

Analysis of general algorithms of numeric solutions
of ordinary differential equations of higher order one
with initial conditions.

Carlos Oscar Rodríguez Leal.

3 de junio de 2015

In this paper several existing numerical methods of solution of ordinary differential equations of first order with initial conditions are modified , so that they can be generalized as methods of solution of ordinary differential equations of order n .

1. Introducción

En este trabajo se adaptan los métodos ya existentes de soluciones numéricas de ecuaciones diferenciales ordinarias de orden uno con valores en la frontera para así generalizarlos como métodos de soluciones numéricas de ecuaciones diferenciales ordinarias de orden mayor a uno con valores en la frontera (el caso de orden uno es el método original, sin modificar). Los métodos a adaptar son el de Euler, el de Heun y los de Taylor de orden mayor. Además se hace un análisis de convergencia hacia la solución exacta y del orden de convergencia en cada método.

Nota: En el libro [1] consideran un orden n para los respectivos algoritmos, siendo que yo considero para los mismos el orden del error de truncamiento local de tipo $O(h^{n+1})$, ya que tomo en cuenta el término del error de los polinomios de Taylor, que es del orden $n + 1$ para polinomios de orden n , siendo que éste término nos representa efectivamente el error de nuestra aproximación respecto a la función exacta.

2. Método de Euler-Leal o de Euler generalizado.

2.1. Algoritmo de Euler-Leal

Como sabemos, el método de Euler es un método numérico para solución de ecuaciones diferenciales de orden uno expresadas de la forma $y' = f(x, y)$, con la condición inicial $y(x_0) = y_0$ [1, 2]. Por lo tanto, para una ecuación de segundo orden despejada de la forma

$$y'' = f(x, y, y'), \quad y(x_0) = y_0, \quad y'(x_0) = w_0,$$

podemos modificar el algoritmo de Euler sabiendo que y'_n nos dará la pendiente de la tangente de y en el punto x_n y que $y''_n = f(x_n, y_n, y'_n)$ es la pendiente de la tangente a y' en el punto x_n . Así, el algoritmo adaptado escrito en matlab será:

```
"function z = fn2 (t, y, w)

    z = 1 + (w - t)^2;

end

function y = fa2(x)

    y = -log(abs(x-2)) + 1/2*x^2 + log(2);
```

```

end

function Euler_Leal2 (t0, tf, y0, w0, N)

    dt = (tf - t0) / N;
    t = zeros(N+1,1);
    y = zeros(N+1,1);
    w = zeros(N+1,1);
    x = zeros(N+1,1);

    cero = 1;

    t(cero) = t0;
    y(cero) = y0;
    w(cero) = w0;
    x(cero) = fa2(t(cero));

    for n=cero:N-1+cero
        t(n+1) = t(n) + dt;
        y(n+1) = y(n) + w(n)*dt;
        w(n+1) = w(n) + fn2( t(n),y(n),w(n) )*dt;
        x(n+1) = fa2(t(n+1));
    end

    figure;
    plot (t, x, 'r');
    hold on;
    plot(t, y, 'g');

    Error = x - y;
    Error_relat = Error./x;

    for n=1:N+1
        if (x(n) != 0)
            Error_relat(n) = (x(n) - y(n))/x(n);
        else
            Error_relat(n) = 0;
        end
    end

    end

    M = zeros (N+1,5);
    disp ("t   x   y   error   error_relativo:");
    M = [t x y Error Error_relat]

```

end

donde hemos puesto la ecuación de ejemplo $y'' = 1 + (y' - t)^2$, cuya solución analítica es $y = -\ln|x - 2| + \frac{1}{2}x^2 + \ln(2)$;

Análogamente, podemos generalizar dicho método para una ecuación diferencial

$$y^{(n)} = f(t, y, y', \dots, y^{(n-1)}), y(x_0) = y_0, y'(x_0) = y'_0, \dots, y^{(n-1)}(x_0) = y_0^{(n-1)}$$

mediante un algoritmo general, cuyo núcleo es

```
desde i = 0, hasta i < Numero de particiones, de uno en uno:
{
  t_(i+1) = t_(i) + dt;
  y_(i+1) = y_(i) + y'_(i)*dt;
  y'_(i+1) = y'_(i) + y''_(i)*dt;
  .
  .
  .
  y^(n-1)_(i+1) = y^(n-1)_(i) +
  + f( t_(i), y_(i), y'_(i), ... , y^(n-1)_(i) ) * dt;
}
```

2.2. Análisis de convergencia.

2.2.1. Análisis gráfico.

A continuación ponemos a prueba el método anterior mediante la obtención de soluciones numéricas de ecuaciones para las cuales se conoce su solución analítica y comparando ambas soluciones. Las soluciones analíticas aparecen en rojo y las numéricas en verde.

Para ecuaciones de orden dos tendremos:

1.

$$y'' = 1 + (y' - t)^2, 0 \leq t \leq 1,9, y(0) = 0, y'(0) = 1/2, N = 20,$$

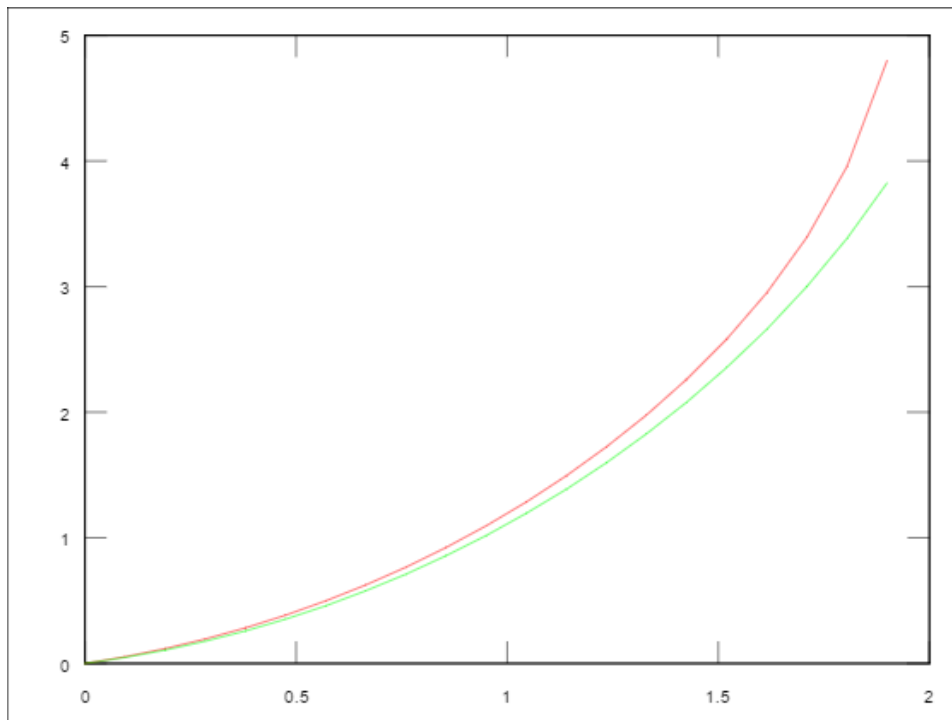
cuya solución analítica es

$$y = -\ln|t - 2| + \frac{1}{2}t^2 + \ln(2).$$

Tabla de resultados:

```
octave:1> Euler_Leal2(0, 1.9, 0, 1/2, 20)
```

t	Sol. ana.	Sol. num.	Error	Error relativo.
0.00000	0.00000	0.00000	0.00000	0.00000
0.09500	0.05318	0.04750	0.00568	0.10677
0.19000	0.11787	0.10628	0.01159	0.09832
0.28500	0.19435	0.17656	0.01778	0.09150
0.38000	0.28292	0.25860	0.02432	0.08597
0.47500	0.38397	0.35268	0.03129	0.08148
0.57000	0.49792	0.45915	0.03878	0.07788
0.66500	0.62533	0.57840	0.04693	0.07504
0.76000	0.76684	0.71092	0.05591	0.07292
0.85500	0.92326	0.85727	0.06598	0.07147
0.95000	1.09561	1.01815	0.07746	0.07070
1.04500	1.28520	1.19440	0.09081	0.07066
1.14000	1.49377	1.38706	0.10671	0.07144
1.23500	1.72364	1.59747	0.12616	0.07320
1.33000	1.97807	1.82734	0.15074	0.07620
1.42500	2.26184	2.07889	0.18295	0.08089
1.52000	2.58232	2.35515	0.22717	0.08797
1.61500	2.95177	2.66028	0.29149	0.09875
1.71000	3.39307	3.00027	0.39280	0.11577
1.80500	3.95692	3.38409	0.57283	0.14477
1.90000	4.80073	3.82593	0.97480	0.20305



2.

$$y'' = e^x, \quad 0 \leq t \leq 3, \quad y(0) = 1, \quad y'(0) = 0, \quad N = 20,$$

cuya solución analítica es

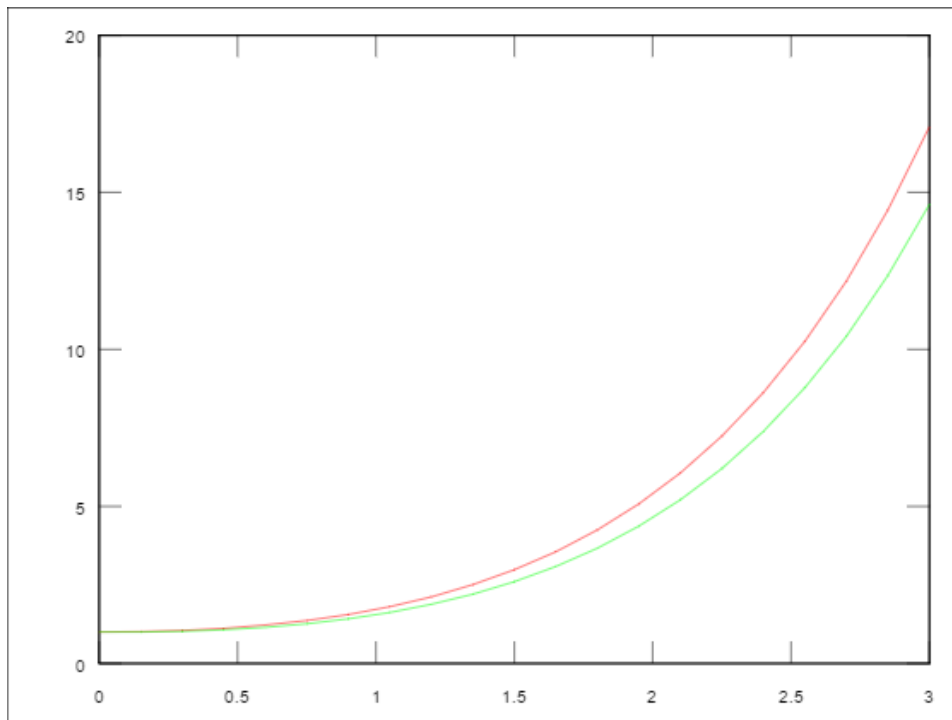
$$y = -\ln|t - 2| + \frac{1}{2}t^2 + \ln(2).$$

Tabla de resultados:

```
octave:1> Euler_Leal2(0, 3, 1, 0, 20)
```

t	Sol. ana.	Sol. num.	Error	Error relativo.
0.00000	1.00000	1.00000	0.00000	0.00000
0.15000	1.01183	1.00000	0.01183	0.01170
0.30000	1.04986	1.02250	0.02736	0.02606
0.45000	1.11831	1.07114	0.04717	0.04218
0.60000	1.22212	1.15015	0.07196	0.05888
0.75000	1.36700	1.26445	0.10255	0.07502
0.90000	1.55960	1.41975	0.13985	0.08967
1.05000	1.80765	1.62268	0.18497	0.10233
1.20000	2.12012	1.88095	0.23916	0.11281
1.35000	2.50743	2.20352	0.30390	0.12120

1.50000	2.98169	2.60079	0.38090	0.12774
1.65000	3.55698	3.08486	0.47212	0.13273
1.80000	4.24965	3.66976	0.57989	0.13646
1.95000	5.07869	4.37182	0.70687	0.13918
2.10000	6.06617	5.20999	0.85618	0.14114
2.25000	7.23774	6.20631	1.03142	0.14251
2.40000	8.62318	7.38637	1.23680	0.14343
2.55000	10.25710	8.77991	1.47720	0.14402
2.70000	12.17973	10.42146	1.75827	0.14436
2.85000	14.43778	12.35118	2.08661	0.14452
3.00000	17.08554	14.61568	2.46985	0.14456



3.

$$y'' = e^x - y' - y, \quad 0 \leq t \leq 10, \quad y(0) = 1, \quad y'(0) = 1, \quad N = 20,$$

cuya solución analítica es

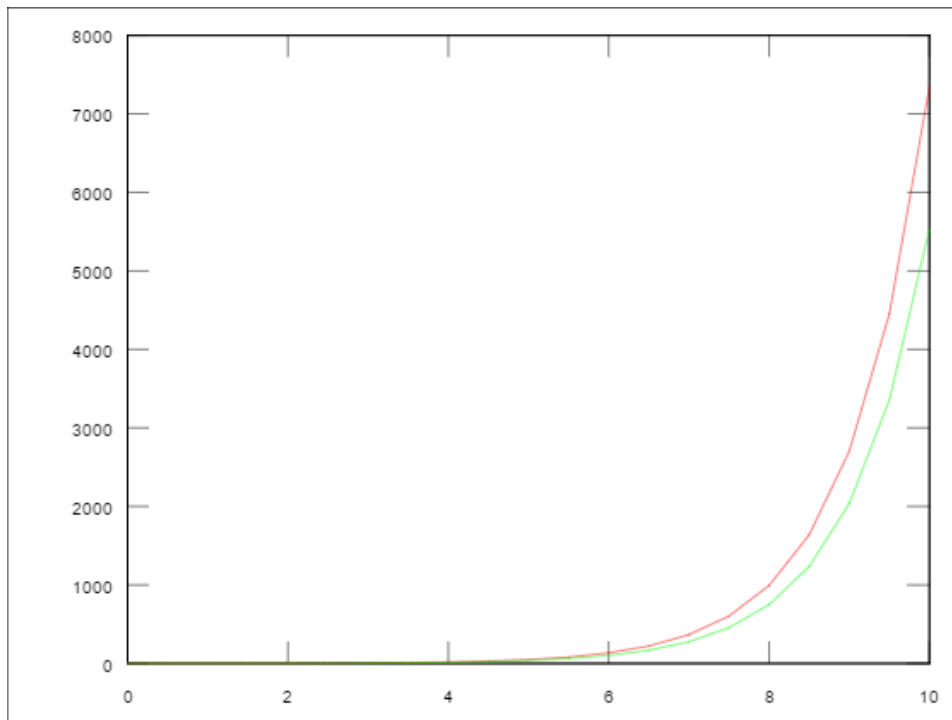
$$y = e^{\frac{-1}{2}x} \left(\frac{2}{3} \cos(\sqrt{3}x) + \frac{1}{\sqrt{3}} \sin(\sqrt{3}x) \right) + \frac{1}{3} e^x.$$

Tabla de resultados:

```
octave:1> Euler_Leal2(0, 10, 1, 1, 20)
```

t	Sol. analt.	Sol. Numr.	Error	Error relativo
0.0000e+00	1.0000e+00	1.0000e+00	0.0000e+00	0.0000e+00
5.0000e-01	1.2285e+00	1.5000e+00	-2.7154e-01	-2.2104e-01
1.0000e+00	1.1868e+00	1.7500e+00	-5.6319e-01	-4.7454e-01
1.5000e+00	1.3654e+00	1.9122e+00	-5.4678e-01	-4.0045e-01
2.0000e+00	2.1631e+00	2.2353e+00	-7.2249e-02	-3.3401e-02
2.5000e+00	3.8361e+00	3.0393e+00	7.9681e-01	2.0771e-01
3.0000e+00	6.6503e+00	4.7297e+00	1.9206e+00	2.8880e-01
3.5000e+00	1.1130e+01	7.8607e+00	3.2688e+00	2.9371e-01
4.0000e+00	1.8318e+01	1.3265e+01	5.0533e+00	2.7586e-01

4.5000e+00	3.0071e+01	2.2281e+01	7.7896e+00	2.5904e-01
5.0000e+00	4.9464e+01	3.7122e+01	1.2342e+01	2.4951e-01
5.5000e+00	8.1518e+01	6.1477e+01	2.0041e+01	2.4585e-01
6.0000e+00	1.3443e+02	1.0148e+02	3.2957e+01	2.4515e-01
6.5000e+00	2.2170e+02	1.6728e+02	5.4418e+01	2.4546e-01
7.0000e+00	3.6556e+02	2.7567e+02	8.9885e+01	2.4589e-01
7.5000e+00	6.0270e+02	4.5433e+02	1.4837e+02	2.4618e-01
8.0000e+00	9.9367e+02	7.4890e+02	2.4476e+02	2.4632e-01
8.5000e+00	1.6383e+03	1.2346e+03	4.0364e+02	2.4639e-01
9.0000e+00	2.7010e+03	2.0355e+03	6.6554e+02	2.4640e-01



Y para ecuaciones de orden tres tendremos:

1.

$$y''' = 10\sin(x) + y', \quad 0 \leq t \leq 10, \quad y(0) = 0, \quad y'(0) = 0, \quad y''(0) = 0 \quad N = 50,$$

con solución

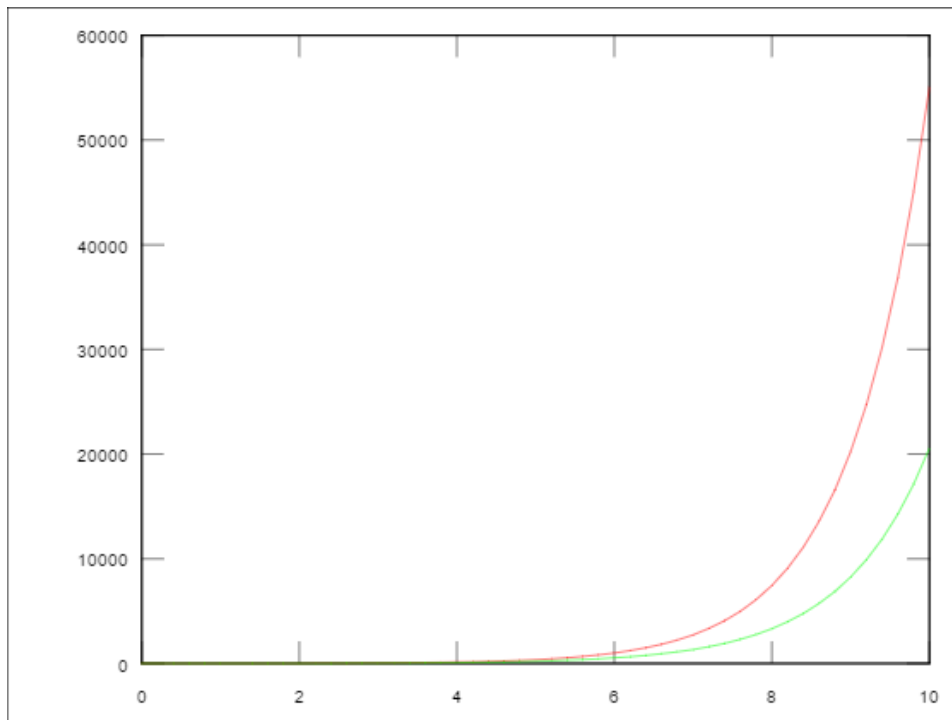
$$y = -10 + \frac{5}{2}e^{-x} + \frac{5}{2}e^x + 5\cos(x).$$

Tabla de resultados:

```
octave:1> Euler_Leal3(0, 10, 0, 0, 0, 50)
```

t	Sol. analt.	Sol. Numr.	Error	Error relativo
0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00
2.0000e-01	6.6667e-04	0.0000e+00	6.6667e-04	1.0000e+00
4.0000e-01	1.0667e-02	0.0000e+00	1.0667e-02	1.0000e+00
6.0000e-01	5.4004e-02	0.0000e+00	5.4004e-02	1.0000e+00
8.0000e-01	1.7071e-01	1.5894e-02	1.5481e-01	9.0690e-01
1.0000e+00	4.1691e-01	7.8834e-02	3.3808e-01	8.1091e-01
1.2000e+00	8.6507e-01	2.3463e-01	6.3044e-01	7.2877e-01
1.4000e+00	1.6043e+00	5.4318e-01	1.0611e+00	6.6143e-01

1.6000e+00	2.7413e+00	1.0780e+00	1.6633e+00	6.0674e-01
1.8000e+00	4.4014e+00	1.9261e+00	2.4752e+00	5.6238e-01
2.0000e+00	6.7302e+00	3.1877e+00	3.5426e+00	5.2637e-01
2.2000e+00	9.8970e+00	4.9765e+00	4.9205e+00	4.9717e-01
2.4000e+00	1.4098e+01	7.4211e+00	6.6767e+00	4.7360e-01
2.6000e+00	1.9561e+01	1.0666e+01	8.8949e+00	4.5474e-01
2.8000e+00	2.6553e+01	1.4873e+01	1.1680e+01	4.3988e-01
3.0000e+00	3.5388e+01	2.0226e+01	1.5162e+01	4.2846e-01
3.2000e+00	4.6442e+01	2.6935e+01	1.9507e+01	4.2002e-01
3.4000e+00	6.0160e+01	3.5241e+01	2.4919e+01	4.1421e-01
3.6000e+00	7.7080e+01	4.5423e+01	3.1657e+01	4.1070e-01
3.8000e+00	9.7854e+01	5.7809e+01	4.0045e+01	4.0923e-01
		.		
		.		
		.		
7.0000e+00	2.7354e+03	1.3303e+03	1.4050e+03	5.1365e-01
7.2000e+00	3.3416e+03	1.5969e+03	1.7447e+03	5.2211e-01
7.4000e+00	4.0822e+03	1.9168e+03	2.1653e+03	5.3044e-01
7.6000e+00	4.9867e+03	2.3007e+03	2.6860e+03	5.3863e-01
7.8000e+00	6.0918e+03	2.7615e+03	3.3303e+03	5.4669e-01
8.0000e+00	7.4417e+03	3.3145e+03	4.1272e+03	5.5460e-01
8.2000e+00	9.0907e+03	3.9783e+03	5.1123e+03	5.6237e-01
8.4000e+00	1.1105e+04	4.7752e+03	6.3299e+03	5.7000e-01
8.6000e+00	1.3566e+04	5.7316e+03	7.8341e+03	5.7749e-01
8.8000e+00	1.6572e+04	6.8797e+03	9.6919e+03	5.8485e-01
9.0000e+00	2.0243e+04	8.2576e+03	1.1986e+04	5.9208e-01
9.2000e+00	2.4728e+04	9.9115e+03	1.4816e+04	5.9918e-01
9.4000e+00	3.0206e+04	1.1896e+04	1.8310e+04	6.0616e-01
9.6000e+00	3.6897e+04	1.4278e+04	2.2619e+04	6.1302e-01
9.8000e+00	4.5070e+04	1.7137e+04	2.7932e+04	6.1976e-01
1.0000e+01	5.5052e+04	2.0568e+04	3.4484e+04	6.2639e-01



Mas si ahora resolvemos la misma ecuación pero con 250 iteraciones tendremos una precisión mayor, tal como se muestra en seguida y como era de esperarse:

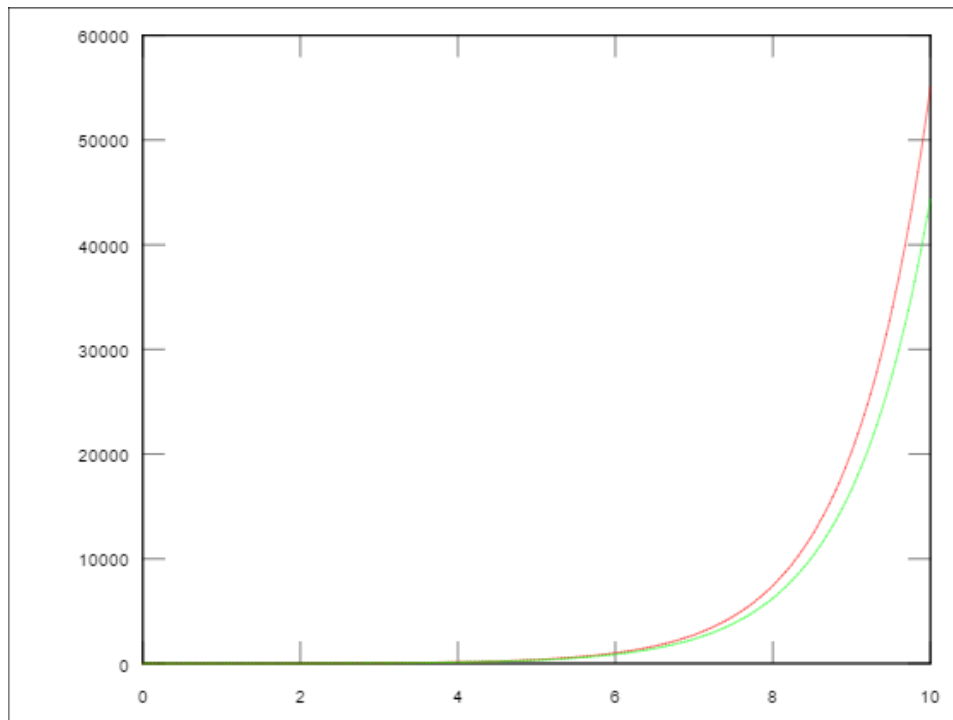
```

octave:1> Euler_Lea13(0, 10, 0, 0, 0, 250)
t          x          y          Error          Error relat.
ans =

0.0000e+00  0.0000e+00  0.0000e+00  0.0000e+00  0.0000e+00
4.0000e-02  1.0667e-06  0.0000e+00  1.0667e-06  1.0000e+00
8.0000e-02  1.7067e-05  0.0000e+00  1.7067e-05  1.0000e+00
1.2000e-01  8.6400e-05  0.0000e+00  8.6400e-05  1.0000e+00
1.6000e-01  2.7307e-04  2.5593e-05  2.4747e-04  9.0628e-01
2.0000e-01  6.6667e-04  1.2792e-04  5.3874e-04  8.0811e-01
2.4000e-01  1.3824e-03  3.8365e-04  9.9875e-04  7.2247e-01
2.8000e-01  2.5611e-03  8.9490e-04  1.6662e-03  6.5058e-01
3.2000e-01  4.3691e-03  1.7892e-03  2.5799e-03  5.9048e-01
3.6000e-01  6.9985e-03  3.2196e-03  3.7789e-03  5.3996e-01
4.0000e-01  1.0667e-02  5.3643e-03  5.3026e-03  4.9711e-01
4.4000e-01  1.5617e-02  8.4269e-03  7.1905e-03  4.6042e-01
4.8000e-01  2.2119e-02  1.2636e-02  9.4828e-03  4.2871e-01
5.2000e-01  3.0466e-02  1.8247e-02  1.2220e-02  4.0109e-01
5.6000e-01  4.0979e-02  2.5537e-02  1.5442e-02  3.7682e-01

```

6.0000e-01	5.4004e-02	3.4813e-02	1.9191e-02	3.5537e-01
6.4000e-01	6.9912e-02	4.6403e-02	2.3509e-02	3.3627e-01
6.8000e-01	8.9100e-02	6.0662e-02	2.8438e-02	3.1917e-01
7.2000e-01	1.1199e-01	7.7971e-02	3.4022e-02	3.0379e-01
7.6000e-01	1.3904e-01	9.8733e-02	4.0303e-02	2.8988e-01
8.0000e-01	1.7071e-01	1.2338e-01	4.7328e-02	2.7725e-01
8.4000e-01	2.0751e-01	1.5237e-01	5.5141e-02	2.6573e-01
8.8000e-01	2.4996e-01	1.8617e-01	6.3789e-02	2.5520e-01
9.2000e-01	2.9862e-01	2.2530e-01	7.3319e-02	2.4552e-01
9.6000e-01	3.5407e-01	2.7029e-01	8.3781e-02	2.3662e-01
1.0000e+00	4.1691e-01	3.2169e-01	9.5222e-02	2.2840e-01
		.		
		.		
		.		
9.0000e+00	2.0243e+04	1.6649e+04	3.5946e+03	1.7757e-01
9.0400e+00	2.1070e+04	1.7315e+04	3.7548e+03	1.7821e-01
9.0800e+00	2.1930e+04	1.8008e+04	3.9221e+03	1.7884e-01
9.1200e+00	2.2826e+04	1.8729e+04	4.0968e+03	1.7948e-01
9.1600e+00	2.3758e+04	1.9479e+04	4.2792e+03	1.8012e-01
9.2000e+00	2.4728e+04	2.0258e+04	4.4696e+03	1.8075e-01
9.2400e+00	2.5738e+04	2.1069e+04	4.6685e+03	1.8139e-01
9.2800e+00	2.6789e+04	2.1913e+04	4.8761e+03	1.8202e-01
9.3200e+00	2.7882e+04	2.2790e+04	5.0929e+03	1.8266e-01
9.3600e+00	2.9021e+04	2.3702e+04	5.3192e+03	1.8329e-01
9.4000e+00	3.0206e+04	2.4650e+04	5.5555e+03	1.8392e-01
9.4400e+00	3.1439e+04	2.5637e+04	5.8023e+03	1.8455e-01
9.4800e+00	3.2723e+04	2.6663e+04	6.0599e+03	1.8519e-01
9.5200e+00	3.4059e+04	2.7730e+04	6.3288e+03	1.8582e-01
9.5600e+00	3.5450e+04	2.8840e+04	6.6096e+03	1.8645e-01
9.6000e+00	3.6897e+04	2.9994e+04	6.9027e+03	1.8708e-01
9.6400e+00	3.8403e+04	3.1195e+04	7.2088e+03	1.8771e-01
9.6800e+00	3.9971e+04	3.2443e+04	7.5283e+03	1.8834e-01
9.7200e+00	4.1603e+04	3.3741e+04	7.8619e+03	1.8897e-01
9.7600e+00	4.3302e+04	3.5092e+04	8.2101e+03	1.8960e-01
9.8000e+00	4.5070e+04	3.6496e+04	8.5736e+03	1.9023e-01
9.8400e+00	4.6910e+04	3.7957e+04	8.9531e+03	1.9086e-01
9.8800e+00	4.8825e+04	3.9476e+04	9.3493e+03	1.9149e-01
9.9200e+00	5.0818e+04	4.1055e+04	9.7629e+03	1.9211e-01
9.9600e+00	5.2893e+04	4.2698e+04	1.0195e+04	1.9274e-01
1.0000e+01	5.5052e+04	4.4407e+04	1.0645e+04	1.9337e-01



En conclusión, el método de Euler-Leal es muy impreciso, característica que hereda del método de Euler para ecuaciones de orden uno; pero al aumentar el número de iteraciones aumenta la precisión, como debe ser.

2.2.2. Análisis del orden de convergencia.

Ahora queremos medir la precisión con la que la ecuación en diferencias se aproxima a la ecuación diferencial, es decir, el error de truncamiento local, el cual está definido como

$$L(t, y; h) = y(t, h) - [y + h \cdot y'(t, y)]$$

donde $y(t + h)$ es la solución de la ecuación que pasa por el punto (t, y) .

Si $y \in C^2$, podemos desarrollar $y(t + h)$ de la forma:

$$y(t + h) = y(t) + hy'(t) + (h^2/2)y''(\xi).$$

Si a esto le restamos $y + hy'(t)$, entonces

$$L = (h^2/2)y''(\xi),$$

lo cual está acotado al ser y'' continua, ya que es de clase C^2 . Así pues, el método es consistente, ya que $L \rightarrow 0$ cuando $h \rightarrow 0$, y nuestro grado de error será del orden de $O(h^2)$.

3. Método de Heun-Leal o de Heun generalizado

3.1. Algoritmo de Heun-Leal

Análogamente al Método de Euler Generalizado, el Método de Heun-Leal o de Heun Generalizado [1,2], para ecuaciones de orden mayor a uno poseerá un núcleo de la forma:

```

desde (n = 0, n < Nmero de particiones, n incrementa de 1 en 1)
  y_{n+1} = y_{n} + y'_{n}*h
  y'_{n+1} = y'_{n} + y''_{n}*h
  .
  .
  .
  y^{k-1}_{n+1} = y^{k-1}_{n} +
  + f( t_{n}, y_{n}, y'_{n}, ... , y^{k-1}_{n} ) * h

  y_{n+1} = y_{n} + (y'_{n} + y'_{n+1})*h/2
  y'_{n+1} = y'_{n} + (y''_{n} + y''_{n+1})*h/2
  .
  .
  .
  y^{k-1}_{n+1} = y^{k-1}_{n} +
  + ( f( t_{n}, y_{n}, y'_{n}, ... , y^{k-1}_{n} ) +
      f( t_{n+1}, y_{n+1}, y'_{n+1}, ... , y^{k-1}_{n+1} ) ) * h/2
fin_desde

```

3.2. Análisis de convergencia

3.2.1. Análisis gráfico

Mostramos ahora unos gráficos del método puesto a prueba.

Ecuaciones de orden 2:

1. Primero volveremos a usar la tercera ecuación usada en el método de Euler-Leal pero ahora con este método, para así poder hacer la comparación directa entre ambos métodos. Así pues, tenemos

$$y'' = e^x - y' - y, \quad 0 \leq t \leq 10, \quad y(0) = 1, \quad y'(0) = 1, \quad N = 20,$$

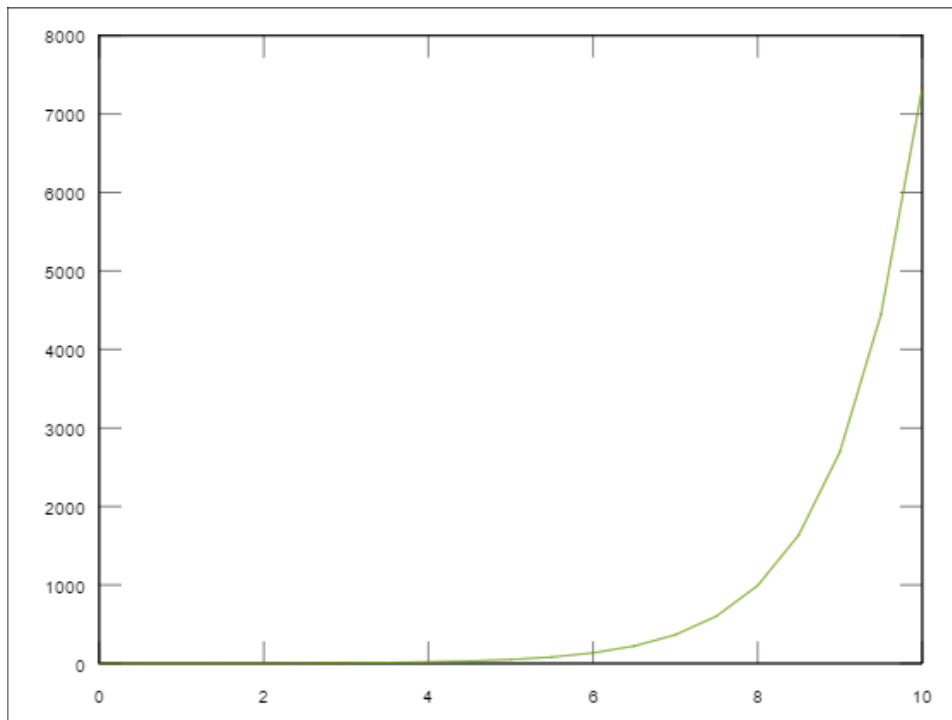
cuya solución analítica es

$$y = e^{-\frac{1}{2}x} \left(\frac{2}{3} \cos \sqrt{3}x + \frac{1}{\sqrt{3}} \sin \sqrt{3}x \right) + \frac{1}{3} e^x.$$

Tabla de resultados:

```
octave:1> Heun_Leal2(0, 10, 1, 1, 20)
```

t	Sol. anal.	Sol. num.	Error	Error relativo
0.0000e+00	1.0000e+00	1.0000e+00	0.0000e+00	0.0000e+00
5.0000e-01	1.2285e+00	1.3750e+00	-1.4654e-01	-1.1929e-01
1.0000e+00	1.1868e+00	1.6693e+00	-4.8244e-01	-4.0650e-01
1.5000e+00	1.3654e+00	2.0741e+00	-7.0868e-01	-5.1902e-01
2.0000e+00	2.1631e+00	2.8210e+00	-6.5789e-01	-3.0414e-01
2.5000e+00	3.8361e+00	4.2118e+00	-3.7571e-01	-9.7941e-02
3.0000e+00	6.6503e+00	6.6819e+00	-3.1605e-02	-4.7525e-03
3.5000e+00	1.1130e+01	1.0909e+01	2.2082e-01	1.9841e-02
4.0000e+00	1.8318e+01	1.7989e+01	3.2916e-01	1.7969e-02
4.5000e+00	3.0071e+01	2.9729e+01	3.4175e-01	1.1365e-02
5.0000e+00	4.9464e+01	4.9110e+01	3.5419e-01	7.1605e-03
5.5000e+00	8.1518e+01	8.1062e+01	4.5585e-01	5.5920e-03
6.0000e+00	1.3443e+02	1.3372e+02	7.1106e-01	5.2893e-03
6.5000e+00	2.2170e+02	2.2052e+02	1.1785e+00	5.3160e-03
7.0000e+00	3.6556e+02	3.6360e+02	1.9536e+00	5.3441e-03
7.5000e+00	6.0270e+02	5.9948e+02	3.2171e+00	5.3379e-03
8.0000e+00	9.9367e+02	9.8837e+02	5.2914e+00	5.3252e-03
8.5000e+00	1.6383e+03	1.6295e+03	8.7160e+00	5.3203e-03
9.0000e+00	2.7010e+03	2.6866e+03	1.4375e+01	5.3220e-03
9.5000e+00	4.4532e+03	4.4295e+03	2.3715e+01	5.3254e-03
1.0000e+01	7.3422e+03	7.3030e+03	3.9118e+01	5.3278e-03



Así, en este ejemplo vemos que el método de Heun-Leal converge más rápido que el Euler-Leal, pues la solución analítica (roja) y la numérica (verde) son prácticamente iguales

Ecuaciones de orden 3:

1. Ahora volveremos a tomar la ecuación de tercer orden usada en el algoritmo de Euler-Leal

$$y''' = 10\sin(t)+y', \quad 0 \leq t \leq 10, \quad y(0) = 0, \quad y'(0) = 0, \quad y''(0) = 0, \quad N = 50,$$

cuya solución analítica estaba dada por

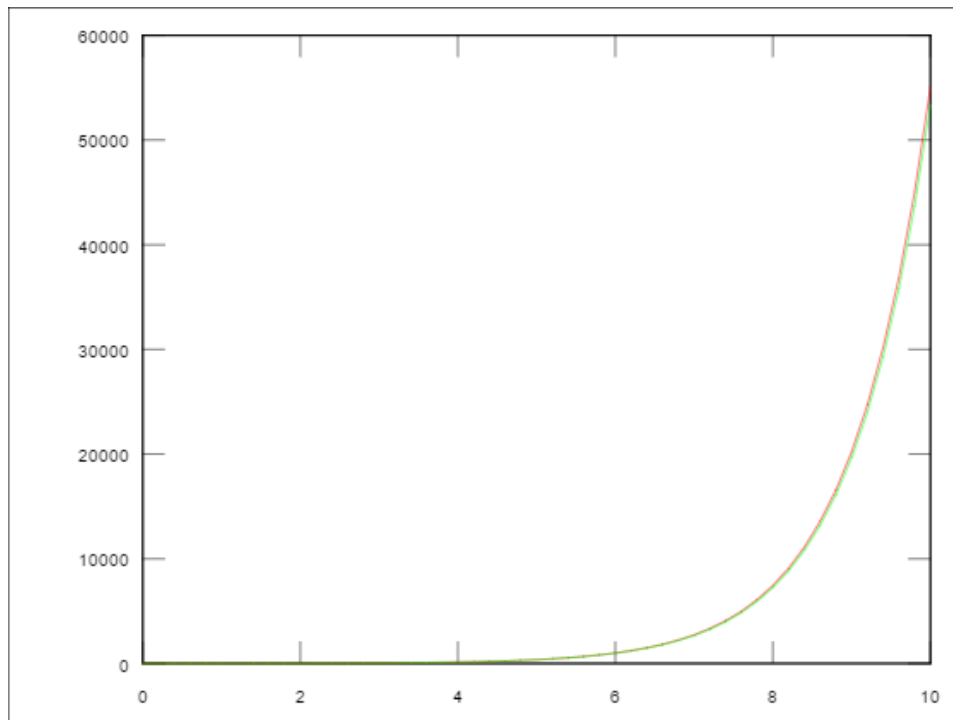
$$y = -10 + \frac{5}{2}e^{-x} + \frac{5}{2}e^x + 5\cos(x).$$

Tabla de resultados:

```
octave:1> Heun_Leal3(0, 10, 0, 0, 0, 50)
```

t	Sol. Analt.	Sol. numr.	Error	Error relativo
0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00
2.0000e-01	6.6667e-04	0.0000e+00	6.6667e-04	1.0000e+00

4.0000e-01	1.0667e-02	3.9734e-03	6.6934e-03	6.2750e-01
6.0000e-01	5.4004e-02	3.5761e-02	1.8243e-02	3.3781e-01
8.0000e-01	1.7071e-01	1.3511e-01	3.5601e-02	2.0855e-01
1.0000e+00	4.1691e-01	3.5772e-01	5.9198e-02	1.4199e-01
1.2000e+00	8.6507e-01	7.7542e-01	8.9649e-02	1.0363e-01
1.4000e+00	1.6043e+00	1.4765e+00	1.2780e-01	7.9658e-02
1.6000e+00	2.7413e+00	2.5665e+00	1.7478e-01	6.3759e-02
1.8000e+00	4.4014e+00	4.1692e+00	2.3212e-01	5.2738e-02
2.0000e+00	6.7302e+00	6.4285e+00	3.0178e-01	4.4840e-02
2.2000e+00	9.8970e+00	9.5107e+00	3.8635e-01	3.9037e-02
2.4000e+00	1.498e+01	1.3609e+01	4.8913e-01	3.4696e-02
2.6000e+00	1.9561e+01	1.8946e+01	6.1436e-01	3.1408e-02
2.8000e+00	2.6553e+01	2.5785e+01	7.6738e-01	2.8901e-02
		.		
		.		
		.		
7.0000e+00	2.7354e+03	2.6659e+03	6.9413e+01	2.5376e-02
7.2000e+00	3.3416e+03	3.2557e+03	8.5884e+01	2.5701e-02
7.4000e+00	4.0822e+03	3.9759e+03	1.0624e+02	2.6026e-02
7.6000e+00	4.9867e+03	4.8553e+03	1.3140e+02	2.6350e-02
7.8000e+00	6.0918e+03	5.9293e+03	1.6248e+02	2.6672e-02
8.0000e+00	7.4417e+03	7.2408e+03	2.0088e+02	2.6994e-02
8.2000e+00	9.0907e+03	8.8424e+03	2.4831e+02	2.7314e-02
8.4000e+00	1.1105e+04	1.0798e+04	3.0688e+02	2.7635e-02
8.6000e+00	1.3566e+04	1.3187e+04	3.7922e+02	2.7954e-02
8.8000e+00	1.6572e+04	1.6103e+04	4.6854e+02	2.8274e-02
9.0000e+00	2.0243e+04	1.9664e+04	5.7881e+02	2.8593e-02
9.2000e+00	2.4728e+04	2.4013e+04	7.1494e+02	2.8912e-02
9.4000e+00	3.0206e+04	2.9323e+04	8.8297e+02	2.9232e-02
9.6000e+00	3.6897e+04	3.5807e+04	1.0903e+03	2.9551e-02
9.8000e+00	4.5070e+04	4.3723e+04	1.3463e+03	2.9871e-02
1.0000e+01	5.5052e+04	5.3390e+04	1.6621e+03	3.0191e-02



Aquí observamos nuevamente que la solución analítica y la numérica son prácticamente iguales, a diferencia del algoritmo de Euler-Leal, donde hay gran divergencia.

Por lo tanto concluimos que este método hereda la mayor precisión del método de Heun respecto al método de Euler, como era de esperarse.

3.2.2. Análisis del orden de convergencia

Evidentemente éste método posee un orden del error de $O(h^2)$, al igual que el método de Euler-Leal, pero, debido a la corrección que hace del cálculo de y y sus derivadas, resulta ser mucho más preciso.

4. Métodos de Taylor-Leal de orden mayor

4.1. Algoritmo de Taylor-Leal

Hasta ahora solo hemos utilizado métodos modificados de orden $O(h^2)$ en el error, sin embargo también podemos modificar los métodos de Taylor de orden mayor en la precisión [1] y aplicarlos a ecuaciones diferenciales de cualquier orden en la derivada mayor. El núcleo del algoritmo será expuesto para dos casos particulares, pudiendo generalizarse fácilmente para los demás casos:

Caso Particular. Método de Taylor-Leal de orden m en el error para ecuaciones de orden n en la máxima derivada, cuando $m > n$.

```

desde i = 0 hasta N-1, incrementando de uno en uno
  t_{i+1} = t_{i} + dt
  y_{i+1} = y_{i} + y'_{i}*dt + (1/2)*y''_{i}*dt^2
            + ...
            + [1/(n)!]*f(t_{i}, y_{i}, ..., y^{n-1}_{i})*dt^n
            + [1/(n+1)!]*f'(t_{i}, y_{i}, ..., y^{n-1}_{i})*dt^{n+1}
            + ...
            + [1/(m-1)!]*f^{m-n-1}(t_{i},y_{i},...,y^{n-1}_{i})*dt^{m-1}
            .
            .
            .
  y^{n-1}_{i+1} = y^{n-1}_{i} + (1/2)*f(t_{i}, y_{i}, ..., y^{n-1}_{i})*dt^2
                + ...
                + [1/(m-1)!]*f^{m-2}(t_{i},y_{i},...,y^{n-1}_{i})*dt^{m-1}
fin_desde

```

Caso 2. Método de Taylor-Leal de orden n en el error para ecuaciones de orden n en la máxima derivada.

```

desde i = 0 hasta N-1, incrementando de uno en uno
  t_{i+1} = t_{i} + dt
  y_{i+1} = y_{i} + y'_{i}*dt + (1/2)*y''_{i}*dt^2
            + ... + [1/(n-1)!]*y^{n-1}_{i}*dt^{n-1}
  y'_{i+1} = y'_{i} + y''_{i}*dt + (1/2)*y'''_{i}*dt^2
            + ...
            + [1/(n-1)!]*f(t_{i},y_{i},...,y^{n-1}_{i})*dt^{n-1}
            .
            .
            .
  y^{n-1}_{i+1} = y^{n-1}_{i} + (1/2)*f(t_{i},y_{i},...,y^{n-1}_{i})*dt^2
                + ...
                + [1/(n-1)!]*f^{n-2}(t_{i},y_{i},...,y^{n-1}_{i})*dt^{n-1}
fin_desde

```

4.2. Análisis de convergencia

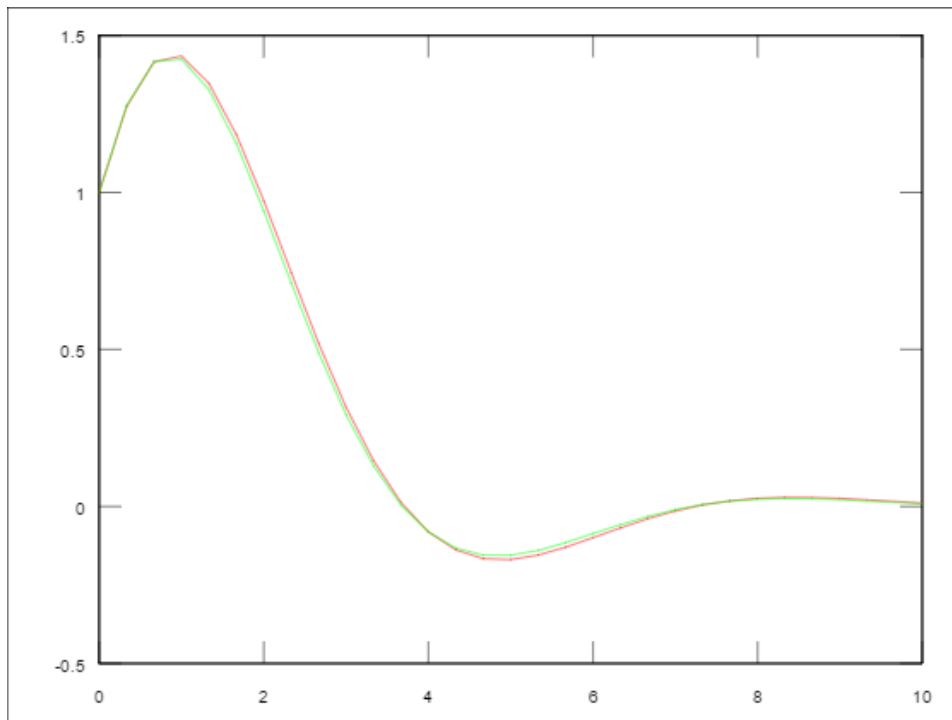
4.2.1. Análisis gráfico

Ponemos a prueba el método con una ecuación diferencial de orden 2 y un orden de error de $O(h^3)$. La ecuación diferencial es $y'' = e^{-x} - y - y'$, con solución analítica $y = \frac{4}{\sqrt{3}}e^{-\frac{1}{2}t}\sin\left(\frac{\sqrt{3}}{2}t\right) + e^{-t}$.

Tabla de resultados:

```
octave:1> Taylor_Leal2_3(0, 10, 1, 1, 30)
```

t	Sol. Anal.	Sol. num.	Error	Error relativo
0.00000	1.00000	1.00000	0.00000	0.00000
0.33333	1.27305	1.27778	-0.00473	-0.00372
0.66667	1.41659	1.41635	0.00024	0.00017
1.00000	1.43489	1.42499	0.00990	0.00690
1.33333	1.34811	1.32787	0.02024	0.01501
1.66667	1.18440	1.15591	0.02849	0.02406
2.00000	0.97389	0.94077	0.03312	0.03401
2.33333	0.74455	0.71088	0.03367	0.04522
2.66667	0.51960	0.48911	0.03048	0.05867
3.00000	0.31627	0.29183	0.02445	0.07729
3.33333	0.14563	0.12895	0.01669	0.11459
3.66667	0.01307	0.00470	0.00838	0.64071
4.00000	-0.08074	-0.08127	0.00053	-0.00651
4.33333	-0.13869	-0.13261	-0.00609	0.04388
4.66667	-0.16600	-0.15502	-0.01098	0.06615
5.00000	-0.16915	-0.15518	-0.01396	0.08255
5.33333	-0.15494	-0.13985	-0.01509	0.09739
5.66667	-0.12979	-0.11519	-0.01461	0.11253
6.00000	-0.09931	-0.08643	-0.01288	0.12966
6.33333	-0.06793	-0.05761	-0.01032	0.15192
6.66667	-0.03892	-0.03157	-0.00735	0.18893
7.00000	-0.01438	-0.01003	-0.00435	0.30238
7.33333	0.00464	0.00624	-0.00159	-0.34295
7.66667	0.01790	0.01720	0.00070	0.03928
8.00000	0.02577	0.02334	0.00243	0.09417
8.33333	0.02902	0.02548	0.00354	0.12209
8.66667	0.02866	0.02457	0.00409	0.14256
9.00000	0.02573	0.02160	0.00413	0.16067
9.33333	0.02124	0.01744	0.00380	0.17884
9.66667	0.01604	0.01284	0.00320	0.19937
10.00000	0.01082	0.00837	0.00245	0.22647



En conclusión el método funciona muy bien, como era de suponerse para un algoritmo de orden $O(h^3)$ en el tamaño del error.

4.2.2. Análisis del orden de convergencia.

Del algoritmo del método de Taylor de orden mayor se puede ver directamente que el método de Taylor-Leal de orden n tendrá efectivamente un error del orden de $O(h^{n+1})$.

5. Conclusiones

En conclusión podemos decir que mis métodos desarrollados para aproximación de soluciones de ecuaciones diferenciales ordinarias de orden mayor a uno con condiciones iniciales sí funcionan, de acuerdo a la teoría, siendo además que entre más pequeño sea el intervalo de discretización, mayor será la precisión de dichos métodos, como también era de esperarse. Sin embargo, el método de Euler-Leal es el más impreciso (característica que hereda del método de Euler), por lo cual dicho método no es muy bueno en la práctica, no obstante que sí es muy didáctico cuando uno comienza a aprender estos nuevos métodos (al igual que el método de Euler). Y en general, considero que los demás métodos sí son satisfactorios.

También debo señalar que en este artículo solo presenté el análisis de

algunos ejemplos, en aras de la brevedad, siendo que en mis archivos personales he desarrollado meticulosamente muchos más ejemplos, con lo cual pude establecer con más seguridad y fundamento las conclusiones de los análisis de convergencia de este documento.

Por último debo decir que actualmente estoy desarrollando aún más este trabajo, mejorando algunos de mis algoritmos (como el algoritmo de Euler, el cual estoy mejorándolo bastante para hacerlo más aplicable a la realidad) e inventando (o descubriendo) nuevos algoritmos, por lo que de este trabajo inicial se pueden desarrollar muchas más cosas.

Referencias

- [1] Análisis Numérico. Burden, Faires. Grupo Editorial Iberoamérica.
. México, D.F. 1996.
- [2] Elementos de métodos numéricos para ingeniería. Gómez J., Escobar,
. Gómez A., Guerrero, Gutiérrez, Hernández, Puerto, Sandoval.
. Mc Graw Hill. México, D.F. 2002.