Capítulo 3

Interpolação polinomial

A necessidade de obter um valor intermediário que não consta de uma tabela ocorre comumente. Dados experimentais, tabelas estatísticas e de funções complexas são exemplos desta situação. Neste capítulo, serão apresentados alguns métodos numéricos para resolver este problema de grande utilidade prática.

O conceito de interpolação não é importante somente na obtenção de valores intermediários em tabelas. Ele é fundamental em outros tópicos abordados em **Algoritmos Numéricos**, como, por exemplo, integração numérica (Capítulo 5), cálculo de raízes de equações (Capítulo 6) e solução de equações diferenciais ordinárias (Capítulo 7).

3.1 Polinômios interpoladores

Seja a Tabela 3.1 a seguir

Tabela 3.1 Dados para interpolação.

x	0,1	0,6	0,8
y	1,221	3,320	4,953

O problema consiste em encontrar o valor correspondente de y para um dado x não pertencente à tabela. Um modo de resolver este problema é obter uma função que relaciona as variáveis x e y. Considerando que os polinômios são as funções mais simples e estudadas, então eles são os mais utilizados para determinar esta relação. Um polinômio construído com o intuito de aproximar uma função é denominado polinômio interpolador. Deste modo, para resolver o problema basta avaliar o polinômio obtido no ponto desejado.

Existem vários métodos para construir um polinômio interpolador a partir de um conjunto de pares (x, y). O esquema mais simples, em termos conceituais, envolve a solução de um sistema de equações lineares.

3.1.1 Interpolação linear

Sejam dois pontos-base (x_0, y_0) e (x_1, y_1), com $x_0 \neq x_1$, de uma função $y = f(x)$. Para obter uma aproximação de $f(z)$, $z \in (x_0, x_1)$, faz-se

$$f(x) \approx P_1(x) = a_0 + a_1 x,$$

onde $P_1(x)$ é um polinômio interpolador de grau 1. Impondo que o polinômio interpolador passe pelos dois pontos-base, tem-se o seguinte sistema de equações lineares de ordem 2

$$\begin{matrix} P_1(x_0) = y_0 \\ P_1(x_1) = y_1 \end{matrix} \longrightarrow \left\{ \begin{matrix} a_0 + a_1 x_0 = y_0 \\ a_0 + a_1 x_1 = y_1 \end{matrix} \right. \iff \begin{bmatrix} 1 & x_0 \\ 1 & x_1 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \end{bmatrix} = \begin{bmatrix} y_0 \\ y_1 \end{bmatrix}.$$

Aplicando uma operação l-elementar para transformar este sistema linear em um sistema triangular equivalente, obtém-se

$$\begin{bmatrix} 1 & x_0 \\ 0 & x_1 - x_0 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \end{bmatrix} = \begin{bmatrix} y_0 \\ y_1 - y_0 \end{bmatrix},$$

cuja solução é

$$a_1 = \frac{y_1 - y_0}{x_1 - x_0} \quad \text{e} \quad a_0 = y_0 - a_1 x_0.$$

Portanto, o polinômio interpolador torna-se

$$P_1(x) = a_0 + a_1 x = (y_0 - a_1 x_0) + a_1 x = y_0 + a_1(x - x_0) \text{ e}$$

$$\boxed{P_1(x) = y_0 + \frac{y_1 - y_0}{x_1 - x_0}(x - x_0)}. \tag{3.1}$$

Como o determinante da matriz do sistema linear acima é igual a $x_1 - x_0 \neq 0$, então o sistema admite uma única solução, ou seja, por dois pontos passa um único polinômio de grau 1. Deve ser verificado que, por (3.1), $P_1(x_0) = y_0$ e $P_1(x_1) = y_1$, conforme imposto ao se construir o sistema linear. Isso mostra que, de fato, o polinômio interpolador passa pelos dois pontos-base.

Exemplo 3.1 Calcular $P_1(0{,}2)$ e $P_1(0{,}3)$ a partir da tabela

i	0	1
x_i	0,1	0,6
y_i	1,221	3,320

Pelo uso de (3.1),

$$P_1(0{,}2) = 1{,}221 + \frac{3{,}320 - 1{,}221}{0{,}6 - 0{,}1}(0{,}2 - 0{,}1) \rightsquigarrow P_1(0{,}2) = 1{,}641 \text{ e}$$

$$P_1(0{,}3) = 1{,}221 + \frac{3{,}320 - 1{,}221}{0{,}6 - 0{,}1}(0{,}3 - 0{,}1) \rightsquigarrow P_1(0{,}3) = 2{,}061.$$

Sabendo que a função neste exemplo é $f(x) = e^{2x}$, então, os erros cometidos foram

em $x = 0{,}2$ tem-se $1{,}641 - e^{2\times 0{,}2} = 0{,}149$ e

em $x = 0{,}3$ tem-se $2{,}061 - e^{2\times 0{,}3} = 0{,}239$. ∎

Quanto mais próximo o valor a ser interpolado for de um ponto-base, melhor será o resultado obtido pela interpolação. O resultado da interpolação pode ser melhorado pelo aumento do grau do polinômio interpolador.

A Figura 3.1 mostra os dados da Tabela 3.1 representados por pontos o e o polinômio interpolador de grau 1 esboçado por uma linha tracejada --. São apresentados também o polinômio interpolador de grau 2, desenhado com uma linha traço-ponto -., e a função $f(x) = e^{2x}$, representada por uma linha sólida —.

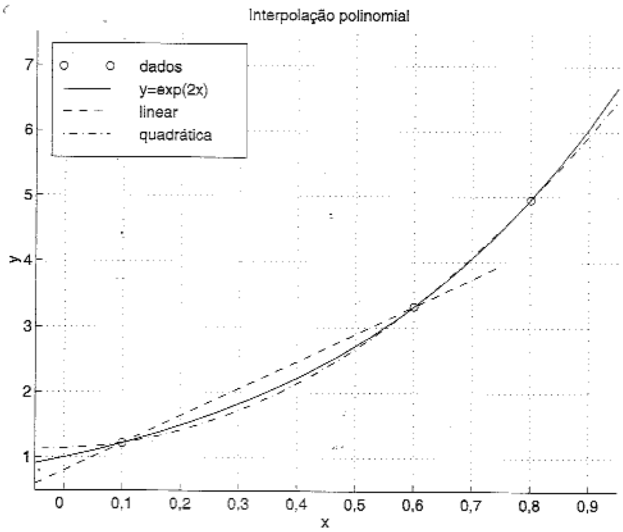


Figura 3.1 Interpretação geométrica da interpolação polinomial.

3.1.2 Interpolação quadrática

Sejam três pontos-base (x_0, y_0), (x_1, y_1) e (x_2, y_2), com x_i distintos, de uma função $y = f(x)$. Para aproximar $f(z)$, $z \in (x_0, x_2)$, faz-se

$$f(x) \approx P_2(x) = a_0 + a_1 x + a_2 x^2,$$

sendo $P_2(x)$ um polinômio interpolador de grau 2. Impondo que o polinômio interpolador passe pelos três pontos-base, tem-se o sistema linear de ordem 3

$$\begin{matrix} P_2(x_0) = y_0 \\ P_2(x_1) = y_1 \\ P_2(x_2) = y_2 \end{matrix} \rightarrow \begin{cases} a_0 + a_1 x_0 + a_2 x_0^2 = y_0 \\ a_0 + a_1 x_1 + a_2 x_1^2 = y_1 \\ a_0 + a_1 x_2 + a_2 x_2^2 = y_2 \end{cases} \Longleftrightarrow \begin{bmatrix} 1 & x_0 & x_0^2 \\ 1 & x_1 & x_1^2 \\ 1 & x_2 & x_2^2 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} y_0 \\ y_1 \\ y_2 \end{bmatrix}.$$

A matriz dos coeficientes é a matriz de Vandermonde, e o sistema de equações lineares admite uma única solução, pois $\det(X) = (x_2 - x_0)(x_2 - x_1)(x_1 - x_0) \neq 0$. Conseqüentemente, por três pontos passa um único polinômio de grau 2. Este fato pode ser generalizado, dizendo-se que por $n + 1$ pontos passa um único polinômio de grau n.

Exemplo 3.2 Calcular $P_2(0{,}2)$ usando os dados da Tabela 3.1.

Os coeficientes do polinômio interpolador são determinados pela solução do sistema linear

$$\begin{bmatrix} 1 & 0{,}1 & 0{,}01 \\ 1 & 0{,}6 & 0{,}36 \\ 1 & 0{,}8 & 0{,}64 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} 1{,}221 \\ 3{,}320 \\ 4{,}953 \end{bmatrix}.$$

Usando a decomposição LU com pivotação parcial, vista na Seção 2.4, têm-se os fatores

$$L = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 0{,}714 & 1 \end{bmatrix}; \; U = \begin{bmatrix} 1 & 0{,}1 & 0{,}01 \\ 0 & 0{,}7 & 0{,}63 \\ 0 & 0 & -0{,}1 \end{bmatrix} \quad \text{e} \quad P = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}.$$

O sistema triangular inferior $Lt = Py$ é resolvido pelas substituições sucessivas (2.7), fornecendo $t = [1{,}221 \;\; 3{,}732 \;\; -0{,}567]^T$. Os coeficientes a_i do polinômio interpolador são obtidos pela solução do sistema triangular superior $Ua = t$ usando as substituições retroativas (2.8), o que fornece $a = [1{,}141 \;\; 0{,}231 \;\; 5{,}667]^T$. Conseqüentemente, o polinômio tem a forma

$$P_2(x) = 1{,}141 + 0{,}231x + 5{,}667x^2 \rightsquigarrow P_2(0{,}2) = 1{,}414.$$

Como o polinômio passa pelos pontos-base, pode ser constatado que $P_2(0{,}1) = 1{,}221$, $P_2(0{,}6) = 3{,}320$ e $P_2(0{,}8) = 4{,}953$. ∎

Como visto, esta metodologia de determinar os coeficientes do polinômio interpolador por meio da resolução de um sistema de equações lineares, apesar de ser conceitualmente simples, requer um certo esforço computacional, pois a decomposição LU tem complexidade da ordem de n^3, de acordo com a Tabela 2.4. Considerando que por $n+1$ pontos passa um único polinômio de grau n, deve ser procurada uma metodologia alternativa de modo a evitar a solução de um sistema de equações lineares. Serão vistas, a seguir, outras formas de obter um polinômio interpolador, as quais apresentam uma menor ordem de complexidade.

3.2 Polinômios de Lagrange

Sejam dados $n+1$ pontos (x_0, y_0), $(x_1, y_1), \ldots, (x_n, y_n)$, sendo x_i distintos, tais que $y_i = f(x_i)$ e $x \in (x_0, x_n)$. Deseja-se construir um polinômio $L_n(x)$ de grau não superior a n e possuindo nos pontos x_i o mesmo valor da função $f(x)$, isto é,

$$L_n(x_i) = y_i, \; i = 0, 1, 2, \ldots, n. \tag{3.2}$$

3.2.1 Fórmula de Lagrange

Inicialmente, sejam os polinômios de grau n, $P_i(x)$, $i = 0, 1, 2, \ldots, n$, tais que

$$P_i(x_i) \neq 0 \;\text{ e }\; P_i(x_j) = 0, \; \forall \, i \neq j.$$

Assim,

$$P_0(x) = (x - x_1)(x - x_2)(x - x_3) \ldots (x - x_n),$$

$$P_1(x) = (x - x_0)(x - x_2)(x - x_3)\ldots(x - x_n),$$
$$P_2(x) = (x - x_0)(x - x_1)(x - x_3)\ldots(x - x_n),$$
$$\vdots$$
$$P_n(x) = (x - x_0)(x - x_1)(x - x_2)\ldots(x - x_{n-1}),$$

ou seja,

$$P_i(x) = \prod_{\substack{j=0 \\ j \neq i}}^{n} (x - x_j),\ i = 0, 1, 2, \ldots, n. \tag{3.3}$$

Considerando que $L_n(x)$ é de grau não superior a n, ele pode ser escrito como uma combinação linear dos polinômios $P_i(x)$, isto é,

$$L_n(x) = c_0P_0(x) + c_1P_1(x) + c_2P_2(x) + \cdots + c_nP_n(x)$$
$$L_n(x) = \sum_{i=0}^{n} c_iP_i(x). \tag{3.4}$$

Comparando (3.2) e (3.4) e em vista de que $P_i(x_j) = 0,\ \forall\, i \neq j$, tem-se

$$L_n(x_i) = y_i = c_iP_i(x_i) \longrightarrow c_i = \frac{y_i}{P_i(x_i)}.$$

Substituindo o valor de c_i em (3.4), tem-se

$$L_n(x) = \sum_{i=0}^{n} \frac{y_i}{P_i(x_i)} P_i(x).$$

Em vista de (3.3), tem-se a fórmula do polinômio interpolador de Lagrange de grau n

$$\boxed{L_n(x) = \sum_{i=0}^{n} y_i \prod_{\substack{j=0 \\ j \neq i}}^{n} \frac{x - x_j}{x_i - x_j}}. \tag{3.5}$$

Exemplo 3.3 Calcular $L_1(0{,}2)$ a partir dos dados do Exemplo 3.1

i	0	1
x_i	0,1	0,6
y_i	1,221	3,320

Usando (3.5) com $n = 1$, tem-se

$$L_1(x) = y_0\frac{x - x_1}{x_0 - x_1} + y_1\frac{x - x_0}{x_1 - x_0},$$

$$L_1(0{,}2) = 1{,}221\frac{0{,}2 - 0{,}6}{0{,}1 - 0{,}6} + 3{,}320\frac{0{,}2 - 0{,}1}{0{,}6 - 0{,}1} \leadsto L_1(0{,}2) = 1{,}641. \quad \blacksquare$$

Este resultado é igual ao obtido no Exemplo 3.1 usando a fórmula de interpolação linear (3.1). De fato, (3.5) com $n = 1$ é idêntica a (3.1)

$$P_1(x) = y_0 + \frac{y_1 - y_0}{x_1 - x_0}(x - x_0) = \frac{y_0x_1 - y_0x_0 + y_1x - y_1x_0 - y_0x + y_0x_0}{x_1 - x_0},$$

$$P_1(x) = \frac{y_0(x_1 - x) + y_1(x - x_0)}{x_1 - x_0} = y_0\frac{x - x_1}{x_0 - x_1} + y_1\frac{x - x_0}{x_1 - x_0} \leadsto P_1(x) = L_1(x).$$

Exemplo 3.4 Calcular $L_2(0{,}2)$ usando os dados da tabela do Exemplo 3.2

i	0	1	2
x_i	0,1	0,6	0,8
y_i	1,221	3,320	4,953

A fórmula de Lagrange (3.5) com $n = 2$ torna-se

$$L_2(x) = y_0\frac{(x-x_1)(x-x_2)}{(x_0-x_1)(x_0-x_2)} + y_1\frac{(x-x_0)(x-x_2)}{(x_1-x_0)(x_1-x_2)} + y_2\frac{(x-x_0)(x-x_1)}{(x_2-x_0)(x_2-x_1)},$$

$$L_2(0{,}2) = 1{,}221\frac{(0{,}2 - 0{,}6)(0{,}2 - 0{,}8)}{(0{,}1 - 0{,}6)(0{,}1 - 0{,}8)} + 3{,}320\frac{(0{,}2 - 0{,}1)(0{,}2 - 0{,}8)}{(0{,}6 - 0{,}1)(0{,}6 - 0{,}8)} +$$

$$4{,}953\frac{(0{,}2 - 0{,}1)(0{,}2 - 0{,}6)}{(0{,}8 - 0{,}1)(0{,}8 - 0{,}6)} \leadsto L_2(0{,}2) = 1{,}414.$$

■

Considerando que $f(0{,}2) = e^{2\times 0{,}2} \approx 1{,}492$, o erro cometido foi menor do que quando $L_1(0{,}2)$ foi utilizado. Portanto, quando o grau do polinômio interpolador é aumentado, a exatidão do resultado é melhorada. Obviamente, a interpolação de Lagrange requer um menor esforço computacional do que resolver um sistema de equações lineares, como será mostrado mais adiante.

3.2.2 Dispositivo prático

Um dispositivo prático pode ser construído para facilitar o uso dos polinômios de Lagrange. Para tal, seja a matriz

$$G = \begin{bmatrix} x - x_0 & x_0 - x_1 & x_0 - x_2 & \cdots & x_0 - x_n \\ x_1 - x_0 & x - x_1 & x_1 - x_2 & \cdots & x_1 - x_n \\ x_2 - x_0 & x_2 - x_1 & x - x_2 & \cdots & x_2 - x_n \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_n - x_0 & x_n - x_1 & x_n - x_2 & \cdots & x - x_n \end{bmatrix}.$$

Acrescentando o termo $(x - x_i)/(x - x_i)$ na fração de (3.5), esta não se modificará numericamente

$$L_n(x) = \sum_{i=0}^{n} y_i \prod_{\substack{j=0 \\ j \neq i}}^{n} \frac{x - x_j}{x_i - x_j} \times \frac{x - x_i}{x - x_i} \longrightarrow \boxed{L_n(x) = G_d \sum_{i=0}^{n} \frac{y_i}{G_i}}, \tag{3.6}$$

onde G_d é o produto dos elementos da diagonal principal da matriz G e G_i é o produto dos elementos da $(i+1)$-ésima linha de G.

Exemplo 3.5 Determinar $L_2(0{,}2)$ por (3.6) usando os dados do Exemplo 3.4.

A matriz G é

$$G = \begin{bmatrix} 0{,}2 - 0{,}1 & 0{,}1 - 0{,}6 & 0{,}1 - 0{,}8 \\ 0{,}6 - 0{,}1 & 0{,}2 - 0{,}6 & 0{,}6 - 0{,}8 \\ 0{,}8 - 0{,}1 & 0{,}8 - 0{,}6 & 0{,}2 - 0{,}8 \end{bmatrix} = \begin{bmatrix} 0{,}1 & -0{,}5 & -0{,}7 \\ 0{,}5 & -0{,}4 & -0{,}2 \\ 0{,}7 & 0{,}2 & -0{,}6 \end{bmatrix}.$$

Conseqüentemente,

$$G_d = (0{,}1)(-0{,}4)(-0{,}6) = 0{,}024,$$

$$G_0 = (0{,}1)(-0{,}5)(-0{,}7) = 0{,}035,$$

$$G_1 = (0{,}5)(-0{,}4)(-0{,}2) = 0{,}040 \text{ e}$$

$$G_2 = (0{,}7)(0{,}2)(-0{,}6) = -0{,}084.$$

Usando (3.6) com $n = 2$, tem-se

$$L_2(x) = G_d \left(\frac{y_0}{G_0} + \frac{y_1}{G_1} + \frac{y_2}{G_2} \right),$$

$$L_2(0{,}2) = 0{,}024 \left(\frac{1{,}221}{0{,}035} + \frac{3{,}320}{0{,}040} + \frac{4{,}953}{-0{,}084} \right) \leadsto L_2(0{,}2) = 1{,}414.$$

■

3.2.3 Algoritmo e complexidade computacional

Um algoritmo para interpolação, usando os polinômios de Lagrange, é mostrado na Figura 3.2. Os parâmetros de entrada são o número *m* de pontos, o vetor *x* contendo as *m* abscissas x_i, o vetor *y* com as *m* ordenadas y_i e o ponto *z* a ser interpolado. O parâmetro de saída é a ordenada *r* do polinômio de Lagrange de grau *m – 1* avaliado no ponto *z*.

Neste algoritmo, (3.5) é implementada de forma a reduzir o número de operações de divisão, se comparado com o algoritmo da Figura 1.9, na página 18.

Algoritmo Polinômio_Lagrange
{ **Objetivo:** Interpolar valor em tabela usando polinômio de Lagrange }
parâmetros de entrada m, x, y, z
  { número de pontos, abscissas, ordenadas e valor a interpolar }
parâmetros de saída r { valor interpolado }
  $r \leftarrow 0$
  para $i \leftarrow 1$ **até** m **faça**
    $c \leftarrow 1$; $d \leftarrow 1$
    para $j \leftarrow 1$ **até** m **faça**
      se $i \neq j$ **então**
        $c \leftarrow c * (z - x(j))$; $d \leftarrow d * (x(i) - x(j))$
      fim se
    fim para
    $r \leftarrow r + y(i) * c/d$
  fim para
fim algoritmo

Figura 3.2 Interpolação por polinômio de Lagrange.

A complexidade computacional do algoritmo da Figura 3.2 é mostrada na Tabela 3.2, a qual deve ser comparada com aquela apresentada na Tabela 1.6, na página 19.

Tabela 3.2 Complexidade da interpolação de Lagrange.

(n: grau do polinômio interpolador.)

Operações	Complexidade
adições	$2n^2 + 3n + 1$
multiplicações	$2n^2 + 3n + 1$
divisões	$n + 1$

Exemplo 3.6 Resolver o problema do Exemplo 3.4 utilizando uma implementação do algoritmo da Figura 3.2.

```
% Os parametros de entrada
m = 3
x = 0.1000    0.6000    0.8000
y = 1.2210    3.3200    4.9530
z = 0.2000
% fornecem o resultado
r = 1.4141
```

■

3.3 Polinômios de Newton

Como visto na seção anterior, os polinômios de Lagrange constituem um modo de interpolar sem a necessidade de resolver um sistema de equações lineares. Será mostrado, a seguir, um esquema alternativo para a construção de um polinômio interpolador.

3.3.1 Operador de diferença dividida

Seja a função $y = f(x)$, que passa pelos pontos (x_i, y_i), $i = 0, 1, 2, \ldots, n$. O operador de diferença dividida Δ^i de ordem i é definido como

a) ordem 0: $\Delta^0 y_i = y_i = [x_i]$,

b) ordem 1: $\Delta y_i = \dfrac{\Delta^0 y_{i+1} - \Delta^0 y_i}{x_{i+1} - x_i} = \dfrac{y_{i+1} - y_i}{x_{i+1} - x_i} = [x_i, x_{i+1}]$,

c) ordem 2: $\Delta^2 y_i = \dfrac{\Delta y_{i+1} - \Delta y_i}{x_{i+2} - x_i} = [x_i, x_{i+1}, x_{i+2}]$,

d) ordem n: $\Delta^n y_i = \dfrac{\Delta^{n-1} y_{i+1} - \Delta^{n-1} y_i}{x_{i+n} - x_i} = [x_i, x_{i+1}, \ldots, x_{i+n}]$.

O Teorema 3.1 [3, Teorema 6.9] mostra uma importante propriedade das diferenças divididas de um polinômio. Essa propriedade é fundamental para a construção dos polinômios de Newton.

Teorema 3.1 *Se $y = f(x)$ for um polinômio de grau n, então suas diferenças divididas de ordem $n + 1$ são identicamente nulas, isto é,*

$$[x, x_0, x_1, \ldots, x_n] = 0 \; \forall \; x.$$

Exemplo 3.7 Verificar a tabela de diferenças divididas do polinômio $y = 5x^3 - 2x^2 - x + 3$ para alguns pontos x_i no intervalo [0; 0,9]

i	x_i	y_i	Δy_i	$\Delta^2 y_i$	$\Delta^3 y_i$	$\Delta^4 y_i$
0	0,0	3,000	$-1{,}20$	0,5	5	0
1	0,2	2,760	$-1{,}05$	2,5	5	0
2	0,3	2,655	$-0{,}55$	5,0	5	
3	0,4	2,600	1,45	8,0		
4	0,7	3,035	5,45			
5	0,9	4,125				

■

3.3.2 Fórmula de Newton

Sejam os $n+1$ pontos (x_i, y_i), $i = 0, 1, 2, \ldots, n$, com x_i distintos, tais que $y_i = P(x_i)$, sendo $P(x)$ um polinômio de grau n. A diferença dividida de ordem 1 é

$$[x, x_0] = \frac{P(x) - P(x_0)}{x - x_0}$$

ou

$$P(x) = P(x_0) + [x, x_0](x - x_0). \tag{3.7}$$

No entanto, a diferença dividida de ordem 2

$$[x, x_0, x_1] = \frac{[x, x_0] - [x_0, x_1]}{x - x_1} \leadsto [x, x_0] = [x_0, x_1] + [x, x_0, x_1](x - x_1).$$

Substituindo esta equação em (3.7), tem-se

$$P(x) = P(x_0) + [x_0, x_1](x - x_0) + [x, x_0, x_1](x - x_0)(x - x_1). \tag{3.8}$$

Contudo, a diferença dividida de ordem 3

$$[x, x_0, x_1, x_2] = \frac{[x, x_0, x_1] - [x_0, x_1, x_2]}{x - x_2} \leadsto$$

$$[x, x_0, x_1] = [x_0, x_1, x_2] + [x, x_0, x_1, x_2](x - x_2).$$

Substituindo a equação acima em (3.8), obtém-se

$$P(x) = P(x_0) + [x_0, x_1](x - x_0) + [x_0, x_1, x_2](x - x_0)(x - x_1) +$$
$$+ [x, x_0, x_1, x_2](x - x_0)(x - x_1)(x - x_2). \tag{3.9}$$

Continuando o desenvolvimento de $[x, x_0, x_1, x_2]$, chega-se a

$$P(x) = P(x_0) + [x_0, x_1](x - x_0) + [x_0, x_1, x_2](x - x_0)(x - x_1) +$$
$$+ [x_0, x_1, x_2, x_3](x - x_0)(x - x_1)(x - x_2) + \ldots +$$
$$+ [x_0, x_1, \ldots, x_n](x - x_0)(x - x_1)\ldots(x - x_{n-1}) +$$
$$+ [x, x_0, x_1, \ldots, x_n](x - x_0)(x - x_1)\ldots(x - x_n).$$

Considerando que $P(x)$ é um polinômio de grau n, então pelo Teorema 3.1 resulta que

$$[x, x_0, x_1, \ldots, x_n] = 0.$$

Fazendo $y_0 = P(x_0)$ e usando a notação de diferenças divididas com o operador Δ, tem-se o polinômio de Newton de grau n

$$P_n(x) = y_0 + \Delta y_0(x - x_0) + \Delta^2 y_0(x - x_0)(x - x_1) + \ldots + \Delta^n y_0(x - x_0)\ldots(x - x_{n-1})$$

$$\boxed{P_n(x) = y_0 + \sum_{i=1}^{n} \Delta^i y_0 \prod_{j=0}^{i-1}(x - x_j)}. \tag{3.10}$$

Uma vantagem dos polinômios de Newton sobre os de Lagrange é que para aumentar o grau basta acrescentar um novo termo em vez de montar todo o polinômio novamente.

Exemplo 3.8 Calcular $P_1(0{,}2)$ a partir dos dados

x	0,1	0,6
y	1,221	3,320

Pelo uso de (3.10) com $n = 1$, tem-se que

$$P_1(x) = y_0 + \Delta y_0(x - x_0).$$

A tabela de diferenças divididas é

i	x_i	y_i	Δy_i
0	0,1	1,221	4,198
1	0,6	3,320	

$$P_1(0{,}2) = 1{,}221 + 4{,}198(0{,}2 - 0{,}1) \rightsquigarrow P_1(0{,}2) = 1{,}641.$$ ■

Deve ser observado que este resultado é idêntico aos obtidos nos Exemplos 3.1 e 3.3. É fácil verificar que

$$P_1(x) = y_0 + \Delta y_0(x - x_0) = y_0 + \frac{y_1 - y_0}{x_1 - x_0}(x - x_0) \quad \text{(comparar com (3.1))}.$$

Exemplo 3.9 Determinar $P_2(1{,}2)$ usando os dados

x	0,9	1,1	2,0
y	3,211	2,809	1,614

Por (3.10) com $n = 2$, tem-se que

$$P_2(x) = y_0 + \Delta y_0(x - x_0) + \Delta^2 y_0(x - x_0)(x - x_1).$$

A tabela de diferenças divididas é

i	x_i	y_i	Δy_i	$\Delta^2 y_i$
0	0,9	3,211	−2,010	0,620
1	1,1	2,809	−1,328	
2	2,0	1,614		

$$P_2(1{,}2) = 3{,}211 + (-2{,}010)(1{,}2 - 0{,}9) + (0{,}620)(1{,}2 - 0{,}9)(1{,}2 - 1{,}1) \rightsquigarrow$$

$$P_2(1{,}2) = 2{,}627.$$

■

Exemplo 3.10 Calcular $P_4(0{,}2)$ a partir de

x	0,1	0,3	0,4	0,6	0,7
y	0,3162	0,5477	0,6325	0,7746	0,8367

Por (3.10) com $n = 4$, tem-se

$$P_4(x) = y_0 + \Delta y_0(x - x_0) + \Delta^2 y_0(x - x_0)(x - x_1) +$$

$$\Delta^3 y_0(x - x_0)(x - x_1)(x - x_2) + \Delta^4 y_0(x - x_0)(x - x_1)(x - x_2)(x - x_3).$$

A tabela de diferenças divididas é

i	x_i	y_i	Δy_i	$\Delta^2 y_i$	$\Delta^3 y_i$	$\Delta^4 y_i$
0	0,1	0,3162	1,1575	−1,0317	1,1468	−1,2447
1	0,3	0,5477	0,8480	−0,4583	0,4000	
2	0,4	0,6325	0,7105	−0,2983		
3	0,6	0,7746	0,6210			
4	0,7	0,8367				

$$P_4(0{,}2) = 0{,}3162 + 1{,}1575(0{,}1) + (-1{,}0317)(0{,}1)(-0{,}1) +$$

$$1{,}1468(0{,}1)(-0{,}1)(-0{,}2) + (-1{,}2447)(0{,}1)(-0{,}1)(-0{,}2)(-0{,}4) \rightsquigarrow$$

$$P_4(0{,}2) = 0{,}4456.$$

■

3.3.3 Algoritmo e complexidade computacional

A Figura 3.3 exibe um algoritmo para interpolar um valor z em uma tabela definida pelos vetores x e y usando um polinômio de Newton de diferenças divididas de grau n. Para tal, o polinômio (3.10)

$$P_n(x) = y_0 + \sum_{i=1}^{n} \Delta^i y_0 \prod_{j=0}^{i-1}(x - x_j)$$

é escrito de forma a ser avaliado pelo processo de Horner (ver Figura 6.2, na página 273)

$$\boxed{P_n(z) = (\ldots(\Delta^n y_0(z - x_{n-1}) + \Delta^{n-1} y_0)(z - x_{n-2}) + \ldots + \Delta^2 y_0)(z - x_1) + \Delta y_0)(z - x_0) + y_0}.$$

Os parâmetros de entrada do algoritmo são o número m de pontos, o vetor x contendo as m abscissas x_i, o vetor y com as m ordenadas y_i e o ponto z a ser interpolado. O parâmetro de saída é o valor r do polinômio de Newton de grau $m - 1$ avaliado no ponto z.

```
Algoritmo Polinômio_Newton
{ Objetivo: Interpolar valor em tabela usando polinômio de Newton }
parâmetros de entrada m, x, y, z
  { número de pontos, abscissas, ordenadas e valor a interpolar }
parâmetros de saída r  { valor interpolado }
  para i ← 1 até m faça
    Dely(i) ← y(i)
  fim para
  { construção das diferenças divididas }
  para k ← 1 até m − 1 faça
    para i ← m até k + 1 passo −1 faça
      Dely(i) ← (Dely(i) − Dely(i − 1))/(x(i) − x(i − k))
    fim para
  fim para
  { avaliação do polinômio pelo processo de Horner }
  r ← Dely(m)
  para i ← m − 1 até 1 passo −1 faça
    r ← r ∗ (z − x(i)) + Dely(i)
  fim para
fim algoritmo
```

Figura 3.3 Interpolação por polinômio de Newton com diferenças divididas.

Este algoritmo utiliza um vetor auxiliar *Dely*, no qual é construída a tabela de diferenças divididas. Para economizar espaço de memória, os valores de $\Delta^0 y_0$, Δy_0, ... , $\Delta^n y_0$ são armazenados nas primeiras $n + 1$ posições de *Dely*, conforme o esquema para cinco pontos

i	x_i	y_i	$Dely_i^{(1)}$	$Dely_i^{(2)}$	$Dely_i^{(3)}$	$Dely_i^{(4)}$
1	x_0	y_0	$\Delta^0 y_0$	$\Delta^0 y_0$	$\Delta^0 y_0$	$\Delta^0 y_0$
2	x_1	y_1	Δy_0	Δy_0	Δy_0	Δy_0
3	x_2	y_2	Δy_1	$\Delta^2 y_0$	$\Delta^2 y_0$	$\Delta^2 y_0$
4	x_3	y_3	Δy_2	$\Delta^2 y_1$	$\Delta^3 y_0$	$\Delta^3 y_0$
5	x_4	y_4	Δy_3	$\Delta^2 y_2$	$\Delta^3 y_1$	$\Delta^4 y_0$

,

onde $Dely_i^{(k)}$ significa i-ésima posição do vetor *Dely* na k-ésima repetição do comando **para** $k \leftarrow 1$ **até** $m - 1$ **faça**. A complexidade computacional do algoritmo da Figura 3.3 é mostrada na Tabela 3.3.

Exemplo 3.11 Calcular $P_4(0{,}2)$ a partir dos dados da tabela do Exemplo 3.10 usando o algoritmo da Figura 3.3.

Tabela 3.3 Complexidade da interpolação de Newton.

(n: grau do polinômio interpolador.)

Operações	Complexidade
adições	$n^2 + 4n$
multiplicações	n
divisões	$\frac{1}{2}n^2 + \frac{1}{2}n$

```
% Os parametros de entrada
m = 5
x = 0.1000    0.3000    0.4000    0.6000    0.7000
y = 0.3162    0.5477    0.6325    0.7746    0.8367
z = 0.2000
% produzem o resultado
r = 0.4456
```

A seqüência de vetores *Dely* produzida pelo algoritmo é

i	x_i	y_i	$Dely_i^{(1)}$	$Dely_i^{(2)}$	$Dely_i^{(3)}$	$Dely_i^{(4)}$
1	0,1	0,3162	0,3162	0,3162	0,3162	0,3162
2	0,3	0,5477	1,1575	1,1575	1,1575	1,1575
3	0,4	0,6325	0,8480	−1,0317	−1,0317	−1,0317
4	0,6	0,7746	0,7105	−0,4583	1,1468	1,1468
5	0,7	0,8367	0,6210	−0,2983	0,4000	−1,2447

Exemplo 3.12 Calcular $P_1(0{,}2)$, $P_2(0{,}2)$ e $P_3(0{,}2)$ usando os dados da tabela do Exemplo 3.10 e o algoritmo da Figura 3.3.

```
% Calculo de P1(0,2)
m = 2
x = 0.1000    0.3000
y = 0.3162    0.5477
z = 0.2000
r = 0.4320
% Calculo de P2(0,2)
m = 3
x = 0.1000    0.3000    0.4000
y = 0.3162    0.5477    0.6325
z = 0.2000
r = 0.4423
% Calculo de P3(0,2)
m = 4
x = 0.1000    0.3000    0.4000    0.6000
y = 0.3162    0.5477    0.6325    0.7746
z = 0.2000
r = 0.4446
```

Considerando que $y = \sqrt{x}$, portanto, o valor exato é $\sqrt{0{,}2} \approx 0{,}4472$. A partir dos resultados acima e mais o valor de $P_4(0{,}2)$ obtido no Exemplo 3.11, pode-se verificar que a diferença entre o valor interpolado e o exato diminui à medida que o grau do polinômio interpolador aumenta

n	$P_n(0{,}2)$	$\lvert P_n(0{,}2) - \sqrt{0{,}2}\rvert$
1	0,4320	0,0152
2	0,4423	0,0049
3	0,4446	0,0026
4	0,4456	0,0016

■

3.4 Polinômios de Gregory-Newton

Quando os valores das abscissas x_i forem igualmente espaçados, a fórmula de Newton pode ser simplificada, resultando na fórmula de Gregory-Newton. Portanto, o polinômio de Gregory-Newton é um caso particular do polinômio de Newton para pontos igualmente espaçados.

3.4.1 Operador de diferença finita ascendente

Seja a função $y = f(x)$ que passa pelos pontos (x_i, y_i), $i = 0, 1, 2, \ldots, n$, sendo $x_{i+1} - x_i = h \ \forall\, i$. O operador de diferença finita ascendente Δ^i de ordem i é definido como

a) ordem 0: $\Delta^0 y_i = y_i$,

b) ordem 1: $\Delta y_i = \Delta^0 y_{i+1} - \Delta^0 y_i = y_{i+1} - y_i$,

c) ordem 2: $\Delta^2 y_i = \Delta y_{i+1} - \Delta y_i$,

d) ordem n: $\Delta^n y_i = \Delta^{n-1} y_{i+1} - \Delta^{n-1} y_i$.

Exemplo 3.13 Verificar a tabela de diferenças finitas

i	x_i	y_i	Δy_i	$\Delta^2 y_i$	$\Delta^3 y_i$	$\Delta^4 y_i$
0	3,5	9,82	1,09	0,05	−0,10	2,11
1	4,0	10,91	1,14	−0,05	2,01	
2	4,5	12,05	1,09	1,96		
3	5,0	13,14	3,05			
4	5,5	16,19				

■

A relação entre os operadores de diferença finita e dividida é dada pela expressão

$$\boxed{\Delta^n y_i = \frac{\Delta^n y_i}{n!h^n}}. \tag{3.11}$$

Exemplo 3.14 Para a tabela do Exemplo 3.13,

$$\Delta y_0 = \frac{\Delta y_0}{1!h} \rightsquigarrow \frac{10{,}91 - 9{,}82}{4{,}0 - 3{,}5} = \frac{1{,}09}{1!\ 0{,}5} = 2{,}18 \text{ e}$$

$$\Delta^2 y_1 = \frac{\Delta^2 y_1}{2!h^2} \rightsquigarrow$$

$$\frac{\Delta y_2 - \Delta y_1}{x_3 - x_1} = \frac{\frac{y_3-y_2}{x_3-x_2} - \frac{y_2-y_1}{x_2-x_1}}{x_3 - x_1} = \frac{\frac{13{,}14-12{,}05}{5{,}0-4{,}5} - \frac{12{,}05-10{,}91}{4{,}5-4{,}0}}{5{,}0 - 4{,}0} = \frac{-0{,}05}{2!\ 0{,}5^2} = -0{,}10.$$

■

3.4.2 Fórmula de Gregory-Newton

Seja a fórmula do polinômio interpolador de Newton (3.10) na forma

$$P_n(x) = y_0 + \Delta y_0(x-x_0) + \Delta^2 y_0(x-x_0)(x-x_1) + \ldots + \Delta^n y_0(x-x_0)\ldots(x-x_{n-1})$$

e uma variável auxiliar

$$u_x = u(x) = \frac{x - x_0}{h},$$

da qual verifica-se que

$$x - x_0 = hu_x,$$

$$x - x_1 = x - (x_0 + h) = x - x_0 - h = hu_x - h \rightsquigarrow x - x_1 = h(u_x - 1),$$

$$x - x_2 = x - (x_0 + 2h) = x - x_0 - 2h = hu_x - 2h \rightsquigarrow x - x_2 = h(u_x - 2),$$

$$\vdots$$

$$x - x_{n-1} = x - (x_0 + (n-1)h) = x - x_0 - (n-1)h \rightsquigarrow x - x_{n-1} = h(u_x - n + 1).$$

Substituindo esses valores na fórmula de Newton e aplicando a relação (3.11) entre operadores, tem-se

$$P_n(x) = y_0 + \frac{\Delta y_0}{1!h} hu_x + \frac{\Delta^2 y_0}{2!h^2} hu_x h(u_x - 1) + \ldots +$$

$$\frac{\Delta^n y_0}{n!h^n} hu_x h(u_x - 1) \ldots h(u_x - n + 1).$$

Assim,

$$P_n(x) = y_0 + \Delta y_0 u_x + \frac{\Delta^2 y_0}{2!} u_x(u_x - 1) + \ldots + \frac{\Delta^n y_0}{n!} u_x(u_x - 1)\ldots(u_x - n + 1)$$

$$\boxed{P_n(x) = y_0 + \sum_{i=1}^{n} \frac{\Delta^i y_0}{i!} \prod_{j=0}^{i-1} (u_x - j)}. \tag{3.12}$$

Exemplo 3.15 Calcular $P_1(0{,}2)$, usando os dados da tabela do Exemplo 3.1.

Usando (3.12) com $n = 1$, obtém-se

$$P_1(x) = y_0 + \Delta y_0 u_x.$$

A tabela de diferenças finitas é

i	x_i	y_i	Δy_i
0	0,1	1,221	2,099
1	0,6	3,320	

e a variável $u_x = \dfrac{x - x_0}{h} = \dfrac{0{,}2 - 0{,}1}{0{,}5} = 0{,}2$. Então

$$P_1(0{,}2) = 1{,}221 + 2{,}099(0{,}2) \rightsquigarrow P_1(0{,}2) = 1{,}641. \qquad \blacksquare$$

Este resultado é o mesmo obtido por (3.1) no Exemplo 3.1, por (3.5) no Exemplo 3.3 e por (3.10) no Exemplo 3.8. De fato,

$$P_1(x) = y_0 + \Delta y_0 u_x = y_0 + (y_1 - y_0)\frac{x - x_0}{x_1 - x_0} \qquad \text{(comparar com (3.1))}.$$

Exemplo 3.16 Calcular $P_2(115)$ a partir da tabela

x	110	120	130
y	2,041	2,079	2,114

Usando (3.12) com $n = 2$, tem-se

$$P_2(x) = y_0 + \Delta y_0 u_x + \frac{\Delta^2 y_0}{2!} u_x(u_x - 1).$$

A tabela de diferenças finitas é

i	x_i	y_i	Δy_i	$\Delta^2 y_i$
0	110	2,041	0,038	−0,003
1	120	2,079	0,035	
2	130	2,114		

e a variável $u_x = \dfrac{x - x_0}{h} = \dfrac{115 - 110}{10} = 0{,}5$. Assim,

$$P_2(115) = 2{,}041 + (0{,}038)(0{,}5) + \frac{-0{,}003}{2}(0{,}5)(0{,}5 - 1) \rightsquigarrow P_2(115) = 2{,}060. \qquad \blacksquare$$

3.4.3 Algoritmo e complexidade computacional

Um algoritmo para interpolar um valor z em uma tabela definida pelos vetores x e y, de dimensões m, usando um polinômio de Gregory-Newton de diferenças finitas de grau $m-1$, é mostrado na Figura 3.4. Os parâmetros de entrada são o número *m* de pontos, o vetor *x* contendo as *m* abscissas x_i, o vetor *y* com as *m* ordenadas y_i e o ponto *z* a ser interpolado. O parâmetro de saída é o valor *r* do polinômio de Gregory-Newton de grau *m – 1* avaliado no ponto *z*.

```
Algoritmo Polinômio_Gregory-Newton
{ Objetivo: Interpolar valor em tabela usando polinômio de Gregory-Newton }
parâmetros de entrada m, x, y, z
   { número de pontos, abscissas, ordenadas e valor a interpolar }
parâmetros de saída r  { valor interpolado }
   para i ← 1 até m faça
      Dely(i) ← y(i)
   fimpara
   { construção das diferenças finitas }
   para k ← 1 até m − 1 faça
      para i ← m até k + 1 passo −1 faça
         Dely(i) ← Dely(i) − Dely(i − 1)
      fimpara
   fimpara
   { avaliação do polinômio pelo processo de Horner }
   u ← (z − x(1))/(x(2) − x(1))
   r ← Dely(m)
   para i ← m − 1 até 1 passo −1 faça
      r ← r * (u − i + 1)/i + Dely(i)
   fimpara
fimalgoritmo
```

Figura 3.4 Interpolação por polinômio de Gregory-Newton com diferenças finitas.

Também neste caso, o polinômio (3.12)

$$P_n(x) = y_0 + \sum_{i=1}^{n} \frac{\Delta^i y_0}{i!} \prod_{j=0}^{i-1} (u_x - j)$$

pode ser escrito de forma a ser avaliado pelo processo de Horner

$$\boxed{P_n(x) = \left(\left(\left(\ldots \left(\Delta^n y_0 \frac{u_x - n+1}{n}\right) + \ldots + \Delta^2 y_0\right) \frac{u_x - 1}{2} + \Delta y_0\right) \frac{u_x - 0}{1}\right) + y_0}.$$

De forma análoga ao método de Newton, este algoritmo utiliza um vetor auxiliar *Dely*, no qual é construída a tabela de diferenças finitas. Os valores de $\Delta^0 y_0$, Δy_0, $\ldots$, $\Delta^n y_0$ são armazenados nas primeiras $n+1$ posições de *Dely*, para economizar espaço de memória, conforme o esquema para quatro pontos

i	x_i	y_i	$\text{Dely}_i^{(1)}$	$\text{Dely}_i^{(2)}$	$\text{Dely}_i^{(3)}$
1	x_0	y_0	$\Delta^0 y_0$	$\Delta^0 y_0$	$\Delta^0 y_0$
2	x_1	y_1	Δy_0	Δy_0	Δy_0
3	x_2	y_2	Δy_1	$\Delta^2 y_0$	$\Delta^2 y_0$
4	x_3	y_3	Δy_2	$\Delta^2 y_1$	$\Delta^3 y_0$

,

onde $Dely_i^{(k)}$ significa *i*-ésima posição do vetor *Dely* na *k*-ésima repetição do comando **para** *k* $\leftarrow$ *1* **até** $m-1$ **faça**. A Tabela 3.4 mostra a complexidade computacional do algoritmo da Figura 3.4. Comparando com a complexidade do algoritmo de Newton, dado na Tabela 3.3, verifica-se que o algoritmo de Gregory-Newton apresenta uma complexidade menor em relação à operação de divisão.

Tabela 3.4 Complexidade da interpolação de Gregory-Newton.

(n: grau do polinômio interpolador.)

Operações	Complexidade
adições	n^2+4n+2
multiplicações	n
divisões	$n+1$

Exemplo 3.17 Calcular $P_2(115)$ com os dados da tabela do Exemplo 3.16 usando o algoritmo da Figura 3.4.

```
% Os parametros de entrada
m = 3
x = 110   120   130
y = 2.0410    2.0790    2.1140
z = 115
% produzem o resultado
r = 2.0604
```

■

Além disso, a seqüência de vetores *Dely* produzida pelo algoritmo é

i	x_i	y_i	$Dely_i^{(1)}$	$Dely_i^{(2)}$
1	110	2,041	2,041	2,041
2	120	2,079	0,038	0,038
3	130	2,114	0,035	−0,003

.

3.5 Escolha dos pontos para interpolação

Uma questão importante que surge no uso da interpolação é como fazer a escolha dos pontos a serem utilizados. Até agora foram vistos exemplos nos quais todos os pontos da tabela eram usados. No entanto, na prática, faz-se necessário escolher $n+1$ pontos dentre os m valores de uma tabela, sendo $m > n+1$, para que seja construído um polinômio interpolador de grau n. Esta escolha é importante, pois não se devem construir polinômios de grau elevado por causa do erro de arredondamento, e deve-se evitar uma extrapolação na qual o valor de z esteja fora do intervalo dos pontos utilizados para construir o polinômio.

Exemplo 3.18 Interpolar $z = 1{,}4$ usando um polinômio de terceiro grau com os dados

x	0,7	1,2	1,3	1,5	2,0	2,3	2,6
y	0,043	1,928	2,497	3,875	9,000	13,467	19,176

.

Para determinar um polinômio interpolador de grau 3 são necessários 4 pontos. O ponto a ser interpolado deve ser o mais próximo possível destes 4 pontos. Inicialmente são escolhidos 2 pontos, devendo o valor a ser interpolado (1,4) estar entre eles, a saber 1,3 e 1,5. O terceiro ponto será 1,2 e não 2,0, porque $1{,}4 - 1{,}2 < 2{,}0 - 1{,}4$. Analogamente, o quarto ponto será 2,0 e não 0,7, pois $2{,}0 - 1{,}4 < 1{,}4 - 0{,}7$. Assim, a interpolação cúbica utilizará os quatro pontos

i	0	1	2	3
x_i	1,2	1,3	1,5	2,0
y_i	1,928	2,497	3,875	9,000

.

Usando o polinômio de Lagrange, a matriz G é

$$G = \begin{bmatrix} 1{,}4-1{,}2 & 1{,}2-1{,}3 & 1{,}2-1{,}5 & 1{,}2-2{,}0 \\ 1{,}3-1{,}2 & 1{,}4-1{,}3 & 1{,}3-1{,}5 & 1{,}3-2{,}0 \\ 1{,}5-1{,}2 & 1{,}5-1{,}3 & 1{,}4-1{,}5 & 1{,}5-2{,}0 \\ 2{,}0-1{,}2 & 2{,}0-1{,}3 & 2{,}0-1{,}5 & 1{,}4-2{,}0 \end{bmatrix} = \begin{bmatrix} 0{,}2 & -0{,}1 & -0{,}3 & -0{,}8 \\ 0{,}1 & 0{,}1 & -0{,}2 & -0{,}7 \\ 0{,}3 & 0{,}2 & -0{,}1 & -0{,}5 \\ 0{,}8 & 0{,}7 & 0{,}5 & -0{,}6 \end{bmatrix}.$$

Por isso,

$$G_d = (0{,}2)(0{,}1)(-0{,}1)(-0{,}6) = 1{,}2{\times}10^{-3},$$
$$G_0 = (0{,}2)(-0{,}1)(-0{,}3)(-0{,}8) = -4{,}8{\times}10^{-3},$$
$$G_1 = (0{,}1)(0{,}1)(-0{,}2)(-0{,}7) = 1{,}4{\times}10^{-3},$$
$$G_2 = (0{,}3)(0{,}2)(-0{,}1)(-0{,}5) = 3{,}0{\times}10^{-3} \text{ e}$$
$$G_3 = (0{,}8)(0{,}7)(0{,}5)(-0{,}6) = -1{,}68{\times}10^{-1}.$$

Usando (3.6) com $n = 3$, tem-se

$$L_3(x) = G_d\left(\frac{y_0}{G_0} + \frac{y_1}{G_1} + \frac{y_2}{G_2} + \frac{y_3}{G_3}\right),$$
$$L_3(1{,}4) = 1{,}2{\times}10^{-3}\left(\frac{1{,}928}{-4{,}8{\times}10^{-3}} + \frac{2{,}497}{1{,}4{\times}10^{-3}} + \frac{3{,}875}{3{,}0{\times}10^{-3}} + \frac{9{,}000}{-1{,}68{\times}10^{-1}}\right) \rightsquigarrow$$
$$L_3(1,4) = 3{,}144.$$

■

3.6 Erro de truncamento da interpolação polinomial

Erro de truncamento é o erro cometido ao aproximar uma função $f(x)$ por um polinômio interpolador. Pode ser mostrado que se $P_n(x)$ for um polinômio interpolador de grau n de Lagrange, Newton ou Gregory-Newton, então o erro de truncamento [22] é dado pela expressão

$$\boxed{T_n(x) = \frac{f^{n+1}(\xi)}{(n+1)!}\prod_{i=0}^{n}(x-x_i),\ x_0 < \xi < x_n}. \tag{3.13}$$

Sendo $f(x)$ definida no intervalo $[a, b]$ que contém os pontos $x_0, x_1, \ldots, x_n$, e supondo que a derivada $f^{n+1}(x)$ existe e é contínua no intervalo (a, b), então, devido à dificuldade de localizar ξ, na prática ele é tomado como o ponto no intervalo $[x_0, x_n] \subset (a, b)$, onde $f^{n+1}(x)$ apresenta o maior valor em módulo. Deste modo, (3.13) fornecerá a cota máxima do erro de truncamento cometido.

Exemplo 3.19 Sendo $f(x) = 2x^4 + 3x^2 + 1$, calcular $P_2(0{,}1)$ e $T_2(0{,}1)$ a partir da tabela

x	0,0	0,2	0,4
y	1,0000	1,1232	1,5312

Cálculo de $P_2(0{,}1)$:

Utilizando o polinômio de Gregory-Newton de grau 2, dado por (3.12), tem-se

$$P_2(x) = y_0 + \Delta y_0 u_x + \frac{\Delta^2 y_0}{2} u_x(u_x - 1).$$

A tabela de diferenças finitas é

i	x_i	y_i	Δy_i	$\Delta^2 y_i$
0	0,0	1,0000	0,1232	0,2848
1	0,2	1,1232	0,4080	
2	0,4	1,5312		

e a variável $u_x = \dfrac{x - x_0}{h} = \dfrac{0{,}1 - 0{,}0}{0{,}2} \leadsto u_x = 0{,}5$. Deste modo,

$$P_2(0{,}1) = 1{,}0000 + 0{,}1232(0{,}5) + \frac{0{,}2848}{2}(0{,}5)(0{,}5 - 1) \leadsto P_2(0{,}1) = 1{,}0260.$$

Cálculo de $T_2(0{,}1)$:

$$f(x) = 2x^4 + 3x^2 + 1,\ f'(x) = 8x^3 + 6x,\ f''(x) = 24x^2 + 6,\ f'''(x) = 48x \leadsto \xi = 0{,}4.$$

Logo,

$$T_2(x) = \frac{f'''(\xi)}{3!}(x - x_0)(x - x_1)(x - x_2) \text{ e}$$

$$T_2(0{,}1) = \frac{48(0{,}4)}{6}(0{,}1 - 0{,}0)(0{,}1 - 0{,}2)(0{,}1 - 0{,}4) \rightsquigarrow T_2(0{,}1) = 0{,}0096.$$

Pode ser verificado que, de fato, neste exemplo, (3.13) fornece a cota máxima do erro de truncamento, pois o erro real cometido foi

$$|f(0{,}1) - P_2(0{,}1)| = |1{,}0302 - 1{,}0260| = 0{,}0042 < T_2(0{,}1).$$

∎

Apesar de (3.13) ser de maior interesse na análise teórica da interpolação, ela mostra claramente que o erro de truncamento é diretamente proporcional ao produto das distâncias entre o valor interpolado e os pontos-base. Portanto, os pontos escolhidos para construir o polinômio interpolador devem ser os mais próximos do ponto a ser interpolado (ver Seção 3.5), conforme será mostrado no próximo exemplo.

Exemplo 3.20 Seja a tabela da função $f(x) = e^x - x^2 - x$

x	1,1	1,4	1,9	2,1	2,5	3,0	3,2
y	0,6942	0,6952	1,1759	1,6562	3,4325	8,0855	11,0925

Para calcular $P_2(2{,}2)$, são necessários 3 pontos. A fim de garantir que não seja efetuada uma extrapolação, devem ser usados os pontos de abscissas $x = 2{,}1$ e $x = 2{,}5$. O terceiro ponto pode ser escolhido entre $x_a = 1{,}9$ e $x_b = 3{,}0$. Como os pontos não são igualmente espaçados, o método de Lagrange ou de Newton deve ser utilizado.

Cálculo de $P_{2,a}(2{,}2)$ pelo método de Newton utilizando os pontos

x	1,9	2,1	2,5
y	1,1759	1,6562	3,4325

Por (3.10) com $n = 2$,

$$P_2(x) = y_0 + \Delta y_0(x - x_0) + \Delta^2 y_0(x - x_0)(x - x_1).$$

A tabela de diferenças divididas é

i	x_i	y_i	Δy_i	$\Delta^2 y_i$
0	1,9	1,1759	2,4015	3,3988
1	2,1	1,6562	4,4408	
2	2,5	3,4325		

$$P_{2,a}(2{,}2) = 1{,}1759 + 2{,}4015(2{,}2 - 1{,}9) + 3{,}3988(2{,}2 - 1{,}9)(2{,}2 - 2{,}1) \rightsquigarrow$$

$$P_{2,a}(2{,}2) = 1{,}9983.$$

O erro de truncamento é, por (3.13),

$$T_{2,a}(x) = \frac{f'''(\xi)}{3!}(x-x_0)(x-x_1)(x-x_2), \text{ para algum } \xi,\ x_0 < \xi < x_2.$$

Como $f'''(x) = e^x$, então para $\xi = 2{,}5$, $|T_{2,a}(2{,}2)|$ é a cota máxima do erro de truncamento

$$T_{2,a}(2{,}2) = \frac{e^{2,5}}{6}(2{,}2-1{,}9)(2{,}2-2{,}1)(2{,}2-2{,}5) \rightsquigarrow T_{2,a}(2{,}2) = -0{,}0183,$$

onde o sinal negativo indica que a interpolação foi por excesso, isto é, $P_{2,a}(2{,}2) > f(2{,}2)$. O erro real cometido é

$$|f(2{,}2) - P_{2,a}(2{,}2)| = |1{,}9850 - 1{,}9983| = 0{,}0133 < |T_{2,a}(2{,}2)|.$$

Cálculo de $P_{2,b}(2{,}2)$ pelo método de Newton utilizando os pontos

x	2,1	2,5	3,0
y	1,6562	3,4325	8,0855

A tabela de diferenças divididas é

i	x_i	y_i	Δy_i	$\Delta^2 y_i$
0	2,1	1,6562	4,4408	5,4058
1	2,5	3,4325	9,3060	
2	3,0	8,0855		

$$P_{2,b}(2{,}2) = 1{,}6562 + 4{,}4408(2{,}2-2{,}1) + 5{,}4058(2{,}2-2{,}1)(2{,}2-2{,}5) \rightsquigarrow$$

$P_{2,b}(2{,}2) = 1{,}9381$, e para $\xi = 3{,}0$, tem-se

$$T_{2,b}(2{,}2) = \frac{e^{3,0}}{6}(2{,}2-2{,}1)(2{,}2-2{,}5)(2{,}2-3{,}0) \rightsquigarrow T_{2,b}(2{,}2) = 0{,}0803.$$

Neste caso, o valor positivo do erro de truncamento indica que a interpolação foi por falta, ou seja, $P_{2,b}(2{,}2) < f(2{,}2)$. O erro real é

$$|f(2{,}2) - P_{2,b}(2{,}2)| = |1{,}9850 - 1{,}9381| = 0{,}0469 < |T_{2,b}(2{,}2)|.$$

Conseqüentemente, como o ponto-base $x_a = 1{,}9$ está mais próximo do valor interpolado $z = 2{,}2$ do que $x_b = 3{,}0$, tem-se que

$$|f(2{,}2) - P_{2,a}(2{,}2)| < |f(2{,}2) - P_{2,b}(2{,}2)| \text{ e } |T_{2,a}(2{,}2)| < |T_{2,b}(2{,}2)|. \quad \blacksquare$$

3.7 Comparação das complexidades

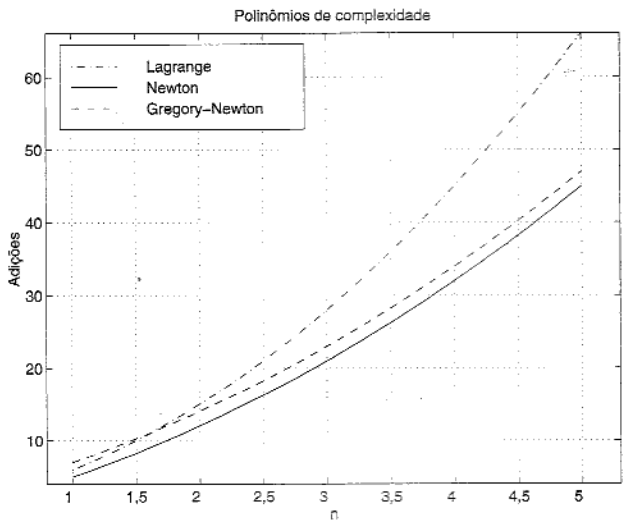
A Tabela 3.5 apresenta uma compilação das Tabelas 3.2, 3.3 e 3.4 referentes às complexidades dos algoritmos de interpolação utilizando polinômios de Lagrange, Newton e Gregory-Newton, respectivamente.

Tabela 3.5 Complexidade dos algoritmos de interpolação.

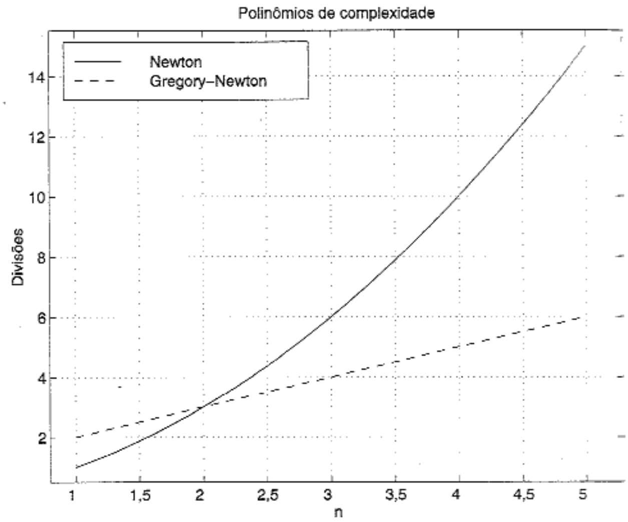
(n: grau do polinômio interpolador.)

Polinômio	Adições	Multiplicações	Divisões
Lagrange	$2n^2+3n+1$	$2n^2+3n+1$	$n+1$
Newton	n^2+4n	n	$\frac{1}{2}n^2+\frac{1}{2}n$
Gregory-Newton	n^2+4n+2	n	$n+1$

A Figura 3.5 exibe os gráficos dos polinômios de complexidade da interpolação para as operações de adição (a) e divisão (b).



(a) Adição.



(b) Divisão.

Figura 3.5 Valores dos polinômios de complexidade para interpolação.

Pela Figura 3.5(a), os polinômios de Newton são os que apresentam menor complexidade para a operação de adição. No que diz respeito à operação de divisão, as complexidades dos algoritmos de Lagrange e Gregory-Newton são lineares, enquanto o de Newton é quadrático. Entretanto, pela Figura 3.5(b), o polinômio de Newton é o mais indicado quando $n \leq 2$. É óbvio, pela Tabela 3.5, que os polinômios de Lagrange são os menos eficientes em termos da operação de multiplicação. Resumindo, para polinômios de grau $n \leq 2$ é preferível utilizar a fórmula de Newton; caso contrário, um polinômio de Gregory-Newton é o mais indicado se os pontos forem igualmente espaçados.