

Visualização de Campo de Direção para EDOs com Python

Luigi Carvalho Euzebio

Março 2025

Objetivo

O código apresentado feito utilizando a linguagem de programação Python - mais especificamente as bibliotecas `numpy` e `matplotlib` - tem como objetivo **plotar o campo de direção** de uma equação diferencial ordinária (EDO) de primeira ordem da forma:

$$\frac{dy}{dx} = f(x, y)$$

O campo de direção é uma representação gráfica que mostra, em diferentes pontos do plano (x, y) , a inclinação das soluções da EDO. Ele é útil para entender o comportamento qualitativo das soluções, mesmo quando a equação não pode ser resolvida analiticamente.

Definição da Função

A função principal é:

```
def campo_direcao(f, xlim=(-5, 5), ylim=(-5, 5), densidade=20, titulo="Campo de Direcao"):
```

Ela recebe os seguintes parâmetros:

- `f`: uma função de duas variáveis $f(x, y)$, que representa a derivada $\frac{dy}{dx}$.
- `xlim`, `ylim`: tuplas que determinam os limites dos eixos x e y .

- **densidade**: número de vetores a serem desenhados por eixo.
- **titulo**: título do gráfico.

Construção da Malha de Pontos

```
x = np.linspace(xlim[0], xlim[1], densidade)
y = np.linspace(ylim[0], ylim[1], densidade)
X, Y = np.meshgrid(x, y)
```

- Gera uma grade de pontos no plano (x, y) , onde os vetores serão desenhados.
- `np.linspace` cria vetores igualmente espaçados.
- `np.meshgrid` transforma esses vetores em matrizes 2D com todas as combinações possíveis.

Cálculo dos Vetores

```
U = np.ones_like(X)
V = f(X, Y)
```

- $U = 1$: o deslocamento horizontal (dx) é constante.
- $V = f(X, Y)$: o deslocamento vertical (dy) é dado pela função derivada.

Cada vetor representa a inclinação local da solução da EDO, ou seja:

$$(dx, dy) = (1, f(x, y))$$

Normalização dos Vetores

```
magnitude = np.sqrt(U**2 + V**2)
U_norm = U / magnitude
V_norm = V / magnitude
```

- Todos os vetores são normalizados para terem o mesmo comprimento.
- Isso facilita a visualização da direção, sem que vetores muito grandes dominem o gráfico.

Plotagem do Campo de Direção

```
plt.figure(figsize=(10, 6))
plt.quiver(X, Y, U_norm, V_norm, angles='xy')
plt.title(titulo, fontsize=14)
plt.xlabel("x")
plt.ylabel("y")
plt.grid(True)
plt.xlim(xlim)
plt.ylim(ylim)
plt.show()
```

- A função `quiver` do `matplotlib` é usada para desenhar os vetores.
- O gráfico é configurado com rótulos, limites e título.

Exemplos de Uso

Exemplo 1: $\frac{dy}{dx} = 3xy$

```
campo_direcao(lambda x, y: 3 * x * y, xlim=(-3, 3), ylim=(-3, 3), titulo=r"$\frac{dy}{dx} = 3xy$")
```

Exemplo 2: $\frac{dy}{dx} = y(2y - 1)$

```
campo_direcao(lambda x, y: y * (2 * y - 1), xlim=(-3, 3), ylim=(-2, 2), titulo=r"$\frac{dy}{dx} = y(2y - 1)$")
```

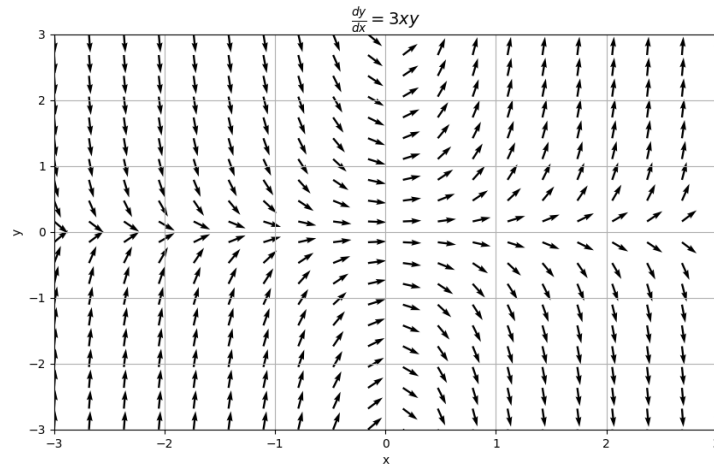


Figura 1: Campo de direção para $\frac{dy}{dx} = 3xy$

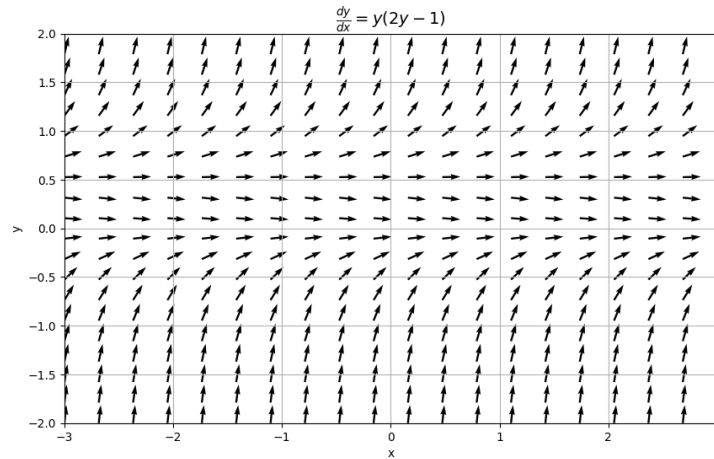


Figura 2: Campo de direção para $\frac{dy}{dx} = y(2y - 1)$

Conclusão

O código apresentado é uma ferramenta eficaz para a **visualização qualitativa** de EDOs de primeira ordem. Com o uso de `numpy` e `matplotlib`, é possível observar o comportamento das soluções de maneira gráfica, sem precisar resolver analiticamente a equação.