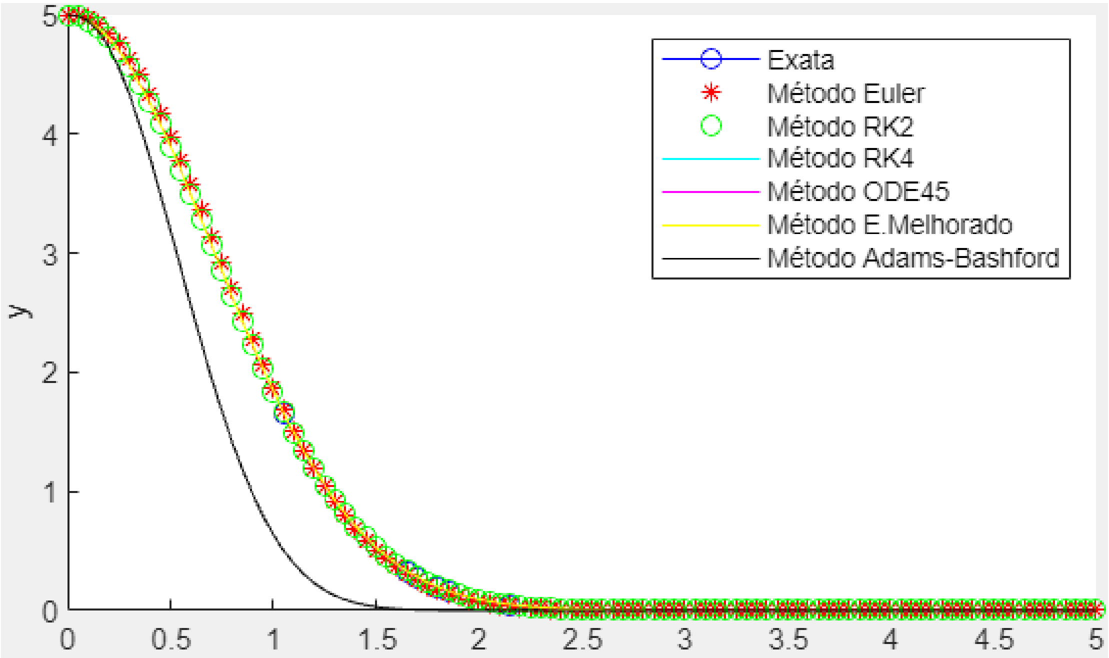
Métodos Numéricos para Equações Diferenciais Ordinárias / Problemas de Valor Inicial



Análise Matemática II – Atividade 03 Trabalho

Renato Alexandre Oliveira Craveiro | 2018011392

Ano Letivo 2022/2023 – ISEC – Lic. Engenharia Informática – Regime Pós-Laboral

Índice

1. Introdução

Este trabalho tem como objetivo aplicar Métodos de resolução/aproximação de Equações Diferenciais Ordinárias / Problemas de Valor Inicial através de *software* desenvolvido pelos alunos na plataforma MATLAB, por forma a aprofundar os conhecimentos acerca dos diversos métodos e de como desenvolvê-los em MATLAB.

1.1 Equação diferencial: Definição e propriedades

Uma equação diferencial é uma equação que envolve uma função desconhecida e uma ou mais das suas derivadas em relação a uma ou mais variáveis independentes. Uma equação diferencial ordinária (EDO) é um tipo de equação diferencial que envolve apenas uma variável independente e as derivadas de uma função desconhecida em relação a essa variável independente. As EDOs são usadas para modelar fenómenos físicos e naturais e existem métodos para resolvê-las analítica e numericamente.

As propriedades das equações diferenciais ordinárias dependem do tipo de equação. Algumas propriedades comuns incluem:

- Ordem: a ordem de uma EDO é a ordem mais alta das derivadas da função desconhecida que aparecem na equação. Por exemplo, uma EDO de primeira ordem envolve apenas a primeira derivada da função desconhecida, enquanto uma EDO de segunda ordem envolve a segunda derivada da função desconhecida.

- Linearidade: uma EDO é linear se a função desconhecida e suas derivadas aparecem apenas em termos lineares na equação. Caso contrário, a equação é não linear.

- Homogeneidade: uma EDO é homogênea se o termo constante for igual a zero. Caso contrário, a equação é não homogênea.

- Solução geral: a solução geral de uma EDO é uma função que satisfaz a equação para todos os valores da variável independente. É possível que existam várias soluções gerais para uma EDO.

- Condições iniciais: as condições iniciais são valores conhecidos da função desconhecida e suas derivadas em um ponto específico. A solução de uma EDO pode depender dessas condições iniciais.

1.2 Definição de PVI

Um problema de valor inicial (PVI) é um tipo de problema de equação diferencial ordinária (EDO) que consiste em encontrar uma solução para a EDO que satisfaça uma condição inicial especificada. Essa condição é dada em termos da função desconhecida e das suas derivadas num ponto específico da variável independente.

Um PVI é composto por uma EDO, uma condição inicial e um intervalo de interesse para a variável independente. O objetivo é encontrar a solução da EDO que satisfaça a condição inicial no ponto inicial e que esteja definida no intervalo de interesse.

Os problemas de valor inicial são muito importantes na matemática aplicada, especialmente na física e na engenharia, pois permitem modelar fenômenos dinâmicos que evoluem ao longo do tempo. Há muitas técnicas para resolver PVIs, incluindo métodos analíticos e numéricos, como o método de Euler e o método de Runge-Kutta.

2. Métodos Numéricos para resolução de PVI

2.1 Método de Euler

O método de Euler é um método simples para resolver problemas de valor inicial (PVI) de equações diferenciais ordinárias (EDO). Ele envolve dividir o intervalo de interesse em um número finito de subintervalos, estimar o valor da solução em cada subintervalo usando a EDO e a largura do subintervalo e repetir esse processo até chegar ao final do intervalo. Embora seja fácil de entender e implementar, ele nem sempre é o método mais preciso ou eficiente.

2.1.1 Fórmulas

$$y_{n+1} = y_n + h.f(y_n, t_n)$$

Onde:

- y_n é o valor aproximado da solução da EDO no ponto inicial (t_n)
- y_{n+1} é o valor aproximado da solução da EDO no ponto seguinte (t_{n+1})
- h é o tamanho do passo: a largura entre t_n até t_{n+1}
- $f(y_n,t_n)$ é a taxa de variação no ponto (y_n,t_n)

2.1.2 Algoritmo/Função

Dado um problema de valor inicial y' = f(t,y), y(t0) = y0 e um intervalo de tempo [t0, tf]:

1. Definir o ponto inicial (t_0, y_0) e o intervalo de interesse [t_0, t_f].
2. Dividir o intervalo [t_0, t_f] em N subintervalos de tamanho igual h = (t_f - t_0) / N.
3. Para cada subintervalo i = 0, 1, ..., N-1:
   a. Calcular o valor da taxa de variação da EDO no ponto (t_i, y_i) usando f(t_i, y_i).
   b. Utilizar a fórmula do método de Euler para calcular o valor aproximado da solução no ponto (t_{i+1}, y_{i+1}):
4. $y_{i+1} = y_i + h.f(t_i, y_i)$
5. Retornar os valores aproximados da solução em cada ponto do intervalo [t_0, t_f], ou seja, (t_0, y_0), (t_1, y_1), ..., (t_N, y_N).

Em MATLAB:

```
%NEULER  Método de Euler para ED/PVI.
%   y = NEuler(f,a,b,n,y0) Método numérico para a resolução de um PVI
%   y'= f(t,y) com t=[a, b] e y(a)=y0 condição inicial
%
%INPUT:
%   f - função do 2.º membro da Equação Diferencial
%   [a, b] - extremos do intervalo da variável independente t
%   n - número de subintervalos ou iterações do método
%   y0 - condição inicial t=a -> y=y0
%OUTPUT:
%   y - vector das soluções aproximações
%   y(i+1) = y(i)+h*f(t(i),y(i)) , i =0,1,...,n-1
%
%   03/03/2020 - ArménioCorreia .: armenioc@isec.pt

function y = NEuler(f,a,b,n,y0)
    h    = (b-a)/n;

    t = a:h:b;   %Modificado a 16/04/2023 por Renato Craveiro para alocação de memória
                 % e preenchimento de vetor t (abcissas) (de a a b com intervalos de tamanho h)
    y = zeros(1, n+1);  % Modificado a 16/04/2023 por Renato Craveiro para alocação de memória a zero
                        %criando um vetor limpo com tamanho n+1

    t(1) = a;
    y(1) = y0;
    for i=1:n
        y(i+1)=y(i)+h*f(t(i),y(i));
        t(i+1)=t(i)+h;
    end
end
```

2.2 Método de Euler Melhorado ou Modificado (Heun)

O método de Euler melhorado (também conhecido como método de Heun ou Euler modificado) é um método numérico para resolver problemas de valor inicial de equações diferenciais ordinárias. É uma variação do método de Euler que é mais preciso do que o método de Euler padrão, mas ainda é de primeira ordem.

Este método é uma melhoria do método de Euler, já que faz uma correção na inclinação da solução no ponto inicial antes de calcular a aproximação da solução no ponto final.

2.2.1 Fórmulas

$$y_{i+1} = y_i + \left(\frac{h}{2}\right).\left(f(t_i, y_i) + f\left(t_{i+h}, y_i + h.f(t_i, y_i)\right)\right)$$

Onde:

- y_i é a aproximação da solução no tempo t_i
- y_{i+1} é a aproximação da solução no tempo t_{i+h}
- $f(t_i, y_i)$ é a inclinação da solução no ponto (t_i, y_i)
- h é o tamanho do passo de tempo.

2.2.2 Algoritmo/Função

Dado um problema de valor inicial $y' = f(t,y)$, $y(t_0) = y_0$ e um intervalo de tempo $[t_0, t_f]$:

1. Definir o tamanho do passo de tempo h [$(t_f-t_0)/N$] (N: número de intervalos).
2. Definir a aproximação inicial $y_1 = y_0$.
3. Para i = 1 até i = N, onde N é o número de passos de tempo:
    a. Calcular a inclinação da solução no ponto (t_i, y_i) usando f(t_i, y_i).
    b. Faça uma predição da solução no ponto (t_{i+h}, y_{i+h}.f(t_i,y_i)) usando o método de Euler (indicado anteriormente).
    c. Calcular a inclinação da solução no ponto predito: f(t_{i+h}, y_{i+h}.f(t_i,y_i)).
    d. Calcular a média ponderada das duas inclinações: m = ½.(f(t_i,y_i) + f(t_i+h, y_i+h.f(t_i,y_i))).
    e. Calcular a aproximação da solução no tempo (t_{i+h}): $y_{i+1} = y_i + h.m$.

As etapas 3.c) até 3.e) correspondem à etapa de correção através de uma média ponderada das inclinações da solução no ponto inicial e no ponto predito. A solução é aproximada em cada passo de tempo usando uma média ponderada das inclinações da solução no ponto inicial e no ponto predito, o que torna o método de Euler melhorado mais preciso do que o método de Euler anterior.

Em MATLAB:

```
%NRK2  Método de Heun.
%   y = NHeun(f,a,b,n,y0) Método numérico para a resolução de um PVI
%   y'= f(t,y) com t=[a, b] e y(a)=y0 condição inicial
%
%INPUT:
%   f - função do 2.º membro da Equação Diferencial
%   [a, b] - extremos do intervalo da variável independente t
%   n - número de subintervalos ou iterações do método
%   y0 - condição inicial t=a -> y=y0
%OUTPUT:
%   y - vector das soluções aproximações
%
%   28/04/2023 - Renato Craveiro | 2018011392 | Lic. Eng. Informática ISEC
%   Ano Letivo 2022/23

function y = NHeun(f,a,b,n,y0)
    h = (b-a)/n;                    % Tamanho de cada subintervalo (passo)

    t = a:h:b;                      % Alocação de memória - vetor das abcissas
    y = zeros(1, n+1);              % Alocação de memória - vetor das ordenadas


    y(1) = y0;                      % O primeiro valor de y é sempre y0 (condição inicial do pvi)

    for i=1:n                       % O número de iterações vai ser igual a n
        k1 = f(t(i),y(i));          % Inclinação no início do intervalo
        k2 = f(t(i+1), y(i) + k1*h); % Inclinação no fim do intervalo
        k = 0.5*(k1+k2);            % Cálculo da média das inclinações
        y(i+1)=y(i)+h*k;            % Próximo valor aproximado da solução do problema original
    end
end
```

2.3 Método de RK2

O método de Runge-Kutta é um conjunto de algoritmos para resolver problemas de valor inicial (PVI) de equações diferenciais ordinárias (EDO). O método de Runge-Kutta de ordem n (RK-n) envolve o uso de n estimativas intermediárias da derivada da solução em pontos específicos, para calcular uma estimativa da solução no próximo ponto no tempo.

O método de RK2 é um método de segunda ordem que calcula a solução aproximada da EDO usando duas estimativas intermediárias da derivada.

2.3.1 Fórmulas

$$y_{n+1} = y_n + h.f(y_n + 0{,}5.h.f(y_n, t_n), t_n + 0{,}5.h)$$

Onde:

- y_n é o valor aproximado da solução da EDO no ponto inicial t_n.
- y_{n+1} é o valor aproximado da solução da EDO no próximo ponto t_{n+1}.
- h é o tamanho do passo de tempo, ou seja, a largura do intervalo de tempo entre t_n e t_{n+1}.
- $f(y_n, t_n)$ é a taxa de variação da solução da EDO no ponto (y_n, t_n).

2.3.2 Algoritmo/Função

Dado um problema de valor inicial $y' = f(t,y)$, $y(t_0) = y_0$ e um intervalo de tempo $[t_0, t_f]$:

1. Definir o tamanho do passo de tempo h.
2. Definir a aproximação inicial $y_1 = y_0$.
3. Para i = 1 até i = N, onde N é o número de passos de tempo:
   a. Calcular $k_1 = h.f(t_i, y_i)$.
   b. Calcular $k2 = h.f(t_i + h, y_i + k_1)$.
   c. Calcule $y_{i+1} = y_i + ½ . (k_1 + k_2)$.

Em MATLAB:

```
%NRK2  Método de Runge-Kutte de Ordem 2.
%   y = N_RK2(f,a,b,n,y0) Método numérico para a resolução de um PVI
%   y'= f(t,y) com t=[a, b] e y(a)=y0 condição inicial
%
%INPUT:
%   f - função do 2.º membro da Equação Diferencial
%   [a, b] - extremos do intervalo da variável independente t
%   n - número de subintervalos ou iterações do método
%   y0 - condição inicial t=a -> y=y0
%OUTPUT:
%   y - vector das soluções aproximações
%
%   16/04/2023 - Renato Craveiro | 2018011392 | Lic. Eng. Informática ISEC
%   Ano Letivo 2022/23

function y = NRK2(f,a,b,n,y0)
    h = (b-a)/n;                        % h = tamanho de cada passo (subintervalo)

    t = a:h:b;                          % Alocação de memória e preenchimento de vetor t (abcissas) (de a a b com intervalos de tamanho h)
    y = zeros(1, n+1);                  %Alocação de memória a zero criando um vetor limpo com tamanho n+1

    y(1) = y0;                          % Inserção da condição inicial no primeiro elemento do vetor y

    for i=1:n                           % n iterações
        k1 = h * f(t(i), y(i));         % Inclinação inicial do intervalo
        k2 = h * f(t(i + 1), y(i) + k1);    % Inclinação final do intervalo

        y(i + 1)=y(i) + (k1 + k2)/2;    % Aproximação do próximo valor da solução do problema original
    end
end
```

2.4 Método de RK4

O método de RK4 é o método de quarta ordem do método de Runge-Kutta, que calcula a solução aproximada da EDO usando quatro estimativas intermediárias da derivada.

2.4.1 Fórmulas

- $k_1 = h.f(y_n, t_n)$
- $k_2 = h.f(y_n + 0.5 . k_1, t_n + 0.5 . h)$
- $k_3 = h.f(y_n + 0.5 . k_2, t_n + 0.5 . h)$
- $k_4 = h.f(y_n + k_3, t_n + h)$
- $y_{n+1} = y_n + (1/6) . (k_1 + 2 . k_2 + 2 . k_3 + k_4)$

Onde:

- k_1, k_2, k_3 e k_4 são estimativas intermediárias da derivada da solução em pontos específicos.
- Os restantes são os mesmos utilizados no Runge-Kutta de ordem 2.

2.4.2 Algoritmo/Função

Dado um problema de valor inicial $y' = f(t,y)$, $y(t_0) = y_0$ e um intervalo de tempo $[t_0, t_f]$:

1. Definir o tamanho do passo de tempo h.
2. Definir a aproximação inicial $y_1 = y_0$.
3. Para i = 1 até i = N, onde N é o número de passos de tempo:
   a. Calcular $k_1 = h.f(t_i, y_i)$.
   b. Calcular $k_2 = h.f(t_i + ½ . h, y_i + ½ . k_1)$.
   c. Calcular $k_3 = h.f(ti + ½ . h, y_i + ½ . k_2)$.
   d. Calcular $k_4 = h.f(ti + h, yi + k_3)$.
   e. Calcular $y_{i+1} = yi + 1/6 . (k_1 + 2 . k_2 + 2 . k_3 + k_4)$.

Em MATLAB:

```matlab
%NRK2  Método de Runge-Kutte de Ordem 4.
%   y = N_RK2(f,a,b,n,y0) Método numérico para a resolução de um PVI
%   y'= f(t,y) com t=[a, b] e y(a)=y0 condição inicial
%
%INPUT:
%   f - função do 2.º membro da Equação Diferencial
%   [a, b] - extremos do intervalo da variável independente t
%   n - número de subintervalos ou iterações do método
%   y0 - condição inicial t=a -> y=y0
%OUTPUT:
%   y - vector das soluções aproximações
%
%   16/04/2023 - Renato Craveiro | 2018011392 | Lic. Eng. Informática ISEC
%   Ano Letivo 2022/23


function y = NRK4(f,a,b,n,y0)
    h = (b-a)/n;                                % h = tamanho de cada passo (subintervalo)

    t = a:h:b;                                  % Alocação de memória e preenchimento de vetor t (abcissas) (de a a b com intervalos de tamanho h)
    y = zeros(1, n+1);                          %Alocação de memória a zero criando um vetor limpo com tamanho n+1

    y(1) = y0;                                  % Inserção da condição inicial no primeiro elemento do vetor y

    for i=1:n                                   % n iterações
        k1 = h*f(t(i), y(i));                   % Inclinação inicial do intervalo
        k2 = h*f(t(i) + h/2, y(i) + 0.5*k1);    % 1ª Inclinação média (ponto médio) do intervalo
        k3 = h*f(t(i) + h/2, y(i) + 0.5*k2);    % 2ª Inclinação média (ponto médio) do intervalo
        k4 = h*f(t(i+1), y(i) + k3);            % Inclinação final do intervalo

        y(i + 1) = y(i) + (k1 + 2*k2 + 2*k3 + k4)/6;    % Aproximação do próximo valor da solução do problema original
    end
end
```

2.5 Função ODE45 do MATLAB

A função "ode45" do MATLAB é uma implementação do método Runge-Kutta-Fehlberg de quinta ordem (RK45) para resolver problemas de valor inicial (PVI) de equações diferenciais ordinárias (EDO).

Este é um algoritmo adaptativo que usa duas fórmulas diferentes (uma de quinta ordem e outra de quarta ordem) para estimar a solução em cada passo de tempo. Utiliza-se a diferença entre essas duas estimativas para avaliar a precisão da solução. Se a precisão desejada não for atingida, o tamanho do passo de tempo é ajustado para tentar atingir a precisão desejada. Este processo de ajuste do tamanho do passo é chamado de controlo de erro.

2.5.1 Fórmulas e Função:

$$[t,y] = \text{ode45}(\text{odefun},\text{tspan},y0)$$

Onde:

- *odefun* é uma função que descreve a EDO a ser resolvida. Esta função deve ter a seguinte forma: dydt = odefun(t,y), onde "t" é o tempo, "y" é a solução num determinado ponto no tempo e *dydt* é a taxa de variação da solução em relação ao tempo.
- *tspan* é um vetor que define o intervalo de tempo a ser resolvido. O primeiro elemento de *tspan* é o tempo inicial e o último elemento é o tempo final.
- *y0* é o valor inicial da solução da EDO em *tspan(1)*.

A função "ode45" retorna dois argumentos:

- *t* é um vetor que contém os pontos de tempo onde a solução foi avaliada.
- *y* é uma matriz que contém a solução da EDO em cada ponto de tempo em "t".

Em MATLAB:

```
%N_ODE45  Método ODE45 (Método disponibilizado pelo MATLAB).
%   y = N_ODE45(f,a,b,n,y0) Método numérico para a resolução de um PVI
%   y'= f(t,y) com t=[a, b] e y(a)=y0 condição inicial
%
%INPUT:
%   f - função do 2.º membro da Equação Diferencial
%   [a, b] - extremos do intervalo da variável independente t
%   n - número de subintervalos ou iterações do método
%   y0 - condição inicial t=a -> y=y0
%OUTPUT:
%   y - vector das soluções aproximações
%   y(i+1) = y(i)+h*f(t(i),y(i)) , i =0,1,...,n-1
%
%   16/04/2023 - Renato Craveiro | 2018011392 | Lic. Eng. Informática ISEC
%   Ano Letivo 2022/23

function y = N_ODE45(f,a,b,n,y0)
    h = (b-a)/n;                      % h = tamanho de cada passo (subintervalo)

    t = a:h:b;                        % Alocação de memória e preenchimento de vetor t (abcissas) (de a a b com intervalos de tamanho h)

    [~,y] = ode45(f, t, y0);          % Preenchimento do vetor y com aproximações utilizando a função ODE45
    y = y';                           % Mudança da orientação do vetor
end
```

2.6 Método Adams-Bashford (2 passos)

O método de Adams-Bashford é um método numérico para a resolução de equações diferenciais ordinárias (EDOs). É um método de passo múltiplo, ou seja, usa informações de passos anteriores para obter uma solução para o próximo passo.

Para começar o método, precisamos de valores iniciais para y_0 e y_1. Normalmente, estes valores são obtidos com base num método de passo único (ex.: RK2).

Depois de se obter os valores iniciais, podemos usar a fórmula referenciada na secção seguinte (2.6.1 Fórmulas) para calcular y_{i+1} para cada passo seguinte.

Este método é chamado de método de dois passos porque utiliza dois pontos anteriores (condição inicial e seguinte) para calcular os próximos pontos.

2.6.1 Fórmulas

$$y_{i+1} = \frac{h}{2}.(3.f(t_i, y_i) + f(t_{i-1}, y_{i-1}))$$

- Onde y_i é uma aproximação da solução da EDO no ponto t_i
- h é o tamanho do passo
- f(t,y) é a função que descreve a EDO.

2.6.2 Algoritmo/Função

1. Definir o tamanho do passo h e o intervalo de tempo $[t_0, t_f]$.
2. Inicializar o vetor de tempo t e o vetor de solução y com seus valores iniciais t_0 e y_0.
3. Utilizar um método de passo único (ex.: RK2), para obter o valor inicial para y_1.
4. Para cada passo i a partir de 2 até o número total de passos N:
    a. Calcular o próximo tempo t(i+1) = t(i) + h.
    b. Utilizar o método de Adams-Moulton de 2 passos para obter o valor de y_{i+1}:
    $$y_{i+1} = y_i + \frac{h}{2}.(3.f(t_i, y_i) + f(t_{i-1}, y_{i-1}))$$
    c. Atualizar os valores de t e y para o próximo passo.
5. Retornar os vetores t e y que contêm as soluções para cada ponto de tempo.

Em MATLAB:

```
%NAdamsMoulton Método de Adams-Bashford.
%   y = NAdamsMoulton(f, a, b , N, y0) Método numérico para a resolução de um PVI
%   y'= f(t,y) com t=[a, b] e y(a)=y0 condição inicial
%
%INPUT:
%   f - função do 2.º membro da Equação Diferencial
%   [a, b] - extremos do intervalo da variável independente t
%   n - número de subintervalos ou iterações do método
%   y0 - condição inicial t=a -> y=y0
%OUTPUT:
%   y - vector das soluções aproximações
%
%   16/04/2023 - Renato Craveiro | 2018011392 | Lic. Eng. Informática ISEC
%   Ano Letivo 2022/23

function y = NAdamsBashford(f, a, b , N, y0)

% Define o tamanho h

h = (b - a)/N;

t= a:h:b; %define os valores de 7

y = zeros(1, N+1); %inicializa y
y(1) = y0; %CI

% Usa o método de RK2 para obter valor seguinte
 k1 = h * f(t(1), y(1));             % Inclinação inicial do intervalo
 k2 = h * f(t(2), y(1) + k1);    % Inclinação final do intervalo
 y(2)=y(1) + (k1 + k2)/2;

%y(2) = y(1) + h*f(t(1), y(1));

% Usa o método de Adams-Moulton para os passos subsequentes
    for i = 2:N
        y(i+1) = y(i) + (h/2)*(3*f(t(i), y(i)) + f(t(i-1), y(i-1)));
    end
end
```

3. Exemplos de aplicação e teste dos métodos

3.1 Exercício 3 do Teste Farol

3. Considere o problema de valor inicial $y' = -2ty, \ y(0) = 2, \ t \in [0{,}1.5]$

(a) Verifique que $y(t) = 2\exp(-t^2)$ é a solução exata do problema.

(b) Complete a tabela seguinte e interprete os resultados obtidos. Para o preenchimento da coluna das aproximações de Euler, deve apresentar os cálculos das iterações da aplicação da fórmula do método de Euler.

			Aproximações		Erros	
i	t_i	$y(t_i)$ Exata	y_i Euler	y_i RK2	$\lvert y(t_i) - y_i \rvert$ Euler	$\lvert y(t_i) - y_i \rvert$ RK2
0	0	2			0	0
1		1.5576		1.5000		0.0576
2	1					0.0142
3	1.5	0.2108		0.3750		

a) Começar pelo cálculo da integral de y'=-2ty

$$y' = \frac{dy}{dt} \Leftrightarrow -2ty = \frac{dy}{dt} \Leftrightarrow (-2ty) - dy = 0 \Leftrightarrow y(-2t)dt - 1dy = 0$$

$$\Leftrightarrow (-2t\,)dt - \left(\frac{1}{y}dy\right) = 0 \Leftrightarrow \int (-2t)dt - \int \left(\frac{1}{y}\right)dy = \int 0\,dt$$

$$\Leftrightarrow -\frac{2t^2}{2} - \ln|y| = C, \qquad C \in \mathbb{R} \Leftrightarrow \ln|y| = -t^2 - c, \qquad c \in \mathbb{R}$$

$$\Leftrightarrow |y| = e^{-t^2-C}, \qquad C \in \mathbb{R} \Leftrightarrow |y| = \frac{1}{e^C}e^{-t^2}, \qquad C \in \mathbb{R}$$

$$\Leftrightarrow |y| = Ce^{-t^2}, \qquad C \in \mathbb{R} \Leftrightarrow y = Ce^{-t^2}, \qquad C \in \mathbb{R}$$

Calcular C com valor inicial

$$\begin{cases} y(t) = Ce^{-t^2} \\ y(0) = 2 \end{cases} \Leftrightarrow Ce^{0^2} = 2 \Leftrightarrow C.1 = 2 \Leftrightarrow C = 2$$

Solução exata: $2e^{-t^2} = 2\exp\left(-t^2\right)$ verificando a equação apresentada.

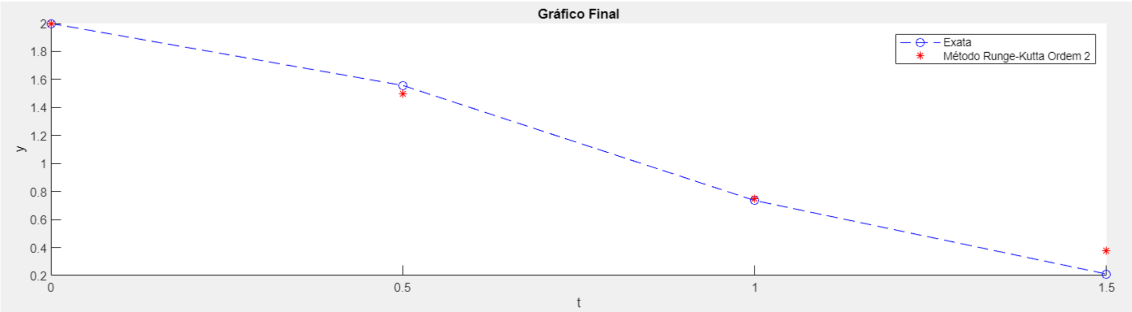
b)

			Aproximações		Erros	
i	t_i	$y(t_i)$ Exata	y_i Euler	y_i RK2	$\lvert y(t_i) - y_i \rvert$ Euler	$\lvert y(t_i) - y_i \rvert$ RK2
0	0	2	2	2	0	0
1	0.5	1.5576	2	1.5000	0.4424	0.0576
2	1	0.735	1	0.75	0.2642	0.0142
3	1.5	0.2108	0	0.3750	0.2108	0.1642

3.1.1 PVI – Equação Diferencial de 1ª ordem e Condições Iniciais

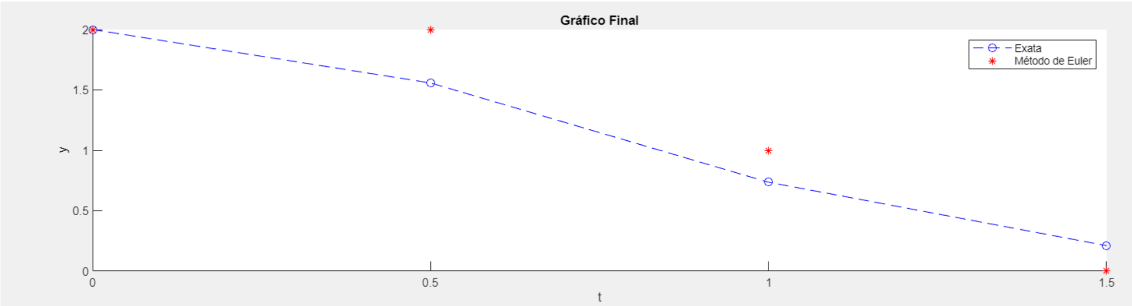
$$\begin{cases} y' = -2ty \rightarrow Equação\ Diferencial\ de\ primeira\ ordem \\ y(0) = 2 \rightarrow Condição\ inicial \\ t \in [0\ ,\ \ 1.5] \end{cases}$$

3.1.2 Exemplos de Output – App com gráfico e Tabela



Solução Exata: 2*exp(-t^2)

t	Exata	RK2	Erro
0	2.0000	2.0000	0
0.5000	1.5576	1.5000	0.0576
1.0000	0.7358	0.7500	0.0142
1.5000	0.2108	0.3750	0.1642



Solução Exata: 2*exp(-t^2)

t	Exata	Euler	Erro
0	2.0000	2	0
0.5000	1.5576	2	0.4424
1.0000	0.7358	1	0.2642
1.5000	0.2108	0	0.2108

3.2 Problemas de aplicação do livro https://moodle.isec.pt/moodle/mod/page/view.php?id=237035

3.2.1 Modelação Matemática do problema

1. If air resistance is proportional to the square of the instantaneous velocity, then the velocity v of a mass m dropped from a height h is determined from

$$m\frac{dv}{dt} = mg - kv^2, \quad k > 0$$

Let $v(0) = 0$, $k = 0.125$, $m = 5$ slugs, and $g = 32\, ft/s^2$.

   (a) Use the Runge-Kutta method with $h = 1$ to find an approximation to the velocity of the falling mass at $t = 5\,s$.

   (b) Use a numerical solver to graph the solution of the initial-value problem.

   (c) Use separation of variables to solve the initial-value problem and find the true value $v(5)$.

Exercício 1:

Dados do problema

$$\begin{cases} m.\dfrac{dv}{dt} = mg - kv^2, \qquad k > 0 \\ v(0) = 0 \\ k = 0.125 \\ m = 5 \\ g = 32ft/s^2 \end{cases}$$

Simplificando:

$$m.\frac{dv}{dt} = m.g - k.v^2, \qquad k > 0 \;\Leftrightarrow\; \frac{dv}{dt} = \frac{mg - kv^2}{m}, k > 0 \;\Leftrightarrow\; \frac{dv}{dt} = g - \frac{kv^2}{m}, \qquad k > 0$$

Utilizando os dados do problema:

$$\frac{dv}{dt} = 32 - \frac{0.125v^2}{5} \Leftrightarrow \frac{dv}{dt} = 32 - 0.025v^2 \Leftrightarrow v' = 32 - 0.025v^2$$

PVI final:

$$\begin{cases} v' = 32 - 0.025\, v^2 \\ v(0) = 0 \\ t \in [0,5] \end{cases}$$

2. A mathematical model for the area A (in cm^2) that a colony of bacteria (B. forbiddenkeyworddendroides) occupies is given by

$$\frac{dA}{dt} = A(2.128 - 0.0432A).$$

Suppose that the initial area is $0.24\,cm^2$.

(a) Use the Runge-Kutta method with $h = 0.5$ to complete the following table.

$t(days)$	1	2	3	4	5
$A(observed)$	2.78	13.53	36.30	47.50	49.40
$A(approximated)$					

(b) Use a numerical solver to graph the solution of the initial-value problem. Estimate the values $A(1)$, $A(2)$, $A(3)$, $A(4)$, and $A(5)$ from the graph.

(c) Use separation of variables to solve the initial-value problem and compute the values $A(1)$, $A(2)$, $A(3)$, $A(4)$, and $A(5)$.

Exercício 2:

Dados do problema

$$\begin{cases} \dfrac{dA}{dt} = A(2.128 - 0.0432A) = A' \\ A(0) = 0.24 \\ t \in [0,5] \end{cases}$$

Sabendo que h=0.5 podemos calcular N:

$$h = \frac{b-a}{n} \Leftrightarrow 0.5 = \frac{(5-0)}{n} \Leftrightarrow n = \frac{5}{0.5} \Leftrightarrow n = 10$$

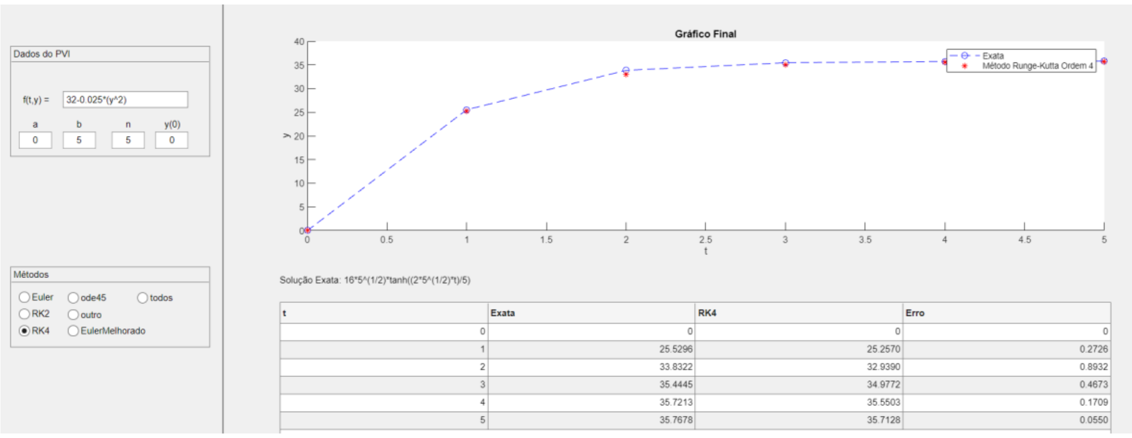
Ficando:

$$\begin{cases} \dfrac{dA}{dt} = A(2.128 - 0.0432A) = A' \\ A(0) = 0.24 \\ t \in [0,5] \\ n = 10 \end{cases}$$

3.2.2 Resolução através da App desenvolvida

<u>Exercício 1:</u>

Resolução através de Runge-Kutta (enunciado não explicíta – utiliza-se RK de ordem 4):



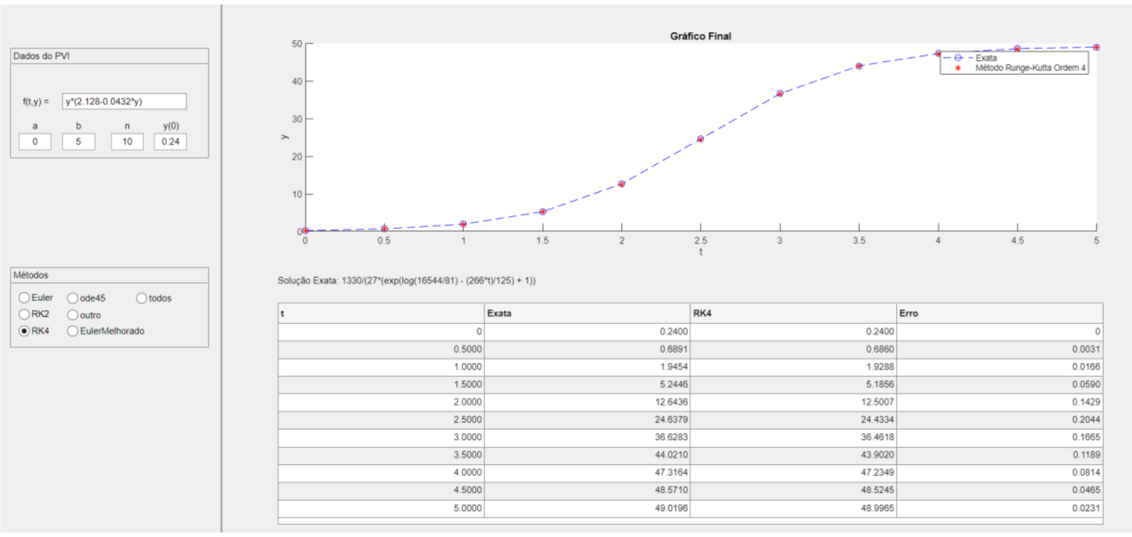
t	Exata	RK4	Erro
0	0	0	0
1	25.5296	25.2570	0.2726
2	33.8322	32.9390	0.8932
3	35.4445	34.9772	0.4673
4	35.7213	35.5503	0.1709
5	35.7678	35.7128	0.0550

(utilizou-se y em vez de v)

$$v(5) \approx 35.7128\ (arrendondado\ a\ RK4) \rightarrow \quad Sol\ Exata\ de\ v(5) = 35.7678$$

<u>Exercício 2:</u>

Resolução através de Runge-Kutta (enunciado não explicíta – utiliza-se RK de ordem 4):



t	Exata	RK4	Erro
0	0.2400	0.2400	0
0.5000	0.6891	0.6860	0.0031
1.0000	1.9454	1.9288	0.0166
1.5000	5.2446	5.1856	0.0590
2.0000	12.6436	12.5007	0.1429
2.5000	24.6379	24.4334	0.2044
3.0000	36.6283	36.4618	0.1665
3.5000	44.0210	43.9020	0.1189
4.0000	47.3164	47.2349	0.0814
4.5000	48.5710	48.5245	0.0465
5.0000	49.0196	48.9965	0.0231

3.3 Problemas de aplicação da alínea 2.b do Teste Farol

2. Qual o valor lógico das seguintes afirmações? Justifique a sua resposta.

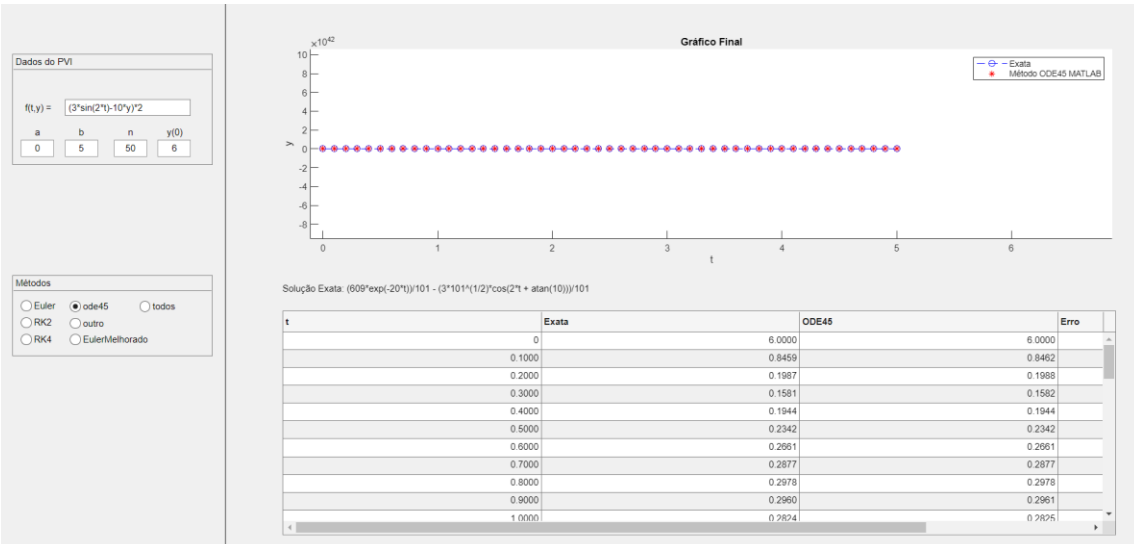
(b) A força eletromotriz e de um circuito RL com intensidade i, resistência $R = 10\ \Omega$ (ohms) e indutância $L = 0.5\ h$ (henry), é igual à queda de tensão Ri mais a força eletromotriz de autoindução $L\frac{di}{dt}$. Assim, a intensidade de corrente i, no instante t, se $e = 3\sin(2t)$ (em volts) e $i = 6$ quando $t = 0$ é dada pela solução particular $i(t) = \frac{609}{101}e^{-20t} - \frac{30}{101}\sin 2t + \frac{3}{101}\cos 2t$. À medida que o tempo aumenta, o termo que envolve e^{-20t} perde influência no valor da intensidade da corrente. Diz-se que este termo é o termo do *estado transitório* e o outro é o termo do *estado permanente*.

3.3.1 Modelação Matemática do problema

$$e = Ri + L\frac{di}{dt} \Leftrightarrow 3.sen(2t) = 10i + 0.5\,.\frac{di}{dt} \Leftrightarrow \frac{3.sen(2t)}{10i} = 0.5\frac{di}{dt} \Leftrightarrow i' = \left(\frac{3.sen(2t)}{10i}\right).2$$

$$\begin{cases} i' = \left(\dfrac{3.sen(2t)}{10i}\right).2 \\ i(0) = 6 \end{cases}$$

3.3.2 Resolução através da App desenvolvida



t	Exata	ODE45	Erro
0	6.0000	6.0000	
0.1000	0.8459	0.8462	
0.2000	0.1987	0.1988	
0.3000	0.1581	0.1582	
0.4000	0.1944	0.1944	
0.5000	0.2342	0.2342	
0.6000	0.2661	0.2661	
0.7000	0.2877	0.2877	
0.8000	0.2978	0.2978	
0.9000	0.2960	0.2961	
1.0000	0.2824	0.2825	

4 Conclusão

Em conclusão, as equações diferenciais ordinárias (EDOs) são importantes para modelar fenómenos dinâmicos em várias áreas da ciência e engenharia.

A resolução de problemas de valor inicial (PVI) de EDOs pode ser feita por métodos analíticos ou numéricos, sendo estes últimos mais comuns.

Os métodos apresentados são métodos numéricos que usam informações de passos anteriores para obter uma solução para o próximo passo.

A resolução de PVIs de EDOs é essencial para a compreensão e modelação de fenómenos em diversos campos, entre eles, a engenharia.

5 Bibliografia

Moodle ISEC – AM2 (04/2023):

https://moodle.isec.pt/moodle/course/view.php?id=15679

GitHub – John S Butler Dit (04/2023:

https://john-s-butler-dit.github.io/NumericalAnalysisBook/Chapter%2004%20-%20Multistep%20Methods/402_Adams%20Bashforth%20Population%20Equations.html

MATLAB Answers:

https://www.mathworks.com/matlabcentral/answers/index

6 Autoavaliação e heteroavaliação do trabalho submetido numa escala de 0 a 5 valores

Renato Alexandre Oliveira Craveiro – 4 valores em 5.