OBLIGATORIO 3

MÉTODOS NUMÉRICOS

Curso 2017.

DMEL, CenUR LN

Universidad de la Republica

UdelaR.

Profesor: José Vieitez

Sebastián Castro

Grupo:

- Juan Ferrand
- Luciano Acosta

Contenido

Descripción del Problema

Se consideran las ecuaciones diferenciales de primer orden siguientes:

$$y' = f(x,y) = 2x - y, \qquad y(0) = -1, \qquad (1)$$
$$y' = g(x,y) = 2x * y, \qquad y(0) = 1, \qquad (2)$$

El problema planteado busca analizar y evaluar diferentes métodos numéricos comparándolos con las soluciones analíticas estimando los errores.

Ejercicio N° 1:

Hallar las soluciones analíticas de ambas ecuaciones (1) y (2).

Ecuación (1)

$$y' = 2x - y \,, y(0) = -1$$

$$\rightarrow y' + y = 2x$$

Factor Integrante:

$$FI = e^{\int p(x)dx} \quad p(x) = 1 \,, \; e^{\int 1} = e^x = FI$$

$$\rightarrow \; e^x y' + e^x y = 2xe^x$$

Del Factor Integrante obtenemos que

$$e^x y' + e^x y = (e^x y)'$$

$$\rightarrow \; \int (e^x y)' dx = \int 2xe^x \; dx$$

$$\int (e^x y)' dx = e^x y$$

$$2\int_0^x xe^x \, dx = 2(xe^x - \int e^x) = 2(xe^x - e^x) + c = 2xe^x - 2e^x + c$$

$$e^x y = 2xe^x - 2e^x$$

$$y(x) = 2x - 2 + c/e^x$$

Sabemos que $y(0) = -1$

→ -2 + c = -1, c=1

$$\boldsymbol{y(x) = 2x + e^{-x} - 2}$$

Ecuación (2)

$$y' = g(x,y) = 2x * y, \qquad y(0) = 1$$

$$y' = 2xy \rightarrow \frac{y'}{y} = 2x \rightarrow \int_{x0}^{x} \frac{y(s)'}{y(s)}\, ds = \int_{x0}^{x} 2s\, ds$$

$$\int_{x0}^{x} \frac{y(s)'}{y(s)}\, ds = \int_{x0}^{x} \frac{du}{u} = Ln(u) \begin{cases} x \\ x0 \end{cases} = Ln(y(s)) \begin{cases} x \\ x0 \end{cases}$$

$$\rightarrow \mathrm{Ln}(\mathrm{y(x)}) - Ln(y(x0)) = Ln(\frac{y(x)}{y(x0)})$$

$$\int_{x0}^{x} 2s\, ds = s^2 \begin{cases} x \\ x0 \end{cases} = x^2$$

$$\rightarrow Ln\left(\frac{y(x)}{y(x0)}\right) = x^2$$

$$\frac{y(x)}{y(x0)} = e^{x^2}$$

$$y(0) = 1, \rightarrow \boldsymbol{y(x) = e^{x^2}}$$

Ejercicio N° 2:

Aproximar las soluciones de ambas ecuaciones con el método de Forward Euler y de Backward Euler en el intervalo [0; 3] con un paso igual a h = 0,2 (i.e., dividiendo [0; 3] en 15 subintervalos iguales de longitud h = 3=15 = 0,2).

Método de Forward Euler:

$$\int_{a=x0}^{x1} f(x_0, y_0)\, dt = h\, f(x_0, y_0)$$

Donde h es el "paso" de la aproximación numérica de la E.D.O

$$\rightarrow y_1 = y_0 + h\, f(x_0, y_0)$$

$$y_2 = y_1 + h\, f(x_1, y_1)$$

.

.

.

$$y_{j+1} = y_j + h\, f(x_j, y_j)$$

Método de Backward Euler:

$$y_{j+1} = y_j + h\, f(x_{j+1}, y_{j+1})$$

$$h\, f(x_{j+1}, y_{j+1}) - y_{j+1} + y_j = 0$$

debemos hallar y_{j+1} con algún método iterativo (como por ejemplo Raphson-Newton)

Para ambos métodos se implementaron módulos que realizan los cálculos correspondientes brindándoles como parámetro de entrada “h”, obteniendo los siguientes resultados:

Soluciones con h = 0.2			
Forward Euler		Backward Euler	
Ecuación 1	Ecuación 2	Ecuación 1	Ecuación 2
-1	1	-1	1
-0.8	1	-0.766667	1
-0.56	1.08	-0.505556	1.08
-0.288	1.2528	-0.221296	1.2528
0.0096	1.5535	0.082253	1.5535
0.32768	2.0506	0.401878	2.0506
0.662144	2.8708	0.734898	2.8708
1.0097152	4.2488	1.079082	4.2488
1.3677722	6.6281	1.432568	6.6281
1.7342177	10.8702	1.793807	10.8702
2.1073742	18.6967	2.161506	18.6967
2.4858993	33.654	2.534588	33.654
2.8687195	63.2695	2.912157	63.2695
3.2549756	124.0082	3.293464	124.0082
3.6439805	252.9767	3.677887	252.9767
4.0351844	536.3107	4.064905	536.3107

Tabla N°1: Resultados obtenidos de los Métodos mencionados anteriormente.

A continuación, se presentan gráficos que comparan los resultados obtenidos con los métodos de Backward Euler y Forward Euler (para las Ecuaciones 1 y 2) con las soluciones analíticas de las respectivas ecuaciones.

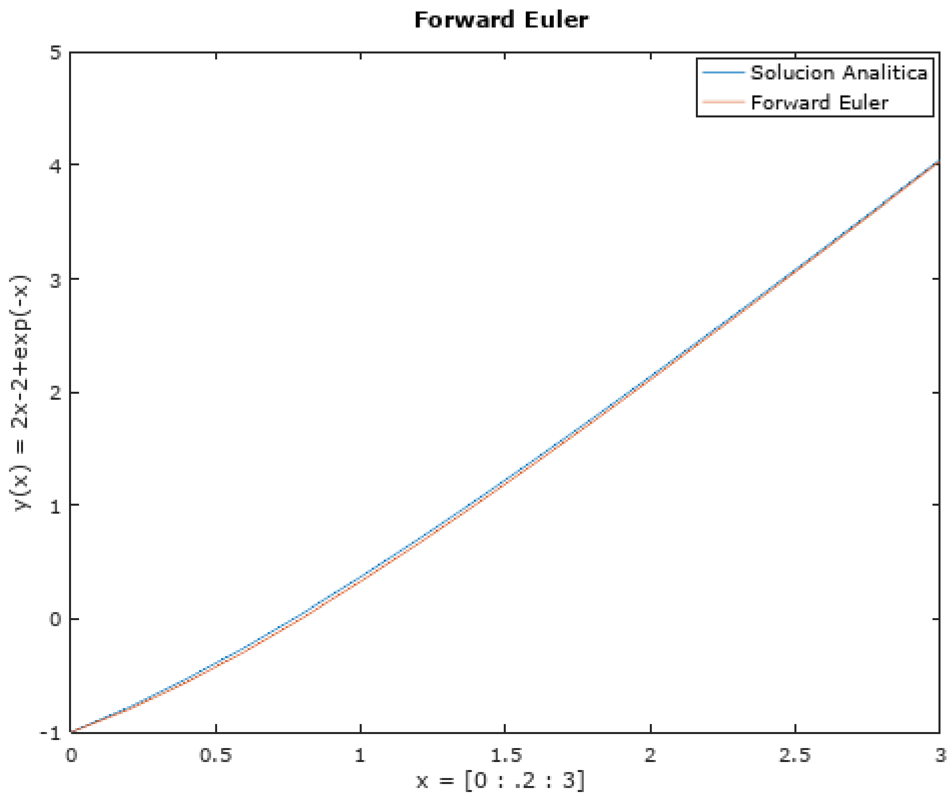


Figura N°1: Método de Forward Euler Ecuación 1.

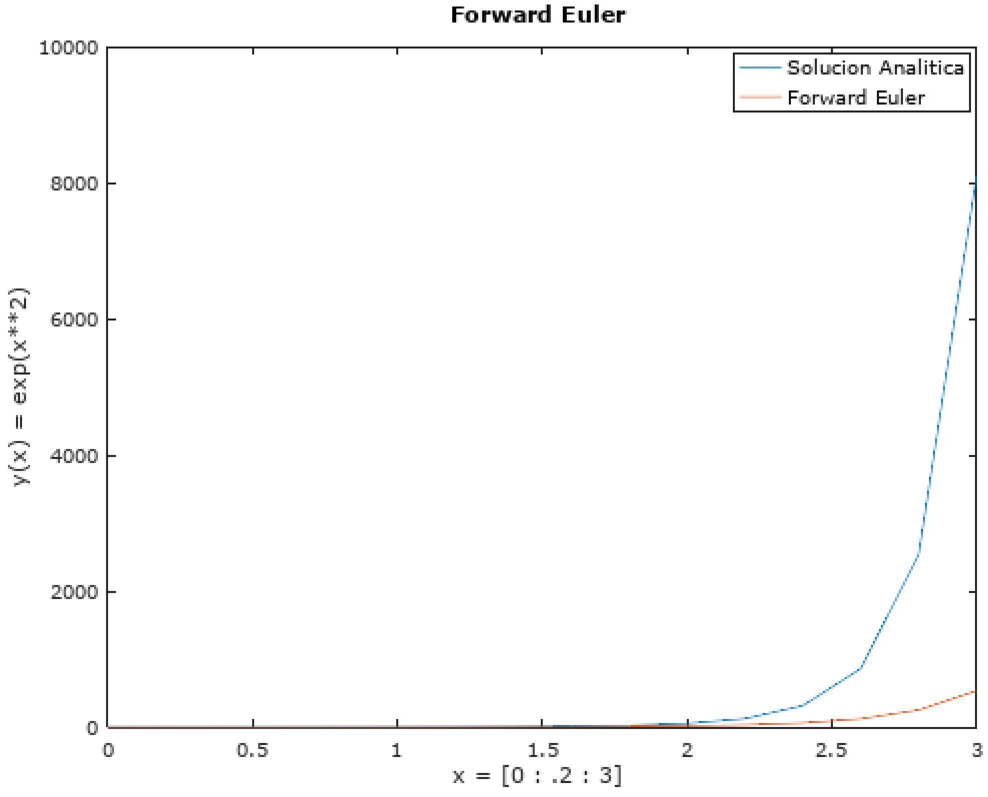


Figura N°1: Método de Forward Euler Ecuación 2.

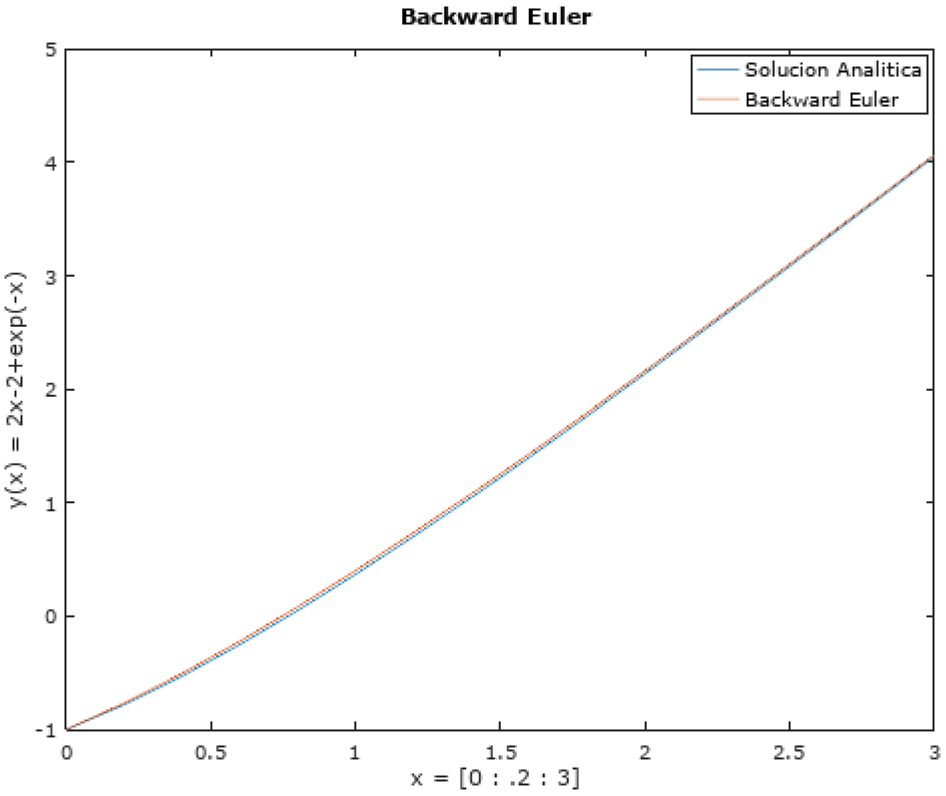


Figura N°3: Método de Backward Euler Ecuación 1.

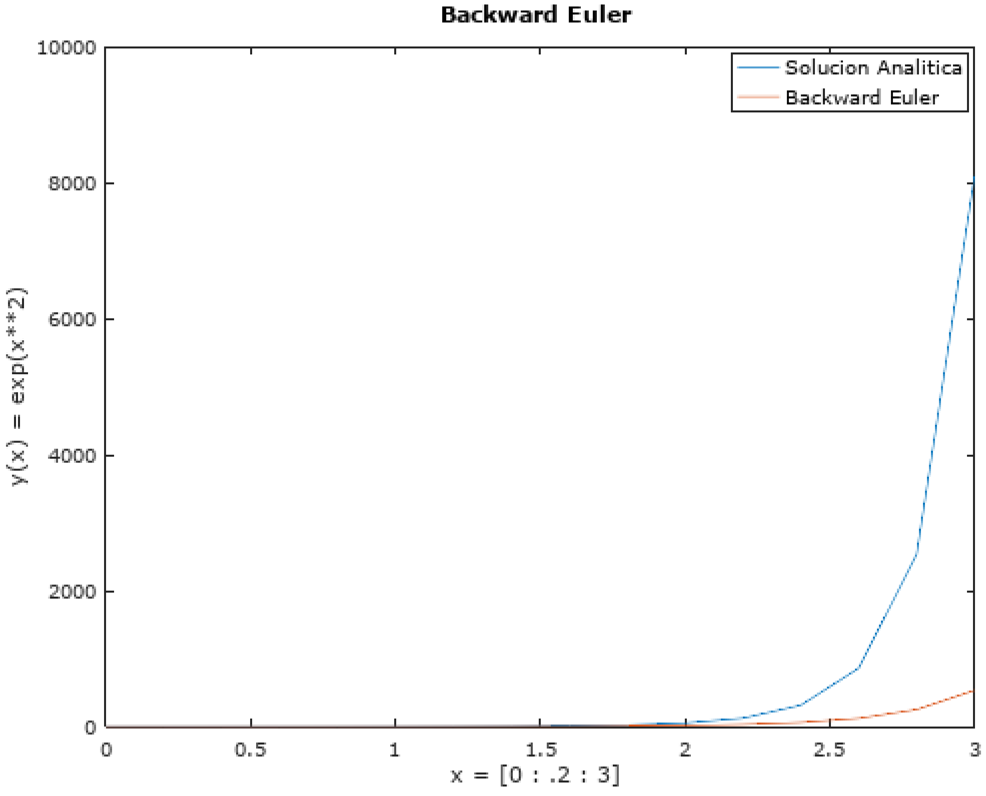


Figura N°4: Método de Backward Euler Ecuación 2.

Ejercicio N° 3:

Aproximar las soluciones de ambas ecuaciones por el método Predictor - Corrector y el método de Heun.

Método Predictor – Corrector:

$$y_{j+1}^{P} = y_j + h\, f(x_j, y_j)$$

$$y_{j+1}^{C} = y_{j+1} + h\, f(x_{j+1}, y_{j+1}^{P})$$

Método de Heun:

$$y_{j+1} = y_j + h\, f(\gamma_1 + \gamma_2)(x_j, y_j) + h^2\left[f_x(x_j, y_j)\gamma_2\alpha + \gamma_2\beta k_1 + f_x(x_j, y_j)\gamma_2\beta k_1\right]$$

Donde:

$$\gamma_1 = 1/4\ ;\ \gamma_2 = 3/4\ ;\ \alpha = \beta = 2/3$$

$$k_1 = f(x_j, y_j)$$

Para ambos métodos se implementaron módulos que realizan los cálculos correspondientes brindándoles como parámetro de entrada “h”, ver los módulos en “anexos”.

Ejercicio N°4:

Considerar ahora un paso igual a h = 0:1 (i.e., dividiendo en 30 subintervalos iguales a [0; 3]). Repetir lo anterior con este nuevo paso para ambas ecuaciones diferenciales.

A continuación, se presentan gráficos que comparan los resultados obtenidos con los métodos de Backward Euler, Forward Euler, Predictor Corrector y Heun (para las Ecuaciones 1 y 2) con las soluciones analíticas de las respectivas ecuaciones para un paso h = 0.1.

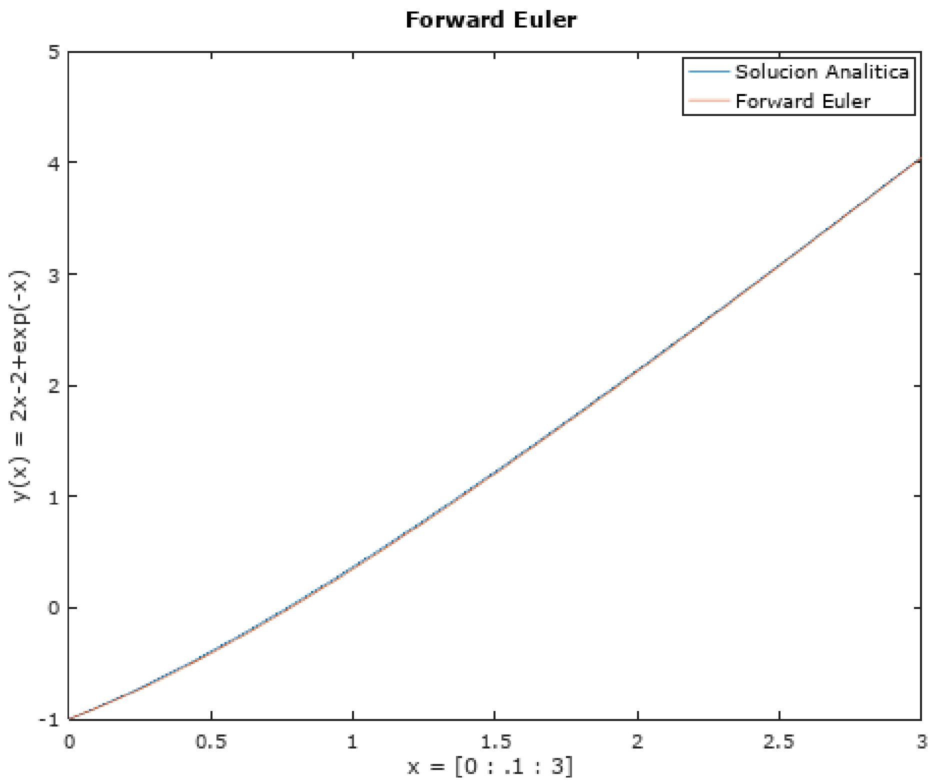


Figura N°5: Método de Forward Euler Ecuación 1.

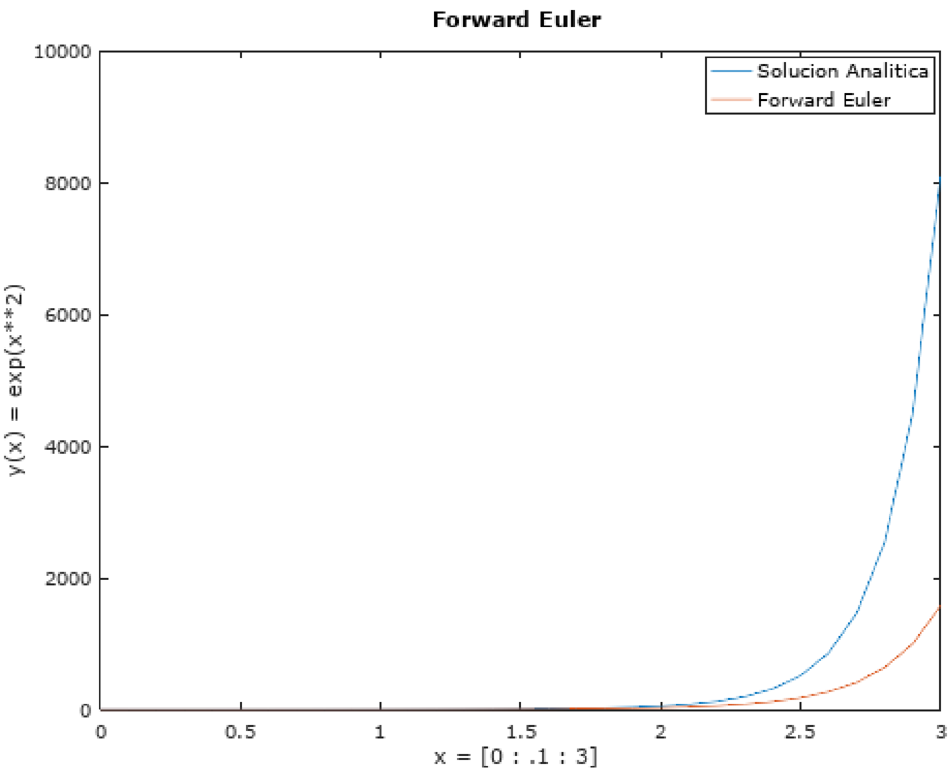


Figura N°6: Método de Forward Euler Ecuación 2.

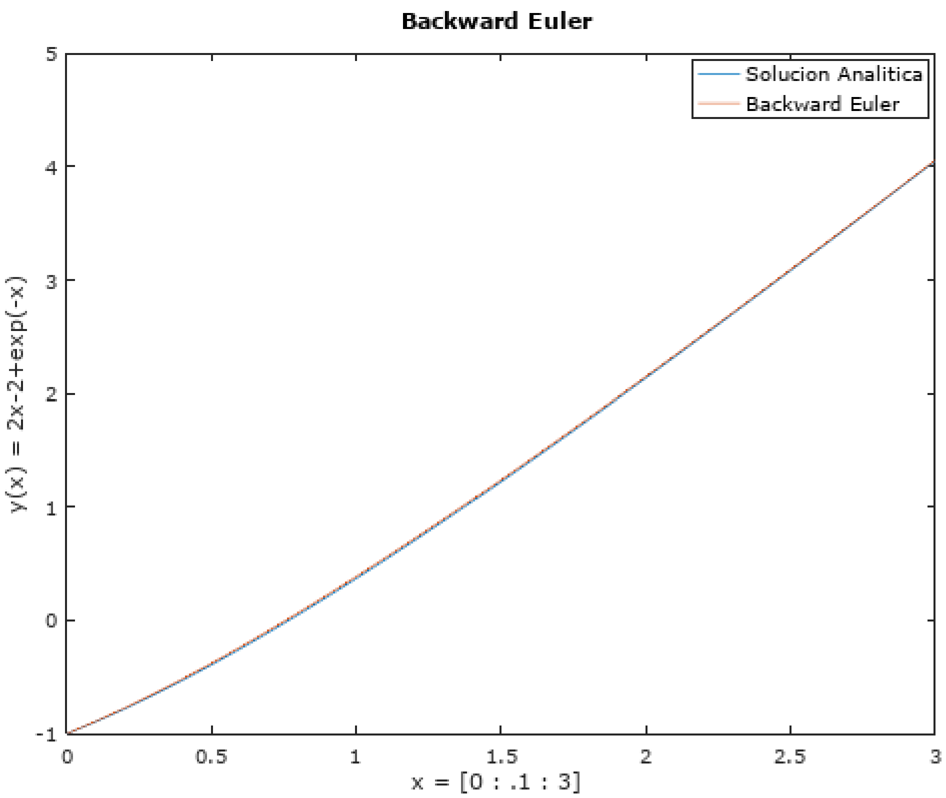


Figura N°7: Método de Backward Euler Ecuación 1.

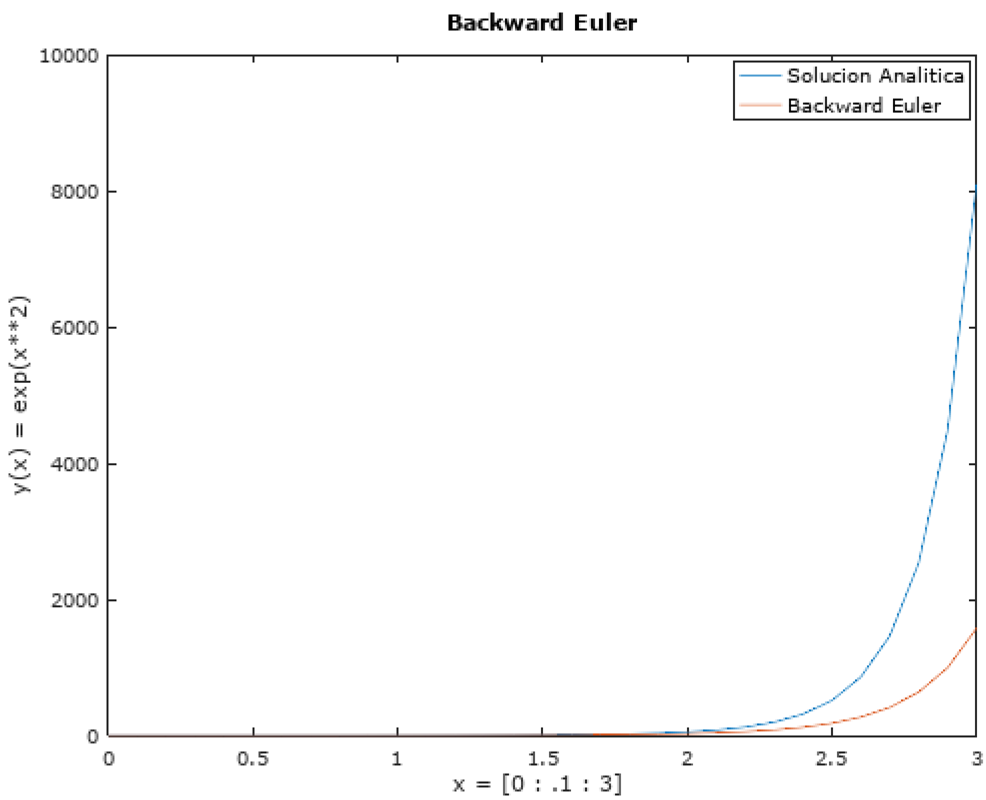


Figura N°8: Método de Backward Euler Ecuación 2.

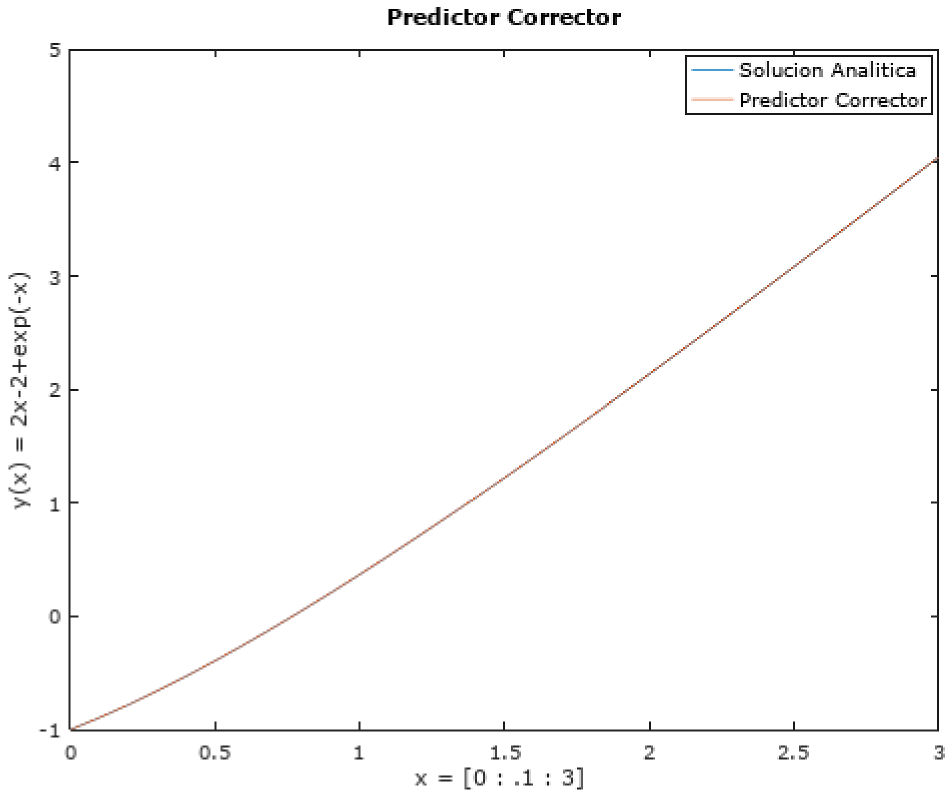


Figura N°9: Método de Predictor Corrector Ecuación 1.

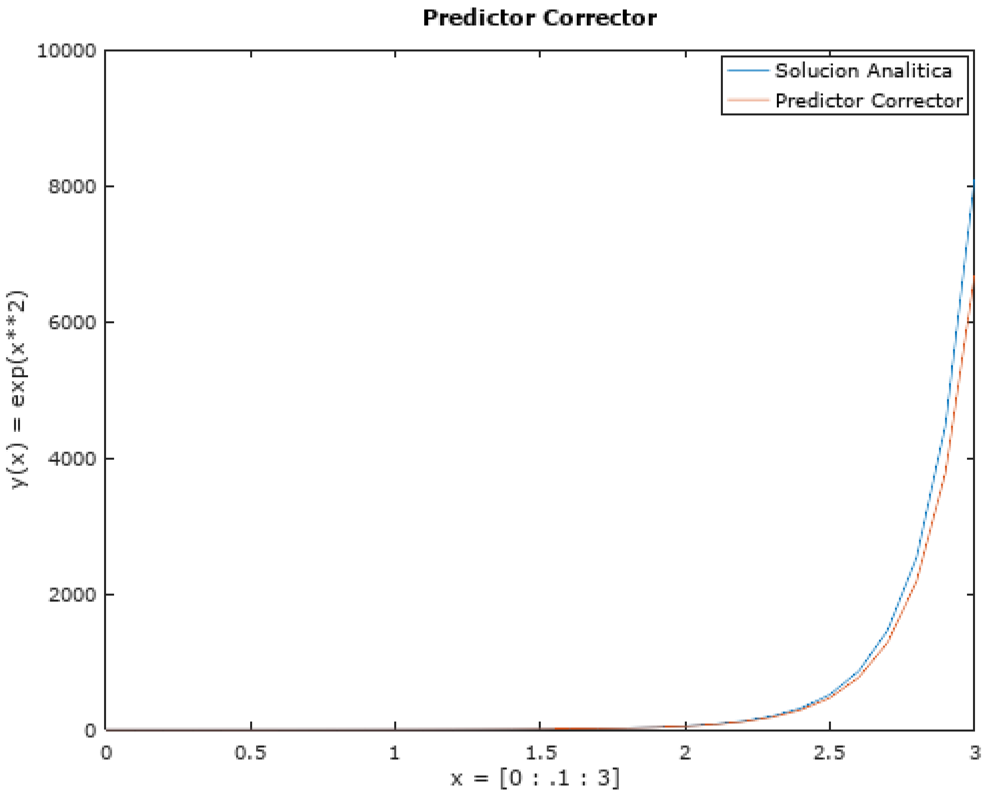


Figura N°10: Método de Predictor Corrector Ecuación 2

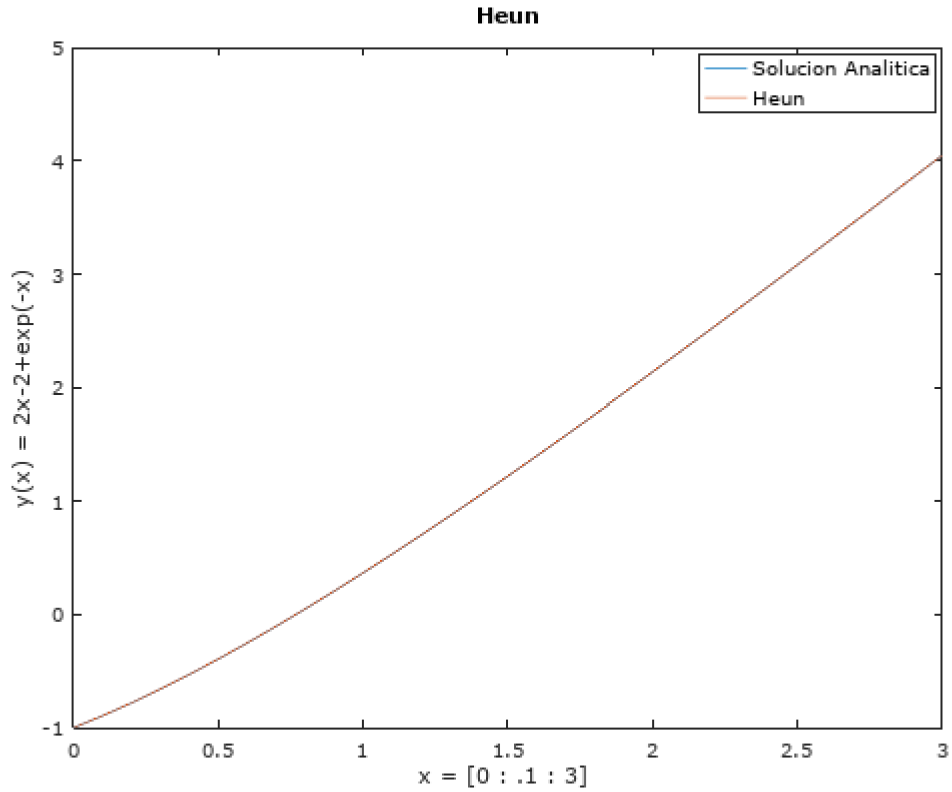


Figura N°11: Método de Heun Ecuación 1.

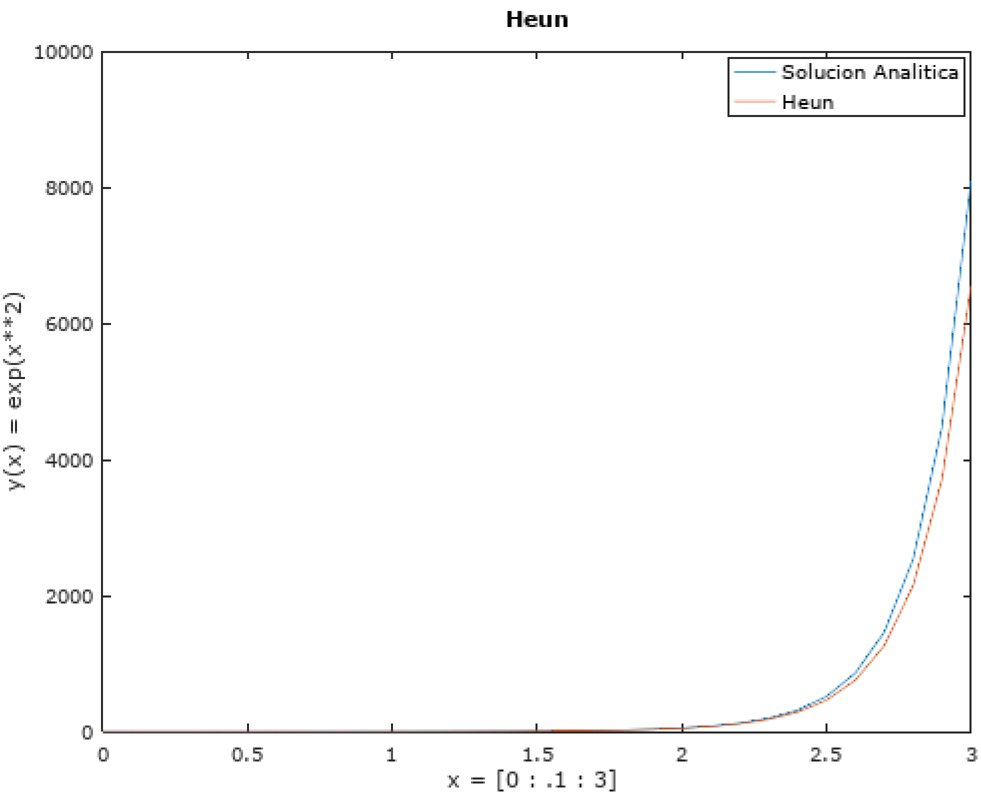


Figura N°12: Método de Heun Ecuación 2.

Ejercicio N°5:

Octave difiere de Matlab en lo que refiere a la resolución de ecuaciones diferenciales. En Matlab paquetes de solución se encuentran con el nombre ode23 y ode45 y otros. En Octave la función que se usa es lsode. Escribir en la ventana de Octave "help lsode" para ver su uso.

Use el paquete lsode para hallar aproximaciones de las soluciones de las ecuaciones (1) y (2) en [0; 3]. La función linspace (0,3,15) y linspace (0,3,30) se puede usar para poder comparar el método dado en Octave con los implementados.

Ecuación 1 (y' = 2x - y)	
Octave lsode	Solución Analítica
-1	-1
-0.76431	-0.76431
-0.49142	-0.49142
-0.1885	-0.1885
0.13866	0.13866
0.48538	0.48538
0.84788	0.84788
1.22313	1.22313
1.60866	1.60866
2.0025	2.0025
2.40303	2.40303
2.80898	2.80898
3.21928	3.21928
3.63311	3.63311
4.04979	4.04979

Ecuación 2 (y' =2x * y)	
Octave lsode	Solución Analítica
1	1
1.047	1.047
1.2016	1.2016
1.5117	1.5117
2.0848	2.0848
3.1518	3.1518
5.2229	5.2229
9.4877	9.4877
18.893	18.8927
41.239	41.2391
98.676	98.6755
258.82	258.8173
744.15	744.1517
2345.4	2345.3832
8103.1	8103.0839

Ejercicio N°6:

Hacer tablas en que se muestren los valores dados por la solución analítica de cada ecuación y las correspondientes aproximaciones dadas por los métodos numéricos (incluyendo el obtenido con lsode).

A continuación, se presentan tablas de valores con los resultados de los diferentes métodos para un "paso" de aproximación numérica derivada de las siguientes funciones linspace (0,3,15) y linspace (0,3,30).

Tabla de Valores						
Ecuación 1 (y' = 2x - y)		Solución Analítica (2x+ e^(-x)-2)			x pertenece a [0; 3], h = 0.214	
Puntos	Forward Euler	Backward Euler	Predictor - Corrector	Heun	Octave lsode	Solución Analítica
0	-1	-1	-1	-1	-1	-1
0.21429	-0.78571	-0.7479	-0.76276	-0.76276	-0.76431	-0.76431
0.42857	-0.52551	-0.46466	-0.4889	-0.4889	-0.49142	-0.49142
0.64286	-0.22923	-0.15577	-0.18545	-0.18545	-0.1885	-0.1885
0.85714	0.0954	0.17424	0.14194	0.14194	0.13866	0.13866
1.07143	0.44231	0.52164	0.48869	0.48869	0.48538	0.48538
1.28571	0.80671	0.88337	0.85109	0.85109	0.84788	0.84788
1.5	1.18486	1.25689	1.22616	1.22616	1.22313	1.22313
1.71429	1.57382	1.64013	1.61146	1.61146	1.60866	1.60866
1.92857	1.97127	2.03137	2.00504	2.00504	2.0025	2.0025
2.14286	2.37538	2.42919	2.40531	2.40531	2.40303	2.40303
2.35714	2.78474	2.83245	2.811	2.811	2.80898	2.80898
2.57143	3.19821	3.24017	3.22107	3.22107	3.21928	3.21928
2.78571	3.61492	3.65156	3.63468	3.63468	3.63311	3.63311
3	4.03417	4.06599	4.05115	4.05115	4.04979	4.04979

Ecuación 2 (y' =2x * y)		Solución Analítica (e^(x^2))			x pertenece a [0; 3], h = 0.214	
Puntos	Forward Euler	Backward Euler	Predictor - Corrector	Heun	Octave lsode	Solución Analítica
0	1	1	1	1	1	1
0.21429	1	1	1.0459	1.0459	1.047	1.047
0.42857	1.0918	1.0918	1.1988	1.1974	1.2016	1.2016
0.64286	1.2924	1.2924	1.5044	1.4992	1.5117	1.5117
0.85714	1.6484	1.6484	2.0641	2.0506	2.0848	2.0848
1.0714	2.254	2.254	3.0912	3.0595	3.1518	3.1518
1.2857	3.289	3.289	5.0436	4.9704	5.2229	5.2229
1.5	5.1013	5.1013	8.9476	8.7757	9.4877	9.4877
1.7143	8.3807	8.3807	17.223	16.806	18.893	18.8927
1.9286	14.538	14.538	35.898	34.839	41.239	41.2391
2.1429	26.554	26.554	80.841	78.017	98.676	98.6755
2.3571	50.94	50.94	196.29	188.34	258.82	258.8173
2.5714	102.4	102.4	512.87	489.18	744.15	744.1517
2.7857	215.25	215.25	1439	1364.3	2345.4	2345.3832
3	472.23	472.23	4327.5	4077.9	8103.1	8103.0839

Ecuación 1 (y' = 2x - y)		Solución Analítica (2x+ e^(-x)-2)			x pertenece a [0; 3], h = 0.103	
Puntos	Forward Euler	Backward Euler	Predictor - Corrector	Heun	Octave lsode	Solución Analítica
0	-1	-1	-1	-1	-1	-1
0.10345	-0.89655	-0.88685	-0.8912	-0.8912	-0.89138	-0.89138
0.2069	-0.7824	-0.76492	-0.77278	-0.77278	-0.7731	-0.7731
0.31034	-0.65866	-0.63502	-0.64568	-0.64568	-0.64612	-0.64612
0.41379	-0.52631	-0.4979	-0.51075	-0.51075	-0.51128	-0.51128
0.51724	-0.38625	-0.35424	-0.36876	-0.36876	-0.36935	-0.36935
0.62069	-0.23928	-0.20465	-0.2204	-0.2204	-0.22105	-0.22105
0.72414	-0.08611	-0.04969	-0.0663	-0.0663	-0.06698	-0.06698
0.82759	0.07262	0.11014	0.09297	0.09297	0.09228	0.09228
0.93103	0.23633	0.27439	0.25692	0.25692	0.25621	0.25621
1.03448	0.40451	0.44263	0.42509	0.42509	0.42438	0.42438
1.13793	0.5767	0.61449	0.59705	0.59705	0.59634	0.59634
1.24138	0.75247	0.78964	0.77244	0.77244	0.77174	0.77174
1.34483	0.93147	0.96777	0.95092	0.95092	0.95024	0.95024
1.44828	1.11335	1.14859	1.13218	1.13218	1.13153	1.13153
1.55172	1.29782	1.33186	1.31597	1.31597	1.31533	1.31533
1.65517	1.48461	1.51734	1.50201	1.50201	1.5014	1.5014
1.75862	1.67348	1.70483	1.69011	1.69011	1.68952	1.68952
1.86207	1.86421	1.89414	1.88005	1.88005	1.87949	1.87949
1.96552	2.05662	2.0851	2.07165	2.07165	2.07112	2.07112
2.06897	2.25052	2.27756	2.26475	2.26475	2.26425	2.26425
2.17241	2.44577	2.47136	2.45921	2.45921	2.45873	2.45873
2.27586	2.64223	2.6664	2.65488	2.65488	2.65443	2.65443
2.37931	2.83976	2.86254	2.85166	2.85166	2.85124	2.85124
2.48276	3.03826	3.0597	3.04943	3.04943	3.04903	3.04903
2.58621	3.23763	3.25776	3.2481	3.2481	3.24772	3.24772
2.68966	3.43778	3.45666	3.44757	3.44757	3.44721	3.44721
2.7931	3.63863	3.6563	3.64777	3.64777	3.64744	3.64744
2.89655	3.8401	3.85663	3.84863	3.84863	3.84832	3.84832
3	4.04214	4.05757	4.05008	4.05008	4.04979	4.04979

Ecuación 2 (y' =2x * y)		Solución Analítica (e^(x^2))			x pertenece a [0; 3], h = 0.103	
Puntos	Forward Euler	Backward Euler	Predictor - Corrector	Heun	Octave lsode	Solución Analítica
0	1	1	1	1	1	1
0.10345	1	1	1.0107	1.0107	1.0108	1.0108
0.2069	1.0214	1.0214	1.0436	1.0435	1.0437	1.0437
0.31034	1.0651	1.0651	1.1009	1.1006	1.1011	1.1011
0.41379	1.1335	1.1335	1.1864	1.1859	1.1868	1.1868
0.51724	1.2306	1.2306	1.3061	1.3052	1.3067	1.3067
0.62069	1.3622	1.3622	1.4688	1.4673	1.47	1.47
0.72414	1.5372	1.5372	1.6873	1.6848	1.6894	1.6894
0.82759	1.7675	1.7675	1.9798	1.976	1.9836	1.9836
0.93103	2.0701	2.0701	2.3726	2.3669	2.3793	2.3793
1.0345	2.4689	2.4689	2.9039	2.8953	2.9158	2.9158
1.1379	2.9973	2.9973	3.6297	3.6167	3.6506	3.6506
1.2414	3.703	3.703	4.6328	4.6132	4.6694	4.6694
1.3448	4.654	4.654	6.0378	6.008	6.1017	6.1017
1.4483	5.949	5.949	8.0341	7.9885	8.1458	8.1458
1.5517	7.7316	7.7316	10.914	10.843	11.11	11.11
1.6552	10.214	10.214	15.134	15.024	15.481	15.4807
1.7586	13.711	13.711	21.422	21.248	22.038	22.0375
1.8621	18.7	18.7	30.947	30.668	32.05	32.0501
1.9655	25.905	25.905	45.625	45.171	47.62	47.6202
2.069	36.439	36.439	68.638	67.89	72.285	72.2851
2.1724	52.037	52.037	105.36	104.1	112.1	112.0989
2.2759	75.426	75.426	164.99	162.86	177.6	177.6025
2.3793	110.94	110.94	263.56	259.89	287.47	287.4698
2.4828	165.56	165.56	429.45	423.01	475.37	475.3685
2.5862	250.6	250.6	713.66	702.18	803.09	803.0895
2.6897	384.69	384.69	1209.4	1188.6	1386.1	1386.0939
2.7931	598.76	598.76	2089.8	2051.5	2444.1	2444.0869
2.8966	944.77	944.77	3681.8	3610.1	4402.9	4402.87
3	1511	1511	6612.3	6475.9	8103.1	8103.0839

Ejercicio N°7:

Graficar los resultados obtenidos y graficar el error local y global en cada caso. Analice las situaciones dadas por la ecuación (1) y por la ecuación (2) al respecto de cómo se comporta el error global en cada caso.

Error Local: es el error cometido en cada paso contra la solución $y(x, x_j, y_j)$ que cumple $y(x, x_j, y_j) = y_j$

Calculamos $y(x, x_j, y_j)$ mediante los siguientes módulos:**"sol_discreta1"** y **"sol_discreta2"**

Error Global: es el error entre la verdadera solución $y(x) = y(x, x_j, y_j)$ e y_{j+1} , o sea:

$$|y(x_{j+1}) - y_{j+1}|$$

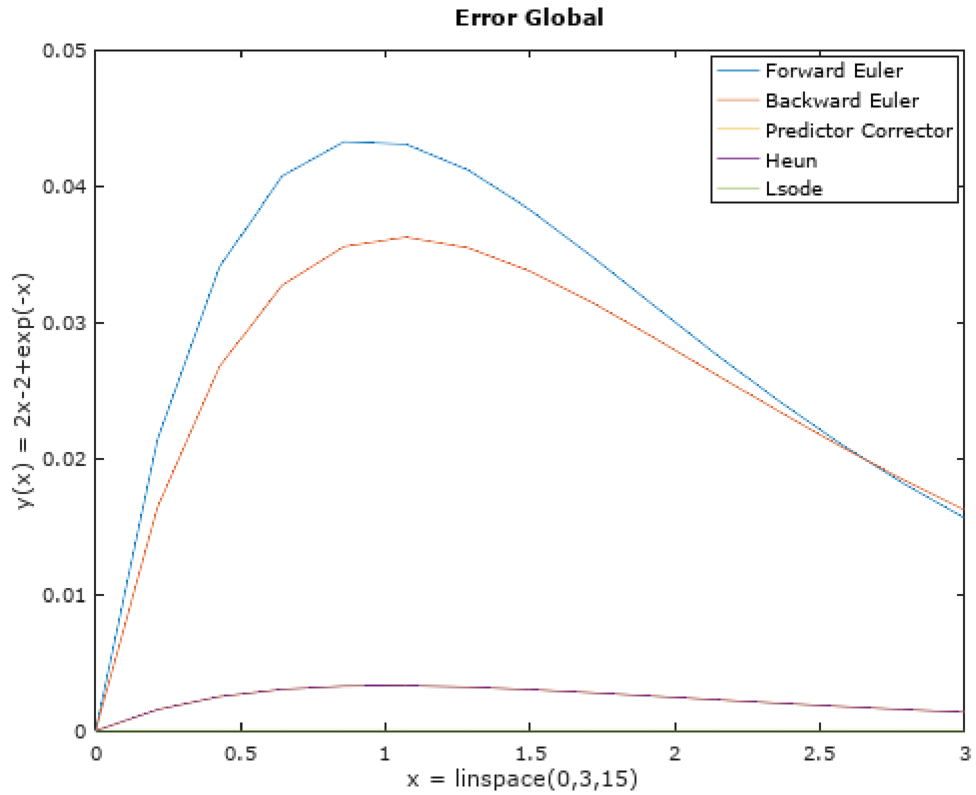


Figura N°13: Error Global por Método Ecuación 1, h =0.2.

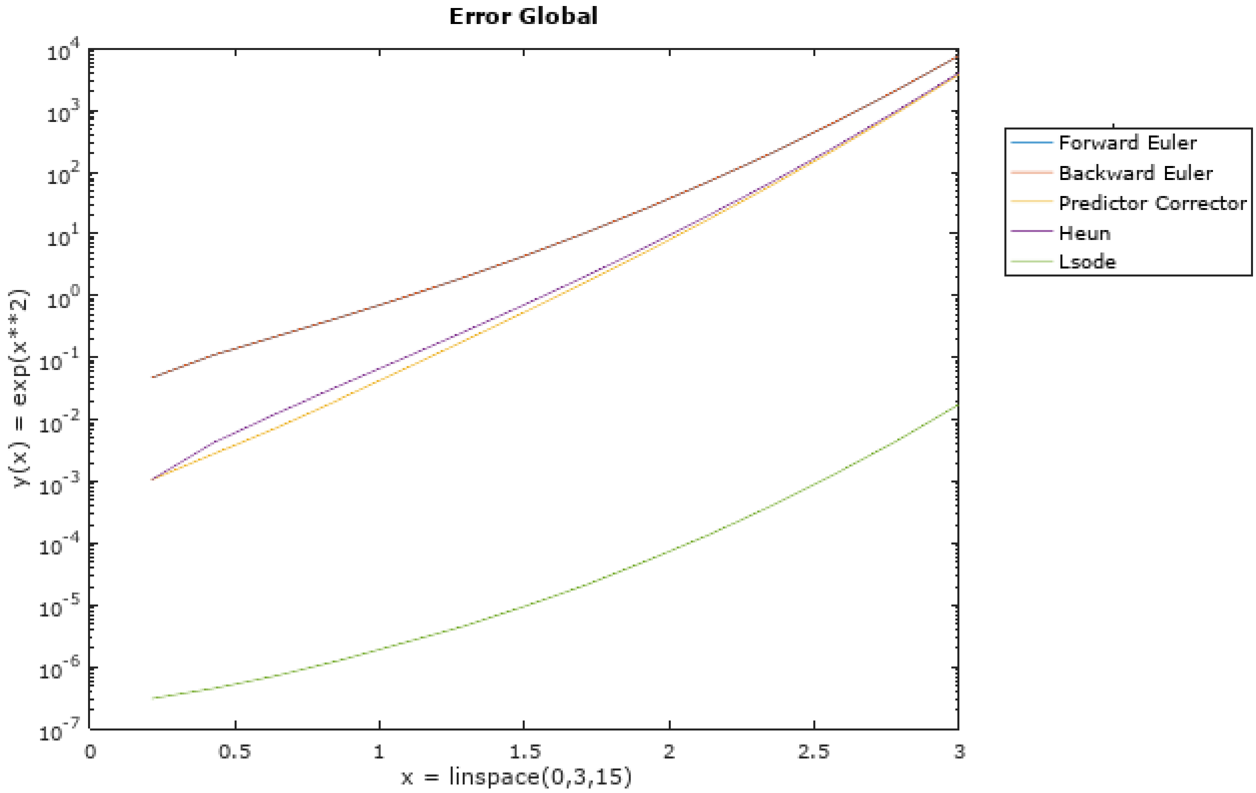


Figura N°14: Error Global por Método Ecuación 2, h =0.2.

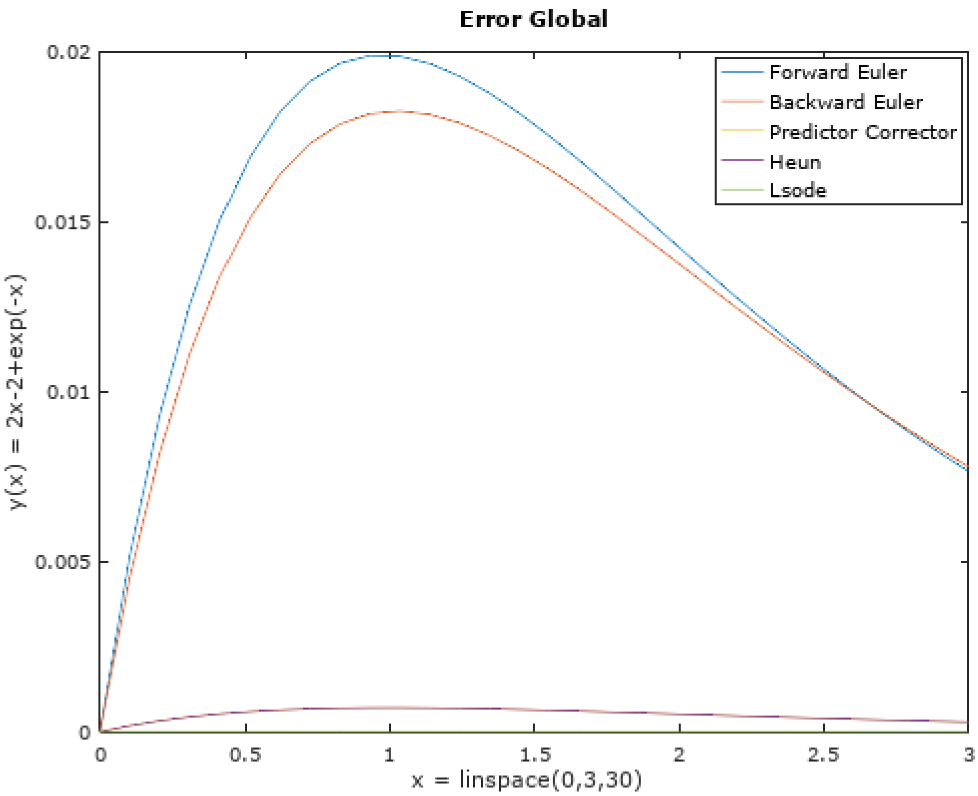


Figura N°15: Error Global por Método Ecuación 1, h =0.1.

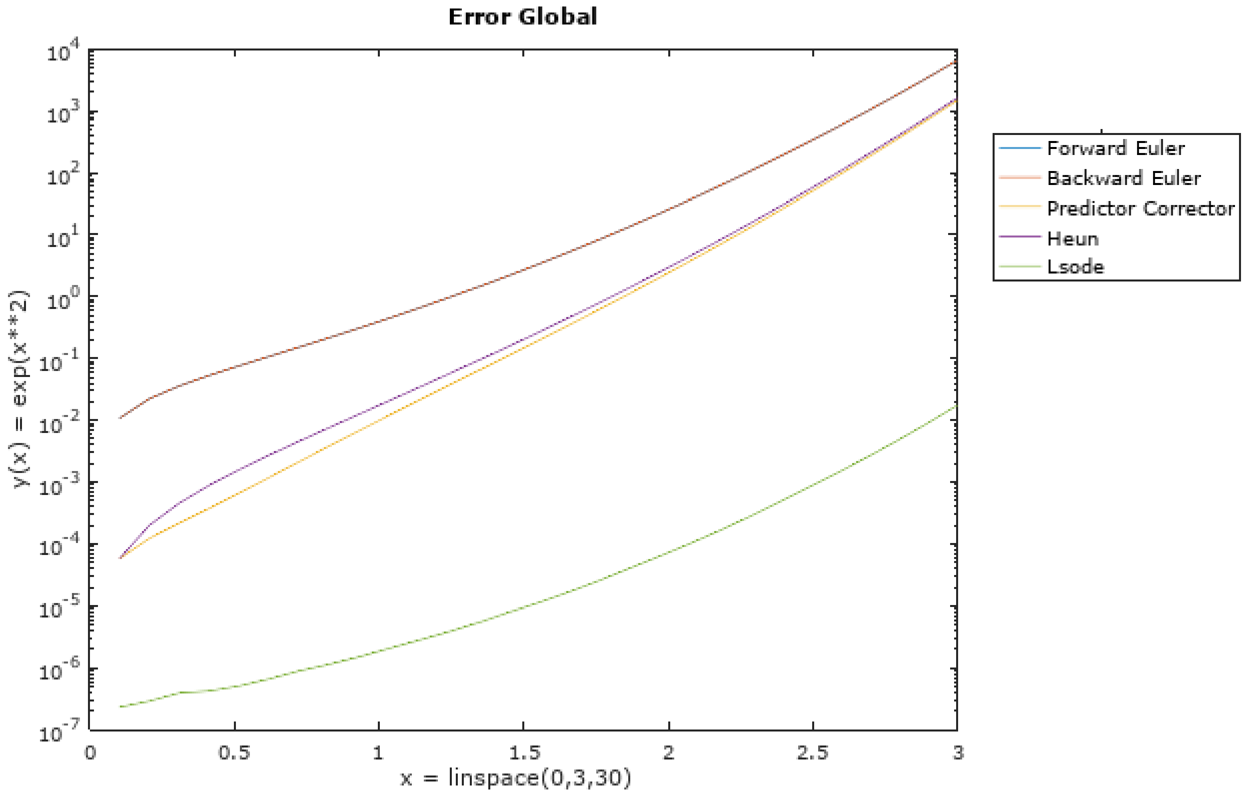


Figura N°16: Error Global por Método Ecuación 2, h =0.1.

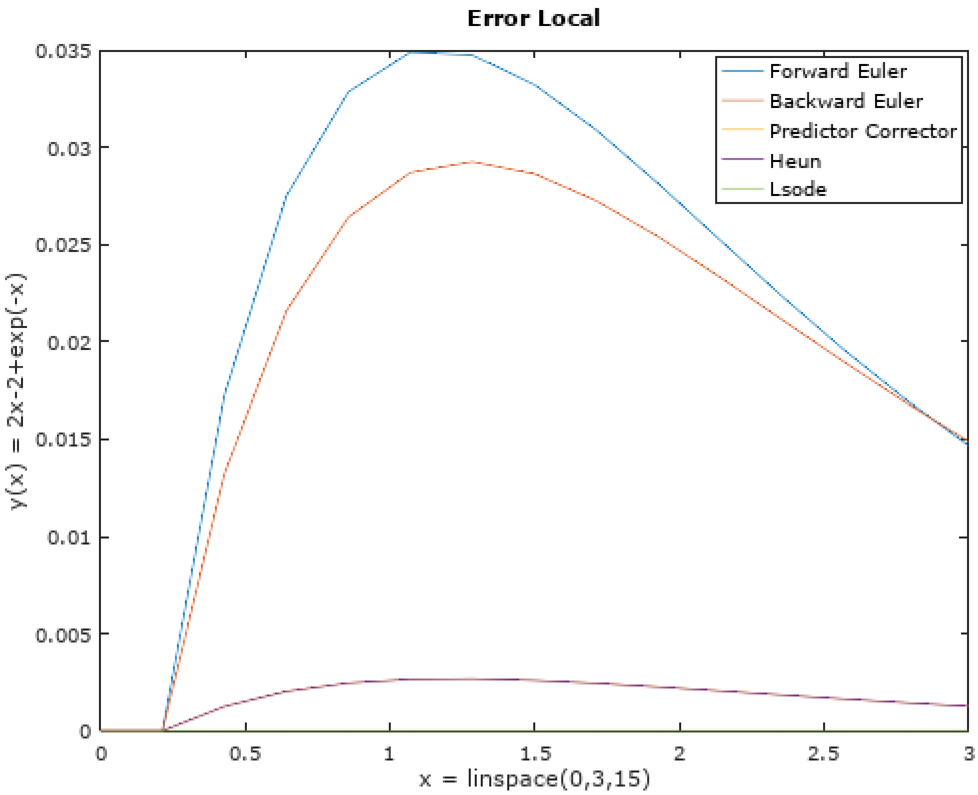


Figura N°17: Error Local por Método Ecuación 1, h =0.2.

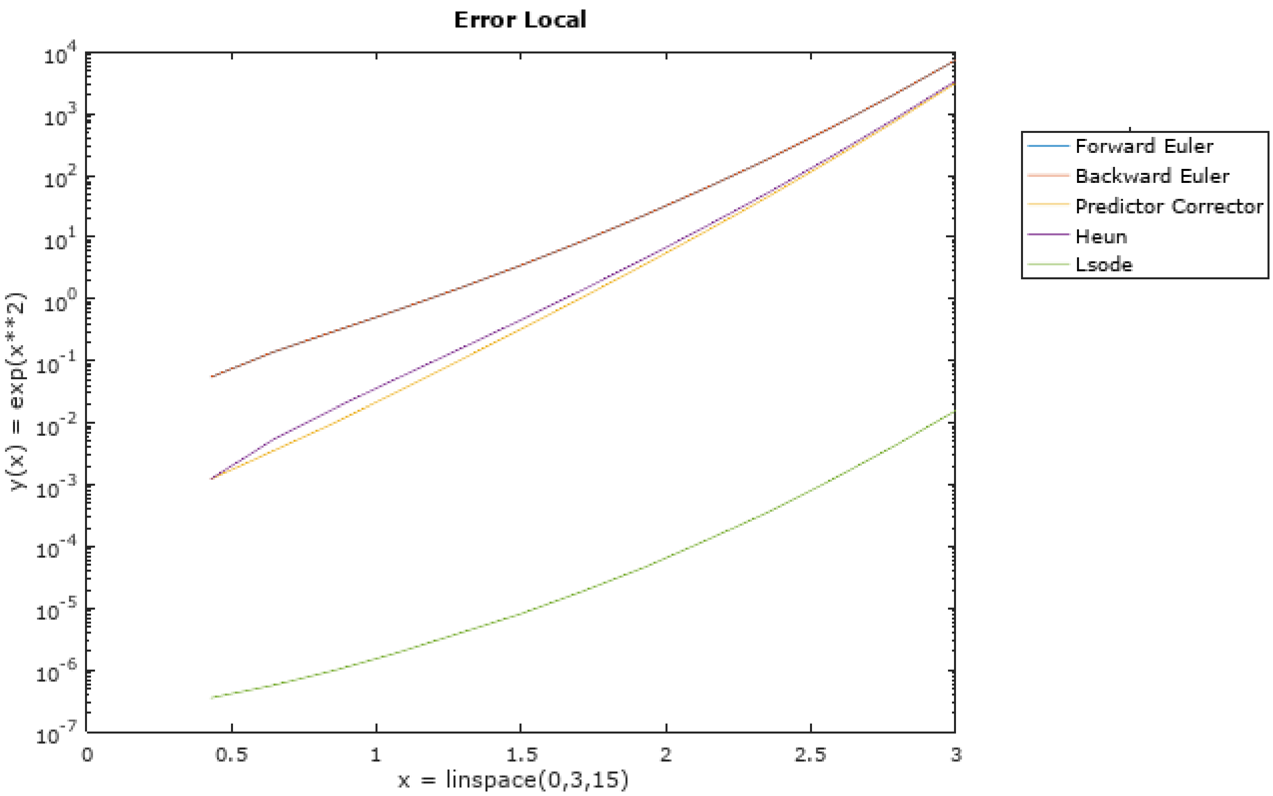


Figura N°18: Error Local por Método Ecuación 2, h =0.2.

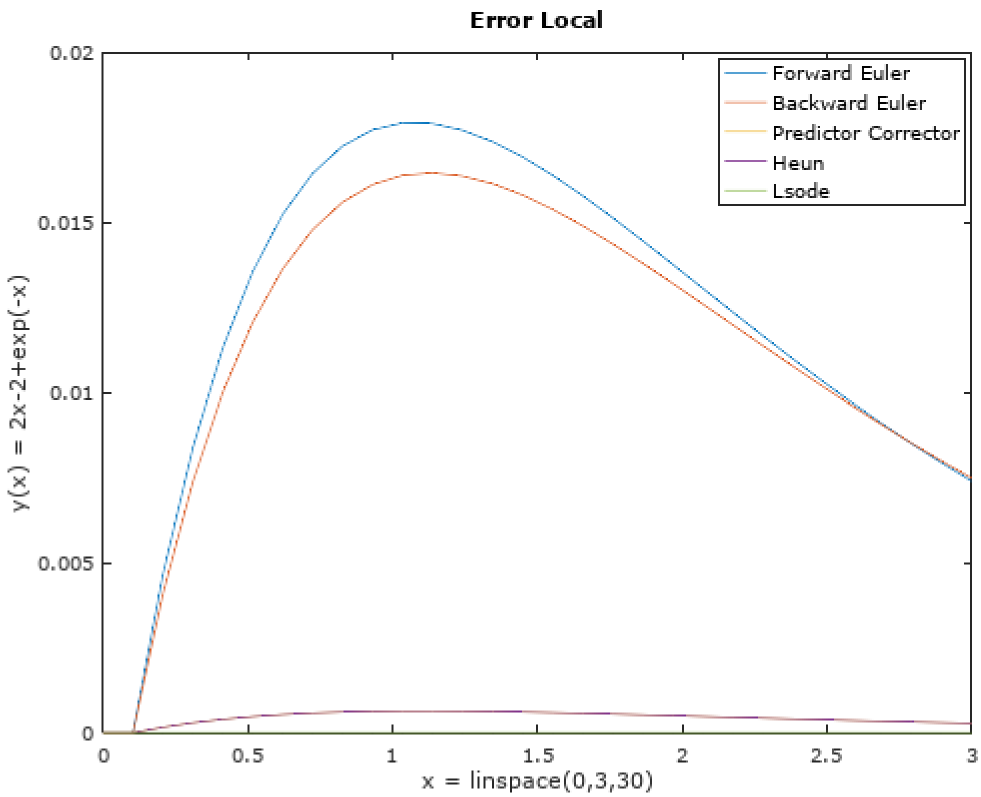


Figura N°19: Error Local por Método Ecuación 1, h =0.1.

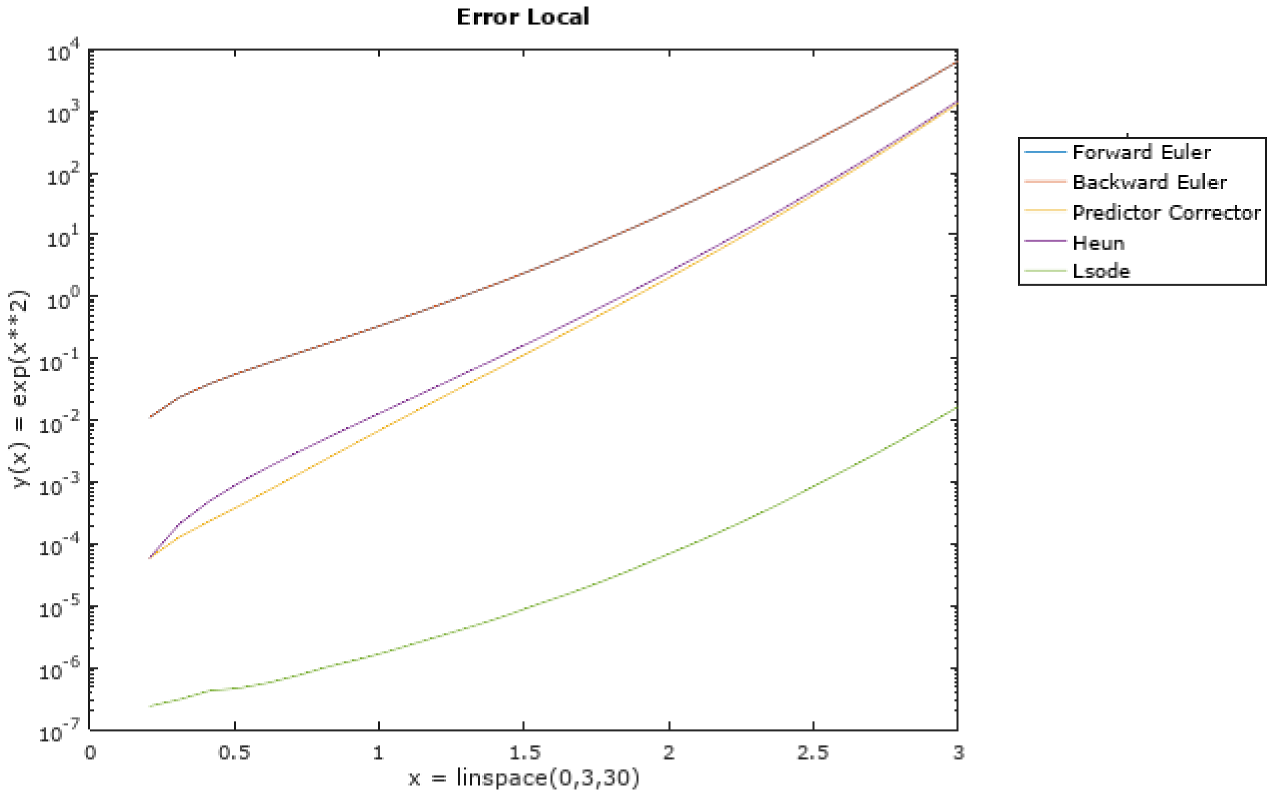


Figura N°20: Error Local por Método Ecuación 2, h =0.1.

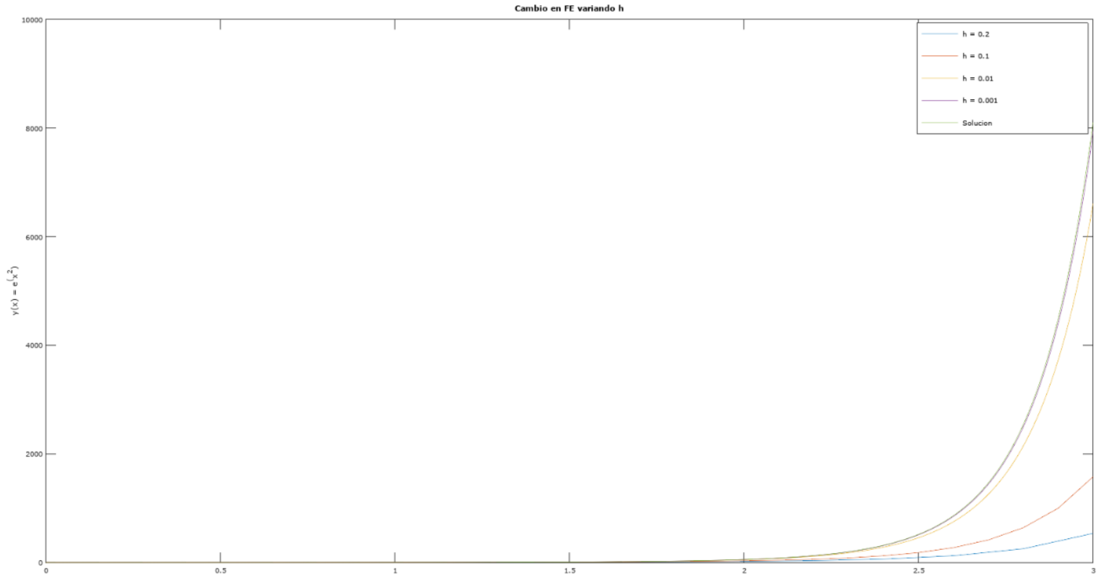
Conclusiones:

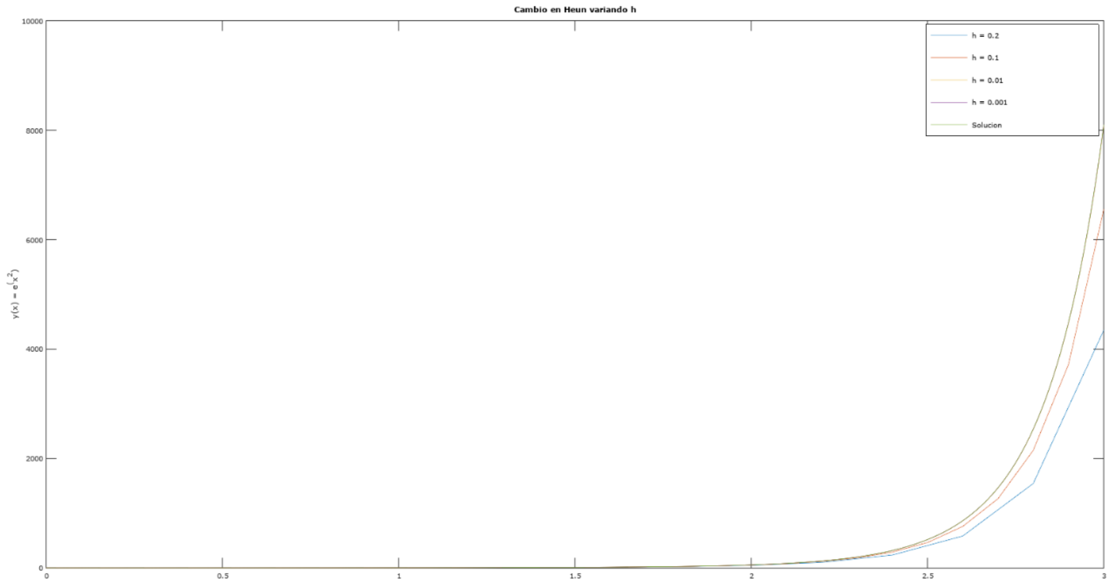
Al comparar los resultados obtenidos por los métodos desarrollados anteriormente (backward Euler, forward Euler, predictor corrector y Heun) se concluye que los métodos de Forward Euler y Backward Euler necesitan un paso de aproximación numérica “h” más pequeño para lograr obtener resultados un más preciso. Mientras que con Heun y predictor corrector se obtienen resultados aceptables con valores de “h” sugerido.

En la primera ecuación podemos observar que con cualquiera de los métodos utilizados con los valores de “h” sugerido obtenemos buenos resultados, con errores menores que 10^{-2}, aunque como se puede observar en las gráficas y tablas de errores, los metros de Heun y predictor correceptor aproximan significativamente mejor a la solución. Creemos que este fenómeno es debido al tipo de ecuación diferencial, ecuaciones que cambian de forma gradual.

En la segunda ecuación se nota una diferencia importante entre los métodos implementados. Para los métodos de Forward y backward Euler con los “h” sugerido, la aproximación numérica no es precisa dándonos errores aproximadamente de $7 * 10^{3}$. Mientras que con los métodos de Heun y predictor corrector en los mismos valores de “h” la aproximación numérica es más precisa pero aun presenta errores de $1.5 * 10^{3}$. Esto se debe a que la ecuación diferencial cambia de forma drástica en intervalos pequeños, lo que hace que la aproximación numérica sea más compleja.

Para obtener una aproximación más precisa en la segunda ecuación, se recomienda utilizar valores de “h” más pequeños, a continuación se muestra un gráfico que refleja como varía la aproximación numérica al cambiar “h”





Últimos valores obtenidos al cambiar "h" y su diferencia				
h	0.2	0.1	0.01	0.001
Backward Euler	536.3107	1579.184	6608	7935.1915
Heun	4346.1919	6556.071	8079.6717	8102.8417
Solucion Analitica	8103.0839	8103.0839	8103.0839	8103.0839
Error Euler	7566.7732	6523.8999	1495.0839	167.8924
Error Heun	3756.892	1547.0129	23.4122	0.2422

Anexo:

```
function [y1 y2] = forward_euler (h)
  #Generamos vector con los puntos de evaluación
  x = [0:h:3];

  #Calculamos el tamaño
  tam = length(x);

  #Generamos los vectores donde guardar los resultados
  y1 = ones(1,tam);
  y2 = ones(1,tam);

  #Actualizamos valores iniciales
  y1(1) = -1;
  y2(1) = 1;

  for i = 1 : tam-1
    #Calculamos soluciones mediante el metodo explicito de Euler
    y1(i+1) = y1(i) + (2*x(i)-y1(i))*h;
    y2(i+1) = y2(i) + (2*x(i)*y2(i))*h;
  endfor
```

```
function [y1 y2] = backward_euler(h);
  f1=inline('2*x-y');
  f2=inline('2*x*y');

  x = [0:h:3];
  tam = length(x);
#vector con los resultados obtenidos
  y1 = [-1 ones(1,tam-1)];
  y2 = [1 ones(1,tam-1)];
  for i = 1 : tam-1
    ynew1 = y1(i) + f1(x(i),y1(i))*h;
    ynew2 = y2(i) + f2(x(i),y2(i))*h;
    check1 = 1000;
    check2 = 1000;
  #NR del la primera ecuación
    while check1 > 10**-9
     yold1 = ynew1;
     ynew1 = yold1 - ( (y1(i) - yold1 + h*(2*x(i+1)-yold1)) / (h-1));
     check1 = abs(ynew1 - yold1);
      endwhile
      y1(i+1) = ynew1;
  #NR de la segunda ecuacion
    while check2 < 10**-4
      yold2 = ynew2;
      ynew2 = yold2 - (( y2(i) - yold2 + (h*2*x(i+1)*yold2) )/(-1+(h*2*x(i+1)) ));
      check2 = abs(yold2 -ynew2);
      endwhile
      y2(i+1) = ynew2;
     endfor
```

```
function [y1 y2]= predictor_corrector(h)

  #definimos las funciones que utilisaremos

  f1 = inline('2*x - y');

  f2 = inline('2*x*y');

  x = [0:h:3] ;

  #calculamos el tamano correspondiente

    tam = length(x);

  #vector que contiene las sol

  y1 = [-1 ones(1,tam-1)];

  y2 = [1 ones(1,tam-1)];

  for i = 1:tam-1

    #predictor

    y1(i+1) = y1(i) + f1(x(i),y1(i))*h;

    y2(i+1) = y2(i) + f2(x(i),y2(i))*h;

    #corrector

    y1(i+1) = y1(i) + (h/2)*(
f1(x(i),y1(i))+f1(x(i+1),y1(i+1)));

    y2(i+1) = y2(i) + (h/2)*(
f2(x(i),y2(i))+f2(x(i+1),y2(i+1)));

    endfor
```

```
function [y1 y2]= heun(h)
  #definimos las funciones a utilizar
  f1 = inline('2*x - y');
  f2 = inline('2*x*y');
  #generamos el vector con los puntos de evaluación
  x = [0:h:3] ;
  #calculamos el tamaño del vector que contiene los
resultados
  tam = length(x);
  #vector solucion con los valores iniciales
  y1 = [-1 ones(1,tam-1)];
  y2 = [1 ones(1,tam-1)];
  #los valores definido por el metodo de Heun
  g1 = 1/4;
  g2 = 3/4;
  a = 2/3;
  b = 2/3;
  #vector que conitene los valores de k correspondiente a
la ecuacion
  k1 = ones(1,2);
  k2 = ones(1,2);
  for i = 1:tam-1
    #calculamos los k1
    k1(1)  = f1(x(i),y1(i));
    k1(2)  = f2(x(i),y2(i));
    #calculamos los k2
    k2(1)  = f1( (x(i)+a*h) , (y1(i)+b*k1(1)*h) );
    k2(2)  = f2( (x(i)+a*h) , (y2(i)+b*k1(2)*h) );
    #calculamos los valores los valores de la ecuación
mediante el método

    #de Runge-Kuta con los valores definido para el
método de Heun

    y1(i+1) = y1(i) + h*(g1*k1(1)+g2*k2(1));

    y2(i+1) = y2(i) + h*(g1*k1(2)+g2*k2(2));

    endfor
```

```
function res =
sol_discreta1(y0,x0,x1)

  #calculamos el C
correspondiente a la funcion

  c = (y0 - 2*x0 + 2)*exp(x0);

  #retornamos el valor de xi+1

  res = 2*x1 - 2 + c*exp(-x1);

function sol =
sol_discreta2(y0,x0,x1)

  if (y0 == 0)

    c = 0;

    else

    c = log(y0) - x0**2;

    endif


    sol = exp( ((x1**2)+c) );
```