



# Capítulo 20

## Interpolação por splines cúbicas

# Resumo do capítulo

- Quando polinômios de grau muito alto são necessários para interpolar uma função amostrada em um dado conjunto de pontos, erros numéricos são mais prováveis.
- Como alternativa, em vez de um só polinômio, usa-se vários polinômios, um para cada par de abcissas consecutivas, ajustando-se os coeficientes de forma a permitir uma transição suave de um polinômio para outro, garantindo continuidade sempre que possível.
- Esses “pedaços” de polinômios, em geral cúbicos, são chamados de *splines (cúbicas)*. Neste capítulo veremos algumas funções do **MATLAB** associadas ao uso de splines.
- Consulte também Help → **MATLAB** Help → **MATLAB** → Mathematics → Polynomials and Interpolation → Interpolation.

# A função *spline*

- Já vimos a função *spline* no capítulo sobre interpolação, no uso da função *interp1*.
- Há duas sintaxes possíveis:
  - *spline(x,y,xi)*: encontra os polinômios interpoladores dados os pontos com coordenadas em  $x, y$  e calcula o valor da spline no ponto  $xi$ .
  - *spline(x,y)*: encontra os polinômios interpoladores e retorna uma estrutura contendo todas as informações a respeito desses polinômios.
- Veja o exemplo no arquivo *mm2001.m*; experimente agora usar a segunda forma e olhar cada campo da estrutura resultante.
- A função *unmkpp(pp)* extrai todos os campos da estrutura de *pp* criada com o comando *pp=spline(x,y)*. Utilize essa função nos exemplos acima.

# Polinômios cúbicos hermitianos

- Quando os vetores  $x, y$  originam-se de uma função razoavelmente suave, splines cúbicas são uma boa solução.
- Caso contrário, podem resultar em funções com novos máximos e mínimos muito distantes dos dados originais, o que, entre outras coisas, pode destruir a monotonicidade dos dados originais.
- Uma solução é usar uma aproximação por pedaços de polinômios que respeitam a monotonicidade e convexidade da função descrita pelos pontos em  $x, y$ .
- Em **MATLAB**, a função para esse fim é *pchip*. Veja a descrição em *help pchip*.
- Veja também um exemplo no arquivo *mm2002.m*.

# Integração

- A integral de uma função interpolada por splines é, como era de se esperar, a soma das integrais de cada pedaço de polinômio, de cada spline.
- Em **MATLAB**, a função *mmpint(pp,c)*, onde *pp* é uma estrutura devolvida por *spline* ou *pchip* e devolve outra estrutura que contém os pedaços de polinômios que compõe as integrais dos pedaços de polinômios correspondentes de *pp*.
- Veja um exemplo no arquivo *mm2003.m*

# Diferenciação

- O argumento dado acima para a integração se aplica à diferenciação.
- A função correspondente, neste caso, é *mmppder(pp)*, onde *pp*, como antes, é uma estrutura devolvida por *spline*.
- Veja um exemplo no arquivo *mm2004.m*