Y_3 = 1.856$, $t_3 = t_2 + h$, $t_3 = 0.6$

$$Y_4 = Y_3 + hF_3 \quad Y_4 = Y_3 + h(t_3 + Y_3) \quad Y_4 = 2.3472$$

Para $j = 4$, se tiene: $Y_4 = 2.3472$, $t_4 = t_3 + h$, $t_4 = 0.8$

$$Y_5 = Y_4 + h(t_4 + Y_4) \quad Y_5 = 2.97664$$

Tabla 1. Resultados paso a paso, tabulados por el Método de Euler

j	t_j	$Y(t_j)$
1	0.2	1.2
2	0.4	1.48
3	0.6	1.856
4	0.8	2.3472
5	1.0	2.97664

A fin de mejorar la potencialidad de cálculo, y automatizar el proceso, se procedió a implementar un archivo script en MATLAB™, cuyo nombre es **EulerSolve.m**.

```

Editor - C:\EulerSolve.m*
File Edit Text Cell Tools Debug Desktop Window Help
[Icons] Stack: Base
1 % Metodo de Euler para Resolver una Ecuacion Diferencial Ordinaria
2 % Resuelve por el Metodo de Euler Simple Hacia Adelante
3 % y' = f(t,y), y(a) = y0, tinit < t < tfinal
4 % hay N+1 numero de subintervalos igualmente espaciados [tinit,tfinal]
5 %
6 % Autor: Francisco M. Gonzalez-Longatt
7 % Fecha: March 9, 2006
8 %
9 % Precaucion: Solamente para usos de ensenanza.
10 %
11 tinit =0; % Comienzo del intervalo
12 tfinal =1; % Final del intervalo
13 Y0 =1; % Condicion inicial
14 h=0.2; % Paso de intefracion
15 n=(tfinal-tinit)/h; % Numero de Puntos Necesarios
16 Y=zeros(n,1); % Inicializar el vector de Salida
17 t=zeros(n,1); % Inializar el vector de tiempo
18 Y(1)=Y0;
19 t(1)=tinit;
20 disp('          j          tj          Yj')
21 disp('-----')
22 for i=1:n
23     Y(i+1)=Y(i)+h*f(t(i),Y(i));
24     t(i+1)=tinit+i*h;
25     disp(sprintf(' %i          %g          %g', i, t(i+1), Y(i+1)))
26 end % for

```

A continuación se presenta el contenido de este programa.

```
% Método de Euler para Resolver una Ecuación Diferencial Ordinaria
% Resuelve por el Método de Euler Simple Hacia Adelante
% y' = f(t,y), y(a) = y0, tinit < t < tfinal
% hay N+1 numero de sub-intervalos igualmente espaciados [tinit,tfinal]
%
% Autor: Francisco M. González-Longatt
% Fecha: March 9, 2006
%
% Precaución: Solamente para usos de enseñanza.
%
tinit =0;           % Comienzo del intervalo
tfinal =1;         % Final del intervalo
Y0 =1;             % Condición inicial
h=0.2;            % Paso de integración
n=(tfinal-tinit)/h; % Número de Puntos Necesarios
Y=zeros(n,1);     % Inicializar el vector de Salida
t=zeros(n,1);     % Inicializar el vector de tiempo
Y(1)=Y0;
t(1)=tinit;
for i=1:n
    Y(i+1)= Y(i) + h*F(t(i),Y(i));
    t(i+1)= tinit + i*h;
end % for
```

Al ejecutar éste programa, en el workspace de MATLAB™, se obtiene la solución de la ecuación diferencial¹:

k	tk	Yk
1	0.2	1.2
2	0.4	1.48
3	0.6	1.856
4	0.8	2.3472
5	1.0	2.97664

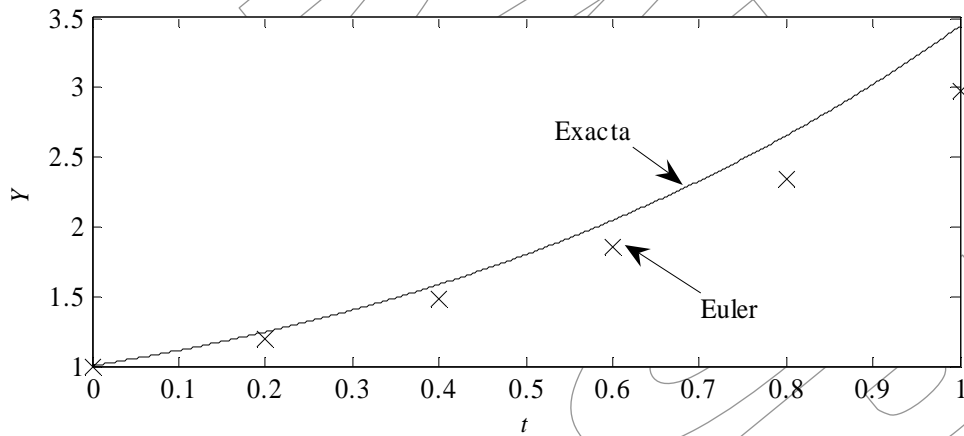


Figura 1. Grafico de la solución punto a punto de la ecuación diferencial empleando el método de Euler hacia adelante.

¹ Es importante, aclarar que el archivo presentado, no muestra resultados en el workspace, para que se publiquen en el modo mostrado antes, se deben agregar unas líneas 20, 21 y 25:

```
disp('      j      tj      Yj')
disp('-----')
disp(sprintf('%i      %g      %g',i,t(i+1),Y(i+1)))
```

2.1.2. Método de Euler Modificado

El *método de Euler modificado* tiene dos motivaciones. La primera es que es más preciso que el método de Euler por diferencias hacia delante o hacia atrás. La segunda es que este método es más estable que su homólogo hacia delante. El método de Euler modificado también es reconocido como el método de punto medio, y las ecuaciones aproximante:

$$Y_{i+\frac{1}{2}} = Y_i + \frac{h}{2} F(t_i, Y_i)$$

$$Y_{i+1} = Y_i + hF\left(t_{i+\frac{1}{2}}, Y_{i+\frac{1}{2}}\right)$$

El error local atribuible a este método es del orden de $O(h^3)$ mientras que el error global es $O(h^2)$

Se va a proceder a resolver el mismo ejemplo, empleando el método de Euler modificado:

Para $j = 0$, se tiene: $Y_0 = 1, t_0 = 0$.

$$Y_{0+\frac{1}{2}} = Y_0 + hF_0 \quad Y_{0+\frac{1}{2}} = Y_0 + h(t_0 + Y_0) \quad Y_{0+\frac{1}{2}} = 1 + 0.2(0+1) \quad Y_{0+\frac{1}{2}} = 1.2$$

$$Y_1 = Y_0 + hF\left(t_{0+\frac{1}{2}}, Y_{0+\frac{1}{2}}\right) \quad Y_1 = Y_0 + h\left(t_{0+\frac{1}{2}} + Y_{0+\frac{1}{2}}\right) \quad Y_1 = 1.24$$

Se sigue calculando para los restantes subintervalos y se obtiene:

Tabla 2. Resultados paso a paso, tabulados por el Método de Euler Modificado

j	t_j	$Y(t_j)$
1	0.2	1.24
2	0.4	1.5768
3	0.6	2.0317
4	0.8	2.63067
5	1.0	3.40542

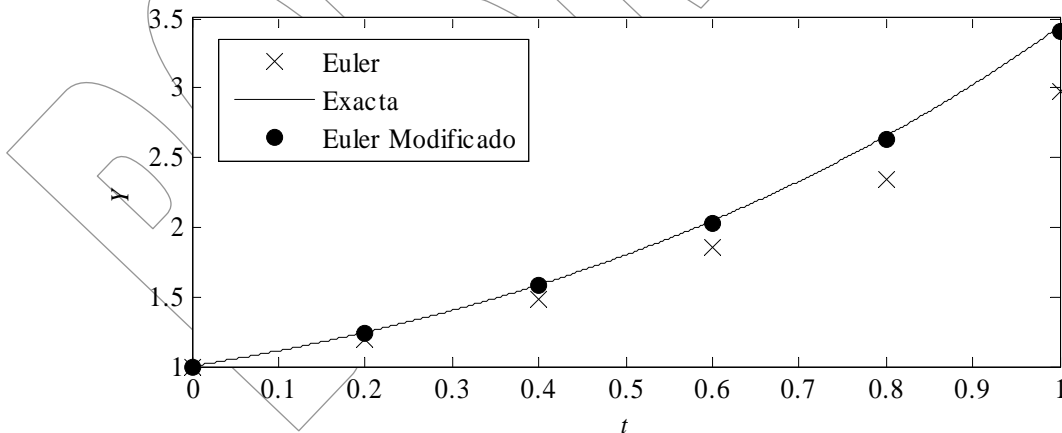


Figura 2. Gráfico de la solución punto a punto de la ecuación diferencial empleando el método de Euler Modificado (Medio Punto).

Se procedió a implementar un archivo script en MATLAB™, cuyo nombre es EulerModSolve.m.

⌘ Método de Euler Modificado para Resolver una Ecuación Diferencial Ordinaria

```

% Autor: Francisco M. Gonzalez-Longatt
% Fecha: March 9, 2006
% Precaución: Solamente para usos de enseñanza.
tinit =0; % Comienzo del intervalo
tfinal =1; % Final del intervalo
Y0 =1; % Condición inicial
h=0.2; % Paso de integrasion
n=(tfinal-tinit)/h; % Numero de Puntos Necesarios
Y=zeros(n,1); % Inicializar el vector de Salida
t=zeros(n,1); % Inializar el vector de tiempo
Y(1)=Y0;
t(1)=tinit;
for i=1:n
    t(i+1)=tinit+i*h;
    Y0(i+1)=Y(i)+h*F(Y(i),t(i))/2;
    Y(i+1)=Y(i)+h*F(Y0(i+1),t(i)+h/2);
end % for
    
```

Los resultados obtenidos con **EulerModSolve.m** son:

j	tj	Yj
1	0.2	1.24
2	0.4	1.5768
3	0.6	2.0317
4	0.8	2.63067
5	1	3.40542

2.2 Método de Heun (Predictor Corrector)

El método de Euler posee un serio defecto en su aproximación para determinar la pendiente usando en cada paso de la iteración. Una de las mejoras que se aplica el método de Euler, es el denominado Predictor-Corrector, cuya ecuaciones interactivas resultan ser:

$$Y_{i+1}^0 = Y_i + hF(t_i, Y_i)$$

$$Y_{i+1} = Y_i + \frac{h}{2} [F(t_i, Y_i) + F(t_{i+1}, Y_{i+1}^0)]$$

El error local es de $O(h^3)$ y el error global es de orden $O(h^3)$. Al sustituir valores se tiene:

Tabla 3. Resultados paso a paso, tabulados por el Método Predictor Corrector

j	t _i	Y(t _i)
1	0.2	1.24
2	0.4	1.5768
3	0.6	2.0317
4	0.8	2.63067
5	1.0	3.40542

Se verifica que a primera vista los resultados son idénticos a los obtenidos por del método de Euler Modificado.

Solo para ser empleado con objetivo de evaluación, o académicos. Prohibido la reproducción total o parcial de este documento.

```

% Método de Heun para Resolver una Ecuación Diferencial Ordinaria
% Autor: Francisco M. González-Longatt
% Fecha: March 9, 2006
% Precaución: Solamente para usos de enseñanza.
tinit =0; % Comienzo del intervalo
tfinal =1; % Final del intervalo
Y0 =1; % Condición inicial
h=0.2; % Paso de integración
n=(tfinal-tinit)/h; % Numero de Puntos Necesarios
Y=zeros(n,1); % Inicializar el vector de Salida
t=zeros(n,1); % Inicializar el vector de tiempo
Y(1)=Y0;
t(1)=tinit;
for i=1:n
    t(i+1)=tinit+i*h;
    Y0(i+1)=Y(i)+h*F(Y(i),t(i));
    Y(i+1)=Y(i)+h/2*(F(Y0(i+1),t(i+1))+F(Y(i),t(i)));
end % for

```

Los resultados obtenidos por el programa codificado **HeunSolve.m**, es:

j	tj	Yj
1	0.2	1.24
2	0.4	1.5768
3	0.6	2.0317
4	0.8	2.63067
5	1	3.40542

2.3 Métodos de Runge-Kutta

2.3.1.1. Segundo Orden (Método de Ralston)

Para el caso del Runge-Kutta de segundo orden, es también conocido como el Método de Ralston, se tiene que las ecuaciones iterativas son:

$$k_1 = F(t_i, Y_i)$$

$$k_2 = F\left(t_i + \frac{3}{4}h, Y_i + \frac{3}{4}k_1h\right)$$

$$Y_{i+1} = Y_i + h\left(\frac{1}{3}k_1 + \frac{2}{3}k_2\right)$$

Al efectuar las respectivas evaluaciones se tiene:

Tabla 4. Resultados paso a paso, tabulados por el Método Runge-Kutta de Segundo Orden

j	tj	Y(tj)
1	0.2	1.24
2	0.4	1.5768
3	0.6	2.0317
4	0.8	2.63067
5	1.0	3.40542

Se codificó las ecuaciones aproximantes en el archivo script de MATLAB™, llamado **RalstonSolve.m**, arrojando los siguientes resultados:

j	tj	K1	K2	Yj
1	0.2	1.24	1	1.3
2	0.4	1.5768	1.44	1.806
3	0.6	2.0317	1.9768	2.42332
4	0.8	2.63067	2.6317	3.17645
5	1	3.40542	3.43067	4.09527

Tiempo Total Empleado: 0.160000 seconds.

```
% Método de Ralston para Resolver una Ecuación Diferencial Ordinaria
% Autor: Francisco M. Gonzalez-Longatt
% Fecha: March 9, 2006
% Precaución: Solamente para usos de enseñanza.
tinit =0; % Comienzo del intervalo
tfinal =1; % Final del intervalo
Y0 =1; % Condición inicial
h=0.2; % Paso de integracion
n=(tfinal-tinit)/h; % Numero de Puntos Necesarios
Y=zeros(n,1); % Inicializar el vector de Salida
t=zeros(n,1); % Inicializar el vector de tiempo
Y(1)=Y0;
t(1)=tinit;
for i=1:n
    t(i+1)=tinit+i*h;
    K1=F(Y(i),t(i));
    K2=F(Y(i)+3/4*K1*h,t(i)+3/4*h);
    Y(i+1)=Y(i)+h*(1/3*K1+2/3*K2);
end % for
```

2.3.1.2. Tercer Orden

Para el caso del Runge-Kutta de tercer orden, se cumple que tiene un error local de $O(h^4)$ y un error global de $O(h^3)$, y sus ecuaciones aproximantes son:

$$k_1 = F(t_i, Y_i)$$

$$k_2 = F\left(t_i + \frac{1}{2}h, Y_i + \frac{1}{2}k_1h\right)$$

$$k_3 = F(t_i + h, Y_i - k_1h + 2k_2h)$$

$$Y_{i+1} = Y_i + \frac{h}{6}(k_1 + 4k_2 + k_3)$$

Evaluando para cada sub-intervalo se tiene:

Tabla 5. Resultados paso a paso, tabulados por el Método Runge-Kutta de Tercer Orden

j	t _i	Y(t _i)
1	0.2	1.24267
2	0.4	1.58331
3	0.6	2.04362
4	0.8	2.65007
5	1.0	3.43502

Se codificó las ecuaciones aproximantes en el archivo script de MATLAB™, llamado **RK3Solve.m**, arrojando los siguientes resultados:

j	t _j	K1	K2	K3	Y _j
1	0.2	1.24267	1	1.2	1.48
2	0.4	1.58331	1.44267	1.68693	2.02891
3	0.6	2.04362	1.98331	2.28164	2.6993
4	0.8	2.65007	2.64362	3.00798	3.51808
5	1	3.43502	3.45007	3.89508	4.51809

Tiempo Total Empleado: 0.210000 seconds.

```
% Método de Runge-Kutta de Tercer Orden Modificado para Resolver una
```

```

% Ecuación Diferencial Ordinaria
% Autor: Francisco M. Gonzalez-Longatt
% Fecha: March 9, 2006
% Precaución: Solamente para usos de enseñanza.
tinit =0; % Comienzo del intervalo
tfinal =1; % Final del intervalo
Y0 =1; % Condicion inicial
h=0.2; % Paso de integracion
n=(tfinal-tinit)/h; % Numero de Puntos Necesarios
Y=zeros(n,1); % Inicializar el vector de Salida
t=zeros(n,1); % Inicializar el vector de tiempo
Y(1)=Y0;
t(1)=tinit;
for i=1:n
    t(i+1)=tinit+i*h;
    K1=F(Y(i),t(i));
    K2=F(Y(i)+1/2*K1*h,t(i)+1/2*h);
    K3=F(Y(i)-K1*h+2*K2*h,t(i)+h);
    Y(i+1)=Y(i)+h*(K1+4*K2+K3)/6;
end % for

```

2.3.1.3. Cuarto Orden

Para el caso del Runge-Kutta de cuarto orden, se posee las siguientes ecuaciones aproximantes:

$$\begin{aligned}
 K_1 &= hF(t_j, Y_j) \\
 K_2 &= hF\left(t_j + \frac{h}{3}, Y_j + \frac{1}{3}K_1\right) \\
 K_3 &= hF\left(t_j + \frac{2h}{3}, Y_j + \frac{1}{3}K_1 + \frac{1}{3}K_2\right) \\
 K_4 &= hF(t_j + h, Y_j + K_1 + K_2 + K_3)
 \end{aligned}
 \tag{40}$$

$$Y_{j+1} = Y_j + \frac{1}{8}(K_1 + 3K_2 + 3K_3 + 4K_4)
 \tag{41}$$

Tabla 6. Resultados paso a paso, tabulados por el Método Runge-Kutta de Cuarto Orden

j	t_j	$Y(t_j)$
1	0.2	1.2516
2	0.4	1.60683
3	0.6	2.08946
4	0.8	2.72874
5	1.0	3.5606

Se codificó las ecuaciones aproximantes en el archivo sript de MATLAB™, llamado **RK4Solve.m**.

j	t_j	K1	K2	K3	K4	Y_j
1	0.2	1.2428	1	1.2	1.22	1.444
2	0.4	1.58364	1.4428	1.68708	1.71151	1.9851
3	0.6	2.04421	1.98364	2.282	2.31184	2.646
4	0.8	2.65104	2.64421	3.00863	3.04508	3.45323
5	1	3.4365	3.45104	3.89615	3.94066	4.43917

Tiempo Total Empleado: 0.370000 seconds.

```

% Método de Runge Kutta de Cuarto Orden para Resolver una Ecuación Diferencial Ordinaria
% Autor: Francisco M. González-Longatt
% Fecha: March 9, 2006
% Precaución: Solamente para usos de enseñanza.
tinit =0; % Comienzo del intervalo
tfinal =1; % Final del intervalo
Y0 =1; % Condición inicial
h=0.2; % Paso de integracion
n=(tfinal-tinit)/h; % Numero de Puntos Necesarios
Y=zeros(n,1); % Inicializar el vector de Salida
t=zeros(n,1); % Inicializar el vector de tiempo
Y(1)=Y0;
t(1)=tinit;
for i=1:n
    t(i+1)=tinit+i*h;
    K1=F(Y(i),t(i));
    K2=F(Y(i)+1/2*K1*h,t(i)+1/2*h);
    K3=F(Y(i)+K2*h,t(i)+1/2*h);
    K4=F(Y(i)+K3*h,t(i)+h);
    Y(i+1)=Y(i)+h*(K1+2*K2+2*K3+K4)/6;
end % for
    
```

2.4 Comparación de Resultados

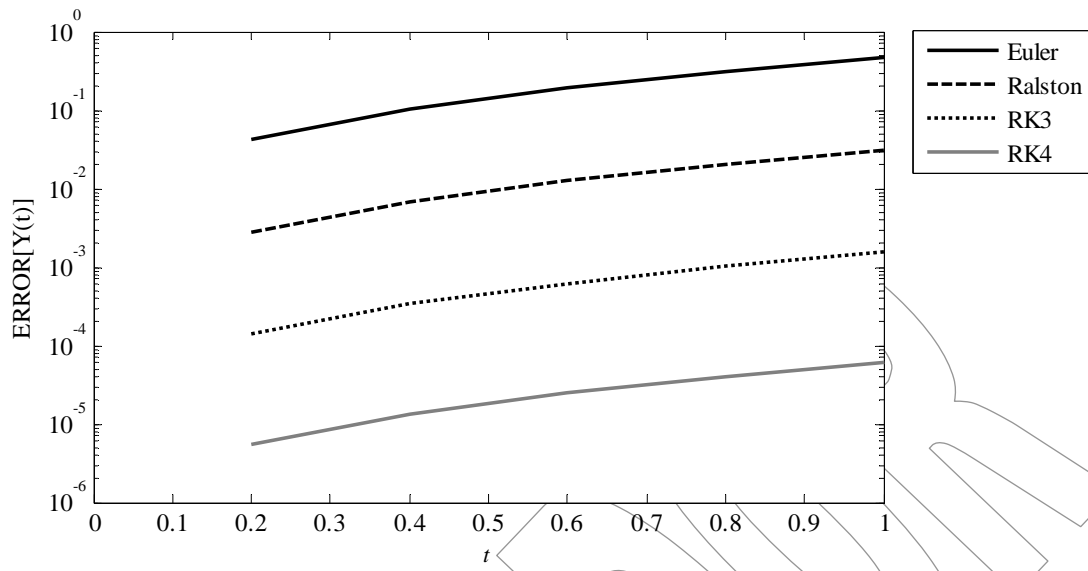
A continuación se muestra una tabla resumen con los resultados obtenidos de los diferentes métodos.

<i>t</i>	<i>Euler</i>	<i>Euler Modificado</i>	<i>Heun</i>	<i>Solución Exacta</i>
0	1.00000000000000	1.00000000000000	1.00000000000000	1.00000000000000
0.2	1.20000000000000	1.24000000000000	1.24000000000000	1.24280551632034
0.4	1.48000000000000	1.57680000000000	1.57680000000000	1.58364939528254
0.6	1.85600000000000	2.03169600000000	2.03169600000000	2.04423760078102
0.8	2.34720000000000	2.63066912000000	2.63066912000000	2.65108185698494
1.0	2.97664000000000	3.40541632640000	3.40541632640000	3.43656365691809

<i>t</i>	<i>Runge-Kutta 2do</i>	<i>Runge-Kutta 3^{ro}</i>	<i>RungeKutta 4^{to}</i>	<i>Solución Exacta</i>
0	1.00000000000000	1.00000000000000	1.00000000000000	1.00000000000000
0.2	1.24000000000000	1.24266666666667	1.24280000000000	1.24280551632034
0.4	1.57680000000000	1.58331022222222	1.58363592000000	1.58364939528254
0.6	2.03169600000000	2.04361621807407	2.04421291268800	2.04423760078102
0.8	2.63066912000000	2.65006994100780	2.65104165155712	2.65108185698494
1.0	3.40541632640000	3.43501875461753	3.43650227321187	3.43656365691809

En la siguiente grafica se muestra una comparación del error cometido por cada uno de los métodos.

Solo para ser empleado con objetivo de evaluación, o académicos. Prohibido la reproducción total o parcial de este documento.



Solo para ser empleado con objetivo de evaluación, o académicos. Prohibido la reproducción total o parcial de este documento.