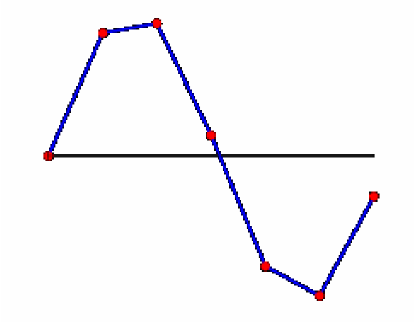
Interpolação

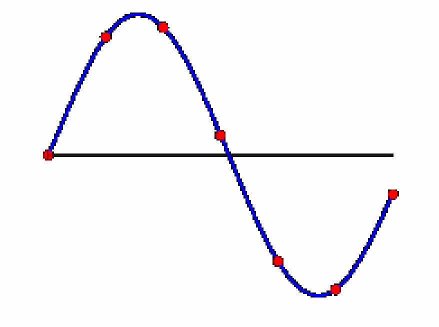
Interpolação é um método que permite construir um novo conjunto de dados a partir de um conjunto discreto de dados pontuais conhecidos. Em engenharia e ciências, dispõe-se habitualmente de dados pontuais, obtidos a partir de uma amostragem ou experimento. Através da interpolação pode-se construir uma função que aproximadamente se "ajuste" nestes dados pontuais.

Outra aplicação da interpolação é aproximação de funções complexas por funções mais simples. Suponha que tenhamos uma função, mas que seja muito complicada para avaliar de forma eficiente. Podemos então, escolher alguns dados pontuais da função complicada e tentar interpolar estes dados para construir uma função mais simples. Obviamente, quando utilizamos a função mais simples para calcular novos dados, normalmente não se obtém o mesmo resultado da função original, mas dependendo do domínio do problema e do método de interpolação utilizado, o ganho de simplicidade pode compensar o erro.

A **interpolação** permite fazer a reconstituição (aproximada) de uma função apenas conhecendo algumas das suas abscissas e respectivas ordenadas (imagens). A função resultante passa nos pontos fornecidos e, em relação aos outros pontos, pode ser um mero ajuste.



Exemplo de Interpolação Linear



Exemplo de Interpolação Polinomial de grau superior a 1.

Tipos de interpolação

- Interpolação linear
- Interpolação polinomial
- Interpolação trigonométrica

Interpolação linear

Em análise numérica, a **interpolação linear** consiste em aproximar uma função num intervalo por uma função linear, ou seja, por utilizando de polinômios de primeiro grau. O principal problema é que se os pontos forem poucos ou muito afastados entre si, a representação gráfica para uma determinada função não seria muito bem representada por tal método. Sendo necessário, talvez, a utilização de polinômios de graus mais elevados (usando-se polinômio interpolador de Lagrange, por exemplo).

Interpolação polinomial

Diz-se **interpolação polinomial** quando a função interpoladora é um polinômio. A função interpoladora é a função $F(x)$.

Chama-se de interpolação ao processo de avaliar $f(x), x \in [a,b]$ substituindo a função $f(x)$ por uma função $F(x)$, tal que $F(x_i) = f(x_i), i = 1(1)n$. O $f(x)$ é uma função real definida em $[a,b] \in IR$, da qual conhecem-se os valores nos pontos de abcissas $x_1, x_2, \cdots x_i \cdots x_n \in [a,b]$.

Métodos de interpolação polinomial

Nos métodos de interpolação utilizam-se polinômios como funções interpoladores (interpolação polinomial). Escolhem-se os polinômios pela sua (relativa) simplicidade e porque permitem uma representação satisfatória da maioria das funções que surgem em aplicações práticas.

Os métodos de interpolação polinomial diferem uns dos outros na táctica escolhida para determinar o polinômio interpolador (os erros de arredondamento são diferentes, porque as operações aritmética são conduzidas de forma distinta em cada método).

- Polinômios de Newton
- Polinômios de Gragory-Newton
- Polinômios de Lagrange
- Outros Polinômios (Chebichev, Bernstein)

Polinômio de Lagrange

Em análise numérica, **polinômio de Lagrange** (nome é devido a Joseph-Louis de Lagrange) é o polinômio de interpolação de um conjunto de pontos na forma de Lagrange.

Dado um conjunto de n+1 pontos:

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

com todos x_j distintos, o polinômio de interpolação de um conjunto de pontos na forma de Lagrange é a combinação linear dos polinômios da base de Lagrange:

$$L(x) = \sum_{i=0}^{n} y_i L_i(x)$$

com polinômios da base de Lagrange dados por:

$$L_i(x) = \prod_{j=0, j\neq i}^{n} \frac{x - x_j}{(x_i - x_j)} = \frac{(x - x_0)}{(x_i - x_0)} \cdots \frac{(x - x_{i-1})}{(x_i - x_{i-1})} \cdot \frac{(x - x_{i+1})}{(x_i - x_{i+1})} \cdots \frac{(x - x_n)}{(x_i - x_n)}$$

Temos abaixo a complexidade da interpolação de Lagrange para um um polinômio de grau n com relação ao número de operações:

Operações	Complexidade
Adições	$2n^2 + 3n + 1$
Multiplicações	$2n^2 + 3n + 1$
Divisões	$n + 1$

Polinômio de Newton

O **polinômio de Newton** (nome é devido a Isaac Newton) é um polinômio interpolador para um dado conjunto de pontos. Os coeficientes do polinômio são calculados através de diferenças divididas.

Dado um conjunto de n+1 pontos:

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

com todos x_j distintos, o polinômio de interpolação de um conjunto de pontos na forma de Newton é dado por:

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

ou

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

Onde

$$\Delta^i y_0$$

representa a diferença dividida de *i*-ésima ordem 0.

Operador de diferença dividida

Seja a função *y = f(x)* cujo gráfico passa pelos pontos (x_i, y_i), i = 0,1,2,...,n. O operador de diferença dividida Δ é definido como sendo

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

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

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

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

Temos abaixo a complexidade da interpolação de Newton para um um polinômio de grau *n* com relação ao número de operações:

Operações	**Complexidade**
Adições	$n^2 + 3n$
Multiplicações	n
Divisões	$\dfrac{n^2 + n}{2}$

Polinômio de Gregory-Newton

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

Fórmula de Gregory-Newton

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

.

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

,

onde utiliza-se uma variável auxiliar $u_x = u(x) = \dfrac{x - x_0}{h}$
ou

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

Operador de diferença finita ascendente

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

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*: $\Delta^n y_i = \Delta^{n-1} y_{i+1} - \Delta^{n-1} y_i$

Temos abaixo a complexidade da interpolação de Gregory-Newton para um polinômio de grau *n* com relação ao número de operações:

Operações	**Complexidade**
Adições	$\frac{1}{2}n^2 + \frac{7}{2}n + 2$
Multiplicações	*n*
Divisões	*n+1*

Estudo Comparativo dos Polinômios

Como vimos acima cada um dos métodos numéricos possui um número específico de operações aritméticas que podem ser resumidos na tabela abaixo:

Método Numérico	**Número de Operações Aritméticas**		
	Adições	**Multiplicações**	**Divisões**
Lagrange	$2n^2 + 3n + 1$	$2n^2 + 3n + 1$	*n+1*
Gregory-Newton	$\frac{1}{2}n^2 + \frac{7}{2}n + 2$	*n*	$n + 1$
Newton	$n^2 + 3n$	*n*	$\frac{1}{2}n^2 + \frac{1}{2}n$

Analisando a tabela acima não podemos afirmar diretamente qual é o método que possui o menor custo de computação. Isto deve-se ao fato de que em geral, um adição gasta

menos ciclos de máquina que uma multiplicação que gasta menos ciclos que uma divisão.
A dependência dos tempos de ciclos em relação a arquitetura de máquina utilizada é um fator importante a ser considerado na análise de eficiência dos métodos apresentados acima.
Foram implementados os 3 métodos utilizando o software MATLAB. Utilizando um conjunto de pontos a serem interpolados foram feitas várias simulações contabilizando o tempo de CPU gasto na execução de cada um dos métodos. As simulações foram feitas utilizando uma máquina PC 400Mhz. Os resultados podem ser sintetizados na figura abaixo.

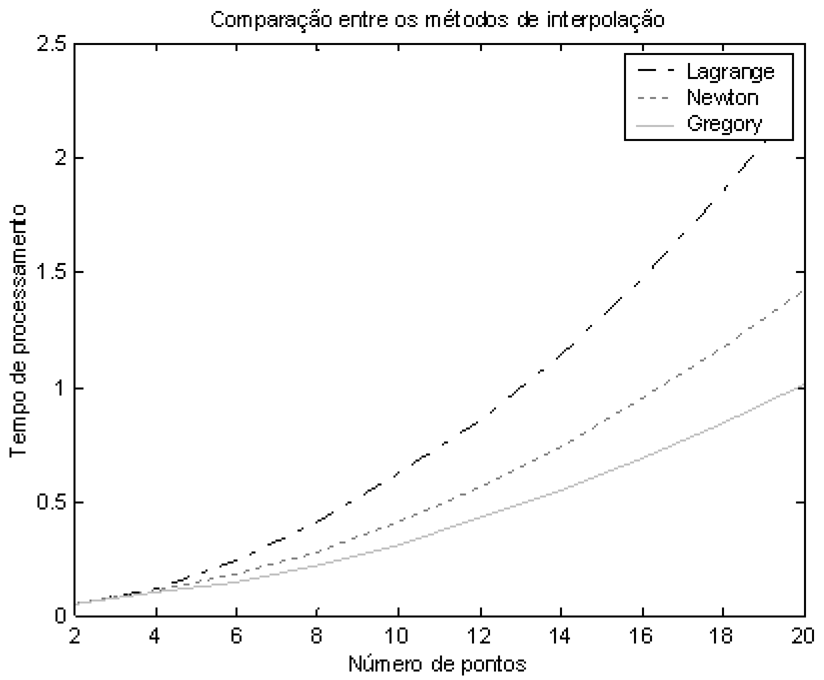


Fig. 1 – Tempo de processamento dos métodos seqüências de interpolação

Na figura 1 podemos verificar que o método Gregory-Newton é o mais eficiente para a arquitetura PC. A desvantagem na utilização deste método, é a exigência de que as abscissas dos pontos a serem utilizados para o polinômio interpolador de grau n, devam ser necessariamente eqüidistante.