Tópico 7 - Interpolação Polinomial

João B. Florindo

Instituto de Matemática, Estatística e Computação Científica
Universidade Estadual de Campinas - Brasil
jbflorindo@ime.unicamp.br

Outline

Introdução

Sejam os seguintes dados tabelados:

Temperatura (C)	20	25	30	35
Calor específico	0.907	0.852	0.826	0.818

Agora supondo que queremos encontrar o calor específico em 27C. Ou a temperatura em que o CE é 0.823.

Este é o típico problema resolvido pela interpolação

Interpolar uma função $f(x)$ consiste em encontrar os melhores parâmetros para uma outra função $g(x)$ tal que esta última aproxime a primeira em algum sentido

Aplicação: valores em pontos não tabelados, operações como diferenciação ou integração de $f(x)$ fornecida em apenas alguns pontos (tabelados) ou com expressão muito complexa, etc.

Definição

Dados os pontos distintos $x_0, x_1, \ldots, x_n$ (nós de interpolação) e os valores de $f(x)$ nestes pontos: $f(x_0), f(x_1), \ldots, f(x_n)$. Interpolar $f(x)$ é encontrar $g(x)$ tal que

$$\begin{cases} g(x_0) = f(x_0) \\ g(x_1) = f(x_1) \\ \vdots \\ g(x_n) = f(x_n) \end{cases}$$

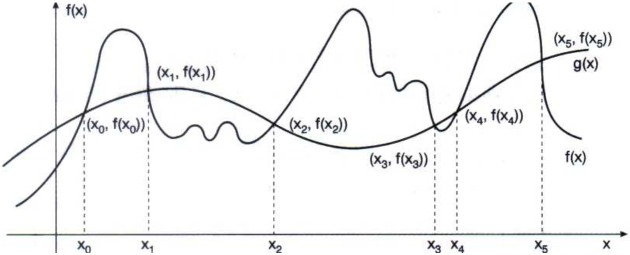

Figura: Fonte: Ruggiero&Lopes

Definição

Note que a interpolação pode usar qualquer tipo de função como exponencial, trigonométrica, etc. Existem também outras formas de interpolação, como Taylor por exemplo, que não satisfazem a restrições acima.

Vamos focar aqui no caso em que $g(x)$ é polinomial.

Introdução

- Polinômios são estruturas com manipulação algébrica simples
- Derivadas e integrais simples: também polinomiais
- São usados como elemento básico de aproximação em virtualmente toda a análise numérica
- Teorema de Weierstrass: Qualquer função f contínua em $[a, b]$ pode ser aproximada tanto quanto se queira de um polinômio, ou seja, para qualquer $\epsilon > 0$, existe um polinômio $P(x)$ tal que

$$|f(x) - P(x)| < \epsilon, \forall x \in [a, b].$$

- Já conhecemos bem a aproximação polinomial de Taylor, porém embora esta se aproxime tanto quanto desejado de um ponto específico da função, o mesmo não ocorre quando nos afastamos do ponto de referência, não sendo então recomendada quando temos informação em vários pontos
- A interpolação é o método mais simples e difundido para tal aproximação

Existência e Unicidade

- O problema da interpolação polinomial consiste em:

Dados $n+1$ pontos distintos $x_0, \dots, x_n$ no eixo dos reais e $f(x)$ uma função real definida em um intervalo $[a, b]$ que contém $x_0, \dots, x_n$. Dizemos que um polinômio $p_n(x)$ de grau $\leq n$ interpola $f(x)$ em $x_0, \dots, x_n$ se:

$$p_n(x_k) = f(x_k), i = 0, \dots, n$$

Perguntas importantes

Existe sempre este polinômio $p_n(x)$? Se sim, é único?

- Para mostrar a unicidade do polinômio interpolador usaremos as restrições da interpolação

Existência e Unicidade

Usando a forma clássica para polinômios:

$$p_n(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_n x^n$$

A condição $p_n(x_k) = f(x_k)$ gera o seguinte sistema linear:

$$\begin{cases} a_0 + a_1 x_0 + a_2 x_0^2 + \cdots + a_n x_0^n = f(x_0) \\ a_0 + a_1 x_1 + a_2 x_1^2 + \cdots + a_n x_1^n = f(x_1) \\ \vdots \\ a_0 + a_1 x_n + a_2 x_n^2 + \cdots + a_n x_n^n = f(x_n) \end{cases}$$

que é um sistema $(n+1) \times (n+1)$ e com matriz A

$$\begin{bmatrix} 1 & x_0 & x_0^2 & \ldots & x_0^n \\ 1 & x_1 & x_1^2 & \ldots & x_1^n \\ \vdots & & & & \\ 1 & x_n & x_n^2 & \ldots & x_n^n \end{bmatrix}$$

Esta é chamada matriz de Vandermonde e o sistema respectivo admite solução única

Existência e Unicidade

E assim temos garantido o seguinte teorema de existência e unicidade:

Teorema 1

Dada uma função real $f(x)$ definida em $n+1$ pontos distintos $x_0, \ldots, x_n$, então existe **exatamente um** polinômio $p_n(x)$ de grau $\leq n$ que interpola $f(x)$ em $x_0, \ldots, x_n$, ou seja, tal que $p_n(x_k) = f(x_k)$, dado que $x_k \neq x_j$ sempre que $k \neq j$.

Veremos três formas de se obter este polinômio: resolvendo o sistema linear anterior, por Lagrange e Newton.

Sistema Linear - Exemplo

Polinômio com $n \leq 2$ que interpola os dados da tabela:

x	-1	0	2
f(x)	4	1	-1

$$p_2(x) = a_0 + a_1 x + a_2 x^2$$

$p_2(x_0) = f(x_0) \Leftrightarrow a_0 - a_1 + a_2 = 4$
$p_2(x_1) = f(x_1) \Leftrightarrow a_0 = 1$
$p_2(x_2) = f(x_2) \Leftrightarrow a_0 + 2a_1 + 4a_2 = -1$

Resolvendo o sistema obtemos $[a_0 \quad a_1 \quad a_2]^T = [1 \quad -7/3 \quad 2/3]^T$.

Logo o polinômio interpolador será $p_2(x) = 1 - \frac{7}{3}x + \frac{2}{3}x^2$.

O problema é que esta solução nem sempre é viável, pois a matriz de Vandermonde pode ser mal-condicionada.

Outline

Forma de Lagrange

Dados $(n+1)$ pontos distintos $x_0, x_1, \dots, x_n$ e $y_i = f(x_i)$, na forma de Lagrange o polinômio $p_n(x)$ é obtido de modo que

$$p_n(x) = y_0 L_0(x) + y_1 L_1(x) + \cdots + y_n L_n(x),$$

em que $L_k(x)$ tem grau n. Para cada i devemos satisfazer $p_n(x_i) = y_i$, ou seja,

$$p_n(x_i) = y_0 L_0(x_i) + y_1 L_1(x_i) + \cdots + y_n L_n(x_i) = y_i.$$

Uma forma simples de satisfazer isso é exigir que $L_k(x_i)$ seja 1 se $k = i$ e 0 se $k \neq i$ e para isso definimos $L_k(x)$ como

$$L_k(x) = \frac{(x - x_0)(x - x_1)\dots(x - x_{k-1})(x - x_{k+1})\dots(x - x_n)}{(x_k - x_0)(x_k - x_1)\dots(x_k - x_{k-1})(x_k - x_{k+1})\dots(x_k - x_n)}$$

Como a função $L_k(x)$ é o produto de n fatores lineares, então $L_k(x)$ tem grau n e $p_n(x)$ obtido desta forma é um polinômio de grau $\leq n$

Forma de Lagrange

Forma de Lagrange

$$p_n(x) = y_0 L_0(x) + y_1 L_1(x) + \cdots + y_n L_n(x)$$

com

$$L_k(x) = \prod_{\substack{i=0 \\ i \neq k}}^{n} \frac{x - x_i}{x_k - x_i}, k = 0, \ldots, n$$

Forma de Lagrange - Exemplo

x	-1	0	2
f(x)	4	1	-1

$$p_2(x) = y_0 L_0(x) + y_1 L_1(x) + y_2 L_2(x)$$

em que

$$L_0(x) = \frac{(x - x_1)(x - x_2)}{(x_0 - x_1)(x_0 - x_2)} = \frac{(x - 0)(x - 2)}{(-1 - 0)(-1 - 2)} = \frac{x^2 - 2x}{3}$$

$$L_1(x) = \frac{(x - x_0)(x - x_2)}{(x_1 - x_0)(x_1 - x_2)} = \frac{(x + 1)(x - 2)}{(0 + 1)(0 - 2)} = \frac{x^2 - x - 2}{-2}$$

$$L_2(x) = \frac{(x - x_0)(x - x_1)}{(x_2 - x_0)(x_2 - x_1)} = \frac{(x + 1)(x - 0)}{(2 + 1)(2 - 0)} = \frac{x^2 + x}{6}$$

Forma de Lagrange:

$$p_2(x) = 4\left(\frac{x^2 - 2x}{3}\right) + 1\left(\frac{x^2 - x - 2}{-2}\right) + (-1)\left(\frac{x^2 + x}{6}\right)$$

Forma de Lagrange - Exemplo

Rearranjando:

$$p_2(x) = 1 - \frac{7}{3}x + \frac{2}{3}x^2$$

que no caso é idêntica à forma que obtemos resolvendo o SL.

NOTA

Isso era esperado lembrando que o polinômio é ÚNICO. O que muda é apenas a forma de apresentação!

Forma de Lagrange

- A avaliação de um polinômio na forma de Lagrange em um ponto x exige $(2n+1)$ adições/subtrações e $(2n+1)$ multiplicações/divisões, além dos denominadores
- Na forma aninhada apenas n adições e n multiplicações são necessárias
- Outro problema é a identificação do número ideal de pontos de interpolação
- É comum para isso ir aumentando o número de pontos, porém na fórmula de Lagrange, quando se calcula $p_k(x)$ não se aproveita nada de $p_{k-1}(x)$ já calculado
- Neste caso, a forma de Newton vista mais adiante é mais eficiente

Outline

Forma de Newton

- Representação clássica de um polinômio $p(x)$ de grau n:

$$p(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_n x^n$$

em que $a_0, a_1, \ldots, a_n$ são chamados coeficientes e $a_0 \neq 0$.

- A forma acima pode acarretar perda de significância
- Uma potencial solução é usar a forma deslocada:

$$p(x) = a_0 + a_1(x - c) + a_2(x - c)^2 + \cdots + a_n(x - c)^n$$

Recomenda-se usar c dentro de um intervalo limitado $[a, b]$ dentro do qual o polinômio está sendo analisado.

Forma de Newton

- A forma deslocada é obtida pela expansão de Taylor de $p(x)$, tal que

$$a_i = p^{(i)}(c)/i!$$

- Uma generalização da forma deslocada é a forma de Newton:

$$p(x) = a_0 + a_1(x - c_1) + a_2(x - c_1)(x - c_2) + \cdots + a_n(x - c_1)(x - c_2)\ldots(x - c_n)$$

- Uma representação mais eficiente para cálculos numéricos da forma de Newton é a forma aninhada:

$$p(x) = a_0 + (x - c_1)\{a_1 + (x - c_2)[a_2 + (x - c_3) < a_3 + \ldots \\ + (x - c_{n-1})(a_{n-1} + (x - c_n)a_n) \cdots >]\}$$

Forma de Newton

- A forma de Newton considera os pontos de interpolação $x_0, \ldots, x_{n-1}$ como centros:

$$p_n(x) = A_0 + A_1(x - x_0) + A_2(x - x_0)(x - x_1) + \cdots + A_n(x - x_0)\ldots(x - x_{n-1})$$

Pode-se mostrar então que nesta forma o polinômio pode ser construído iterativamente:

$$p_k(x) = p_{k-1}(x) + A_k(x - x_0)\ldots(x - x_{k-1})$$

Os coeficientes dependem apenas do valor de $f(x)$ em $x_0, \ldots, x_k$ e são chamados de k-**ésima diferença dividida de** $f(x)$ **nos pontos** $x_0, \ldots, x_k$, denotados por $f[x_0, \ldots, x_k]$. A forma de Newton pode ser resumida em

$$p_n(x) = \sum_{i=0}^{n} f[x_0, \ldots, x_i] \prod_{j=0}^{i-1}(x - x_j)$$

Diferenças Divididas

- As diferenças divididas da forma de Newton podem ser calculadas por:

$$f[x_0, \ldots, x_k] = \frac{f[x_1, \ldots, x_k] - f[x_0, \ldots, x_{k-1}]}{x_k - x_0}$$

partindo de $f[x_0] = f(x_0)$ e $f[x_1] = f(x_1)$.

Diferenças Divididas - Exemplo

x	Ordem 0	Ordem 1	Ordem 2	Ordem 3	...	Ordem n
x_0	$f[x_0]$					
		$f[x_0, x_1]$				
x_1	$f[x_1]$		$f[x_0, x_1, x_2]$			
		$f[x_1, x_2]$		$f[x_0, x_1, x_2, x_3]$		
x_2	$f[x_2]$		$f[x_1, x_2, x_3]$			
		$f[x_2, x_3]$		$f[x_1, x_2, x_3, x_4]$		
x_3	$f[x_3]$		$f[x_2, x_3, x_4]$			$f[x_0, x_1, x_2, \ldots, x_n]$
		$f[x_3, x_4]$				
x_4	$f[x_4]$			$f[x_{n-3}, x_{n-2}, x_{n-1}, x_n]$		
			$f[x_{n-2}, x_{n-1}, x_n]$			
		$f[x_{n-1}, x_n]$				
x_n	$f[x_n]$					

x	-1	0	1	2	3
f(x)	1	1	0	-1	-2

Diferenças divididas:

$$f[x_i] = f(x_i)$$

$$f[x_0, x_1] = \frac{f[x_1]-f[x_0]}{x_1-x_0} = \frac{1-1}{1} = 0$$

$$f[x_1, x_2] = \frac{f[x_2]-f[x_1]}{x_2-x_1} = \frac{0-1}{1-0} = -1$$

$$\vdots$$

$$f[x_0, x_1, x_2] = \frac{f[x_1,x_2]-f[x_0,x_1]}{x_2-x_0} = \frac{-1-0}{1+1} = \frac{-1}{2}$$

$$f[x_1, x_2, x_3] = \frac{f[x_2,x_3]-f[x_1,x_2]}{x_3-x_1} = \frac{-1+1}{2-0} = 0$$

Diferenças Divididas - Exemplo

⋮

$$f[x_0, x_1, x_2, x_3] = \frac{f[x_1,x_2,x_3]-f[x_0,x_1,x_2]}{x_3-x_0} = \frac{0+1/2}{2+1} = \frac{1}{6}$$

⋮

Sumarizando em tabela:

x	Ordem 0	Ordem 1	Ordem 2	Ordem 3	Ordem 4
-1	1				
		0			
0	1		$-\frac{1}{2}$		
		-1		$\frac{1}{6}$	
1	0		0		$-\frac{1}{24}$
		-1		0	
2	-1		0		
		-1			
3	-2				

Diferenças Divididas

Algoritmo 2: Tabela de diferenças divididas

Dadas as duas primeiras colunas da tabela com $x_0, \ldots, x_n$ e $f[x_0], \ldots, f[x_n]$:

for $k = 1$ **to** n **do**
  for $i = 0$ **to** $n - k$ **do**
    $f[x_i, \ldots, x_{i+k}] \leftarrow \frac{f[x_{i+1},\ldots,x_{i+k}] - f[x_i,\ldots,x_{i+k-1}]}{x_{i+k} - x_i}$
  end for
end for

Forma de Newton-Construção

Seja $f(x)$ contínua e com derivadas até ordem n contínuas em um intervalo $[a, b]$, o qual será particionado em $(n+1)$ pontos:

$$a = x_0, x_1, x_2, \ldots, x_n = b$$

Vamos inicialmente obter um polinômio de grau zero que interpola $f(x)$ em $x = x_0$, de modo que $p_0(x_0) = f(x_0) = f[x_0]$. Assim, $\forall x \in [a, b], x \neq x_0$:

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

$$\Rightarrow (x - x_0) f[x_0, x] = f(x) - f(x_0) \Rightarrow$$

$$\Rightarrow \quad f(x) = \underbrace{f(x_0)}_{p_0(x)} + \underbrace{(x - x_0) f[x_0, x]}_{E_0(x)}$$

em que $E_0(x)$ é o erro da interpolação.

Forma de Newton-Construção

Similarmente, vamos obter um polinômio de grau ≤ 1 interpolante em x_0, x_1:

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

$$\frac{\frac{f(x)-f(x_0)}{x-x_0} - f[x_1, x_0]}{(x - x_1)} = \frac{f(x) - f(x_0) - (x - x_0)f[x_1, x_0]}{(x - x_1)(x - x_0)} \Rightarrow$$

$$\Rightarrow f[x_0, x_1, x] = \frac{f(x) - f(x_0) - (x - x_0)f[x_1, x_0]}{(x - x_0)(x - x_1)} \Rightarrow$$

$$\Rightarrow \quad f(x) = \underbrace{f(x_0) + (x - x_0)f[x_1, x_0]}_{p_1(x)} + \underbrace{(x - x_0)(x - x_1)f[x_0, x_1, x]}_{E_1(x)}$$

Verificando nos pontos de interpolação:

$$p_1(x_0) = f(x_0)$$
$$p_1(x_1) = f(x_0) + (x - x_0)\frac{f(x_1)-f(x_0)}{x_1-x_0} = f(x_1)$$

Forma de Newton-Construção

Obtemos agora o polinômio de grau ≤ 2 interpolante em x_0, x_1, x_2:

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

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

$$= \frac{\frac{\frac{f(x)-f(x_0)}{(x-x_0)}-f[x_1,x_0]}{(x-x_1)} - f[x_2, x_1, x_0]}{(x - x_2)} =$$

$$= \frac{f(x) - f(x_0) - (x - x_0)f[x_1, x_0] - (x - x_0)(x - x_1)f[x_2, x_1, x_0]}{(x - x_0)(x - x_1)(x - x_2)} \Rightarrow$$

$$\Rightarrow f(x) = \underbrace{f(x_0) + (x - x_0)f[x_0, x_1] + (x - x_0)(x - x_1)f[x_0, x_1, x_2]}_{p_2(x)} +$$

Forma de Newton-Construção

$$+\underbrace{(x-x_0)(x-x_1)(x-x_2)f[x_0,x_1,x_2,x]}_{E_2(x)}.$$

Usando o mesmo raciocínio sucessivamente obtemos a forma de Newton do polinômio de grau $\leq n$ que interpola $f(x)$ em $x_0,\ldots,x_n$:

$$p_n(x)=f(x_0)+(x-x_0)f[x_0,x_1]+(x-x_0)(x-x_1)f[x_0,x_1,x_2]+\cdots+$$
$$+\cdots+(x-x_0)(x-x_1)\ldots(x-x_{n-1})f[x_0,x_1,\ldots,x_n]$$

com erro

$$E_n(x)=(x-x_0)(x-x_1)\ldots(x-x_n)f[x_0,x_1,\ldots,x_nx,x]$$

Note que nos pontos tabelados:

$$f(x_k)=p_n(x_k)+\underbrace{E_n(x_k)}_{=0}=p_n(x_k)$$

Forma de Newton-Exemplo

x	-1	0	2
f(x)	4	1	-1

x	Ordem 0	Ordem 1	Ordem 2
-1	4		
		-3	
0	1		$\frac{2}{3}$
		-1	
2	-1		

$$p_2(x) = f(x_0) + (x - x_0)f[x_0, x_1] + (x - x_0)(x - x_1)f[x_0, x_1, x_2] =$$

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

Se desenvolvermos voltamos à forma já obtida $p_2(x) = 1 - \frac{7}{3}x + \frac{2}{3}x^2$, porém é conveniente não desenvolver pois assim evitamos as potências na avaliação numérica de $p_n(x)$ em $x = \alpha$.

Forma de Newton-Exemplo

O número de operações é ainda menor usando-se a forma de Newton com parêntesis encaixados:

$$p_n(x) = f(x_0) + (x - x_0)\{f[x_0, x_1] + (x - x_1)\{f[x_0, x_1, x_2] +$$

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

Forma de Newton-Algoritmo

Algoritmo 3: Coeficientes da fórmula de Newton

Sejam dados $n+1$ pontos distintos $x_0, \ldots, x_n$ e d_i contendo incialmente o valor de $f(x_i)$,

```
for k = 1 to n do
    for i = 0 to n − k do
        d_i ← (d_{i+1} − d_i)/(x_{i+k} − x_i)
    end for
end for
```

Outline

Erro na Interpolação Polinomial

Seja uma função real $f(x)$ definida no intervalo $[a, b]$ e interpolada em $n+1$ pontos distintos $x_0, \ldots, x_n$ neste intervalo pelo polinômio $p_n(x)$ de grau $\leq n$. O erro de interpolação é dado por:

$$E_n(x) = f(x) - p_n(x)$$

Seja agora então $p_{n+1}(x)$ um polinômio de grau $\leq n+1$ e que interpola $f(x)$ nos pontos $x_0, \ldots, x_n$ e no ponto $\overline{x}$ diferente de $x_0, \ldots, x_n$:

$$p_{n+1}(x) = p_n(x) + f[x_0, \ldots, x_n, \overline{x}] \prod_{j=0}^{n} (x - x_j)$$

Portanto:

$$f(\overline{x}) = p_{n+1}(\overline{x}) = p_n(\overline{x}) + f[x_0, \ldots, x_n, \overline{x}] \prod_{j=0}^{n} (\overline{x} - x_j)$$

e assim:

$$E_n(\overline{x}) = f(\overline{x} - p_n(\overline{x})) = f[x_0, \ldots, x_n, \overline{x}] \prod_{j=0}^{n} (\overline{x} - x_j)$$

Erro na Interpolação Polinomial

- A expressão anterior não pode ser calculada diretamente sem conhecer $f(\overline{x})$. Mas podemos estimá-la sabendo que $f[x_0, \ldots, x_n, \overline{x}]$ está bem próximo da $n+1$-ésima derivada de $f(x)$

Teorema 2

Seja $f(x)$ uma função real definida em $[a, b]$ e k vezes diferenciável em (a, b). Se $x_0, \ldots, x_k$ são $k+1$ pontos distintos em $[a, b]$, então existe $\xi \in (a, b)$ tal que:

$$f[x_0, \ldots, x_k] = \frac{f^{(k)}(\xi)}{k!}$$

- Como consequência, temos o teorema a seguir para a estimativa do erro:

Erro na Interpolação Polinomial

Teorema

Seja $f(x)$ uma função real definida em $[a, b]$ e $n+1$ vezes diferenciável em (a, b) e $p_n(x)$ um polinômio de grau $\leq n$ interpolando $f(x)$ em $n+1$ pontos distintos $x_0, \ldots, x_n$ em $[a, b]$. Então para todo $\overline{x} \in [a, b]$ existe $\xi(\overline{x}) \in (a, b)$ tal que

$$E_n(\overline{x}) = f(\overline{x}) - p_n(\overline{x}) = \frac{f^{(n+1)}(\xi)}{(n+1)!} \prod_{j=0}^{n} (\overline{x} - x_j)$$

Este mesmo teorema se aplica também à forma de Lagrange

Erro na Interpolação Polinomial

Forma de Lagrange amplamente usada em diferenciação e integração numérica tornando esta expressão para o erro de grande importância

Note a semelhança com o erro de Taylor, apenas em vez de $(x - x_0)^n$ entra $(x - x_0)(x - x_1) \ldots (x - x_n)$

A real utilidade da expressão anterior existe quando se conhece um limitante para $|f^{(n+1)}(x)|$

Outro ponto é que a função $\psi_{n+1}(x) = \prod_{j=0}^{n}(x - x_j)$ depende totalmente da posição dos pontos de interpolação. Esta função pode ser minimizada pelos pontos de Chebyshev

Erro na Interpolação Polinomial

Para pontos igualmente espaçados, $\psi_{n+1}(x)$ e consequentemente o erro aumenta à medida que se distancia do centro do intervalo

Situação ainda pior ocorre fora do intervalo definido: **extrapolação**

No caso em que um limitante superior para $f^{(n+1)}(x)$ é conhecido temos os corolários a seguir

Corolário

Se $f^{(n+1)}(x)$ for contínua em $[a, b]$:

$$|E_n(x)| = |f(x) - p_n(x)| \leq |(x - x_0)(x - x_1)\dots(x - x_n)| \frac{M_{n+1}}{(n+1)!}$$

em que $M_{n+1} = max_{x \in [a,b]} |f^{(n+1)}(x)|$.

Erro na Interpolação Polinomial

Corolário 2

Se além disso os pontos forem igualmente espaçados:

$$x_1 - x_0 = x_2 - x_1 = \cdots = x_n - x_{n-1} = h,$$

então

$$|f(x) - p_n(x)| < \frac{h^{n+1} M_{n+1}}{4(n+1)},$$

independendo portanto de x.

Exemplo

Fazer interpolação linear de Newton de $f(x) = e^x + x - 1$, obter $f(0.7)$ e analisar o erro.

x	0	0.5	1	1.5	2
f(x)	0.0	1.1487	2.7183	4.9811	8.3890

$$p_1(x) = f(x_0) + (x - x_0)f[x_0, x_1]$$

$$x = 0.7 \in [0.5, 1] \Rightarrow x_0 = 0.5, x_1 = 1$$

$$p_1(x) = 1.1487 + (x - 0.5)\left(\frac{2.7183 - 1.1487}{1 - 0.5}\right) = 1.1487 + (x - 0.5)3.1392$$

Estimativa do erro pelo Corolário 1:

$$|E_1(0.7)| \leq |(0.7 - 0.5)(0.7 - 1)|\frac{M_2}{2}$$

em que $M_2 = \max_{x \in [0.5,1]} |f''(x)| = e^1 = 2.7183$. Portanto:

$$|E_1(0.7)| \leq 0.0815$$

Exemplo

Estimativa do erro pelo Corolário 2:

$$|E_1(x)| \leq \frac{h_2}{8} M_2 = \frac{(0.5)^2}{8}(2.7183) = 0.0850$$

Podemos neste caso calcular o erro exato:

$$|E_1(0.7)| = |f(0.7) - p_1(0.7)| = |1.7137 - 1.7765| = 0.0628,$$

como esperado, menor do que os limitantes obtidos por ambos os corolários.

Erro na Interpolação Polinomial

No caso em que $f(x)$ é fornecida por uma tabela e sua expressão analítica não é conhecida, pode-se aproximar $\frac{M_{n+1}}{(n+1)!}$ pelo máximo em módulo entre todas as diferenças divididas no intervalo $[x_0, x_n]$ até ordem $(n+1)$

Exemplo

x	0.2	0.34	0.4	0.52	0.6	0.72
f(x)	0.16	0.22	0.27	0.29	0.32	0.37

Obter $f(0.47)$ com grau 2 e estimar o erro.

x	Ordem 0	Ordem 1	Ordem 2	Ordem 3
0.2	0.16			
		0.4286		
0.34	0.22		2.0235	
		0.8333		
$x_0 = 0.4$	$\underline{0.27}$		-3.7033	
		$\underline{0.1667}$		$\underline{18.2494}$
$x_1 = 0.52$	0.29		$\underline{1.0415}$	
		0.375		-2.6031
$x_2 = 0.6$	0.32		0.2085	
		0.4167		
0.72	0.37			

Exemplo

Três pontos de interpolação. Como $0.47 \in [0.4, 0.52]$ estes dois pontos deverão ser usados. O terceiro pode ser tanto 0.34 quanto 0.6. Vamos usar $x_0 = 0.4$, $x_1 = 0.52$ e $x_2 = 0.6$.

$$p_2(x) = f(x_0) + (x - x_0)f[x_0, x_1] + (x - x_0)(x - x_1)f[x_0, x_1, x_2] =$$

$$= 0.27 + (x - 0.4)0.1667 + (x - 0.4)(x - 0.52)(1.0415)$$

Finalmente

$$f(0.47) \approx p_2(0.47) = 0.2780$$

com erro

$$|E(0.47)| \approx |(0.47-0.4)(0.47-0.52)(0.47-0.6)||18.2492| \approx 8.303 \times 10^{-3}$$

Outline

Interpolação com Pontos Igualmente Espaçados em Tabelas de Funções

- **Tabelas de função** são tabelas contendo valores de uma função $f(x)$ em pontos regularmente espaçados $x_0, \ldots, x_n$
- O usuário precisa determinar o valor em um ponto qualquer que não esteja listado
- Interpolaçao polinomial inicialmente desenvolvida para este problema
- O uso de pontos igualmente espaçados permite algumas simplificações

Interpolação com Pontos Igualmente Espaçados em Tabelas de Funções

- Seja a função $f(x)$ tabelada em N pontos x_i do intervalo $[a, b]$ com espaçamento h, isto é:

$$x_i = a + ih, i = 0, \ldots, N, \text{ com } N = \frac{b-a}{h}$$

- Aplica-se então uma mudança de variável:

$$s = s(x) = \frac{x - x_0}{h}, \text{ tal que } x = x(s) = x_0 + sh$$

e seguindo a mesma notação:

$$f(x) = f(x_0 + sh) = f_s$$

Interpolação com Pontos Igualmente Espaçados em Tabelas de Funções

- Como o intervalo é uniforme não precisamos mais de diferenças divididas, mas apenas da *tabela de diferenças* ou **diferenças progressivas**:

$$\Delta^i f_s = \begin{cases} f_s & \text{se } i = 0 \\ \Delta(\Delta^{i-1} f_s) = \Delta^{i-1} f_{s+1} - \Delta^{i-1} f_s & \text{se } i > 0 \end{cases}$$

Lema 3

Para qualquer $i \geq 0$:

$$f[x_k, \ldots, x_{k+i}] = \frac{1}{i! h^i} \Delta^i f_k$$

E por fim temos a fórmula de diferenças progressivas de Newton:

$$p_n(x_0 + sh) = \sum^{n} \Delta^i f_0 \binom{s}{i}$$

Outline

Interpolação Inversa

x	x_0	x_1	x_2	$\dots$	x_n
$f(x)$	$f(x_0)$	$f(x_1)$	$f(x_2)$	$\dots$	$f(x_n)$

Dado $\overline{y}$ obter $\overline{x}$ tal que $\overline{y} = f(\overline{x})$

Duas abordagens:

1. Obter polinômio interpolador $p_n(x)$ e encontrar $\overline{x}$ tal que $p_n(\overline{x}) = \overline{y}$.
2. Interpolação inversa: interpolar a função inversa $x = f^{-1}(y)$. Para que isto seja possível, é necessário que $f(x)$ seja inversível e isso ocorre se $f(x)$ for monótona (crescente ou decrescente) no intervalo $[a, b]$.

Exemplo Abordagem 1

x	0.5	0.6	0.7	0.8	0.9	1.0
f(x)	1.65	1.82	2.01	2.23	2.46	2.72

$2 \in [1.82, 2.01]$, vamos portanto interpolar linearmente com $x_0 = 0.6$ e $x_1 = 0.7$:

$$p_1(x) = f(x_0)\frac{x - x_0}{x_0 - x_1} + f(x_1)\frac{x - x_0}{x_1 - x_0} = 1.82\frac{x - 0.7}{-0.1} + 2.01\frac{x - 0.6}{0.1} =$$

$$-1.82x + 12.74 + 20.1x - 12.06 = 1.9x + 0.68 = 2 \Leftrightarrow \overline{x} = \frac{2 - 0.68}{1.9} \approx 0.69$$

Vamos então encontrar $\overline{x}$ tal que $p_1(\overline{x}) = 2$:

$$p_1(\overline{x}) = 2 \Leftrightarrow 1.9\overline{x}$$

Neste caso o erro não pode ser estimado pois o que vimos até agora estima o desvio em $f(x)$ e não em x como no caso

Exemplo Abordagem 2

x	0	0.1	0.2	0.3	0.4	0.5
$y = e^x$	1	1.1052	1.2214	1.3499	1.4918	1.6487

Obter x tal que $e^x = 1.3165$. Vamos a usar a forma de Newton quadrática em y. Diferenças divididas:

y	Ordem 0	Ordem 1	Ordem 2	Ordem 3
1	0			
		0.9506		
1.1052	0.1		-0.4065	
		0.8606		0.1994
1.2214	0.2		-0.3367	
		0.7782		0.1679
1.3499	0.3		-0.2718	
		0.7047		0.1081
1.4918	0.4		-0.2256	
		0.6373		
1.6487	0.5			

Exemplo Abordagem 2

Assim:

$$p_2(y) = g(y_0) + (y - y_0)g[y_0, y_1] + (y - y_0)(y - y_1)g[y_0, y_1, y_2] =$$
$$= 0.2 + (y - 1.2214)0.7782 + (y - 1.2214)(y - 1.3499)(-0.2718)$$
$$\Rightarrow p_2(1.3165) = 0.27487$$

Neste caso, podemos medir o erro $E(y) = f^{-1}(y) - p_n(y) = g(y) - p_n(y)$:

$$|E_2(y)| \leq |(y - y_0)(y - y_1)(y - y_2)| \frac{M_3}{3!}$$

em que $M_3 = \max |g'''(y)|, \forall y \in [y_0, y_2]$. No caso:

$$g'''(y) = \frac{2}{y^3} \Rightarrow M_3 = \frac{2}{(1.2214)^3} = 1.0976$$

Portanto temos o limitante superior: $|E(1.3165)| \leq 1.0186 \times 10^{-4}$ e a estimativa

$|E(y)| \approx |(y - y_0)(y - y_1)(y - y_2)||\text{máxima diferença dividida de ordem 3}| \approx 1.11028 \times 10^{-4}$

Outline

Fenômeno de Runge

Se $n \to \infty$ temos garantia que polinômio interpolador em $x_0, \ldots, x_n$ converge para $f(x)$ no intervalo?

No caso de pontos igualmente espaçados: NÃO. E mais que isso, na verdade ocorre DIVERGÊNCIA! (Fenômeno de Runge)

Alternativas:

1. Usar funções não-polinomiais (foge do nosso escopo)
2. Interpolar em nós de Chebyshev:

$$x_i = \frac{x_0 + x_n}{2} + \frac{x_n - x_0}{2} \cos\left(\frac{2i+1}{2n+2}\pi\right)$$

3. Interpolar por partes

Outline

Interpolação por partes - *splines*

Ideia da **interpolação por partes** é dividir o intervalo de interpolação em subintervalos e usar polinômios (geralmente distintos) de grau menor em cada uma destas regiões

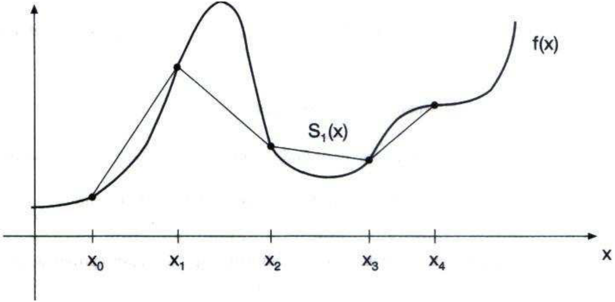

Figura: Fonte: Ruggiero&Lopes

Interpolação por partes - *splines*

No caso específico de **splines** a função interpolante deve satisfazer algumas restrições extras.

Spline

Uma função $S_p(x)$ é chamada *spline* de grau p interpolante nos pontos $x_0, x_1, \ldots, x_n$ em $[a, b]$ se

1. Em cada subintervalo $[x_i, x_{i+1}]$, $S_p(x)$ é um polinômio de grau p: $s_p(x)$
2. $S_p(x)$ é contínua e tem derivadas contínuas até ordem $p-1$ em $[a, b]$
3. $S_p(x_i) = f(x_i)$

Exemplo típico de *splines* são as linhas curvas usadas para desenho ou contorno semi-automático no *photoshop/GIMP* (tesoura mágica)

Splines lineares

Em cada subintervalo $[x_{i-1}, x_i]$ a *spline* linear é dada por

$$s_i(x) = f(x_{i-1})\frac{x_i - x}{x_i - x_{i-1}} + f(x_i)\frac{x - x_{i-1}}{x_i - x_{i-1}}, \forall x \in [x_{i-1}, x_i]$$

Verificando as condições:

1. Pela definição, $S_1(x)$ tem grau 1 em cada subintervalo $[x_{i-1}, x_i]$
2. $S_1(x)$ é contínua em cada subintervalo pela definição e em cada nó:

$$s_i(x_i) = f(x_{i-1})\frac{x_i - x_i}{x_i - x_{i-1}} + f(x_i)\frac{x_i - x_{i-1}}{x_i - x_{i-1}} = f(x_i)$$

$$s_{i+1}(x_i) = f(x_i)\frac{x_{i+1} - x_i}{x_{i+1} - x_i} + f(x_{i+1})\frac{x_i - x_i}{x_{i+1} - x_i} = f(x_i) = s_i(x_i)$$

3. $$S_1(x_i) = s_i(x_i) = f(x_{i-1})\frac{x_i - x_i}{x_i - x_{i-1}} + f(x_i)\frac{x_i - x_{i-1}}{x_i - x_{i-1}} = f(x_i)$$

Splines lineares - Exemplo

Interpolar por *splines* lineares:

	x_0	x_1	x_2	x_3
x	1	2	5	7
f(x)	1	2	3	2.5

$$s_1(x) = f(x_0)\frac{x_1 - x}{x_1 - x_0} + f(x_1)\frac{x - x_0}{x_1 - x_0} = 1\frac{2 - x}{2 - 1} + 2\frac{x - 1}{2 - 1} = 2 - x + 2x - 2 = x, x \in [1, 2]$$

$$s_2(x) = f(x_1)\frac{x_2 - x}{x_2 - x_1} + f(x_2)\frac{x - x_1}{x_2 - x_1} = 2\frac{5 - x}{5 - 2} + 3\frac{x - 2}{5 - 2} = \frac{2}{3}(5 - x) + x - 2 = \frac{1}{3}(x + 4), x \in [2, 5]$$

$$s_3(x) = f(x_2)\frac{x_3 - x}{x_3 - x_2} + f(x_2)\frac{x - x_2}{x_3 - x_2} = 3\frac{7 - x}{7 - 5} + 2.5\frac{x - 5}{7 - 5} = \frac{1}{2}(-0.5x + 8.5), x \in [5, 7]$$

Splines cúbicas

Splines lineares possuem derivadas primeiras descontínuas nos nós. *Splines quadráticas* podem ter descontinuidade na derivada segunda (curvatura) nos nós. Daí o interesse em *splines* cúbicas.

Definição: *Spline* de grau 3 sobre os nós x_i em que cada polinômio s_k (também de grau 3 pela definição) satisfaz:

1. $S_3(x) = s_k(x)$ para $x \in [x_{k-1}, x_k]$, $k = 1, \ldots, n$
2. $S_3(x_i) = f(x_i)$, $i = 0, 1, \ldots, n$
3. $s_k(x_k) = s_{k+1}(x_k), k = 1, \ldots, (n-1)$
4. $s'_k(x_k) = s'_{k+1}(x_k), k = 1, \ldots, (n-1)$
5. $s''_k(x_k) = s''_{k+1}(x_k), k = 1, \ldots, (n-1)$

Vamos denotar o polinômio $s_k(x)$ daqui para frente como

$$s_k(x) = a_k(x - x_k)^3 + b_k(x - x_k)^2 + c_k(x - x_k) + d_k, k = 1, \ldots, n$$

Splines cúbicas

A condição 1 resulta da própria definição de *spline* interpolante

Já para a condição 2 temos 2 casos:

$$\begin{aligned} s_k(x_k) &= d_k = f(x_k), k = 1, \ldots, n \\ s_1(x_0) &= -a_1 h_1^3 + b_1 h_1^2 - c_1 h_1 + d_1 = f(x_0) \end{aligned} \tag{1}$$

em que $h_k = x_k - x_{k-1}$

Para a condição 3:

$$s_{k+1}(x_k) = f(x_k) \Rightarrow -a_{k+1} h_{k+1}^3 + b_{k+1} h_{k+1}^2 - c_{k+1} h_{k+1} + d_{k+1} = f(x_k) \tag{2}$$

Para as condições 4 e 5 devemos primeiramente definir as derivadas:

$$\begin{aligned} s'_k(x) &= 3a_k(x - x_k)^2 + 2b_k(x - x_k) + c_k \\ s''_k(x) &= 6a_k(x - x_k) + 2b_k \end{aligned} \tag{3}$$

Splines cúbicas

Veja que $s_k''(x_k) = 2b_k$ e portanto b_k sai de

$$b_k = \frac{s_k''(x_k)}{2}$$

Dado que $s_k''(x_{k-1}) = -6a_k h_k + 2b_k$, também podemos obter a_k:

$$a_k = \frac{2b_k - s_k''(x_{k-1})}{6h_k} = \frac{s_k''(x_k) - s_k''(x_{k-1})}{6h_k}$$

Usando a condição 5 ($s_k''(x_{k-1}) = s_{k-1}''(x_{k-1})$):

$$a_k = \frac{s_k''(x_k) - s_{k-1}''(x_{k-1})}{6h_k}$$

em que, quando $k = 1$, $s_0''(x_0)$ é arbitrária pois não temos a derivada segunda naquele ponto.
Dado que a estas alturas já temos uma expressão em função de f e suas derivadas para a_k, b_k e d_k podemos obter expressão semelhante para c_k usando (1) e (2). Da (2) obtemos c_1:

Splines cúbicas

$$c_1 = \frac{-f(x_0) + a_1 h_1^3 - b_1 h_1^2 - d_1}{h_1}$$

e da (2):

$$c_k = \frac{-f(x_{k-1}) - a_k h_k^3 + b_k h_k^2 + d_k}{h_k} = \frac{f(x_k) - f(x_{k-1})}{h_k} - (a_k h_k^2 - b_k h_k) =$$

$$= \frac{f(x_k) - f(x_{k-1})}{h_k} - \left\{ \frac{[s_k''(x_k) - s_k''(x_{k-1})]}{6} h_k - \frac{s_k''(x_k)}{2} h_k \right\}$$

Veja que a expressão acima engloba o caso $k = 1$. Então temos uma expressão geral para $k = 1, \ldots, n$:

$$c_k = \frac{f(x_k) - f(x_{k-1})}{h_k} - \frac{-2s_k''(x_k)h_k - s_{k-1}''(x_{k-1})h_k}{6}.$$

Para simplificar vamos denotar $s_k''(x_k) = g_k$ e $f(x_k) = y_k$

Splines cúbicas

Deste modo:

$$a_k = \frac{g_k - g_{k-1}}{6h_k} \tag{4}$$

$$b_k = \frac{g_k}{2} \tag{5}$$

$$c_k = \frac{y_k - y_{k-1}}{h_k} - \frac{2h_k g_k + g_{k-1}h_k}{6} \tag{6}$$

$$d_k = y_k \tag{7}$$

Vamos agora finalmente introduzir a condição 4: $s'_k(x_k) = s'_{k+1}(x_k)$. Pela expressão da derivada (3):

$$s'_k(x_k) = c_k = 3a_{k+1}h_{k+1}^2 - 2b_{k+1}h_{k+1} + c_{k+1}$$

$$\Rightarrow c_{k+1} = c_k - 3a_{k+1}h_{k+1}^2 + 2b_{k+1}h_{k+1}$$

Splines cúbicas

Substituindo pelas expressões em (4), (5) e (6):

$$\frac{y_{k+1}-y_k}{h_{k+1}}+\frac{2h_{k+1}g_{k+1}+g_kh_{k+1}}{6}=$$

$$=\frac{y_k-y_{k-1}}{h_k}+\frac{2h_kg_k+g_{k-1}h_k}{6}-3\left(\frac{g_{k+1}-g_k}{6}\right)h_{k+1}+2\left(\frac{g_{k+1}h_{k+1}}{2}\right)$$

Agrupando:

$$\frac{1}{6}[h_kg_{k-1}+(2h_k+3h_{k+1}-h_{k+1})g_k+(6h_{k+1}-3h_{k+1}-2h_{k+1})g_{k+1}]=$$

$$=\frac{y_{k+1}-y_k}{h_{k+1}}-\frac{y_k-y_{k-1}}{h_k}$$

ou ainda

$$h_kg_{k-1}+2(h_k+h_{k+1})g_k+h_{k+1}g_{k+1}=6\left(\frac{y_{k+1}-y_k}{h_{k+1}}-\frac{y_k-y_{k-1}}{h_k}\right)$$

Splines cúbicas

Esta expressão recai em um sistema com $(n-1)$ equações $(k = 1, \ldots, (n-1))$ e $(n+1)$ incógnitas $(g_0, \ldots, g_n)$ e portanto indeterminado:

$$A = \begin{bmatrix} h_1 & 2(h_1+h_2) & h_2 & & & \\ & h_2 & 2(h_2+h_3) & h_3 & & \\ & & \ddots & \ddots & \ddots & \\ & & h_{n-1} & 2(h_{n-1}+h_n) & h_n \end{bmatrix}_{(n-1)}$$

$$b = \begin{bmatrix} \frac{y_2-y_1}{h_2} - \frac{y_1-y_0}{h_1} \\ \frac{y_3-y_2}{h_3} - \frac{y_2-y_1}{h_2} \\ \vdots \\ \frac{y_n-y_{n-1}}{h_n} - \frac{y_{n-1}-y_{n-2}}{h_{n-1}} \end{bmatrix}_{(n-1)\times 1}$$

Splines cúbicas

Para solução única precisamos impor mais duas condições. Por exemplo:

- $S_3''(x_0) = g_0 = 0$ e $S_3''(x_n) = g_n = 0$ (próximos de funções lineares nos extremos): *spline* natural
- $g_0 = g_1$, $g_n = g_{n-1}$ (próximos de parábolas nos extremos)
- $S_3'(x_0) = A$, $S_3'(x_n) = B$ (inclinações específicas nos extremos), gerando as equações extras:

$$s_1'(x_0) = 3a_1h^2 - 2b_1h + c_1 = A$$

$$s_n'(x_n) = c_n = B$$

Splines cúbicas - Exemplo

Aproximar $f(0.25)$ por *spline* cúbica natural:

x	0	0.5	1.0	1.5	2.0
f(x)	3	1.8616	-0.5571	-4.1987	-9.0536

Como temos 4 subdivisões do intervalo $[0, 2.0]$ precisamos obter $s_1(x)$, $s_2(x)$, $s_3(x)$ e $s_4(x)$ e devemos então resolver o sistema

$$h_k g_{k-1} + 2(h_k + h_{k+1})g_k + h_{k+1}g_{k+1} = 6\left(\frac{y_{k+1} - y_k}{h_{k+1}} - \frac{y_k - y_{k-1}}{h_k}\right), 1 \leq k \leq$$

Como no caso $h_k = h = 0.5$:

$$hg_{k-1} + 4hg_k + hg_{k+1} = \frac{6}{h}(y_{k+1} - 2y_k + y_{k-1})$$

Substituindo para cada k:

$$\begin{cases} hg_0 + 4hg_1 + hg_2 = \frac{6}{h}(y_2 - 2y_1 + y_0) \\ hg_1 + 4hg_2 + hg_3 = \frac{6}{h}(y_3 - 2y_2 + y_1) \\ hg_2 + 4hg_3 + hg_4 = \frac{6}{h}(y_4 - 2y_3 + y_2) \end{cases}$$

Splines cúbicas - Exemplo

Como queremos a *spline* natural temos ainda as restrições $g_0 = g_4 = 0$:

$$\begin{cases} 4hg_1 + hg_2 & = \frac{6}{h}(y_2 - 2y_1 + y_0) \\ hg_1 + 4hg_2 + hg_3 & = \frac{6}{h}(y_3 - 2y_2 + y_1) \\ hg_2 + 4hg_3 & = \frac{6}{h}(y_4 - 2y_3 + y_2) \end{cases}$$

Na forma matricial:

$$\begin{bmatrix} 4h & h & 0 \\ h & 4h & h \\ 0 & h & 4h \end{bmatrix} \begin{bmatrix} g_1 \\ g_2 \\ g_3 \end{bmatrix} = \frac{6}{h} \begin{bmatrix} y_2 - y_1 + y_0 \\ y_3 - 2y_2 + y_1 \\ y_4 - 2y_3 + y_2 \end{bmatrix}$$

Substituindo pelos valores já conhecidos de h e y_i

$$\begin{bmatrix} 2 & 0.25 & 0 \\ 0.5 & 2 & 0.5 \\ 0 & 0.5 & 2 \end{bmatrix} \begin{bmatrix} g_1 \\ g_2 \\ g_3 \end{bmatrix} = \begin{bmatrix} -15.3636 \\ -14.6748 \\ -14.5598 \end{bmatrix}$$

Por eliminação de Gauss:

$$[g_1 \quad g_2 \quad g_3]^T = [-6.6541 \quad -4.111 \quad -6.252]^T$$

Splines cúbicas - Exemplo

Partindo destes valores obtemos a_k, b_k, c_k, d_k e consequentemente $s_1(x), s_2(x), s_3(x), s_4(x)$. Como neste caso em particular queremos aproximar $f(0.25)$ usamos $s_1(x)$:
$s_1(x) = a_1(x - x_1)^3 + b_1(x - x_1)^2 + c_1(x - x_1) + d_1$, o que usando os valores de g_i já obtidos:

$$a_1 = \frac{g_1 - g_0}{6h} = \frac{-6.6541}{3} = -2.2180$$

$$b_1 = \frac{g_1}{2} = -3.3270$$

$$c_1 = \frac{y_1 - y_0}{h} + \frac{2hg_1 - g_0 h}{6} = -3.3858$$

$$d_1 = y_1 = 1.8616$$

Finalmente:

$$f(0.25) \approx s_1(0.25) = -2.2180(-0.25)^3 - 3.3270(0.25)^2 - 3.3858(-0.25) + 1.8616 = 2.5348$$

Observações finais

- Seguindo as condições aqui impostas um polinômio seria interpolado por outro exatamente igual o original (unicidade)
- O erro é minimizado quando o ponto que queremos calcular pela interpolação está mais central am relação a $[x_0, x_n]$
- De um modo geral, são os nós de Chebyshev que minimizam o erro na interpolação