Análise Matemática II

Atividade 01 – Métodos Numéricos para EDO/PVI

Docente: Arménio Correia

Miguel Ângelo Rodrigues Ferreira nº 2020107016 - LEI

Pablo Oliveira Amaral nº 2020143935 - LEI

Paulo Henrique Figueira Pestana de Gouveia nº 2020121705 – LEI

Coimbra, 3 de maio de 2021

Índice

1 Introdução ... 3
1.1 Equação diferencial: definição e propriedades ... 3
1.2 Definição de PVI ... 4
2 Métodos numéricos para resolução de PVI ... 5
2.0 Cálculo do Passo ... 5
2.1 Método de Euler ... 6
2.1.1 Fórmulas ... 6
2.1.2 Algoritmo/Função ... 6
2.2 Método de Euler melhorado ou modificado ... 8
2.2.1 Fórmulas ... 8
2.2.2 Algoritmo/Função ... 8
2.3 Método de RK2 ... 10
2.3.1 Fórmulas ... 10
2.3.2 Algoritmo/Função ... 11
2.4 Método de RK4 ... 12
2.4.1 Fórmulas ... 12
2.4.2 Algoritmo/Função ... 12
2.5 Função ODE45 do Matlab ... 14
2.5.1 Fórmulas ... 14
2.5.2 Algoritmo/Função ... 14
2.6 Método de Dormand Prince ... 15
2.6.1 Fórmulas ... 15
2.6.2 Algoritmo/Função ... 15
3 Exemplos de aplicação e teste dos métodos ... 17
3.1 Exercício 3 do teste farol ... 17
3.1.1 PVI – Equação diferencial de 1ª ordem e condições iniciais ... 17
3.1.2 Exemplos de output – App com gráfico e tabela ... 18
3.2 Problemas de aplicação ... 20
3.2.1 Modelação matemática do problema ... 20
3.2.2 Resolução através da app desenvolvida ... 23
4 Conclusão ... 25

1 Introdução

Este trabalho foi realizado no âmbito da unidade curricular de Análise Matemática II com o intuito de adquirir e aprofundar conhecimentos sobre os métodos numéricos para desenvolvimento de equações diferenciais ordinárias e problemas de valor iniciar e também na linguagem de Matlab.

Ao longo deste relatório iremos abordar os temas propostos no enunciado, tais como, equações diferenciais ordinárias, problemas de valor inicial e os diferentes métodos e formulas para resolver os problemas e também mostrando o respetivo código para cada um em Matlab.

1.1 Equação diferencial: definição e propriedades

Uma equação diferencial é uma equação cuja incógnita é uma função que aparece na equação na forma de uma derivada correspondente. Dada uma variável x, função de uma variável y, a equação diferencial envolve, x, y, derivadas de y e eventualmente também derivadas de x.

$$x^2 \frac{d^2y}{dx^2} + x\frac{dy}{dx} + (x^2)y = 0$$

Existem dois tipos de equações diferenciais: equações diferenciais ordinárias (EDO) e equações de derivadas parciais (EDP). EDO é uma função que envolve incógnitas e as suas derivadas, enquanto que EDP é uma função com múltiplas variáveis, em que existe uma relação entre elas. Essa relação é o que vai possibilitar a resolução dos valores das incógnitas.

Algumas das propriedades destas equações são:

- Ordem e grau:
    - A Ordem de uma ED corresponde à ordem da derivada mais alta da função.
    - O Grau de uma ED é o expoente da maior derivada da função.
- Pode ou não haver solução;
- No caso de existir solução, estou pode ou não ser única.

1.2 Definição de PVI

Um PVI (Problema de Valor Inicial) trata-se de uma equação diferencial acompanhada do valor da função num certo ponto, denominado de valor inicial ou condição inicial. Maioritariamente, um PVI, é associado a problemas reais, com aplicações em áreas científicas, sendo por norma que a equação diferencia dada, trata-se de uma equação evolutiva que descreve como é que o sistema se vai desenvolver com o tempo, caso as condições iniciais se verifiquem.

Matematicamente, um PVI pode ser apresentado da seguinte forma:

$$\begin{cases} y' = f(t, y) & (1) \\ t \in [a, b] & (2) \\ y(a) = y_0 & (3) \end{cases}$$

Em que:

(1) É a equação diferencial.
(2) É o intervalo pretendido.
(3) É a condição inicial (valor inicial).

Estes PVI's podem ser resolvidos de uma forma exata ou aproximada, e este trabalho trata a segunda forma através do uso de métodos numéricos.

2 Métodos numéricos para resolução de PVI

2.0 Cálculo do Passo

Em todos os métodos numéricos implementados ao longo do trabalho, será utilizado o valor do passo, ***h***. De forma a evitar múltiplas repetições, a definição e a fórmula de calculo deste valor será apresentado aqui.

Este valor, é o tamanho de cada subintervalo no intervalo original [a, b], e pode calculado da seguinte forma:

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

Em que:

- h → Tamanho de cada subintervalo (passo);
- α → Limite esquerdo do intervalo;
- b → Limite direito do intervalo;
- n → Número de subintervalos.

Nota: Quanto menor for o valor ***h***, maior será a quantidade de subintervalos que irão existir dentro de um determinado intervalo e maior será a aproximação ao valor real.

2.1 Método de Euler

2.1.1 Fórmulas

O método de Euler é um procedimento numérico de primeira ordem (y') para aproximação da solução da equação diferencial $y' = f(t, y)$ que satisfaz a condição inicial: $y(t_0) = y_0$.

Para resolver um PVI, o método de Euler é dado pela seguinte equação:

(Fórmula Geral)

$$y_i + 1 = y_i + h * f(x_i, y_i), i = 0, 1, 2, \dots, n - 1$$

Em que:

- $y_i + 1 \rightarrow$ Próximo valor aproximado da solução do problema inicial (na abcisa $t_i + 1$);
- $y_i \rightarrow$ Valor aproximado da solução do problema inicial na abcissa atual;
- $h \rightarrow$ Valor de cada subintervalo (passo);
- $f(t_i, y_i) \rightarrow$ Valor da equação em t_i e y_i.

2.1.2 Algoritmo/Função

Algoritmo:

1. Definir o valor do passo (***h***);
2. Definir um vetor ***y*** para guardar a solução e atribuir $y(1) = y_0$;
3. Atribuir o primeiro valor de ***y***;
4. Para ***i*** de 1 a ***n***, fazemos o cálculo do método de Euler para ***i*** iterações.

Função (Matlab)

```
function y = MEuler(f,a,b,n,y0)
%MEULER Método de Euler para resolução numérica de EDO/PVI
%   y'=f(t,y), t=[a,b], y(a)=y0
%   y(i+1)=y(i)+hf(t(i),y(i)), i=0,1,2,...,n
%INPUT:
%   f - função da EDO y'=f(t,y)
%   [a,b] - intervalo de valores da variável independente t
%   n - núnmero de subintervalos ou iterações do método
%   y0 - aproximação inicial y(a)=y0
%OUTPUT:
%   y - vetor das soluções aproximadas do PVI em cada um dos t(i)
%
%   26/03/2021  Arménio Correia   armenioc@isec.pt
%   26/04/2021  Paulo Gouveia     a2020121705.isec.pt
%   26/04/2021  Miguel Ferreira   a2020107016.isec.pt

    h = (b-a)/n;                            % Tamanho de cada sub-intervalo

    t = a:h:b;                              %Alocação de memória
    y = zeros(1,n+1);                       %Alocação de memória

    y(1) = y0;                              % Condição inicial PVi

    for i =1:n                              % O número de iterações vai ser igual a n
        y(i+1) = y(i)+h*f(t(i),y(i));
    end
end
```

2.2 Método de Euler melhorado ou modificado

2.2.1 Fórmulas

O método de Euler melhorado acaba por ser semelhante ao método de Euler.

A única diferença entre eles está no facto de o método de Euler utilizar a médias das inclinações em cada ponto para cada iteração, enquanto o método melhorado calcula a inclinação em x_0 e x_1, obtendo resultados mais aproximados.

(Formula Geral)

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

$$y_{i+1} = y_i + \frac{h}{2}\big(f(t_i, y_i) + f(t_{i+1}, y_{i+1})\big), i = 0,1, \dots, n-1$$

Em que:

- y_{i+1} → Próximo valor aproximado da solução do problema original (na abcissa t_{i+1});
- y_i → Valor aproximado da solução do problema original na abcissa atual;
- ***h*** → Valor de cada subintervalo (passo).

2.2.2 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 ***n*** iterações.

Função (Matlab)

```matlab
function y = MEulerMelh(f,a,b,n,y0)
%MEULER Método de Euler para resolução numérica de EDO/PVI
%   y'=f(t,y), t=[a,b], y(a)=y0
%   y(i+1)=y(i)+hf(t(i),y(i)), i=0,1,2,...,n
%INPUT:
%   f - função da EDO y'=f(t,y)
%   [a,b] - intervalo de valores da variável independente t
%   n - número de subintervalos ou iterações do método
%   y0 - aproximação inicial y(a)=y0
%OUTPUT:
%   y - vetor das soluções aproximadas do PVI em cada um dos t(i)
%
%   26/03/2021  Arménio Correia   armenioc@isec.pt
%   09/04/2021  Paulo   Gouveia   a2020121705.isec.pt
%   09/04/2021  Miguel Ferreira   a2020107016.isec.pt

    h = (b-a)/n;          % Tamanho de cada sub-intervalo

    t = a:h:b;            % Pré-alocação de memória no vetor das abcissas
    y = zeros(1,n+1);     % Pré-alocação de memória no vetor das ordenadas

    y(1) = y0;            % Condição inicial PVi

    for i = 1:n           % O número de iterações vai ser igual a n
        y(i+1) = y(i)+(h/2)*(f(t(i),y(i))+ f(t(i),y(i)));
            % Próximo valor aproximado da solução do problema original
    end
end
```

2.3 Método de RK2

2.3.1 Fórmulas

Método Runge-Kutta de ordem 2, é um método que requer apenas derivadas de primeira ordem, utilizando para cada iteração dois valores, denominados "***k***", sendo estes a inclinação no início do intervalo e a inclinação no final do intervalo, por fim calcula-se a media das inclinações com o objetivo de obter a inclinação em cada uma das iterações.

(Formula Geral)

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

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

$$y_{i+1} = \ y_i + \frac{1}{2}(k_1 + k_2), i = 0,1, \dots, n-1$$

Em que:

- $y_{i+1} \rightarrow$ Próximo valor aproximado da solução do problema original (na abcissa t_{i+1});
- $y_i \rightarrow$ Valor aproximado da solução do problema original na abcissa atual;

(Cálculo de k1)

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

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

(Cálculo de k2)

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

- $k_2 \rightarrow$ Inclinação no fim do intervalo;
- $t_i \rightarrow$ Valor da abcissa atual;
- $h \rightarrow$ Tamanho de cada subintervalo (passo);
- $y_i \rightarrow$ Valor aproximado da solução do problema original na abcissa atual;
- $k_1 \rightarrow$ Inclinação no início do intervalo

2.3.2 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 do ponto médio das inclinações do intervalo;
7. Cálculo do método de RK2 para ***n*** iterações.

Função (Matlab)

```
function y = NRK2(f,a,b,n,y0)
%NRK2  Método Númerico para resolver um PVI: Runge-Kutta de ordem 2
%    y = NRK2(f,a,b,n,y0) Método numérico para a resolução de um PVI
%
%INPUT:
%    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
%
%OUTPUT:
%    y - vector das soluções aproximadas
%
%
%    25/04/2021  Paulo Gouveia    a2020121705.isec.pt
%    25/04/2021  Miguel Ferreira  a2020107016.isec.pt

h = (b-a)/n;                    %Tamanho de cada sub-intervalo

t = a:h:b;                      %Pré-alocação de memória no vetor das abcissas
y = zeros(1,n+1);               %Pré-alocação de memória no vetor das ordenadas

y(1) = y0;                      %Condição inicial PVi

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

    k1 = h*f(t(i),y(i));        %Inclinação no início do intervalo

    k2 = h*f(t(i+1),y(i)+k1);   %Inclinação no ponto médio do intervalo

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

end
end
```

2.4 Método de RK4

2.4.1 Fórmulas

O método de Runge-Kutta de ordem 4 não necessita do cálculo da derivada ***f***, mas depende de uma função definida através da avaliação de ***f*** em múltiplos pontos.

Para resolver um PVI através do método RK4 são utilizadas as seguintes equações:

Fórmula geral:

$$y_i + 1 = y_i + \frac{h}{6} * (k_1 + 2k_2 + 2k_3 + k_4), i = 0,1,2, \dots, n - 1$$

Em que:

- $y_i + 1 \longrightarrow$ Aproximação pelo método RK4 de y(xn+1);
- $y_i \longrightarrow$ Valor de y em ***n*** iteração;
- $h \longrightarrow$ Valor de cada subintervalo (passo).
- k_1, k_2, k_3 e k_4:
    - $k_1 = h * f(t_i, y_i)$
    - $k_2 = h * f(t_i + \frac{h}{2}, y_i + \frac{1}{2}k_1)$
    - $k_3 = h * f(t_i + \frac{h}{2}, y_i + \frac{1}{2}k_2)$
    - $k_4 = h * f(t_i + 1, y_i + k_3)$
- Onde:
    - $k_1 \longrightarrow$ Inclinação no início do intervalo;
    - $k_2 \longrightarrow$ Inclinação no ponto médio do intervalo;
    - $k_3 \longrightarrow$ Inclinação no ponto médio do intervalo;
    - $k_4 \longrightarrow$ Inclinação no final do intervalo.

Média ponderada das inclinações:

$$\frac{k_1 + 2k_2 + 2k_3 + k_4}{6}$$

2.4.2 Algoritmo/Função

Algoritmo

1. Definir o passo ***h***;
2. Definir um vetor ***y*** para guardar a solução e atribuir $y(1) = y_0$;
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 (novamente) da inclinação no ponto médio do intervalo;
7. Cálculo da inclinação no fim do intervalo;
8. Cálculo do método RK4 para ***n*** iterações.

Função (Matlab)

```
%NRK4  Método Númerico para resolver um PVI: Runge-Kutta de ordem 4
%    y = NRK4(f,a,b,n,y0) Método numérico para a resolução de um PVI
%
%INPUT:
%    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
%
%OUTPUT:
%    y - vector das soluções aproximadas
%
%    26/04/2021  Paulo Gouveia    a2020121705.isec.pt
%    26/04/2021  Miguel Ferreira  a2020107016.isec.pt

function y = NRK4(f,a,b,n,y0)
    h = (b-a)/n;        % Tamanho de cada sub-intervalo

    t = a:h:b;          % Pré-alocação de memória no vetor das abcissas
    y = zeros(1,n+1);   % Pré-alocação de memória no vetor das ordenadas

    y(1) = y0;          % Condição inicial PVi

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

     k1 = h*f(t(i), y(i));                       % Inclinação no início do intervalo

     k2 = h*f(t(i) + h/2, y(i) + 0.5*k1);        % Inclinação no ponto médio do intervalo

     k3 = h*f(t(i) + h/2, y(i) + 0.5*k2);        % Inclinação (novamente) no ponto médio do intervalo

     k4 = h*f(t(i+1), y(i) + k3);                % Inclinação no final do intervalo

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

2.5 Função ODE45 do Matlab

2.5.1 Fórmulas

A função ODE45 é uma função nativa do Matlab e baseada no método de Runge-Kutta. Para solucionar um PVI com através de uma EDO de ordem 2, a chamada deverá ser feita da seguinte forma:

$$[t, y] = ode45(f, t, y_0)$$

Em que:

- t → É o vetor das abcissas;
- f → Equação diferencial em ***t*** e em ***y***;
- y_0 → Valor inicial do PVI (condição inicial).

2.5.2 Algoritmo/Função

Algoritmo:

1. Definir o passo (h);
2. Aproximação através da função ODE45.

Função (Matlab):

```
%NODE45  Método Númerico para resolver um PVI: ODE45 função do MATLAB
%   y = NODE45(f,a,b,n,y0) Método numérico para a resolução de um PVI
%
%INPUT:
%   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
%
%OUTPUT:
%   y - vector das soluções aproximadas
%
%   26/04/2021  Paulo Gouveia    a2020121705.isec.pt
%   26/04/2021  Miguel Ferreira  a2020107016.isec.pt

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

    h = (b-a)/n;               % Tamanho de cada sub-intervalo

    t = a:h:b;                 % Pré-alocação de memória no vetor das abcissas

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

    y = y.';                   % Condição inicial PVi

end
```

2.6 Método de Dormand Prince

2.6.1 Fórmulas

O método de Dormand Prince é baseado em RK4, este avalia seis vezes a função para calcular soluções de quinta e quarta ordem.

Atualmente o método de Dormand Prince é o método padrão na função ODE45 do Matlab.

Formula Geral:

$$y_{i+1} = y_i + h(\frac{35k_1}{384} + \frac{500k_3}{1113} + \frac{125k_4}{192} - \frac{2187k_5}{6784} + \frac{11k_6}{84})$$

Em que:

- $y_i + 1 \longrightarrow$ Aproximação pelo método RK4 de y(xn+1);
- $y_i \longrightarrow$ Valor de y em ***n*** iteração;
- $h \longrightarrow$ Valor de cada subintervalo (passo).
- k_1, k_2, k_3, k_4, k_5 e k_6:
    - $k_1 = f(t_i, y_i)$
    - $k_2 = f(t_i + \frac{h}{5}, y_i + \frac{hk_1}{5})$
    - $k_3 = f(t_i + \frac{3h}{10}, y_i + \frac{3hk_1}{40} + \frac{9hk_2}{40})$
    - $k_4 = f(t_i + \frac{4h}{5}, y_i + \frac{44hk_1}{45} - \frac{56hk_2}{15} + \frac{32hk_3}{9})$
    - $k_5 = f(t_i + \frac{8h}{9}, y_i + \frac{19372hk_1}{6561} - \frac{25360hk_2}{2187} + \frac{6448hk_3}{6561} - \frac{212hk_4}{729})$
    - $k_6 = f(t_i + h, y_i + \frac{9017hk_1}{3168} - \frac{355hk_2}{33} + \frac{46732hk_3}{5247} + \frac{49hk_4}{176} + \frac{5103hk_5}{18656})$
- Onde:
    - $k_1 \longrightarrow$ Inclinação no início do intervalo;
    - $k_2 \longrightarrow$ Inclinação no ponto médio do intervalo;
    - $k_3 \longrightarrow$ Inclinação no ponto médio do intervalo;
    - $k_4 \longrightarrow$ Inclinação no ponto médio do intervalo;
    - $k_5 \longrightarrow$ Inclinação no ponto médio do intervalo;
    - $k_6 \longrightarrow$ Inclinação no final do intervalo.

2.6.2 Algoritmo/Função

Algoritmo:

1. Definir o passo ***h***;
2. Definir um vetor ***y*** para guardar a solução e atribuir $y(1) = y_0$;
3. Atribuir o primeiro valor de y;
4. Cálculo da inclinação no início do intervalo;
5. Cálculo da inclinação no ponto ***n***;
6. Cálculo (novamente) da inclinação no ponto ***n***;
7. Cálculo da inclinação no fim do intervalo;
8. Cálculo do método RK4 para ***n*** iterações.

Função (Matlab):

```matlab
function y = DormandPrince(f,a,b,n,y0)
%DormandPrince  Método Número para resolver um PVI: Dormand Prince
%   y = DormandPrince(f,a,b,n,y0) Método numérico para a resolução de um PVI
%
%INPUT:
%   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
%
%OUTPUT:
%   y - vector das soluções aproximadas

%referência: http://depa.fquim.unam.mx/amyd/archivero/DormandPrince_19856.pdf

%Ir ver https://www.mathworks.com/matlabcentral/answers/494191-planet-orbit-dormand-prince-algorithm

%   26/04/2021  Paulo Gouveia    a2020121705.isec.pt
%   26/04/2021  Miguel Ferreira  a2020107016.isec.pt

h = (b-a)/n;      %Tamanho de cada sub-intervalo

t = a : h : b;    %Pré-alocação de memória no vetor das abcissas
y = zeros(1,n+1); %Pré-alocação de memória no vetor das ordenadas

y(1) = y0;        % Condição inicial PVi

    for i=1:n

        k1 = f(t(i), y(i));
        k2 = f(t(i) + h/5, y(i) + ((h*k1)/5));
        k3 = f(t(i) + ((3*h)/10), y(i) + ((3*h*k1)/40)+((9*h*k2)/40));
        k4 = f(t(i)+((4*h)/5), y(i) + ((44*h*k1)/45)-((56*h*k2)/15)+((32*h*k3)/9));
        k5=  f(t(i)+((8*h)/9), y(i) + ((19372*h*k1)/6561)- ((25360*h*k2)/2187)+((64448*h*k3)/6561)-((212*h*k4)/729));
        k6=  f(t(i)+ h, y(i) + ((9017*h*k1)/3168)- ((355*h*k2)/33)+((46732*h*k3)/5247)+((49*h*k4)/176)-((5103*h*k5)/18656));
        y(i + 1) = y(i) + h*(((35*k1)/384) + ((500*k3)/1113) + ((125*k4)/192) - ((2187*k5)/6784)+((11*k6)/84));
    end
end
```

3 Exemplos de aplicação e teste dos métodos

3.1 Exercício 3 do teste farol

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

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

(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		

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

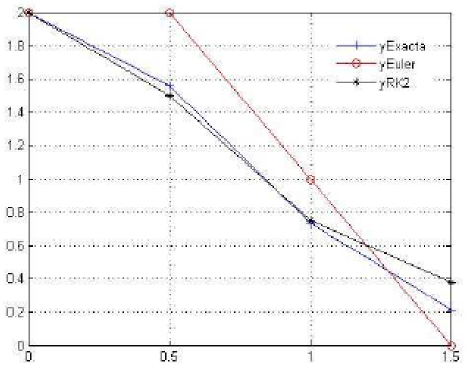


Figura 4

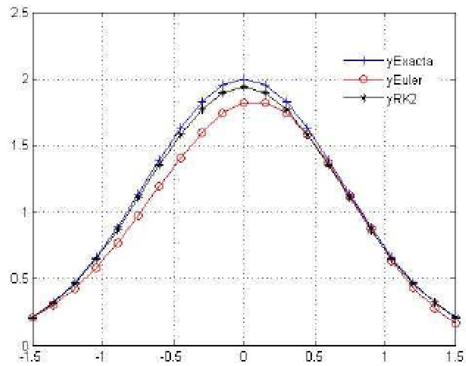


Figura 5

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

(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]`

3.1.1 PVI – Equação diferencial de 1ª ordem e condições iniciais

(a)

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

$$y' = -2ty \leftrightarrow y' + 2ty = 0$$

$$u = e^{\int 2t\, dy} = e^{t2}$$

$$e^{t2} y' + 2tye^{t^2} = 0 \leftrightarrow e^{t^2} y = \int 0 dt \leftrightarrow y = \frac{C}{e^{t^2}} \leftrightarrow y = Ce^{-t^2}$$

Sendo: $y(0) = 2 \leftrightarrow 2 = Ce^0 \leftrightarrow C = 2$

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

(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	2.0000	2.0000	0	0
1	0.5	1.5576	1.5576	1.5000	0.4424	0.0576
2	1	0.7358	0.7358	0.7500	0.2642	0.0142
3	1.5	0.2108	0.2108	0.3750	0.2108	0.1642

(c) Figura 4.

(d)

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

t=-1.5

Então:

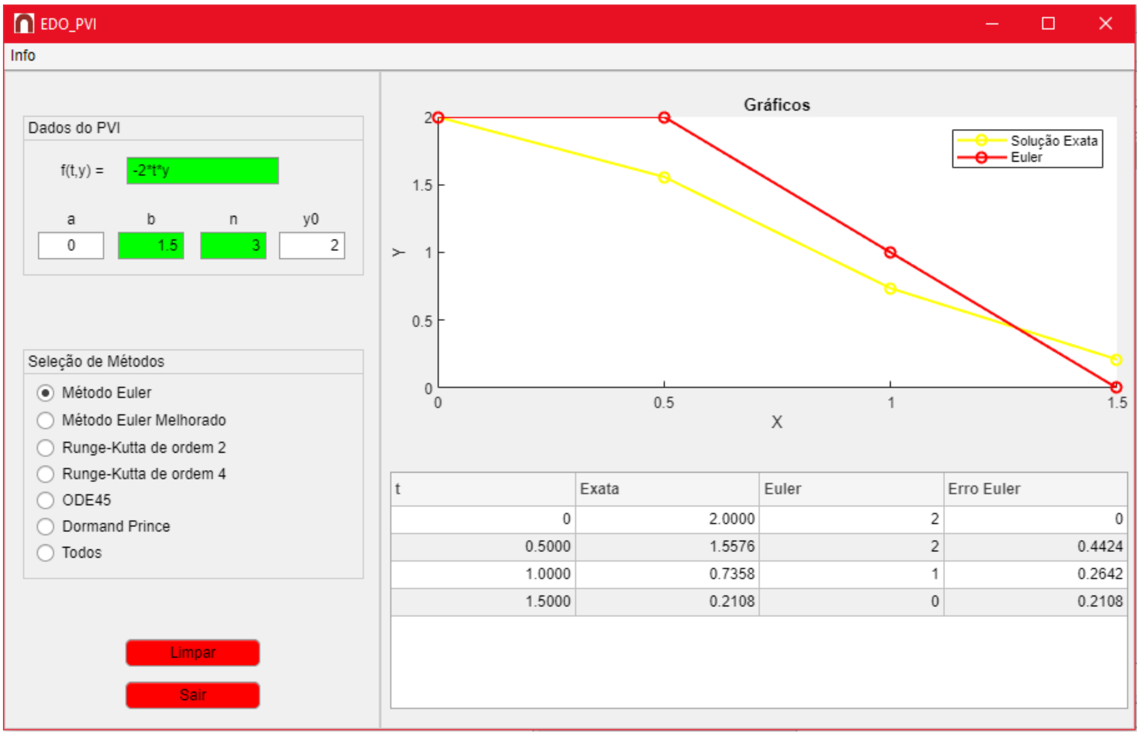
$$y = 2e^{-(-1.5)^2} \leftrightarrow y = 0.2107$$

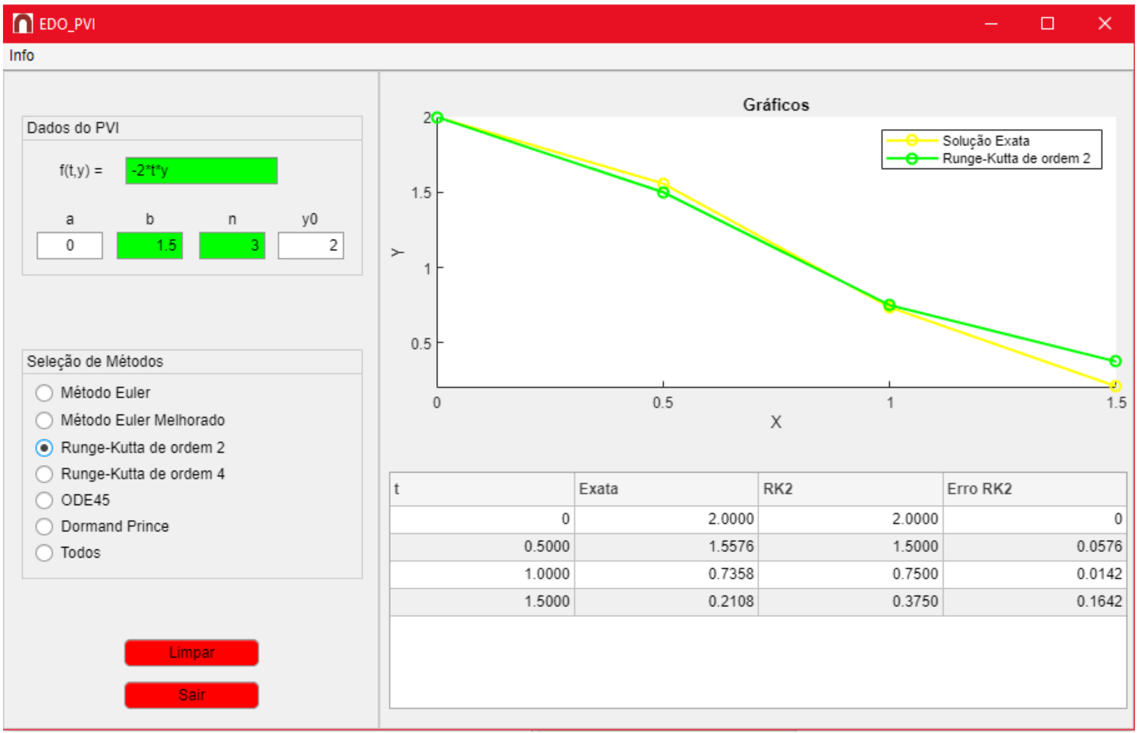
Logo, o PVI é:

$$\begin{cases} y' = -2ty \\ t \in [-1.5, 1.5] \\ y(-1.5) = 0.2107 \end{cases}$$

(e) A e C.

3.1.2 Exemplos de output – App com gráfico e tabela




EDO_PVI
Info
Dados do PVI
f(t,y) =
-2*t*y
a b n y0
0 1.5 3 2
Seleção de Métodos
Método Euler
Método Euler Melhorado
Runge-Kutta de ordem 2
Runge-Kutta de ordem 4
ODE45
Dormand Prince
Todos
Limpar
Sair
Gráficos
Solução Exata
Runge-Kutta de ordem 2
Y
X
t Exata RK2 Erro RK2
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

3.2 Problemas de aplicação

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 $v(5)$.

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 $A(5)$.

3.2.1 Modelação matemática do problema

1.

a)

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

Como:

$$k = 0.125$$
$$m = 5$$

$$g = 32\frac{fd}{s^2}$$

temos:

$$v' = \frac{(5 * 32 - 0.125 * v^2)}{5}$$

$$\begin{cases} y' = \dfrac{5 * 32 - 0.125 * y^2}{5} \\ t \in [0,5] \\ y0 = v(0) = 0 \end{cases}$$

$$a = 0$$

$$b = 5$$

$$h = 1$$

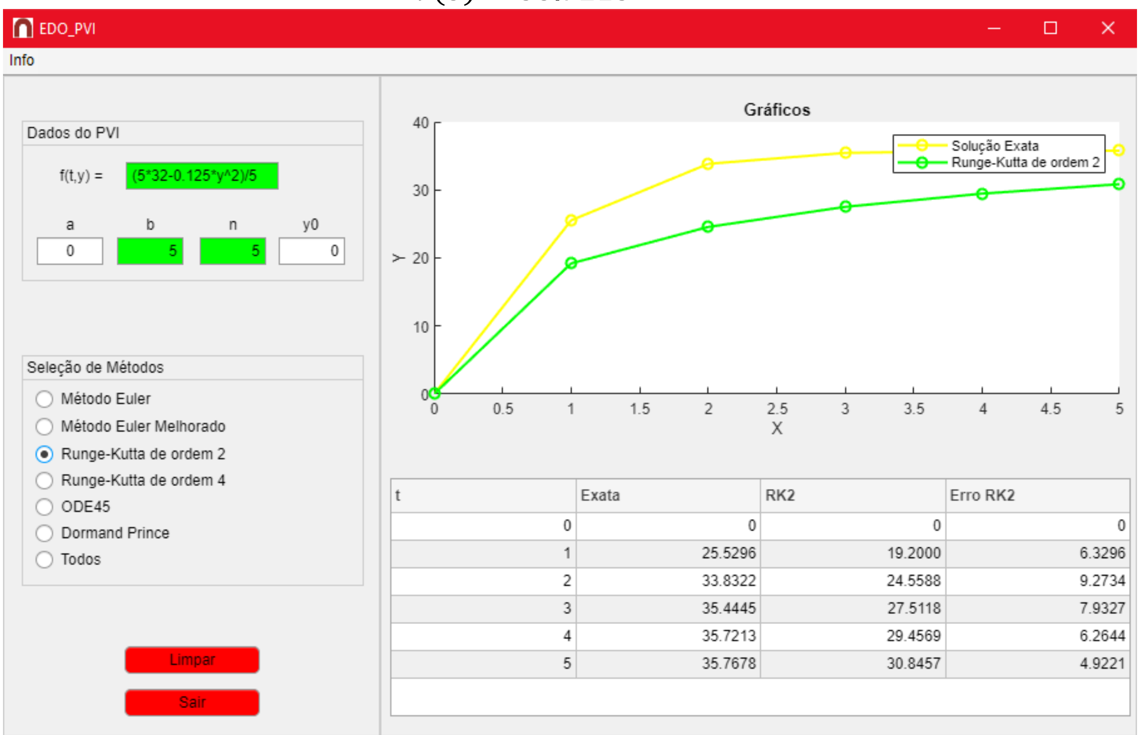
$$n = \frac{b-a}{h} = \frac{5-0}{1} = 5$$

b)

$$\begin{cases} y'(t) = f(t, y(t)) \\ t \in [a, b] \\ y(t_0) = y_0 \end{cases} \leftrightarrow \begin{cases} y'(t, y) = \dfrac{5 * 32 - 0.125 * y^2}{5} \\ t \in [0,5] \\ y(0) = 0 \end{cases}$$

c)

$$v(5) = 35.7213$$



2.

a)

$$\frac{dA}{dt} = A(2.218 - 0.0432A) \leftrightarrow A' = A(2.218 - 0.0432A)$$

$$\begin{cases} y' = y * (2.218 - 0.0432 * y) \\ t \in [1,5] \\ y_0 = 0.24 \end{cases}$$

$a = 1$

$b = 5$

$h = 0.5$

$$n = \frac{(b-a)}{h} = \frac{(5-1)}{0.5} = 8$$

Utilizando o método de RK2 e RK4 obtemos respetivamente os seguintes resultados:

$t(days)$	1	2	3	4	5
$A(observed)$	2.78	13.53	36.30	47.50	49.40
$A(approximated)$	0.2400	1.7401	10.9768	35.4325	47.4155

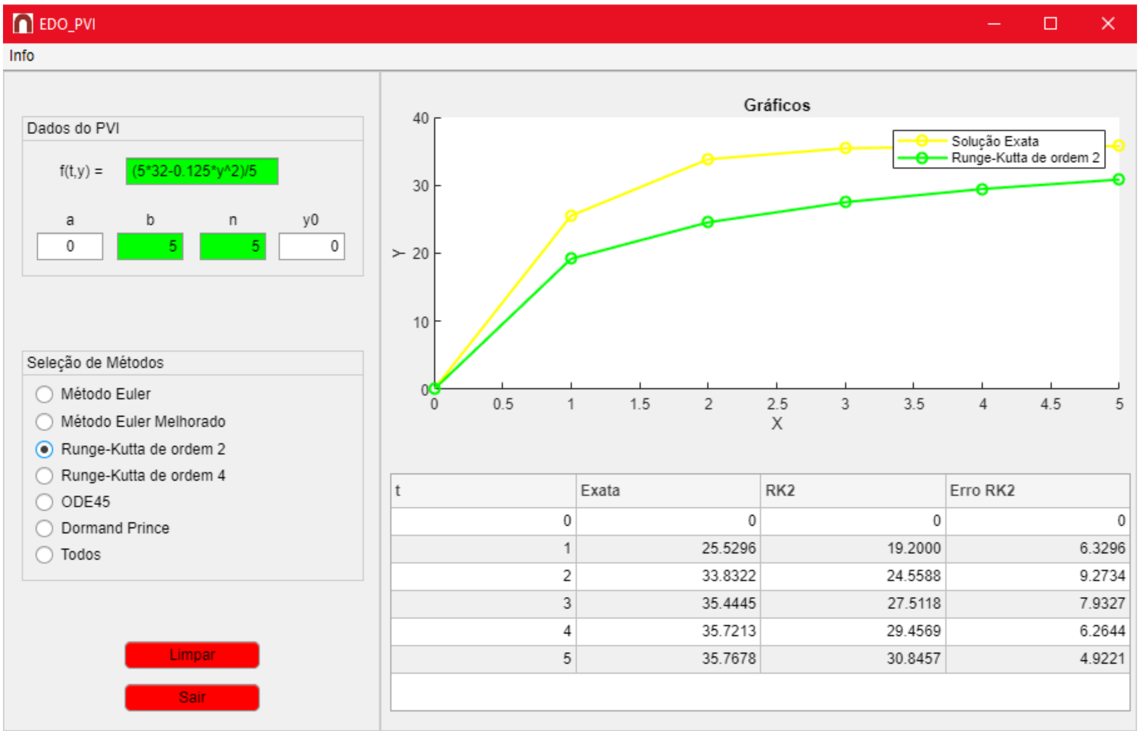
$t(days)$	1	2	3	4	5
$A(observed)$	2.78	13.53	36.30	47.50	49.40
$A(approximated)$	0.2400	2.1025	14.3923	40.1006	49.7660

3.2.2 Resolução através da app desenvolvida

1.

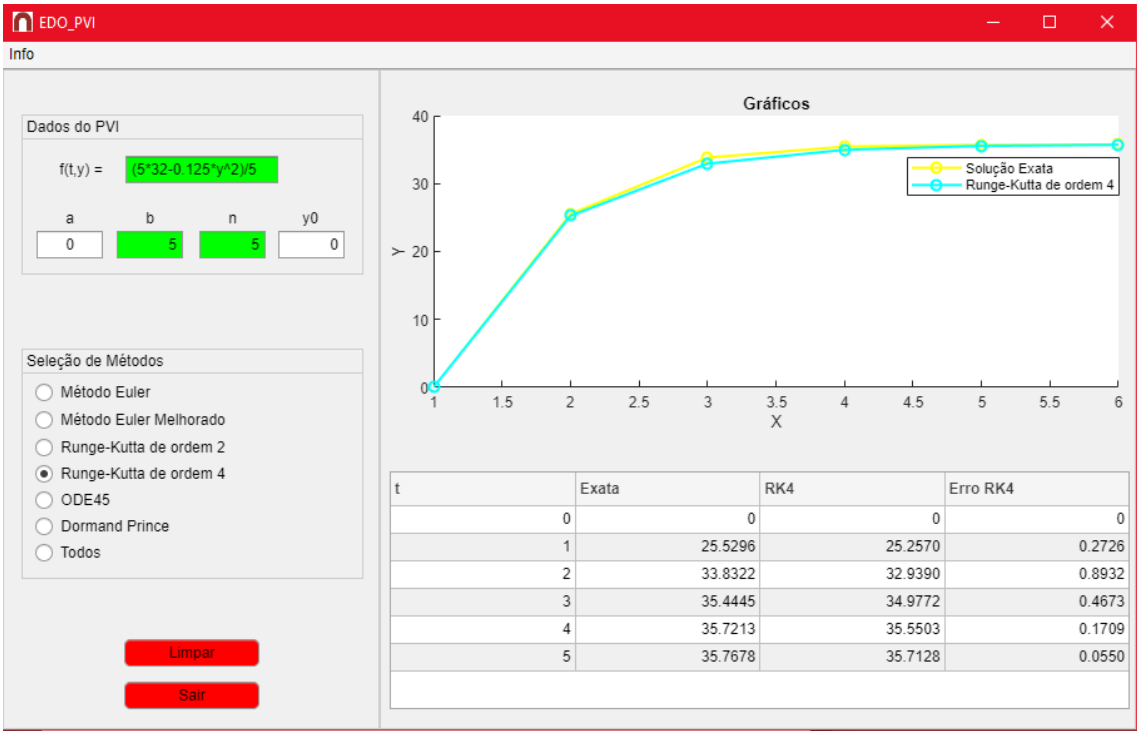
a)

b) Método de RK2:



t	Exata	RK2	Erro RK2
0	0	0	0
1	25.5296	19.2000	6.3296
2	33.8322	24.5588	9.2734
3	35.4445	27.5118	7.9327
4	35.7213	29.4569	6.2644
5	35.7678	30.8457	4.9221

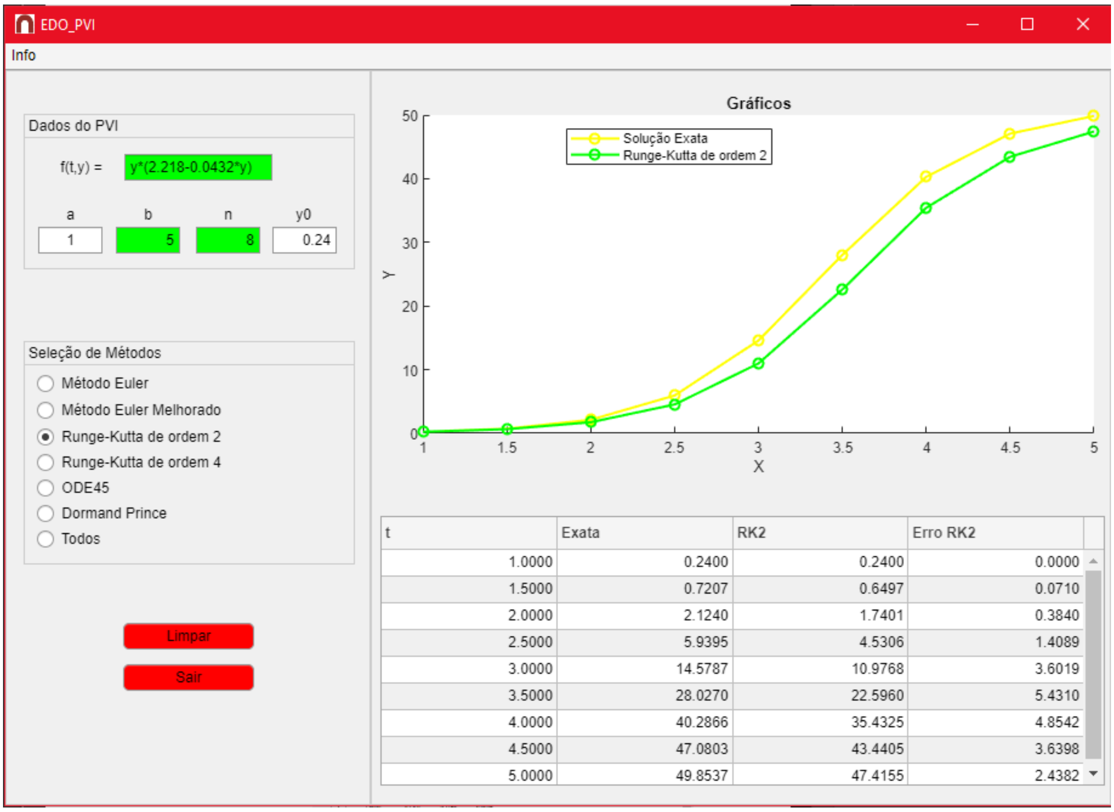
Método de RK4:



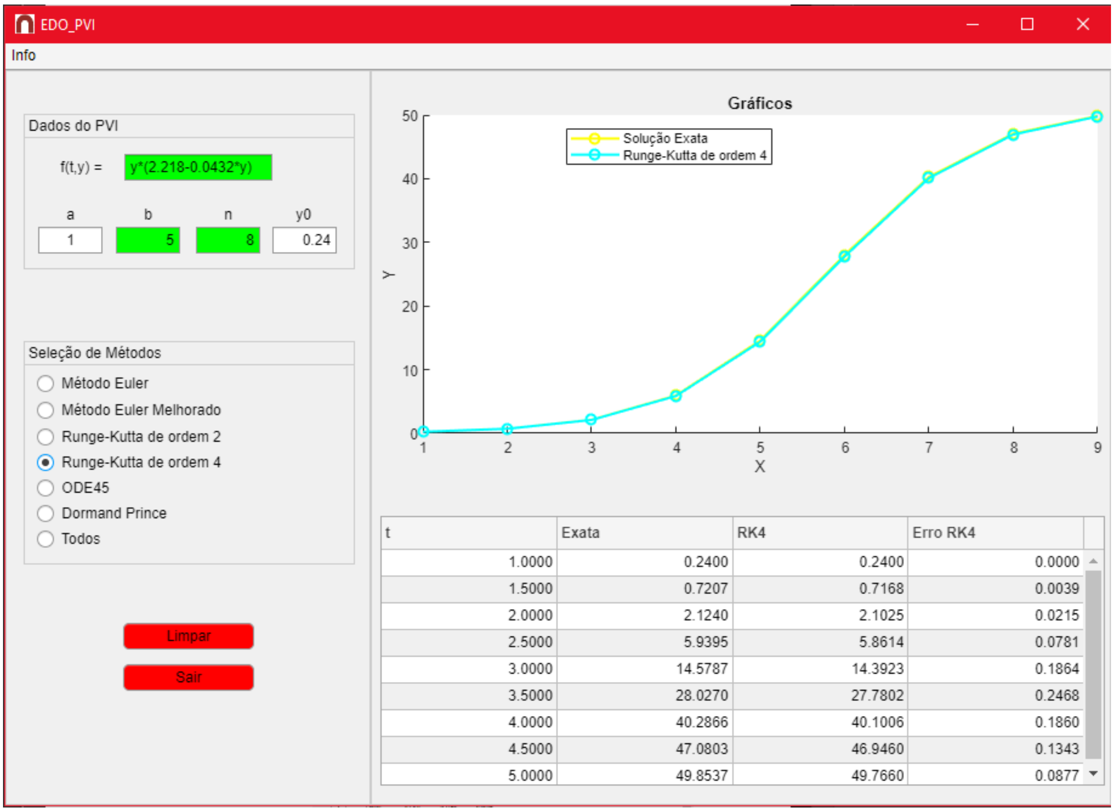
t	Exata	RK4	Erro RK4
0	0	0	0
1	25.5296	25.2570	0.2726
2	33.8322	32.9390	0.8932
3	35.4445	34.9772	0.4673
4	35.7213	35.5503	0.1709
5	35.7678	35.7128	0.0550

2.

b) Método RK2:



Método RK4:



4 Conclusão

Podemos então dizer que os métodos numéricos são bastante úteis para a resolução de problemas de valor inicial, principalmente quando utilizados em contexto real e pratico, uma vez que originam aproximações, dependendo do método utilizado, com erro mínimo.

Como regra geral, pode ser verificado o seguinte: quanto maior o número de subintervalos, menor será o erro dos métodos.

Relativamente à comparação de métodos, é possível observar que os métodos que verificam menor erro e, consequentemente, maior aproximação ao valor exato, são o método de Runge-Kutta de ordem 4 e o método usando a função ODE45 do MATLAB, que proporcionam maioritariamente erros na ordem das milésimas ou menor. Por outro lado, o método de Euler, quando comparado aos restantes métodos explorados, apresenta erros maiores.

Por fim, neste trabalho foram apresentados métodos de implementação simples capazes de produzir soluções para diversos problemas utilizando equações diferenciais e também adquirir/aprofundar técnicas de Matlab.