Ano Letivo: 2022/2023

Atividade 04-Métodos Numéricos para a resolução de SED

Relatório

Análise Matemática II

Docente: Arménio Correia

Trabalho realizado por:

Martim Alexandre Vieira Antunes

nº: 2022141890 **Curso:** LEI

Pedro Lino Neves Faneca

nº: 2022134142 **Curso:** LEI

Índice

1.Introdução

Este trabalho surge do âmbito da unidade curricular de Análise Matemática 2, do curso de Engenharia Informática do Instituto Superior de Engenharia de Coimbra.

O seu foco consiste na redefinição e adaptação das funções implementadas anteriormente na Atividade03 (Métodos Numéricos para a resolução de Equações Diferenciais Ordinárias (EDO) e Problemas de Valor Inicial (PVI)) para a resolução de Sistemas de Equações Diferenciais com condições iniciais.

Pretende-se também, desenvolver uma app em Matlab para resolver exemplos/exercícios (Pêndulo, Sistemas Mecânicos Massa-Mola com amortecimento e sem amortecimento, circuitos elétricos modelados por ED de 2ª ordem, entre outros), de modo a testar as funções implementadas.

2.Métodos Numéricos para a resolução de Sistemas de ED

Neste trabalho, iremos abordar os seguintes métodos numéricos para a resolução de SED:

- Método de Euler;
- Método de Euler Melhorado;
- Método de RK2;
- Método de RK4;

2.1-Método de Euler

O método de Euler é um dos métodos numéricos mais simples e amplamente utilizados para resolver sistemas de equações diferenciais. No entanto, é importante lembrar que ele tem limitações em relação à precisão e pode não ser a melhor escolha em todos os casos.

2.1.1-Fórmulas

Fórmula Geral para EDs de 1ªOrdem

$$\boxed{y_{i+1}= y_i+h*f(t_i,y_i) \text{ ,}i=0,1,\ldots,n-1}$$

Legenda:

- y_{i+1} = Próximo valor aproximado da solução do problema original (na abcissa *ti+1*);
- y_i = Valor aproximado da solução do problema original na abcissa atual;
- h = Valor de cada subintervalo (passo);
- $f(t_i ,y_i)$ = Valor da equação em t_i e y_i;

Fórmula Geral alterada para um Sistemas de Equações Diferenciais

$$u_{i+1} = u_i + h*f(t_i, u_i, v_i) \ ,i=0,1,\ldots,n-1$$

$$v_{i+1} = v_i + h*g(t_i,u_i,v_i) \ ,i=0,1,\ldots,n-1$$

Legenda:

- u_{i+1} = Próxima ordenada da solução aproximada *y(t)*;
- v_{i+1} = Próxima ordenada da solução aproximada *y'(t)*
- u_i = Ordenada atual da solução aproximada *y(t)*;
- v_i = Ordenada da solução aproximada *y'(t)*;
- *h* = Valor de cada subintervalo (passo);
- $f(t_i,u_i,v_i)$ à Valor de *f* no ponto (t_i , u_i, v_i);
- $g(t_i,u_i,v_i)$ à Valor de *g* no ponto (t_i , u_i, v_i).

2.1.2-Algoritmo/Função

Algoritmo:

- Ler f, g, a, b, n, u0, v0;
- Calcular o passo h (h =(b-a)/n);
- t = a:h:b;
- Criar os vetores u e v;
- u(1) = u0 e v(1)=v0;
- Para i de 1 até n, fazer o cálculo o cálculo do Método de Euler para os vetores u e v;

Função (MATLAB):

```
function [t,u,v] = EulerSED(f,g,a,b,n,u0,v0)
h = (b-a)/n;                        % Tamanho de cada subintervalo (passo)
t = a:h:b;                          % Alocação de memória
u = zeros(1,n+1);                   % Alocação de memória
v = zeros(1,n+1);                   % Alocação de memória
u(1) = u0;                          % O primeiro valor de u é sempre u0
v(1) = v0;                          % O primeiro valor de v é sempre v0
    for i = 1:n                     % O número de iterações vai ser igual a n
        u(i+1) = u(i)+h*f(t(i),u(i),v(i));  % Aproximação do método de Euler para
                                            % a iésima iteração
        v(i+1) = v(i)+h*g(t(i),u(i),v(i));  % Aproximação do método de Euler para
                                            % a iésima iteração
    end
end
```

2.2-Método de Euler Melhorado

Este método também se pode referir como Método de Euler Moderado ou Método de Heun. Ele é uma melhoria do método de Euler simples, que é mais simples de implementar, mas pode produzir soluções menos precisas.

2.2.1-Fórmulas

Fórmula Geral para EDs de 1ª Ordem

$$\boxed{y_{i+1} = y_i + h/2*(k1+k2)\ ,i=0,1,\ldots,n-1}$$

Legenda:

- y_{i+1} = Próximo valor de y(valor aproximado da solução ao problema) na abcissa *ti+1* ;
- y_i = Valor aproximado da solução do problema na abcissa atual(abcissa *ti)*;
- h = Valor de cada subintervalo (passo);
- k_1 = Inclinação no início do intervalo;
- k_2 = Inclinação no fim do intervalo;

Fórmula Geral alterada para um Sistemas de Equações Diferenciais

$$\boxed{u_{i+1} = u_i + h/2*(k1u+k2u)\ ,i=0,1,\ldots,n-1}$$

$$\boxed{v_{i+1} = v_i + h/2*(k1v+k2v)\ ,i=0,1,\ldots,n-1}$$

Legenda:

- u_{i+1} = Aproximação do método de Heun para a iésima iteração;
- v_{i+1} = Aproximação do método de Heun para a iésima iteração;
- u_i = Ordenada atual da função aproximada *y(t)*;
- v_i = Ordenada atual da função aproximada *y'(t)*;
- h = Valor de cada subintervalo (passo);
- k1u = Inclinação no início do intervalo;
- k2u = Inclinação no fim do intervalo;
- k1v = Inclinação no inicio do intervalo;
- k2v=Inclinação no fim do intervalo;

Fórmula para calcular k1u

$$\boxed{K1u = f(t_i, u_i, v_i)}$$

Legenda:

- $k1u$ = Inclinação no início do intervalo;
- $f(t_i,u_i,v_i)$ = Valor de f no ponto (t_i,u_i,v_i).

Fórmula para calcular k1v

$$\boxed{K1v = g(t_i, u_i, v_i)}$$

Legenda:

- $k1v$ = Inclinação no início do intervalo;
- $g(t_i, u_i, v_i)$ = Valor de g no ponto (t_i, u_i, v_i).

Fórmula para calcular k2u

$$\boxed{K2u = f(t_{i+1}, u_i + k1u * h, v_i + k1v * h)}$$

Legenda:

- $k2u$ = Inclinação no fim do intervalo;
- t_{i+1} = Próxima abcissa do intervalo escolhido;
- $k1u$ = Inclinação no início do intervalo;
- h = Tamanho de cada subintervalo (passo);
- u_i = Ordenada atual da solução aproximada $y(t)$;
- v_i = Ordenada atual da solução aproximada $y'(t)$;
- $k1v$ = Inclinação no início do intervalo.

Fórmula para calcular k2v

$$\boxed{K2v = g(t_{i+1}, u_i + k1u * h, v_i + k1v * h)}$$

Legenda:

- $k2v$ = Inclinação no fim do intervalo;
- t_{i+1} = Próxima abcissa do intervalo escolhido;
- $k1u$ = Inclinação no início do intervalo;
- h = Tamanho de cada subintervalo (passo);
- u_i = Ordenada atual da solução aproximada $y(t)$;
- v_i = Ordenada atual da solução aproximada $y'(t)$;
- $k1v$ = Inclinação no início do intervalo.

2.2.2- Algoritmo/Função

Algoritmo:

- Ler f, g, a, b, n, u0, v0;
- Calcular h (h =(b-a)/n);
- t = a:h:b;
- Criar os vetores u e v;
- u(1) = u0 e v(1)=v0;
- Calcular k1u, k1v, k2u,k2v;
- Para i de 1 até n, fazer o cálculo do Método de Euler Melhorado para os vetores u e v;

Função (MATLAB):

```
function [t,u,v] = EulerMSED(f,g,a,b,n,u0,v0)

h = (b-a)/n;                          % Tamanho de cada subintervalo (passo)
t = a:h:b;                            % Alocação de memória
u = zeros(1,n+1);                     % Alocação de memória
v = zeros(1,n+1);                     % Alocação de memória
u(1) = u0;                            % O primeiro valor de u é sempre u0
v(1) = v0;                            % O primeiro valor de v é sempre v0
    for i = 1:n                       % O número de iterações vai ser igual a n

        k1u = f(t(i),u(i),v(i));            % Inclinação no início do intervalo
        k1v = g(t(i),u(i),v(i));            % Inclinação no início do intervalo

        k2u = f(t(i+1), u(i) + k1u*h, v(i) + k1v*h); % Inclinação no fim do intervalo
        k2v = g(t(i+1), u(i) + k1u*h, v(i) + k1v*h); % Inclinação no fim do intervalo

        u(i+1) = u(i)+h/2*(k1u+k2u);   % Aproximação do método de
                                       % Euler Melhorado para a iésima iteração
        v(i+1) = v(i)+h/2*(k1v+k2v);   % Aproximação do método de
                                       % Euler Melhorado para a iésima iteração
    end
end
```

2.3-Método de Runge-Kutta de 2ªOrdem

O método de Runge-Kutta de 2ª ordem é um algoritmo numérico utilizado para resolver sistemas de equações de primeira ordem. Tem uma precisão maior que o método de Euler.

2.3.1-Fórmulas

Fórmula Geral para EDs de 1ªOrdem

$$y_{i+1} = y_i + \frac{1}{2}*(k1+k2) \text{ ,} i=0,1,\ldots,n-1$$

Legenda:

- y_{i+1} = Próximo valor de y(valor aproximado da solução ao problema) na abcissa *ti+1* ;
- y_i = Valor aproximado da solução do problema na abcissa atual(abcissa *ti)*;
- k_1 = Inclinação no início do intervalo;
- k_2 = Inclinação no fim do intervalo;

Fórmula Geral alterada para um Sistema de Equações

$$u_{i+1} = u_i + (k1u+k2u)/2 \text{ ,} i=0,1,\ldots,n-1$$

$$v_{i+1} = v_i + (k1v+k2v)/2 \text{ ,} i=0,1,\ldots,n-1$$

Legenda:

- u_{i+1} = Aproximação do método de RK2 para a iésima iteração;
- v_{i+1} = Aproximação do método de RK2 para a iésima iteração;
- u_i = Ordenada atual da função aproximada *y(t)*;
- v_i = Ordenada atual da função aproximada *y'(t)*;
- k1u = Inclinação no início do intervalo;
- k2u = Inclinação no fim do intervalo;
- k1v = Inclinação no inicio do intervalo;
- k2v=Inclinação no fim do intervalo;

Fórmula para calcular k1u

$$\boxed{K1u = h * f(t_i, u_i, v_i)}$$

Legenda:

- *k1u* = Inclinação no início do intervalo;
- $f(t_i,u_i,v_i)$ = Valor de *f* no ponto (t_i,u_i,v_i).

Fórmula para calcular k1v

$$\boxed{K1v = h * g(t_i, u_i, v_i)}$$

Legenda:

- *k1v* = Inclinação no início do intervalo;
- $g(t_i, u_i, v_i)$ = Valor de *g* no ponto (t_i, u_i, v_i).

Fórmula para calcular k2u

$$\boxed{K2u = h * f(t_{i+1}, u_i + k1u, v_i + k1v)}$$

Legenda:

- *k2u* = Inclinação no fim do intervalo;
- t_{i+1} = Próxima abcissa do intervalo escolhido;
- *k1u* = Inclinação no início do intervalo;
- *h* = Tamanho de cada subintervalo (passo);
- u_i = Ordenada atual da solução aproximada *y(t)*;
- v_i = Ordenada atual da solução aproximada *y'(t)*;
- *k1v* = Inclinação no início do intervalo.

Fórmula para calcular k2v

$$\boxed{K2v = h * g(t_{i+1}, u_i + k1u, v_i + k1v)}$$

Legenda:

- *k2v* = Inclinação no fim do intervalo;
- t_{i+1} = Próxima abcissa do intervalo escolhido;
- *k1u* = Inclinação no início do intervalo;
- *h* = Tamanho de cada subintervalo (passo);
- u_i = Ordenada atual da solução aproximada *y(t)*;
- v_i = Ordenada atual da solução aproximada *y'(t)*;
- *k1v* = Inclinação no início do intervalo.

2.3.2-Algoritmo/Função

Algoritmo:

- Ler f, g, a, b, n, u0, v0;
- Calcular h (h =(b-a)/n);
- t= a:h:b;
- Criar os vetores u e v;
- u(1) = u0 e v(1)=v0;
- Calcular k1u, k1v, k2u,k2v;
- Para i de 1 até n, fazer o cálculo o cálculo do Método de Runge-Kutta 2ª Ordem para os vetores u e v;

Função (MATLAB):

```
function [t,u,v] = RK2SED(f,g,a,b,n,u0,v0)

h = (b-a)/n;                       % Tamanho de cada subintervalo (passo)
t = a:h:b;                         % Alocação de memória
u = zeros(1,n+1);                  % Alocação de memória
v = zeros(1,n+1);                  % Alocação de memória
u(1) = u0;                         % O primeiro valor de u é sempre u0
v(1) = v0;                         % O primeiro valor de v é sempre v0
    for i = 1:n                    % O número de iterações vai ser igual a n

        k1u = h*f(t(i),u(i),v(i));        % Inclinação no início do intervalo
        k1v = h*g(t(i),u(i),v(i));        % Inclinação no início do intervalo

        k2u = h*f(t(i+1),u(i)+k1u,v(i)+k1v);% Inclinação no fim do intervalo
        k2v = h*g(t(i+1),u(i)+k1u,v(i)+k1v); % Inclinação no fim do intervalo

        u(i+1) = u(i)+(k1u+k2u)/2;          %Aproximação do método de RK2 para
                                            % a iésima iteração
        v(i+1) = v(i)+(k1v+k2v)/2;          %Aproximação do método de RK2 para
                                            % a iésima iteração

    end

end
```

2.4-Método de Runge-Kutta de 4ª Ordem

O método de Runge-Kutta de 4ª ordem é um algoritmo numérico utilizado para resolver sistemas de equações diferenciais. O método de Runge-Kutta de 4ª ordem é mais preciso do que o método de Runge-Kutta de 2ª ordem.

2.4.1-Fórmulas

Fórmula Geral para EDs de 1ª Ordem

$$y_{i+1} = y_i + \frac{1}{6}*(k1+2*k2+2*k3+k4) \quad ,i=0,1,\ldots,n-1$$

Legenda:

- y_{i+1} = Próximo valor de y(valor aproximado da solução ao problema) na abcissa *ti+1* ;
- y_i = Valor aproximado da solução do problema na abcissa atual(abcissa *ti)*;
- k_1 = Inclinação no início do intervalo;
- k_2 = Inclinação no ponto médio do intervalo;
- k3 = Inclinação no ponto médio do intervalo;
- k4 = Inclinação no final do intervalo

Fórmula alterada para um Sistema de Equações:

$$u_{i+1} = u_i + 1/6*(k1u+2*k2u+2*k3u+k4u) \quad ,i=0,1,\ldots,n-1$$

$$v_{i+1} = v_i + 1/6*(k1v+2*k2v+2*k3v+k4v) \quad ,i=0,1,\ldots,n-1$$

Legenda:

- u_{i+1} = Aproximação do método de RK4 para a iésima iteração;
- v_{i+1} = Aproximação do método de RK4 para a iésima iteração;
- u_i = Ordenada atual da função aproximada *y(t)*;
- v_i = Ordenada atual da função aproximada *y'(t)*;
- k1u = Inclinação no início do intervalo;
- k2u = Inclinação no ponto médio do intervalo;
- k3u = Inclinação (novamente) no ponto médio do intervalo;
- k4u = Inclinação no fim do intervalo;
- k1v = Inclinação no inicio do intervalo;
- k2v = Inclinação no ponto médio do intervalo;
- k3v = Inclinação (novamente) no ponto médio do intervalo;
- k4u = Inclinação no fim do intervalo;

Fórmula para calcular k1u

$$K1u = h^* f(t_i,u_i,v_i)$$

Legenda:

- *k1u* = Inclinação no início do intervalo;
- $f(t_i,u_i,v_i)$ = Valor de *f* no ponto (t_i,u_i,v_i).

Fórmula para calcular k1v

$$K1v = h^*g(t_i,u_i,v_i)$$

Legenda:

- *k1v* = Inclinação no início do intervalo;
- $g(t_i,u_i,v_i)$ = Valor de *g* no ponto (t_i, u_i, v_i).

Fórmula para calcular k2u

$$K2u = h^*f(t_i+h/2,u_i+k1u/2,v_i+k1v/2)$$

Legenda:

- *k2u* = Inclinação no ponto médio do intervalo;
- t_i = Abcissa do intervalo escolhido;
- *k1u* = Inclinação no início do intervalo;
- *h* = Tamanho de cada subintervalo (passo);
- u_i = Ordenada atual da solução aproximada *y(t)*;
- v_i = Ordenada atual da solução aproximada *y’(t)*;
- *k1v* = Inclinação no início do intervalo.

Fórmula para calcular k2v

$$K2v = h^*g(t_i+h/2,u_i+k1u/2,v_i+k1v/2)$$

Legenda:

- *k2v* = Inclinação no ponto médio do intervalo;
- t_{i+1} = Próxima abcissa do intervalo escolhido;
- *k1u* = Inclinação no início do intervalo;
- *h* = Tamanho de cada subintervalo (passo);
- u_i = Ordenada atual da solução aproximada *y(t)*;
- v_i = Ordenada atual da solução aproximada *y’(t)*;
- *k1v* = Inclinação no início do intervalo.

Fórmula para calcular k3u

$$\boxed{K3u = h * f(t_i + h/2, u_i + k2u/2, v_i + k2v/2)}$$

Legenda:

- *k3u* = Inclinação no ponto médio do intervalo;
- t_i = Abcissa do intervalo escolhido;
- *k2u* = Inclinação no ponto médio do intervalo;
- *h* = Tamanho de cada subintervalo (passo);
- u_i = Ordenada atual da solução aproximada *y(t)*;
- v_i = Ordenada atual da solução aproximada *y'(t)*;
- *k2v* = Inclinação no ponto médio do intervalo.

Fórmula para calcular k3v

$$\boxed{K3v = h * g(t_i + h/2, u_i + k2u/2, v_i + k2v/2)}$$

Legenda:

- *k3v* = Inclinação no ponto médio do intervalo;
- t_i = Abcissa do intervalo escolhido;
- *k2u* = Inclinação no ponto médio do intervalo;
- *h* = Tamanho de cada subintervalo (passo);
- u_i = Ordenada atual da solução aproximada *y(t)*;
- v_i = Ordenada atual da solução aproximada *y'(t)*;
- *k2v* = Inclinação no ponto médio do intervalo.

Fórmula para calcular k4u

$$\boxed{K4u = h * f(t_{i+1}, u_i + k3u, v_i + k3v)}$$

Legenda:

- *k4u* = Inclinação no fim do intervalo;
- t_{i+1} = Próxima abcissa do intervalo escolhido;
- *k3u* = Inclinação no ponto médio do intervalo;
- *h* = Tamanho de cada subintervalo (passo);
- u_i = Ordenada atual da solução aproximada *y(t)*;
- v_i = Ordenada atual da solução aproximada *y'(t)*;
- *k3v* = Inclinação no ponto médio do intervalo.

Fórmula para calcular k4v

$$K4v = g*f(t_{i+1},u_i+k3u,v_i+k3v)$$

Legenda:

- *k4v* = Inclinação no fim do intervalo;
- t_{i+1} = Próxima abcissa do intervalo escolhido;
- *k3u* = Inclinação no ponto médio do intervalo;
- *h* = Tamanho de cada subintervalo (passo);
- u_i = Ordenada atual da solução aproximada *y(t)*;
- v_i = Ordenada atual da solução aproximada *y'(t)*;
- *k3v* = Inclinação no ponto médio do intervalo.

2.4.2-Algoritmo/Função

Algoritmo:

- Ler f, g, a, b, n, u0, v0;
- Calcular h (h =(b-a)/n);
- t= a:h:b;
- Criar os vetores u e v;
- u(1) = u0 e v(1)=v0;
- Calcular k1u, k1v, k2u,k2v,k3u,k3v,k4u,k4v;
- Para i de 1 até n, fazer o cálculo o cálculo do Método de Runge-Kutta 4ª Ordem para os vetores u e v;

Função (MATLAB):

```
function [t,u,v] = RK4SED(f,g,a,b,n,u0,v0)

h = (b-a)/n;                       % Tamanho de cada subintervalo (passo)
t = a:h:b;                         % Alocação de memória
u = zeros(1,n+1);                  % Alocação de memória
v = zeros(1,n+1);                  % Alocação de memória
u(1) = u0;                         % O primeiro valor de u é sempre u0
v(1) = v0;                         % O primeiro valor de v é sempre v0
    for i = 1:n                    % O número de iterações vai ser igual a n

      k1u = h*f(t(i), u(i), v(i));    % Inclinação no início do intervalo
      k1v = h*g(t(i), u(i), v(i));    % Inclinação no início do intervalo

      k2u = h*f(t(i) + h/2, u(i) + k1u/2, v(i) + k1v/2);       % Inclinação no ponto
                                                                % médio do intervalo
      k2v = h*g(t(i) + h/2, u(i) + k1u/2, v(i) + k1v/2);       % Inclinação no
                                                                % ponto médio do intervalo

      k3u = h*f(t(i) + h/2, u(i) + k2u/2, v(i) + k2v/2);      % Inclinação (novamente)
                                                               % no ponto médio do intervalo
      k3v = h*g(t(i) + h/2, u(i) + k2u/2, v(i) + k2v/2);      % Inclinação (novamente)
                                                               % no ponto médio do intervalo
      k4u = h*f(t(i+1), u(i) + k3u, v(i) + k3v);              % Inclinação no final do intervalo
      k4v = h*g(t(i+1), u(i) + k3u, v(i) + k3v);              % Inclinação no final do intervalo

      u(i+1) = u(i)+1/6*(k1u+2*k2u+2*k3u+k4u);   % Aproximação do método
                                                  % de RK4 para a iésima iteração

      v(i+1) = v(i)+1/6*(k1v+2*k2v+2*k3v+k4v);   % Aproximação do método
                                                  % de RK4 para a iésima iteração
    end
end
```

3.Exemplos de aplicação e teste dos métodos

3.1-Algoritmo

A resolução e aplicação dos diferentes exercícios na nossa app segue sempre os mesmos 5 passos.

Iremos sempre partir de uma equação do tipo Ay''+By'+Cy+D=0,onde A,B,C,D, y são funções de t. O objetivo é obter ,y(t), solução da ED.

Algoritmo de resolução

1º passo: Resolver em ordem a y'' a equação dada:

$$y'' = -\frac{B}{A}y' - \frac{C}{A}y' - \frac{D}{A}$$

2º passo: Mudar de variáveis:

$$\begin{cases} u = y \\ v = y' \end{cases}$$

3º passo: Derivar u e v e efetuar as devidas substituições:

$$\begin{cases} u' = y' \\ v' = y'' \end{cases} \Leftrightarrow \begin{cases} u' = v \\ v' = -\frac{B}{A}y - \frac{C}{A}y - \frac{D}{A} \end{cases}$$

4º passo: Definir os PVIs e o sistema de EDs

$$\begin{cases} \begin{cases} u' = v \\ v' = -\frac{B}{A}v - \frac{C}{A}u - \frac{D}{A} \end{cases} \\ t \in [a,b] \\ \begin{cases} u(0) = u_0 \\ v(0) = v_0 \end{cases} \end{cases}$$

5º passo: Aplicar Métodos Numéricos na APP, com $f(t,u,v) = u'$ e $g(t,u,v) = v'$, de modo a obter uma aproximação de $y(t)$.

3.2-Problema do Pêndulo

Example 13-A Motion of a Nonlinear Pendulum

The motion of a pendulum of length L subject to damping can be described by the angular displacement of the pendulum from the vertical, θ, as a function of time. (See Fig. 13.1.) If we let m be the mass of the pendulum, g the gravitational constant, and c the damping coefficient (i.e., the damping force is $F = -c\theta'$), then the ODE initial-value problem describing this motion is

$$\theta'' + \frac{c}{mL}\theta' + \frac{g}{L}\sin\theta = 0.$$

The initial conditions give the angular displacement and velocity at time zero; for example, if $\theta(0) = a$ and $\theta'(0) = 0$, the pendulum has an initial displacement, but is released with 0 initial velocity.

Analytic (closed-form) solutions rely on approximating $\sin\theta$; the exact solutions to this approximated system do not have the characteristics of the physical pendulum, namely, a decreasing amplitude and a decreasing period. (See Greenspan, 1974, for further discussion.)

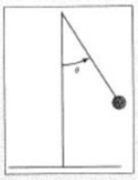


FIGURE 13.1a Simple pendulum.

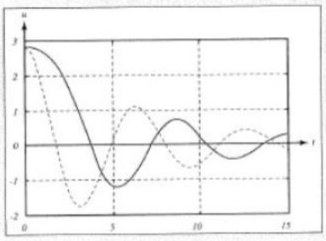


FIGURE 13.1b The motion of a pendulum given by ODE above (solid line) and linearized ODE (dashed line).

3.2.1-Modelação Matemática do problema

$$\boxed{\theta'' + \frac{c}{mL}\theta' + \frac{g}{L}\sin\theta = 0}$$

Exemplo: $\frac{g}{L} = 1$, $\frac{c}{mL} = 0.3$

$\theta = y$

$y'' + 0.3y' + \sin(y) = 0$

$y'' = -\sin(y) - 0.3y'$

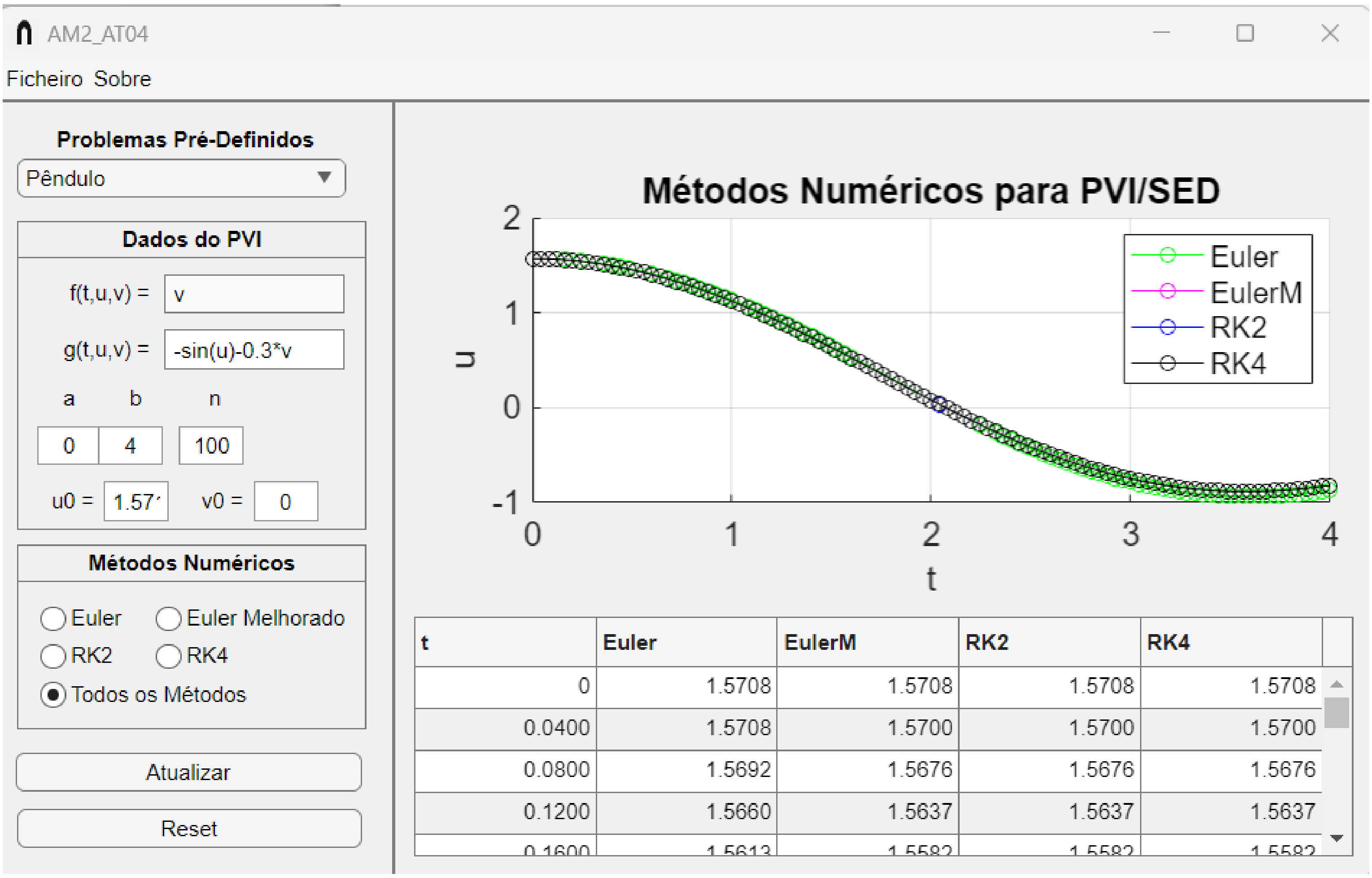
Mudansa variável

$$\begin{cases} u = y \\ v = y' \end{cases} \Leftrightarrow \begin{cases} u' = y' \\ v' = y'' \end{cases} \Leftrightarrow \begin{cases} u' = v = f(t,u,v) \\ v' = -\sin(u) - 0.3v = g(t,u,v) \end{cases}$$

Logo obtemos o seguinte sistema

$$\begin{cases} \begin{cases} u' = v \\ v' = -\sin(u) - 0.3v \end{cases} \\ t \in [0,4] \\ \begin{cases} u(0) = \frac{\pi}{2} \\ v(0) = 0 \end{cases} \end{cases}$$

3.2.2-Resolução através da App Desenvolvida



Nota: A Equação Diferencial do Problema do Pêndulo não é linear, logo não é possível calcular a solução exata através do MATLAB.

3.3-Modelo Vibratório Mecânico

Modelos vibratórios mecânicos

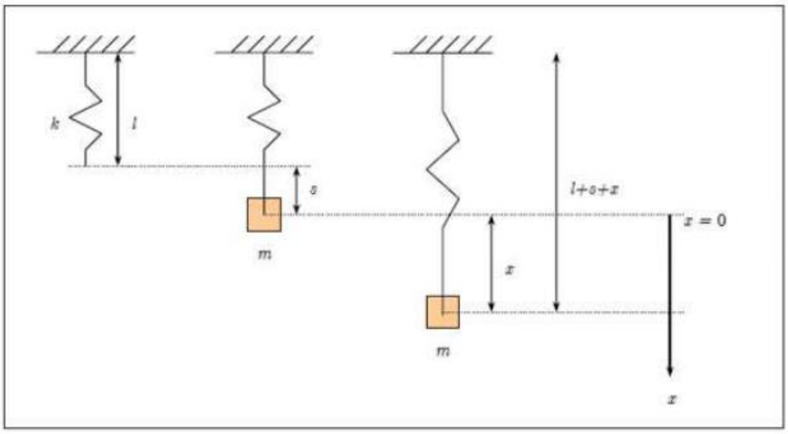
Nestes sistemas, o deslocamento x obedece à equação diferencial linear de 2ª ordem

$$mx'' + bx' + k(x) = f(t)$$

onde:

$m =$ massa; $x =$ deslocamento; $b =$ factor de amortecimento;

$k =$ constante da mola e $f(t) =$ força aplicada



2.

a) $x'' + 2x' + 2x = 4\cos t + 2\sin t, \quad x(0) = 0 \quad x'(0) = 3$

$\Rightarrow x(t) = e^{-t}\sin t + 2\sin t$

3.3.1-Modelação Matemática do problema

$x = y$

Substituindo, com os dados do enunciado

$$y'' + 2y' + 2y - 4\cos(t) - 2\sin(t) = 0$$

$$C.I: \begin{cases} y(0) = 0 \\ y'(0) = 3 \end{cases}$$

$$y'' = -2y' - 2y + 4\cos(t) + 2\sin(t)$$

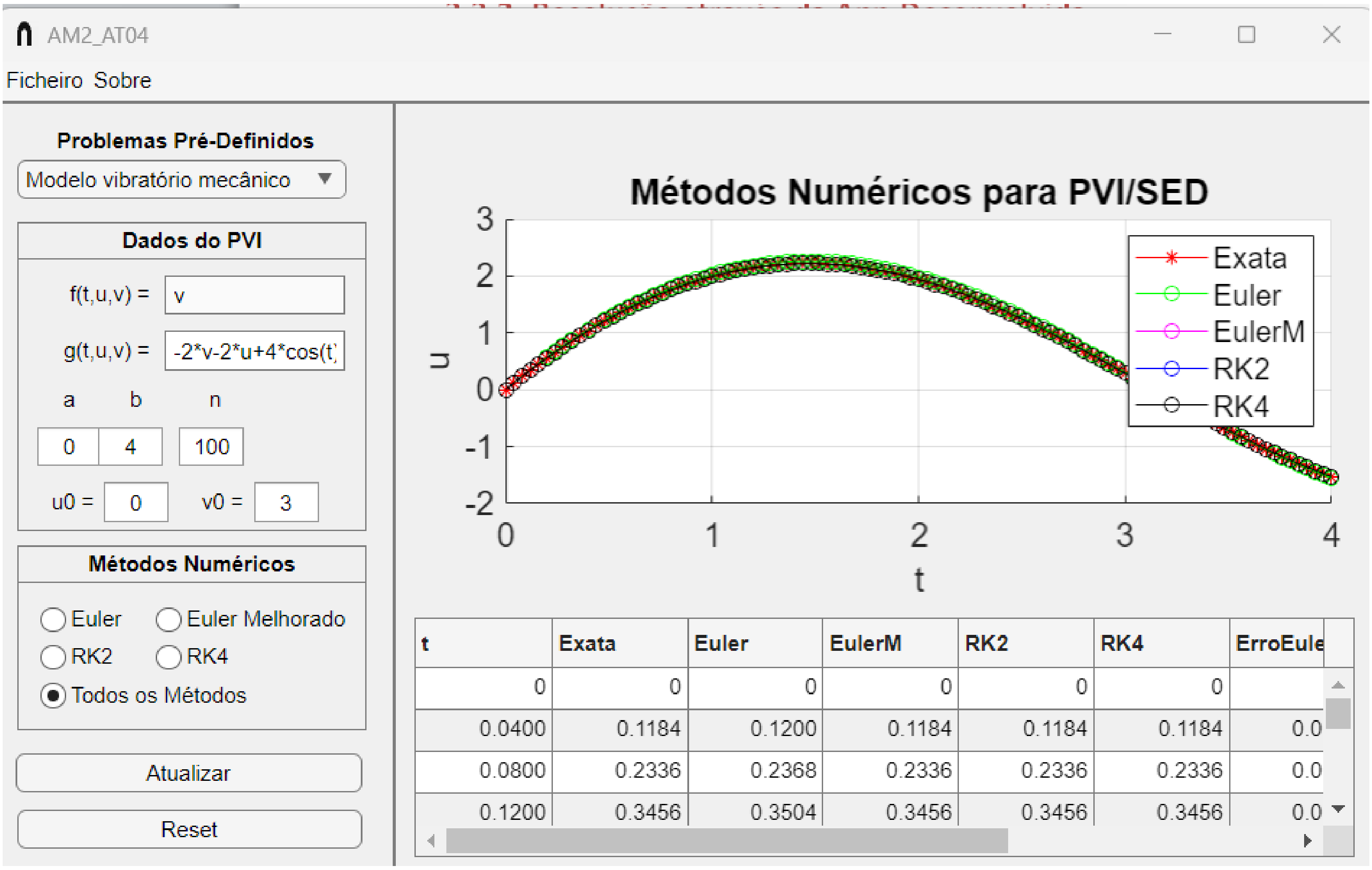
Mudança variável

$$\begin{cases} u = y \\ v = y' \end{cases} \Leftrightarrow \begin{cases} u' = y' \\ v' = y'' \end{cases} \Leftrightarrow \begin{cases} u' = v = f(t,u,v) \\ v' = \underbrace{-2v - 2u + 4\cos(t) + 2\sin(t)}_{g(t,u,v)} \end{cases}$$

Logo obtemos o seguinte sistema:

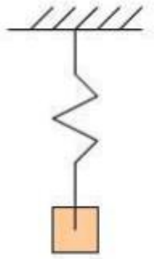
$$\begin{cases} \begin{cases} u' = v \\ v' = -2v - 2u + 4\cos(t) + 2\sin(t) \end{cases} \\ t \in [0,4] \\ \begin{cases} u(0) = 0 \\ v(0) = 3 \end{cases} \end{cases}$$

3.3.2- Resolução através da App Desenvolvida



3.4-Mola da massa sem amortecimento

b) A equação $mx'' + kx = 0$ descreve o movimento harmónico simples, ou movimento livre não amortecido, e está sujeita às condições iniciais $x(0) = a$ e $x'(0) = b$ representando, respectivamente, a medida do deslocamento inicial e a velocidade inicial.



Use este conhecimento para dar uma interpretação física do problema de *Cauchy*

$$x'' + 16x = 0 \quad x(0) = 9 \quad x'(0) = 0$$

e resolva-o

3.4.1-Modelação Matemática do problema

$x = y$

Substituindo, com os dados do enunciado

$$y'' + 16y = 0$$

$$C.I = \begin{cases} y(0) = 9 \\ y'(0) = 0 \end{cases}$$

$$\boxed{y'' = -16y}$$

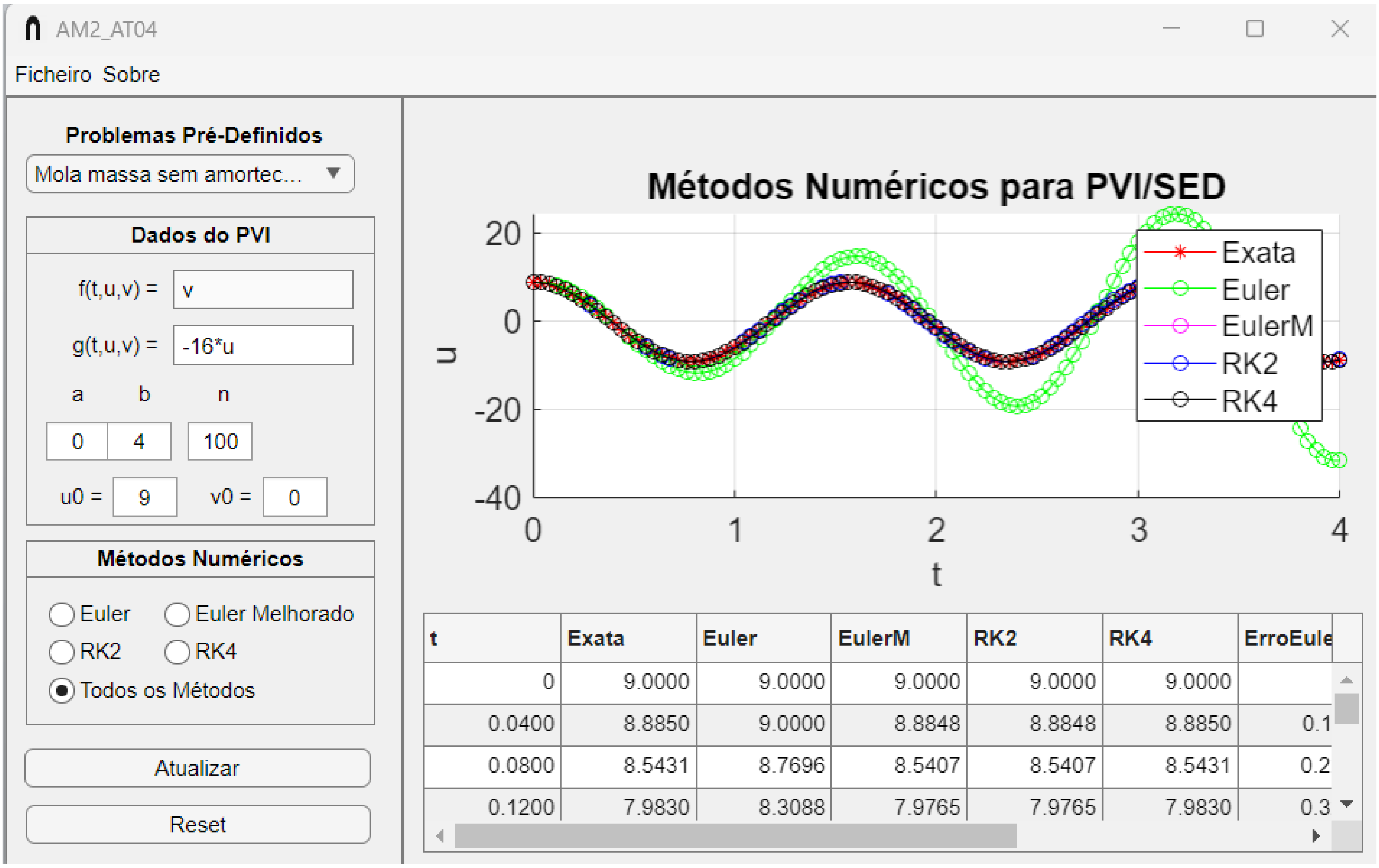
Mudansa variável

$$\begin{cases} u = y \\ v = y' \end{cases} \Leftrightarrow \begin{cases} u' = y' \\ v' = y'' \end{cases} \Leftrightarrow \begin{cases} u' = v = f(t, u, v) \\ v' = -16u = g(t, u, v) \end{cases}$$

Logo, obtemos o seguinte sistema

$$\begin{cases} \begin{cases} u' = v \\ v' = -16u \end{cases} \\ t \in [0, 4] \\ \begin{cases} u(0) = 9 \\ v(0) = 0 \end{cases} \end{cases}$$

3.4.2- Resolução através da App Desenvolvida



3.5-Mola da massa com amortecimento

c) Um peso de 6.4 *lb* provoca, numa mola, um alongamento de 1.28 *ft*. O sistema está sujeito à acção duma força amortecedora, numericamente igual ao dobro da sua velocidade instantânea. Determine a equação do movimento do peso, supondo que ele parte da posição de equilíbrio com uma velocidade dirigida para cima de 4 *ft/s*.

Resolução:

Sabe-se, pela lei de *Hooke*, que $W = ks$

No caso em estudo $k = \frac{6.4}{1.28} \Leftrightarrow k = 5$ lb/ft. Como $W = mg$, tem-se $m = \frac{6.4}{32} \Leftrightarrow m = 0.2$

A equação que descreve o movimento livre amortecido é

$$m\frac{d^2x}{dt^2} = -Kx - b\frac{dx}{dt}$$

onde b é uma constante positiva e o sinal "-" indica que as forças amortecedoras actuam na direcção oposta ao movimento.

Então a equação diferencial de movimento de peso é $0.2x'' = -5x - 2x'$

$\Leftrightarrow x'' + 10x' + 25x = 0$ com $x(0) = 0$ e $x'(0) = -4$

3.5.1-Modelação Matemática do problema

$x = y$

Substituindo, com os dados do enunciado

$$y'' + 10y' + 25y = 0$$

$$\text{C.I.: } \begin{cases} y(0) = 0 \\ y'(0) = -4 \end{cases}$$

$$\boxed{y'' = -10y' - 25y}$$

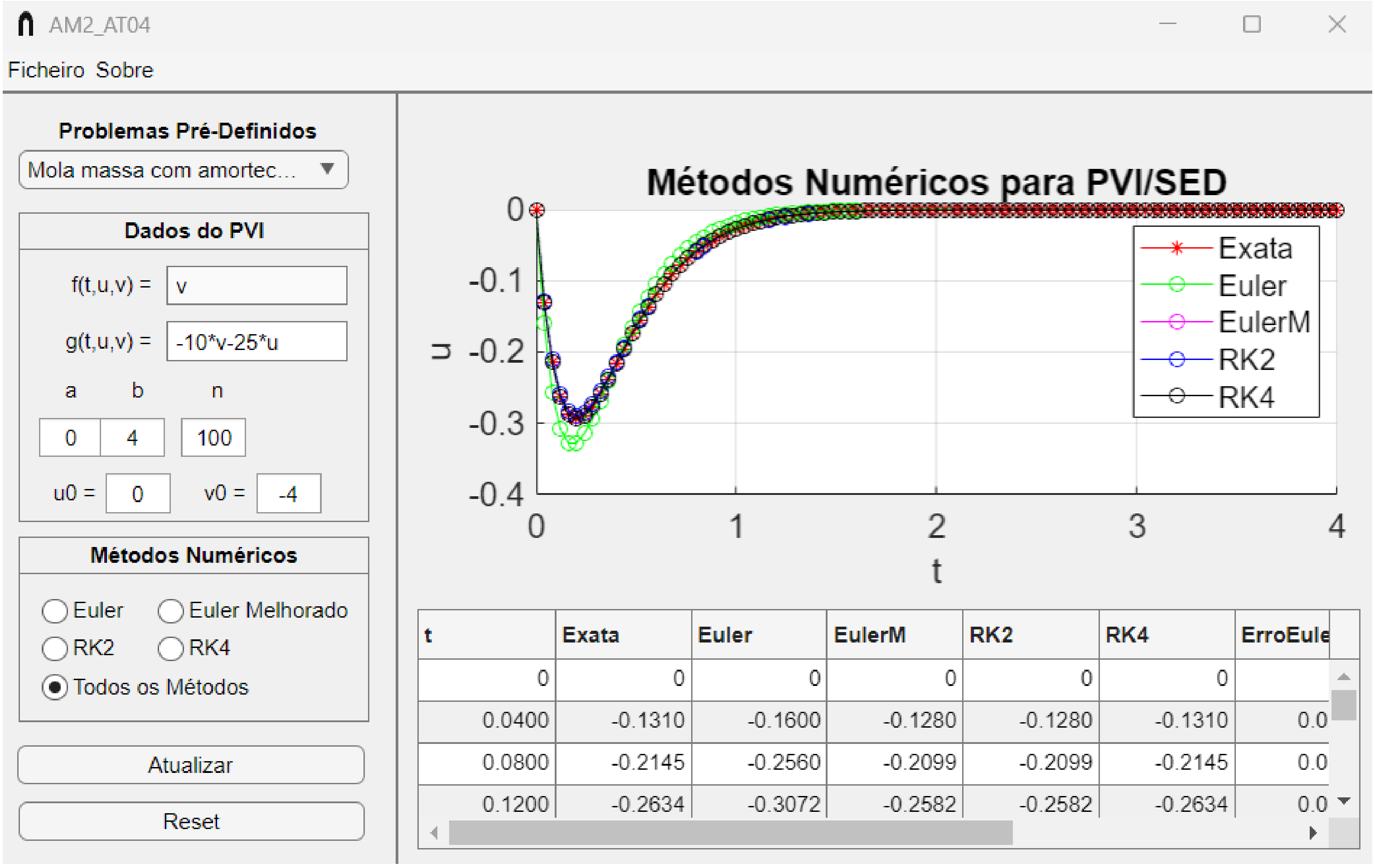
Mudança variável

$$\begin{cases} u = y \\ v = y' \end{cases} \Leftrightarrow \begin{cases} u' = y' \\ v' = y'' \end{cases} \Leftrightarrow \begin{cases} u' = v = f(t, u, v) \\ v' = -10v - 25u \end{cases}$$

Logo obtemos o seguinte sistema

$$\begin{cases} \begin{cases} u' = v \\ v' = -10v - 25u \end{cases} \\ t \in [0, 4] \\ \begin{cases} u(0) = 0 \\ v(0) = -4 \end{cases} \end{cases}$$

3.5.2- Resolução através da App Desenvolvida



3.6-Circuitos elétricos em série

Circuito eléctrico em série

$Lq'' + rq' + \frac{1}{c}q = e(t)$ (*)

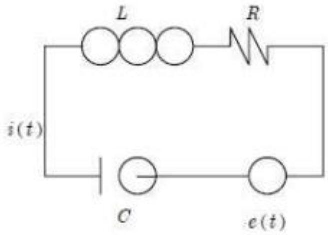
L – Indutância

q – carga

R – Resistência

C – capacidade

$e(t)$ – força electromotriz



Pelas leis de *Kirchoff*, num circuito indutivo-restritivo-capacitivo (L-R-C) série, em que a corrente varia com o tempo, a carga q acumulada no condensador é dada pela equação diferencial linear de 2ª Ordem. (*)

CIRCUITOS ELÉTRICOS

Uma circuito possui um capacitor de $0{,}5 \times 10^{-1}$ F, um resistor de $25\ \Omega$ e um indutor de $5\ H$, em série. O capacitor se encontra descarregado. No instante $t=0$ conecta-se esse circuito a uma bateria cuja tensão é de $10\ e^{-t/4}\ V$, e o circuito é fechado. Determine a carga no capacitor em qualquer inste $t>0$.

3.6.1-Modelação Matemática do problema

$$Lq'' + rq' + \frac{1}{C}q = \mathcal{E}(t)$$

A partir do enunciado, retiramos a seguinte informação:

$L = 5\ H$

$r = 25\ \Omega$

$C = 0{,}5 \times 10^{-1}\ F$

$\mathcal{E}(t) = 10\,e^{-\frac{1}{4}t}\ V$

$C.I = \begin{cases} y(0) = 0 \\ y'(0) = 0 \end{cases}$

$q = y$

Substituindo com os dados do enunciado

$$5y'' + 25y' + 20y - 10e^{-\frac{1}{4}t} = 0$$

$$y'' + 5y' + 4y - 2e^{-\frac{1}{4}t} = 0$$

$$\boxed{y'' = -5y' - 4y + 2e^{-\frac{1}{4}t}}$$

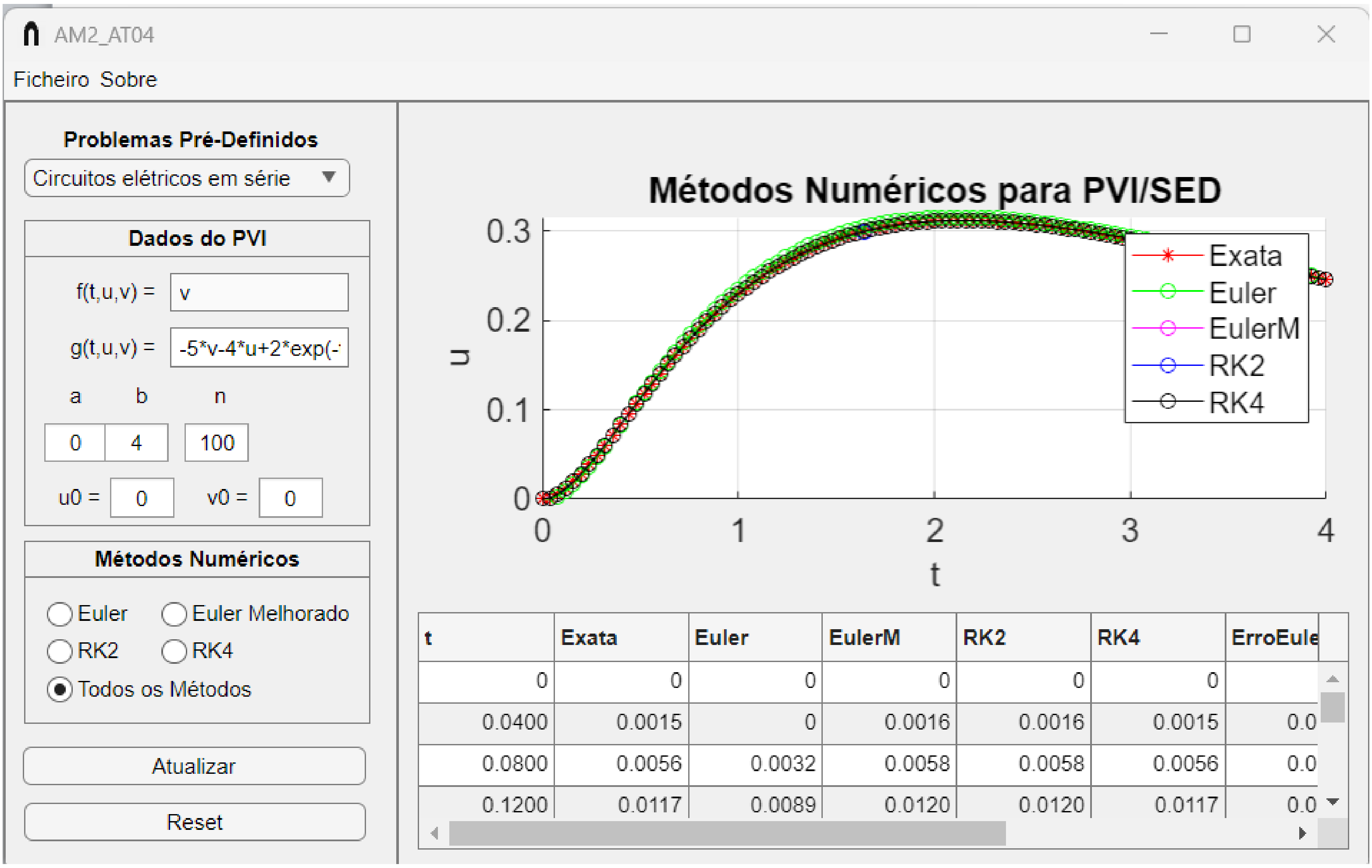
Mudança variável

$$\begin{cases} u = y \\ v = y' \end{cases} \Leftrightarrow \begin{cases} u' = y' \\ v' = y'' \end{cases} \Leftrightarrow \begin{cases} u' = v = f(t,u,v) \\ v' = -5v - 4u + 2e^{-\frac{1}{4}t} = g(t,u,v) \end{cases}$$

Logo obtemos o seguinte sistema

$$\begin{cases} \begin{cases} u' = v \\ v' = -5v - 4u + 2e^{-\frac{1}{4}t} \end{cases} \\ t \in [0, 4] \\ \begin{cases} u(0) = 0 \\ v(0) = 0 \end{cases} \end{cases}$$

3.6.2- Resolução através da App Desenvolvida



4.Conclusão

Concluindo, os mesmos métodos (Euler, Euler Melhorado,RK2,RK4) utilizados no trabalho anterior, com umas pequenas adaptações podem ser utilizados em Sistemas de Equações Diferenciais (SED).

Comparando os diferentes métodos, observamos que os que verificam menor erro e, consequentemente, melhor aproximação ao valor exato, é o método de Runge-Kutta de ordem 4 que muitas vezes apresentou erros muito pequenos (milésimas). Em contrapartida, temos o método de Euler, cujo erro é especialmente grande comparado com todos os outros métodos implementados.

De referir também, que equações diferenciais lineares de 2ªordem, como as que utilizámos neste trabalho, são possíveis de resolver problemas de diversas áreas como na Física, Engenharia, Biologia, Economia entre outras, como demonstramos no trabalho.

5.Bibliografia

- Ficheiros de suporte disponibilizados pelo professor;
- Formulário da cadeira;
- (232) EDO de Segunda Ordem - Circuitos Elétricos - YouTube;

6.Autoavalição e heteroavaliação

Chegando ao fim do trabalho, estamos contentes pelo resultado final.

Pelo esforço e trabalho aplicados a esta atividade, achamos que merecemos numa escala de 0 a 5 valores, um 4.5.

A nível de grupo, não houve quaisquer problemas e ambos trabalhámos bem. O aluno Martim Antunes foi quem distribui as tarefas, explicou como secalhar ficava melhor e quem preocupava-se de sempre colocar um "dedinho" dele no final, mesmo não lhe tivesse sido atribuído aquela tarefa, aperfeiçoar ainda mais.

Por isso achamos que o aluno Martim Antunes mereça um 5 e o aluno Pedro Faneca um 4.