A picture containing text

Description automatically generated



INSTITUTO POLITÉCNICO DE COIMBRA

INSTITUTO SUPERIOR

DE ENGENHARIA

DE COIMBRA

DEPARTAMENTO DE INFORMÁTICA E SISTEMAS

**Métodos Numéricos para Equações Diferenciais Ordinárias e Problemas de Valor Inicial**

Relatório de Licenciatura

Autores

**Ana Rita Conceição Pessoa – 2023112690**

**João Francisco de Matos Claro – 2017010293**

Coimbra, março e 2024

# Índice

## Índice de texto

[Parte pré-textual **Erro! Marcador não definido.**](#_Toc161244872)

[1 Índice 1](#_Toc161244873)

[1.1 Índice de texto 1](#_Toc161244874)

[1.2 Índice de figuras, quadros e afins 2](#_Toc161244875)

[2 Lista de siglas, acrónimos e símbolos 4](#_Toc161244876)

[2.1 Lista de siglas e acrónimos 4](#_Toc161244877)

[2.2 Lista de símbolos 4](#_Toc161244878)

[2.2.1 Exemplos de listas de símbolos 4](#_Toc161244879)

[3 Introdução 5](#_Toc161244880)

[3.1 Equação diferencial: definição e propriedades 5](#_Toc161244881)

[3.2 Definição de PVI 6](#_Toc161244882)

[4 Métodos Numéricos para resolução de PVI 7](#_Toc161244883)

[4.1 Método de Euler 7](#_Toc161244884)

[4.1.1 Fórmulas 7](#_Toc161244885)

[4.1.2 Algoritmo/Função 8](#_Toc161244886)

[4.1 Método de Euler Melhorado ou Modificado 9](#_Toc161244887)

[4.1.1 Fórmulas 9](#_Toc161244888)

[4.1.2 Algoritmo/Função 9](#_Toc161244889)

[4.2 Método de RK2 10](#_Toc161244890)

[4.2.1 Fórmulas 10](#_Toc161244891)

[4.2.2 Algoritmo/Função 10](#_Toc161244892)

[4.3 Método de RK4 10](#_Toc161244893)

[4.3.1 Fórmulas 10](#_Toc161244894)

[4.3.2 Algoritmo/Função 10](#_Toc161244895)

[4.4 Função ODE45 do Matlab 10](#_Toc161244896)

[4.5 Outro método pesquisado... 10](#_Toc161244897)

[5 Exemplos de aplicação e teste dos métodos 11](#_Toc161244898)

[5.1 Exercício 3 do Teste Farol 11](#_Toc161244899)

[5.1.1 PVI - Equação Diferencial de 1ª ordem e Condições Iniciais 11](#_Toc161244900)

[5.1.2 Exemplos de output - App com gráfico e tabela 11](#_Toc161244901)

[5.1 Problemas de aplicação do livro 11](#_Toc161244902)

[5.1.1 Modelação matemática do problema 11](#_Toc161244903)

[5.1.2 Resolução através da App desenvolvida 11](#_Toc161244904)

[5.2 Problemas de aplicação da alínea 2.b do teste Farol 11](#_Toc161244905)

[5.2.1 Modelação matemática do problema 11](#_Toc161244906)

[5.2.2 Resolução através da App desenvolvida 11](#_Toc161244907)

[6 Conclusão 12](#_Toc161244908)

[7 Bibliografia 12](#_Toc161244909)

[8 Autoavaliação e heteroavaliação do trabalho submetido 12](#_Toc161244910)

## Índice de figuras, quadros e afins

[Figura 1.1 – Introdução de índice de texto **Erro! Marcador não definido.**](#_Toc138337193)

[Figura 1.2 – Personalização do índice de texto **Erro! Marcador não definido.**](#_Toc138337194)

[Figura 1.3 – Introdução de índices de figuras, quadros e afins **Erro! Marcador não definido.**](#_Toc138337195)

[Figura 1.4 – Exemplo de atualização de índice automático **Erro! Marcador não definido.**](#_Toc138337196)

[Figura 3.1 - Personalização das margens **Erro! Marcador não definido.**](#_Toc138337197)

[Figura 3.2 – Introdução de quebras de páginas: a) através do separador Inserir; b) através do separador Esquema **Erro! Marcador não definido.**](#_Toc138337198)

[Figura 3.3 - Introdução de cabeçalho **Erro! Marcador não definido.**](#_Toc138337199)

[Figura 3.4 - Propriedades de cabeçalhos **Erro! Marcador não definido.**](#_Toc138337200)

[Figura 3.5 - Formatação do número de páginas **Erro! Marcador não definido.**](#_Toc138337201)

[Figura 4.1 – Localização do painel de estilos **Erro! Marcador não definido.**](#_Toc138337202)

[Figura 4.2 – Estilos em uso no documento. **Erro! Marcador não definido.**](#_Toc138337203)

[Figura 4.3 – Estilos disponíveis para a formatação da relatório de licenciatura/projeto final de licenciatura. **Erro! Marcador não definido.**](#_Toc138337204)

[Figura 4.4 – Visualização e edição de estilos **Erro! Marcador não definido.**](#_Toc138337205)

[Figura 4.5 – Criação de novos estilos **Erro! Marcador não definido.**](#_Toc138337206)

[Figura 5.1 – Exemplo de figura simples. **Erro! Marcador não definido.**](#_Toc138337207)

[Figura 5.2 – Exemplo de figura dupla com fotografias: a) azul de metileno; b) equivalente de areia. **Erro! Marcador não definido.**](#_Toc138337208)

[Figura 5.3 – Ferramenta de introdução de legendas: a) criação de novas etiquetas; b) formatação da numeração da legenda **Erro! Marcador não definido.**](#_Toc138337209)

[Figura 5.4 – Seleção de um quadro para ser posteriormente legendado **Erro! Marcador não definido.**](#_Toc138337210)

[Figura 5.5 – Introdução de referências cruzadas **Erro! Marcador não definido.**](#_Toc138337211)

[Figura 5.6 – Atualização de referências cruzadas **Erro! Marcador não definido.**](#_Toc138337212)

[Figura 6.1 – Introdução e numeração automática de equações. **Erro! Marcador não definido.**](#_Toc138337213)

[Figura 7.1 – Gestor de referências do Microsoft® Word. **Erro! Marcador não definido.**](#_Toc138337214)

## 

# Lista de siglas, acrónimos e símbolos

## Lista de siglas e acrónimos

|  |  |
| --- | --- |
| EDO | Equações Diferenciais Ordinárias |
| PVI | Problemas de Valor Inicial |
| ED | Equação Diferencial |
| RK2 | Runge-Kutta de Segunda Ordem |
| RK4 | Runge-Kutta de Quarta Ordem |
| SI | Sistema Internacional de Unidades |

## Lista de símbolos

### Exemplos de listas de símbolos

As tabelas seguintes exemplificam as recomendações anteriormente descritas.

**Alfabeto latino**

|  |  |
| --- | --- |
|  | Representa "pertence a intervalo" |
| *b* | Termo do 1.º grau num polinómio do segundo grau representado na forma canónica |
| *c* | Termo independente num polinómio do segundo grau representado na forma canónica |
| kN | Quilonewton |
| *p*’ | Tensão média efetiva (kPa) |

**Alfabeto grego**

|  |  |
| --- | --- |
|  | Diferença ou variação entre dois valores |
| ∑ | Soma dos termos da série |
| β | Representa os coeficientes |

# Introdução

Este trabalho, desenvolvido no âmbito da unidade curricular de Análise Matemática II, tem como objetivo principal a aquisição e aprofundamento de conhecimentos adquiridos referentes aos métodos numéricos à resolução de Equações Diferenciais Ordinárias (EDO) e Problemas de Valor Inicial (PVI), assim como na linguagem de programação Matlab.

No decorrer deste relatório, serão abordados os temas propostos no enunciado, examinando conceitos como EDO, PVI e os diferentes métodos e fórmulas para a sua resolução, os quais serão acompanhados pelos respetivos códigos implementados em Matlab.

## Equação diferencial: definição e propriedades

As equações diferencias são equações que envolvem uma função incógnita e uma ou mais de suas derivadas (ou diferenciais). **doc01\_EDO.** Estas equações podemser **lineares** (quando a relação entre a função desconhecida, as suas derivadas e outras variáveis na equação é linear) ou **não lineares** (quando a relação é mais complexa). **Livro**

A resolução das equações diferencias pode ser feita a partir de vários métodos desde os analíticos, como a separação de variáveis, até aos numéricos, como o método de Euler para aproximar as soluções. **Livro**

Uma equação diferencial pode ter várias propriedades, dependendo das suas características e do tipo de equação:

- **Ordem**: A ordem de uma equação diferencial é o número de vezes que a função desconhecida aparece diferenciada na equação.

Por exemplo, uma equação que envolve a primeira derivada de y tem ordem 1, e uma equação que envolve a segunda derivada de y tem ordem 2.

- **Linearidade**: Uma equação diferencial é chamada linear se a sua forma é linear em relação à função desconhecida e as suas derivadas.

Por exemplo, a equação é linear, enquanto a equação é não linear.

- **Grau**: O grau de uma equação diferencial é o grau mais alto das derivadas que aparecem na equação.

Por exemplo, a equação tem grau 2.

- **Autonomia**: Uma equação diferencial é chamada autónoma se ela não depende explicitamente da variável independente.

Por exemplo, a equação é autónoma.

- **Homogeneidade**: Uma equação diferencial é chamada homogénea se ela pode ser escrita na forma , onde f é uma função qualquer.

Por exemplo, a equação é homogénea.

- **Separabilidade**: Uma equação diferencial é chamada separável se pode ser escrita na forma g(y)dy = f(x)dx, onde g e f são funções.

Por exemplo, a equação é separável.

Estas são apenas algumas das propriedades possíveis de equações diferenciais, e cada equação pode ter características únicas que as distinguem de outras equações diferenciais. **Livro**

## Definição de Problema de Valor Inicial

Um Problema de Valor Inicial (PVI) é um tipo específico de problema associado a equações diferenciais, onde o objetivo é encontrar uma solução que satisfaça a equação diferencial, sujeita a uma condição inicial que estipula o valor da solução em um ponto específico. Essa condição inicial é geralmente expressa como uma equação ou relação que define o valor da solução num ponto denominado ponto inicial. A solução obtida é considerada válida para todo o domínio onde a equação diferencial é definida. **doc01\_EDO**.

Em resumo, um PVI pretende determinar a solução de uma equação diferencial juntamente com uma condição inicial que especifica o valor da solução num ponto particular. **Livro**

Por exemplo,

Neste problema, a equação diferencial (ED) é , o intervalo pretendido é e a condição inicial (valor inicial) .

# Métodos Numéricos para resolução de PVI

## Método de Euler

O Método de Euler ou Método da Reta Tangente é um método numérico utilizado para resolver EDO por meio de aproximações. Ou seja, este método baseia-se numa aproximação de primeira ordem, aproximando a curva solução da EDO por meio de uma linha reta. Esta reta é determinada pelos valores iniciais da função e da sua derivada, permitindo obter valores aproximados da solução em pontos subsequentes da trajetória.

O método referido divide o intervalo de tempo da solução em subintervalos menores e, em cada um, estima a solução da EDO por uma reta tangente no ponto inicial. Esta reta é usada para estimar o valor da solução num ponto final dentro do subintervalo, e o processo é repetido para obter sucessivas aproximações da solução.

Embora o Método de Euler seja simples e fácil de implementar, pode gerar estimativas imprecisas da solução, especialmente se o tamanho dos subintervalos for grande. Contudo, é uma ferramenta fundamental para a compreensão de outros métodos numéricos mais precisos para a resolução de EDO. É particularmente eficaz para solucionar equações diferenciais simples, embora a sua precisão possa ser limitada. **Livro**

### Fórmulas

Sabendo que o Método de Euler é um procedimento numérico de primeira ordem para aproximar a solução da equação diferencial que satisfaz a condição inicial:

O Método de Euler para resolver um PVI é dado pela seguinte equação:

**( 1 )**

Onde:

• → Próximo valor aproximado da solução do problema inicial (na abcissa );

• → Valor aproximado da solução do problema inicial na abcissa atual;

• ℎ → Valor de cada subintervalo (passo);

• 𝑓 () → Valor da equação em e .

### Algoritmo/Função

**Algoritmo:**

1. Definir o valor inicial para a variável dependente e uma condição inicial para a variável independente

2. Definir o tamanho de passo que determinará a distância entre os pontos onde a solução será aproximada.

3. Para cada passo de até ao ponto final desejado:

a) Calcular a derivada da função desconhecida ) no ponto atual )

b) Utilizar a derivada calculada para estimar a mudança na variável dependente: .

c) Calcular a próxima aproximação da solução:

d) Atualize a variável independente:

4. Repetir o passo 3 até alcançar o ponto final desejado ou até que o número de passos determinado seja atingido.

**Função (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

%

% Autores: Arménio Correia | armenioc@isec.pt

% Ana Rita Conceição Pessoa .: a2023112690@isec.pt

% João Francisco de Matos Claro .: a21270422@isec.pt

%

% 02/04/2024

% Definição da função NEuler que recebe:

% uma função f;

% os limites de integração a e b;

% o número de passos n;

% o valor inicial y0.

function y = NEuler (f, a, b, n, y0)

% Calcula o tamanho do passo h com base nos limites de integração e no

% número de passos.

h = (b-a) /n;

% Inicializa os vetores para armazenar os valores das variáveis:

% -> independentes (t)

% -> dependentes (y)

% começando com os valores iniciais.

t (1) = a;

y (1) = y0;

% Início do loop para cada passo de integração.

for i = 1:n

% Calcula as novas aproximações usando o Método de Euler e

% atualiza os valores de y e t para o próximo passo.

y(i+1) = y(i)+h\*f(t(i), y(i));

t(i+1) = t(i)+h;

% Finalização do loop.

end

## Método de Euler Melhorado ou Modificado

O Método de Euler Melhorado é uma versão aprimorada do Método de Euler para resolver equações diferenciais ordinárias com problemas de valor inicial. Vai permitir calcular uma média ponderada das inclinações em cada ponto para obter uma solução mais precisa do que o Método de Euler original. Esta abordagem reduz o erro de truncamento local (diferença entre o valor exato e o valor aproximado obtido), resultando em resultados mais precisos.

Em resumo, o Método de Euler Melhorado é uma evolução do Método de Euler, fornecendo soluções mais precisas para PVI.

### Fórmulas

O Método de Euler Melhorado ou Modificado para resolver um PVI é dado pela seguinte equaçõe:

**( 2 )**

Onde:

• → Próximo valor aproximado da solução do problema inicial (na abcissa );

• → Valor aproximado da solução do problema inicial na abcissa atual;

• ℎ → Valor de cada subintervalo (passo);

**Cálculo de :**

• → Inclinação no início do intervalo;

• 𝑓 () → Valor da equação em e .

**Cálculo de :**

• k2 → Inclinação no fim do intervalo;

• ti → Valor da abcissa atual;

• ℎ → Tamanho de cada subintervalo (passo);

• → Valor aproximado da solução do problema original na abcissa atual;

• → Inclinação no início do intervalo.

### Algoritmo/Função

**Algoritmo:**

1. Definir o passo h;

2. Criar um vetor y para guardar a solução;

3. Atribuir o primeiro valor de y (condição inicial) do PVI;

4. Cálculo da inclinação no início do intervalo;

5. Cálculo da inclinação no fim do intervalo;

6. Cálculo da média das inclinações;

7. Cálculo do valor aproximado para a iésima iteração.

**Função (MatLab):**

….. juntar código + comentários

## Método de Runge-Kutta de Segunda Ordem

O Método de Runge-Kutta de segunda ordem (RK2) é um método numérico para resolver equações diferenciais ordinárias de primeira ordem com um problema de valor inicial. Ele envolve calcular duas inclinações da solução em pontos diferentes e usar uma média ponderada delas para atualizar a solução.

Resumidamente, o Método de RK2 é eficiente porque fornece resultados mais precisos que o Método de Euler, e é relativamente simples de implementar, pois requer apenas cálculos envolvendo derivadas de primeira ordem da função.

### Fórmulas

O Método de RK2 para resolver um PVI é dado pela seguinte equação:

**( 3 )**

Onde:

• → Próximo valor aproximado da solução do problema inicial (na abcissa );

• → Valor aproximado da solução do problema inicial na abcissa atual;

**Cálculo de :**

• → Inclinação no início do intervalo;

• ℎ → Tamanho de cada subintervalo (passo);

• 𝑓() → Valor da equação em e .

**Cálculo de :**

• → Inclinação no fim do intervalo;

• → Valor da abcissa atual;

• ℎ → Tamanho de cada subintervalo (passo);

• → Valor aproximado da solução do problema original na abcissa atual;

• → Inclinação no início do intervalo.

### Algoritmo/Função

**Algoritmo:**

1. Definir o passo h;

2. Criar um vetor y para guardar a solução;

3. Atribuir o primeiro valor de y (condição inicial) do PVI;

4. Cálculo da inclinação no início do intervalo;

5. Cálculo da inclinação no fim do intervalo;

6. Cálculo da média das inclinações;

7. Cálculo do método de RK2 para a iésima iteração.

**Função (MatLab):**

% RK2 - Método de Runge-Kutta de ordem 2

% y = RK2(f,a,b,n,y0) Método numérico para a resolução de um PVI

% y' = f(t,y), Equação Diferencial

% t = [a,b]

% y(a) = y0, cI (condição inicial)

%INPUTS:

% 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

%OUTPUTS:

% y - vector das aproximações discretas da solução exacta

% y(i+1) = y(i)+h\*f(t(i)y(i)), i = 0,1,2,...,n-1

%

% Trabalho realizado por:

% Ana Rita Conceição Pessoa - 2023112690

% João Francisco de Matos Claro - 2017010293

% Esta linha define uma função chamada RK2 que aceita cinco argumentos de

% entrada:

% - f (a função diferencial a ser avaliada)

% - a (o limite inferior do intervalo)

% - b (o limite superior do intervalo)

% - n (o número de passos)

% - y0 (o valor inicial)

function y = RK2(f, a, b, n, y0)

% Calcula o tamanho do passo h com base nos limites inferior e superior do

% intervalo a e b, e no número de passos n

h = (b-a) /n;

% Gera um vetor t contendo os pontos discretos ao longo do intervalo de

% a a b, com um espaçamento de h entre os pontos

t = a:h:b;

% Inicializa um vetor y de tamanho n+1 preenchido com zeros para armazenar

% os valores da solução

y = zeros(1,n+1);

% Define o primeiro elemento do vetor y como o valor inicial y0

y(1) = y0;

% Inicia um loop que irá iterar de 1 até n,

% representando cada passo do método de Runge-Kutta de segunda ordem

for i=1:n

% Calcula o primeiro coeficiente k1 utilizando a derivada da função f

% no ponto t(i) e y(i)

k1 = h\*f(t(i), y(i));

% Calcula o primeiro coeficiente k1 utilizando a derivada da função f

% no ponto t(i) e y(i)

k2 = h\*f(t(i+1), y(i)+k1);

% Atualiza o valor de y(i+1) usando uma média ponderada dos

% coeficientes k1 e k2, que representa a solução no próximo ponto

y(i+1) = y(i)+(k1+k2)/2;

end

## Método de Runge-Kutta de Quarta Ordem

O Método de Runge-Kutta de Quarta Ordem (RK4) é um método numérico para resolver EDO de primeira ordem com um PVI. Ele é mais preciso que o Método de RK2, mas requer mais cálculos. O RK4 usa a média ponderada de quatro inclinações calculadas em diferentes pontos para encontrar a solução. Isso torna o RK4 altamente preciso e confiável, mas é necessário ajustar o tamanho do passo de tempo para garantir resultados precisos. Em comparação com outros métodos, o RK4 é considerado o mais eficaz para resolver EDO. Ele não precisa calcular derivadas, apenas avalia a função em diferentes pontos.

### Fórmulas

O Método de RK4 para resolver um PVI é dado pela seguinte equação:

**( 4 )**

Onde:

• → Próximo valor aproximado da solução do problema inicial (na abcissa );

• → Valor aproximado da solução do problema inicial na abcissa atual;

• ℎ → Tamanho de cada subintervalo (passo);

• → Inclinação no início do intervalo;

• → Inclinação no ponto médio do intervalo;

• → Inclinação no ponto médio do intervalo;

• → Inclinação no final do intervalo

Média ponderada das inclinações:

### Algoritmo/Função

**Algoritmo:**

1. Definir e calcular o passo h;

2. Criar um vetor y para guardar a solução e atribuir ;

3. Atribuir o primeiro valor de y;

4. Cálculo da inclinação no início do intervalo;

5. Cálculo da inclinação no ponto médio do intervalo;

6. Cálculo de uma nova inclinação no ponto médio do intervalo;

7. Cálculo da inclinação no final do intervalo;

8. Cálculo do método RK4.

**Função (MatLab):**

% RK4 - Método de Runge-Kutta de ordem 4

% y = RK4(f,a,b,n,y0) Método numérico para a resolução de um PVI

% y' = f(t,y), Equação Diferencial

% t = [a,b]

% y(a) = y0, cI (condição inicial)

%INPUTS:

% 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

%OUTPUTS:

% y - vector das aproximações discretas da solução exacta

% y(i+1) = y(i)+h\*f(t(i)y(i)), i = 0,1,2,...,n-1

%

% Trabalho realizado por:

% Ana Rita Conceição Pessoa - 2023112690

% João Francisco de Matos Claro - 2017010293

% Esta linha define uma função chamada RK4 que recebe cinco argumentos:

% - f, que é a função que representa a equação diferencial

% - a e b, que são os limites do intervalo onde queremos resolver a equação

% - n, que é o número de passos ou subdivisões desse intervalo

% - y0, que é a condição inicial

function y = RK4(f,a,b,n,y0)

% Esta linha calcula o tamanho de cada passo de tempo h,

% dividindo a diferença entre b e a pelo número de passos n

h = (b-a)/n;

% Esta linha cria um vetor t com os pontos discretos ao longo do intervalo

% de a a b, com intervalo h

t = a:h:b;

% Esta linha inicializa um vetor y com zeros, que terá o mesmo tamanho que

% o vetor t mais um, para armazenar as soluções.

y = zeros(1,n+1);

% Esta linha define o primeiro elemento de y como a condição inicial y0

y(1) = y0;

% Este é um loop que itera de 1 até n, representando cada passo de tempo

for i=1:n

% Calcula o valor de k1

% multiplicando o tamanho do passo h pela derivada da função f

% avaliada em (t(i), y(i))

k1 = h\*f(t(i),y(i));

% Calcula o valor de k2

% multiplicando o tamanho do passo h pela derivada da função f

% avaliada em (t(i) + (h/2), y(i) + (1/2)\*k1)

k2 = h\*f(t(i)+(h/2),y(i)+(1/2)\*k1);

% Calcula o valor de k3

% multiplicando o tamanho do passo h pela derivada da função f

% avaliada em (t(i) + (h/2), y(i) + (1/2)\*k2)

k3 = h\*f(t(i)+(h/2),y(i)+(1/2)\*k2);

% Calcula o valor de k4

% multiplicando o tamanho do passo h pela derivada da função f

% avaliada em (t(i) + h, y(i) + k3)

k4 = h\*f(t(i)+(h/2),y(i)+(1/2)\*k3);

% Atualiza o valor de y para o próximo passo

% usando a média ponderada dos valores de k1, k2, k3 e k4.

y(i+1) = y(i)+(1/6)\*(k1+2\*k2+2\*k3+k4);

end

## Função ODE45 do Matlab

A função ODE45 é uma das funções nativas do MATLAB, e é baseada num método de Runge-Kutta.

### Fórmulas

Para resolver um PVI com uma EDO de ordem 2, a função ODE45 pode ser chamada da seguinte forma:

**( 5 )**

Onde:

• → Vetor das abcissas;

• f → Equação diferencial em t e em y;

• → Valor inicial do PVI (condição inicial).

### Algoritmo/Função

**Algoritmo:**

1. Definir o passo h;

2. Aproximação através da função ODE45.

**Função (MatLab):**

% N\_ODE45 Método Númerico para resolver um PVI: Função ODE45 do MATLAB

% y = N\_ODE45(f,a,b,n,y0) Método numérico para a resolução de um PVI

%INPUTS:

% f - Função da equação diferencial, em t e y

% a - Limite esquerdo do intervalo

% b - Limite direito do intervalo

% n - Numero de sub-intervalos

% y0 - Valor (condição) Inicial do PVI

%OUTPUTS:

% y - vetor das soluções aproximadas

% Trabalho realizado por:

% Ana Rita Conceição Pessoa - 2023112690

% João Francisco de Matos Claro - 2017010293

% Esta linha define uma função chamada N\_ODE45 que recebe cinco argumentos:

% - f, que é a função que representa a equação diferencial

% - a e b, que são os limites do intervalo onde queremos resolver a equação

% - n, que é o número de pontos no intervalo

% - y0, que é a condição inicial

function y = N\_ODE45(f,a,b,n,y0)

% Esta linha calcula o tamanho de cada subintervalo h,

% dividindo a diferença entre b e a pelo número de pontos n.

h = (b-a)/n;

% Esta linha cria um vetor t

% com os pontos discretos ao longo do intervalo de a a b,

% com intervalo h.

t = a:h:b;

% Esta linha utiliza a função ode45 do MATLAB

% para resolver numericamente a equação diferencial definida por f.

% Ela calcula a solução y para cada ponto em t,

% iniciando com a condição inicial y0.

[~,y] = ode45(f, t, y0);

% Esta linha transpõe o vetor de soluções y

% para que ele tenha a orientação correta,

% onde as colunas representam as variáveis dependentes

% e as linhas representam os pontos no domínio de tempo.

y = y';

end

## Método de Adams-Bashforth

O Método de Adams-Bashforth é uma técnica numérica para resolver EDO com PVI. Ele utiliza informações de pontos anteriores para estimar a solução no próximo ponto, através de uma fórmula polinomial. A ordem do método determina o grau desse polinómio. É amplamente utilizado pela sua simplicidade e facilidade de implementação, podendo ser estendido para ordens mais elevadas.

### Fórmulas

A fórmula geral do Método de Adams-Bashforth de ordem para estimar a solução no ponto é:

**( 6 )**

Onde:

• → Estimativa da solução no próximo ponto;

• → Solução no ponto atual;

• ℎ → Tamanho de cada subintervalo (passo);

• → Derivada da solução no ponto ;

• → Coeficientes determinados pelo método.

### Algoritmo/Função

**Algoritmo:**

1. Definir o passo h e o número de passos n;

2. Aproximar os primeiros pontos usando um método de passo único (como o método de Runge-Kutta);

3. Usar a fórmula do Método de Adams-Bashforth para calcular o próximo ponto;

4. Repetir o passo 3 até alcançar o ponto final desejado.

**Função (MatLab):**

%AdamBashford Método de Adams-Bashforth para ED/PVI.

% y = AdamBashford(f,a,b,n,y0)

% Método de 2 passos numérico para a resolução de um PVI

% y'= f(t,y) com t=[a, b] e y(a)=y0 condição inicial

%INPUTS:

% 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

%OUTPUTS:

% y - vector das soluções aproximações

% y(i+2)=y(i+1)+(3/2)\*f(t(i+1),y(i+1))-(1/2)\*h\*f(t(0),y(0));

% Autores: Arménio Correia | armenioc@isec.pt

% Ana Rita Conceição Pessoa .: a2023112690@isec.pt

% João Francisco de Matos Claro .: a21270422@isec.pt

%

% 13/03/2024

% Esta linha define uma função chamada AdamBashford que aceita:

% - uma função f

% - um intervalo [a, b]

% - o número de passos n

% - o valor inicial y0

function y = AdamBashford(f,a,b,n,y0)

% Calcula o tamanho do passo h

% com base no intervalo [a, b] e no número de passos n

h = (b-a)/n;

% Pré alocação de memória

% Cria um vetor t contendo os pontos ao longo do intervalo [a, b]

% com intervalo h

t = a:h:b;

% Inicializa um vetor y com zeros para armazenar as soluções

y = zeros(1,n+1);

% Define os valores iniciais de t e y

t(1) = a;

y(1) = y0;

% Usa o Método de Euler para estimar o segundo valor de y

y(2) = NEuler(f,a,a+h,n,y0);

% Inicia um loop que itera de 1 até n

for i=1:n

% Calcula os valores de y usando a fórmula do Método de Adams-Bashforth

y(i+2)=y(i+1)+(3/2)\*f(t(i+1),y(i+1))-(1/2)\*h\*f(t(0),y(0));

% Atualiza os valores de t para o próximo passo

t(i+1)=t(i)+h;

t(i+2)=t(i+1)+h;

end

## Exercício 3 do Teste Farol

### PVI - Equação Diferencial de 1ª ordem e Condições Iniciais

### Exemplos de output - App com gráfico e tabela

## Problemas de aplicação do livro

### Modelação matemática do problema

### Resolução através da App desenvolvida

## Problemas de aplicação da alínea 2.b do teste Farol

### Modelação matemática do problema

### Resolução através da App desenvolvida

# Conclusão

# Bibliografia

# Autoavaliação e heteroavaliação do trabalho submetido

Text

Description automatically generated with medium confidence