Relatório da Questão 3

Erik Davino Vincent - BMAC - Turma 54
NUSP: 10736584

August 13, 2019

IME - USP

1 Introdução

O problema da questão 3 consiste no calculo da função $y(t)$ no ponto $t = 1$, dado que:

$$\begin{aligned} y(t_0) = y(0) = 1 \quad \text{e} \quad f(t, y(t)) = y'(t) = e^t(\sin(\omega t) + \omega\cos(\omega t)) \\ \omega = 2 \end{aligned} \tag{1}$$

Para tal deveria ser implementado algum método de um passo em um programa em python. No caso, foi escolhido o Método de Euler devido a sua simplicidade de implementação.

O método consiste em utilizar a aproximação da derivada

$$y'(t) \approx \frac{y(t + \Delta t) - y(t)}{\Delta t} \tag{2}$$

e de forma recursiva encontrar o valor no ponto desejado, a partir da discretização do domínio do problema, definido por $D(y(t)) = [t_0, t_f]$. Para tal definimos $n \in \mathbb{N}$ o número de passos/intervalos. Definimos $\Delta t = \frac{t_f - t_0}{n}$ o tamanho de cada intervalo. A partir da aproximação da derivada, obtemos:

$$y_{k+1} \approx y_k + \Delta t f(t_k, y_k) \tag{3}$$

e para o caso específico:

$$\begin{aligned} y_{k+1} \approx y_k + \Delta t y'(t) = y_k + \frac{1-0}{n} e^t(\sin(\omega t) + \omega\cos(\omega t)) \\ \omega = 2 \end{aligned} \tag{4}$$

Assim, cada instante do domínio pode ser aproximado a partir do instante anterior, dado um instante inicial.

No problema, devemos fazer o calculo múltiplas vezes, para valores cada vez maiores de n e calcular o erro absoluto. Quanto ao erro, isso só pode ser feito pois $y(t)$ pode ser obtido analiticamente (resolvido nas questões 1 e 2 da Tarefa 1):

$$\begin{aligned} y'(t) &= e^t(\sin(\omega t) + \omega\cos(\omega t)) \\ y(t) &= \int e^t \sin(\omega t) dt + \int e^t \omega\cos(\omega t) dt + 1 \\ y(t) &= e^t \sin(\omega t) - \int e^t \omega \cos(\omega t) dt + \int e^t \omega\cos(\omega t) dt + 1 \\ y(t) &= e^t \sin(\omega t) + 1 \\ \omega &= 2 \end{aligned} \tag{5}$$

2 Implementação

O código implementado em python pode ser encontrado no arquivo *Tarefa1_Q3.py*. O algorítimo não está adaptado para casos multidimensionais e não está otimizado; foi escrito de forma a resolver o problema indicado na questão. Como o programa é curto, não coloquei ele nesse relatório, pode ser facilmente compreendido lendo o arquivo.

O programa consiste em duas partes: a definição das funções necessárias como $y'(t)$ e do Método de Euler, e em seguida de uma função *main()* que serve apenas para gerar os gráficos da função.

Além disso, o erro absoluto ($|y(1) - y_{t_f}|$) e tempo de computação são calculados para cada n escolhido.

3 Resolução do problema

Abaixo seguem os resultados obtidos para $n = \{2, 5, 10, 25, 50, 100\}$. Os valores não são muito grandes, pois o método converge de forma muito suave e rápida para o resultado verdadeiro, como veremos adiante:

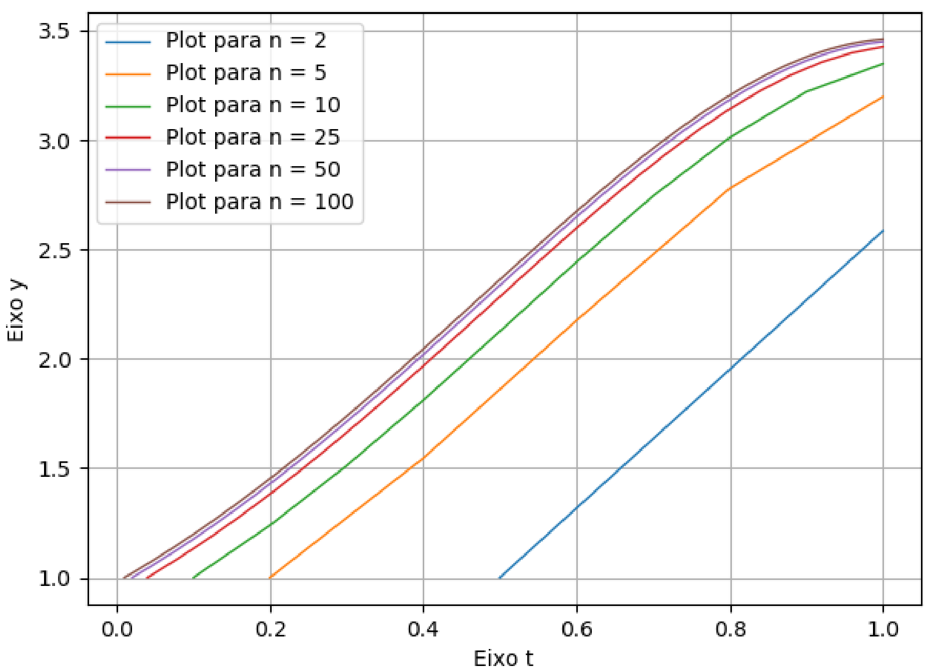
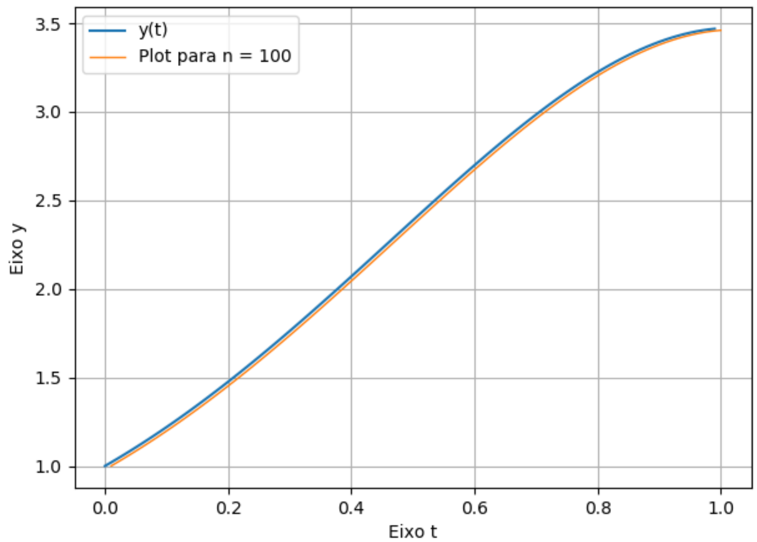


Figure 1: Plot da função $y(t)$ para os diferentes valores de n.

Podemos ver a partir desse gráfico que as curvas convergem para um ponto para quanto maior o n. Se observarmos os valores de y_{t_f} e os erros absolutos, esse resultado se torna ainda mais explícito:

- $n = 2$: erro $= 0.887$; tempo $= 0.0$
- $n = 5$: erro $= 0.274$; tempo $= 0.0$
- $n = 10$: erro $= 0.124$; tempo $= 0.0$
- $n = 25$: erro $= 0.046$; tempo $= 0.0$
- $n = 50$: erro $= 0.023$; tempo $= 0.0$
- $n = 100$: erro $= 0.011$; tempo $= 0.001$

Em seguida um gráfico para comparar $y(t)$ com o resultado obtido pelo método (Figure 2) e um gráfico para analisar a convergência (Figure 3):



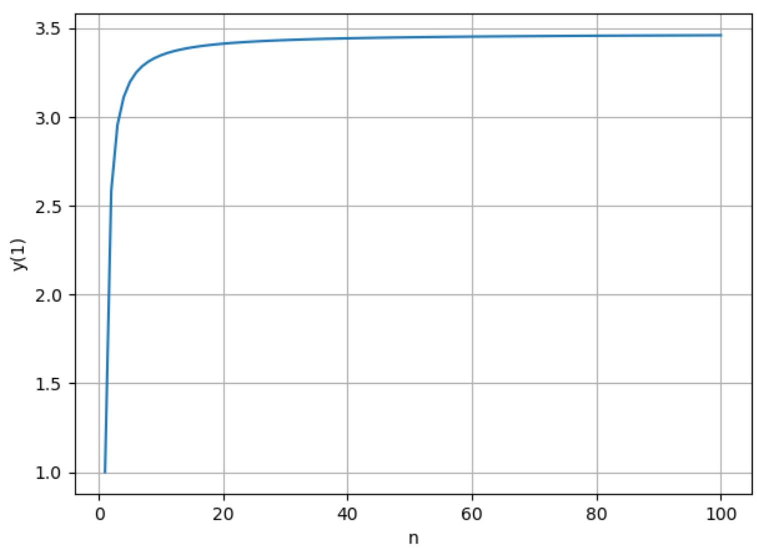


Figure 2: Gráfico comparando a curva real de $y(t)$ com o resultado obtido para $n = 100$.
Figure 3: Convergência do valor de y_1 para $y(t)$.