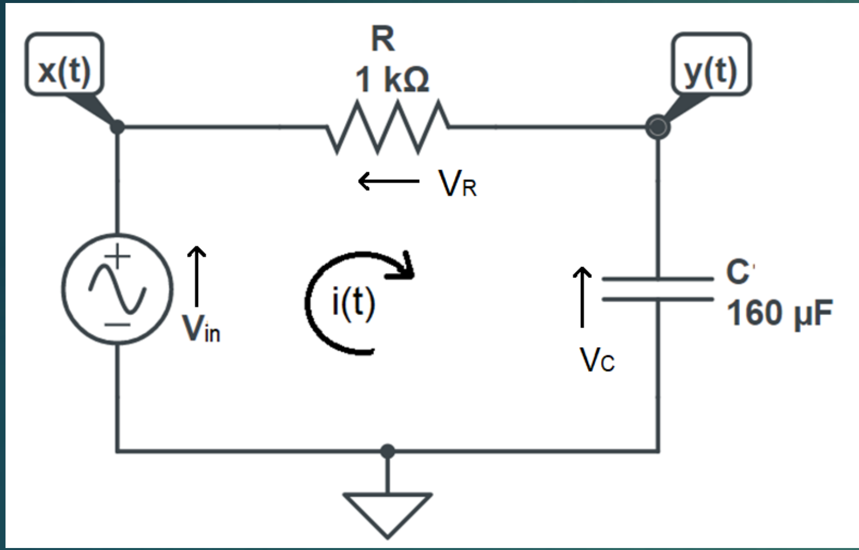
Métodos Numéricos para Engenharia

MÓDULO 15 – EQUAÇÕES DIFERENCIAIS ORDINÁRIAS – PROBLEMAS DE VALOR INICIAL

PROFESSOR LUCIANO NEVES DA FONSECA

Exemplo EDO de 1ª Ordem

Como exemplo, seja o sistema eletrônico RC mostrado abaixo no qual se quer encontrar a tensão no capacitor $v_c(t)$ (saída do do sistema) , resultante da tensão na fonte $v_{in}(t)$ (entrada do sistema)



$R = 1\ k\Omega$
$C = 500\ \mu F$
Condições iniciais $v_C(0^-) = 0$

$Malha \quad v_R(t)+v_C(t) = v_{in}(t)$
$v_R(t) = R\ i(t)$

$v_{in} = R\ i(\mathrm{t}) + v_C(t)$

Mas no capacitor $i(t) = C\ v_c{}'(t)$

Então :

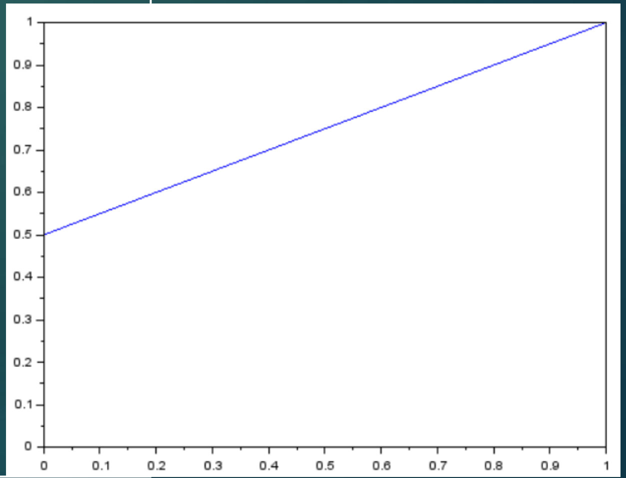
$$R\ Cv_c'(t)\ + v_C(t) = \mathrm{v_{in}}(\mathrm{t})$$
$$v_c'(t) + \frac{1}{\mathrm{RC}} v_c(t) = \frac{1}{RC} v_{in}(t)$$

Como : $\frac{1}{RC} = 2$

$$v_c'(t) + 2v_C(t) = 2v_{in}\ (t)$$
$$x(t) = v_{in}(t)$$
$$y(t) = v_c(t)$$
$$y'(t) + 2y(t) = 2x(t)$$

Para a entrada: $x(t) = 0.5(t+1)\ \ 0 \le t \le 1$

```
--> deff('x=fx(t)','x=0.5*(t+1)')

--> t=linspace(0,1,1000);

--> plot(t,fx(t));
```



$$y'(t) + 2y(t) = t + 1$$

EDO de primeira ordem – Problemas de Valor Inicial

Seja a seguinte Equações Diferenciais Ordinária, definida no domínio $[a, b]$, com condição inicial $y_0 = f(a)$, na qual seja possível se isolar o termo em y' (forma canônica):

$$y' = g(t, y)$$

A solução desta equação diferencial é a função $y = f(t)$

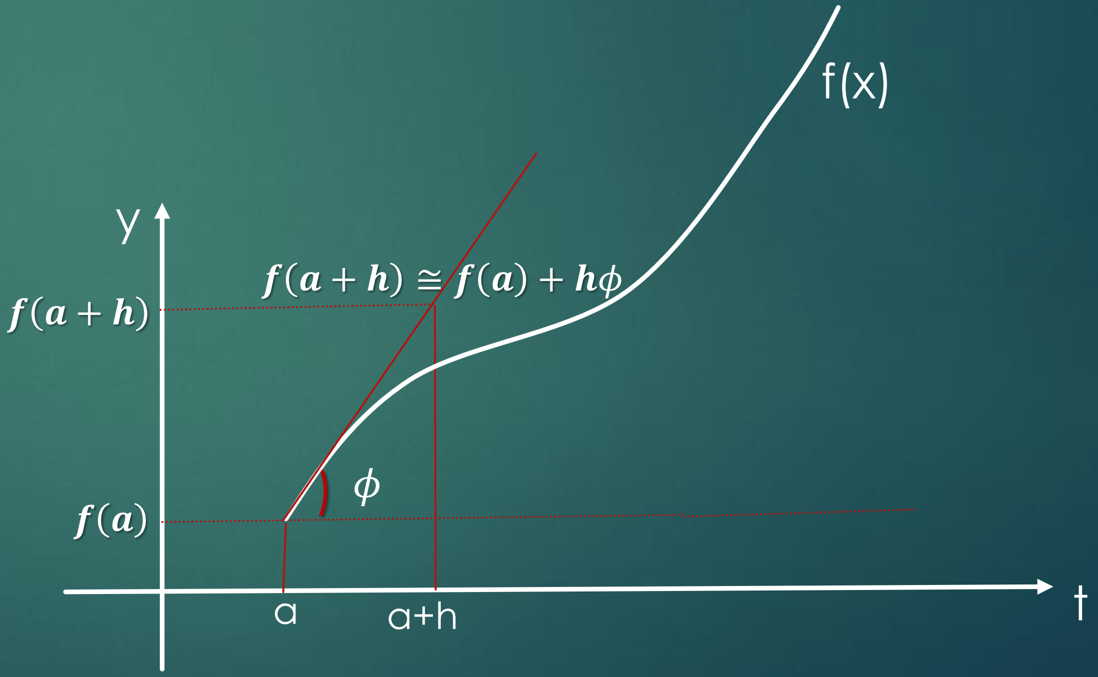
Se soubermos o valor de da função $f(t)$ no ponto inicial $(a, f(a))$, e uma estimativa ϕ da inclinação da curva neste ponto, podemos estimar o valor de $f(a+h)$, com h pequeno, através da seguinte expressão:

$$\boldsymbol{f(a+h) \cong f(a) + h\phi}$$

Em um malha de n soluções equiespaçadas, com espaçamento $h = \frac{b-a}{n-1}$, no domínio [a,b], teremos a seguinte equação recursiva

$$t_i = a + ih, \qquad 0 \leq i \leq n-1$$

$$y_{i+1} = y_i + h\phi_i$$



Os diferentes métodos que serão estudados apresentam soluções alternativas para o cálculo da inclinação ϕ_i

Método de Euler

Seja a seguinte Equações Diferenciais Ordinária, definida no domínio $[a, b]$, com condição inicial $\mathrm{y_o} = f(a)$

$$y' = g(t, y)$$

No método de Euler, a inclinação ϕ_i é aproximada diretamente pela função g(x,y) no início do intervalo :

$$\boxed{\begin{aligned} k_1 &= g(t_i, y_i) \\ \phi_i &= k_1 \end{aligned}}$$

Em um malha de n soluções equiespaçadas, com espaçamento h, no domínio [a,b], teremos a seguinte equação recursiva

$$t_i = [a, a+h, a+2h, a+3h, \dots., a+(n-2)h, b] \qquad onde, \qquad h = \frac{|b-a|}{n-1}$$

$$y_{i+1} = y_i + h\phi_i$$

Método de Euler

Seja a seguinte EDO de 1ª ordem

$y'(t) + 2y(t) = t + 1 \qquad (i)$

$0 \leq t \leq 1.2$

Condição inicial $y(0) = 1$ n=7

Primeiro transformamos (i) para a forma canônica:

$y' = g(t, y) = -2y + t + 1$

Em Euler:

$k_1 = g(x, y) \qquad \phi = k_1$

$$h = \frac{b-a}{n-1} = \frac{1.2-0}{7-1} = 0.2$$

i	0.0	1.0	2.0	3.0	4.0	5.0	6.0
ti	0.0	0.2	0.4	0.6	0.8	1.0	1.2
yi	1.0	y1	y2	y3	y4	y5	y6

$k_1 = g(x_i, y_i) = -2y_i + t_i + 1$

$\phi_i = \mathrm{k}_1$

$t_0 = 0.0\ , \qquad y_0 = 1.000, \qquad \phi_0 = -2y_0 + t_0 + 1 = -1$

$y_1 = y_0 + h\phi_0 = 1.000 + (0.2)(-1) = 0.800$

$t_1 = 0.2\ , \qquad y_1 = 0.800, \qquad \phi_1 = -2y_1 + t_1 + 1 = -0.4$

$y_2 = y_1 + h\phi_1 = 0.800 + (0.2)(-0.4) = 0.720$

$t_2 = 0.4\ , \qquad y_2 = 0.720, \qquad \phi_2 = -2y_2 + t_2 + 1 = -0.04$

$y_3 = y_2 + h\phi_2 = 0.720 + (0.2)(-0.04\ ) = 0.712$

$t_3 = 0.6\ , \qquad y_3 = 0.712, \qquad \phi_3 = -2y_3 + t_3 + 1 = \ 0.176$

$y_4 = y_3 + h\phi_3 = 0.712 + (0.2)(0.176) = 0.747$

$t_4 = 0.8\ , \qquad y_4 = 0.747, \qquad \phi_3 = -2y_4 + t_4 + 1 = \ 0.306$

$y_4 = y_3 + h\phi_3 = 0.747 + (0.2)(0.306) = 0.808$

$t_5 = 1.0\ , \qquad y_5 = 0.808, \qquad \phi_5 = -2y_5 + t_5 + 1 = 0.383$

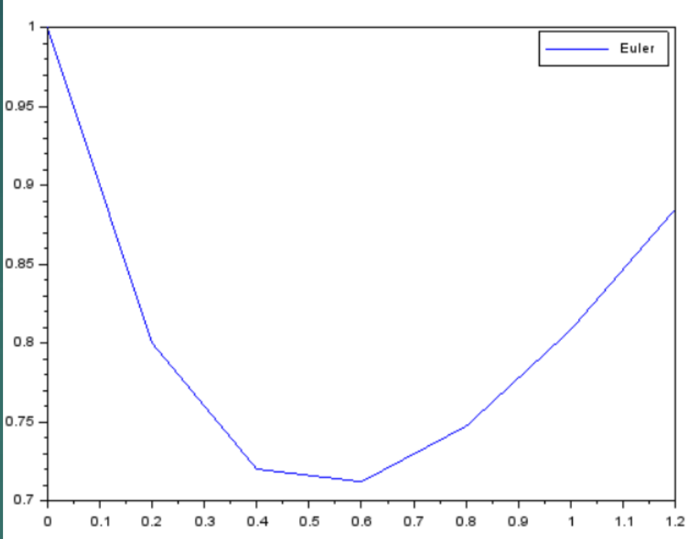
$y_6 = y_5 + h\phi_5 = 0.808 + (0.5)(0.383) = 0.885$

i	0.0	1.0	2.0	3.0	4.0	5.0	6.0
ti	0.0	0.2	0.4	0.6	0.8	1.0	1.2
yi	1.0	**0.800**	**0.720**	**0.712**	**0.747**	**0.808**	**0.885**

EDO de primeira ordem – Problemas de Valor Inicial

Seja a seguinte Equações Diferenciais Ordinária, definida no domínio $[a, b]$, com condição inicial $y_o = f(a)$, na qual seja possível se isolar o termo em y' (forma canônica):

$$y' = g(t, y)$$
$$t_i = a + ih, \qquad 0 \le i \le n-1$$
$$y_{i+1} = y_i + h\phi_i$$



$$y' = g(t, y) = -2y + t + 1$$
$$0 \le t \le 1.2$$
$y(0) = 1$ n=7

```
function [t,y]=edo_euler(g,a,b,ya,n,prt)
    h=(b-a)/(n-1);
    t=linspace(a,b,n)'
    y = [ya zeros(1,n-1)]'
    for i=1:n-1
        phi(i)= g(t(i),y(i))
        y(i+1)= y(i)+phi(i)*h
    end
    if (prt)
        printf("   i    t(i)  y(i)   phi(i)")
        disp([[0:n-1]',t,y,[phi;%nan]]);
        plot(t,y)
        legend("Euler")
    end
endfunction
```

```
--> deff('phi=g(t,y)','phi=-2*y+t+1')

--> [t,y]=edo_euler(g,0,1.2,1,7,%t);
   i     t(i)   y(i)     phi(i)
   0.    0.     1.          -1.
   1.    0.2    0.8         -0.4
   2.    0.4    0.72        -0.04
   3.    0.6    0.712        0.176
   4.    0.8    0.7472       0.3056
   5.    1.     0.80832      0.38336
   6.    1.2    0.884992    Nan
```

EDO de primeira ordem – Problemas de Valor Inicial

Seja a seguinte Equações Diferenciais Ordinária, definida no domínio $[a, b]$, com condição inicial $y_0 = f(a)$, na qual seja possível se isolar o termo em y' (forma canônica):

$$y' = g(x, y)$$

$$x_i = a + ih, \qquad 0 \leq i \leq n - 1$$

$$y_{i+1} = y_i + h\phi_i$$

```
function [t,y]=edo_fga(g,tipo,a,b,ya,n,prt,cor)
  h=(b-a)/(n-1);
  t=linspace(a,b,n)'
  y = [ya zeros(1,n-1)]'
  for i=1:n-1
    phi(i)= calc_phi(g,tipo,t(i),y(i),h)
    y(i+1)= y(i)+phi(i)*h
  end
  if (prt)
    printf("   i    t(i)  y(i)   phi(i)")
    disp([[0:n-1]',t,y,[phi;%nan]]);
    plot(t,y,cor)
    legend(tipo,2)
  end
endfunction
```

```
function [t,y]=edo_euler(g,a,b,ya,n,prt)
  h=(b-a)/(n-1);
  t=linspace(a,b,n)'
  y = [ya zeros(1,n-1)]'
  for i=1:n-1
    phi(i)= g(t(i),y(i))
    y(i+1)= y(i)+phi(i)*h
  end
  if (prt)
    printf("   i    t(i)  y(i)   phi(i)")
    disp([[0:n-1]',t,y,[phi;%nan]]);
    plot(t,y)
    legend("Euler")
  end
endfunction
```

```
function k=calc_phi(g,tipo,xi,yi,h)
    if (tipo=='eul')
        k = g(xi,yi)
    elseif (tipo=='heu')
        k1 = g(xi    , yi);
        k2 = g(xi+h, yi+k1*h);
        k  = (k1+k2)/2;
    elseif (tipo=='mid')
        k1 = g(xi      , yi);
        k2 = g(xi+ h/2, yi+k1*h/2);
        k  = k2;
    elseif (tipo=='ral')
        k1 = g( xi        , yi);
        k2 = g( xi+3/4*h, yi+3/4*k1*h);
        k  = 1/3*(k1 +2*k2);
    elseif (tipo=='rk3')
        k1 = g( xi        , yi);
        k2 = g( xi+1/2*h, yi+1/2*k1*h);
        k3 = g( xi+h      , yi+(2*k2-k1)*h);
        k= 1/6*(k1+4*k2+k3)
    elseif (tipo=='rk4')
        k1 = g( xi        , yi);
        k2 = g( xi+1/2*h, yi+1/2*k1*h);
        k3 = g( xi+1/2*h, yi+1/2*k2*h);
        k4 = g( xi+h      , yi+k3*h);
        k= 1/6*(k1+2*k2+2*k3+k4)
    end
endfunction
```

No método de Euler, usamos apenas uma inclinação calculada no início do intervalo

$$y_{i+1} = y_i + (k_1) * h$$

No método de Heun, usamos 2 inclinações, uma calculada no início e uma estimada do final do intervalo.

$$y_{i+1} = y_i + \left(\frac{k_1 + k_2}{2}\right) * h$$

No método do Midpoint, usamos apenas uma inclinação estimada no meio do intervalo.

$$y_{i+1} = y_i + (k_2) * h$$

No método de Ralston, usamos 2 inclinações, uma calculada no início e outra estimada a ¾ do final do intervalo.

$$y_{i+1} = y_i + \left(\frac{1}{3}k_1 + \frac{2}{3}k_2\right) * h$$

No método de Ralston, usamos 3 inclinações, uma calculada no início , uma estimada no meio, e uma estimada no final do intervalo.

$$y_{i+1} = y_i + \left(\frac{1}{6}k_1 + \frac{4}{6}k_2 + \frac{1}{6}k_3\right) * h$$

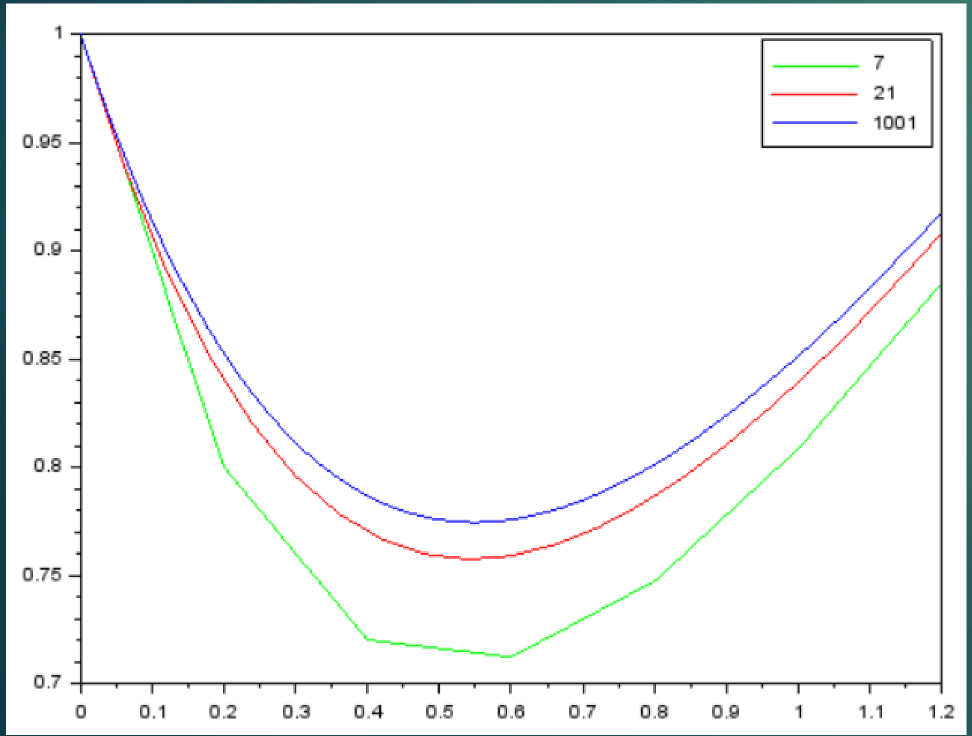
No método de Runge Kutta 4, usamos 4 inclinações, uma calculada no início , duas estimadas no meio, e uma estimada no final do intervalo.

$$y_{i+1} = y_i + \left(\frac{1}{6}k_1 + \frac{2}{6}k_2 + \frac{2}{6}k_3 + \frac{1}{6}k_4\right) * h$$

Método de Euler

```
function k=calc_phi(g,tipo,ti,yi,h)
    if (tipo=='eul')
        k = g(ti,yi)
                ⋮
    end
endfunction
```

```
--> deff('phi=g(t,y)','phi=-2*y+t+1')

--> [t,y]=edo_fga(g,'eul',0,1.2,1,7,%t,'g');
  i     t(i)  y(i)     phi(i)
  0.    0.    1.         -1.
  1.    0.2   0.8        -0.4
  2.    0.4   0.72       -0.04
  3.    0.6   0.712       0.176
  4.    0.8   0.7472      0.3056
  5.    1.    0.80832     0.38336
  6.    1.2   0.884992    Nan
```




```
--> [t,y]=edo_fga(g,'eul',0,1.2,1,21,%t,'r');
  i     t(i)  y(i)     phi(i)
  0.    0.     1.          -1.
  1.    0.06   0.94        -0.82
  2.    0.12   0.8908      -0.6616
  3.    0.18   0.851104    -0.522208
  4.    0.24   0.8197715   -0.399543
  5.    0.3    0.7957989   -0.2915979
  6.    0.36   0.7783031   -0.1966061
  7.    0.42   0.7665067   -0.1130134
  8.    0.48   0.7597259   -0.0394518
  9.    0.54   0.7573588    0.0252824
  10.   0.6    0.7588757    0.0822485
  11.   0.66   0.7638106    0.1323787
  12.   0.72   0.7717534    0.1764933
  13.   0.78   0.782343     0.2153141
  14.   0.84   0.7952618    0.2494764
  15.   0.9    0.8102304    0.2795392
  16.   0.96   0.8270027    0.3059945
  17.   1.02   0.8453624    0.3292752
  18.   1.08   0.8651189    0.3497622
  19.   1.14   0.8861047    0.3677907
  20.   1.2    0.9081721    Nan
```

```
--> [t,y]=edo_fga(g,'eul',0,1.2,1,1001,%t,'b')
  i     t(i)   y(i)       phi(i)
  0.       0.       1.          -1.
  1.       0.0012   0.9988      -0.9964
  2.       0.0024   0.9976043   -0.9928086
  3.       0.0036   0.9964129   -0.9892259
  4.       0.0048   0.9952259   -0.9856518
                ⋮
  997.     1.1964   0.9165333    0.3633334
  998.     1.1976   0.9169693    0.3636614
  999.     1.1988   0.9174057    0.3639886
  1000.    1.2      0.9178425    Nan

--> legend('7','21','1001')
```

Método de Heun

Seja a seguinte Equações Diferenciais Ordinária, definida no domínio $[a, b]$, com condição inicial $y_o = f(a)$

$$y' = g(x, y)$$

No método de Heun, a inclinação ϕ_i é a média função da inclinação $k_1 = g(x_i, y_i)$, no início do intervalo, e uma estimativa $k_2 = g(x_{i+1}, y_i + k_1 h)$ da inclinação no final do intervalo.

$$\boxed{\begin{aligned} k_1 &= g(x_i, y_i) \\ k_2 &= g(x_{i+1}, y_i + k_1 h) \\ \phi_i &= \frac{k_1 + k_2}{2} \end{aligned}}$$

Em um malha de n soluções equiespaçadas, com espaçamento h, no domínio [a,b], teremos a seguinte equação recursiva

$$x_i = [a, a + h, a + 2h, a + 3h, \ldots.., a + (n-2)h, b] \qquad onde, \qquad h = \frac{a-b}{n-1}$$

$$y_{i+1} = y_i + h\phi_i$$

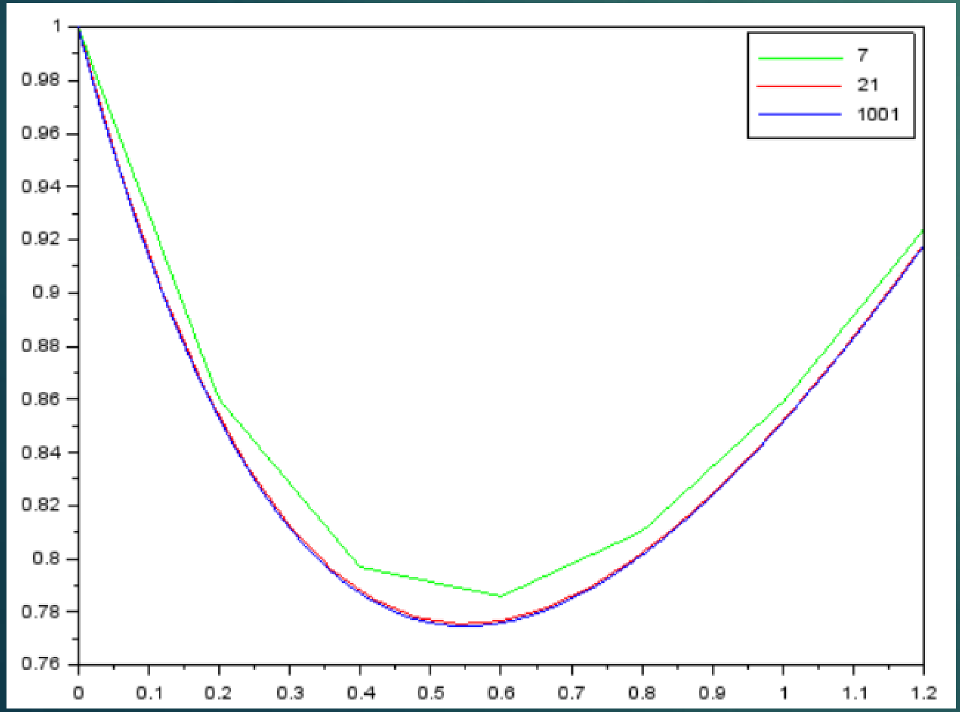
Método de Heun

```
function k=calc_phi(g,tipo,ti,yi,h)
    ⋮
    elseif (tipo=='heu')
        k1 = g(ti    , yi);
        k2 = g(ti+h, yi+k1*h);
        k  = (k1+k2)/2;

    ⋮
    end
endfunction
```

```
--> deff('phi=g(t,y)','phi=-2*y+t+1')

--> [t,y]=edo_fga(g,'heu',0,1.2,1,7,%t,'g');
  i     t(i)   y(i)      phi(i)
  0.    0.     1.          -0.7
  1.    0.2    0.86        -0.316
  2.    0.4    0.7968      -0.05488
  3.    0.6    0.785824     0.1226816
  4.    0.8    0.8103603    0.2434235
  5.    1.     0.859045     0.325528
  6.    1.2    0.9241506    Nan
```

```
--> [t,y]=edo_fga(g,'heu',0,1.2,1,21,%t,'r');
  i     t(i)   y(i)      phi(i)
  0.    0.     1.          -0.91
  1.    0.06   0.9454      -0.750952
  2.    0.12   0.9003429   -0.6098446
  3.    0.18   0.8637522   -0.4846541
  4.    0.24   0.834673    -0.3735852
  5.    0.3    0.8122578   -0.2750447
  6.    0.36   0.7957552   -0.1876197
  7.    0.42   0.784498    -0.1100562
  8.    0.48   0.7778946   -0.0412419
  9.    0.54   0.7754201    0.0198102
  10.   0.6    0.7766087    0.0739756
  11.   0.66   0.7810472    0.1220312
  12.   0.72   0.7883691    0.1646661
  13.   0.78   0.7982491    0.2024917
  14.   0.84   0.8103986    0.2360507
  15.   0.9    0.8245616    0.2658241
  16.   0.96   0.8405111    0.2922392
  17.   1.02   0.8580454    0.3156746
  18.   1.08   0.8769859    0.3364665
  19.   1.14   0.8971739    0.3549131
  20.   1.2    0.9184687    Nan
```

```
--> [t,y]=edo_fga(g,'heu',0,1.2,1,1001,%t,'b')
  i      t(i)     y(i)       phi(i)
  0.     0.       1.          -0.9982
  1.     0.0012   0.9988022   -0.9946086
  2.     0.0024   0.9976086   -0.9910259
  3.     0.0036   0.9964194   -0.9874517
  4.     0.0048   0.9952345   -0.9838861
              ⋮
  997.   1.1964   0.9167303    0.3631039
  998.   1.1976   0.917166     0.3634321
  999.   1.1988   0.9176021    0.3637595
  1000.  1.2      0.9180386    Nan

--> legend('7','21','1001')
```



Método do Midpoint

Seja a seguinte Equações Diferenciais Ordinária, definida no domínio $[a, b]$, com condição inicial $\mathrm{y}_o = f(a)$

$$y' = g(x, y)$$

No método do Midpoint, utilizamos a inclinação $\mathrm{k}_1 = g(x_i, y_i)$, no início do intervalo, para estimar a inclinação $\phi_i = g(x_i + \frac{h}{2}, y_i + \frac{k_1 h}{2})$ no meio do intervalo.

$$\boxed{\begin{aligned} k_1 &= g(x_i, y_i) \\ k_2 &= g\left(x_i + \frac{h}{2}, y_i + k_1\frac{h}{2}\right) \\ \phi_i &= \mathrm{k}_2 \end{aligned}}$$

Em um malha de n soluções equiespaçadas, com espaçamento h, no domínio [a,b], teremos a seguinte equação recursiva

$$x_i = [a, a+h, a+2h, a+3h, \ldots.., a+(n-2)h, b] \qquad onde, \qquad h = \frac{a-b}{n-1}$$

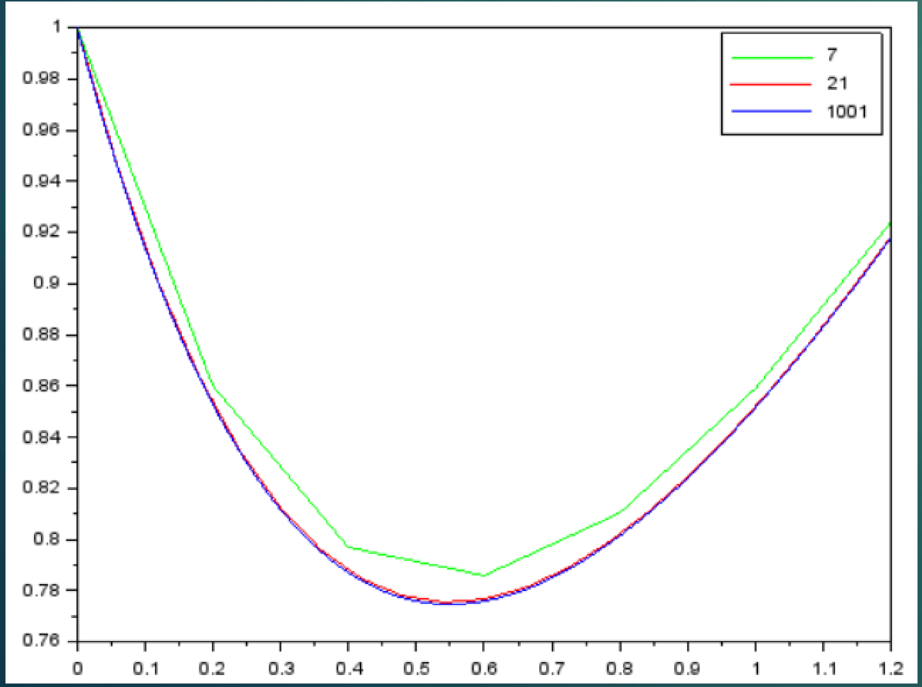
$$y_{i+1} = y_i + h\phi_i$$

Método do Midpoint

```
function k=calc_phi(g,tipo,ti,yi,h)
                  ⋮
    elseif (tipo=='mid')
        k1 = g(ti      , yi);
        k2 = g(ti+ h/2, yi+k1*h/2);
        k  = k2;

                  ⋮
    end
endfunction
```

```
--> [t,y]=edo_fga(g,'mid',0,1.2,1,7,%t,'g');
 i    t(i)  y(i)     phi(i)
 0.   0.    1.          -0.7
 1.   0.2   0.86        -0.316
 2.   0.4   0.7968      -0.05488
 3.   0.6   0.785824     0.1226816
 4.   0.8   0.8103603    0.2434235
 5.   1.    0.859045     0.325528
 6.   1.2   0.9241506    Nan
```

```
--> [t,y]=edo_fga(g,'mid',0,1.2,1,21,%t,'r');
 i     t(i)  y(i)     phi(i)
 0.    0.    1.          -0.91
 1.    0.06  0.9454      -0.750952
 2.    0.12  0.9003429   -0.6098446
 3.    0.18  0.8637522   -0.4846541
 4.    0.24  0.834673    -0.3735852
 5.    0.3   0.8122578   -0.2750447
 6.    0.36  0.7957552   -0.1876197
 7.    0.42  0.784498    -0.1100562
 8.    0.48  0.7778946   -0.0412419
 9.    0.54  0.7754201    0.0198102
 10.   0.6   0.7766087    0.0739756
 11.   0.66  0.7810472    0.1220312
 12.   0.72  0.7883691    0.1646661
 13.   0.78  0.7982491    0.2024917
 14.   0.84  0.8103986    0.2360507
 15.   0.9   0.8245616    0.2658241
 16.   0.96  0.8405111    0.2922392
 17.   1.02  0.8580454    0.3156746
 18.   1.08  0.8769859    0.3364665
 19.   1.14  0.8971739    0.3549131
 20.   1.2   0.9184687    Nan
```




```
--> [t,y]=edo_fga(g,'mid',0,1.2,1,1001,%t,'b')
 i      t(i)    y(i)       phi(i)
 0.     0.      1.           -0.9982
 1.     0.0012  0.9988022    -0.9946086
 2.     0.0024  0.9976086    -0.9910259
 3.     0.0036  0.9964194    -0.9874517
 4.     0.0048  0.9952345    -0.9838861
                 ⋮
 997.   1.1964  0.9167303    0.3631039
 998.   1.1976  0.917166     0.3634321
 999.   1.1988  0.9176021    0.3637595
 1000.  1.2     0.9180386    Nan

--> legend('7','21','1001')
```

Método de Ralston

Seja a seguinte Equações Diferenciais Ordinária, definida no domínio $[a, b]$, com condição inicial $y_o = f(a)$

$$y' = g(x, y)$$

No método de Ralston, a inclinação ϕ_i é a média ponderada da inclinação $k_1 = g(x_i, y_i)$, no início do intervalo, e uma estimativa $k_2 = g(x_i + \frac{3}{4}h, y_i + k_1\frac{3}{4}h)$ da inclinação a ¾ do final do intervalo. Sendo que k_1 terá peso 1/3 e k_2 terá peso 2/3.

$$\boxed{\begin{aligned} k_1 &= g(x_i, y_i) \\ k_2 &= g(x_i + \frac{3}{4}h, y_i + k_1\frac{3}{4}h) \\ \phi_i &= \frac{1}{3}(k_1 + 2k_2) \end{aligned}}$$

Em um malha de n soluções equiespaçadas, com espaçamento h, no domínio [a,b], teremos a seguinte equação recursiva

$$x_i = [a, a + h, a + 2h, a + 3h, \ldots., a + (n-2)h, b] \qquad onde, \qquad h = \frac{b-a}{n-1}$$

$$y_{i+1} = y_i + h\phi_i$$

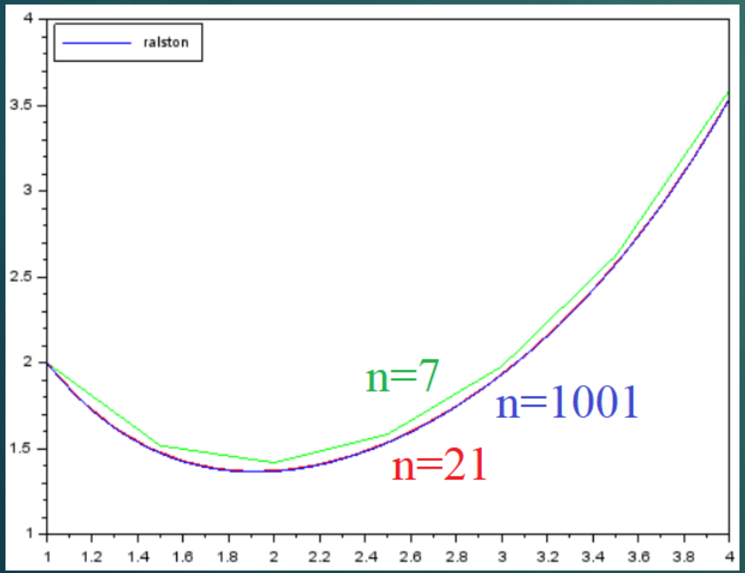
Método de Ralston

```
function k=calc_phi(g,tipo,ti,yi,h)
        ⋮
    elseif (tipo=='ral')
        k1 = g( ti         , yi);
        k2 = g( ti+3/4*h, yi+3/4*k1*h);
        k  = 1/3*(k1 +2*k2);
        ⋮
    end
endfunction
```

```
--> [t,y]=edo_fga(g,'ral',0,1.2,1,7,%t,'g');
 i    t(i)  y(i)    phi(i)
 0.   0.    1.          -0.7
 1.   0.2   0.86        -0.316
 2.   0.4   0.7968      -0.05488
 3.   0.6   0.785824     0.1226816
 4.   0.8   0.8103603    0.2434235
 5.   1.    0.859045     0.325528
 6.   1.2   0.9241506    Nan
```

```
--> [t,y]=edo_fga(g,'ral',0,1.2,1,21,%t,'r');
 i     t(i)  y(i)    phi(i)
 0.    0.    1.           -0.91
 1.    0.06  0.9454       -0.750952
 2.    0.12  0.9003429    -0.6098446
 3.    0.18  0.8637522    -0.4846541
 4.    0.24  0.834673     -0.3735852
 5.    0.3   0.8122578    -0.2750447
 6.    0.36  0.7957552    -0.1876197
 7.    0.42  0.784498     -0.1100562
 8.    0.48  0.7778946    -0.0412419
 9.    0.54  0.7754201     0.0198102
 10.   0.6   0.7766087     0.0739756
 11.   0.66  0.7810472     0.1220312
 12.   0.72  0.7883691     0.1646661
 13.   0.78  0.7982491     0.2024917
 14.   0.84  0.8103986     0.2360507
 15.   0.9   0.8245616     0.2658241
 16.   0.96  0.8405111     0.2922392
 17.   1.02  0.8580454     0.3156746
 18.   1.08  0.8769859     0.3364665
 19.   1.14  0.8971739     0.3549131
 20.   1.2   0.9184687     Nan
```

```
--> [t,y]=edo_fga(g,'ral',0,1.2,1,1001,%t,'b')
 i      t(i)    y(i)       phi(i)
 0.     0.      1.          -0.9982
 1.     0.0012  0.9988022   -0.9946086
 2.     0.0024  0.9976086   -0.9910259
 3.     0.0036  0.9964194   -0.9874517
 4.     0.0048  0.9952345   -0.9838861
              ⋮
 997.   1.1964  0.9167303    0.3631039
 998.   1.1976  0.917166     0.3634321
 999.   1.1988  0.9176021    0.3637595
 1000.  1.2     0.9180386    Nan

--> legend('7','21','1001')
```



Exemplo Runge-Kutta 3

$3y' + 4y - xy - 2x + 1 = 0$

$1 \leq x \leq 4$

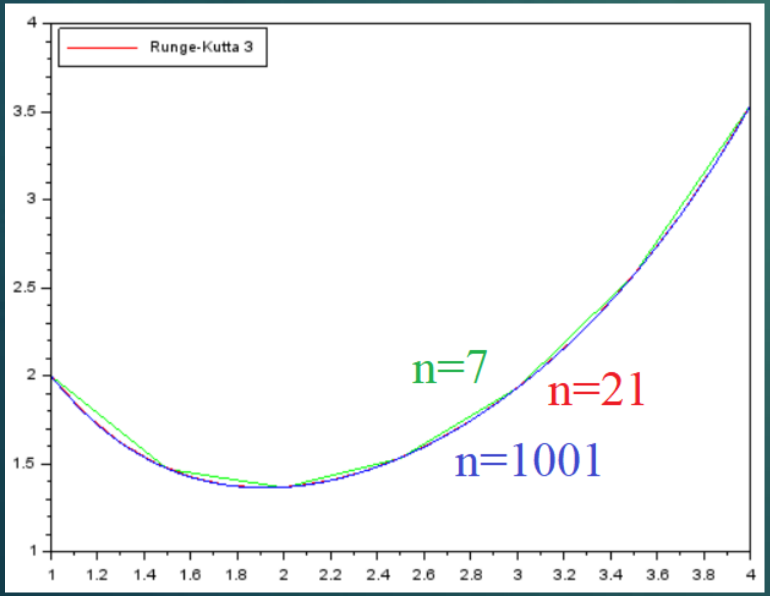
Condição inicial $y(1) = 2$

```
function k=calc_phi(g,tipo,ti,yi,h)
    ⋮
    elseif (tipo=='rk3')
        k1 = g( ti        , yi);
        k2 = g( ti+1/2*h, yi+1/2*k1*h);
        k3 = g( ti+h      , yi+(2*k2-k1)*h);
        k= 1/6*(k1+4*k2+k3)
    ⋮
    end
endfunction
```

```
--> [t,y]=edo_fga(g,'rk3',0,1.2,1,7,%t,'g');
   i     t(i)   y(i)      phi(i)
   0.    0.     1.         -0.74
   1.    0.2    0.852      -0.3299733
   2.    0.4    0.7860053  -0.0555288
   3.    0.6    0.7748996   0.128166
   4.    0.8    0.8005328   0.2511191
   5.    1.     0.8507566   0.3334157
   6.    1.2    0.9174398   Nan
```

```
--> [t,y]=edo_fga(g,'rk3',0,1.2,1,21,%t,'r');
   i     t(i)   y(i)      phi(i)
   0.    0.     1.         -0.9136
   1.    0.06   0.945184   -0.7537388
   2.    0.12   0.8999597  -0.611956
   3.    0.18   0.8632423  -0.4862071
   4.    0.24   0.8340699  -0.3746789
   5.    0.3    0.8115892  -0.2757632
   6.    0.36   0.7950434  -0.1880337
   7.    0.42   0.7837613  -0.1102254
   8.    0.48   0.7771478  -0.0412162
   9.    0.54   0.7746748   0.0199889
   10.   0.6    0.7758742   0.0742724
   11.   0.66   0.7803305   0.122417
   12.   0.72   0.7876755   0.1651171
   13.   0.78   0.7975826   0.2029884
   14.   0.84   0.8097619   0.2365768
   15.   0.9    0.8239565   0.2663668
   16.   0.96   0.8399385   0.2927879
   17.   1.02   0.8575058   0.3162211
   18.   1.08   0.876479    0.3370043
   19.   1.14   0.8966993   0.3554372
   20.   1.2    0.9180255   Nan
```

```
--> [t,y]=edo_fga(g,'rk3',0,1.2,1,1001,%t,'b')
   i     t(i)     y(i)       phi(i)
   0.    0.       1.          -0.9982014
   1.    0.0012   0.9988022   -0.9946101
   2.    0.0024   0.9976086   -0.9910273
   3.    0.0036   0.9964194   -0.9874531
   4.    0.0048   0.9952344   -0.9838875
   ⋮
   997.  1.1964   0.9167301   0.3631041
   998.  1.1976   0.9171658   0.3634323
   999.  1.1988   0.917602    0.3637596
   1000. 1.2      0.9180385   Nan

--> legend('7','21','1001')
```



Exemplo Runge-Kutta 4

$3y' + 4y - xy - 2x + 1 = 0$

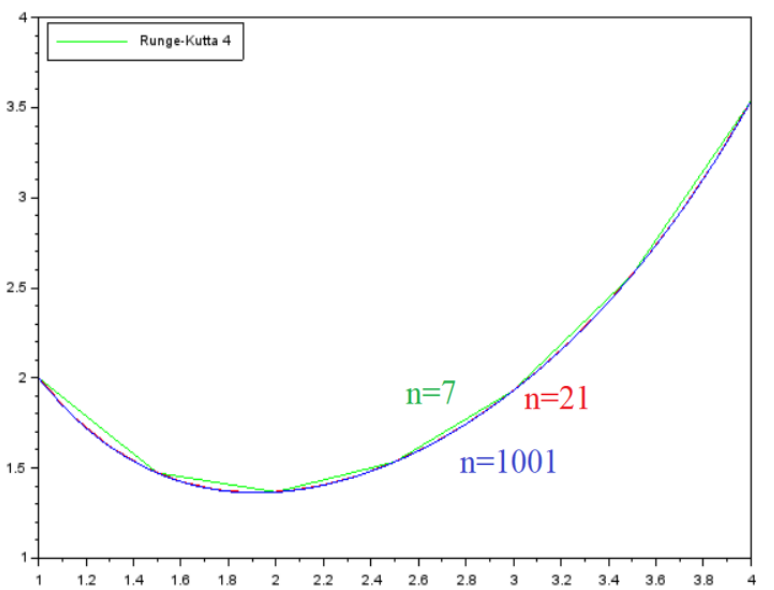
$1 \leq x \leq 4$

Condição inicial $y(1) = 2$

```
function k=calc_phi(g,tipo,ti,yi,h)
:
    elseif (tipo=='rk4')
        k1 = g( ti      , yi);
        k2 = g( ti+1/2*h, yi+1/2*k1*h);
        k3 = g( ti+1/2*h, yi+1/2*k2*h);
        k4 = g( ti+h    , yi+k3*h);
        k= 1/6*(k1+2*k2+2*k3+k4)
    end
endfunction
```

```
--> [t,y]=edo_fga(g,'rk4',0,1.2,1,7,%t,'g');
  i    t(i)   y(i)      phi(i)
  0.   0.     1.         -0.736
  1.   0.2    0.8528     -0.3286144
  2.   0.4    0.7870771  -0.0555031
  3.   0.6    0.7759765   0.1275907
  4.   0.8    0.8014946   0.2503368
  5.   1.     0.851562    0.3326258
  6.   1.2    0.9180872   Nan
```

```
--> [t,y]=edo_fga(g,'rk4',0,1.2,1,21,%t,'r');
  i    t(i)   y(i)       phi(i)
  0.   0.     1.         -0.913492
  1.   0.06   0.9451905  -0.7536552
  2.   0.12   0.8999712  -0.6118927
  3.   0.18   0.8632576  -0.4861606
  4.   0.24   0.834088   -0.3746462
  5.   0.3    0.8116092  -0.2757417
  6.   0.36   0.7950647  -0.1880214
  7.   0.42   0.7837834  -0.1102204
  8.   0.48   0.7771702  -0.041217
  9.   0.54   0.7746972   0.0199835
  10.  0.6    0.7758962   0.0742634
  11.  0.66   0.780352    0.1224054
  12.  0.72   0.7876963   0.1651036
  13.  0.78   0.7976025   0.2029735
  14.  0.84   0.8097809   0.236561
  15.  0.9    0.8239746   0.2663505
  16.  0.96   0.8399556   0.2927715
  17.  1.02   0.8575219   0.3162047
  18.  1.08   0.8764942   0.3369882
  19.  1.14   0.8967135   0.3554215
  20.  1.2    0.9180388   Nan
```

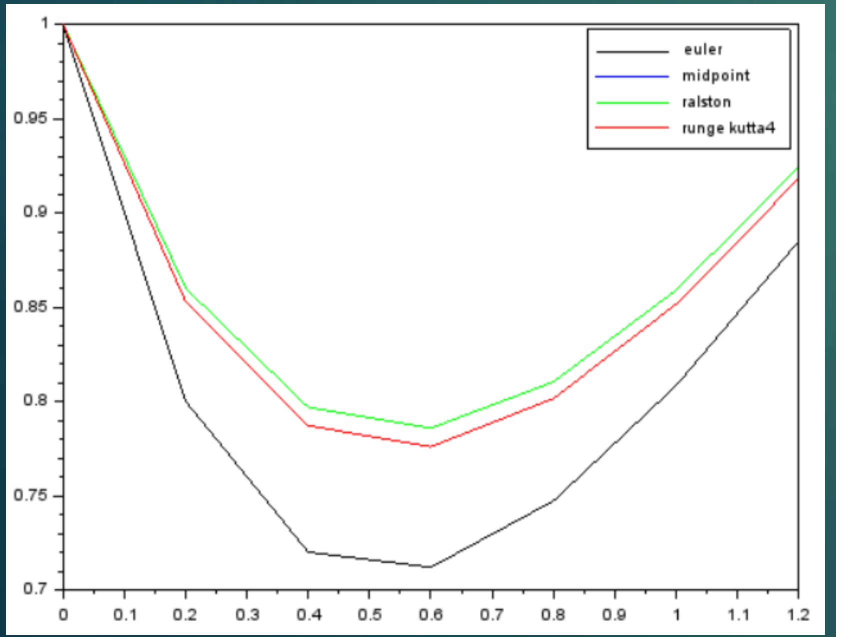



```
--> [t,y]=edo_fga(g,'rk4',0,1.2,1,1001,%t,'b')
  i      t(i)    y(i)       phi(i)
  0.     0.      1.         -0.9982014
  1.     0.0012  0.9988022  -0.9946101
  2.     0.0024  0.9976086  -0.9910273
  3.     0.0036  0.9964194  -0.9874531
  4.     0.0048  0.9952344  -0.9838875
                 :
  997.   1.1964  0.9167301   0.3631041
  998.   1.1976  0.9171658   0.3634323
  999.   1.1988  0.917602    0.3637596
  1000.  1.2     0.9180385   Nan

--> legend('7','21','1001')
```

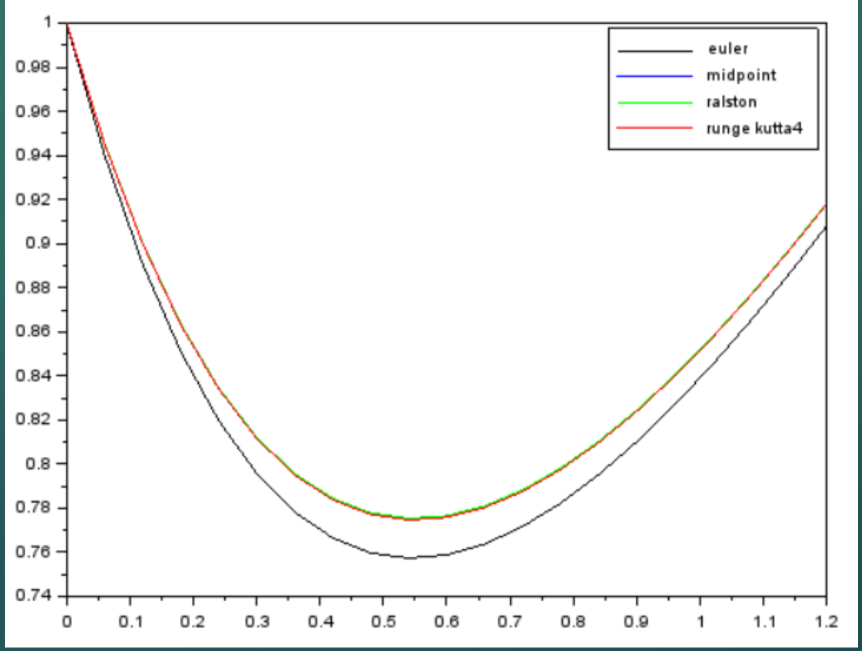
```
function Test_EDO(g,a,b,ya,n)
    [x,y]=edo_fga(g,'eul',a,b,ya,n,%t,'k');
    [x,y]=edo_fga(g,'mid',a,b,ya,n,%t,'b');
    [x,y]=edo_fga(g,'ral',a,b,ya,n,%t,'g');
    [x,y]=edo_fga(g,'rk4',a,b,ya,n,%t,'r');
    legend("euler","midpoint","ralston","runge kutta4",2)
endfunction
```

```
--> deff('phi=g(t,y)','phi=-2*y+t+1')
```

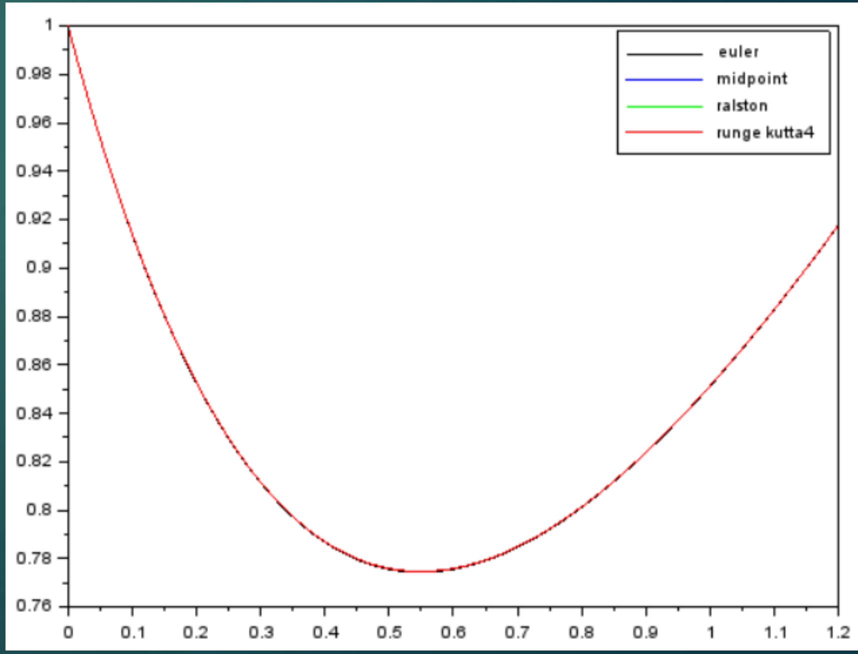
```
--> Test_EDO(g,1,4,2,7);
```




```
--> Test_EDO(g,1,4,2,21);
```

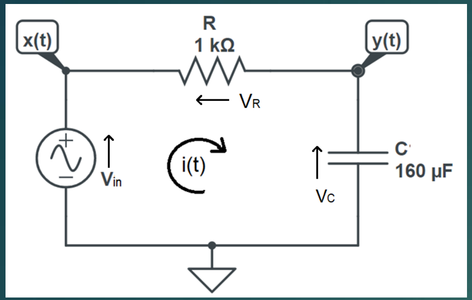



```
--> Test_EDO(g,1,4,2,1001);
```



$$y'(t) + 2y(t) = 2x(t)$$

$x(t)$

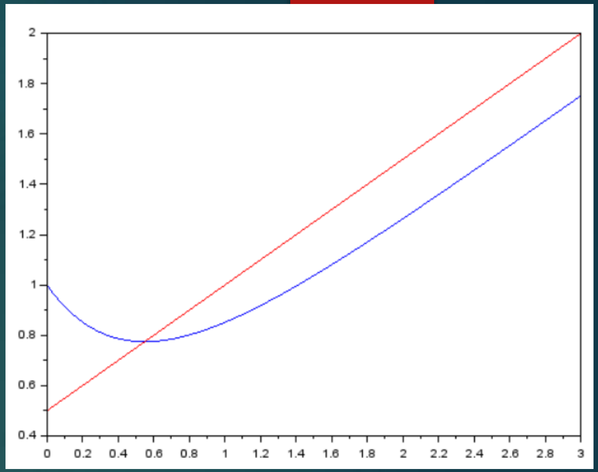


$y(t)$

Uma equação diferencial linear (na qual os coeficientes de y'', y' e y não dependem de t) pode ser interpretada como sendo um Sistema que recebe o sinal de entrada $x(t)$, que é transformado pelo sistema no sinal de saída $y(t)$.

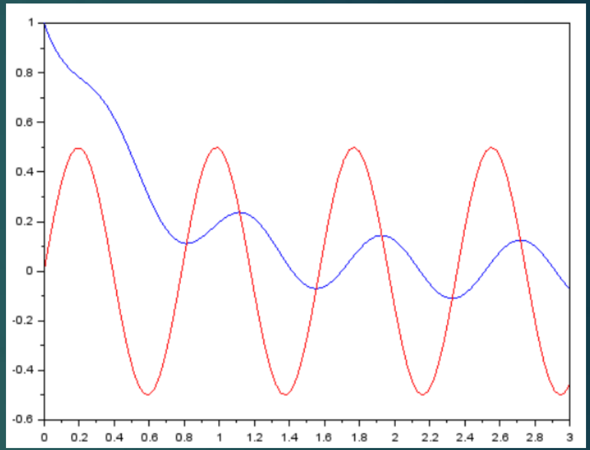
Para a entrada: $x(t) = 0.5(t+1)$;
$0 \leq t \leq 3$; $y(0) = 1$
$y'(t) = -2y(t) + t + 1$

```
--> deff('phi=g(t,y)','phi=-2*y+t+1')

--> [t,y]=edo_fga(g,'rk4',0,3,1,101);

--> plot(t,0.5*(t+1),'r')

--> plot(t,y)
```



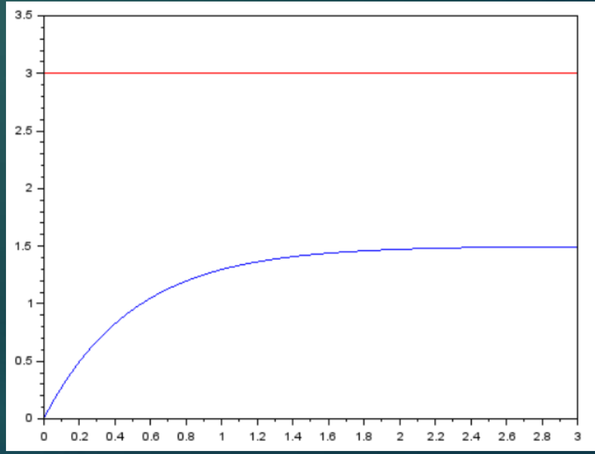
Para a entrada: $x(t) = 0.5 * \sin(8t)$;
$0 \leq t \leq 3$; $y(0) = 1$
$y'(t) = -2y(t) + \sin(8t)$

```
--> deff('phi=g(t,y)','phi=-2*y+sin(8*t)')

--> [t,y]=edo_fga(g,'rk4',0,3,1,101);

--> plot(t,0.5*sin(8*t),'r')

--> plot(t,y)
```



Para a entrada: $x(t) = 1.5$;
$0 \leq t \leq 1$; $y(0) = 0$
$y'(t) = -2y(t) + 3$

```
--> deff('phi=g(t,y)','phi=-2*y+3')

--> [t,y]=edo_fga(g,'rk4',0,3,0,101);

--> plot(t,3,'r')

--> plot(t,y)
```



Conteúdo Adicional
EDO de 2ª ordem

- Seja a seguinte equação diferencial referente a um problema de valor inicial, definido no domínio $t \in [a, b]$, com condições iniciais $f(a) = y_o$ e $f'(a) = y'_0$:

$$y'' - y' + ty = e^t(t^2 + 1) \qquad (i)$$

- Primeiro colocamos e equação diferencial (i) na forma canônica:

$$y'' = y' - ty + e^t(t^2 + 1) \qquad (ii)$$

- Então transformamos esta EDO de 2ª ordem em um sistema de 2 EDO's de 1ª ordem

$$\begin{cases} y' = g_1(t, y, u) = u \\ u' = g_2(t, y, y) = u - ty + e^t(t^2 + 1) \end{cases}$$

Vamos então escolher as seguintes condições iniciais:

Domínio $t \in [2, 3]$,
Condições iniciais : $f(2) = 20$ e $f'(2) = -16.5$

Conteúdo Adicional
EDO de 2ª ordem

- Seja a seguinte equação diferencial referente a um problema de valor inicial, definido no domínio $t \in [a, b]$, com condições iniciais $f(a) = y_o$ e $f'(a) = y'_0$:

$$y'' - y' + ty = e^t(t^2 + 1) \qquad (i)$$

- Primeiro colocamos e equação diferencial (i) na forma canônica:

$$y'' = y' - ty + e^t(t^2 + 1) \qquad (ii)$$

- Então transformamos esta EDO de 2ª ordem em um sistema de 2 EDO's de 1ª ordem

$$\begin{cases} y' = g_1(t, y, u) = u \\ u' = g_2(t, y, y) = u - ty + e^t(t^2 + 1) \end{cases}$$

Vamos então escolher as seguintes condições iniciais:

Domínio $t \in [2, 3]$,
Condições iniciais : $f(2) = 20$ e $f'(2) = -16.5$

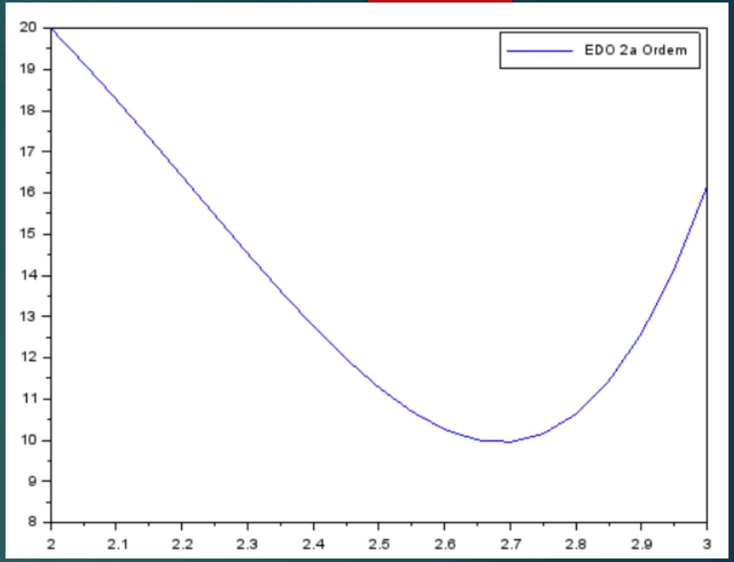
- O sistema de duas EDO's de 1ª ordem pode ser resolvido com qualquer um do métodos analisados, de preferência Runge-Kutta 4ª ordem que é o mais utilizado.

$$y'' - y' + ty = e^t(t^2 + 1)$$

$$\begin{cases} y' = g_1(t, y, u) = g_1(u) = u \\ u' = g_2(t, y, u) = u - ty + e^t(t^2 + 1) \end{cases}$$

$x \in [2, 3]$,

$y(2) = 20$ e $u(2) = y'(2) = -16.5$

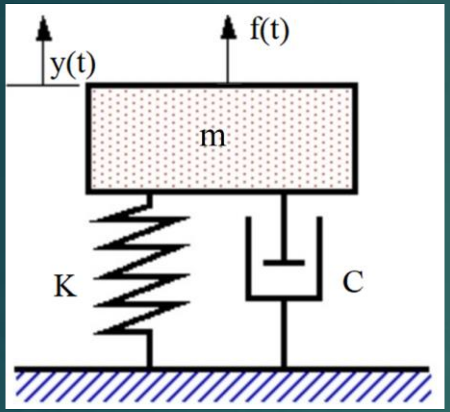


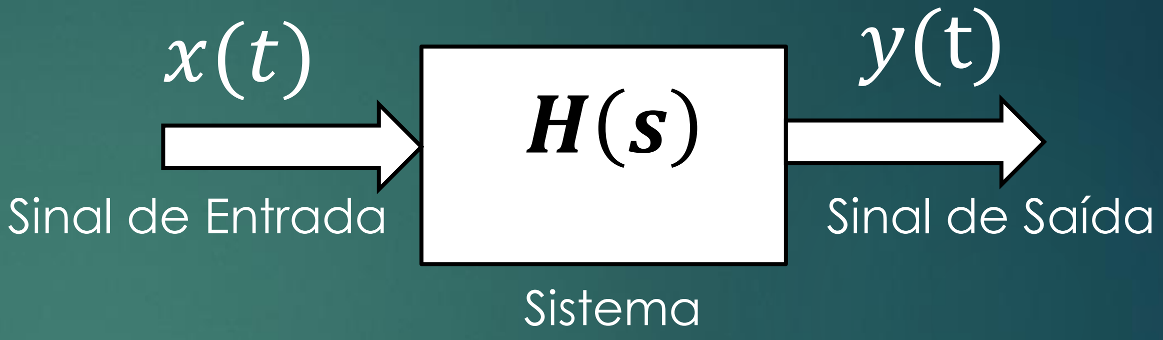

```
function [x,y]=EDO_2a_Ordem(g,a,b,ya,dya,n,prt)
    h=(b-a)/(n-1);
    t=linspace(a,b,n)'
    y = [ya zeros(1,n-1)]'
    u = [dya zeros(1,n-1)]'
    for i=1:n-1
        k1_a = u(i);
        k1_b = g(t(i),y(i),k1_a);
        k2_a = u(i)+1/2*k1_b*h;
        k2_b = g(t(i)+1/2*h, y(i)+1/2*k1_a*h, k2_a);
        k3_a = u(i)+1/2*k2_b*h;
        k3_b = g(t(i)+1/2*h, y(i)+1/2*k2_a*h, k3_a);
        k4_a = u(i)+k3_b*h;
        k4_b = g(t(i)+h, y(i)+k3_a*h, k4_a);
        phi_a=1/6*(k1_a+2*k2_a+2*k3_a+k4_a)
        y(i+1)=y(i)+phi_a*h
        phi_b= 1/6*(k1_b+2*k2_b+2*k3_b+k4_b)
        u(i+1)=u(i)+phi_b*h
    end
    if (prt)
        printf("  i\tx(i)\tu(i)\ty(i)")
        disp([[0:n-1]',t, u, y]);
        plot(t,y,"b")
        legend("EDO 2a Ordem",1)
    end
endfunction
```

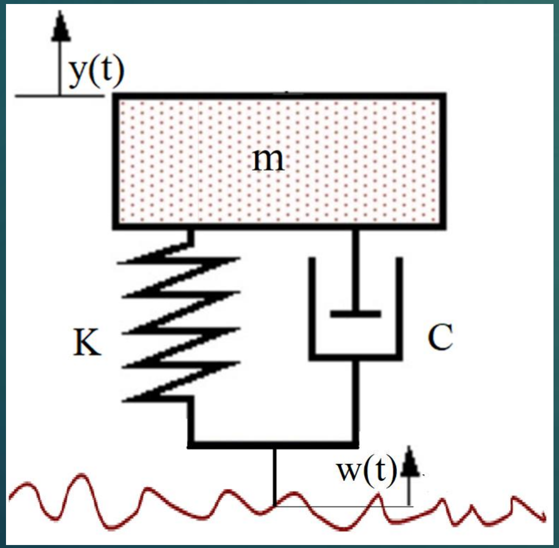
```
function k=calc_phi(g,tipo,ti,yi,h)
        ⋮
    elseif (tipo=='rk4')
        k1 = g( ti       , yi);
        k2 = g( ti+1/2*h, yi+1/2*k1*h);
        k3 = g( ti+1/2*h, yi+1/2*k2*h);
        k4 = g( ti+h     , yi+k3*h);
        k= 1/6*(k1+2*k2+2*k3+k4)
    end
endfunction
```

```
--> deff('phi=g(t,y,u)','phi=u-t*y+exp(t).*(t^2+1)')

--> EDO_2a_Ordem(g,2,3,20,-16.5,21,%t);
  i    x(i)    u(i)      y(i)
  0.    2.    -16.5       20.
  1.    2.05  -17.397651  19.15187
  2.    2.1   -18.113804  18.263259
  3.    2.15  -18.614873  17.344073
  4.    2.2   -18.865217  16.40595
  5.    2.25  -18.827259  15.462356
  6.    2.3   -18.461628  14.528685
  7.    2.35  -17.727352  13.622338
  8.    2.4   -16.582073  12.762801
  9.    2.45  -14.98231   11.971706
  10.   2.5   -12.883757  11.272884
  11.   2.55  -10.241622  10.692393
  12.   2.6   -7.0110033  10.258533
  13.   2.65  -3.1473092  10.001847
  14.   2.7    1.3932917  9.9550883
  15.   2.75   6.6533804  10.153173
  16.   2.8    12.673768  10.633104
  17.   2.85   19.492946  11.433869
  18.   2.9    27.146518  12.596312
  19.   2.95   35.666609  14.162971
  20.   3.     45.081265  16.177892
```

Sistema mecânico







- f será o perfil da estrada
- y será o movimento do carro

$$my'' + C(y' - x') + k(y - x) = 0$$

$$my'' + Cy' + Ky = Cx' + Kx$$

$$y'' + \frac{C}{m}y' + \frac{K}{m}y = \frac{C}{m}x' + \frac{K}{m}x$$

Resolvendo equação Diferencial

$$\boldsymbol{y'' + \frac{C}{m}y' + \frac{K}{m}y = \frac{C}{m}x' + \frac{K}{m}x}$$

- *m=300kg* *Massa do automóvel*
- *K= 13500 N/m* *Rigidez da mola*
- *C=1800 N/(m/s)* *Coeficiente de amortecimento*
- $x(t) = 0.1sen(60t)$ *Oscilação de ~ 10Hz e 10cm de amplitude*
- $y(0^-)$*=0.1m*
- $y'(0^-)$*=0.0 m/s*

$$\boldsymbol{y''(t) + 6y'(t) + 45y(t) = 6x'(t)} + 45x(t)$$

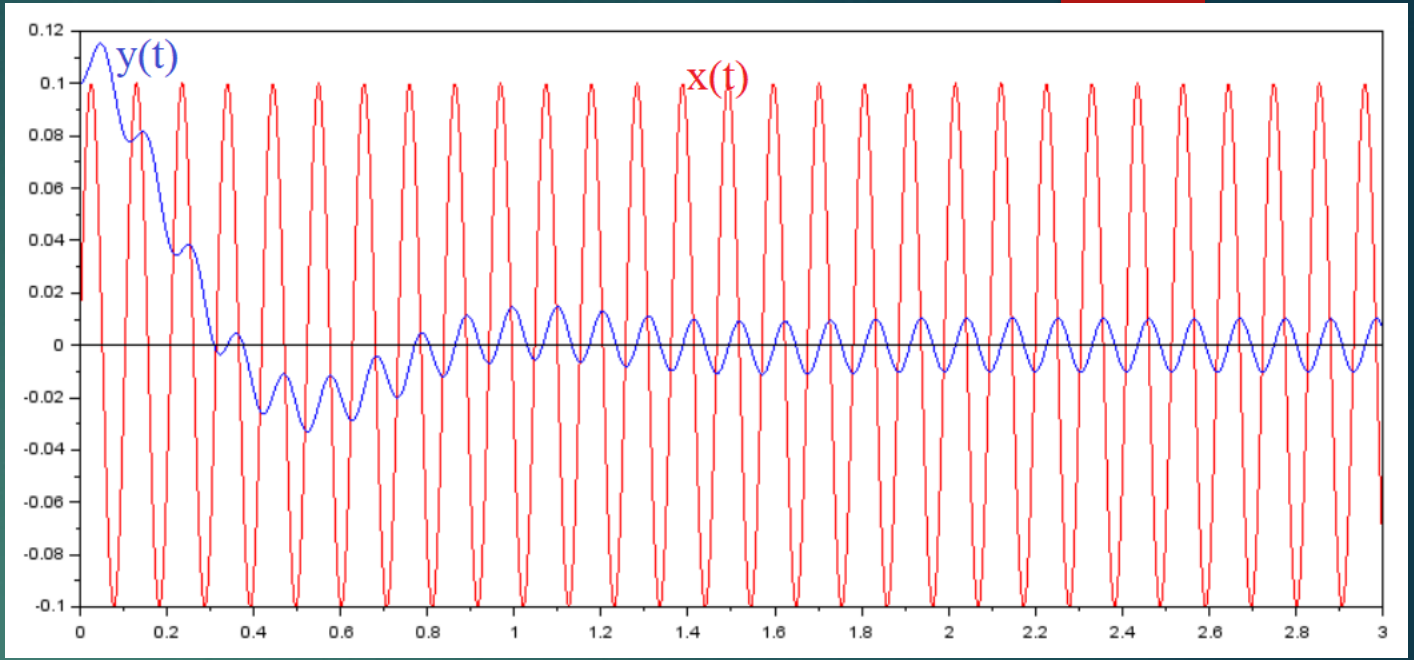
$$x(t) = 0.1sen(60t)$$

$$\boldsymbol{y''(t) + 6y'(t) + 45y(t) = 36cos}(60t) + 4.5\sin(60t)$$

- Na forma canônica:

$$\boldsymbol{y''(t) = -6y'(t) - 45y(t) + 36cos}(60t) + 4.5\sin(60t) \qquad 0 \le t \le 3$$

Symbolab e Scilab



$\boldsymbol{y''(t) + 6y'(t) + 45y(t) = 36cos}(60t) + 4.5\sin(60t)$

$x(t) = 0.1sen(60t)$ $\qquad$ $y(0^-)=0.1m$

$y'(0^-)=0.0\,m/s$

$y = e^{-3t}\big(0.1101507\cos(6t) + 0.0574544\ sen(6t)\big) - 0.0101507\cos(60t) - 0.0002379sen(60t)$

SCILAB

$--> \ t = [0{:}\,0.001{:}\,3];$

$--> \ x = 0.1 * \sin(60 * t);$

$--> \ y = \exp(-3 * t) .* (0.11015 * \cos(6 * t) + 0.05746 * \sin(6 * t))...$

$\quad -0.00024 * \sin(60 * t) - 0.010151 * \cos(60 * t);$

$--> \ plot(t, x, 'r');$

$--> \ plot(t, y)$

Conferir no Symbolab
https://pt.symbolab.com/solver/ordinary-differential-equation-calculator/

Digite uma EDO.:y''+6y'+45y=36cos(60t)+4.5sen(60t),y(0)=0.1,y'(0)=0

$$y = e^{-3t}\left(0.11015067\cos(6t) + \frac{0.344726394}{6}\sin(6t)\right) - 0.0002379064\sin(60t) - 0.01015067\cos(60t)$$

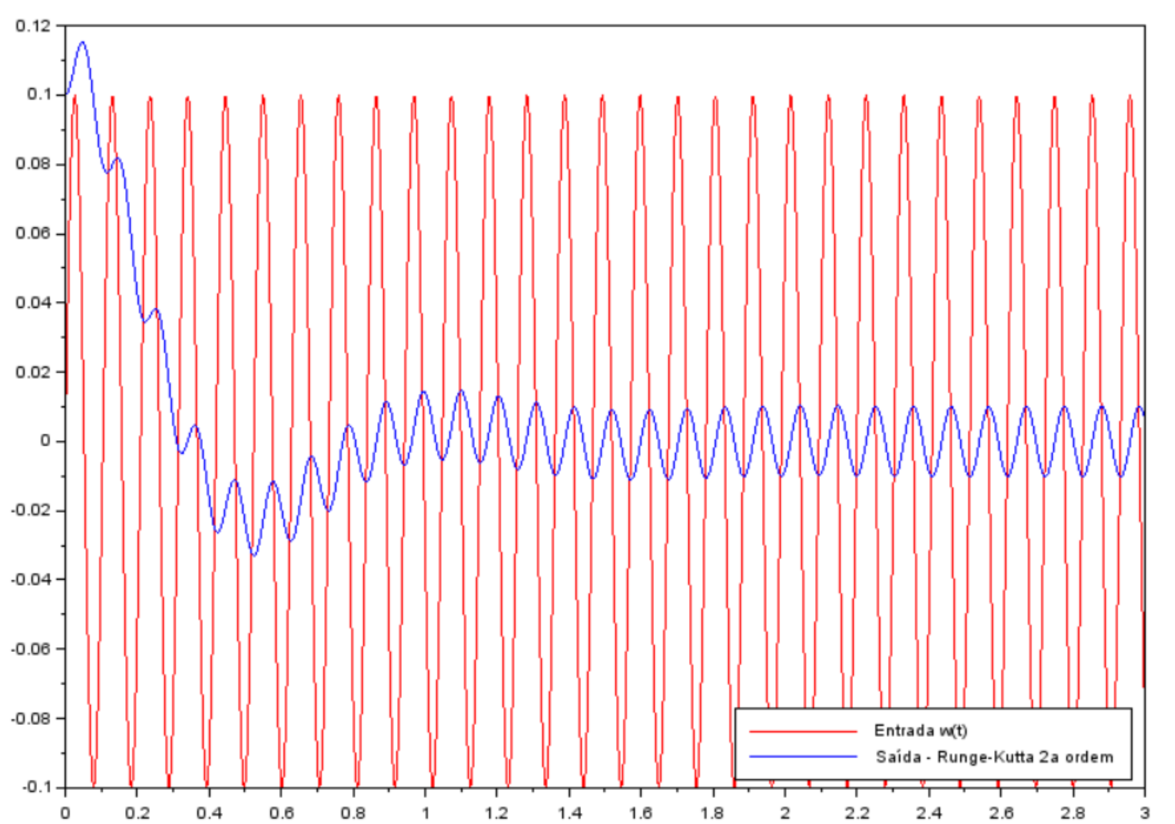
$$\boldsymbol{y}'' + \boldsymbol{6y}' + \boldsymbol{45y} = \boldsymbol{36cos}(60t) + 4.5\sin(60t)$$

$$\begin{cases} y' = g_1(t, y, u) = g_1(u) = u \\ u' = g_2(t, y, u) = -6u - 45y + \boldsymbol{36cos}(60t) + 4.5\sin(60t) \end{cases}$$

$y(0^-)$=*0.1m*　　u(0) =$y'(0^-)$=*0.0 m/s*

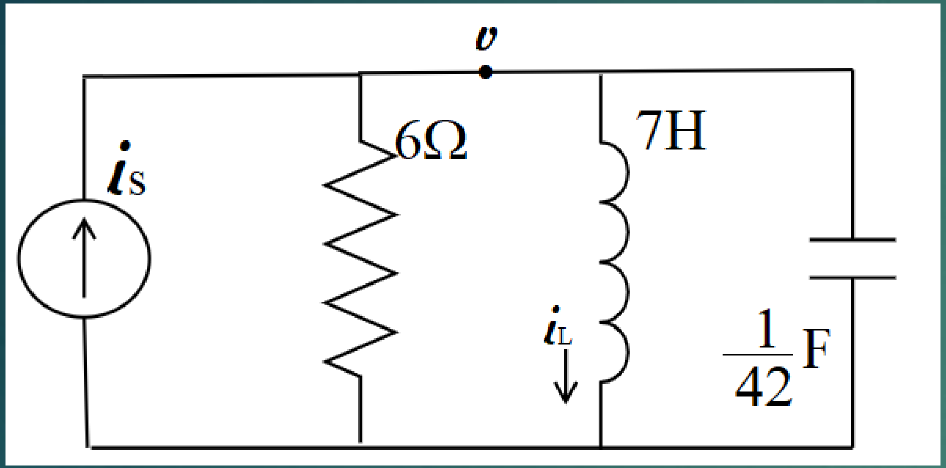
$0 \leq t \leq 3$

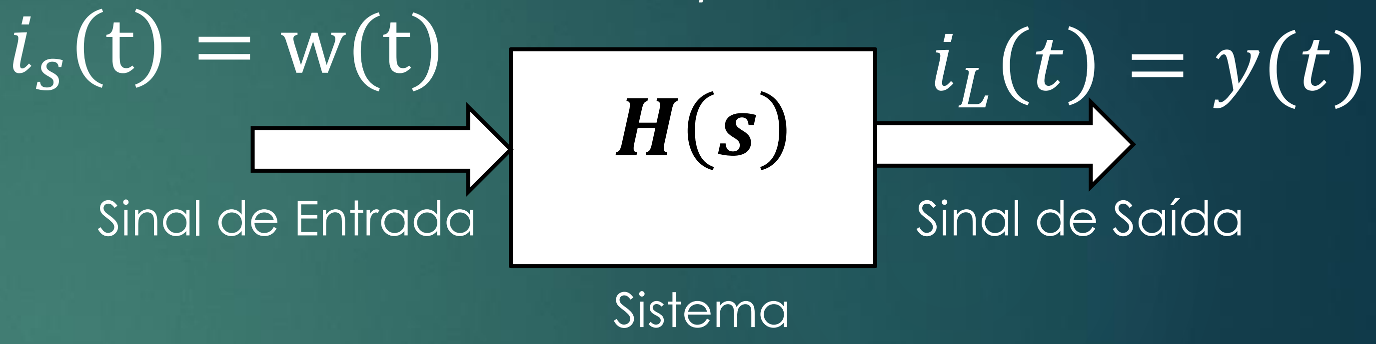
```
--> deff('phi=g2(t,y,u)','phi=-6*u-45*y+36*cos(60*t)+4.5*sin(60*t)');

--> deff('y=w(t)','y=0.1*sin(60*t)');

--> deff('phi=g2(t,y,u)','phi=-6*u-45*y+36*cos(60*t)+4.5*sin(60*t)');

--> t=linspace(0,3,1000);

--> plot(t,feval(t,w),'r')

--> [t,y]=Runge_Kutta4_EDO_2a_Ordem(g2,0,3,0.1,0,1000,%f);

--> plot(t,y,'b')

--> legend("Entrada w(t)","Saída - Runge-Kutta 2a ordem",4)
```



Circuito elétrico

Condições iniciais
$i(0^-) = 5A$
$i'(0^-) = \frac{10}{7}$





- *Nó v* $\quad i_s = i_R + i_L + i_C;$ $\quad \begin{cases} i_R = \frac{v}{R} \\ i_L = \frac{1}{L}\int v\,dt \\ i_C = cv' \end{cases}$
- $i_s = \frac{v}{R} + i_L + Cv'$ (1)
- No indutor $v = L{i_L}'$ (2)
- $\frac{dv}{dt} = Li_L"$ (3)

- Substituindo 2 e 3 em 1, e com $i_L = i$, teremos:
- $\frac{L}{R}i' + i + \mathrm{CL\ i"} = i_s$
- $\mathrm{i}'' + \frac{1}{\mathrm{CR}}\mathrm{i}' + \frac{1}{\mathrm{CL}}\mathrm{i} = \frac{1}{CL}i_s$
- Substituindo R=6, L=7 e C=1/42 e
- $i_s = 8e^{-2t} = entrada\ forçada:$

$$\boldsymbol{i" + 7i' + 6i = 6\ i_s}$$

Resolvendo equação Diferencial

$$y''(t) + 7y'(t) + 6y(t) = 6x(t)$$

Condições iniciais

$y(0^-) = 5A$

$y'(0^-) = \frac{10}{7}$

$$y''(t) + 7y'(t) + 6y(t) = 6x(t)$$

$$x(t) = 8e^{-2t} = entrada\ forçada$$

$$y''(t) + 7y'(t) + 6y(t) = \mathbf{48\ e^{-2t}}$$

Na forma canônica:

$$\boldsymbol{y''(t) = -7y'(t) - 6y + 48\ e^{-2t}} \quad 0 \le t \le 5$$

Symbolab e Scilab

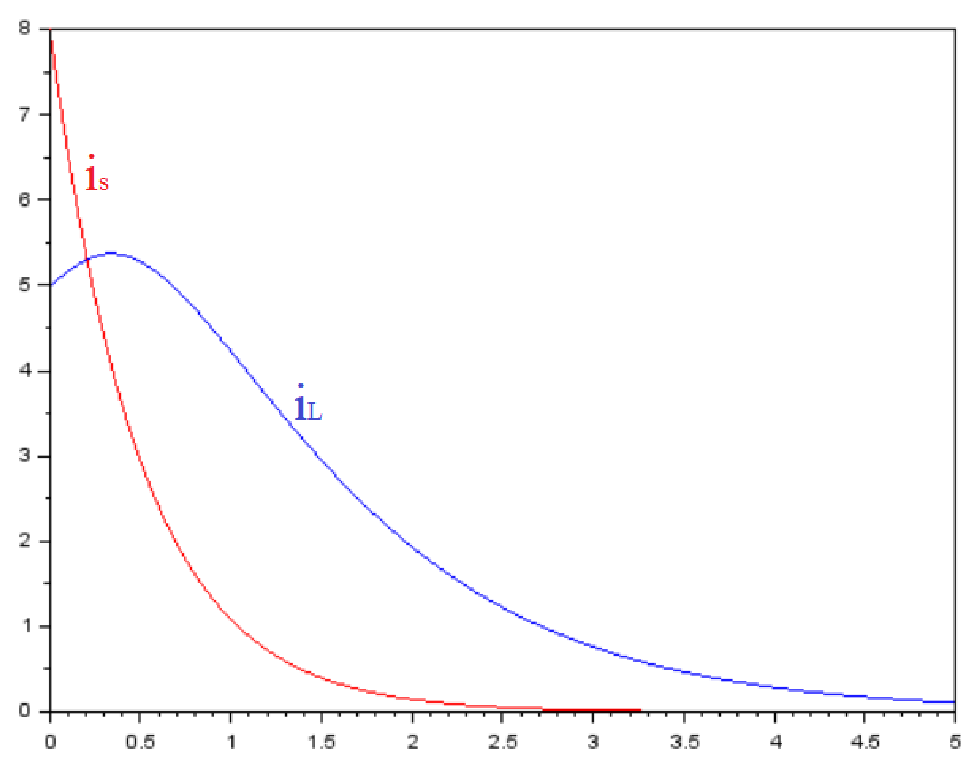
$$\boldsymbol{y''(t) + 7y'(t) + 6y(t) = 6x(t)}$$
$$y(0^-) = 5A$$
$$y'(0^-) = \frac{10}{7}$$
$$x(t) = 8e^{-2t}$$
$$y = \frac{556}{35}e^{-t} + \frac{39}{35}e^{-6t} - 12e^{-2t}$$

Conferir no Symbolab

https://pt.symbolab.com/solver/ordinary-differential-equation-calculator/

$$y'' + 7y' + 6y = 48\exp(-2t);\ y(0) = 5;\ y'(0) = 10/7$$

$$y = \left(-\frac{39}{35} + 17\right)e^{-t} + \frac{39}{35}e^{-6t} - 12e^{-2t}$$



SCILAB

$$--> \ t = [0{:}0.001{:}5];$$
$$--> \ i = \frac{556}{35} * \exp(-t) + \frac{39}{35} * \exp(-6*t) - 12 * \exp(-2*t);$$
$$--> \ is = 8 * \exp(-2*t);$$
$$--> \ plot(t, i);$$
$$--> \ plot(t, is, 'r')$$
$$--> \ plot([0,5], [0,0], 'bk');$$

Condições iniciais

$y(0^-) = 5A$

$y'(0^-) = \frac{10}{7}$

$$y'' = -7y' - 6y + 48\,e^{-2t}$$

$$\begin{cases} y' = g_1(t, y, u) = g_1(u) = u \\ u' = g_2(t, y, y) = -7u - 6y + 48\,e^{-2t} \end{cases}$$

$y(0^-) = 5\,A \qquad u(0) = y'(0^-) = 10/7$

```
--> deff('y=w(t)','y=8*exp(-2*t)');

--> deff('phi=g2(t,y,u)','phi=-7*u-6*y+48*exp(-2*t)');

--> t=linspace(0,5,1000);

--> plot(t,feval(t,w),'r')

--> [t,y]=Runge_Kutta4_EDO_2a_Ordem(g2,0,5,5,10/7,1000,%f);

--> plot(t,y,'b')

--> legend("Entrada w(t)","Saída - Runge-Kutta 2a ordem",4)
```