EP02 - Método de Euler Explícito

Alexsander Benatti da Silva (14555221)
Ana Paula Tavares da Fonseca (8557207)

22 de setembro de 2024

1 Introdução e Conceitos

O método de Euler é uma técnica de aproximação utilizada para resolver problemas de valor inicial. Embora simples e de precisão limitada, ele consegue oferecer uma base para a introdução de métodos mais avançados, sem envolver a complexidade algébrica que normalmente acompanha essas técnicas. Portanto, é uma ferramenta didática que permite compreender princípios básicos de métodos numéricos [1].

O objetivo do método de Euler é gerar aproximações para o problema de valor inicial expresso pela equação diferencial ordinária (EDO):

$$\frac{dy}{dt} = f(t, y), a \leq t \leq b, y(a) = \alpha \tag{1}$$

Em vez de fornecer uma solução contínua para $y(t)$, o método de Euler gera aproximações discretas de y em vários pontos, chamados de **pontos da malha**, dentro do intervalo $[a, b]$. Uma vez obtidas as soluções aproximadas nesses pontos, a solução em outros pontos do intervalo pode ser estimada por **interpolação** [1].

1.1 Definição de Pontos de Malha

Assumimos que os pontos da malha estão uniformemente distribuídos ao longo do intervalo $[a, b]$. Escolhemos um número inteiro positivo N e definimos os pontos da malha como:

$$t_i = a + ih, \quad i = 0, 1, 2, \ldots, N,$$

onde h é o **tamanho do passo**, calculado por

$$h = \frac{b - a}{N}, \tag{2}$$

que também é a distância comum entre os pontos consecutivos da malha, ou seja,

$$h = t_{i+1} - t_i.$$

1.2 Derivação do Método de Euler

O método de Euler pode ser derivado usando o **Teorema de Taylor**. Suponha que $y(t)$, a solução única da equação diferencial 1, possua derivadas contínuas até a segunda ordem no intervalo $[a, b]$. Usando a expansão de Taylor, podemos expressar $y(t_{i+1})$ em termos de $y(t_i)$ e suas derivadas como [1]:

$$y(t_{i+1}) = y(t_i) + hy'(t_i) + \frac{h^2}{2} y''(\xi_i),$$

onde ξ_i é um ponto entre t_i e t_{i+1}, e $h = t_{i+1} - t_i$. Como $y(t)$ satisfaz a equação diferencial original, temos que $y'(t) = f(t, y(t))$, de modo que podemos substituir essa relação na equação acima:

$$y(t_{i+1}) = y(t_i) + hf(t_i, y(t_i)) + \frac{h^2}{2}y''(\xi_i).$$

No método de Euler, ignoramos o termo de erro de ordem h^2 e mantemos apenas os termos de primeira ordem, resultando na seguinte fórmula para calcular as aproximações $w_i \approx y(t_i)$ em cada ponto $i = 1, 2, \dots, N$. Definimos $w_0 = \alpha$ e

$$\begin{aligned} w_0 &= \alpha \\ w_{i+1} &= w_i + hf(t_i, w_i), \quad i = 0, 1, 2, \dots, N-1. \end{aligned} \tag{3}$$

2 Implementação

Este código implementa o **método de Euler explícito** para resolver numericamente uma equação diferencial ordinária (EDO) do tipo 1. A função `step(t0,tf,n)` calcula o tamanho do passo h, que é a distância entre os pontos de malha, com base no intervalo $[a, b]$ e no número de divisões n, como definido em 2. A função `aprox(a, b, n, y0, fuser)` implementa o método de Euler explícito para resolver a equação diferencial dada pelo usuário no intervalo $[a, b]$ com n passos e condição inicial $y(a) = y_0$. Primeiramente, é criado um *array* `t_vals` contendo os valores de t igualmente espaçados. O *array* `y_values` armazena os valores aproximados de $y(t)$, sob a condição inicial `y_values[0] = y0`, em que `y0` é definido pelo usuário. Em seguida, itera-se para cada ponto t_i, calculando o valor aproximado de $y(t_{i+1})$ a partir de $y(t_i)$ conforme a fórmula em 3. Neste caso, $f(t_i, y_i)$ é calculado por meio de `diff_y(t_val, y_val, fuser)`, que calcula o valor da derivada da função $f(t, y)$ no ponto (t, y).

3 Testes

Os passos são feitos para os seguintes valores de n divisões: 16, 64, 256 e 1024, conforme solicitado no EP.

3.1 $y'(t) = t$ com $y(0) = 1$:

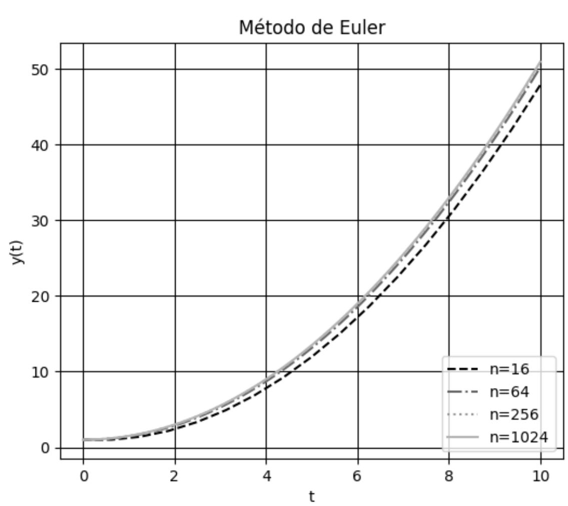

Figura 1: $y'(t) = t$ com $y(0) = 1$

3.2 $y'(t) = sin(t)$ com $y(0) = 1$:

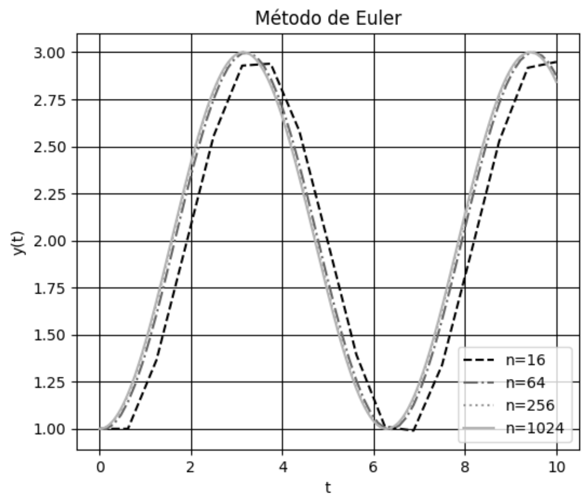

Figura 2: $y'(t) = sin(t)$ com $y(0) = 1$

3.3 $y'(t) = y$ com $y(0) = 1$:

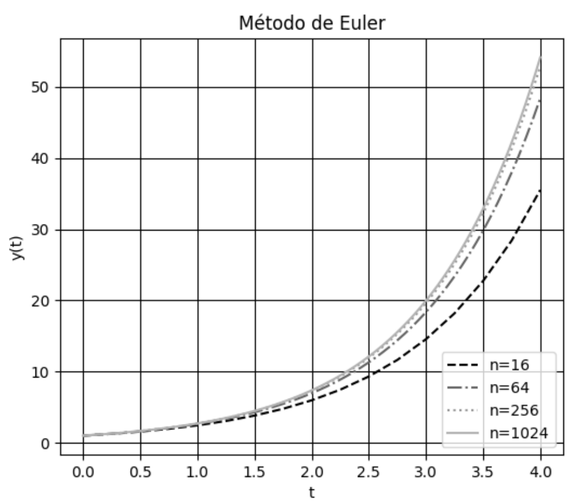

Figura 3: $y'(t) = y$ com $y(0) = 1$

3.4 Problema $\dot{x} = x(\alpha - \beta x)$

Vamos resolver o problema descrito no EP, $\dot{x} = x(\alpha - \beta x)$, sob condições de $t_0 = 0, t_n = 140, x_0 = 10$ e $n = 2500$.

3.4.1 $ab = 21$:

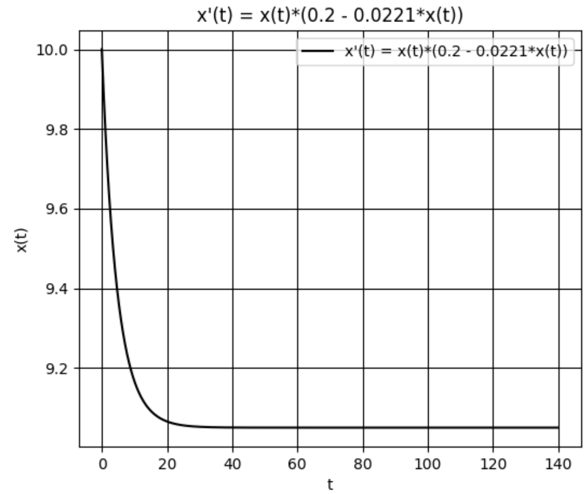

Figura 4: $\beta = 0.0221$

3.4.2 $ab = 07$:

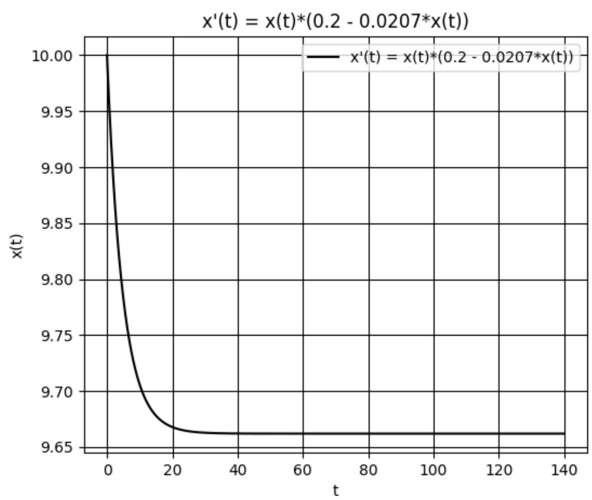

Figura 5: $\beta = 0.0207$

Referências

[1] R. L. Burden and J. D. Faires. *Numerical Analysis*. Brooks/Cole Cengage Learning, 2011.