Ano Letivo: 2022/2023

Atividade 03-Métodos Numéricos para EDO/PVI

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 no estudo de Métodos Numéricos para a resolução de Equações Diferenciais Ordinárias (EDOs) e de Problemas de Valor inicial (PVIs), e na implementação desses métodos através do desenvolvimento de uma app em linguagem de programação MATLAB.

Para uma melhor familiarização com estes conteúdos, incluímos exemplos de aplicação e testes dos métodos numéricos analisados.

1.1-Equação diferencial: definição e propriedades

As equações diferenciais são aquelas que contêm as derivadas de uma ou mais variáveis dependentes em relação a uma ou mais variáveis independentes.

As equações diferenciais são classificadas quanto ao **tipo**, **ordem** e **linearidade**.

- Quanto ao tipo as equações diferenciais são classificadas em: ordinárias e parciais. Equações diferenciais ordinárias (EDO) são aquelas que contêm uma ou mais derivadas de variáveis dependentes em relação a uma variável independente. As equações diferenciais parciais (EDP) são aquelas que envolve as derivadas parciais de uma ou mais variáveis dependentes em relação a uma ou mais variáveis independentes.

- Quanto a ordem uma equação diferencial pode ser de 1ª, 2ª,…,n-ésima ordem dependendo da derivada de maior ordem presente na equação. Uma equação ordinária de ordem n pode ser escrita na forma:

$$F(t, y, y', y''\ldots y^{(n)}) = 0$$

- Quanto a linearidade de uma equação diferencial ela pode ser linear e não linear. Ela é linear se as incógnitas e suas derivadas aparecem de forma linear. Por exemplo uma equação diferencial ordinária de ordem n é uma equação que pode ser escrita como:

$$a_n(x)\frac{d^n y}{dx^n} + a_{n-1}(x)\frac{d^{n-1} y}{dx^{n-1}} + \ldots + a_1(x)\frac{dy}{dx} + a_0(x)y = g(x)$$

  As equações diferenciais ordinárias que não podem ser escritas nessa forma são não lineares.

1.2-Definição de PVI

Um PVI significa Problema de Valor Inicial e é um conceito da matemática aplicada e da análise numérica. Em termos simples, um PVI é uma equação diferencial que descreve o comportamento de uma função desconhecida, juntamente com uma condição inicial que especifica o valor da função em um ponto específico. O objetivo do problema é encontrar uma solução que satisfaça a equação diferencial e a condição inicial. Os PVIs podem ser resolvidos de forma exata ou aproximada.

Um problema de valor inicial (PVI) é uma equação diferencial da forma:

$$\begin{cases} y' = f(t, y) \\ t \,\epsilon\, [a, b] \\ y(a) = y_0 \end{cases}$$

2.Métodos Numéricos para a resolução de PVI

Existem vários métodos numéricos para a resolução de PVI entre eles:

- Método de Euler: Este é o método mais simples e direto para resolver um PVI. Ele usa uma aproximação linear para a solução e é fácil de implementar. No entanto, a precisão deste método é limitada e pode gerar erros significativos em soluções complexas.
- Método de Runge-Kutta: Este é um método iterativo que usa várias estimativas de ordem superior para melhorar a precisão da solução. O método de Runge-Kutta é mais preciso do que o método de Euler, mas é mais complexo de implementar.

- Método de Adams-Bashforth: Este método é uma abordagem de múltiplas etapas que usa as soluções anteriores para estimar a solução atual. Ele tem uma precisão moderada e é fácil de implementar.

- O método de Heun, também conhecido como método de Euler moderado, é um método numérico para resolver problemas de valor inicial (PVI) de equações diferenciais ordinárias (EDOs). Ele é uma melhoria do método de Euler, que consiste em aproximar a solução da EDO por meio de uma reta tangente ao ponto inicial.

Esses são apenas alguns dos métodos disponíveis para resolver PVI. A escolha do método a ser usado depende da complexidade da equação diferencial e das condições iniciais, bem como dos requisitos de precisão e eficiência computacional.

Neste trabalho iremos abordar os métodos de **Euler**, **Euler Moderado**, **Runge-Kutta de ordem 2 e 4**, a **função ODE45** e ainda o **Ponto Médio**.

2.1-Método de Euler

O método de Euler é um dos métodos numéricos mais simples e amplamente utilizados para resolver problemas de valor inicial (PVI) de equações diferenciais ordinárias (EDOs). Ele consiste em aproximar a solução da EDO por meio de uma reta tangente ao ponto inicial, 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

A fórmula do Método de Euler para resolver um PVI é:

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

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;

2.1.2-Algoritmo/Função

Algoritmo:

- Ler f, a, b, y0, n;
- Calcular h (h =(b-a)/n);
- t = a:h:b;
- y = y0;
- Para i de 1 até n fazer
  y=y+h*f(t,y);
- Escrever y;

Função (MATLAB):

```
function yEuler= Euler(f,a,b,n,y0)

h = (b-a)/n;                % Tamanho de cada subintervalo (passo)
yEuler=zeros(1,n+1);        % Alocação de memória - vetor das ordenadas
t=a:h:b;                    % Alocação de memória - vetor das abcissas
yEuler(1) = y0;             % O primeiro valor de y é sempre y0

    for i=1:n                   % O número de iterações vai ser igual a n

    yEuler(i+1)=yEuler(i)+h*f(t(i),yEuler(i)); % Aproximação do método de
                                               %Euler para a iésima iteração
    end

end
```

2.2 Método de Euler Moderado

O método de Euler moderado, ou método de Heun, é um método numérico para resolver equações diferenciais ordinárias (EDOs) de primeira ordem. 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

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

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 para calcular k1

$$\boxed{k1 = f(t_i, y_i)}$$

Legenda:

- k_1 = Inclinação no início do intervalo
- $f(t_i, y_i)$ = Valor da equação em t_i e y_i;

Fórmula para calcular k2

$$\boxed{k2 = f(t_{t+1},\ y_i + k1*h)}$$

Legenda:

- k_2 = Inclinação no fim do intervalo;
- t_{i+1} = Valor da abcissa seguinte;
- y_i = Valor aproximado da solução do problema original na abcissa atual;
- k_1 = Inclinação no início do intervalo;
- h = Valor de cada subintervalo (passo);

2.2.2-Algoritmo/Função

Algoritmo:

- Ler f, a, b, n, y0;
- Calcular h (h =(b-a)/n);
- t = a:h:b;
- y = y0;
- Para i de 1 até n fazer:

  $k1 = f(t_i,y_i)$;

  $k2 = f(t_{t+1}, y_i + k1*h)$;

  $y_{i+1}= y_i + h/2*(k1+k2)$;

- Escrever y_{i+1}

Função (MATLAB):

```
function yEulerM=EulerM(f,a,b,n,y0)

h = (b-a)/n;                 % Tamanho de cada subintervalo (passo)
yEulerM=zeros(1,n+1);        % Alocação de memória - vetor das ordenadas
t=a:h:b;                     % Alocação de memória - vetor das abcissas
yEulerM(1) = y0;             % O primeiro valor de y é sempre y0

    for i=1:n                            % O número de iterações vai ser igual a n
        k1 = f(t(i),yEulerM(i));           % Inclinação no início do intervalo
        k2 = f(t(i+1), yEulerM(i) + k1*h); % Inclinação no fim do intervalo

        yEulerM(i+1)=yEulerM(i)+h/2*(k1+k2);  % Próximo valor aproximado da
                                              % solução do problema original


    end

end
```

2.3 Método de Runge-Kutta de 2ª Ordem

O método de Runge-Kutta de 2ª ordem é um algoritmo numérico utilizado para aproximar soluções de equações diferenciais ordinárias de primeira ordem. Ele utiliza duas estimativas para calcular a solução em cada passo de tempo, resultando numa precisão maior do que a do método de Euler.

2.3.1-Fórmulas

Fórmula Geral

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

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)*;
- k_1 = Inclinação no início do intervalo;
- k_2 = Inclinação no fim do intervalo;

Fórmula para calcular k1

$$\boxed{k1 = h * f(t_i, y_i)}$$

Legenda:

- k_1 = Inclinação no início do intervalo
- $f(t_i, y_i)$ = Valor da equação em t_i e y_i;
- h = Valor de cada subintervalo (passo);

Fórmula para calcular k2

$$\boxed{k2 = h * f(t_{t+1}, y_i + k1)}$$

Legenda:

- k_2 = Inclinação no fim do intervalo;
- t_{i+1} = Valor da abcissa seguinte;
- y_i = Valor aproximado da solução do problema original na abcissa atual;
- k_1 = Inclinação no início do intervalo;
- h = Valor de cada subintervalo (passo);

2.3.2-Algoritmo/Função

Algoritmo:

- Ler f, a, b, n, y0;
- Calcular h (h =(b-a)/n);
- t= a:h:b;
- y = y0;
- Para i de 1 até n fazer:

  $k1 = h*f(t_i,y_i)$;

  $k2 = h* f(t_{t+1}, y_i + k1)$;

  $y_{i+1}= y_i + \frac{1}{2}*(k1+k2)$;

- Escrever y_{i+1}

Função (MATLAB):

```
function yRK2=RK2(f,a,b,n,y0)

h = (b-a)/n;                     % Tamanho de cada subintervalo (passo)
yRK2=zeros(1,n+1);               % Alocação de memória - vetor das ordenadas
t=a:h:b;                         % Alocação de memória - vetor das abcissas
yRK2(1) = y0;                    % O primeiro valor de y é sempre y0

    for i = 1:n                      % O número de iterações vai ser igual a n

    k1 = h*f(t(i), yRK2(i));             % Inclinação no início do intervalo
    k2 = h*f(t(i+1), yRK2(i) + k1);      % Inclinação no fim do intervalo

    yRK2(i+1) = yRK2(i) + 1/2*(k1 + k2);     % Próximo valor aproximado da
                                             % solução do problema original

    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 aproximar soluções de equações diferenciais ordinárias, que utiliza quatro estimativas intercaladas para calcular a solução em cada passo de tempo. O método de Runge-Kutta de 4ª ordem é mais preciso do que o método de Runge-Kutta de 2ª ordem para a aproximação da solução numérica de equações diferenciais ordinárias.

2.4.1-Fórmulas

Fórmula Geral

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

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

Fórmulas para calcular k1,k2,k3,k4

$$k1 = h * f(t_i, y_i)$$

$$k2 = h * f(t_i + \frac{h}{2}, y_i + \frac{k1}{2})$$

$$k3 = h * f(t_i + \frac{h}{2}, y_i + \frac{k2}{2})$$

$$k4 = h * f(t_i + h, y_i + k3)$$

2.4.2-Algoritmo/Função

Algoritmo:

- Ler f, a, b, n, y0;
- Calcular h (h =(b-a)/n);
- t=a:h:b;
- y= y0;
- Para i de 1 até n fazer:

  $k1 = h*f(t_i,y_i);$

  $k2 = h* f(t_i+\frac{h}{2}, y_i + \frac{k1}{2});$

  $k3 = h*f(t_i+\frac{h}{2}, y_i + \frac{k2}{2});$

  $k4 = h*f(t_i+h, y_i + k3);$

  $y_{i+1} = y_i + \frac{1}{6}*(k1+2*k2+2*k3+k4);$

- Escrever y_{i+1}

Função (MATLAB):

```
function yRK4 = RK4(f,a,b,n,y0)

h = (b-a)/n;                    % Tamanho de cada subintervalo (passo)
yRK4=zeros(1,n+1);              % Alocação de memória - vetor das ordenadas
t=a:h:b;                        % Alocação de memória - vetor das abcissas

yRK4(1) = y0;                   % O primeiro valor de y é sempre y0

    for i=1:n                   % O número de iterações vai ser igual a n
        k1 = h*f(t(i), yRK4(i));     % Inclinação no início do intervalo

        k2 = h*f(t(i)+h/2, yRK4(i)+k1/2);   % Inclinação no ponto médio do
                                            % intervalo
        k3 = h*f(t(i)+h/2, yRK4(i)+k2/2);   % Inclinação (novamente) no
                                            % ponto médio do intervalo
        k4 = h*f(t(i)+h, yRK4(i)+k3);       % Inclinação no final do intervalo

        yRK4(i+1) = yRK4(i) + 1/6*(k1 + 2*k2 + 2*k3 + k4);
        % Próximo valor aproximado da solução do problema original
    end

end
```

2.5-Função ODE45 do Matlab

A função ode45 nativa do Matlab é um método numérico utilizado para resolver equações diferenciais ordinárias de primeira ordem e segunda ordem, que é um método de passo variável baseado num método de Runge-Kutta.

2.5.1-Fórmulas

$$[t, y] = \mathrm{ode45}(f, t, y0)$$

Legenda:

- t → Vetor das abcissas;
- f → Equação diferencial de t e y;
- y_0 → Condição inicial do PVI (valor inicial de y);

2.5.2-Algoritmo/Função

Algoritmo:

- Ler f, a, b, n, y0;
- Calcular h (h=(b-a)/n);
- t= a:h:b
- y= y0;
- Aproximar as soluções através da função ODE45;

Função (MATLAB):

```
function yODE45=ODE45(f,a,b,n,y0)

h = (b-a)/n;                      % Tamanho de cada subintervalo (passo)
t=a:h:b;                          % Alocação de memória - vetor das abcissa
[~,yODE45] = ode45(f, t, y0);     % Aproximação através da função ODE45 e
                                  % colocação dos valores no vetor y
yODE45 = yODE45';                 % Mudança da orientação do vetor

end
```

2.6-Método do Ponto Médio

O método do Ponto Médio é um método numérico para resolver Equações Diferenciais Ordinárias (ODE).

2.6.1-Fórmulas

Formula Geral

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

Fórmula para calcular k1

$$\boxed{k1 = \frac{1}{2} * f(t_i, y_i)}$$

2.6.2-Algoritmo/Função

Algoritmo:

- Ler f, a, b, n, y0;
- Calcular h (h =(b-a)/n);
- t= a:h:b;
- y= y0;
- Para i de 1 até n fazer:

  $k1 = \frac{1}{2} * f(t_i, y_i)$;

  $y_{i+1} = y_i + h*f(t_i+h/2, y_i + h*k1)$;

- Escrever y_{i+1}

Função:

```
function yPM = PM(f,a,b,n,y0)

h = (b-a)/n;                    % Tamanho de cada subintervalo (passo)
yPM=zeros(1,n+1);               % Alocação de memória - vetor das ordenadas
t=a:h:b;                        % Alocação de memória - vetor das abcissas
yPM(1) = y0;                    % O primeiro valor de y é sempre y0

    for i=1:n
     k1 = 0.5 * f(t(i), yPM(i));        % variável auxiliar
     yPM(i+1) = yPM(i) + h*f(t(i) + h/2, yPM(i) + h*k1);
     % Próximo valor aproximado da solução do problema original

    end
end
```

3.Exemplos de aplicação e teste dos métodos

3.1-Exercício 3 do Teste Farol

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

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

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

1º Passo:

$$\left\{\begin{array}{l} y' = -2ty \\ t \in [0, 1.5] \\ y(0) = 2 \end{array}\right. \longrightarrow \text{PVI}$$

2º Passo:

$$y' = -2ty$$

$$\Leftrightarrow \frac{dy}{dt} = -2ty$$

$$\Leftrightarrow \frac{1}{y}\, dy = -2t\, dt$$

$$\Leftrightarrow \int \frac{1}{y}\, dy = \int -2t\, dt$$

$$\Leftrightarrow \ln|y| = \frac{-\cancel{2}t^2}{\cancel{2}} + c$$

$$\Leftrightarrow \ln|y| = -t^2 + c$$

$$\Leftrightarrow y = e^{-t^2+c}$$

$$\Leftrightarrow y = e^{c} \times e^{-t^2}$$

$$\Leftrightarrow y = C e^{-t^2}$$

3º Passo:

$$y(0) = 2$$

Substituindo:

$$2 = C e^{-0^2}$$

$$\Leftrightarrow 2 = C$$

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

3.1.2-Exemplos de output - App com gráfico e tabela

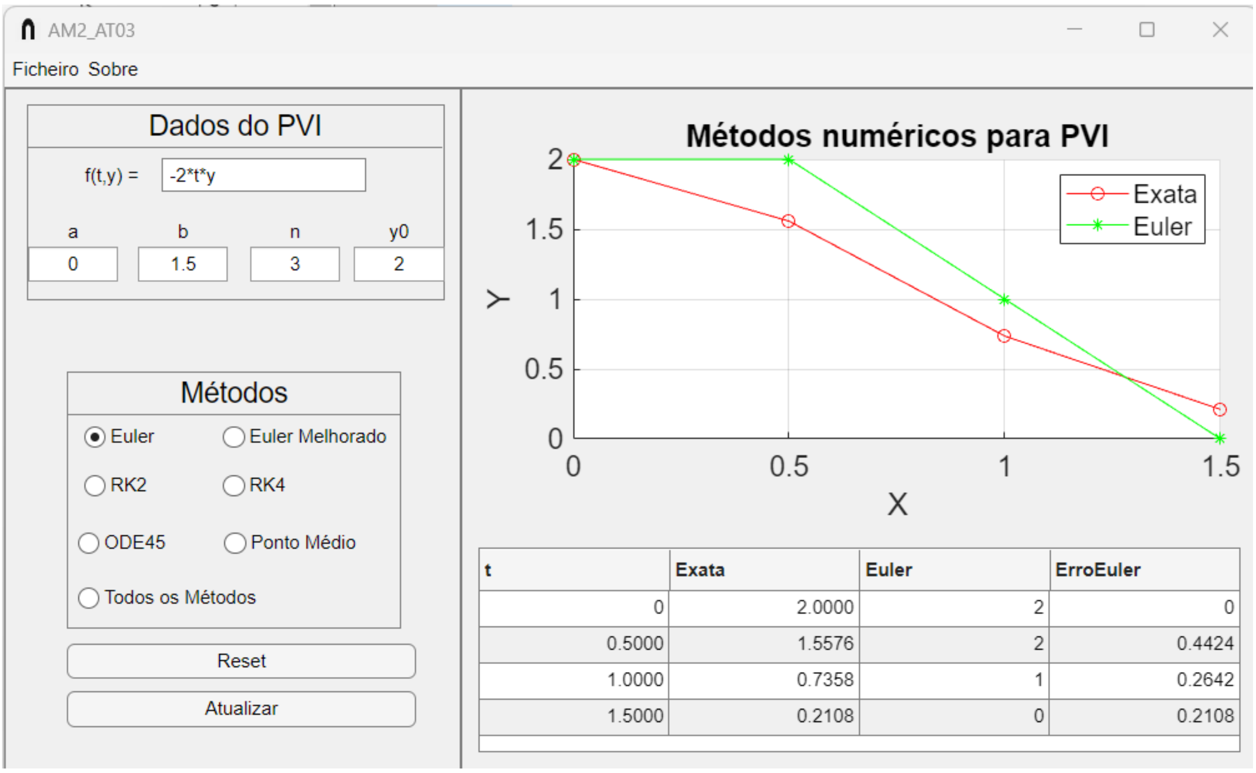
(b) Complete a tabela seguinte e interprete os resultados obtidos. Para o preenchimento da coluna das aproximações de Euler, deve apresentar os cálculos das iterações da aplicação da fórmula do método de Euler.

		Aproximações			Erros	
i	t_i	$y(t_i)$ Exata	y_i Euler	y_i RK2	$\lvert y(t_i) - y_i \rvert$ 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		

Pela tabela verificamos que h = 0.5, a = 0 e b = 1.5.

Como h = (b-a)/n, logo n = 3

Euler



RK2

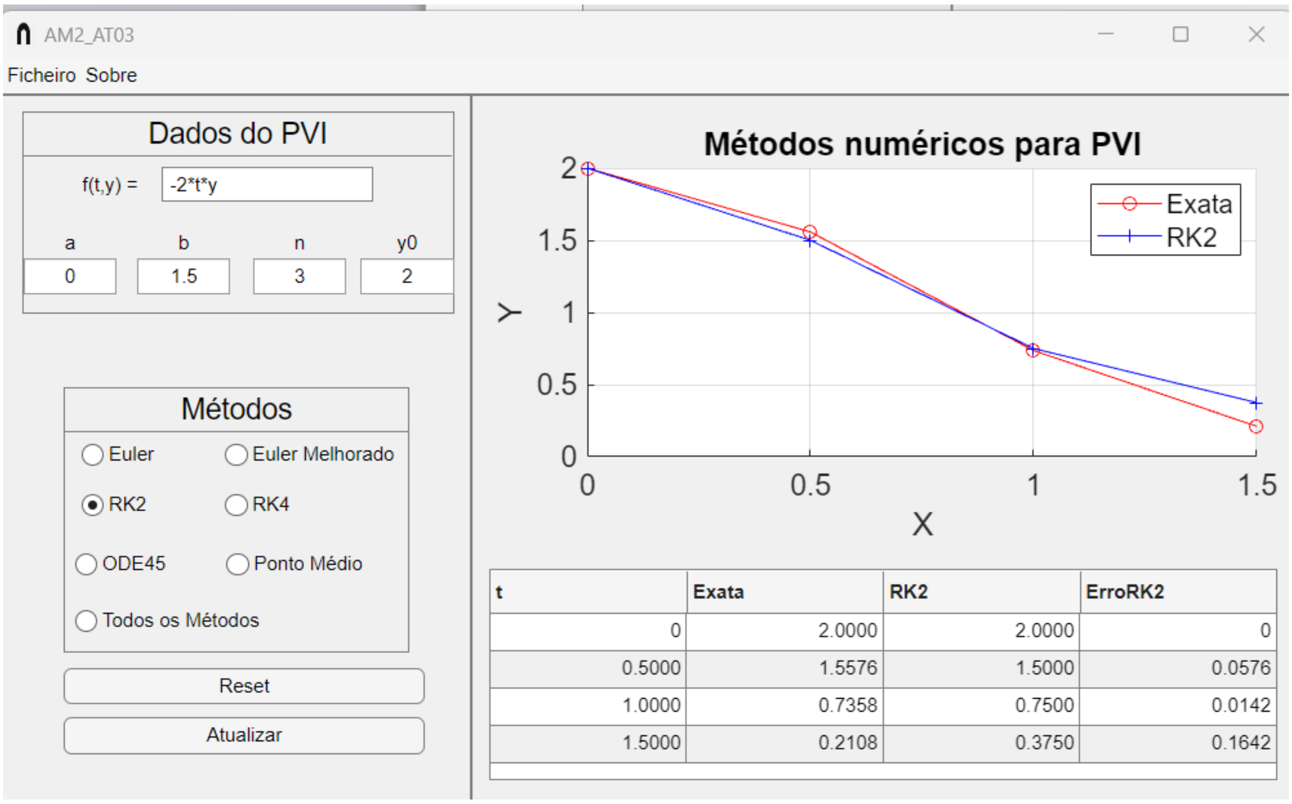


Tabela preenchida

			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	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.

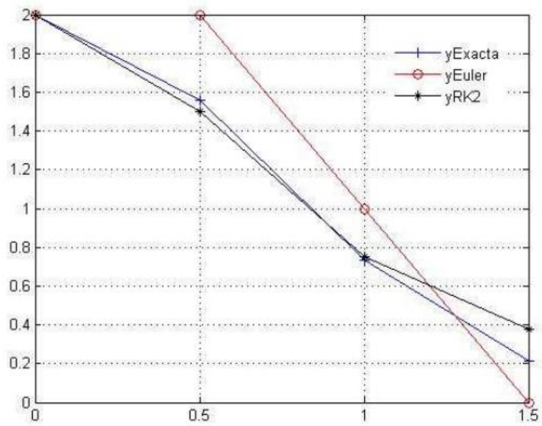


Figura 4

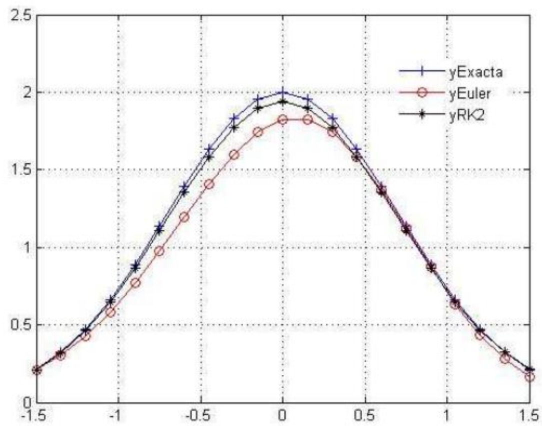


Figura 5

Como podemos ver pelos gráficos obtidos na nossa aplicação na alínea anterior, a figura 4 é a que representa corretamente uma solução do PVI dado.

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

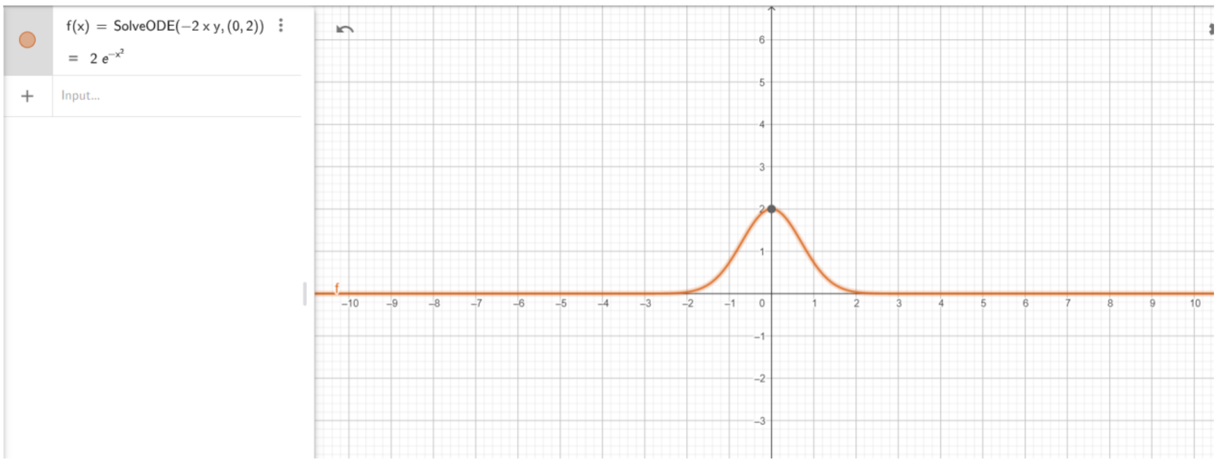
PVI

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

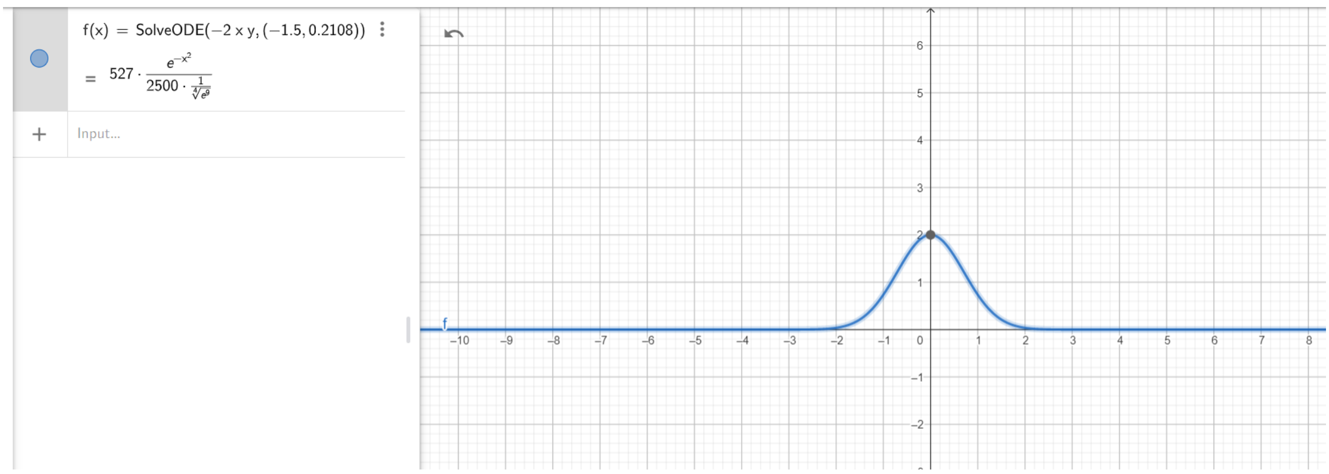
(e) Quais dos comandos seguintes em GeoGebra lhe permitiriam determinar a solução exata do PVI e a solução aproximada do mesmo.

(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]`

(A) - Solução exata



(B) - Solução aproximada



3.2-Problemas de aplicação do livro “Differential Equations with Modeling Applications”

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 $v(5)$.

3.2.1-Modelação Matemática do problema

Sabemos que:

$$\begin{cases} m\dfrac{dv}{dt} = mg - kv^2, \quad k>0 \\ v(0)=0 \\ t \in [0,5] \end{cases}$$

$\rightarrow$ ambiguo pois 0 valor inicial e 5 pois valor que pedem

$$\boxed{\begin{array}{l} m=5 \text{ slugs} \\ g = 32\ ft/s^2 \\ k = 0.125 \end{array}}$$

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

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

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

$$\Leftrightarrow \frac{dv}{dt} = \frac{160 - 0.125v^2}{5}$$

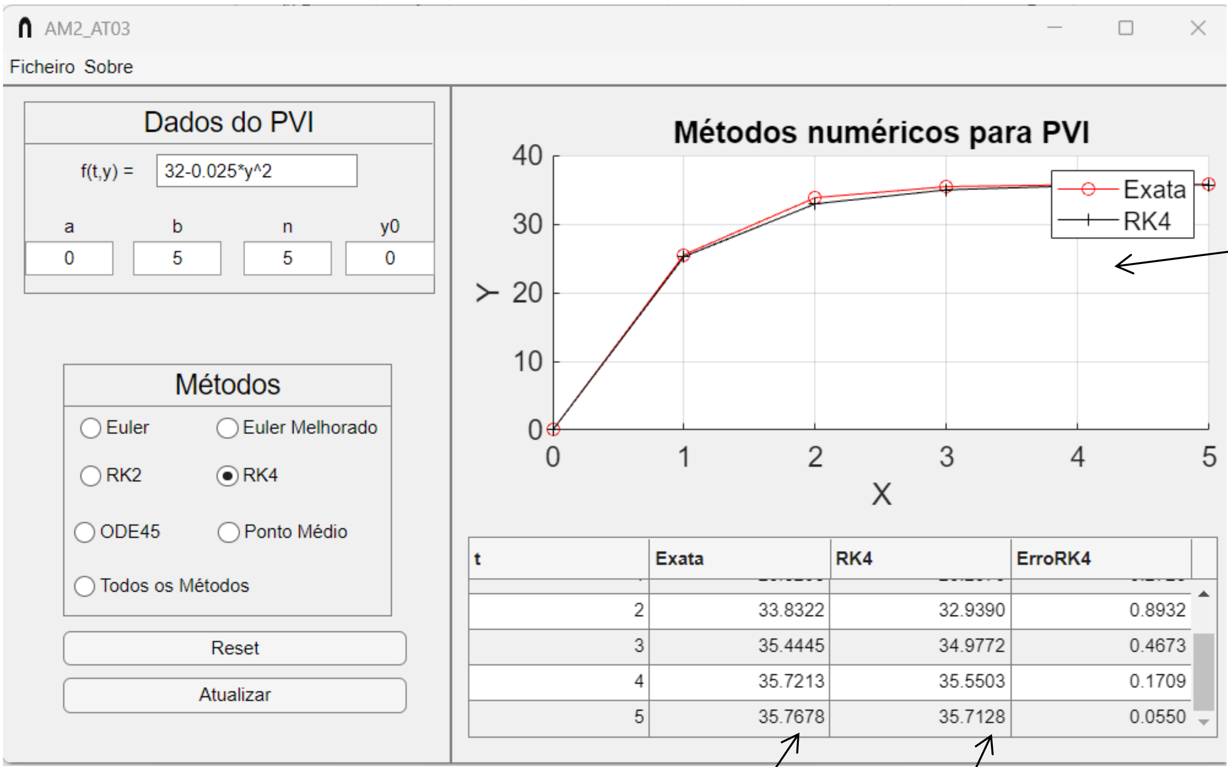
$$\Leftrightarrow \frac{dv}{dt} = 32 - 0.025v^2$$

$$\Leftrightarrow v' = 32 - 0.025v^2$$

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

3.2.2-Resolução através da App Desenvolvida

Usámos o método de Runge-Kutta de ordem 4 (RK4) em vez do Rk2 pois é mais preciso, o que conduzes a um erro menor.



Resposta a)- Gráfico da solução do PVI

Resposta c)- Valor real de v(5)

Resposta b)- aproximação pelo método de Runge-Kutta da velocidade massa em queda em ***t*** = 5s

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
$A(approximated)$					

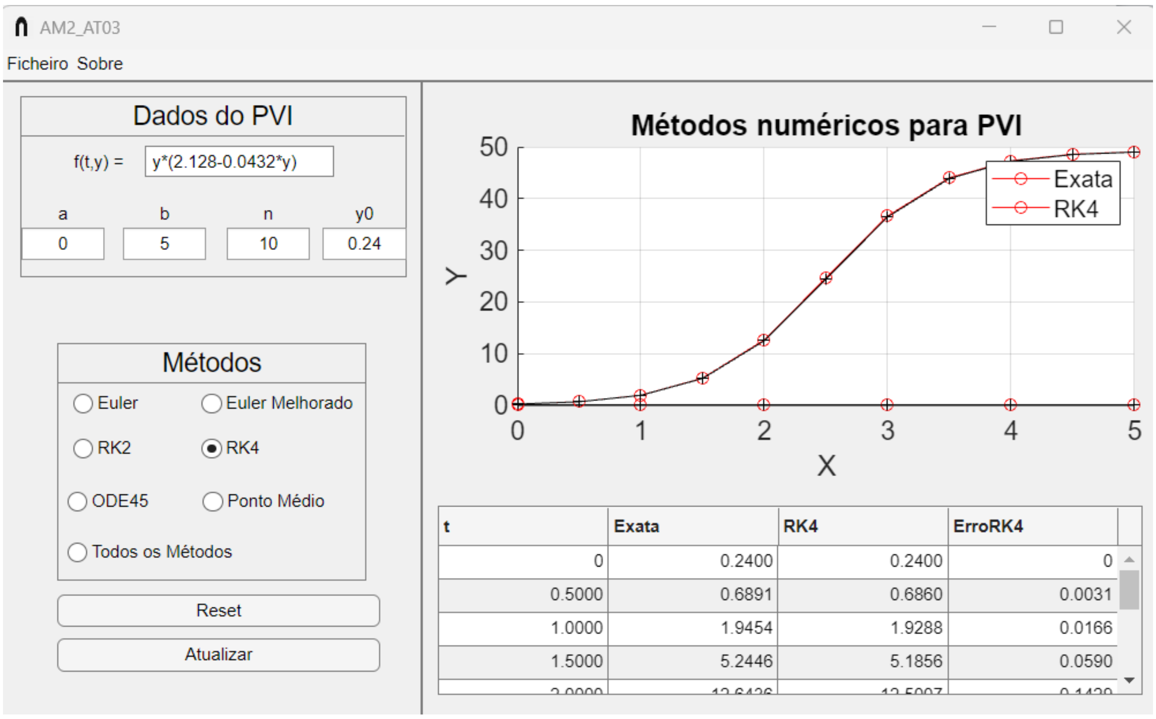
(b) Use a numerical solver to graph the solution of the initial-value problem. Estimate the values $A(1)$, $A(2)$, $A(3)$, $A(4)$, and $A(5)$ from the graph.

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

PVI

$$\left\{ \boxed{\begin{array}{l} y' = y*(2.128-0.0432*y) \\ t \in [0,5] \\ y(0) = 0.24 \end{array}} \right.$$

Como h=0.5, a=0, b=5 , pela formula h = (b-a)/n, o valor de **n = 10**



Pela app podemos retirar os valores que precisamos para preencher a tabela

t	Exata	RK4	erroRK4
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

$t(days)$	1	2	3	4	5
$A(observed)$	2.78	13.53	36.30	47.50	49.40
$A(approximated)$	1.93	12.50	36.46	47.23	49.00
A (exact)	1.95	12.64	36.63	47.32	49.02

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

(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

2-b) Através da interpretação da questão ficamos com a seguinte equação:

e - força eletromotriz

$e = Ri + L\frac{di}{dt}$

Manipulando a equação para isolar $\frac{di}{dt}$:

$\Leftrightarrow L\frac{di}{dt} + Ri = e \Leftrightarrow Li' + Ri = e \Leftrightarrow Li' = e - Ri$

$\Leftrightarrow i' = \frac{e}{L} - \frac{Ri}{L}$

Como queremos a equação diferencial deste modo: $y' = f(t,y)$, troca-se a variável i por y

$\Leftrightarrow y' = \frac{e}{L} - \frac{R}{L}y$ como $e = 3\sin(3t)$, $L = 0.5h$ e $R = 10\Omega$

$\Leftrightarrow y' = \frac{3sen(2t)}{0.5} - \frac{10}{0.5}y \Leftrightarrow y' = \underbrace{-20y + 6sen(2t)}_{f(t,y)}$

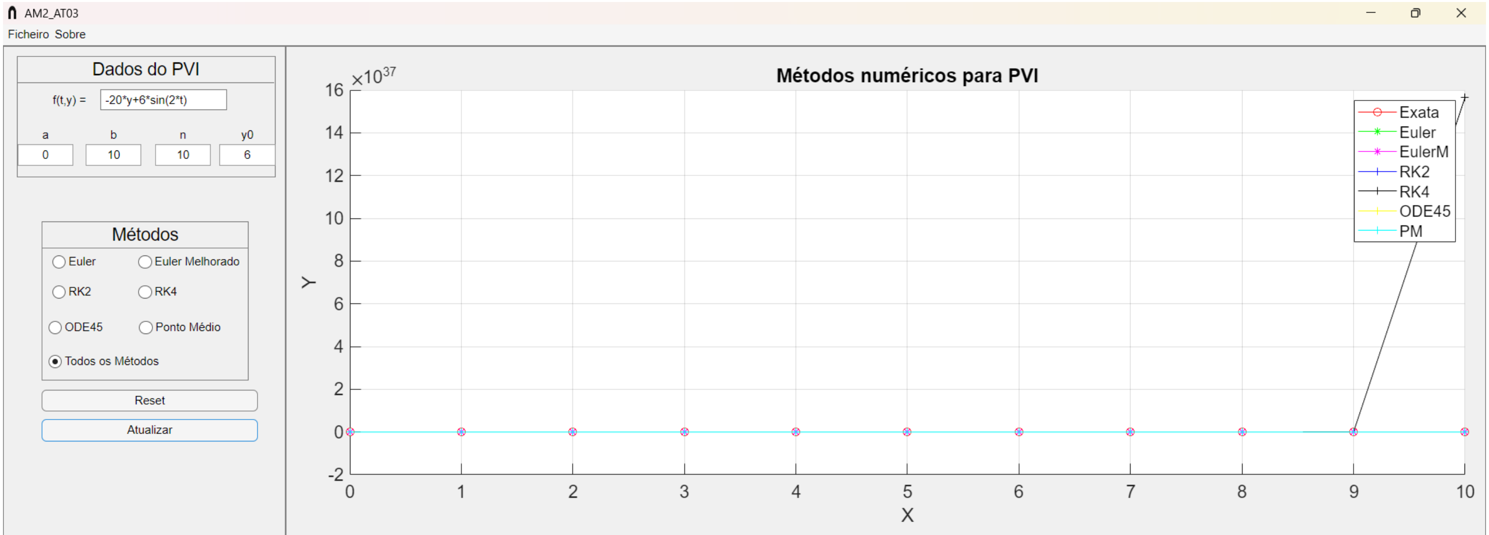
Quando $t=0$ $i=6$, logo $i(0)=6$, mudando de variável, $y(0)=6=y_0$

Como o tempo não pode ser negativo, $a=0$ como valor inicial de t.

Para uma melhor observação de resultados, definiu-se $b=10$ (valor final de t) e $n=10$ (passos).

3.3.2-Resolução através da app desenvolvida

Com a informação retirada e explicada anteriormente, utilizámos a nossa app com todos os métodos implementados.



t	Exata	Euler	EulerM	RK2	RK4	ODE45	PM	ErroEuler	ErroEulerM	ErroRK2	ErroRK4	ErroODE45	ErroPM
0	6.0000	6	6	6	6	6.0000	6	0	0	0	0	0	0
1	0.2824	-114	1.0887e+03	1.0887e+03	3.3225e+04	0.2825	1.0910e+03	114.2824	1.0884e+03	1.0884e+03	3.3225e+04	6.6343e-05	1.0908e+03
2	-0.2054	2.1715e+03	1.9701e+05	1.9701e+05	1.8321e+08	-0.2053	1.9743e+05	2.1717e+03	1.9701e+05	1.9701e+05	1.8321e+08	6.5552e-05	1.9743e+05
3	-0.1115	-4.1262e+04	3.5658e+07	3.5658e+07	1.0103e+12	-0.1115	3.5734e+07	4.1262e+04	3.5658e+07	3.5658e+07	1.0103e+12	1.0873e-05	3.5734e+07
4	0.2982	7.8398e+05	6.4541e+09	6.4541e+09	5.5711e+15	0.2982	6.4679e+09	7.8398e+05	6.4541e+09	6.4541e+09	5.5711e+15	5.2331e-05	6.4679e+09
5	-0.1367	-1.4896e+07	1.1682e+12	1.1682e+12	3.0721e+19	-0.1366	1.1707e+12	1.4896e+07	1.1682e+12	1.1682e+12	3.0721e+19	9.1355e-05	1.1707e+12
6	-0.1844	2.8302e+08	2.1144e+14	2.1144e+14	1.6940e+23	-0.1845	2.1189e+14	2.8302e+08	2.1144e+14	2.1144e+14	1.6940e+23	1.7280e-05	2.1189e+14
7	0.2902	-5.3773e+09	3.8271e+16	3.8271e+16	9.3415e+26	0.2901	3.8353e+16	5.3773e+09	3.8271e+16	3.8271e+16	9.3415e+26	7.2653e-05	3.8353e+16
8	-0.0571	1.0217e+11	6.9271e+18	6.9271e+18	5.1512e+30	-0.0571	6.9419e+18	1.0217e+11	6.9271e+18	6.9271e+18	5.1512e+30	9.4473e-07	6.9419e+18

4.Conclusão

Em suma, os métodos numéricos são uma ferramenta poderosa e indispensável para a solução de problemas matemáticos e científicos na atualidade. Porém, é importante ressaltar que a utilização dos métodos numéricos requer um conhecimento sólido de matemática e programação, além de cuidados na escolha do método e dos parâmetros a serem utilizados para evitar erros e imprecisões nos resultados obtidos.

Comparando os diferentes métodos, observamos que os que verificam menor erro e, consequentemente, melhor aproximação ao valor exato, são o método de Runge-Kutta de ordem 4 e o método usando a função ODE45 do MATLAB, que muitas vezes apresentaram erros muito pequenos (milésimas). Em contrapartida, temos o método de Euler, cujo erro é especialmente grande comparado com todos os outros métodos implementados.

Com este trabalho, adquirimos várias técnicas, não só de programação de Matlab como também ficamos a entender melhor ainda, a matéria dos métodos numéricos lecionada nas aulas.

5.Bibliografia

- Ficheiros de suporte disponibilizados pelo professor

- Formulário da cadeira

- Métodos Numéricos (Visualizado a 17 de abril de 2023)
  Disponível em: https://paginas.fe.up.pt/~faf/mnum/mnum-faf-handout.pdf

- Midpoint method-Wikipedia (Visualizado a 17 de abril de 2023)
  Disponível em: https://en.wikipedia.org/wiki/Midpoint_method

- Equações Diferencias-Método de Heun (Visualizado a 15 de abril de 2023
  Disponível em: https://cn.ect.ufrn.br/index.php?r=conteudo%2Fedo-heun

- Problema de valor inicial-Wikipedia (Visualizado a 23 de abril de 2023)
  Disponivel em: https://pt.wikipedia.org/wiki/Problema_de_valor_inicial

- Introdução as equações diferenciais-Luso Academia (Visualizado a 20 de abril de 2023)
  Disponível em: https://lusoacademia.org/2015/11/19/1-introducao-as-equacoes-diferenciais/

6.Autoavalição e heteroavaliação

Chegando ao fim do trabalho, estamos contentes pelo resultado final, tanto a nível da app quanto ao nível da LiveScript implementada.

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.