

Atividade 1 – Métodos numéricos para EDO de Problema de valor inicial

Análise Matemática 2

Docente: Arménio Correia

Autores:

Daniel Duarte Dias Ferreira Albino - a2020134077 – LEI

Miguel Ferreira Neves - a2020146521 - LEI

Nuno Alexandre Domingues - a2020109910 - LEI

Engenharia Informática

Coimbra, abril de 2021

| | |
|--|----|
| 1. Introdução..... | 3 |
| 1.1. Equação diferencial: definição e propriedades..... | 4 |
| 1.2. Definição de PVI | 5 |
| 2. Métodos Numéricos para resolução de PVI..... | 6 |
| 2.1. Método de Euler..... | 6 |
| 2.1.1. Fórmula | 6 |
| 2.1.2. Algoritmo / Função..... | 7 |
| 2.2. Método de Euler Melhorado | 9 |
| 2.2.1. Fórmula | 9 |
| 2.2.2. Algoritmo/Função..... | 10 |
| 2.3. Método de RK2 | 12 |
| 2.3.1. Fórmula | 12 |
| 2.3.2. Algoritmo/Função..... | 13 |
| 2.4. Método de RK4 | 15 |
| 2.4.1. Fórmula | 15 |
| 2.4.2. Algoritmo/Função..... | 16 |
| 2.5. Função ODE45 do Matlab..... | 18 |
| 2.5.1. Fórmula | 18 |
| 2.6. Método Adams-Bashforth (ordem4) | 20 |
| 2.6.1. Fórmula | 20 |
| 2.6.2. Algoritmo/Função..... | 21 |
| 3. Exemplos de aplicação e teste dos métodos | 24 |
| 3.1. Exercício 3 do Teste3 2015/2016..... | 24 |
| 3.1.1. PVI – Equação Diferencial de 1ª ordem e Condições Iniciais..... | 24 |
| 3.1.2. Exemplos de output – App com gráfico e tabela..... | 27 |
| 3.2. Problemas de aplicação do livro | 32 |
| 3.2.1. Modelação matemática do problema..... | 32 |
| 3.2.2. Resolução através da App desenvolvida..... | 35 |
| 4. Conclusão | 38 |
| 5. Referências Bibliográficas..... | 39 |

1. Introdução

Este trabalho foi realizado no âmbito da unidade curricular de Análise Matemática 2, do curso de Licenciatura Informática do Instituto Superior de Engenharia de Coimbra.

O principal objetivo consiste, na aplicação de conhecimentos sobre a resolução de Equações Diferenciais Ordinárias (EDOs), como também de Problemas de Valor Inicial (PVI) e aplicações de Métodos numéricos, que permitem uma aproximação a solução exata.

A programação destes métodos numéricos na linguagem de programação *MatLab* é um elemento-chave deste trabalho, senão o principal.

Com este trabalho procura-se desenvolver o sentido de responsabilidade, investigação, competências algorítmicas e claro, a aprendizagem da linguagem de programação *Matlab*.

É também importante referir, que existe a introdução em relação ao desenvolvimento de *APPS* (Interface gráfica), que permitem uma interação mais intuitiva ao utilizador.

Este relatório subdivide-se em três partes:

1º Parte:

- Breve definição de Equação diferencial e propriedades;
- Definição de PVI.

2º Parte:

- Explicitação dos vários Métodos numéricos.

3º Parte:

- Exemplos de aplicação e teste dos métodos.

1.1. Equação diferencial: definição e propriedades

Definição 1: Uma Equação diferencial, é uma equação com derivadas onde a incógnita é uma função desconhecida.

Definição 2: As Equações diferenciais, contêm as derivadas de uma ou mais variáveis dependentes em relação a uma ou mais variáveis independentes. São muito úteis para modelagem e simulações de fenómenos que queremos analisar e compreender como operam, como por exemplo: mecânica de fluidos, termodinâmica, entre outros.

Propriedades:

As Equações diferenciais têm algumas propriedades, que são importantes de realçar.

- As Equações diferenciais podem dar-se na forma implícita ou explícita;
- A ordem da derivada mais elevada que aparece na equação diferencial determina a ordem da equação;
- As Equações diferenciais, podem ser equações diferenciais ordinárias (EDO), ou seja, que dependem de uma variável independente. Ou podem ser equações diferenciais parciais (EDP), que dependem de várias variáveis.

1.2. Definição de PVI

PVI é uma sigla, que significa Problema de Valor Inicial, este tipo de problema surge, quando estamos perante uma condição inicial, que é um ponto conhecido. Os PVIs podem ser resolvidos de forma exata ou de forma aproximada, como se vai verificar mais abaixo, neste relatório.

Por norma um PVI, representa-se matematicamente, pela Equação diferencial, o intervalo pretendido e a condição inicial (valor inicial).

$$\begin{cases} y' = F(t, y) \rightarrow ED & \text{Equação diferencial} \\ t \in [a, b] \rightarrow I & \text{Intervalo} \\ y(a) = y_0 \rightarrow CI & \text{Condição inicial} \end{cases}$$

2. Métodos Numéricos para resolução de PVI

2.1. Método de Euler

O Método de Euler é um método numérico para aproximar a solução da equação diferencial, sendo que é o método com mais erro, em relação aos outros métodos que iremos analisar.

2.1.1.Fórmula

$$y_{i+1} = y_i + hf(t_i, y_i)$$

$$i = 0, 1, \dots, n - 1$$

Onde:

$y_{i+1} \rightarrow$ Próximo valor a calcular

$y_i \rightarrow$ Valor aproximado na abcissa atual

$f(t_i, y_i) \rightarrow$ Valor da equação na abcissa t_i e ordenada y_i

$h = \frac{b - a}{n} \rightarrow$ Amplitude de cada subintervalo (PASSO)

$[a, b] \rightarrow$ Intervalo pretendido

$n \rightarrow$ número de vezes, que o intervalo vai ser dividido

2.1.2. Algoritmo / Função

Algoritmo:

1º Calcular o h (passo).

Nota: O h irá dar a amplitude da cada partição do intervalo $[a, b]$, quanto menor o h menor o erro.

$$h = \frac{b - a}{n}$$

2º Criar um vetor, por hipótese y e atribuir ao $y(1)$ a condição inicial y_0 .

$$y(1) = y_0$$

3º Calcular os $y(n)$ seguintes, seguindo o método de Euler, até à n iteração.

$$y_{i+1} = y_i + hf(t_i, y_i)$$

Pseudocódigo:

$$h \leftarrow \frac{b-a}{n}$$

$$t \leftarrow a:h:b$$

$$y(1) \leftarrow y_0$$

Para $i = 1$ até n

$$y_{i+1} = y_i + hf(t_i, y_i)$$

Fim para

Função Euler (MatLab):

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

    h = (b-a)/n;           % Cálculo do passo
    t = a:h:b;             % Alocação de memória
    y = zeros(1,n+1);      % Alocação de memória
    y(1) = y0;             % Atribuição do valor y0 ao primeiro índice do vetor y

    for i = 1:n             % Ciclo com n iterações
        y(i+1) = y(i)+h*f(t(i),y(i)); % Cálculo do Método de Euler até n
    end                    % Fim do ciclo FOR
end                        % Indicação do fim da função
```


2.2. Método de Euler Melhorado

O Método de Euler Melhorado é um também método numérico para aproximar a solução da equação diferencial, mas como o nome indica é um melhoramento do método de Euler original, pois consegue melhores aproximações. **Nota:** As suas aproximações coincidem às do RK2.

2.2.1. Fórmula

$$y_{i+1} = y_i + hf(t_i, y_i),$$

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

$$i = 0, 1, \dots, n - 1$$

Onde:

$y_{i+1} \rightarrow$ Próximo valor a calcular

$y_i \rightarrow$ Valor aproximado na abcissa atual

$f(t_i, y_i) \rightarrow$ Valor da equação na abcissa t_i e ordenada y_i

$h = \frac{b - a}{n} \rightarrow$ Amplitude de cada subintervalo (PASSO)

$[a, b] \rightarrow$ Intervalo pretendido

$n \rightarrow$ número de vezes, que o intervalo vai ser dividido

2.2.2. Algoritmo/Função

Algoritmo:

1º Calcular o h (passo).

Nota: O h irá dar a amplitude da cada partição do intervalo $[a, b]$, quanto menor o h menor o erro.

$$h = \frac{b - a}{n}$$

2º Criar um vetor, por hipótese y e atribuir ao $y(1)$ a condição inicial y_0 .

$$y(1) = y_0$$

3º Calcular os $y(n)$ seguintes, seguindo o método de Euler Melhorado, até à n iteração.

$$y_{i+1} = y_i + hf(t_i, y_i)$$

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

Pseudocódigo:

$$h \leftarrow \frac{b - a}{n}$$

$$t \leftarrow a:h:b$$

$$y(1) \leftarrow y_0$$

Para $i = 1$ até n

$$y_{i+1} = y_i + hf(t_i, y_i)$$

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

Fim para

Função Euler Melhorado (MatLab):

```
function y = NEulerM(f,a,b,n,y0)
    h = (b-a)/n;           % Cálculo do passo
    t = a:h:b;             % Alocação de memória
    y = zeros(1,n+1);      % Alocação de memória
    y(1) = y0;             % Atribuição do valor y0 ao primeiro índice do vetor y
    h2 = h/2;              % Atribuição da divisão h/2, para eficiência no ciclo FOR

    for i = 1:n              % Ciclo com n iterações
        y(i+1) = y(i)+h*f(t(i),y(i));
        y(i+1) = y(i)+h2*(f(t(i),y(i))+f(t(i+1),y(i+1))); % Cálculo do Método de Euler Melhorado até n
    end                     % Fim do ciclo FOR
end                         % Indicação do fim da função
```

2.3. Método de RK2

O Método de Runge-Kutta de ordem 2 (RK2) é método numérico de passo simples, onde utiliza derivadas para aproximar a solução da equação diferencial. **Nota:** As suas aproximações coincidem às do Euler melhorado.

2.3.1. Fórmula

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

$$i = 0, 1, \dots, n - 1$$

Onde:

$y_{i+1} \rightarrow$ Próximo valor a calcular

$y_i \rightarrow$ Valor aproximado na abcissa atual

$f(t_i, y_i) \rightarrow$ Valor da equação na abcissa t_i e ordenada y_i

$k_1 = hf(t_i, y_i)$

$k_2 = hf(t_{i+1}, y_i + k_1)$

$h = \frac{b - a}{n} \rightarrow$ Amplitude de cada subintervalo (PASSO)

$[a, b] \rightarrow$ Intervalo pretendido

$n \rightarrow$ número de vezes, que o intervalo vai ser dividido

2.3.2. Algoritmo/Função

Algoritmo:

1º Calcular o h (passo).

Nota: O h irá dar a amplitude da cada partição do intervalo $[a, b]$, quanto menor o h menor o erro.

$$h = \frac{b - a}{n}$$

2º Criar um vetor, por hipótese y e atribuir ao $y(1)$ a condição inicial y_0 .

$$y(1) = y_0$$

3º Cálculo dos parâmetros k_1 e k_2 .

$$k_1 = hf(t_i, y_i)$$

$$k_2 = hf(t_{i+1}, y_i + k_1)$$

4º Calcular os $y(n)$ seguintes, seguindo o método de Runge-Kutta de ordem 2, até à n iteração.

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

Pseudocódigo:

$$h \leftarrow \frac{b-a}{n}$$

$$t \leftarrow a:h:b$$

$$y(1) \leftarrow y_0$$

Para $i = 1$ até n

$$k_1 = hf(t_i, y_i)$$

$$k_2 = hf(t_{i+1}, y_i + k_1)$$

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

Fim para

Função RK2 (MatLab):

```
function y = NRK2(f,a,b,n,y0)

    h = (b-a)/n;           % Cálculo do passo
    t = a:h:b;             % Alocação de memória
    y = zeros(1,n+1);      % Alocação de memória
    y(1) = y0;             % Atribuição do valor y0 ao primeiro índice do vetor y

    for i = 1:n             % Ciclo com n iterações
        k1=h*f(t(i),y(i));  % Parâmetro k1
        k2=h*f(t(i+1),y(i)+ k1); % Parâmetro k2
        y(i+1) = y(i)+1/2*(k1 + k2); % Cálculo do método RK2 até n
    end                    % Fim do ciclo FOR
end                        % Indicação do fim da função
```

2.4. Método de RK4

O Método de Runge-Kutta de ordem 4 (RK4) é método numérico de passo simples, que depende de uma função calculada em diferentes pontos.

2.4.1. Fórmula

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

$$i = 0, 1, \dots, n - 1$$

Onde:

$y_{i+1} \rightarrow$ Próximo valor a calcular

$y_i \rightarrow$ Valor aproximado na abcissa atual

$f(t_i, y_i) \rightarrow$ Valor da equação na abcissa t_i e ordenada y_i

$$k_1 = hf(t_i, y_i)$$

$$k_2 = hf\left(t_i + \frac{h}{2}, y_i + \frac{1}{2}k_1\right)$$

$$k_3 = hf\left(t_i + \frac{h}{2}, y_i + \frac{1}{2}k_2\right)$$

$$k_4 = hf(t_{i+1}, y_i + k_3)$$

$$h = \frac{b - a}{n} \rightarrow \text{Amplitude de cada subintervalo (PASSO)}$$

$[a, b] \rightarrow$ Intervalo pretendido

$n \rightarrow$ número de vezes, que o intervalo vai ser dividido

2.4.2. Algoritmo/Função

Algoritmo:

1º Calcular o h (passo).

Nota: O h irá dar a amplitude da cada partição do intervalo $[a, b]$, quanto menor o h menor o erro.

$$h = \frac{b - a}{n}$$

2º Criar um vetor, por hipótese y e atribuir ao $y(1)$ a condição inicial y_0 .

$$y(1) = y_0$$

3º Calcular os diferentes parâmetros k (k_1, k_2, k_3, k_4), pela ordem indicada.

$$k_1 = hf(t_i, y_i)$$

$$k_2 = hf\left(t_i + \frac{h}{2}, y_i + \frac{1}{2}k_1\right)$$

$$k_3 = hf\left(t_i + \frac{h}{2}, y_i + \frac{1}{2}k_2\right)$$

$$k_4 = hf(t_{i+1}, y_i + k_3)$$

4º Calcular os $y(n)$ seguintes, seguindo o método de Runge-Kutta de ordem 4, até à n iteração.

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

Pseudocódigo:

$$h \leftarrow \frac{b-a}{n}$$

$$t \leftarrow a:h:b$$

$$y(1) \leftarrow y_0$$

Para $i = 1$ até n

$$k_1 = hf(t_i, y_i)$$

$$k_2 = hf\left(t_i + \frac{h}{2}, y_i + \frac{1}{2}k_1\right)$$

$$k_3 = hf\left(t_i + \frac{h}{2}, y_i + \frac{1}{2}k_2\right)$$

$$k_4 = hf(t_{i+1}, y_i + k_3)$$

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

Fim para

Função RK4 (MatLab):

```
function y = NRK4(f,a,b,n,y0)

h = (b-a)/n;           % Cálculo do passo
t = a:h:b;             % Alocação de memória
y = zeros(1,n+1);      % Alocação de memória
y(1) = y0;             % Atribuição do valor y0 ao primeiro índice do vetor y

for i =1:n
    k1=h*f(t(i),y(i)); % Parâmetro k1
    k2=h*f(t(i)+(h/2),y(i)+(k1/2)); % Parâmetro k2
    k3=h*f(t(i)+(h/2),y(i)+(k2/2)); % Parâmetro k3
    k4=h*f(t(i+1),y(i)+k3); % Parâmetro k4

    y(i+1) = y(i)+1/6*(k1+(2*k2)+(2*k3)+k4); % Cálculo do método RK4 até n
end % Fim do ciclo FOR
end % Indicação do fim da função
```

2.5. Função ODE45 do Matlab

O Função ODE45, originária do *MatLab* é uma função com melhor aproximação à solução exata.

2.5.1.Fórmula

$$[t, y] = ODE45(f, t, y0)$$

Onde:

$t \rightarrow$ Vetor das abcissas

$f \rightarrow$ Equação diferencial

$y0 \rightarrow$ Valor inicial do PVI

2.5.2. Algoritmo/Função

Algoritmo:

1º Calcular o h (passo).

Nota: O h irá dar a amplitude da cada partição do intervalo $[a, b]$, quanto menor o h menor o erro.

$$h = \frac{b - a}{n}$$

2º Aproximação através da função ODE45

Função ODE (MatLab):

```
function y = NODE45(f,a,b,n,y0)

    h = (b-a)/n;           % Cálculo do passo
    t = a:h:b;             % Alocação de memória
    [~,y] = ode45(f,t,y0); % Cálculo através da função ODE45 até n
    y = y.';               % Equivalente a fazer uma transposta de um vetor
end
```

2.6. Método Adams-Bashforth (ordem4)

O Método de Adams-Bashforth é um método numérico um pouco diferente dos outros métodos, porque é de passo múltiplo. que depende de outro método da mesma ordem, para calcular os pontos iniciais, para que depois o método Adams, possa calcular os restantes.

2.6.1.Fórmula

$$y_{i+1} = y_i + \frac{h}{24} (55f_i - 59f_{i-1} + 37f_{i-2} - 90f_{i-3})$$

Onde:

$y_{i+1} \rightarrow$ Próximo valor a calcular

$y_i \rightarrow$ Valor aproximado na abcissa atual

$f_i = f(t_i, y_i)$

$f_{i-1} = f(t_{i-1}, y_{i-1})$

$f_{i-2} = f(t_{i-2}, y_{i-2})$

$f_{i-3} = f(t_{i-3}, y_{i-3})$

$h = \frac{b-a}{n} \rightarrow$ Amplitude de cada subintervalo (PASSO)

$[a, b] \rightarrow$ Intervalo pretendido

$n \rightarrow$ número de vezes, que o intervalo vai ser dividido

2.6.2. Algoritmo/Função

Algoritmo:

1º Calcular o h (passo).

Nota: O h irá dar a amplitude da cada partição do intervalo $[a, b]$, quanto menor o h menor o erro.

$$h = \frac{b - a}{n}$$

2º Criar um vetor, por hipótese y e atribuir ao $y(1)$ a condição inicial y_0 .

$$y(1) = y_0$$

3º Calcular os diferentes parâmetros k (k_1, k_2, k_3, k_4), pela ordem indicada.

$$k_1 = hf(t_i, y_i)$$

$$k_2 = hf\left(t_i + \frac{h}{2}, y_i + \frac{1}{2}k_1\right)$$

$$k_3 = hf\left(t_i + \frac{h}{2}, y_i + \frac{1}{2}k_2\right)$$

$$k_4 = hf(t_{i+1}, y_i + k_3)$$

4º Calcular os 4 y seguintes, seguindo o método de Runge-Kutta de ordem 4.

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

5º Com os 4 pontos iniciais, calculamos os restantes pontos até à n iteração, seguindo o método de Adams-Bashforth de ordem 4.

$$y_{i+1} = y_i + \frac{h}{24}(55f_i - 59f_{i-1} + 37f_{i-2} - 9f_{i-3})$$

Pseudocódigo:

$$h \leftarrow \frac{b - a}{n}$$

$$t \leftarrow a: h: b$$

$$y(1) \leftarrow y_0$$

$$n1 \leftarrow 3$$

Para $i = 1$ até $n1$

$$k_1 = hf(t_i, y_i)$$

$$k_2 = hf\left(t_i + \frac{h}{2}, y_i + \frac{1}{2}k_1\right)$$

$$k_3 = hf\left(t_i + \frac{h}{2}, y_i + \frac{1}{2}k_2\right)$$

$$k_4 = hf(t_{i+1}, y_i + k_3)$$

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

Fim para

Para $i = 4$ até n

$$y_{i+1} = y_i + \frac{h}{24}(55f(i) - 59f(i - 1) + 37f(i - 2) - 9f(i - 3))$$

Fim para

Função Adams-Bashforth (MatLab):

```
function y = NAdams(f,a,b,n,y0)

    h = (b-a)/n;      % Cálculo do passo
    t = a:h:b;        % Alocação de memória

    % Para iniciar: Runge-Kutta de ordem 4
    b1 = a+3*h;        % Determinação do último valor do intervalo para RK4
    n1 = 3;            % número de partições para calcular em RK4
    yNAux = NAux(f,a,b1,n1,y0); %Função auxiliar que calcular os primeiros 4 valores pelo RK4
    y = yNAux;         % Atribui os valores calculados em yNAux ao vetor y

    % Método de Adams - Bashforth de ordem 4(Passo múltiplo)
    for i=4:n
        y(i+1)=y(i)+ (h/24) *(55*f(t(i),y(i)) - 59*f(t(i-1),y(i-1)) + 37*f(t(i-2),y(i-2)) - 9*f(t(i-3),y(i-3)));
    end
end
```

3. Exemplos de aplicação e teste dos métodos

3.1. Exercício 3 do Teste3 2015/2016

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

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.

$$\begin{cases} y' = -2ty & \text{Equação diferencial} \\ t \in [0, 1.5] & \text{Intervalo} \\ y(0) = 2 & \text{Condição inicial} \end{cases}$$

Iremos apresentar três formas, para verificar se a função proposta é a solução exata do problema.

1º teste: Consiste em verificar se a função proposta, satisfaz a condição inicial.

$$y(0) = 2 \rightarrow \text{Condição inicial}$$

$$y(t) = 2e^{-t^2} \rightarrow \text{Solução exata proposta}$$

Então se substituirmos o t por 0 na solução proposta, terá que dar 2, para ser a solução exata do problema:

$$y(0) = 2e^{-0^2} \leftrightarrow y(0) = 2e^0 \leftrightarrow y(0) = 2$$

Verificamos, que ao substituirmos o t por 0 na expressão, efetivamente é igual a 2.

2º teste: Consiste em derivar a solução exata proposta e verificar se a derivada é igual à expressão do PVI.

Derivar solução proposta $\rightarrow y(t) = 2e^{-t^2}$

$$y' = (2e^{-t^2})' \leftrightarrow y' = 2(-2t)e^{-t^2} \leftrightarrow y' = -4te^{-t^2}$$

Derivada da solução proposta do problema

Então: Será que a derivada que calculamos acima verifica a expressão do PVI.

$$y' = -2ty ?$$

Vamos verificar a igualdade:

$$-2ty = -2ty \leftrightarrow -2ty = -2t(2e^{-t^2}) \leftrightarrow -2ty = -4te^{-t^2}$$

$-2ty = y'$

Verificamos, que efetivamente a solução exata proposta está correta, a derivada é igual.

3º teste: Consiste em verificar se é a solução exata, se resolvermos efetivamente a equação diferencial.

Então: Como é uma equação diferencial linear de 1º ordem, temos que calcular o fator integrante.

Fator integrante: $e^{\int 2t \, dt} = e^{t^2}$

Resolução da equação diferencial:

$$\begin{aligned}
 y' + 2ty &= 0 \quad \Leftrightarrow \\
 \Leftrightarrow y'e^{t^2} + 2te^{t^2}y &= 0 \quad \Leftrightarrow \\
 \Leftrightarrow \int (y_e t^2)' &= \int 0 \, dt \quad \Leftrightarrow \\
 \Leftrightarrow y_e t^2 &= 0 + c \quad \Leftrightarrow \\
 \Leftrightarrow y &= 0 + C e^{-t^2} \quad \Leftrightarrow \\
 \Leftrightarrow y = c e^{-t^2} &\quad \rightarrow \quad \text{Solução geral}
 \end{aligned}$$

Regra da multiplicação das derivadas (indicated by arrows pointing to the boxed equations in the derivation)

Chegámos à solução geral, agora com o valor inicial, podemos encontrar a solução particular do PVI.

Solução particular do PVI:

$$y(0) = 2 \rightarrow y(0) = c e^{-0^2}$$

$$\begin{aligned}
 y(0) &= c e^{-0^2} \quad \Leftrightarrow \\
 \Leftrightarrow 2 &= c e^{-0^2} \quad \Leftrightarrow \\
 \Leftrightarrow 2 &= c e^0 \quad \Leftrightarrow \\
 \Leftrightarrow 2 &= c
 \end{aligned}$$

Solução particular:

$$y = 2e^{-t^2}$$

Então sim a solução proposta é a correta.

3.1.2.Exemplos de output – App com gráfico e tabela

(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 | $ y(t_i) - y_i $ Euler | $ y(t_i) - y_i $ RK2 |
| 0 | 0 | 2 | | | 0 | 0 |
| 1 | | 1.5576 | | 1.5000 | | 0.0576 |
| 2 | 1 | | | | | 0.0142 |
| 3 | 1.5 | 0.2108 | | 0.3750 | | |

No seguimento do tópico 3.1.2, iremos utilizar a nossa APP, para construirmos a tabela, para isso, necessitamos de saber a expressão derivada do PVI que já sabemos, o intervalo $[a, b]$ que também sabemos implicitamente da tabela, o n que não sabemos e o valor inicial y_0 .

- Expressão derivada do PVI, que sabemos $\rightarrow y' = -2ty$;
- O intervalo $[a, b]$, que também sabemos implicitamente da tabela $[0, 1.5]$;
- O n , que não sabemos, mas podemos calcular pela equação:

$$h \rightarrow 0.5$$

$$a \rightarrow 0$$

$$b \rightarrow 1.5$$

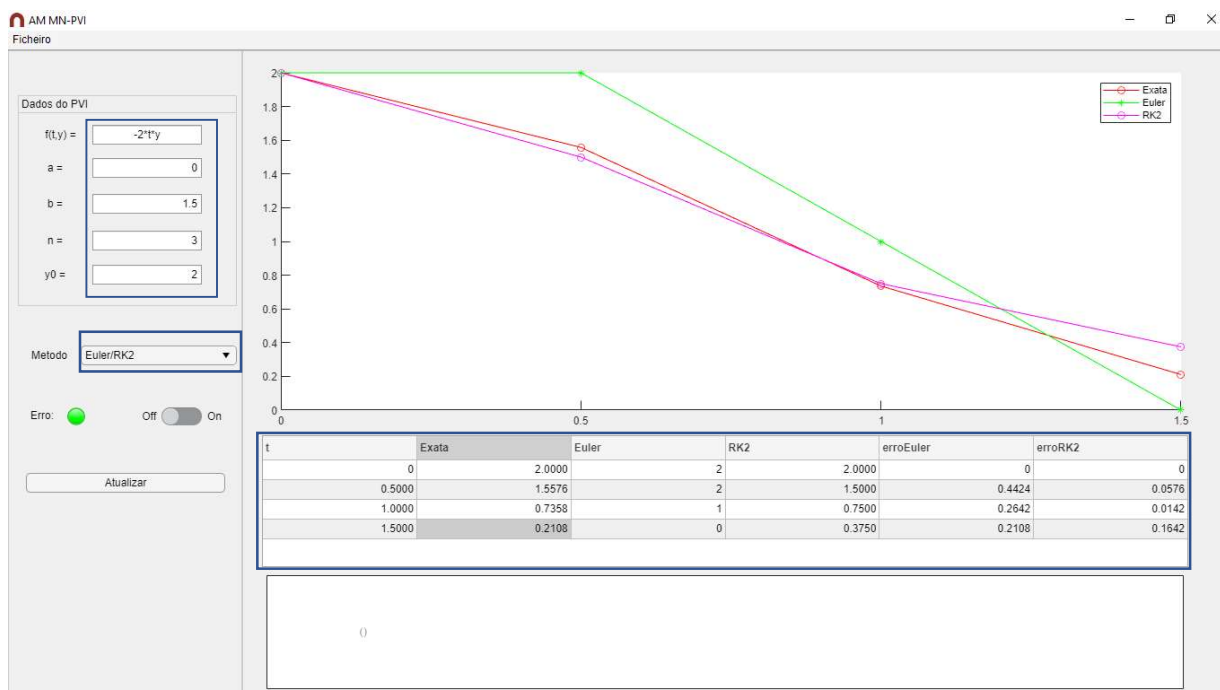
$$h = \frac{b - a}{n}$$

$$0,5 = \frac{1,5 - 0}{n}$$

$$n = \frac{1,5}{0,5} = 3$$

- Valor inicial, que é $\rightarrow y(0) = 2$

Solução exata e aplicação do método Euler e do RK2:



De Acordo com a imagem acima a tabela fica:

| Aproximações | | | | | Erros | |
|--------------|-------|-------------------|----------------|--------------|---------------------------|-------------------------|
| i | t_i | $y(t_i)$ Exata | y_i Euler | y_i RK2 | $ y(t_i) - y_i $ Euler | $ y(t_i) - y_i $ RK2 |
| 0 | 0 | 2 | 2 | 2 | 0 | 0 |
| 1 | 0.5 | 1.5576 | 2 | 1.5000 | 0.4424 | 0.0576 |
| 2 | 1 | 0.7358 | 1 | 0.7500 | 0.2642 | 0.0142 |
| 3 | 1.5 | 0.2108 | 0 | 0.3750 | 0.2108 | 0.1642 |

(c) Qual das figuras seguintes representa graficamente uma solução do PVI dado? Justifique a sua resposta.

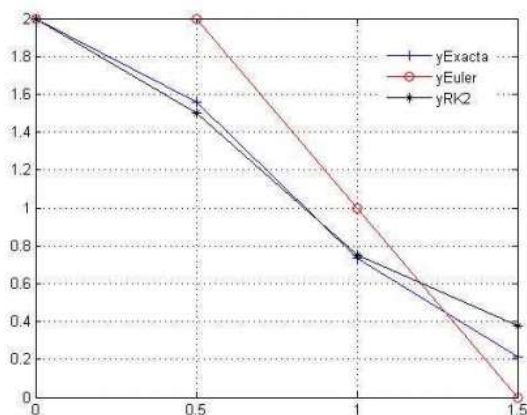


Figura 4

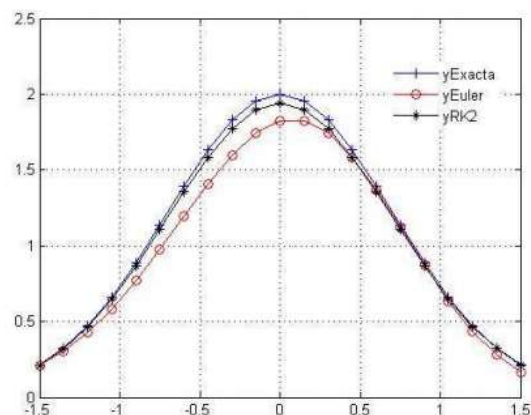
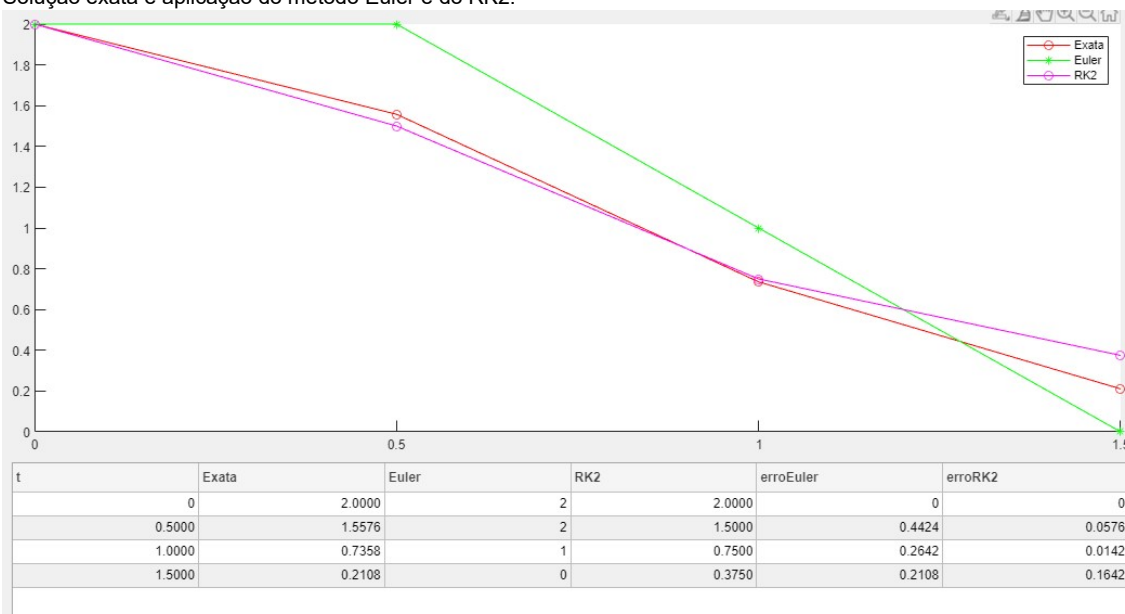


Figura 5

(d) Estabeleça um PVI cuja solução em modo gráfico coincida com a figura que excluiu na alínea anterior.

De acordo com o gráfico que obtivemos na nossa APP, que se apresenta abaixo, a figura 4 corresponde ao PVI dado.

Solução exata e aplicação do método Euler e do RK2:



A figura que excluímos foi a figura 5, porque efetivamente pertence a outro PVI.

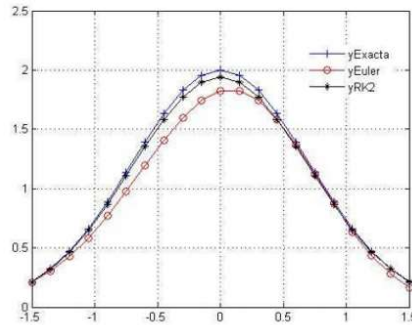


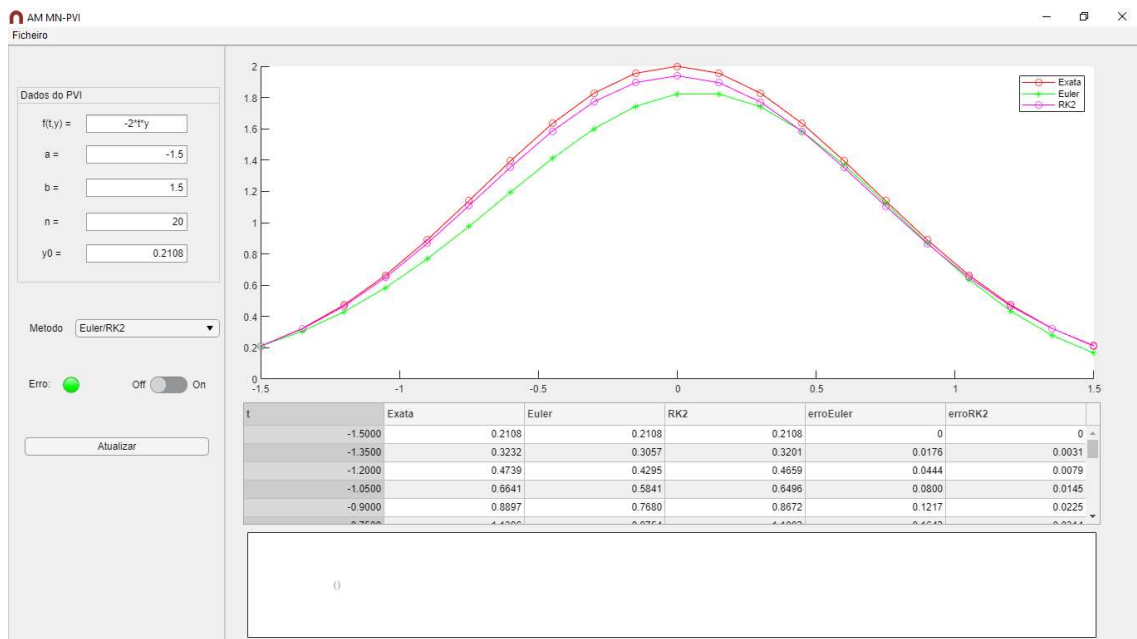
Figura 5

Da figura conseguimos perceber que o intervalo que está a ser utilizado é $[-1.5, 1.5]$, o numero de sub-intervalos são igual a 20, e se calcularmos o $y(-1.5)$ obtemos o valor inicial. **Nota:** Tem a mesma equação diferencial.

$$y = 2e^{-t^2} \rightarrow y(-1.5) = 2e^{-(1.5)^2} \rightarrow y(-1.5) \cong 0.2108$$

Então ficamos com:

$$\begin{cases} y' = -2ty \\ t \in [-1.5, 1.5] \\ y(0) = 0.2108 \end{cases}$$



(e) Quais dos comandos seguintes em GeoGebra lhe permitiriam determinar a solução exata do PVI e a solução aproximada do mesmo.

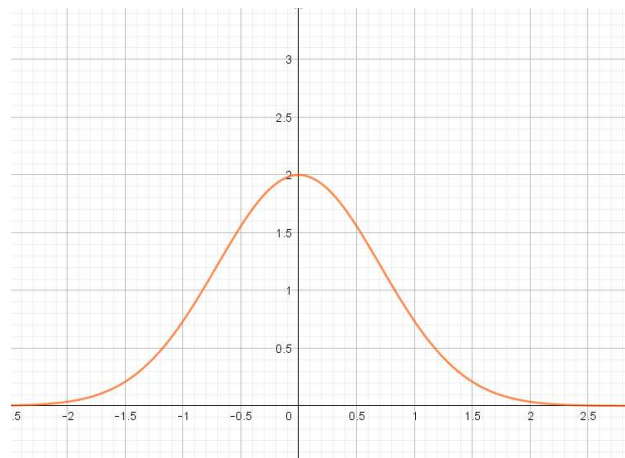
(A) `SolveODE[-2xy, (0,2)]`

(B) `SolveODE[-2xy, (-1.5,0.2108)]`

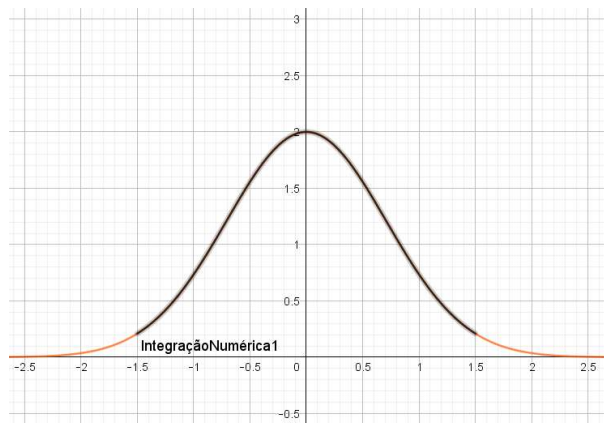
(C) `NSolveODE[{-2xy}, 0, {2}, 1.5]`

(D) `NSolveODE[{-2xy}, -1.5, {0.2108}, 1.5]`

O comando que permitiu determinar a solução exata foi o comando (A) `SolveODE(-2 x y, (0, 2))`, porque `SolveODE(<f(x, y)>, <Ponto em f>)`



O comando que permitiu determinar a solução aproximada foi o comando (D) `NSolveODE({-2 x y}, -1.5, {0.2108}, 1.5)`, porque `NSolveODE(<Lista de Derivadas>, <x Inicial>, <Lista de y Inicial>, <x Final>)`.



3.2. Problemas de aplicação do livro

3.2.1. Modelação matemática do problema

Exercício 1.

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 $g = 32 \text{ ft/s}^2$.

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

O Problema 1 evidentemente, trata-se de um Problema de valor inicial. Sabemos a expressão derivada, o valor inicial, o intervalo e outros valores importantes como a constante de aceleração gravitacional, a massa do corpo e uma constante k .

Um primeiro passo será simplificar a expressão do PVI, ficando na forma explícita.

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

$$m \frac{dv}{dt} = mg - kv^2$$

$$\frac{dv}{dt} = \frac{mg - kv^2}{m}$$

$$\frac{dv}{dt} = g - \frac{kv^2}{m}$$

Substituindo pelos valores mencionados:

$$k = 0.125$$

$$m = 5 \text{ slugs}$$

$$g = 32 \text{ ft/s}^2$$

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

$$\begin{cases} v' = 32 - 0.125v^2 \\ t \in [0, 5] \\ v(0) = 0 \end{cases}$$

Exercício 2.

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.

| $t(\text{days})$ | 1 | 2 | 3 | 4 | 5 |
|--------------------------|------|-------|-------|-------|-------|
| $A(\text{observed})$ | 2.78 | 13.53 | 36.30 | 47.50 | 49.40 |
| $A(\text{approximated})$ | | | | | |

- (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 $A(5)$.

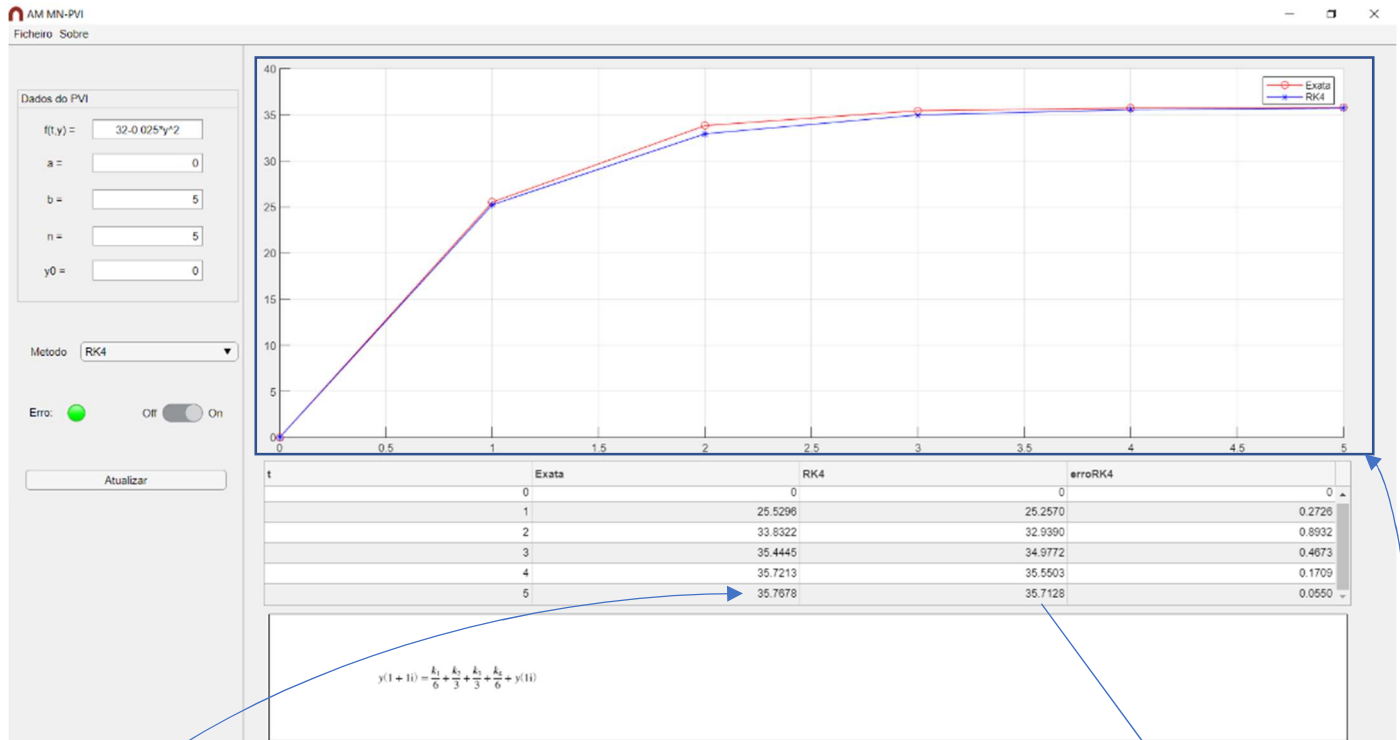
O Problema 2 também, trata-se de um Problema de valor inicial, onde sabemos a expressão derivada, valor inicial e o intervalo.

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

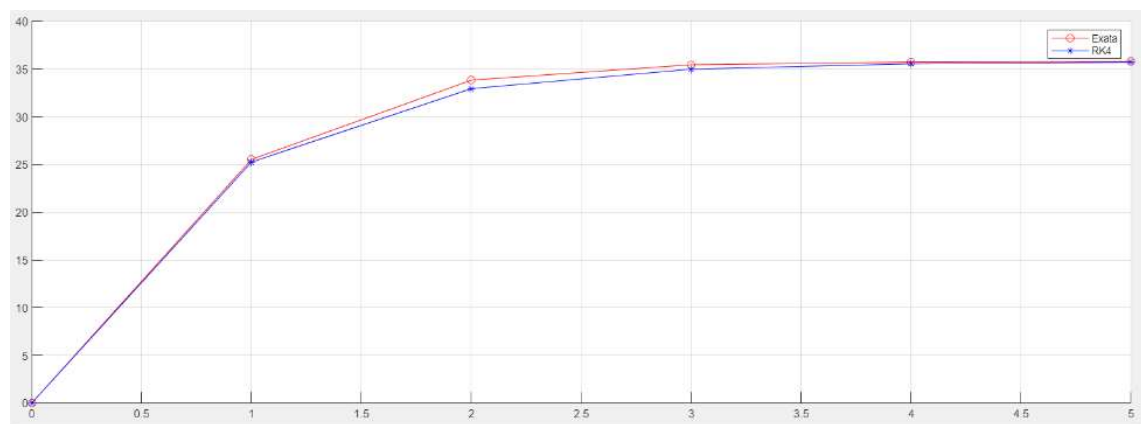
$$\begin{cases} A' = A(2.128 - 0.0432A) \\ t \in [0, 5] \\ A(0) = 0.24 \text{ cm}^2 \end{cases}$$

3.2.2. Resolução através da App desenvolvida

Exercício 1 – Resolução

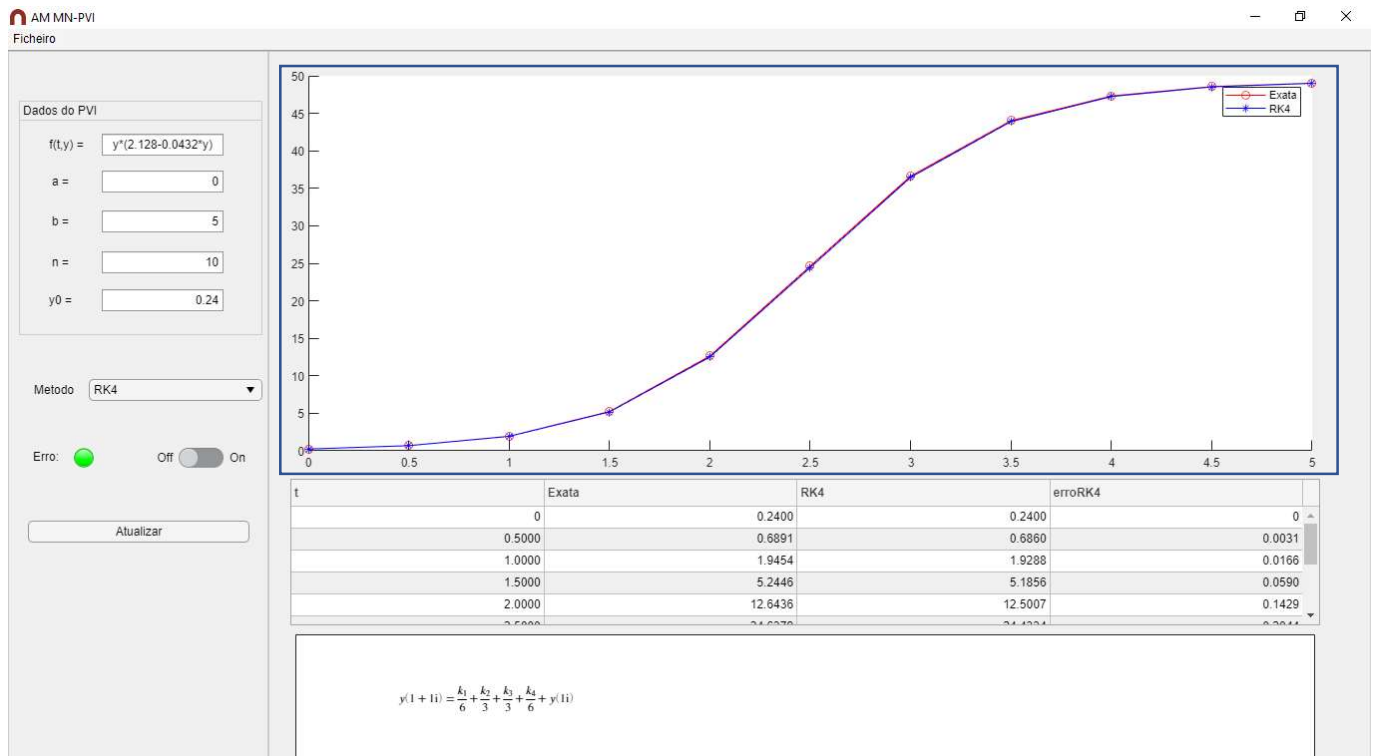


(a) A aproximação da velocidade da massa em queda em $t = 5s$ é de **35.7128**
(b)



(c) O valor real/exato de $v(5) = 35.7678$

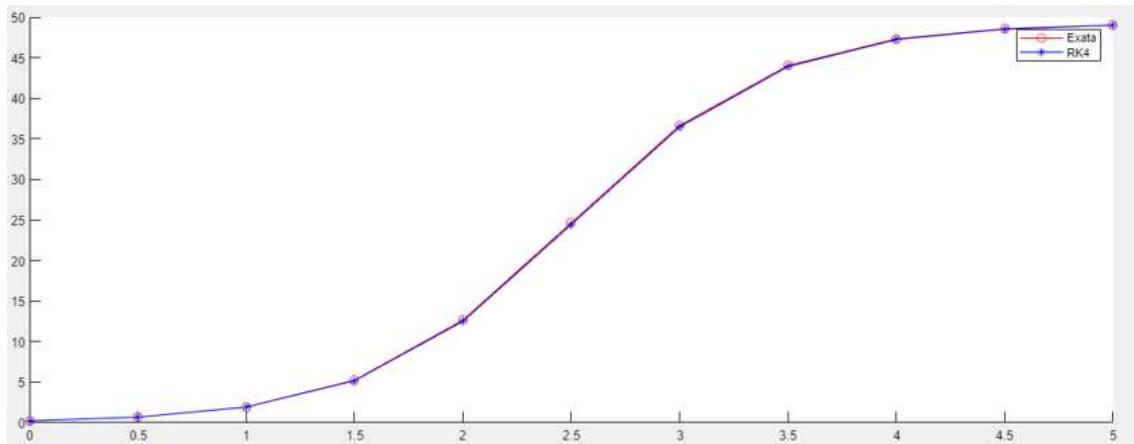
Exercício 2 – Resolução



(a)

| | | | | | | |
|-----------------|------|------|-------|-------|-------|-------|
| t(days) | 0 | 1 | 2 | 3 | 4 | 5 |
| A(observed) | 0.24 | 2.78 | 13.53 | 36.30 | 47.50 | 49.40 |
| A(approximated) | 0.24 | 1.93 | 12.50 | 36.46 | 47.23 | 49.00 |

(b)



| | | t | Exata | RK4 | erroRK4 |
|--------|----|--------|---------|---------|---------|
| A(0) = | 1 | 0 | 0.2400 | 0.2400 | 0 |
| | 2 | 0.5000 | 0.6891 | 0.6860 | 0.0031 |
| A(1) = | 3 | 1 | 1.9454 | 1.9288 | 0.0166 |
| | 4 | 1.5000 | 5.2446 | 5.1856 | 0.0590 |
| A(2) = | 5 | 2 | 12.6436 | 12.5007 | 0.1429 |
| | 6 | 2.5000 | 24.6379 | 24.4334 | 0.2044 |
| A(3) = | 7 | 3 | 36.6283 | 36.4618 | 0.1665 |
| | 8 | 3.5000 | 44.0210 | 43.9020 | 0.1189 |
| A(4) = | 9 | 4 | 47.3164 | 47.2349 | 0.0814 |
| | 10 | 4.5000 | 48.5710 | 48.5245 | 0.0465 |
| A(5) = | 11 | 5 | 49.0196 | 48.9965 | 0.0231 |

(c) $A(0) = 0.2400$

$A(1) = 1.9454$

$A(2) = 12.6436$

$A(3) = 36.6283$

$A(4) = 47.3164$

$A(5) = 49.0196$

Solução exata

4. Conclusão

Em suma, os métodos numéricos, têm uma enorme importância no que diz respeito à resolução de Problemas de Valor Inicial, pois permitem aproximações à solução exata com um erro relativamente pequeno, obviamente tendo em consideração o contexto em que estamos a aplicá-los.

Note-se que, se quisermos um erro ainda menor então a solução será reduzir a amplitude dos subintervalos, ou seja repartir o intervalo principal em mais partes, geralmente, iguais.

No entanto existe diferença, se aplicarmos métodos diferentes, por exemplo, se aplicarmos o método de Euler, que irá apresentar mais erro em relação aos outros e verificamos que a função ODE45 do MatLab, consegue aproximações muito boas à solução exata.

5. Referências Bibliográficas

<https://lusoacademia.org/2015/11/19/1-introducao-as-equacoes-diferenciais/>

https://www.ufrgs.br/reatmat/CalculoNumerico/livro-oct/pdvi-metodo_de_adams-bashforth.html

<http://www.mat.uc.pt/~amca/MPII0607/folha3.pdf>