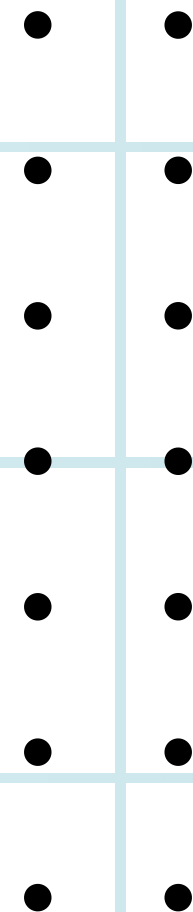


INTERPOLAÇÃO POR PARTES



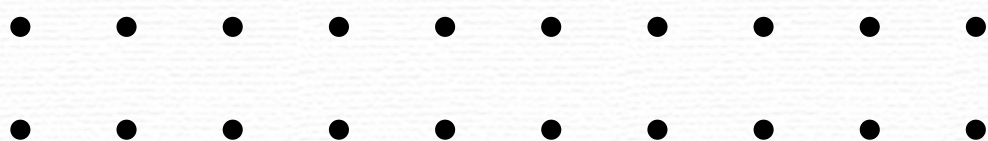


Interpolação por Partes - splines (lineares ou cúbicos)

Onde o método é usado? Na engenharia; física ...

Para modelar curvas de comportamento de materiais, vibrações, ou trajetórias.

Gráficos Computacionais e Animação: Curvas suaves como splines cúbicos são usadas para criar movimentos naturais e suaves em animações.





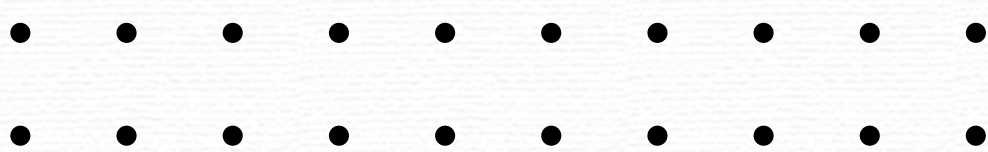
Interpolação por Partes - splines (lineares ou cúbicos)

Por que é importante?

Evita Oscilações Indesejadas; (Splines cúbicos, por exemplo, garantem suavidade e precisão sem oscilações).

Continuidade de Derivadas; (Splines cúbicos garantem continuidade da função e de suas derivadas até a segunda ordem).

Alta Precisão ,Flexível e Adaptável; (Pode lidar com grandes quantidades de dados experimentais e permitir ajustes locais sem alterar toda a curva).





CONCEITO

Interpolação por partes é um método de construir vários polinômios de grau baixo (geralmente 1 ou 3) entre pares de pontos consecutivos de uma tabela de dados.

É a construção de diferentes funções interpoladoras em subintervalos dos dados. O objetivo é reduzir oscilações indesejadas e melhorar a suavidade.





TIPOS DE SPLINES

Lineares:

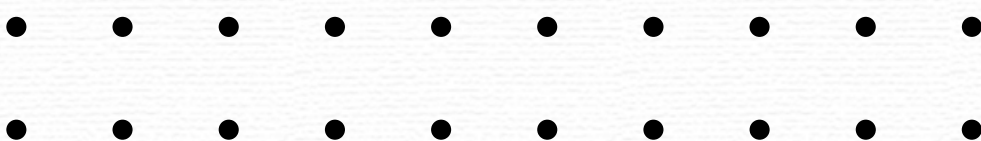
Grau do Polinômio : 1

Continuidade: Contínuo

Cúbico:

Grau do Polinômio : 3

Continuidade: Contínuo + derivadas 1ª e 2ª





TIPOS DE SPLINES

Spline Linear entre dois pontos (x_i, x_{i+1}):

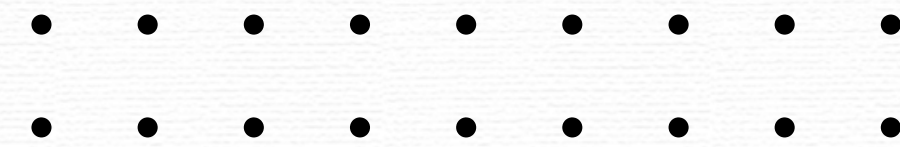
Se $f(x_i)$ e $f(x_{i+1})$ são dados:

$$S_i(x) = f(x_i) + \frac{f(x_{i+1}) - f(x_i)}{x_{i+1} - x_i} (x - x_i)$$

Spline Cúbico Natural (mais comum):

Cada spline $S_i(x)$ é um polinômio cúbico:

$$S_i(x) = a_i + b_i(x - x_i) + c_i(x - x_i)^2 + d_i(x - x_i)^3$$



TIPOS DE SPLINES

Com condições:

- $S_i(x_i) = f(x_i), S_i(x_{i+1}) = f(x_{i+1})$
- Derivadas de 1ª e 2ª ordem contínuas
- Condição natural: $S''(x_0) = S''(x_n) = 0$

Sistema é resolvido para obter coeficientes: b_i, c_i, d_i .



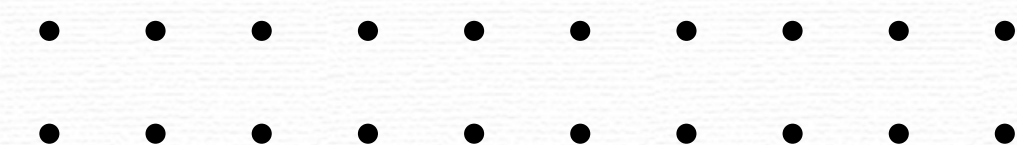
SPLINES LINEARES

Splines Lineares:

- Usam funções lineares entre cada par de pontos sucessivos.
- São simples e rápidas, mas não garantem suavidade nas derivadas.
- Conectam os pontos com segmentos de reta.
- Fáceis de implementar, mas com transições abruptas nas mudanças de inclinação.

Tipos de Splines:

- Naturais: impõem que a segunda derivada seja zero nas extremidades.
- Condicionados: utilizam informações adicionais, como inclinação nas bordas.





SPLINES CÚBICOS

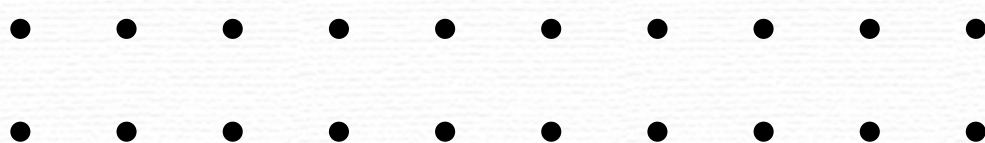
Splines Cúbicos :

Usam polinômios de grau 3; São os mais usados porque garantem continuidade da função; continuidade da 1ª e 2ª derivadas.

Ou seja utilizam polinômios cúbicos em cada subintervalo, com condições de continuidade da 1ª e 2ª derivadas.

Continuidade:

- C^0 : função contínua (sem saltos)
- C^1 : derivada contínua (sem "quebra" na inclinação)
- C^2 : segunda derivada contínua (suavidade na curvatura)





Continuidade de 1ª e 2ª derivadas.

Em uma interpolação por partes, temos vários polinômios diferentes que definem a função em intervalos distintos. Para que a função seja suave e realista, exigimos que:

A função seja contínua: Sem saltos, a curva deve passar "colada" entre os pontos.

$$S_i(x_{i+1}) = S_{i+1}(x_{i+1})$$

A 1ª derivada seja contínua: A inclinação da curva deve ser contínua, sem "quebras" visíveis. $S'_i(x_{i+1}) = S'_{i+1}(x_{i+1})$

A 2ª derivada também seja contínua: Garante uma transição suave da curvatura, isso evita mudanças bruscas na concavidade da curva. $S''_i(x_{i+1}) = S''_{i+1}(x_{i+1})$

Por que isso é importante? Garante que a curva seja suave e realista; Evita descontinuidades que podem causar falhas em simulações mecânicas ou visuais; Permite representar movimentos naturais e simulações confiáveis.



SPLINES NATURAIS VS CONDICIONADOS

Naturais:

- Impõem que a 2ª derivada nas extremidades seja zero, ou seja, as curvas "terminam retas".

Condição de Contorno:

$$S''(x_0) = 0 \text{ e } S''(x_n) = 0$$

Suavidade da curva:

- Mais suave nas extremidades, mas pode ser menos precisa

Controle de inclinação:

- Sem controle, pode achatar nas pontas

Condicionados:

- A derivada primeira nas extremidades é conhecida e fixada

Condição :

$$S'(x_0) = f'(x_0) \text{ e } S'(x_n) = f'(x_n)$$

Suavidade da curva:

- Segue a inclinação do fenômeno real nas bordas

Controle de inclinação:

- Total controle da inclinação inicial e final



SPLINES NATURAIS VS CONDICIONADOS

- **Splines naturais** :

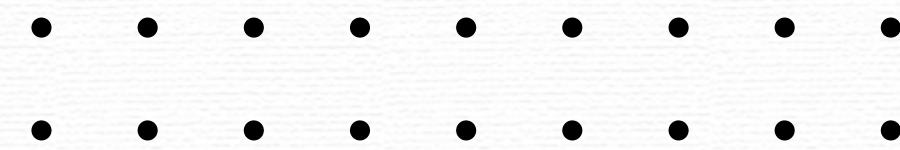
Tende a "relaxar" nas pontas. Bom quando não sabemos como o fenômeno se comporta nas bordas.

São mais usados quando não sabemos a inclinação nas extremidades.

- **Splines condicionados** :

Força a curva a ter determinada inclinação no início e no fim. Ideal quando isso é essencial, como em trajetórias, rampas, engenharia.

São ideais quando conhecemos as derivadas (inclinações) nas bordas, como em problemas físicos.



EXEMPLO

Exemplo	
x	f(x)
1	2
2	3
3	5

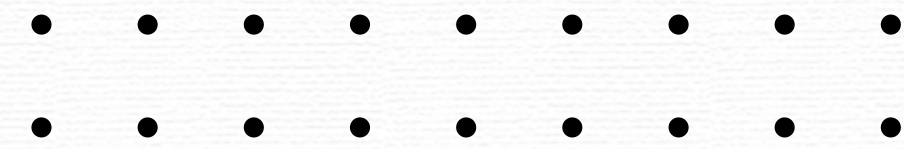
Spline Linear entre os 3 pontos:

Entre $x = 1$ e $x = 2$:

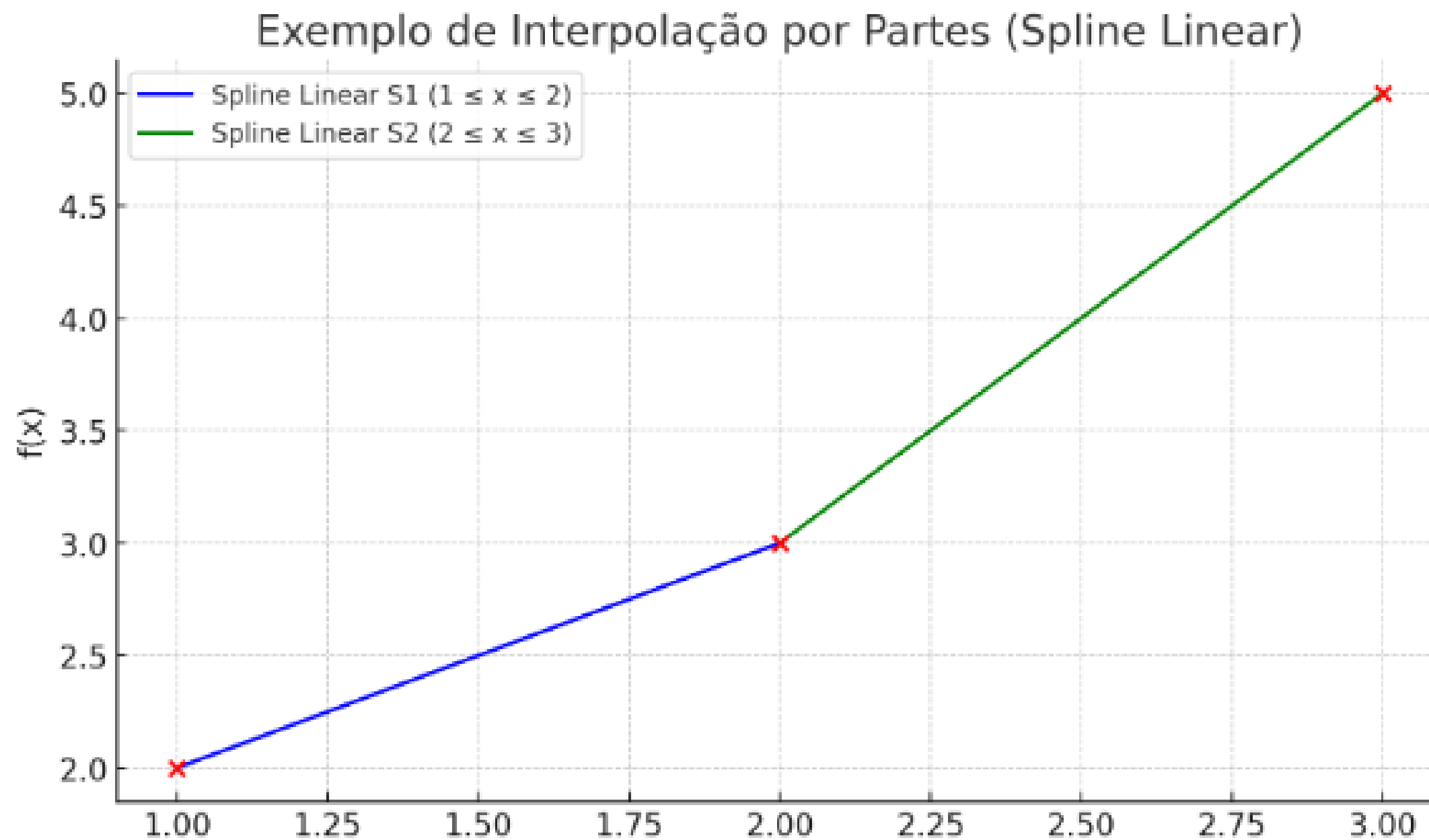
$$S_1(x) = 2 + \frac{3 - 2}{2 - 1}(x - 1) = 2 + (x - 1)$$

Entre $x = 2$ e $x = 3$:

$$S_2(x) = 3 + \frac{5 - 3}{3 - 2}(x - 2) = 3 + 2(x - 2)$$

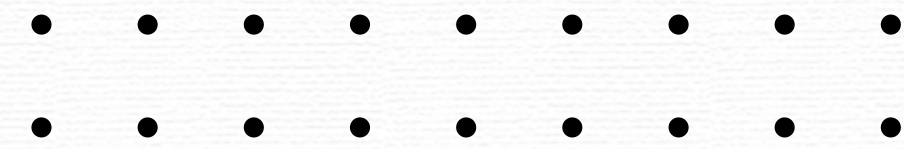


EXEMPLO

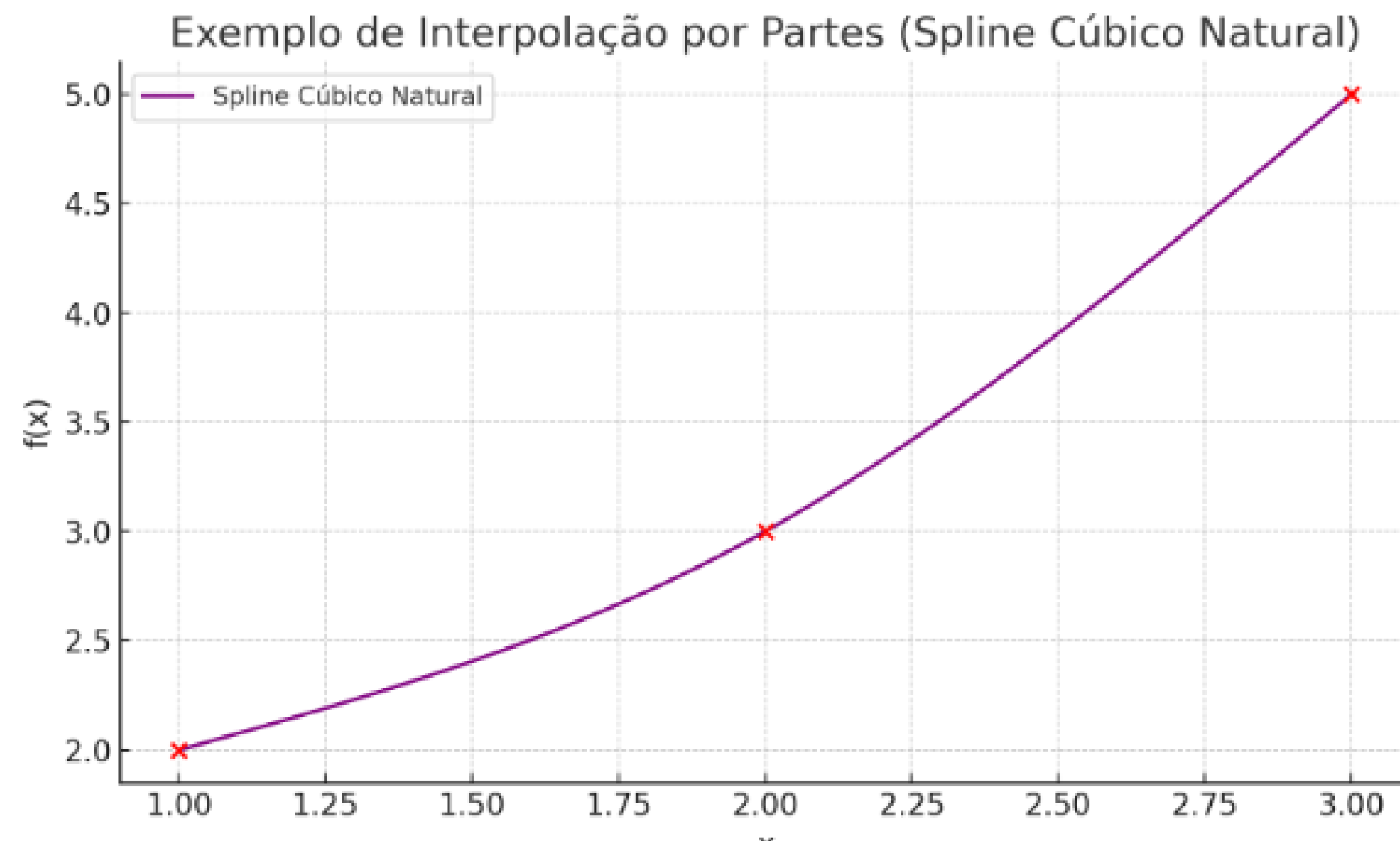


Cada segmento conecta dois pontos consecutivos com uma reta (grau 1)

A curva resultante é contínua, mas não suave nas junções (derivadas não contínuas)



EXEMPLO



A curva é suave, contínua e com derivadas também contínuas

Passa exatamente pelos pontos dados

Ideal para modelar dados reais com suavidade (como trajetórias, perfis, sinais etc.)



EXEMPLO (PYTHON)

- Aplicação: Previsão suave de temperatura ao longo do dia.
- Suponha que temos medições de temperatura em 4 horários:

Hora (x)	Temp (°C)
6	18
9	21
12	25
15	23

- Queremos interpolar suavemente a temperatura entre esses horários.



EXEMPLO (PYTHON)

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.interpolate import CubicSpline

# 1. Dados conhecidos (tempo em horas e temperatura)
horas = [6, 9, 12, 15]          # x
temperaturas = [18, 21, 25, 23] # f(x)

# 2. Criar a spline cúbica natural
spline = CubicSpline(horas, temperaturas, bc_type='natural')

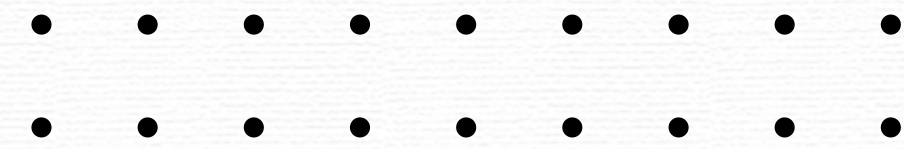
# 3. Gerar valores para interpolação (por exemplo, de hora em hora)
x_interp = np.linspace(6, 15, 200)
y_interp = spline(x_interp)

# 4. Plotar
plt.figure(figsize=(9, 5))
plt.plot(x_interp, y_interp, label='Spline Cúbico', color='purple')
plt.scatter(horas, temperaturas, color='red', label='Medições', zorder=5)
plt.title('Interpolação por Partes: Temperatura ao Longo do Dia')
plt.xlabel('Hora')
plt.ylabel('Temperatura (°C)')
plt.grid(True)
plt.legend()
plt.tight_layout()
plt.show()
```

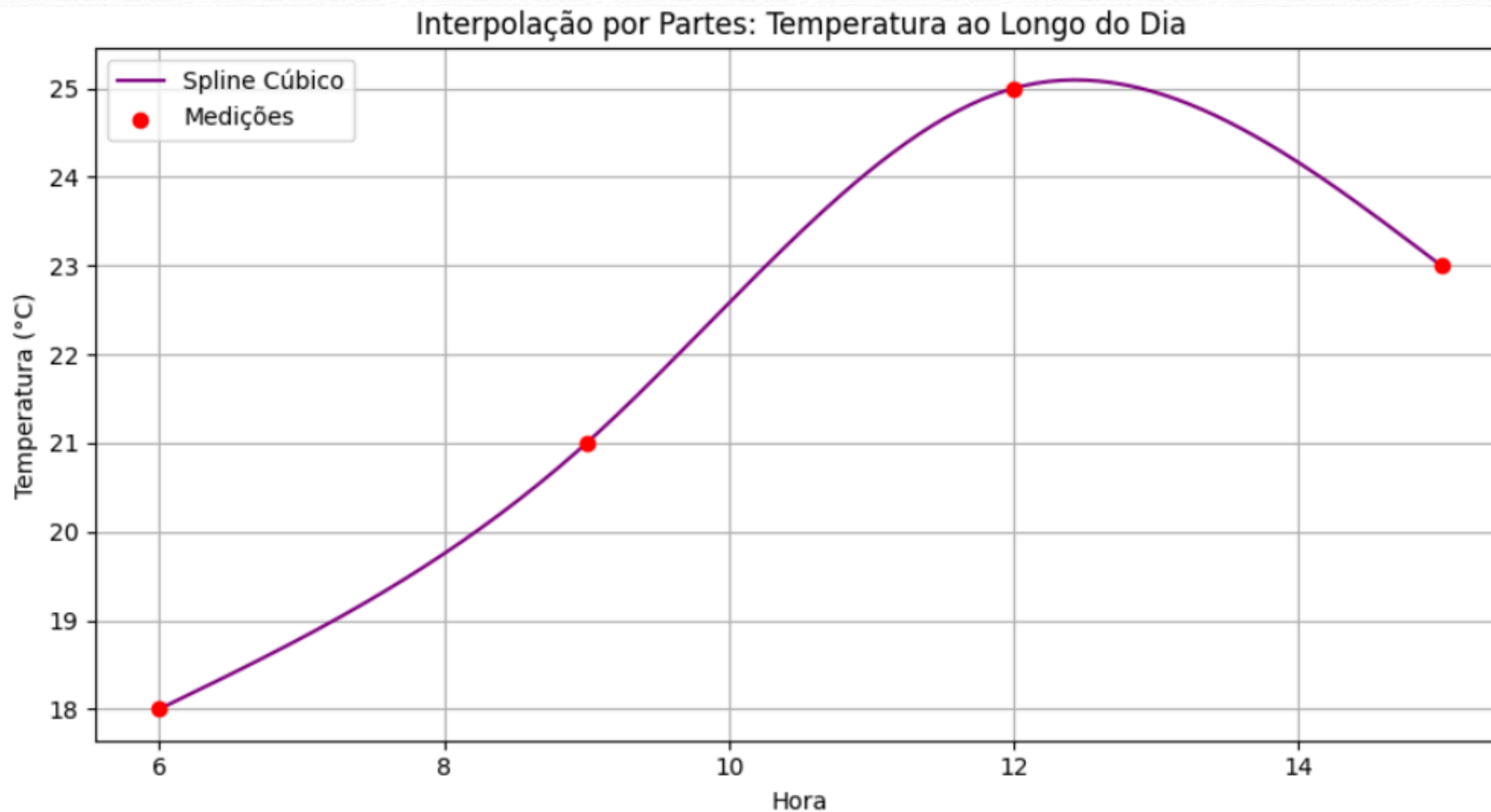
Os dados representam 4 medições de temperatura em horários diferentes.

Construção da spline cúbica:
Usamos CubicSpline da biblioteca scipy, que cria uma interpolação com suavidade.

Interpolação:
Geramos vários pontos entre 6h e 15h para formar a curva suave.



EXEMPLO (PYTHON)



O gráfico mostra as medições em vermelho e a curva estimada em roxo.

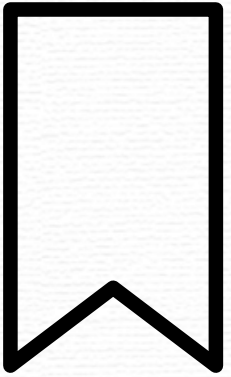
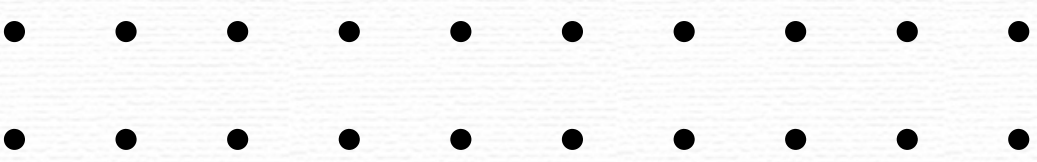
A curva suave que passa pelos 4 pontos e modela o comportamento da temperatura durante o dia com suavidade (sem quebras ou oscilações indesejadas).



Conclusão

A interpolação por partes, especialmente via splines, é essencial em cálculo numérico porque permite construir funções suaves, estáveis e precisas a partir de dados discretos. É amplamente utilizada na prática científica, engenharia, computação gráfica





Referências

- BURDEN, Richard L.; FAIRES, J. Douglas. Análise Numérica. 9. ed. São Paulo: Cengage Learning, 2011.
- CHAPRA, Steven C.; CANALE, Raymond P. Métodos Numéricos para Engenharia. 7. ed. Porto Alegre: AMGH, 2017.
- BORGES, Luiz Henrique de Figueiredo; RIBEIRO, Wilson Rosa. Cálculo Numérico: Aspectos Teóricos e Computacionais. 2. ed. Rio de Janeiro: LTC, 2009.
- ATKINSON, Kendall E. Introdução à Análise Numérica. 2. ed. Rio de Janeiro: LTC, 2008.

