Docente: Arménio Correia

RELATÓRIO

Atividade01Trabalho

Métodos Numéricos para EDO/PVI

Licenciatura em Engenharia Informática

A análise numérica é o estudo de algoritmos de aproximação para a solução de problemas matemáticos.

Trabalho realizado por:
Tomás Gomes Silva - 2020143845
Tomás da Cunha Pinto - 2020144067
Francisco Santos Seabra Mendes - 2020143982

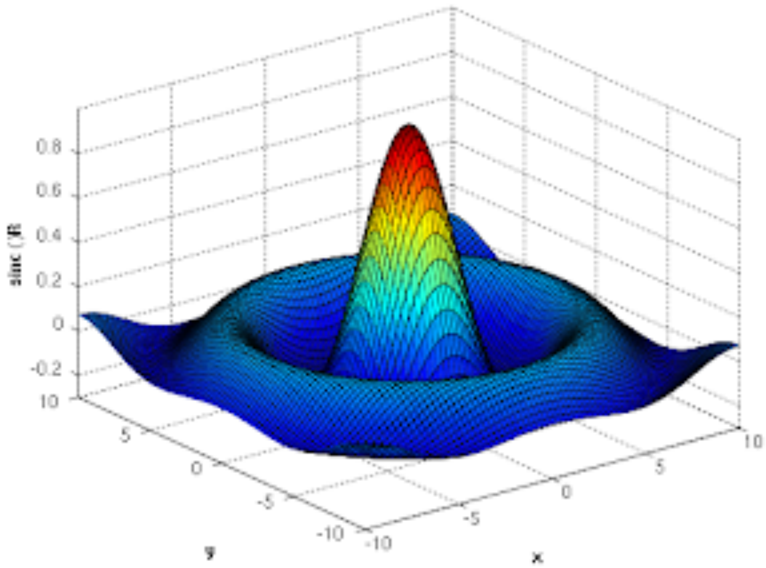
Índice

Introdução

O primeiro trabalho proposto para a unidade curricular de Análise Matemática 2 consiste no estudo e aplicação de Métodos Numéricos para resolução de Equações Diferenciais Ordinárias (EDO) e Problemas de Valor Inicial (PVI).

Este relatório vai abordar todos os métodos utilizados e alguns pontos chave da criação da interface de texto bem como a interface gráfica em MATLAB.

Os cinco métodos utilizados neste trabalho foram: o método de Euler, o método de Euler Melhorado/Modificado, o método de Runge-Kutta de ordem 2, o método de Runge-Kutta de ordem 4, a função ODE45 do próprio MATLAB e o método do Ponto Médio que foi procurado pelo nosso grupo de trabalho.



Equação diferencial: definição e propriedades

Uma equação diferencial é uma equação com derivadas de uma ou mais funções. As equações diferenciais podem ser classificadas por tipo, ordem e linearidade.

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

Figura 1 - Exemplo de uma Equação Diferencial

TIPO: Uma equação diferencial diz-se ordinária (EDO) quando a função depende apenas de uma única variável independente. Caso dependa de mais do que uma variável então dizemos que se trata de uma equação diferencial parcial (EDP).

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

Figura 2 - Equação Diferencial Ordinária (EDO)

ORDEM: A ordem de uma equação diferencial depende do número de derivadas que aparecem na equação. Se na equação aparecerem 2 derivadas então a equação diferencial é de segunda ordem.

$t^2y'' - 3ty' + 4y = 0$ **(EDO de ordem 2)**

LINEARIDADE: Uma equação diferencial ordinária é linear se tivermos um coeficiente a multiplicar pela derivada, ou por outras palavras, se puder ser escrita nesta forma:

$$a_0(t)y^{(n)} + a_1(t)y^{(n-1)} + \ldots + a_n(t)y = g(t),$$

Figura 3 - Outra maneira de escrever uma EDO

$t^2y'' - 3ty' + 4y = 0$ é uma EDO Linear $a_0 = 1$, $a_1 = -3$, $a_2 = 4$.
$y''' + 2e^t y'' + yy' = 0$ não é uma EDO Linear pois temos o uma incógnita (y) a multiplicar pela derivada (y')

Definição de PVI

Um **Problema de Valor Inicial**, também conhecido por *PVI*, é uma equação diferencial que é acompanhada do valor da função objetivo em um determinado ponto, chamada de condição inicial.

$$\textbf{PVI} \quad \begin{cases} y' = f(t,y) & \longleftarrow \text{Equação Diferencial} \\ t \in [a,b] & \longleftarrow \text{Variável Independente} \\ y(a) = y_0 & \longleftarrow \text{Condição Inicial} \end{cases}$$

$$\begin{cases} y' - \frac{1}{2}y = e^{-t} \\ t \in [a,b] \\ y(0) = -1 \end{cases} \quad \leftrightarrow \quad y = -\frac{2}{3}e^{-t} - \frac{1}{3}e^{\frac{t}{2}}$$

Figura 4 – Exemplo e solução de um PVI

Métodos Numéricos para resolução de PVI

Método de Euler

Fórmulas:

O **método de Euler** é um procedimento numérico de primeira ordem para solucionar equações diferenciais ordinárias $y' = f(t, y)$ com um valor inicial dado $y(t_0) = y_0$.

Para resolvermos um PVI recorrendo ao método de Euler usamos a fórmula geral:

$$\boldsymbol{y_{i+1} = y_i + hf(t_i, y_i)}$$
$$(\text{com } i = 0, 1, 2, \dots, n-1)$$

$\boldsymbol{y_{i+1}}$: Próximo valor aproximado da solução do problema na abcissa t_{i+1}
$\boldsymbol{y_i}$: Próximo valor aproximado da solução do problema na abcissa t_i
$\boldsymbol{h}$: Valor de cada subintervalo
$\boldsymbol{f(t_i, y_i)}$: Valor da equação em t_i e y_i

Algoritmo/Função:

1. Definir o valor do passo do **h**
2. Criar um vetor que contém os valores para **t** a partir de **h**
3. Criar um vetor para armazenar as soluções nos diferentes pontos **t**
4. Definir o primeiro elemento do vetor ***y*** para o valor da condição inicial
5. Fazer um ciclo iterativo que percorra desde **1 a n** para calcular a solução aproximada utilizando a fórmula geral

```
h = (b-a)/n; %Amplitude de cada subintervalo

t = zeros(1, n+1); %Alocação de Memória
y = zeros(1, n+1); %Alocação de Memória

t(1) = a; %Definir o primeiro elemento do array
y(1) = y0; %Definir o primeiro elemento do array

for i =1:n %Aplicar o Método de Euler (iteração)
    y(i+1) = y(i)+h*f(t(i),y(i));
    t(i+1) = t(i)+h;
end
```

Figura 4 - Algoritmo do método de Euler

Método de Euler Melhorado

Fórmulas:

O **método de Euler Melhorado**, ou método de Euler Modificado ou método de Heun é uma melhoria ao método de Euler para solucionar equações diferenciais ordinárias $y' = f(t, y)$ com um valor inicial dado $y(t_0) = y_0$.

Para resolvermos um PVI recorrendo ao método de Euler Melhorado usamos a fórmula geral:

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

(com $i = 0, 1, 2, \dots, n-1$*)*

$\boldsymbol{y_{i+1}}$: Próximo valor aproximado da solução do problema na abcissa t_{i+1}
$\boldsymbol{y_i}$: Próximo valor aproximado da solução do problema na abcissa t_i
$\boldsymbol{h}$: Valor de cada subintervalo
$\boldsymbol{f(t_i, y_i)}$: Valor da equação em t_i e y_i

Algoritmo/Função:

1. Definir o valor do passo do **h**
2. Criar um vetor que contém os valores para **t** a partir de **h**
3. Criar um vetor para armazenar as soluções nos diferentes pontos **t**
4. Definir o primeiro elemento do vetor ***y*** para o valor da condição inicial
5. Fazer um ciclo iterativo que percorra desde **1 a n** para calcular a solução aproximada utilizando a fórmula geral

```
h = (b-a)/n; %Amplitude de cada subintervalo

t = a:h:b; %Criar vetor que vai de "a" a "b" com step de "h"
y = zeros(1,n+1); %Alocamento de memória

y(1) = y0; %Definir o primeiro elemento do vetor

for i =1:n %Aplicar o Método de Euler (iteração)
    y(i+1) = y(i)+h*f(t(i),y(i));
end
```

Figura 5 - Algoritmo do método de Euler Melhorado

Método de RK2

Fórmulas:

O método de **Runge-Kutta de ordem 2 (RK2)** é um método bastante simples de se aplicar que permite resolver equações diferenciais ordinárias.
Requer apenas derivadas de primeira ordem e pode fornecer aproximações precisas.

Para resolvermos um problema de valor inicial (PVI) temos de utilizar a fórmula geral do método de RK2:

$$\boldsymbol{y}_{i+1} = \boldsymbol{y}_i + \frac{\mathbf{1}}{\mathbf{2}}(\boldsymbol{k}_1 + \boldsymbol{k}_2)$$

(com $i = 0, 1, 2, \dots, n-1$*)*

$\boldsymbol{y}_{i+1}$: Próximo valor aproximado da solução do problema na abcissa t_{i+1}
$\boldsymbol{y}_i$: Próximo valor aproximado da solução do problema na abcissa t_i

> Precisamos também de calcular dois valores, o k_1 e o k_2:

$$\boldsymbol{k}_1 = \boldsymbol{hf}(\boldsymbol{t}_i, \boldsymbol{y}_i)$$

$\boldsymbol{k}_1$: Inclinação no início do intervalo
$\boldsymbol{h}$: Valor de cada subintervalo
$\boldsymbol{f}(\boldsymbol{t}_i, \boldsymbol{y}_i)$: Valor da equação em t_i e y_i

$$\boldsymbol{k}_2 = \boldsymbol{hf}(\boldsymbol{t}_{i+1}, \boldsymbol{y}_i + \boldsymbol{k}_1)$$

$\boldsymbol{k}_2$: Inclinação no fim do intervalo
$\boldsymbol{t}_i$: Valor na abcissa atual
$\boldsymbol{h}$: Valor de cada subintervalo
$\boldsymbol{y}_i$: Valor aproximado da solução do problema na abcissa atual
$\boldsymbol{k}_1$: Inclinação no início do intervalo

Algoritmo/Função:

1. Definir o valor do passo do **h**
2. Criar um vetor que contém os valores para **t** a partir de **h**
3. Criar um vetor para armazenar as soluções nos diferentes pontos **t**
4. Definir o primeiro elemento do vetor ***y*** para o valor da condição inicial
5. Calcular o k_1 e o k_2 para os podermos utilizar na fórmula geral
6. Fazer um ciclo iterativo que percorra desde **1 a n** para calcular a solução aproximada utilizando a fórmula geral

```
h = (b-a)/n; %Amplitude de cada subintervalo

t = a:h:b; %Criar vetor que vai de "a" a "b" com step de "h"
y = zeros(1,n+1); %Alocamento de memória

y(1) = y0; %Definir o primeiro elemento do vetor

for i =1:n %Aplicar o Método de RK2 (iteração)
    k1 = h*f(t(i),y(i));
    k2 = h*f(t(i+1),y(i)+k1);
    y(i+1) = y(i)+(k1+k2)/2;
end
```

Figura 6 - Algoritmo do método de RK2

Método de RK4

Fórmulas:

Um método fiável e que atinge soluções bem próximas das soluções exatas é precisamente o método de **Runge-Kutta de ordem 4 (RK4)** que nos permite resolver equações diferenciais ordinárias.
O ponto positivo é que não precisamos de calcular nenhuma derivada, mas por outro lado precisamos calcular outra função que é definida avaliando **f** em diferentes pontos.

Para resolvermos um problema de valor inicial (PVI) temos de utilizar a fórmula geral do método de RK4:

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

$\boldsymbol{y_{i+1}}$: Próximo valor aproximado da solução do problema na abcissa t_{i+1}
$\boldsymbol{y_i}$: Próximo valor aproximado da solução do problema na abcissa t_i
$\boldsymbol{h}$: Valor de cada subintervalo

> Precisamos também de calcular quatro valores, o k_1, k_2, k_3 e o k_4:

$$\boldsymbol{k_1 = hf(t_i, y_i)}$$

$\boldsymbol{k_1}$: Inclinação no início do intervalo
$\boldsymbol{h}$: Valor de cada subintervalo
$\boldsymbol{f(t_i, y_i)}$: Valor da equação em t_i e y_i

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

$\boldsymbol{k_2}$: Inclinação no ponto médio do intervalo
$\boldsymbol{t_i}$: Valor na abcissa atual
$\boldsymbol{h}$: Valor de cada subintervalo
$\boldsymbol{y_i}$: Valor aproximado da solução do problema na abcissa atual
$\boldsymbol{k_1}$: Inclinação no início do intervalo

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

k_3 : Inclinação no ponto médio do intervalo
t_i : Valor na abcissa atual
h : Valor de cada subintervalo
y_i : Valor aproximado da solução do problema na abcissa atual
k_2 : Inclinação no ponto médio do intervalo

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

k_4 : Inclinação no final do intervalo
t_i : Valor na abcissa atual
h : Valor de cada subintervalo
y_i : Valor aproximado da solução do problema na abcissa atual
k_3 : Inclinação no ponto médio do intervalo

Algoritmo/Função:

1. Definir o valor do passo do **h**
2. Criar um vetor que contém os valores para **t** a partir de **h**
3. Criar um vetor para armazenar as soluções nos diferentes pontos **t**
4. Definir o primeiro elemento do vetor ***y*** para o valor da condição inicial
5. Calcular o k_1, k_2, k_3 e o k_4 para os podermos utilizar na fórmula geral
6. Fazer um ciclo iterativo que percorra desde **1 a n** para calcular a solução aproximada utilizando a fórmula geral

```
h = (b-a)/n; %Amplitude de cada subintervalo

t = a:h:b; %Criar vetor que vai de "a" a "b" com step de "h"
y = zeros(1,n+1); %Alocamento de memória

y(1) = y0; %Definir o primeiro elemento do vetor

for i = 1:n %Aplicar o Método de RK4 (iteração)
    k1 = h*f(t(i),y(i));
    k2 = h*f(t(i) + (h/2), y(i) + (1/2) * k1);
    k3 = h*f(t(i) + (h/2), y(i) + (1/2) * k2);
    k4 = h*f(t(i + 1), y(i) + k3);
    y(i+1) = y(i)+(k1+2*k2+2*k3+k4)/6;
end
```

Figura 7 - Algoritmo do método de RK4

Função ODE45

Fórmulas:

A **função ODE45**, ou *Ordinary Differntial Equation 45* é função nativa do MATLAB baseada no método de Runge-Kutta e é utilizada para resolver equações diferenciais ordinárias.

Ao chamarmos a função é-nos devolvido dois vetores: um vetor com os pontos de testagem e outro vetor com as soluções aproximadas em cada um desses pontos.

$$\textbf{ode45}(\boldsymbol{f}, \boldsymbol{t}, \boldsymbol{y_0})$$

$\boldsymbol{f}$: Equação diferencial
$\boldsymbol{t}$: Vetor que contém os pontos onde a função vai ser calculada
$\boldsymbol{y_0}$: Valor inicial do PVI (condição inicial)

Algoritmo/Função:

1. Definir o valor do passo do **h**
2. Gerar o vetor **t** a partir do **h** *(vetor a começar em **a** e a acabar em **b** com passo **h**)*
3. Criar um vetor que contém os valores para **t** a partir de **h**
4. Chamar a função ODE45 com os parâmetros

Método do Ponto Médio

Fórmulas:

A **função ODE45**, ou *Ordinary Differntial Equation* é função nativa do MATLAB baseada no método de Runge-Kutta e é utilizada para resolver equações diferenciais ordinárias.

Ao chamarmos a função é-nos devolvido dois vetores: um vetor com os pontos de testagem e outro vetor com as soluções aproximadas em cada um desses pontos.

$$\boldsymbol{y_{i+1} = y_i + hf\left(t_i + \frac{h}{2}, y_i \; + h * k_1\right)}$$

Método Explícito

$$\boldsymbol{y_{i+1} = y_i + hf\left(t_i + \frac{h}{2}, \frac{1}{2}(y_i \; + y_{i+1})\right)}$$

Método Implícito

$\boldsymbol{y_{i+1}}$: Próximo valor aproximado da solução do problema na abcissa t_{i+1}
$\boldsymbol{y_i}$: Próximo valor aproximado da solução do problema na abcissa t_i
$\boldsymbol{t_i}$: Valor na abcissa atual
$\boldsymbol{h}$: Valor de cada subintervalo

> Precisamos também de calcular mais um valor adicional, o k_1:

$$\boldsymbol{k_1 = \frac{1}{2} f(t_i, y_i)}$$

$\boldsymbol{k_1}$: Inclinação no início do intervalo
$\boldsymbol{h}$: Valor de cada subintervalo
$\boldsymbol{f(t_i, y_i)}$: Valor da equação em t_i e y_i

Algoritmo/Função:

1. Definir o valor do passo do **h**
2. Criar um vetor que contém os valores para **t** a partir de **h**
3. Criar um vetor para armazenar as soluções nos diferentes pontos **t**
4. Definir o primeiro elemento do vetor ***y*** para o valor da condição inicial
5. Calcular o $\mathbf{k_1}$ para o podermos utilizar na fórmula geral
6. Fazer um ciclo iterativo que percorra desde **1 a n** para calcular a solução aproximada utilizando o ponto médio explícito e o ponto médio implícito

```
h = (b-a)/n; %Amplitude de cada subintervalo

t = a:h:b; %Criar vetor que vai de "a" a "b" com step de "h"
y = zeros(1, n+1); %Alocamento de memória

y(1) = y0; %Definir o primeiro elemento do vetor

for i=1:n %Aplicar o Método do Ponto Médio (iteração)
    k1 = 0.5 * f(t(i), y(i));
    y(i+1) = y(i) + h*f(t(i) + h/2, y(i) + h*k1);
    y(i+1) = y(i) + h*f(t(i) + h/2, 0.5*(y(i) + y(i+1)));
end
```

Figura 8 - Algoritmo do método do Ponto Médio

Exemplos de aplicação e teste dos métodos

Exercício 3 do Teste Farol

$$\begin{cases} y' = -2ty \\ t \in (0, 1.5) \\ y_0 = 2 \end{cases}$$

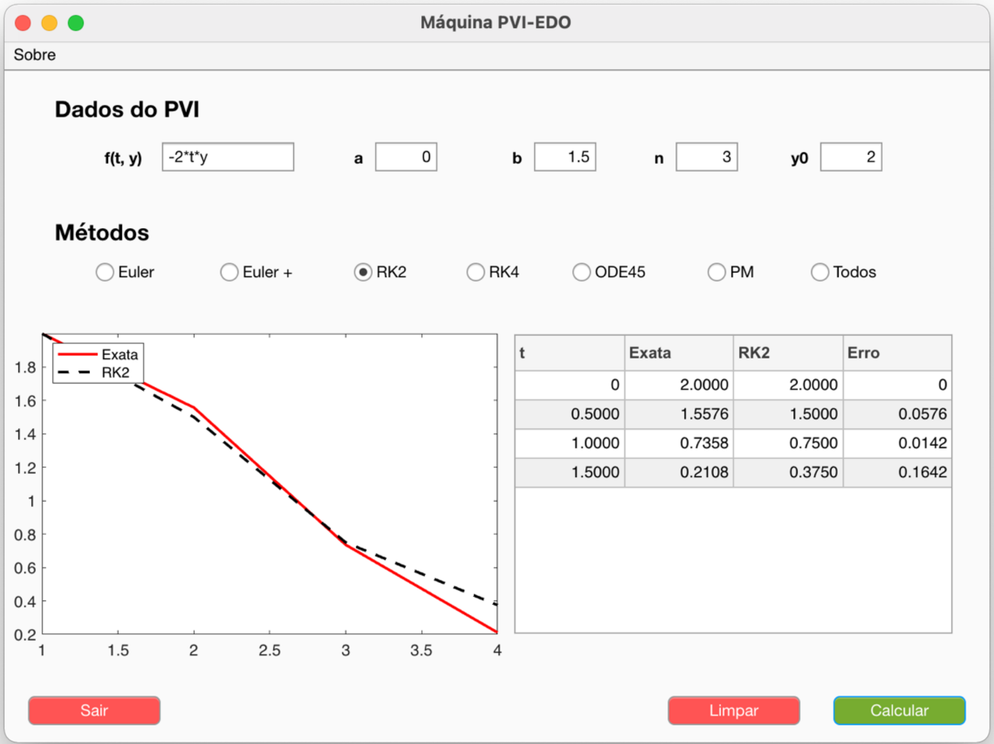
Equação Diferencial

Variável Independente

Condição Inicial

a) A solução **exata** do problema de valor inicial (PVI) é **0.2108**.

b)



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

Problemas de aplicação do livro

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 $g = 32\, ft/s^2$.

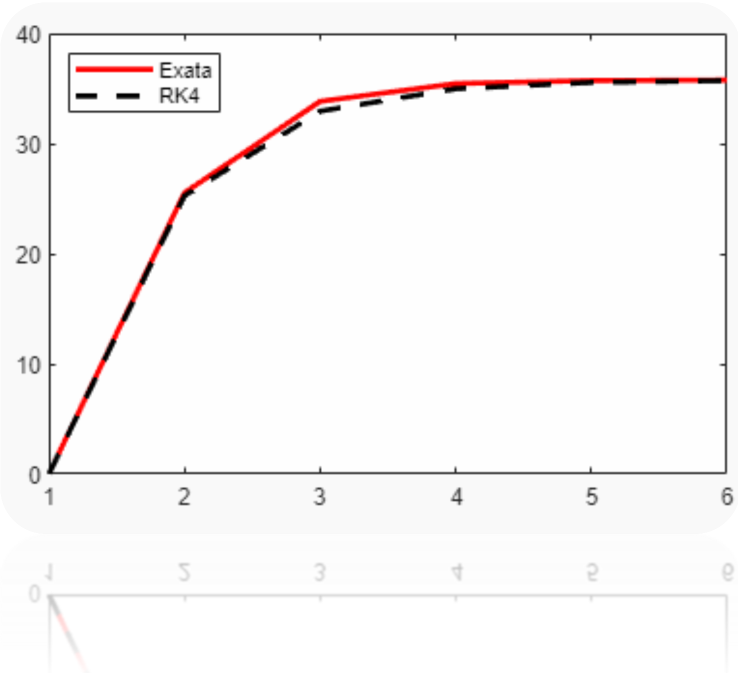
(a) Use the Runge-Kutta method with $h = 1$ to find an approximation to the velocity of the falling mass at $t = 5\, s$.

(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 $v(5)$.

a) Usando o método de Runge-Kutta 4 (RK4), com h = 1 e no intervalo de 0 a 5, o valor **aproximado** da velocidade é **35.7128ft/s** no instante t = 5.

b) Como se pode ver na figura abaixo, a velocidade **aproximada** é de **35.7128ft/s**.



c) O valor **exato** da velocidade no instante t = 5 é igual a **35.7678ft/s**.

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
$A(approximated)$					

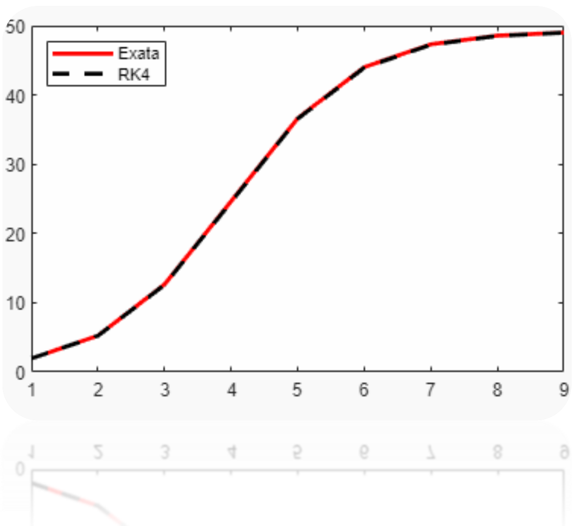
(b) Use a numerical solver to graph the solution of the initial-value problem. Estimate the values $A(1)$, $A(2)$, $A(3)$, $A(4)$, and $A(5)$ from the graph.

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

a)

$t(days)$	1	2	3	4	5
$A(observed)$	2.78	13.53	36.30	47.50	49.40
$A(approximated)$	1.9454	12.5844	36.5464	47.2519	48.9988

b) Como se pode ver na figura abaixo, a área **aproximada** está representada pela linha tracejada a preto.

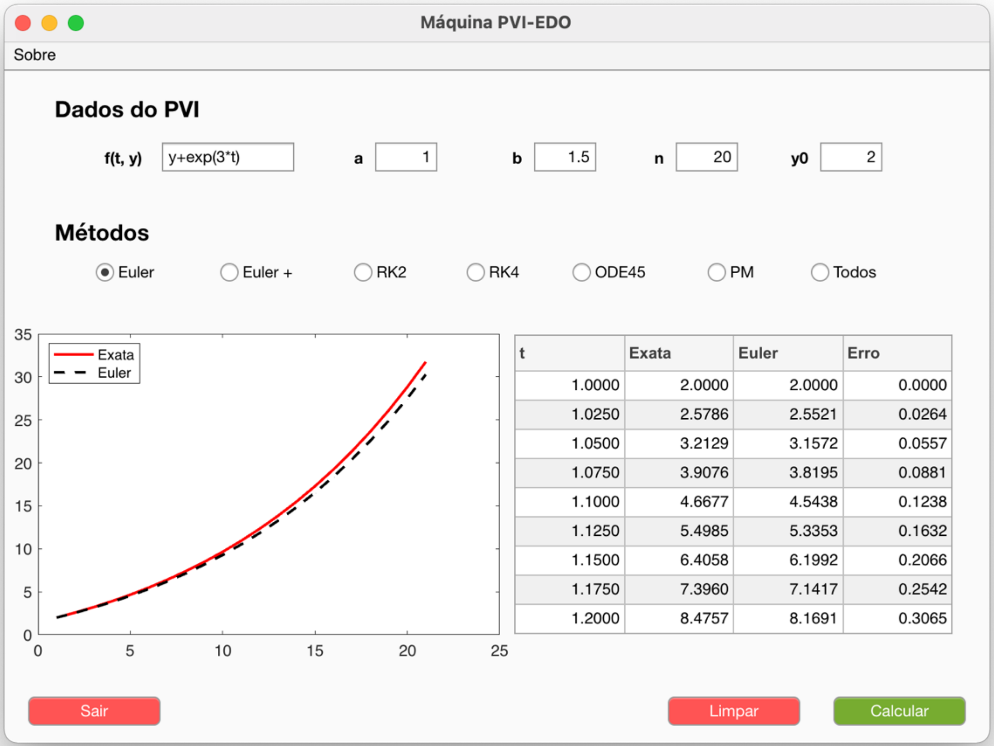


c)

$t(days)$	1	2	3	4	5
$A(observed)$	2.78	13.53	36.30	47.50	49.40
$A(approximated)$	1.9454	12.5844	36.5464	47.2519	48.9988
A(exata)	1.9454	12.6435	36.6282	47.3163	49.0196

Manual de Utilização

Ao abrir a aplicação ser-lhe-á apresentada a seguinte interface onde poderá encontrar 4 painéis distintos: um painel para a **introdução dos dados do problema de valor inicial (PVI)**, um painel para a **escolha do método** que deseja utilizar para resolver o PVI, um painel com o **plot dos gráficos** e por fim um painel que mostra a **tabela preenchida** com os valores.

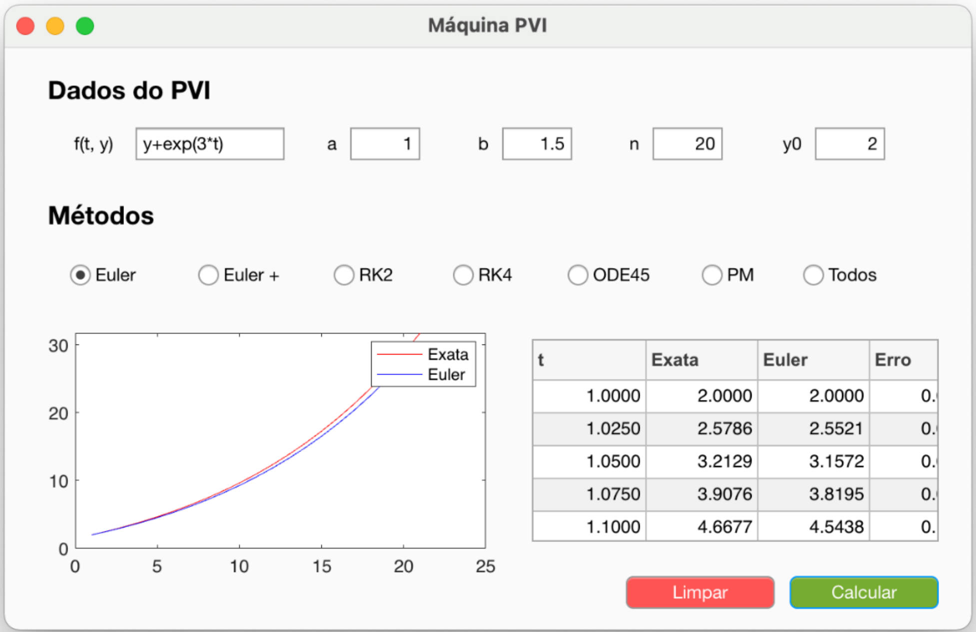


Os três botões que fazem parte da aplicação são os botões: **Limpar**, **Sair** e **Calcular**.

O botão **Limpar** vai limpar todos os campos, gráficos e tabelas que tenham sido preenchidos anteriormente. O botão **Calcular** vai calcular a solução do PVI com base nos valores introduzidos se todos os campos tiverem sido preenchidos corretamente.

Caso algum campo tenha sido preenchido de maneira incorreta o programa irá apresentar um pop-up a explicar o que é que correu mal.

Partindo do princípio que tudo correu bem, a aplicação mostrará finalmente o gráfico da solução exata assim como o gráfico do método selecionado, e também a tabela que mostra os valores de **t**, a **solução exata** no ponto t, a **solução aproximada** no ponto t do método selecionado e o **erro** que esse método cometeu.



Conclusão

Este trabalho permitiu-nos observar as diferenças entre os métodos em questão e perceber que existem métodos mais fiáveis do que outros, mas mais importante do que isso, tivemos a oportunidade de ver o quão úteis estes métodos são para a resolução de problemas do mundo real.

De um modo geral, concluímos que a função ODE45 se destacou imenso dos outros métodos tendo tanto uma melhor precisão como também conseguiu chegar à solução exata em muito poucas iterações. O método de Runge-Kutta de ordem 4 também foi um excelente competidor aproximando-se imenso da solução exata do problema. O método de Euler deixou um pouco a desejar sendo que os erros do método eram bastante maiores do que os erros dos outros métodos e a solução dada pelo método de Euler ficava bastantes vezes aquém da solução exata.

Graças ao tempo que despendemos neste trabalho, podemos dizer com toda a certeza que as nossas competências no que toca à programação em MATLAB beneficiaram imenso.