Capítulo 6

interpolação e Extrapolação

6.1 Introdução

Suponhamos um conjunto de $n+1$ pontos com duas coordenadas x e y, conhecidos por um processo qualquer

$$(x_0, y_0), (x_1, y_1), \ldots, (x_n, y_n)$$

onde

$$x_0 < x_1 < x_2 < \ldots < x_n \,.$$

Os valores y_i podem resultar, por exemplo, de um experimento físico, de um conjunto de observações astronômicas, ou mesmo de uma longa série de cálculos numéricos que não podem ser colocados em uma forma funcional simples. O problema de interpolação consiste em achar, para um determinado valor de x, $x \neq x_0, x_1, \ldots, x_n$ e $x_0 < x < x_n$, um valor razoável para y. Já o problema da extrapolação consiste em estimar y para valores de x fora do intervalo $[x_0, x_n]$.

A interpolação é feita determinando-se uma *função interpolante*, $y = f(x)$, a partir das coordenadas conhecidas. De forma geral, quando se procura determinar a função interpolante existem duas situações distintas:

1. Se o conjunto de coordenadas existente tem uma alta precisão, ou seja, se os valores y_i são bem conhecidos, é razoável exigir que a função interpolante satisfaça: $y_i = f(x_i),\ i = 0, 1, \ldots, n$;

2. Caso contrário esta exigência não é justificável, e podemos ter $y_i \neq f(x_i)$, o que poderá inclusive corrigir valores obtidos imprecisamente.

A interpolação é relacionada a (mas distinta de) outro problema muito comum, que é a *aproximação de funções*. Esta tarefa consiste em encontrar uma função que seja facilmente computável para ser usada no lugar de uma função mais complicada. No caso da interpolação, em geral conhece-se um conjunto de valores y_i para valores da abcissa sobre os quais não se tem controle algum (os valores x_i e y_i são simplesmente dados). No caso de aproximações de funções, pode-se computar a função original para quaisquer valores de x com o propósito de se desenvolver a aproximação.

A interpolação sempre assume algum grau de suavidade da função interpolada, o que frequentemente pode não ser o caso. Por exemplo, considere a função

$$f(x) = 3x^2 + \frac{1}{\pi^4} \ln[(\pi - x)^2] + 1 \,, \tag{6.1}$$

que é muito bem comportada exceto para $x = \pi$. Se fornecemos os valores de $f(x)$ calculados em $x = 3.13$, 3.14, 3.15 e 3.16 para qualquer interpolador, certamente teremos uma resposta muito errada se interpolarmos para $x = 3.1416$, embora um gráfico unindo estes 4 pontos pareça muito suave, como ilustrado na figura abaixo.

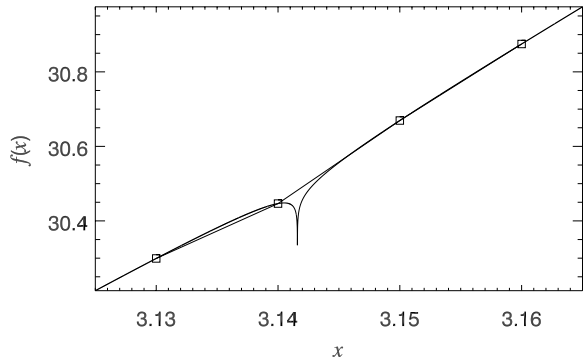

Figura 6.1: Exemplo de função problemática para qualquer interpolador.

Conceitualmente, a interpolação transcorre em dois estágios:

1. ajusta-se (uma vez) uma função interpolante para os pontos fornecidos, e
2. avalia-se (tantas vezes quantas forem necessárias) a função interpolante para os valores de x desejados.

Na prática, este método, conhecido como interpolação global, não é o melhor de se utilizar pois é computaciolmente ineficiente. Em geral inicia-se com o valor tabulado mais próximo ao valor de x desejado e realiza-se uma sequência de correções à medida que outros pontos próximos são considerados (*interpolação local*). O procedimento tipicamente toma $\mathcal{O}(m^2)$ operações, onde $m \ll n$ é o número de pontos efetivamente usados. Se tudo correr bem, a última correção será a menor de todas, e seu valor pode ser usado como uma estimativa informal (mas não rigorosa) do erro. Em esquemas como este, pode-se também dizer que há duas etapas:

1. encontra-se ponto inicial adequado na tabela (digamos, x_i), e
2. faz-se a interpolação usando-se os m pontos vizinhos a i
   (por exemplo, centrados em x_i).

A interpolação local dá valores interpolados que em geral não têm derivadas contínuas. Isso ocorre porque para valores x diferentes usa-se um subconjunto diferente de pontos para a interpolação. Para situações em que a continuidade das derivadas é uma questão importante, deve-se usar a chamada função *spline*, que veremos abaixo. Splines cúbicas são as mais populares, elas garantem que a função interpolada seja contínua até a segunda derivada, e tendem a produzir resultados melhores e mais estáveis que polinômios.

O número m de pontos usado na interpolação menos 1 é chamado de ordem da interpolação. Aumentar a ordem não necessariamente aumenta a acurácia, especialmente no caso de interpolação polinomial (ver figura 6.2).

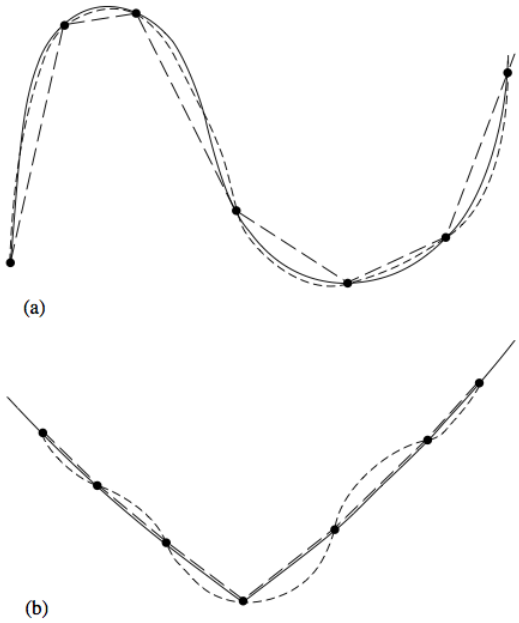

Figura 6.2: (a) Uma função suave (linha sólida) é interpolada de forma mais acurada por um polinômio de ordem mais alta (traços curtos) do que polinômios de ordem baixa (traços longos). (b) Uma função com cantos (derivadas descontínuas) é pior representada por polinômios de alta ordem. [extraído de Numerical Recipes]

Os métodos de interpolação abaixo são também métodos para extrapolação. Neste caso, deve-se tomar muito cuidado em monitorar os erros, caso contrário os resultados podem ser catastróficos. Uma função interpolante, quando usada numa extrapolação, pode dar muitos problemas quando x estiver mais distante dos limites do intervalo $[x_0, x_n]$ do que o espaçamento típico dos valores tabulados.

6.2 Funções interpolantes

São inúmeros os tipos de funções utilizadas para interpolação e devemos ter um certo critério para escolher uma delas, levando em consideração o seu grau de suavidade no intervalo considerado e a sua simplicidade.

De forma geral, uma função interpolante $f(x)$ pode ser escrita como

$$f(x) = a_0 f_0(x) + a_1 f_1(x) + \ldots + a_n f_n(x)$$

onde $f_i(x),\ i = 0, 1, \ldots, n$ representa uma classe de funções. Veremos abaixo exemplos de algumas classes de funções comumente utilizadas para construir funções interpolantes.

1. **Monômios**: $f_i(x) = x^i$

$$f(x) = a_0 + a_1 x + a_2 x^2 + \ldots + a_n x^n = \sum_{i=0}^{n} a_i x^i \,.$$

Claramente, $f(x) = P_n(x)$ é um polinômio de grau n

2. **Funções de Fourier**: $f_i(x) = a_i \cos(ix) + b_i \sin(ix)$

$$\begin{aligned} f(x) &= a_0 + a_1 \cos(x) + a_2 \cos(2x) + \ldots + a_n \cos(nx) + \\ &\quad + b_1 \sin(x) + b_2 \sin(2x) + \ldots + b_n \sin(nx) \\ f(x) &= a_0 + \sum_{i=1}^{n} [a_i \cos(ix) + b_i \sin(ix)] \,. \end{aligned}$$

3. **Exponenciais**: $f_i(x) = e^{b_i x}$

$$f(x) = a_0 e^{b_0 x} + a_1 e^{b_1 x} + \ldots + a_n e^{b_n x} = \sum_{i=0}^{n} a_i e^{b_i x} \,.$$

Existem outras classes de funções, porém são menos usadas. A classe mais usada é a dos monômios (ou polinômios) e citamos as seguintes vantagens:

1. Sua teoria é simples e bem desenvolvida;
2. São fáceis de ser calculados;
3. Somas, produtos e diferenças de polinômios são polinômios;
4. Se $P_n(x)$ é um polinômio, $P_n(x + a)$ e $P_n(ax)$ também o são.
5. Outras classes de funções podem ser aproximadas por polinômios. Para isso utiliza-se o Teorema de Aproximação de Weierstrass:

   "Se $f(x)$ é contínua em $[a, b]$, então para $\forall \epsilon > 0$, existe um $P_n(x)$ de grau n, $n = g(\epsilon)$, tal que $|f(x) - P_n(x)| < \epsilon$, com $a \leq x \leq b$".

6.3 Passo preliminar: buscando em uma tabela ordenada

Vimos acima que uma forma usual de se fazer interpolação é considerar apenas os m pontos vizinhos de x na tabela de dados. Faz-se necessário, portanto, uma rotina que procure o índice i do conjunto de abcissas x_i, $i = 1, \ldots, n$ tal que $x_i < x < x_{i+1}$, para o caso em que o conjunto de abcissas seja monotonicamente crescente ($x_0 < x_1 < x_2 < \ldots < x_n$), ou $x_i > x > x_{i+1}$, para o caso de abcissas monotonicamente decrescentes ($x_n < x_{n-1} < \cdots < x_2 < x_1$).

Dois eficientes métodos para se implementar esta procura são o método da bissecção e o método da caçada. O primeiro consiste em bisseccionar sucessivamente o intervalo $[1, n]$, sempre verificando, em cada passo qual subintervalo contém o valor x procurado (Figura 6.3-a). O outro método, chamado método da caçada, parte do princípio de que na maioria das aplicações faz-se chamadas consecutivas de uma rotina de interpolação para valores da abcissa muito próximos. Dessa forma, pode-se guardar a informação da última posição da tabela e "caçar" o próximo valor, seja para cima ou para baixo, em incrementos de 1, 2, 4 e assim sucessivamente, até que o valor de x deseja esteja confinado em um dado subintervalo, quando então a posição na tabela é encontrada por bisseção (Figura 6.3-b).

Exercício de programação: implemente um subrotina em Fortran, usando o método da bissecção, que encontre a posição de x em uma tabela de abcissas dada. A rotina deve retornar um número entre 1 e $n - 1$ caso $x_1 < x < x_n$, 0 caso $x < x_1$ e n caso $x > x_n$.

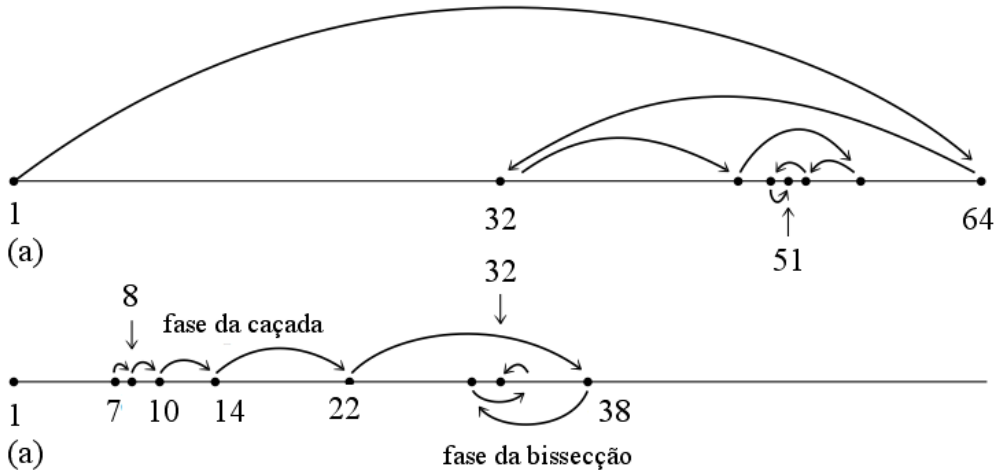

Figura 6.3: (a) Usando a bissecção para encontrar um ponto em uma tabela. São mostrados os passos que convergem para o elemento 51 de uma tabela de 64 pontos. (b) Exemplo do uso do algoritmo da caçada. Mostra-se um caso particularmente desfavorável em que inicia-se no elemento 7 (que fora a último ponto conhecido) para se chegar no ponto 32. [adaptado de Numerical Recipes]

6.4 Interpolação linear

Dado um conjunto de pontos $(x_0, y_0), (x_1, y_1), \ldots, (x_n, y_n)$, a interpolação linear para x, $x_i < x < x_{i+1}$ é

$$y = (1 - f)y_i + f y_{i+1} \tag{6.2}$$

onde

$$f \equiv \frac{x - x_i}{x_{i+1} - x_i}\,.$$

Conceitualmente, a Eq. (6.2) pode ser interpretada como uma média ponderada de y_i e y_{i+1} com peso f, onde f é a distância relativa de x com respeito a x_i.

A interpolação acima é chamada *piecewise*, pois apenas os extremos de um certo subintervalo são usados na interpolação. Como vimos acima, este tipo de função interpolante é contínua mas possui arestas, ou seja, não possui uma derivada primeira contínua nos pontos x_i.

6.4.1 Interpolação Bilinear

A formulação acima para a interpolação linear pode ser facilmente estendida para interpolação linear em duas dimensões (dita bilinear) e dimensões maiores. Vamos explicitar abaixo o problema para duas dimensões, notando que a extensão a outras dimensões é trivial.

Suponhamos uma função desconhecida $f = f(x, y)$, e que temos um conjunto de valores discretos dessa função para vários pontos x e y

$$f_{ij} = f_i j(x_i, y_j)$$

onde $i = 0, \ldots, n$ e $j = 0, \ldots, n$.

queremos estimar o valor de f em um ponto (x, y)

6.4.2 Interpolação $\log \times \log$

DIGITAR...

6.5 Interpolação polinomial

A interpolação polinomial consiste em utilizar-se um polinômio de certo grau como função interpolante. Para um polinômio de ordem n, descrito por $P_n(x) = a_0 + a_1 x + a_2 x^2 + \ldots + a_n x^n$, as incógnitas são os coeficientes $a_0, a_1, a_2, \ldots, a_n$. A interpolação linear, vista acima, é um caso particular da interpolação polinomial para $n = 1$.

A questão que se apresenta é encontrar o melhor polinômio interpolante para cada caso. Como visto na seção 6.1, há duas situações limites, dependendo se a condição $P_n(x_i) = y_i,\ i = 0, 1, \ldots, n$ será imposta ou não. Estas duas situações são descritas abaixo.

6.5.1 Polinômio Interpolante

Um polinômio interpolante pode ser utilizado quando os dados são precisos o suficiente, caso em que podemos impor que $P_n(x_i) = y_i,\ i = 0, 1, \ldots, n$.

Podemos fazer uso de um bem-conhecido teorema da Álgebra:

Teorema: existe um e só um polinômio de grau n ou menor que assume valores específicos para $n + 1$ valores de x.

Este teorema assegura que existe apenas um polinômio interpolante de ordem n ou menor, que passa por um conjunto de $n + 1$ pontos. Entretanto, nada garante que o polinômio interpolante seja uma boa aproximação para $x \neq x_i, i = 0, 1, \ldots, n$.

6.5.2 Polinômio dos Mínimos Quadrados

Quando os valores y_i não são precisos, não se deve exigir que $P_n(x_i) = y_i$. Neste caso, se m é o número de pontos, em geral procura-se um $P_n(x)$ tal que $n \ll m$ (ou seja, a ordem do polinômio interpolante deve ser muito menor que o número de pontos disponíveis).

Para se achar o polinômio interpolante aplica-se o critério dos mínimos quadrados, que estabelece que a soma dos quadrados das diferenças entre y_i e $P_n(x_i)$ deve ser mínima. Isso será visto com maiores detalhes no capítulo 7.

6.5.3 Avaliação de Polinômios

Antes de discutir como determinar o polinômio interpolante, vamos fazer uma breve digressão sobre como avaliar numericamente um polinômio. Seja o polinômio $P_n(x) = a_0 + a_1 x + a_2 x^2 + \ldots + a_n x^n$, onde são conhecidos $a_0, a_1, \ldots, a_n$. Dado um valor de x, vamos calcular $P_n(x)$. Um programa em Fortran seria:

```
REAL :: P
REAL, ALLOCATABLE :: a(:)
INTEGER :: i,n

...
        ALLOCATE(a(n))
```

```
...

        P=0.
        DO i=0,n
                P=P+a(i)*x**i
        ENDDO
```

Desta forma temos $n(n+1)/2$ multiplicações e n adições de ponto flutuante.

Uma forma mais eficiente de implementar o cálculo é usar a regra de Horner, segundo a qual $P_n(x)$ é transformado em

$$P_n(x) = a_0 + x(a_1 + x(a_2 + \ldots + x(a_{n-1} + xa_n)\ldots))$$

Em Fortran:

```
...
        P=a(n)
        DO i=n-1,0,-1
                P=P*x+a(i)
        ENDDO
```

6.6 Polinômio interpolante

Veremos agora como construir um polinômio interpolante de ordem n, que passa por um conjunto de $m = n + 1$ pontos. Veremos inicialmente um algoritmo para calcular os coeficientes a_i do polinômio (seção 6.6.1) e depois um algoritmo mais eficiente usado apenas para avaliar o polinômio em um ponto x, situação mais comumente encontrada (seção 6.6.2).

6.6.1 Sistema de equações

Nesta seção descrevemos uma maneira de se determinar os coeficientes do polinômio interpolante. Um uso válido destes coeficientes poderia ser, por exemplo, avaliar simultaneamente o valor interpolado da função e de algumas das suas derivadas. Um ponto importante a ser considerado é que devido a erros de arredondamento, em geral os coeficientes do polinômio podem ser determinados com precisão muito menor do que o valor do polinômio para uma certa abcissa (seção 6.6.2). Assim, a não ser que os coeficientes sejam realmente necessários, o método abaixo deve ser evitado.

Seja uma tabela de $n+1$ pontos, (x_i, y_i), $i = 0, \ldots, n$. Para determinar os coeficientes do polinômio $P_n(x)$ que passa por estes pontos, montamos o seguinte sistema de equações

$$\begin{cases} a_0 + a_1 x_0 + a_2 x_0^2 + \ldots + a_n x_0^n = y_0 \\ a_0 + a_1 x_1 + a_2 x_1^2 + \ldots + a_n x_1^n = y_1 \\ \vdots \\ a_0 + a_1 x_n + a_2 x_n^2 + \ldots + a_n x_n^n = y_n \end{cases}$$

onde $a_0, a_1, a_2, \ldots, a_n$ são os coeficientes do polinômio, que são as incógnitas deste sistema. Uma maneira de se determinar os coeficientes pode ser resolver o sistema abaixo pelo

método de Gauss

$$\begin{bmatrix} 1 & x_0 & x_0^2 & \dots & x_0^n \\ 1 & x_1 & x_1^2 & \dots & x_1^n \\ 1 & x_2 & x_2^2 & \dots & x_2^n \\ \vdots & & & & \vdots \\ 1 & x_n & x_n^2 & \dots & x_n^n \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ \vdots \\ a_n \end{bmatrix} = \begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} .$$

A matriz acima é conhecida como *matriz de Vandermonde*. A existência de várias potências de x numa mesma linha desta matriz tende a tornar o sistema desbalanceado, ou seja, haverá um grande intervalo dinâmico entre a segunda e a última coluna de cada linha. Por exemplo, se $x_i = 0.1$ e $n = 10 \rightarrow x_i^{10} = 10^{-10}$. Isso pode causar problemas sérios de arredondamento, que comprometerão a acurácia na determinação dos coeficientes. Existem métodos específicos para a resolução deste tipo de sistema, que são descritos na seção 2.8 do Numerical Recipes.

6.6.2 Polinômios de Lagrange

Existem outros métodos de se avaliar o polinômio interpolante de modo mais simples que a resolução do sistema. Nestes métodos, a característica fundamental é que o que se avalia é o polinômio para um dado valor de x, e não os coeficientes propriamente ditos.

Seja uma função tabelada para $n+1$ pontos distintos, $(x_i, y_i),\ i = 0, \dots, n$, e sejam os polinômios de grau n definidos como

$$L_i(x) \equiv \frac{(x - x_0)(x - x_1) \dots (x - x_{i-1})(x - x_{i+1}) \dots (x - x_n)}{(x_i - x_0)(x_i - x_1) \dots (x_i - x_{i-1})(x_i - x_{i+1}) \dots (x_i - x_n)} .$$

Os polinômios $L_i(x)$ são conhecidos como polinômios de Lagrange. Um uma forma mais compacta, pode-se escrevê-los como

$$L_i(x) = \prod_{\substack{j=0 \\ j \neq i}}^{n} \frac{x - x_j}{x_i - x_j} .$$

É fácil ver que para $i \geq 0$ e $j \leq n$,

$$L_i(x_j) = \begin{cases} 1 & \text{se } i = j \\ 0 & \text{se } i \neq j \end{cases} . \tag{6.3}$$

Desta forma, podemos determinar o polinômio interpolador de y relativamente aos pontos $x_i, i = 1, \dots, n$, utilizando os polinômios de Lagrange, da seguinte maneira

$$P_n(x) = L_0(x)y_0 + L_1(x)y_1 + ... + L_n(x)y_n = \sum_{i=0}^{n} L_i(x) f(x_i) . \tag{6.4}$$

Como os polinômios de Lagrange satisfazem as condições (6.3), é evidente que $P_n(x_i) = y_i, i = 1, \dots, n$. Além disso, o grau de P_n é menor ou igual a n. Segue-se, portanto, que o polinômio P_n é o único polinômio interpolador que passa pelos $n + 1$ pontos.

Exemplo: Calcule o polinômio interpolante para os pontos

$$(-1, -11); (1, 3); (2, 7); (3, 17)$$

$$(x_0, y_0); (x_1, y_1); (x_2, y_2); (x_3, y_3)$$

$$
\begin{aligned}
P_3(x) &= \frac{(x-1)(x-2)(x-3)}{(-1-1)(-1-2)(-1-3)}(-11) \\
&+ \frac{(x+1)\quad(x-2)(x-3)}{(1+1)\quad(1-2)(1-3)}(3) \\
&+ \frac{(x+1)(x-1)\quad(x-3)}{(2+1)(2-1)\quad(2-3)}(7) \\
&+ \frac{(x+1)(x-1)(x-2)}{(3+1)(3-1)(3-2)}(17) \\
P_3(x) &= \frac{11}{24}(x^3 - 6x^2 + 11x - 6) + \frac{18}{24}(x^3 - 4x^2 + x + 6) \\
&- \frac{56}{24}(x^3 - 3x^2 - x + 3) + \frac{51}{24}(x^3 - 2x^2 - x + 2) \\
P_3(x) &= x^3 - 3x^2 + 6x - 1\,.
\end{aligned}
$$

Algumas observações importantes:

- O polinômio é o mesmo que aquele calculado por um sistema de equações, pois ele é único;
- Em geral as fórmulas de Lagrange não são usadas para determinar os coeficientes do polinômio interpolante, devido à complexidade dos cálculos e do algoritmo necessário (ver seção 6.6.3) ;
- Estas fórmulas são usadas diretamente para interpolar $y = P_n(x)$;
- Com dois "loops", um para o cálculo de $\prod$ e outro para o cálculo de $\sum$, obtém-se o valor de $y = P_n(x)$ interpolado.

6.6.3 Um método alternativo

Digitar. Numerical Recipes, 3a. edição, seção 3.5.

6.6.4 Falhas do polinômio interpolante

O método de interpolação por polinômios não funciona indiscriminadamente para todas as funções, e em todo o intervalo analisado. Ele pode ser particularmente problemático quando a tabela interpolada tem os pontos da abcissa igualmente espaçados.

Um exemplo clássico dos problemas que podem ocorrer com a interpolação polinomial é o fenômeno de Runge (*Runge's Phenomenon*). Consideremos como função geradora dos pontos base a função de Runge:

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

A Fig. 6.4 mostra o gráfico da função de Runge no intervalo $[-1, 1]$ e as curvas de pois polinômios interpolantes ($P_5(x)$ e $P_{20}(x)$), construídos a partir de pontos igualmente espaçados no intervalo

$$x_i = -1 + (i-1)\frac{2}{n}, i = 1, \ldots, n\,.$$

Vemos que a divergência é tão maior quanto maior for o grau do polinômio. Pode-se, inclusive, mostrar que o erro na interpolação tende a infinito quando o grau do polinômio aumenta.

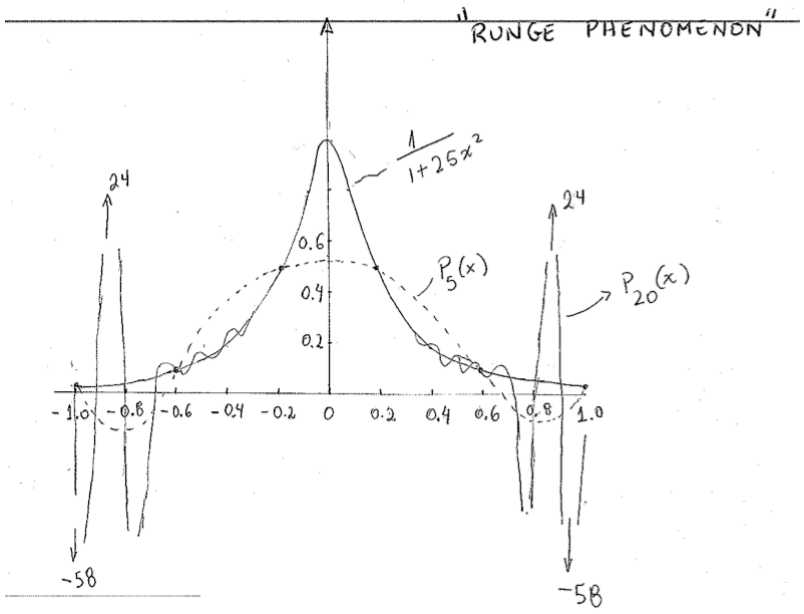

Figura 6.4: O fenômento de Runge.

6.7 Erro do Polinômio Interpolante

DIGITAR

6.8 Interpolação por Spline Cúbica

Em matemática, uma *spline* é uma função polinomial suave que é definida por partes (*piecewise*) e possui uma alta suavidade nos locais onde as diferentes partes se conectam (conhecidas como nós ou pontos-base). O termo spline vem do técnica empregada por desenhistas para traçarem curvas suaves: imagine uma régua flexível e indeformável, que é presa na superfície de desenho em alguns pontos: esta régua assumira a forma de uma spline.

Splines cúbicas são frequentemente empregadas em interpolação com a finalidade de garantir a suavidade da função interpolante. Trata-se de uma solução muito eficiente para o fenômeno de Runge visto anteriormente. Matematicamente, o método consiste na definição de um polinômio de 3º grau para cada par de pontos consecutivos. Ou seja, dados os pontos $y_i = y(x_i)$, $i = 0, \ldots, n$, define-se um polinômio cúbico para o intervalo $[x_0, x_1]$, $p_1(x)$, outro para o intervalo $[x_1, x_2]$, $p_2(x)$, e assim por diante. Para $n+1$ pontos, temos assim n polinômios. Obviamente, os polinômios devem ser tais que a função seja contínua nos nós. Além disso, para que a variação do raio de curvatura seja contínua nos nós, em cada um a 2ª derivada do polinômio anterior deve ser igual à do polinômio posterior. O mesmo se dá com a 1ª derivada. As condições explicitadas acima traduzem-se em $4n - 2$ equações, listadas na Tabela 6.1.

Equações	Condições
n	$p_i(x_i) = y_i$
n	$p_i(x_{i+1}) = y_{i+1}$
$n-1$	$p_i''(x_i) = p_{i-1}''(x_i)$
$n-1$	$p_i'(x_i) = p_{i-1}'(x_i)$
$4n-2$	

Tabela 6.1: condições necessárias para se determinar a função spline que passa por $n+1$ pontos.

6.8.1 Dedução das fórmulas de spline

O procedimento básico consiste em determinar os 4 coeficientes de cada um dos n polinômios de 3º grau, ou seja, $4n$ incógnitas precisam ser determinadas. Como vimos, os valores dos polinômios e suas derivadas primeira e segunda nos $n+1$ pontos fornecem apenas $4n-2$ equações, pois não podemos calcular $p_1'(x_0)$ e $p_1''(x_0)$. Para contornar esse problema, assumiremos que tanto no primeiro quanto no último segmento da função spline a inclinação é constante (ou seja, nos extremos do intervalo a função converte-se em uma reta). Logo, para estes segmentos a 1ª derivada é uma constante e a 2ª derivada é nula. Dessa forma, ao impormos que $p_1''(x_0) = 0$ e $p_n''(x_n) = 0$, ficamos com as $4n$ equações necessárias para tornar o sistema determinado. Uma função spline assim construída chama-se função spline natural.

Para facilitar a notação, vamos definir

$$h_i \equiv x_{i+1} - x_i \,,$$

$$\phi_i \equiv p_{i-1}''(x_i) = p_i''(x_i) \,.$$

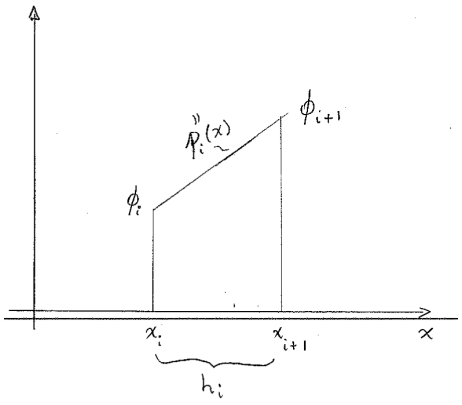

Figura 6.5: Representação gráfica da segunda derivada de p_i.

Como os p_i são polinômios de 3º grau, temos que cada $p_i''(x)$ é um segmento de reta (Fig. 6.5). Dessa forma, podemos escrever, usando a Eq. (6.2)

$$p_i''(x) = \frac{(x_{i+1} - x)}{h_i}\phi_i + \frac{(x - x_i)}{h_i}\phi_{i+1} \,.$$

Integrando $p_i''(x)$ duas vezes, obtemos

$$p_i'(x) = -\frac{1}{2}\frac{(x_{i+1}-x)^2}{h_i}\phi_i + \frac{1}{2}\frac{(x-x_i)^2}{h_i}\phi_{i+1} + C_1,$$

e

$$p_i(x) = \left(-\frac{1}{2}\right)\left(-\frac{1}{3}\right)\frac{(x_{i+1}-x)^3}{h_i}\phi_i + \frac{1}{2}\frac{1}{3}\frac{(x-x_i)^3}{h_i}\phi_{i+1} + C_1 x + C_2\,.$$

Vamos agora fazer uma mudança de variáveis, de forma a reescrever o último termo da equação acima como

$$C_1 x + C_2 = C_1'(x_{i+1}-x) + C_2'(x-x_i)\,.$$

Dessa forma, obtemos

$$p_i(x) = \frac{1}{6}\frac{(x_{i+1}-x)^3}{h_i}\phi_i + \frac{1}{6}\frac{(x-x_i)^3}{h_i}\phi_{i+1} + C_1'(x_{i+1}-x) + C_2'(x-x_i)\,.$$

Para obtermos as constantes de integração C_1' e C_2', impomos as condições $p_i(x_i) = y_i$ e $p_i(x_{i+1}) = y_{i+1}$ (tabela 1)

$$y_i = \frac{1}{6}\frac{(x_{i+1}-x_i)^3}{h_i}\phi_i + C_1'(x_{i+1}-x_i) \to C_1' = \frac{y_i}{h_i} - \frac{h_i\phi_i}{6}$$

$$y_{i+1} = \frac{1}{6}\frac{(x_{i+1}-x_i)^3}{h_i}\phi_{i+1} + C_2'(x_{i+1}-x_i) \to C_2' = \frac{y_{i+1}}{h_i} - \frac{h_i\phi_{i+1}}{6}$$

. Finalmente, chegamos à seguinte expressão para $p_i(x)$, que representa cada uma das partes da função spline

$$p_i(x) = \frac{\phi_i}{6h_i}(x_{i+1}-x)^3 + \frac{\phi_{i+1}}{6h_i}(x-x_i)^3 + \left(\frac{y_i}{h_i} - \frac{h_i\phi_i}{6}\right)(x_{i+1}-x) + \left(\frac{y_{i+1}}{h_i} - \frac{h_i\phi_{i+1}}{6}\right)(x-x_i)\,, \tag{6.5}$$

onde os $\phi_1, \ldots, \phi_n$ são incógnitas.

Para obtermos os ϕ_i, devemos impor as condições da tabela 6.1 que ainda não foram utilizadas, a saber, as condições sobre a primeira derivada de $p_i(x)$. Imporemos a condição que a derivada primeira é contínua nos nós das função spline, ou seja,

$$p_i'(x_i) = p_{i-1}'(x_i)\,. \tag{6.6}$$

Fazendo a derivada da Eq. (6.5) e depois de alguma manipulação algébrica trivial, obtemos

$$p_i'(x_i) = -\frac{1}{2}h_i\phi_i - \left(\frac{y_i}{h_i} - \frac{h_i\phi_i}{6}\right) + \frac{y_{i+1}}{h_i} - \frac{h_i\phi_{i+1}}{6}\,,$$

e

$$p_{i-1}'(x_i) = \frac{1}{2}h_{i-1}\phi_i - \left(\frac{y_{i-1}}{h_{i-1}} - \frac{h_{i-1}\phi_{i-1}}{6}\right) + \left(\frac{y_i}{h_{i-1}} - \frac{h_{i-1}\phi_i}{6}\right)\,.$$

Aplicando a condição (6.6), obtemos, depois de alguma manipulação algébrica

$$h_{i-1}\phi_{i-1} + 2(h_{i-1}+h_i)\phi_i + h_i\phi_{i+1} = 6\left(\frac{y_{i+1}-y_i}{h_i} - \frac{y_i - y_{i-1}}{h_{i-1}}\right) \tag{6.7}$$

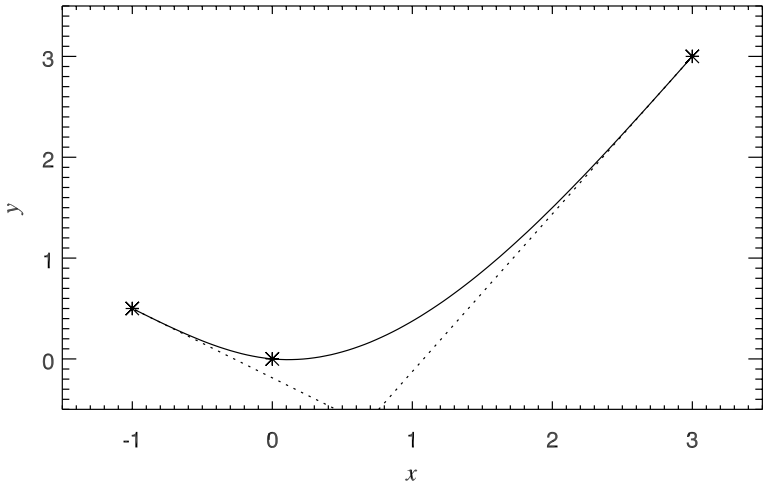

Figura 6.6: Ajuste de spline cúbica que passa pelos pontos $(-1, 1/2)$, $(0, 0)$ e $(3, 3)$. As linhas tracejadas mostram as retas para as quais a função converge nos extremos no intervalo

As Eqs. (6.7), definidas para $i = 1, \ldots, n-1$, representam um sistema de $n-1$ equações. O número de incógnitas é $n+1$, mas duas já estão determinadas por termos imposto que a segunda derivada da primeira e última parte da função é nula (ou seja $\phi_0 = 0$ e $\phi_n = 0$). Antes de mostrar um método geral de obter os coeficientes ϕ_i, vamos ver um exemplo simples.

Exemplo: vamos calcular a função spline cúbica que passa pelos pontos $(-1, 1/2)$, $(0, 0)$ e $(3, 3)$.

Neste caso, temos que determinar $p_1(x)$ e $p_2(x)$. Das definições acima, temos que $h_0 = 1$, $h_1 = 3$, $\phi_0 = 0$ e $\phi_2 = 0$. Falta-nos apenas determinar ϕ_1. Da Eq. (6.7) temos

$$0 + 8\phi_1 + 0 = 6(1 + 0.5)\,,$$

ou seja, $\phi_1 = 9/8$. Da Eq. (6.6) temos que

$$p_1(x) = 0 + \frac{9/8}{6}(x - (-1))^3 + (\frac{1/2}{1} - 0)(0 - x) + (0 - \frac{9/8}{6})(x + 1)\,,$$

ou

$$p_1(x) = \frac{3}{16}(x + 1)^3 + \frac{11}{16}x - \frac{3}{16}\,.$$

e

$$p_2(x) = \frac{9/8}{6 \times 3}(3 - x)^3 + 0 + \left(0 - \frac{3 \times 9/8}{6}\right)(3 - x) + \left(\frac{3}{3} - 0\right)(x - 0)\,,$$

ou

$$p_2(x) = \frac{1}{16}(3 - x)^3 + \frac{25}{16}x - \frac{27}{16}\,.$$

A Fig. 6.6 mostra o função spline determinada sobreposta aos três pontos dados. Note que nos extremos do intervalo a função se transforma em uma linha reta, como esperado.

Exercício: mostrar que as derivadas primeira e segunda da função spline determinada acima é contínua em $(0, 0)$.

As Eqs. (6.7) podem ser escritas em forma matricial da seguinte maneira, formando um sistema $(n-1) \times (n-1)$

$$\begin{bmatrix} 2(h_0+h_1) & h_1 & & & \\ h_1 & 2(h_1+h_2) & h_2 & & \\ & \cdots & \cdots & \cdots & \\ & h_{n-3} & 2(h_{n-3}-h_{n-2}) & h_{n-2} & \\ & & h_{n-2} & 2(h_{n-2}+h_{n-1}) & \end{bmatrix} \begin{bmatrix} \phi_1 \\ \phi_2 \\ \vdots \\ \phi_{n-2} \\ \phi_{n-1} \end{bmatrix} = \begin{bmatrix} e_1 - e_0 \\ e_2 - e_1 \\ \vdots \\ e_{n-2} - e_{n-3} \\ e_{n-1} - e_{n-2} \end{bmatrix}$$

onde definimos

$$e_i = 6\left(\frac{y_{i+1} - y_i}{h_i}\right),$$

para simplificar a notação. Este sistema tridiagonal pode ser resolvido através de uma implementação direta usando a decomposição LU, vista no Cap. 5.

Inicialmente, efetuamos a decomposição LU da matriz tridiagonal acima, escrevendo

$$\begin{cases} u_1 = 2(h_0 + h_1) \\ l_j = \frac{h_{j-1}}{u_{j-1}},\ j = 2, \ldots, n-1 \\ u_j = 2(h_{j-1} + h_j) - \frac{h_{j-1}^2}{u_{j-1}},\ j = 2, \ldots, n-1 \end{cases}$$

Uma vez feita a decomposição LU, o sistema é resolvido em dois passos, como mostrado na Seção 5.10.2. Inicialmente calculamos o vetor y por substituição para frente

$$\begin{cases} y_1 = e_1 - e_0 \\ y_j = (e_j - e_{j-1}) - l_j y_{j-1},\ j = 2, \ldots, n-1 \end{cases}$$

e finalmente calcula-se os ϕ's da seguinte forma

$$\begin{cases} \phi_{n-1} = \frac{y_{n-1}}{u_{n-1}} \\ \phi_j = \frac{y_j - h_j \phi_{j+1}}{u_j},\ j = n-2, \ldots, 1 \end{cases} . \tag{6.8}$$

A interpolação por spline pode ser facilmente implementada computacionalmente. Sugere-se que sejam feitas duas subrotinas, a primeira lê os $n+1$ valores dos pontos tabelados e retorna um array com os h's e outro com os ϕ's. Uma outra subrotina usa estes dois arrays e a Eq. (6.5) para avaliar a função no intervalo i desejado.