INSTITUTO FEDERAL
CATARINENSE

Objetivos:

- Compreender que splines minimizam oscilações ao ajustar polinômios de menor ordem a partições do domínio;
- Aprender a desenvolver um código para procurar valores em tabelas;
- Compreender o processo de dedução de uma spline.

Introdução

Vimos anteriormente como ajustar polinômios de ordem *(n-1)* para um conjunto de *n* pontos de dados. Por exemplo, podemos ajustar um polinômio de sétima ordem para um conjunto de oito pontos de dados. A curva resultante captaria todas as curvas e nuances sugeridas pelos pontos.

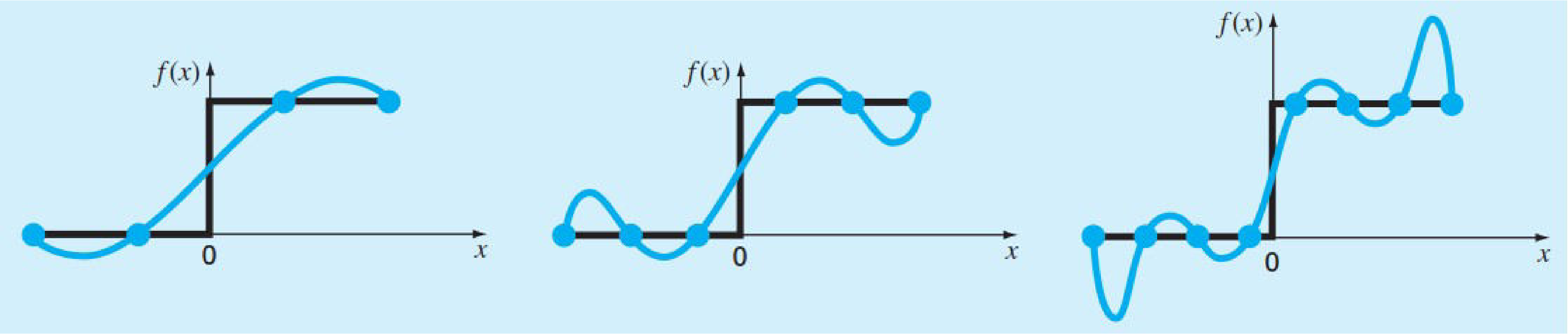
Entretanto, existem casos onde estas funções fornecem resultados ruim devido a erros de arredondamento e oscilações (lembre-se que quanto maior o grau de um polinômio, maior é o seu mau condicionamento).

Assim uma abordagem alternativa pode ser obtida ao se particionar o domínio em pequenos intervalos se aplicar polinômios de baixa ordem a eles. Tais polinômios em série são chamados de **funções spline**.

Prof. MSc. David Roza José
david.jose@luzerna.ifc.edu.br

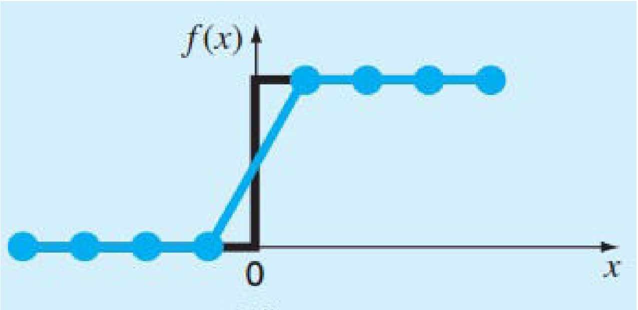
Por exemplo, curvas de terceira ordem utilizadas para conectar conjuntos de dados são chamadas de **splines cúbicas**. Estas funções podem ser construídas de forma que as conexões entre funções adjacentes sejam suaves.

Fica-se com a impressão de que uma spline cúbica é inferior a uma função de interpolação de sétimo grau; e questiona-se até mesmo por qual razão se escolheria uma spline.



A figura ilustra situações unde polinômios de terceira, quinta e sétima ordem demonstram problemas para captar o funcionamento da função.

Em compensação, uma spline linear – criada a partir dos mesmos 8 pontos que deram origem ao polinômio de sétimo grau – demonstra um ajuste muito mais adequado à função além de não apresentar oscilações.



Spline Linear

A notação utilizada para splines é s_i, tal que o índice *i* representa o intervalo da spline. Para *n* conjunto de dados, existem *(n-1)* intervalos. Cada intervalo *i* possui sua própria spline $s_i(x)$. Para splines lineares, cada função é uma linha reta conectando os dois pontos de cada intervalo, cuja formulação é dada por:

$$s_i = a_i + b_i\left(x - x_i\right)$$

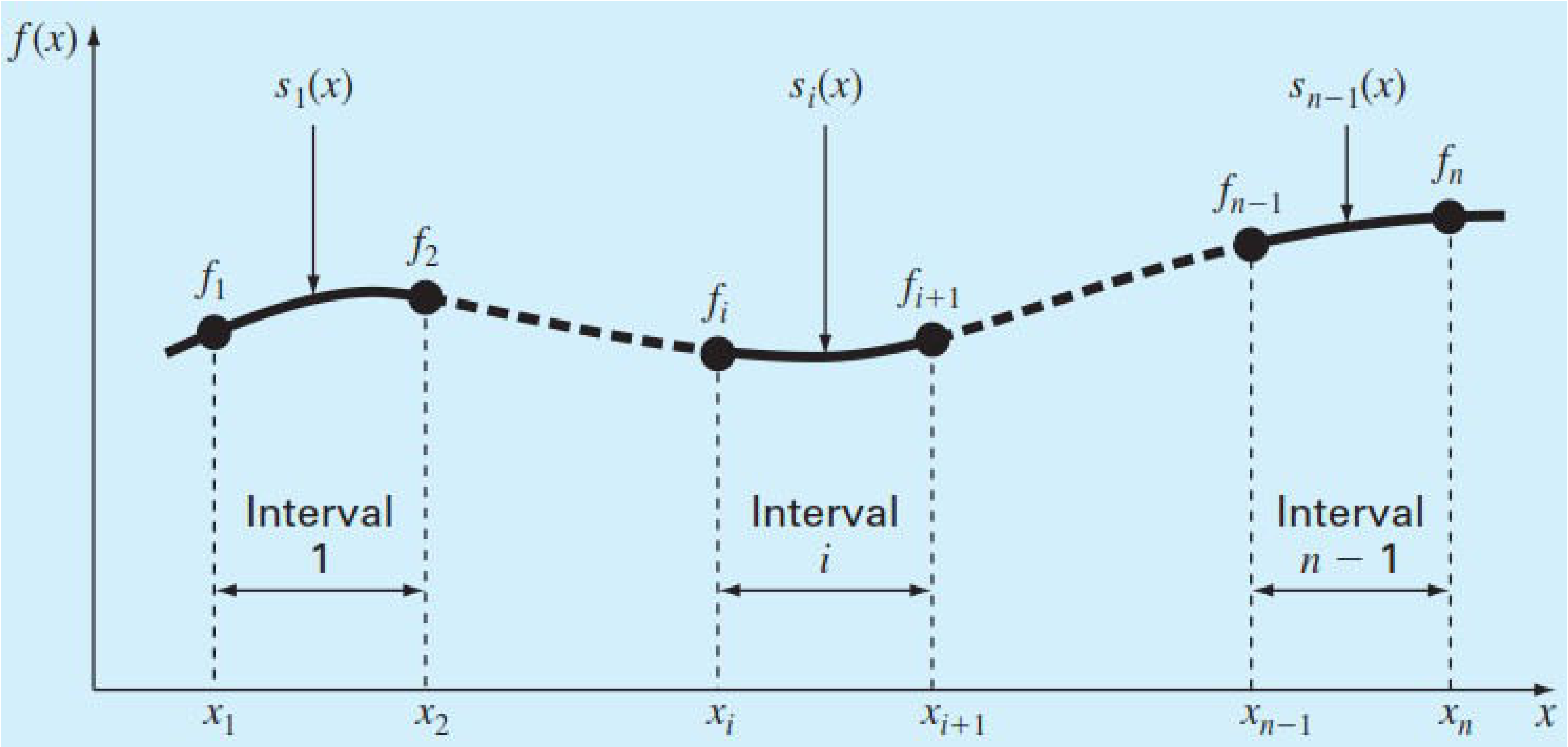
e

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

Tal que f_i é uma notação para $f(x_i)$. Assim, a equação final torna-se:

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

Spline Linear



Exemplo – Spline Linear

Ajustar os dados da tabela com Splines de primeira ordem. Avaliar a função em *x=5*.

i	x_i	f_i
1	3.0	2.5
2	4.5	1.0
3	7.0	2.5
4	9.0	0.5

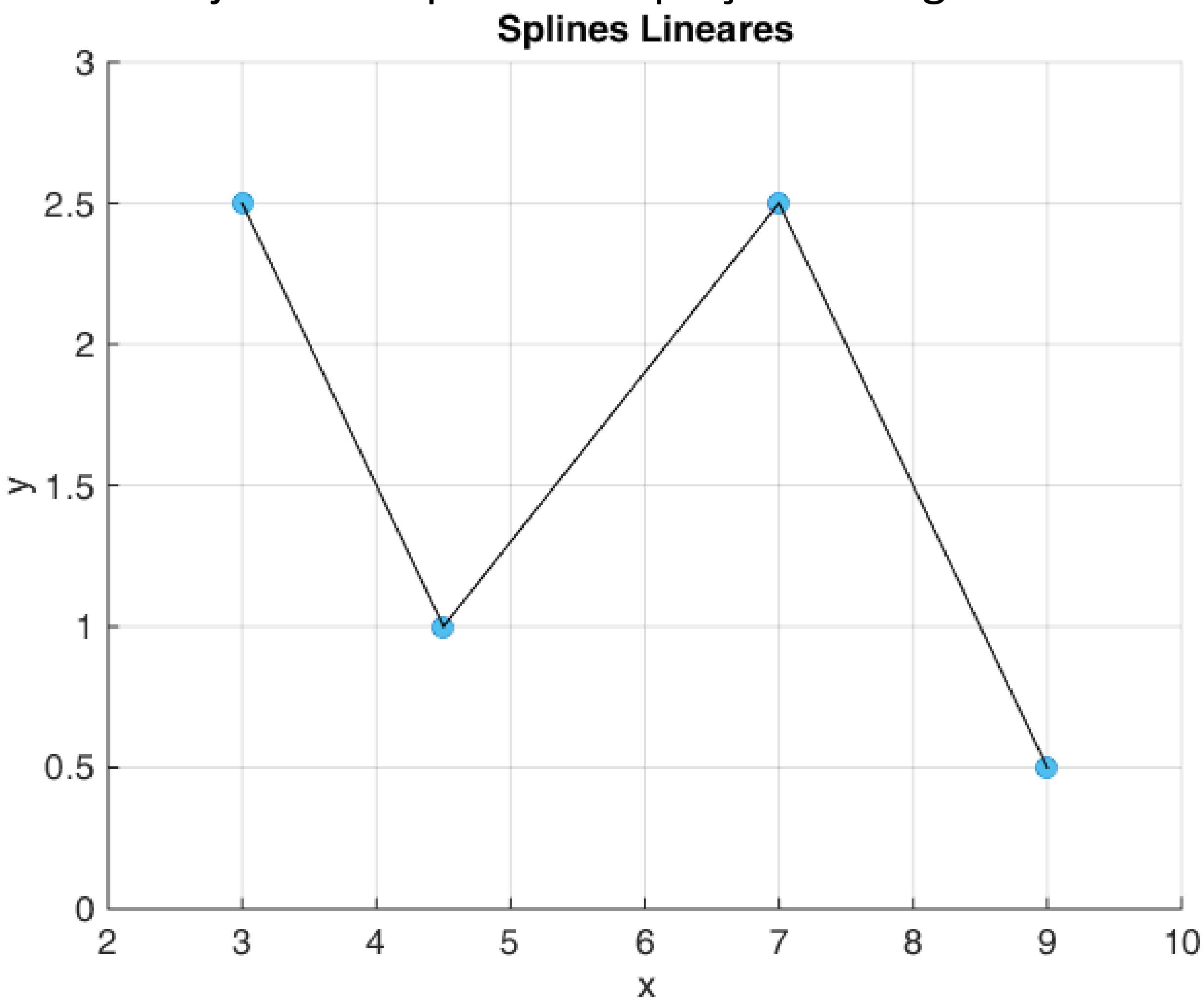
$$s_i\left(x\right) = f_i + \frac{f_{i+1} - f_i}{x_{i+1} - x_i}\left(x - x_i\right)$$

$$s_2\left(x\right) = 1.0 + \frac{2.5 - 1.0}{7.0 - 4.5}\left(x - 4.5\right)$$

$$s_2\left(5\right) = 1.0 + \frac{2.5 - 1.0}{7.0 - 4.5}\left(5 - 4.5\right) = 1.3$$

Exemplo – Spline Linear

O conjunto completo de equações e os gráficos são mostrados abaixo:



$$s_1(x) = 2.5 - (x - 3.0)$$

$$s_2(x) = 1.0 + 0.6\,(x - 4.5)$$

$$s_3(x) = 2.5 - 1\,(x - 7)$$

Prof. MSc. David Roza José
david.jose@luzerna.ifc.edu.br

Spline Linear

A observação do gráfico do exemplo anterior deixa claro que a grande desvantagem de splines lineares é que a curva resultante não é suave. Nos nós – locais de encontro de duas retas distintas – a inclinação da reta muda abruptamente. Ou seja: a derivada primeira é descontínua nestes pontos.

Este problema pode ser resolvido através do uso de splines polinomiais de maior grau, que venham a garantir a suavidade da curva nos nós ao igualar as derivadas primeiras de cada partição do domínio.

Existem situações, porém, onde a interpolação linear é útil e desejada. Em diversas aplicações de engenharia deve-se proceder à interpolação de dados oriundos de tabelas. Uma aplicação que deve ser bastante recorrente ocorre nas disciplina de termodinâmica e transferência de calor, ao se buscar dados de entropia, entalpia, pressão e temperatura nas tabelas de propriedades de substâncias.

Prof. MSc. David Roza José
david.jose@luzerna.ifc.edu.br

Busca em Tabela

Uma das maneiras de se facilitar a busca de dados seria de se construir uma função na qual se coloca o dado desejado e a temperatura na qual buscamos tal dado, e esta função realizaria a interpolação.

Assim, a função realizaria dois atos distintos: buscaria a variável independente para localizar o intervalo que contém a incógnita; e em seguida realizaria a interpolação linear utilizando um dos ajustes lineares já vistos.

Para dados devidamente ordenados, existem duas maneiras de se localizar o intervalo. A primeira é através da **busca sequencial**. Assim como o nome implica, o método envolve comparar o valor desejado com cada elemento do vetor de maneira sequencial até que o intervalo seja encontrado.

Para dados em ordem crescente, compara-se o valor em questão com o do intervalo sendo testado até que o valor em questão seja menor que o sendo testado.

Busca Sequencial

As variáveis independentes são armazenadas em ordem crescente no vetor *x*, e as variáveis dependentes são armazenadas no vetor *y.*

Antes de se iniciar a busca, uma armadilha de erros é inclusa para garantir que o valor desejado *xx* está dentro dos limites de *x*.

A partir daí, um laço *while* compara o valor no qual a interpolação é desejada, *xx*, com o valor superior do intervalo, *x(i+1).* Caso *xx > x(i+1)*, o contador *i* é aumentando de uma unidade para que, na próxima iteração, *xx* seja comparado com o valor superior do próximo intervalo.

O laço é repetido até que *xx* seja menor ou igual ao limite superior do intervalo sendo testado, caso no qual o laço é encerrado. Na sequência, a interpolação é realizada.

Busca Sequencial

Um simples arquivo do MATLAB faz este objetivo:

```
function yi = TableLook(x, y, xx)
n = length(x);
if xx < x(1) | xx > x(n)
    error('Interpolação fora do intervalo')
end
% busca sequencial
i = 1;
while(1)
    if xx <= x(i + 1), break, end
    i = i + 1;
end
% interpolaão linear
yi = y(i) + (y(i+1)-y(i))/(x(i+1)-x(i))*(xx-x(i));
```

Busca Binária

Nas situações para as quais existem muitos dados, a busca sequencial é ineficiente porque ela deve varrer todos os pontos anteriores para encontrar o valor que desejado. Nestes casos, a alternativa é a **busca binária**.

O método da busca binária é parecido com o método da bisseção para localização de raízes. Assim como na bisseção, um índice referente ao ponto médio é calculado – tomando por base o índice inferior e o índice superior. A incógnita é então comparada com o valor do ponto médio para saber se o valor de interesse está no intervalo inferior ou superior.

O processo é então repetido até que a diferença entre os índices superior e inferior seja menor que ou igual a zero. Neste ponto o laço é encerrado e a interpolação linear é efetuada.

Busca Binária

Um simples código que efetua a busca binária:

```
function yi = TableLookBin(x, y, xx)
n = length(x);
if xx < x(1) | xx > x(n)
	error('Interpolação fora do intervalo')
end
% busca binária
iL = 1; iU = n;
while (1)
	if iU - iL <= 1, break, end
	iM = fix((iL + iU) / 2);
	if x(iM) < xx
		iL = iM;
	else
		iU = iM;
	end
end
% linear interpolation
yi = y(iL) + (y(iL+1)-y(iL))/(x(iL+1)-x(iL))*(xx - x(iL));
```

Splines Quadráticas

Para que se possa garantir que uma derivada de n-ésima ordem seja contínua nos nós, uma spline de ordem mínima de *n+1* deve ser utilizada. Splines cúbicas garantem continuidade da derivada primeira e segunda, sendo por esta razão as mais utilizadas. Apesar da derivada terceira e de ordens mais altas acabarem sendo descontínuas ao se utilizar splines cúbicas, tais descontinuidades não podem ser detectadas visualmente – sendo assim ignoradas.

As splines quadradas possuem, portanto, derivada primeira contínua nos nós. Apesar da importância prática das mesmas ser baixa, elas possuem grande utilidade para demonstrar o procedimento geral para desenvolver splines de alta ordem.

Splines Quadráticas

A forma geral de uma spline quadrática é dada por:

$$s_i\left(x\right) = a_i + b_i\left(x - x_i\right) + c_i\left(x - x_i\right)^2$$

de forma que os índices *i* referem-se ao intervalo onde se aplica a equação. Para *n* pontos, existem *(n-1)* intervalos e, consequentemente, *3(n-1)* constantes desconhecidas (os *a's*, *b's* e *c's*) a serem determinadas. Assim, *3(n-1)* equações devem ser deduzidas para se poder determinar o valor das constantes.

Prof. MSc. David Roza José
david.jose@luzerna.ifc.edu.br

Splines Quadráticas

Sua dedução obedece aos seguintes princípios gerais:

1 – A função deve passar por todos os pontos. Esta é a **condição de continuidade**, e pode ser expressada matematicamente como

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

que pode ser simplificado para $a_i = f_i$

Assim, a constante em cada equação quadrática deve ser igual ao valor da variável dependente no começo do intervalo em questão. Podemos, assim, incorporar este resultado à equação geral da spline.

$$s_i\left(x\right) = f_i + b_i\left(x - x_i\right) + c_i\left(x - x_i\right)^2$$

Restam 2(n-1) condições a serem determinadas.

Prof. MSc. David Roza José
david.jose@luzerna.ifc.edu.br

Splines Quadráticas

2 – O valor das splines adjacentes a um mesmo nó devem possuir valores iguais no nó em questão. Esta condição pode ser escrita para o nó *i+1* como:

$$s_i\left(x_{i+1}\right) = s_{i+1}\left(x_{i+1}\right)$$

$$\begin{aligned} f_i + b_i\left(x_{i+1} - x_i\right) + c_i\left(x_{i+1} - x_i\right)^2 = f_{i+1} \\ + b_{i+1}\left(x_{i+1} - x_{i+1}\right) + c_{i+1}\left(x_{i+1} - x_{i+1}\right)^2 \end{aligned}$$

Podemos simplificar a equação ao definir o tamanho de cada intervalo como

$$h_i = x_{i+1} - x_i \qquad\qquad f_i + b_i h + c_i h^2 = f_{i+1}$$

Restam (n-1) condições a serem determinadas.

Splines Quadráticas

3 – A derivada primeira nos nós deve ser igual para funções em nós adjacentes. Esta é uma importante condição, pois significa que splines adjacentes serão unidas de maneira suave; ao invés de observarmos as descontinuidades vistas nas splines lineares. A derivada da forma geral da spline resulta em:

$$s_i^{'}(x) = b_i + 2c_i(x - x_i) = b_i + 2c_i(h_i)$$

De maneira que a condição a ser imposta é:

$$s_i^{'}(x_{i+1}) = s_{i+1}^{'}(x_{i+1})$$

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

Resta somente uma condição a ser determinada.

4 – Podemos impor, então, que a derivada segunda no primeiro ponto (global) é zero. Como a derivada segunda é dada por

$$s_i^{''}(x) = 2c_i$$

Isso implica que $c_1 = 0$. A interpretação visual desta condição é que os dois primeiros pontos serão conectados por uma linha reta.

Prof. MSc. David Roza José
david.jose@luzerna.ifc.edu.br

Exemplo – Splines Quadráticas

Adequar uma spline quadrática para os dados a seguir e estime o valor em *x=5*.

i	x_i	f_i
1	3.0	2.5
2	4.5	1.0
3	7.0	2.5
4	9.0	0.5

Para este problema possuímos quatro pontos de dados e, consequentemente, três intervalos. Aplicando a condição da continuidade (funções adjacentes devem possuir o mesmo valor no mesmo nó) e a condição da derivada segunda do primeiro intervalo ser zero, obtemos o seguinte conjunto de equações:

$$f_1 + b_1 h_1 = f_2$$

$$f_2 + b_2 h_2 + c_2 h_2^2 = f_3$$

$$f_3 + b_3 h_3 + c_3 h_3^2 = f_4$$

A continuidade das derivadas implica que

$$b_1 + 2c_1 h_1 = b_2 \rightarrow b_1 = b_2 \quad \therefore \quad c_1 = 0$$

$$b_2 + 2c_2 h_2 = b_3$$

Prof. MSc. David Roza José
david.jose@luzerna.ifc.edu.br

Exemplo – Splines Quadráticas

As funções necessárias e os valores dos tamanhos dos intervalos são:

$$
\begin{aligned}
f_1 &= 2.5 \qquad & h_1 &= 4.5 - 3.0 = 1.5 \\
f_2 &= 1.0 \qquad & h_2 &= 7.0 - 4.5 = 2.5 \\
f_3 &= 2.5 \qquad & h_3 &= 9.0 - 7.0 = 2.0 \\
f_4 &= 0.5
\end{aligned}
$$

Esses valores podem ser substituídos nas equações obtidas anteriormente, o que gera um sistema de equações que pode ser expressado na forma matricial.

Exemplo – Splines Quadráticas

$$2.5 + 1.5b_1 = 1.0$$
$$1.0 + 2.5b_2 + 6.25c_2 = 2.5$$
$$2.5 + 2.0b_3 + 4.0c_3 = 0.5$$
$$b_2 + 5.0c_2 - b_3 = 0$$

$$\Rightarrow$$

$$1.5b_1 = -1.5$$
$$2.5b_2 + 6.25c_2 = 1.5$$
$$2.5 + 2.0b_3 + 4.0c_3 = -2.0$$
$$b_2 + 5.0c_2 - b_3 = 0$$

$$\begin{bmatrix} 1.5 & 0 & 0 & 0 & 0 \\ 0 & 2.5 & 6.25 & 0 & 0 \\ 0 & 0 & 0 & 2.0 & 4.0 \\ 0 & 1 & 5 & -1 & 0 \\ 1 & -1 & 0 & 0 & 0 \end{bmatrix} \begin{Bmatrix} b_1 \\ b_2 \\ c_2 \\ b_3 \\ c_3 \end{Bmatrix} = \begin{Bmatrix} -1.5 \\ 1.5 \\ -2.0 \\ 0 \\ 0 \end{Bmatrix}$$

Prof. MSc. David Roza José
david.jose@luzerna.ifc.edu.br

Exemplo – Splines Quadráticas

A solução do sistema fornece a seguinte solução:

$$\left\{\begin{array}{c} b_1 \\ b_2 \\ c_2 \\ b_3 \\ c_3 \end{array}\right\} = \left\{\begin{array}{c} -1.0 \\ -1.0 \\ 0.64 \\ 2.2 \\ -1.6 \end{array}\right\}$$

E as equações das splines quadráticas tornam-se:

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

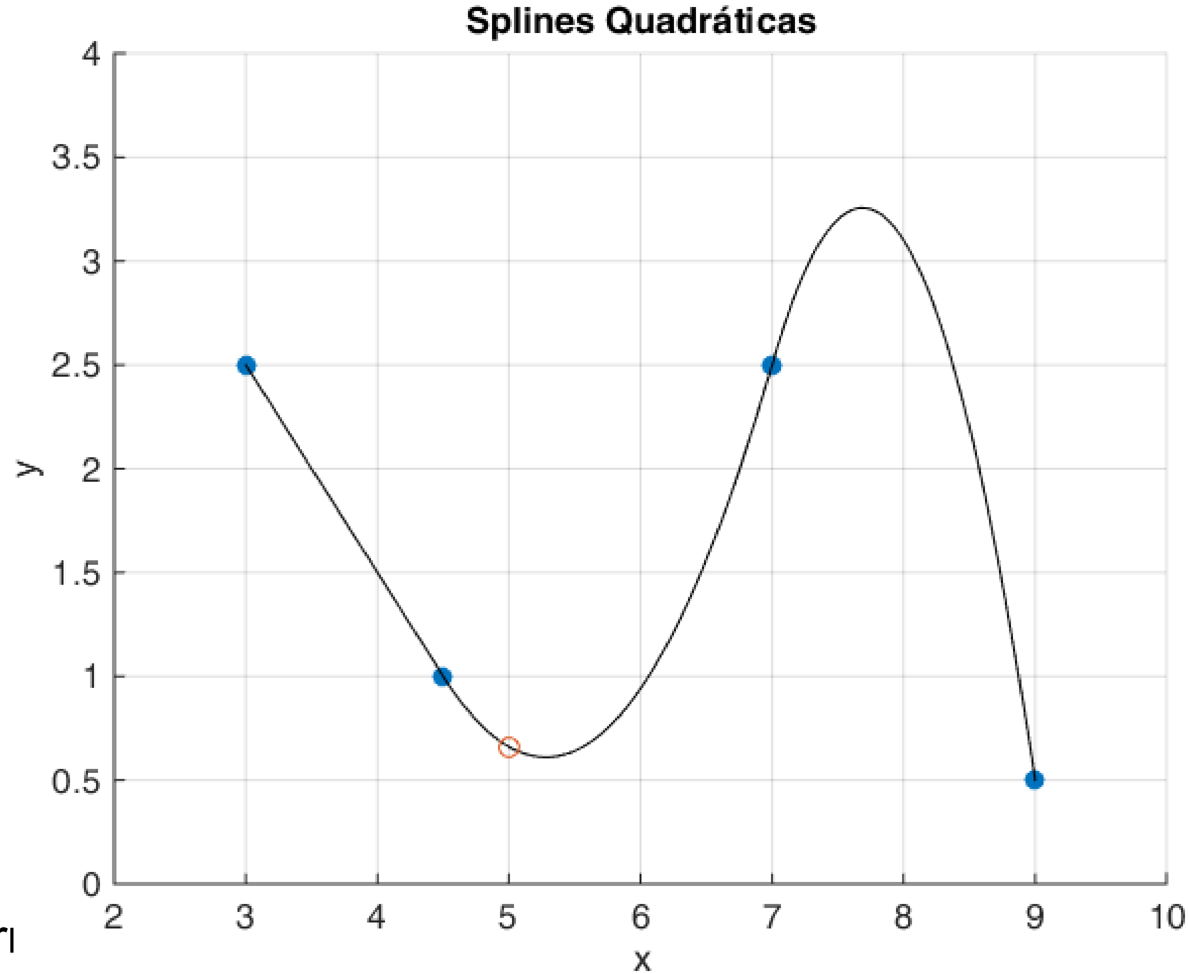
$$s_1(x) = 2.5 - (x - 3)$$
$$s_2(x) = 1.0 - (x - 4.5) + 0.64(x - 4.5)^2$$
$$s_3(x) = 2.5 + 2.2(x - 7.0) - 1.6(x - 7.0)^2$$

Prof. MSc. David Roza José
david.jose@luzerna.ifc.edu.br

Exemplo – Splines Quadráticas

Como *x=5* está no segundo intervalo, utilizamos s_2 para fazer a previsão.

$$s_2(5) = 1.0 - (5 - 4.5) + 0.64(5 - 4.5)^2 = 0.66$$



Prof. MSc. David
david.jose@luzeri

Exemplo – Splines Quadráticas

Perceba que uma das desvantagens é que a primeira spline acaba sendo uma reta. A spline para o último intervalo também "oscila" demais. Nas splines cúbicas, objeto da próxima aula, veremos que estes comportamentos não são apresentados; fornecendo assim, melhores métodos para interpolação.