INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE SANTA CATARINA
CAMPUS FLORIANÓPOLIS
DEPARTAMENTO ACADÊMICO DE ELETRÔNICA

COMPUTAÇÃO CIENTÍFICA

Prof. Marco Valério Miorim Villaça

Capítulo IV

Interpolação

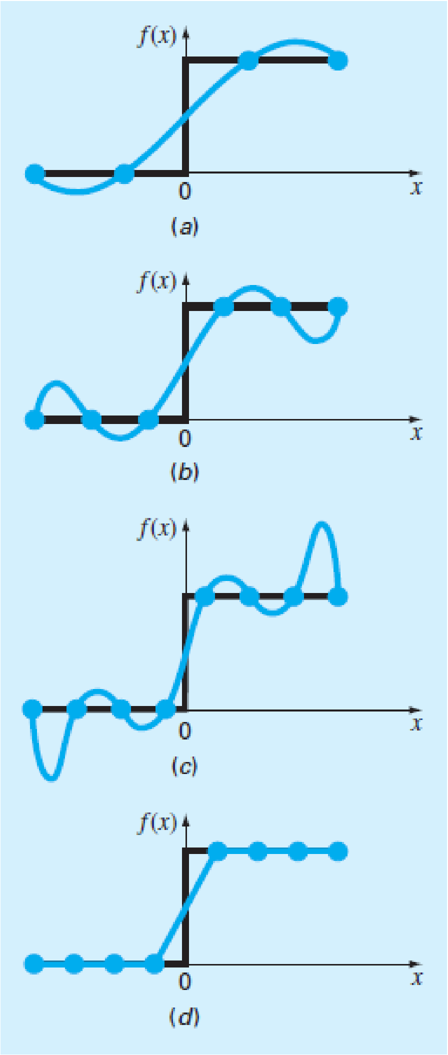
Parte II – Interpolação por splines e por partes

Interpolação Spline

- É uma técnica que divide o intervalo de interesse em vários subintervalos e interpola, da forma mais suave possível, nestes subintervalos com polinômios de pequeno grau .
- É uma alternativa as situações em que o número de pontos de interpolação é grande e a inexatidão na aproximação obtida com um polinômio leva a erros de arredondamento e oscilações.

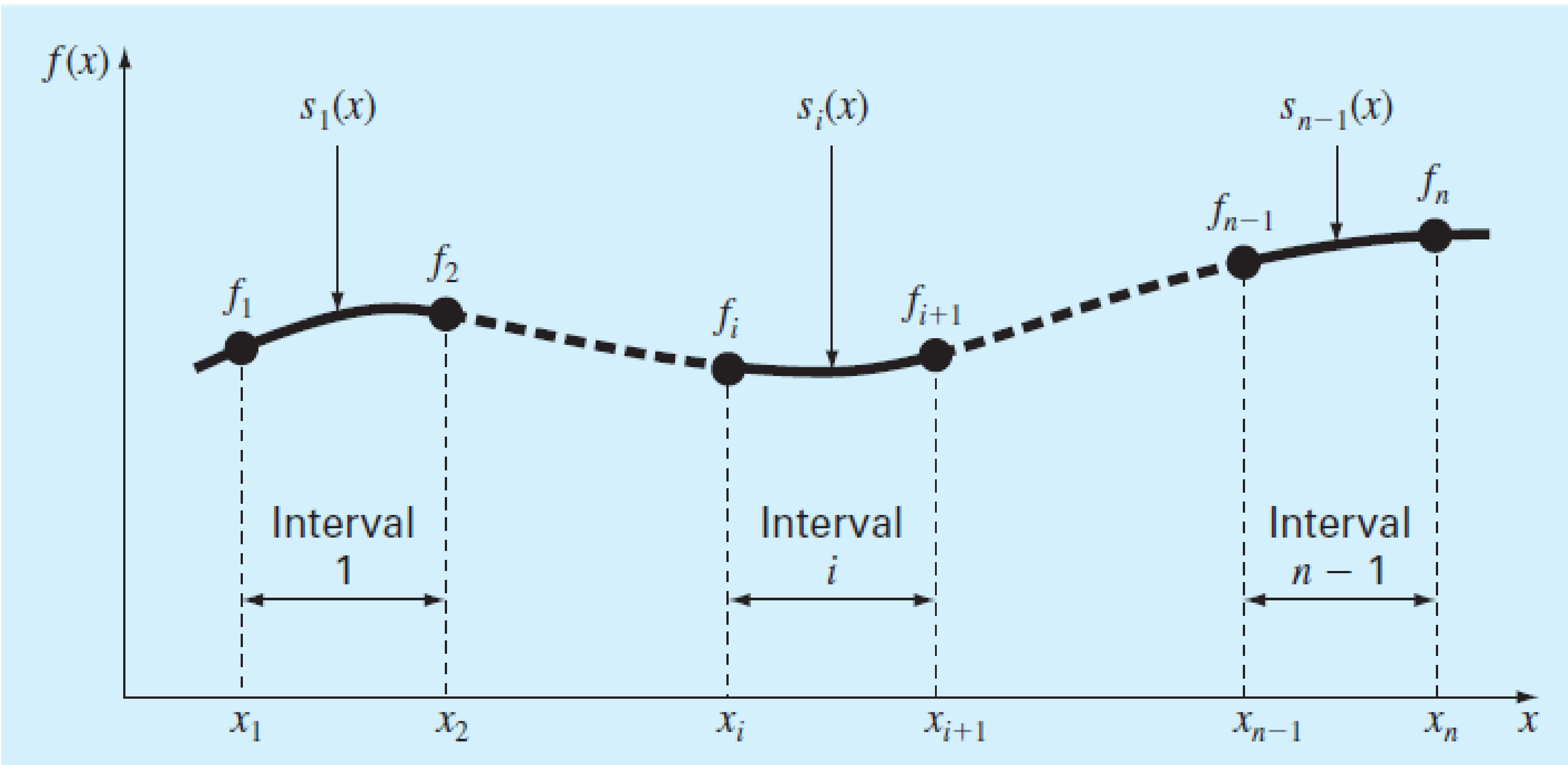
Interpolação Spline

- A figura ao lado ilustra uma situação na qual os *splines* são superiores aos polinômios de grau mais altas. Perceba que a variação abrupta da função em t = 0 provoca oscilações nos polinômios interpoladores (a, b e c).
- A *spline* linear (d) fornece uma aproximação muito mais aceitável.



Splines
Notação

- Notação usada para deduzir splines.
  - Observe que há *n-1* intervalos para *n* pontos dados.



Splines lineares

- Cada função é uma reta que liga dois pontos das extremidades dos intervalos:

$$s_i(x) = a_i + b_i(x - x_i)$$

- Para x = x_i,

$$s_i(x_i) = a_i$$

ou

$$a_i = f_i$$

e b_i é a inclinação da reta ligando os dois pontos:

$$b_i = \frac{f_{i+1} - f_i}{x_{i+1} - x_i}$$

onde f_i é uma abreviação para $f(x_i)$.

Splines lineares

- Substituindo os valores de a_i e b_i na equação de $s_i(x)$, obtém-se

$$s_i(x)=f_i+\frac{f_{i+1}-f_i}{x_{i+1}-x_i}(x-x_i)$$

- Essa equação podem ser usada para calcular o valor da função em qualquer intervalo

- Exemplo 1: Ajuste os dados da Tabela com *splines* de1^0 grau:

i	xi	fi
1	3,0	2,5
2	4,5	1
3	7,0	2,5
4	9,0	0,5

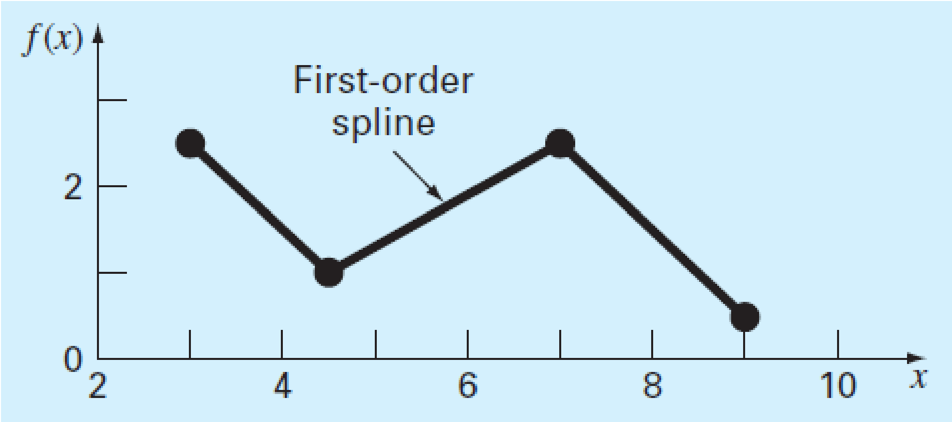
Splines lineares

- Para o primeiro intervalo

$$s_1(x) = 2{,}5 + \frac{1{,}0 - 2{,}5}{4{,}5 - 3{,}0}(x-3)$$

$$s_1(x) = 2{,}5 - (x-3)$$

- As equações em outros intervalos podem ser calculadas e o spline de primeiro grau resultante é representado abaixo.



Splines lineares

- A principal desvantagem desses splines é que eles não são suaves.
- Nos pontos dados, onde dois splines se encontram (nós), a inclinação varia abruptamente.
- Em termos formais a derivada primeira da função é descontínua nesses pontos.
- Isso indica a utilização de splines polinomiais de grau superior.

Splines quadráticos

- Para garantir que as *n-ésimas* derivadas sejam contínuas, um spline de grau *n+1* deve ser usado.
- Os splines cúbicos ou de terceiro grau garantem a continuidade da primeira e segundas derivadas, sendo por isso mais utilizados.
- Devido a maior complexidade da dedução dos splines cúbicos, nesse material será apresentada apenas a dedução dos splines quadráticos.
- Para os splines cúbicos, apresentaremos apenas as equações finais.

Splines quadráticos

- O polinômio para cada intervalo pode ser representado de forma geral como:

$$s_i(x) = a_i + b_i(x - x_i) + c_i(x - x_i)^2 \qquad (i)$$

- Observando a figura da página 5, perceba que para n pontos dados, existem n-1 intervalos e, consequentemente 3(n-1) constantes indeterminadas – *a's, b's e c's.*
- Portanto, *3(n-1)* equações ou condições são necessárias para calcular as incógnitas.

Splines quadráticos
Equações

1. Condição de continuidade: a função deve passar por todos os pontos. Assim:

$$f_i = a_i + b_i(x_i - x_i) + c_i(x_i - x_i)^2$$

o que fornece:

$$a_i = f_i \qquad (ii)$$

Resultado que pode ser incorporado a (i):

$$f_i(x) = f_i + b_i(x - x_i) + c_i(x - x_i)^2 \qquad (iii)$$

A determinação dos coeficientes a_i, reduz o número de condições para *2(n-1).*

Splines quadráticos
Equações

2. Os valores da função de polinômios adjacentes devem ser iguais nos nós, condição que pode ser escrita para o nó *i +1*

$$f_i + b_i(x_{i+1} - x_i) + c_i(x_{i+1} - x_i)^2 = f_{i+1} + b_{i+1}(x_{i+1} - x_{i+1}) + c_{i+1}(x_{i+1} - x_{i+1})^2 \quad (iv)$$

como:

$$h_i = x_{i+1} - x_i$$

A eq. (iv) simplifica para :

$$f_i + b_i h_i + c_i h_i^2 = f_{i+1} \qquad (v)$$

Essa equação pode ser escrita para i = 1, 2, …, n-1 o que equivale *a (n-1)* condições, significando que agora faltam apenas *(n-1)* condições.

Splines quadráticos
Equações

3. Condição de “suavidade” - as derivadas primeiras nos nós internos devem ser iguais. Derivando-se (i), obtém-se

$$s_i'(x) = b_i + 2c_i(x - x_i)$$

Condição que escrita para o nó (i+1) resulta em:

$$b_i + 2c_i(x_{i+1} - x_i) = b_{i+1} + 2c_{i+1}(x_{i+1} - x_{i+1})$$

ou:

$$b_i + 2c_i h_i = b_{i+1} \qquad (vi)$$

Essa equação escrita para os nós interiores resulta em (n-2) condições, significando que agora faltam apenas (n- 1) – (n-2) = 1 condição.

Splines quadráticos
Equações

4. Arbitrariamente, considere a derivada segunda no primeiro ponto nula. Como a derivada segunda de (i) é 2 c_i, resulta que: $c_1 = 0$

- Condição que indica que os dois primeiros pontos serão ligados por uma reta.
- Exemplo 2: Ajuste um *spline quadrático* aos mesmos dados empregados no Exemplo 1. Use os resultados para fazer uma estimativa em x = 5.

  4 pontos → (n – 1) = 3 intervalos.

  Isso implica que após aplicar a condição de continuidade (1) e a condição da derivada segunda nula, restam

  2(n-1) – 1 = 5 condições.

Splines quadráticos
Exemplo

A equação (v) é escrita para i = 1 a 3 com $c_1 = 0$, resultando em:

$$f_1 + b_1 h_1 = f_2$$
$$f_2 + b_2 h_2 + c_2 h_2^2 = f_3$$
$$f_3 + b_3 h_3 + c_3 h_3^2 = f_4$$

A continuidade das derivadas, equação (vi) produz (n-2) equações adicionais, o que resulta (lembrando que $c_1 = 0$):

$$b_1 = b_2$$
$$b_2 + 2 c_2 h_2 = b_3$$

Splines quadráticos
Exemplo

As equações podem ser reordenadas:

$$b_1 h_1 = f_2 - f_1$$
$$b_2 h_2 + c_2 h_2^2 = f_3 - f_2$$
$$b_3 h_3 + c_3 h_3^2 = f_4 - f_3$$
$$b_1 - b_2 = 0$$
$$b_2 - b_3 + 2 c_2 h_2 = 0$$

- Colocando na forma matricial:

$$\begin{bmatrix} 1{,}5 & 0 & 0 & 0 & 0 \\ 0 & 2{,}5 & 0 & 6{,}25 & 0 \\ 0 & 0 & 2 & 0 & 4 \\ 1 & -1 & 0 & 0 & 0 \\ 0 & 1 & -1 & 5 & 0 \end{bmatrix} \cdot \begin{Bmatrix} b_1 \\ b_2 \\ b_3 \\ c_2 \\ c3 \end{Bmatrix} = \begin{Bmatrix} -1{,}5 \\ 1{,}5 \\ -2 \\ 0 \\ 0 \end{Bmatrix}$$

Splines quadráticos
Exemplo

Resolvendo a o sistema no Scilab, resulta:

$$b_1 = -1$$
$$b_2 = -1$$
$$b_3 = 2{,}2$$
$$c_2 = 0{,}64$$
$$c_3 = 1{,}6$$

- Esses resultados juntamente com os valores $a_i = f_i$ e $c_1 = 0$ são substituídos na equação (i) para construção dos *splines* quadráticos:

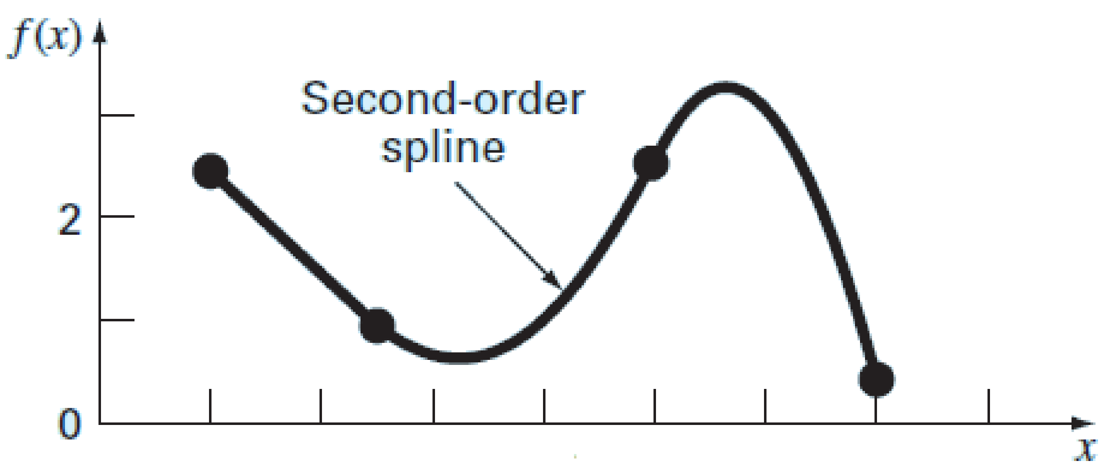
$$s_1(x) = 2{,}5 - (x-3)$$
$$s_2(x) = 1 - (x-4{,}5) + 0{,}64(x-4{,}5)^2$$
$$s_3(x) = 2{,}5 - 2{,}2(x-7) + 1{,}6(x-7)^2$$

Splines quadráticos
Exemplo

Como x = 5 está no segundo intervalo:

$$s_2(5)=1-(5-4{,}5)+0{,}64(5-4{,}5)^2=0{,}66$$

- O ajuste completo do *spline quadrático* é mostrado na figura abaixo. O ajuste é prejudicado pela reta unindo os 2 primeiros pontos e pelos valores mínimo do segundo intervalo e o máximo do terceiro parecerem inadequados



Splines cúbicos

- São os mais utilizados pois oferecem a representação mais simples com a aparência desejada de suavidade.
- Polinômios de grau superior tendem a exibir instabilidades.
- O polinômio para cada intervalo pode ser representado de forma geral como:

$$s_i = a_i + b_i(x - x_i) + c_i(x - x_i)^2 + d_i(x - x_i)^3 \qquad (vii)$$

- Nesse caso existem 4(n-1) constantes indeterminadas e, portanto, 4*(n-1)* condições são necessárias.

Splines cúbicos
Equações

- Nestas equações, assume-se que as segundas derivadas no primeiro e último nós são iguais a zero.
- A especificação de tais condições nas extremidades recebe o nome de *spline* natural.
- Determinação das constantes *a*

$$a_i = f_i \qquad (viii)$$

Splines cúbicos
Equações

Determinação das constantes *c*

$$\begin{bmatrix} 1 & & & & & \\ h_1 & 2(h_1+h_2) & h_2 & & & \\ & \ddots & \ddots & \ddots & & \\ & & h_{n-2} & 2(h_{n-2}+h_{n-1}) & h_{n-1} \\ & & & & 1 \end{bmatrix} \cdot \begin{Bmatrix} c_1 \\ c_2 \\ \vdots \\ c_{n-1} \\ c_n \end{Bmatrix} = \begin{Bmatrix} 0 \\ 3(f[x_3, x_2] - f[x_2, x_1]) \\ \vdots \\ 3(f[x_n, x_{n-1}] - f[x_{n-1}, x_{n-2}]) \\ 0 \end{Bmatrix}$$

(ix)

- Note que além dos coeficientes c_1, c_2, …, c_{n-1}, o sistema envolve um coeficiente c_n que não está diretamente relacionado com nenhum dos (n-1) polinômios s_i.

Splines cúbicos
Equações

Determinação das constantes *c*

- Na realidade c_n está diretamente relacionado com as condições no extremo do intervalo de interpolação e sua determinação depende do tipo de spline que está sendo construído

Splines cúbicos
Equações

Determinação das constantes *b e d*

$$b_i = \frac{f_{i+1} - f_i}{h_i} - \frac{h_i}{3}(2c_i + c_{i+1}) \qquad (x)$$

$$d_i = \frac{c_{i+1} - c_i}{3h_1} \qquad (xi)$$

Splines cúbicos naturais
Exercício

Exemplo 3 – Ajuste splines cúbicos naturais aos mesmos dados usados nos exemplos 1 e 2. Utilize os resultados para fazer uma estimativa do valor em x = 5.

- O primeiro passo é empregar a equação (ix) para gerar um conjunto de equações simultâneas que nos fornecerão os coeficientes c

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 1{,}5 & 8 & 2{,}5 & 0 \\ 0 & 2{,}5 & 9 & 2 \\ 0 & 0 & 0 & 1 \end{bmatrix} . \begin{Bmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \end{Bmatrix} = \begin{Bmatrix} 0 \\ 4{,}8 \\ -4{,}8 \\ 0 \end{Bmatrix}$$

$$c_1 = 0$$
$$c_2 = 0.839543726$$
$$c_3 = -0{,}766539924$$
$$c_4 = 0$$

Splines cúbicos naturais
Exercício

- As equações (x) e (xii) podem ser usadas para calcular os coeficientes *b* e *d*:

$$b_1 = -1{,}4197719 \qquad d_1 = 0{,}1865653$$
$$b_2 = -0{,}1604563 \qquad d_2 = -0{,}2141445$$
$$b_3 = 0{,}0220532 \qquad d_3 = 0{,}1277567$$

- Esses resultados, juntamente com os valores de *a* (eq. viii) podem ser substituídos na eq. (vii) para obter os splines cúbicos:

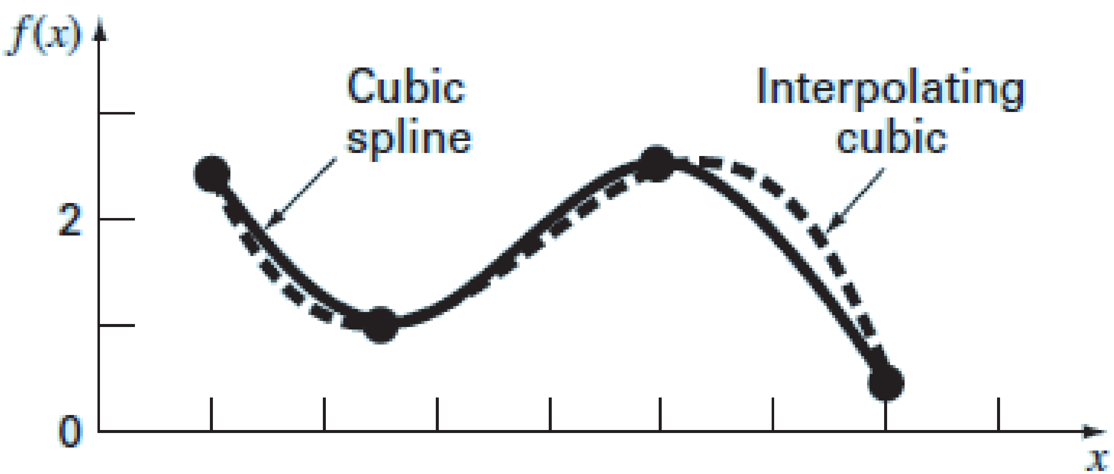
$$s_1(x) = 2{,}5 - 1{,}4997719(x-3) + 0{,}1865653(x-3)^3$$
$$s_2(x) = 1{,}0 - 0{,}1604563(x-4{,}5) + 0{,}8395437(x-4{,}5)^2 - 0{,}2141445(x-4{,}5)^3$$
$$s_3(x) = 2{,}5 + 0{,}0220532(x-7) - 0{,}7665399(x-7)^2 + 0{,}1277567(x-7)^3$$

Splines cúbicos naturais
Exercício

- As equações anteriores podem ser utilizadas para calcular valores dentro da cada intervalo. Como x = 5 está contido no segundo intervalo, resulta:

$$s_2(5)=1-0{,}1604563(0{,}5)+0{,}8395437(0{,}5)^2-0{,}2141445(0{,}5)^3=1{,}1028897$$

- A Figura abaixo mostra o ajuste completo do spline cúbico.



Splines cúbicos
Condições nas extremidades

Além do spline natural, outros dois tipos de splines são polulares:

- Condição de extremidade amarrada (*clamped*):

  Essa condição envolve a especificação das primeiras derivadas no primeiro e últimos nós (amarra a inclinação nesses nós).

- Condição de extremidade sem um nó (*Not-a-Knot*):

  Força a continuidade da derivada terceira no segundo e penúltimo nó. Desse modo, esses nós não representam mais a junção de duas funções cúbicas diferentes, não sendo mais nós verdadeiros.

Perigos da interpolação com polinômios de grau elevado

- A função de Runge é um exemplo conhecido de uma função que não pode ser bem ajustada com polinômios:

$$f(x) = \frac{1}{1+25x^2}$$

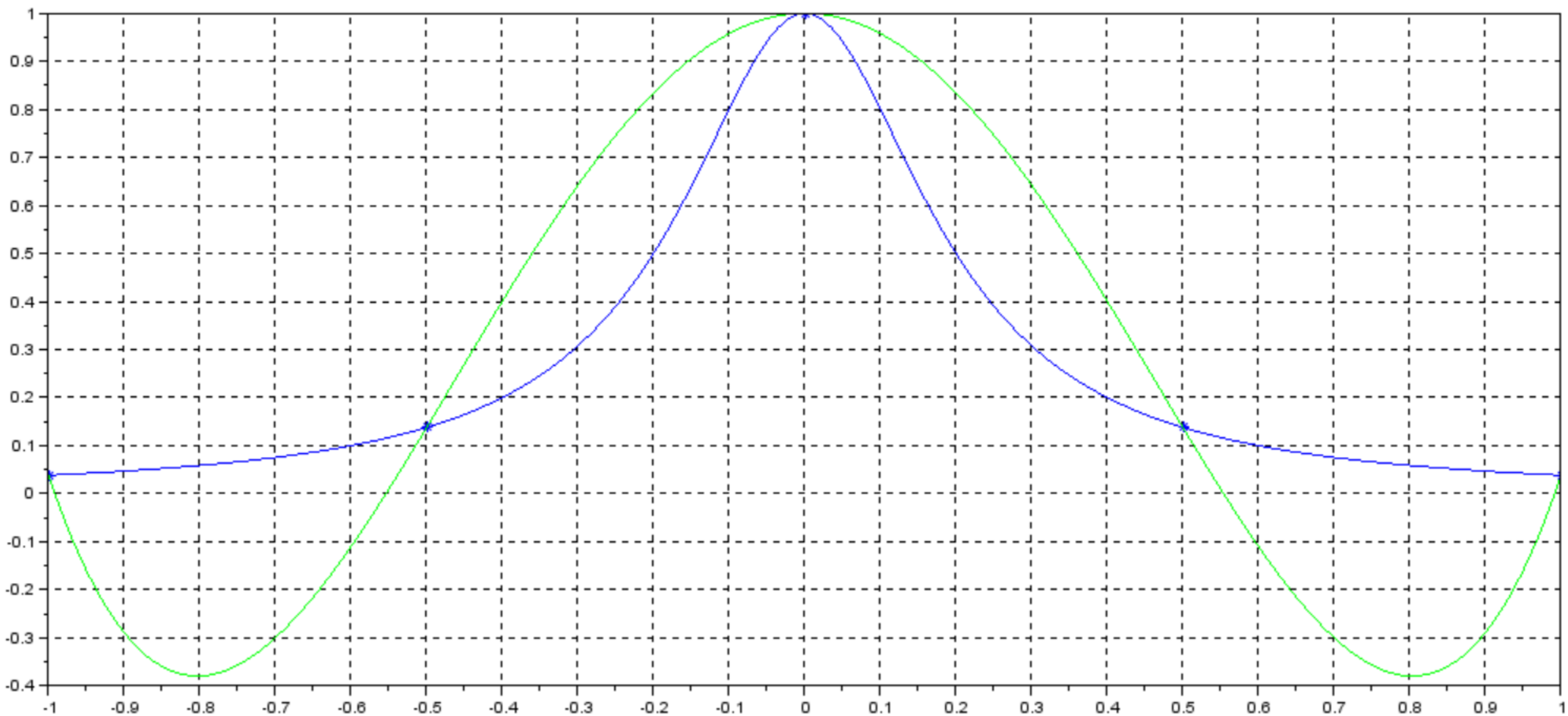
- O script Scilab a seguir ajusta um polinômio de grau 4 a 5 pontos igualmente espaçados da função de Runge e gera uma figura comparando a função de Runge com o polinômio gerado.
- Como mostra a figura, o polinômio não segue satisfatoriamente a função de Runge, sendo que, quanto maior o grau, pior será o ajuste (CHAPRA, 2012, p.424).

Perigos da interpolação com polinômios de grau elevado

```
n = 5;
x = linspace(-1,1,n);
y = 1 ./ (1+25*x.^2)
for i = 1:n
    for j = 1:n
    A(i,j) = x(i)^(j-1);
    end
end
b = y';
p = A\b;
xx = linspace(-1,1,100);
P = p(1) + p(2)*xx + p(3)*xx.^2 + p(4)*xx.^4 +
p(5)*xx.^4;
yy = 1 ./ (1+25*xx.^2)
plot(x,y,'*',xx,yy,'b', xx,P,'g'); xgrid;
```

Perigos da interpolação com polinômios de grau elevado

Comparação da função de Runge (linha azul) com a interpolação polinomial de quarto grau (linha verde)



Perigos da interpolação com polinômios de grau elevado

- O script Scilab a seguir ajusta um polinômio interpolador de Lagrange de grau 10 a 11 pontos igualmente espaçados da função de Runge e gera uma figura comparando a função de Runge com o polinômio gerado.
- O script chama a função Lagrange, apresentada na sequência.
- Como mostra a figura, o polinômio não segue precariamente a função de Runge,

Perigos da interpolação com polinômios de grau elevado

```
exec('path\Lagrange.sci', -1)
n = 1 + input('Entre com o grau do polinômio: ');
x = linspace(-1,1,n);
y = 1 ./ (1+25*x.^2)
xx = linspace(min(x),max(x),200)
yy = 1 ./ (1+25*xx.^2)
yint = Lagrange(x,y,xx)
plot(x,y,'*',xx,yy,'b',xx,yint,'g'); xgrid
```

Perigos da interpolação com polinômios de grau elevado

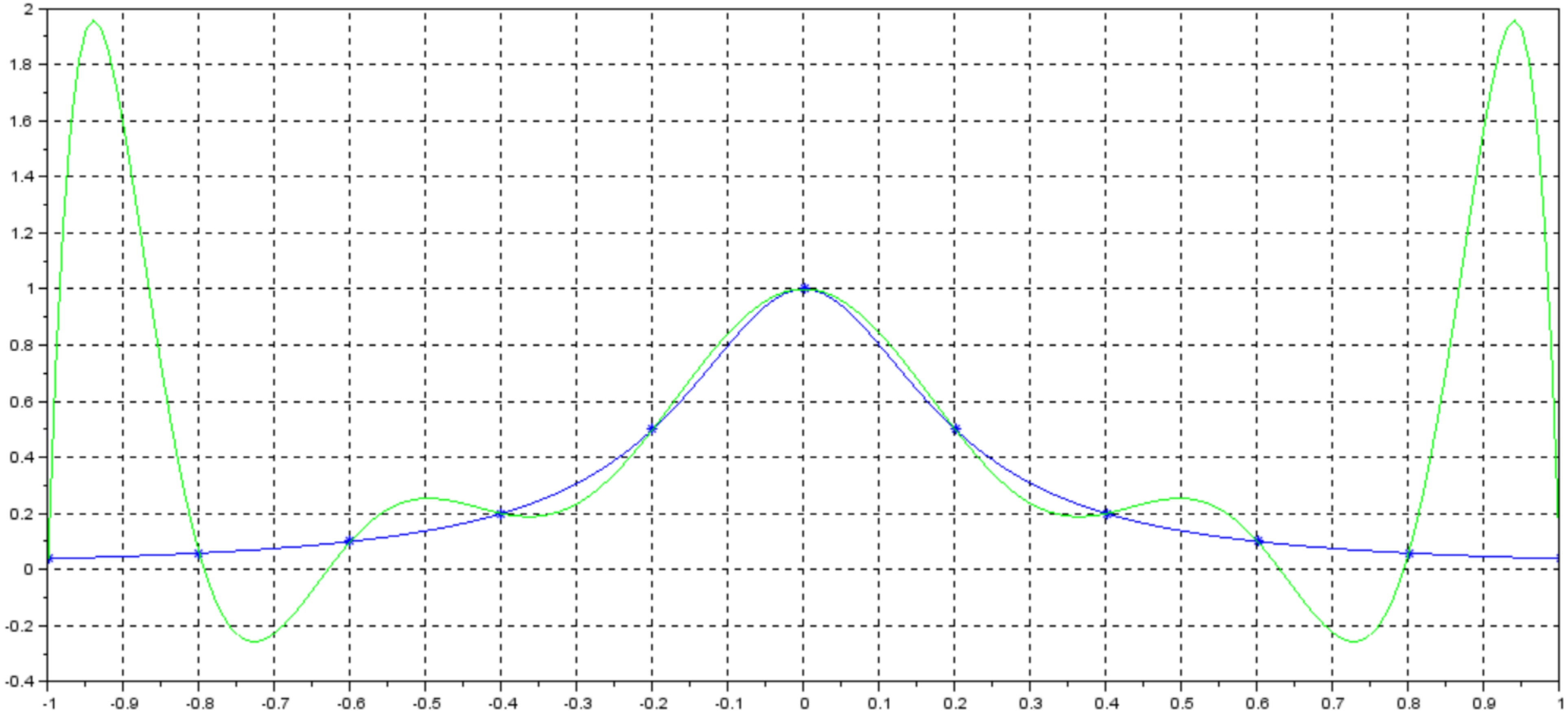
```
function yi = Lagrange(x,y,xi)
// Utiliza um polinômio interpolador de Lagrange
// de grau (n-1) para determinar o valor da
// variável dependente yi em um dado valor xi da
// variável independente
// x é a variável independente
// y é a variável independente
    n = length(x);
    s = 0;
```

Perigos da interpolação com polinômios de grau elevado

```
    for i = 1:n
        produto = y(i);
        for j= 1:n
            if i~=j then
              produto=produto .* (xi-x(j))/(x(i)-x(j));
            end
        end
        s = s + produto;
    end
    yi = s;
endfunction
```

Perigos da interpolação com polinômios de grau elevado

Comparação da função de Runge (linha azul) com a interpolação polinomial de décimo grau (linha verde)



Interpolação por partes no Scilab

- O script a seguir usa funções nativas do Scilab para ajustar 9 pontos dados igualmente espaçados e amostrados dessa função no intervalo [-1,1], empregando um spline cúbico natural.
- A figura gerada mostra que o spline natural segue bem a função de Runge sem exibir oscilações pronunciadas entre os pontos.

Interpolação por partes no Scilab

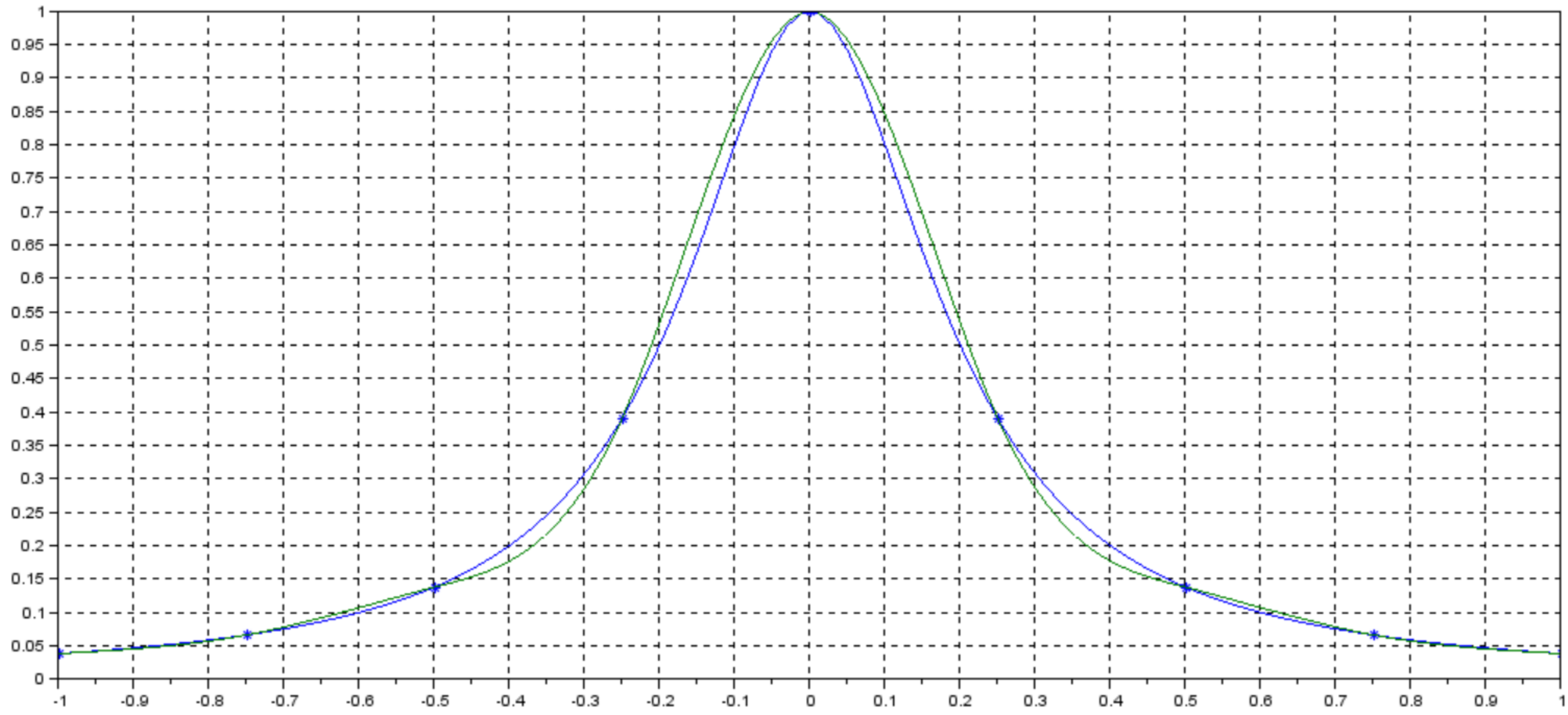
```
// gera os nove pontos
deff("y=runge(x)","y=1 ./(1 + 25*x.^2)")
a = -1; b = 1; n = 9;
x = linspace(a, b, n)';
y = runge(x);
// Esta função computa o spline cúbico s que
// interpola os pontos (xi,yi) i.e., temos os
// s(xi) = yi para todos i = 1,..,n. O spline
// resultante s é completamente definido pela
// tripla (x,y,d) onde d é o vetor com as derivadas
// nos xi: s'(xi)= di
d = splin(x,y,'natural');
// com s''(x1) = s''(xn) = 0
```

Interpolação por partes no Scilab

```
// gera-se em vetor mais finamente espaçado
xx = linspace(a, b, 100);
//  e  100 pontos da função de runge
yx = runge(xx);
// Dados três vetores (x,y,d) definindo uma
função de spline cúbico com yi = s(xi), di =
s'(xi) esta função avalia s em xx
yi =  interp(xx, x, y, d);
// gera os gráficos
// pontos - *, runge - azul, splines - verde
plot(x,y,'*', xx, yx,'b',xx,yi,'g');
```

Interpolação por partes no Scilab

Comparação da função runge (linha azul) com um ajuste de 9 pontos por um *spline natural gerado* com o Scilab (linha verde)



Interpolação por partes no Scilab

- Para gerar o spline sem um nó (padrão) :

```
d = splin(x,y);
```

- Para gerar o spline amarrado:

```
d = splin( x, y, 'clamped', [0 0] );
```

onde o vetor [0 0] contém os valores arbitrados para a primeira derivada no primeiro e no último nós

Exercício

- A partir do conjunto de dados com cinco pontos a seguir

x	8	11	15	18	22
y	5	9	10	8	7

a) Determine splines cúbicos naturais que façam o ajuste dos dados;

b) Determine o valor interpolado de y em x =12,7.

c) Trace um gráfico com os pontos do conjunto de dados e os polinômios interpoladores.

d) Construir um Script Scilab para encontrar um valor interpolado.

Bibliografia e crédito das figuras

- CHAPRA, Steven. **Applied numerical methods with MATHLAB for engineers and scientists.** McGrawHill, 2012.
- CHAPRA, Steven e CANALE, Raymond. **Numerical methods for engineers.** McGrawHill, 2010.