

# Interpolação Polinomial

## (6-9-2004)

**Carlos A. de Moura**

*demoura@ime.uerj.br*

Instituto de Matemática e Estatística, UERJ

## 1 Introdução

As funções polinomiais são, num certo sentido, **naturais, intuitivas**, por só envolver seu cálculo as “*quatro operações*”. Mas é exatamente este fato que lhes passa uma importante característica computacional: para avaliar um polinômio em qualquer ponto da reta real, o computador utiliza apenas programas já *embutidos* na sua estrutura (*hardware*), não sendo envolvidos aplicativos (*software*), os quais podem ser mais ou menos eficientes, dependendo de detalhes na sua implementação.

A expressão para um polinômio é em geral escrita na forma

$$P(s) := \sum_{j=0}^N a_j s^j, \quad (1)$$

onde  $N$  é dito o **grau** do polinômio e  $a_j$ ,  $j = 1, 2, \dots, N$  são os coeficientes. A notação da somatória significa

$$\sum_{j=0}^N a_j s^j := a_0 + a_1 s + a_2 s^2 + a_3 s^3 + \dots + a_N s^N,$$

o que sugere, também de forma natural, um algoritmo de cálculo, o qual passamos a descrever<sup>1</sup>.

$$\begin{array}{l}
 \text{passo 0} \quad p \leftarrow a_0 \\
 \text{passo 1} \quad \left| \begin{array}{l} v \leftarrow s \\ c \leftarrow a_1 \\ p \leftarrow p + cv \end{array} \right. \quad \text{passo 2} \quad \left| \begin{array}{l} v \leftarrow vs \\ c \leftarrow a_2 \\ p \leftarrow p + cv \end{array} \right. \quad \text{passo 3} \quad \left| \begin{array}{l} v \leftarrow vs \\ c \leftarrow a_3 \\ p \leftarrow p + cv \end{array} \right.
 \end{array}$$

O algoritmo segue até o passo  $N$ , quando a variável  $p$  estará armazenando o valor procurado.

Podemos então simplificar (ou automatizar) esta expressão, reescrevendo-a na forma

$$\begin{array}{l}
 \text{entrada: } N \text{ (inteiro), } \{a_j, j = 1, 2, \dots, N\} \text{ (reais)} \\
 \\
 \begin{array}{l}
 \text{passo} \leftarrow 0 \quad p \leftarrow a_{\text{passo}} \quad v \leftarrow 1 \\
 \text{enquanto } \text{passo} \leq N \text{ execute:} \\
 \quad \left| \begin{array}{l} \text{passo} \leftarrow \text{passo} + 1 \\ v \leftarrow vs \\ c \leftarrow a_{\text{passo}} \\ p \leftarrow p + cv \end{array} \right. \\
 \text{saída: } P(s) \leftarrow p
 \end{array}
 \end{array}$$

Como dissemos acima, este algoritmo parece natural, poderíamos mesmo dizer que não se procuraria uma outra alternativa, já tão incorporado este caminho está na nossa rotina de cálculo. Observemos que ele demanda em cada passo<sup>2</sup> uma adição e duas multiplicações, ou seja, requer um total de  $N$  adições e  $2N - 1$  multiplicações, valor que não aparenta poder ser melhorado. (A variável  $\text{passo}$  não está sendo considerada para a presente estimativa. Primeiro, ela é um artifício de programação; segundo, ela é automatizada internamente no sistema computacional, não afetando o desempenho.)

Se em vez de escrever  $P$  na forma da expressão (1), começando com as potências mais baixas, fizermos o contrário, iniciando com o termo de maior potência, obtemos, agora denotando por  $P_N$  o polinômio genérico de grau  $N$ :

$$P_0(s) = a_0 ; \quad P_1(s) = a_1s + a_0 ; \quad P_2(s) = (a_2s + a_1)s + a_0 ;$$

---

<sup>1</sup>A notação informal que usamos deixa implícito que os dados à direita da seta são conhecidos do (ou fornecidos ao) computador.

<sup>2</sup>Exceto no primeiro passo, que não pede o produto.

$$P_3(s) = [(a_3s + a_2)s + a_1]s + a_0 \quad , \quad P_4(s) = \{[(a_4s + a_3)s + a_2]s + a_1\}s + a_0 \quad ,$$

o que sugere o esquema geral

**entrada:**  $N$  (*inteiro*),  $\{a_j, j = 1, 2, \dots, N\}$  (*reais*)

passo $\leftarrow N$	$p \leftarrow a_N$
enquanto passo $> 0$ execute:	$c \leftarrow ps$ $\text{passo} \leftarrow \text{passo} - 1$ $p \leftarrow c + a_{\text{passo}}$

**saída:**  $P(s) \leftarrow p$

O mesmo exame feito anteriormente nos informa que este algoritmo reduziu o total de multiplicações à metade do que era anteriormente requerido, enquanto o total de adições se manteve. Trata-se de uma economia significativa, tendo em vista que a operação de multiplicação demanda muito mais tempo que a de adição.

Este esquema computacional é conhecido como a **fórmula de Horner**.

## 2 Interpolação

Considere os seguintes exemplos.

**Dados do censo populacional** Os órgãos demográficos realizam censos populacionais periodicamente, digamos que correspondam aos instantes de tempo  $t_1 < t_2 < \dots < t_N$ . Suponhamos que se necessitam informações sobre os dados, mas agora para instantes intermediários, digamos  $\hat{t} \in (t_j, t_{j+1})$ . Como variaram esses dados durante o intervalo? Observe que temos definitivamente de recorrer ao estudo teórico, pois não se pode mais realizar o censo no instante  $\hat{t}$ , que já é passado.

**Levantamento de dados meteorológicos** Em uma dada região terrestre coletam-se alguns dados meteorológicos em uma rede de pontos onde estão disponíveis aparelhos de medição. Estes dados variam de ponto para ponto, o que se poderia deduzir sobre seus valores em alguns dos locais onde não estão localizadas essas estações?

Em ambas as situações – e em muitas outras – a alternativa é encontrar uma função que se comporte como *deveria se comportar* a função em estudo,

e usar os valores daquela como se fossem os desta. Esta técnica é chamada de *interpolação*, a função escolhida – que sabemos calcular – se chama **interpolante** ou **interpoladora** e a original, **interpolada**. Sua escolha é feita a partir de determinadas propriedades como rapidez ou facilidade do cálculo de seus valores, conhecimento *a priori* de que seu perfil se assemelha ao da interpolada, etc.

Pelas características já citadas e algumas outras, como o conhecimento aprofundado de suas propriedades teóricas, a escolha de polinômios como interpolantes se justifica em muitas situações. É a chamada *interpolação polinomial*. Este problema pode ser formulado em um contexto mais abstrato na forma:

Dados  $N + 1$  reais distintos  $x_j$  e os correspondentes  $N + 1$  valores  $y_j = f(x_j)$  de uma dada função  $f : \mathbb{R} \rightarrow \mathbb{R}$ , é possível determinar um (único) polinômio de grau  $N$ ,  $P = P_N$  para o qual se tenha

$$P(x_j) = y_j, \quad j = 1, 2, \dots, N + 1 ?$$

**Exercício** Existe mesmo algum interesse (e qual?) em se ter unicidade nesse problema?

Demonstra-se de forma razoavelmente simples a **unicidade**. De fato, suponhamos que dois polinômios  $P_N, Q_N$ , ambos de grau  $N$ , satisfaçam

$$P_N(x_j) = Q_N(x_j) = y_j, \quad j = 1, 2, \dots, N + 1$$

Conclui-se que o polinômio  $\Pi_N := P_N - Q_N$  possui  $N + 1$  raízes. Sendo ele de grau  $N$ , deduz-se ser ele identicamente nulo, donde a unicidade.

E para provar a existência? Uma demonstração se baseia na matriz (ou no determinante) de van der Monde, mas preferimos a que segue, por ser construtiva.

**Exercício** Pesquise essa demonstração (em qualquer livro de Cálculo Numérico.)

**Caso**  $N = 1$  Tomemos o caso onde conhecemos dois valores da função, sejam  $A := f(a)$  e  $B := f(b)$ . Trata-se de determinar a função linear

cujos gráficos passam pelos pontos  $(a, A)$ ,  $(b, B)$ . Este problema pode ser reduzido ao problema (aparentemente mais complicado) de encontrar duas funções  $\phi_a$ ,  $\phi_b$  satisfazendo

$$\phi_a(x) := \begin{cases} 1 & \text{se } x = a \\ 0 & \text{se } x = b \end{cases} \quad \phi_b(x) := \begin{cases} 1 & \text{se } x = b \\ 0 & \text{se } x = a \end{cases},$$

de forma que a interpolante que buscamos é necessariamente

$$P(x) = A \phi_a(x) + B \phi_b(x).$$

Evidentemente devem elas ter a forma

$$\phi_a(x) = \Lambda(x - b), \quad \phi_b(x) = \Gamma(x - a),$$

e, portanto,

$$\phi_a(x) = \frac{(x - b)}{(a - b)}, \quad \phi_b(x) = \frac{(x - a)}{(b - a)}.$$

Para a interpolação de ordem dois, com funções quadráticas, procuram-se

$$\phi_a(x) := \begin{cases} 1, & x = a \\ 0, & x = b \\ 0, & x = c \end{cases} \quad \phi_b(x) := \begin{cases} 1, & x = b \\ 0, & x = a \\ 0, & x = c \end{cases} \quad \phi_c(x) := \begin{cases} 1, & x = c \\ 0, & x = a \\ 0, & x = b \end{cases}$$

cujas soluções são

$$\phi_a(x) = \frac{(x - b)(x - c)}{(a - b)(a - c)}, \quad \phi_b(x) = \frac{(x - a)(x - c)}{(b - a)(b - c)}, \quad \phi_c(x) = \frac{(x - a)(x - b)}{(c - a)(c - b)}.$$

**Exercício** Escrever o caso geral.

Esta expressão recebe na literatura o nome de **fórmula de Lagrange para a interpolação**.