Interpolação Polinomial

Algorítmo de Neville

Fórmula de Lagrange

Lembre que para cada “x” o polinômio é calculado como a soma de N termos

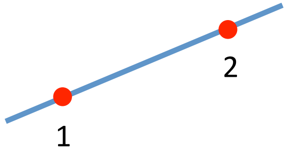
$$P(x) = \sum_{i=1}^{N} Y_i \; L_i(x)$$

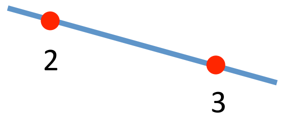
onde cada termo possui um produtório de N-1 fatores

$$L_i(x) = \prod_{j=1, j \neq i}^{N} \frac{x - X_j}{X_i - X_j}$$

Ou seja, para cada “y” calculado a partir da fórmula “y = P(x)” no mínimo N*(N-1) atribuições deverão ser calculadas. É por isto que na implementação da Fórmula de Lagrange temos um laço de N ciclos, dentro do qual tem um outro laço de N-1 ciclos.

Base do Algorítmo de Neville: Fórmula de recorrência





$$P_{12}(x) = \frac{1}{X_1 - X_2}[(x - X_2)Y_1 - (x - X_1)Y_2] \qquad P_{23}(x) = \frac{1}{X_2 - X_3}[(x - X_3)Y_2 - (x - X_2)Y_3]$$

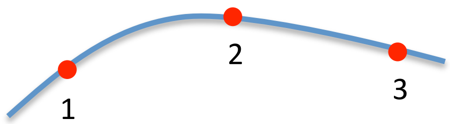
$$P_{123}(x) = \frac{1}{X_1 - X_3}[(x - X_3)P_{12}(x) - (x - X_1)P_{23}(x)]$$

Ou escrito de outra forma

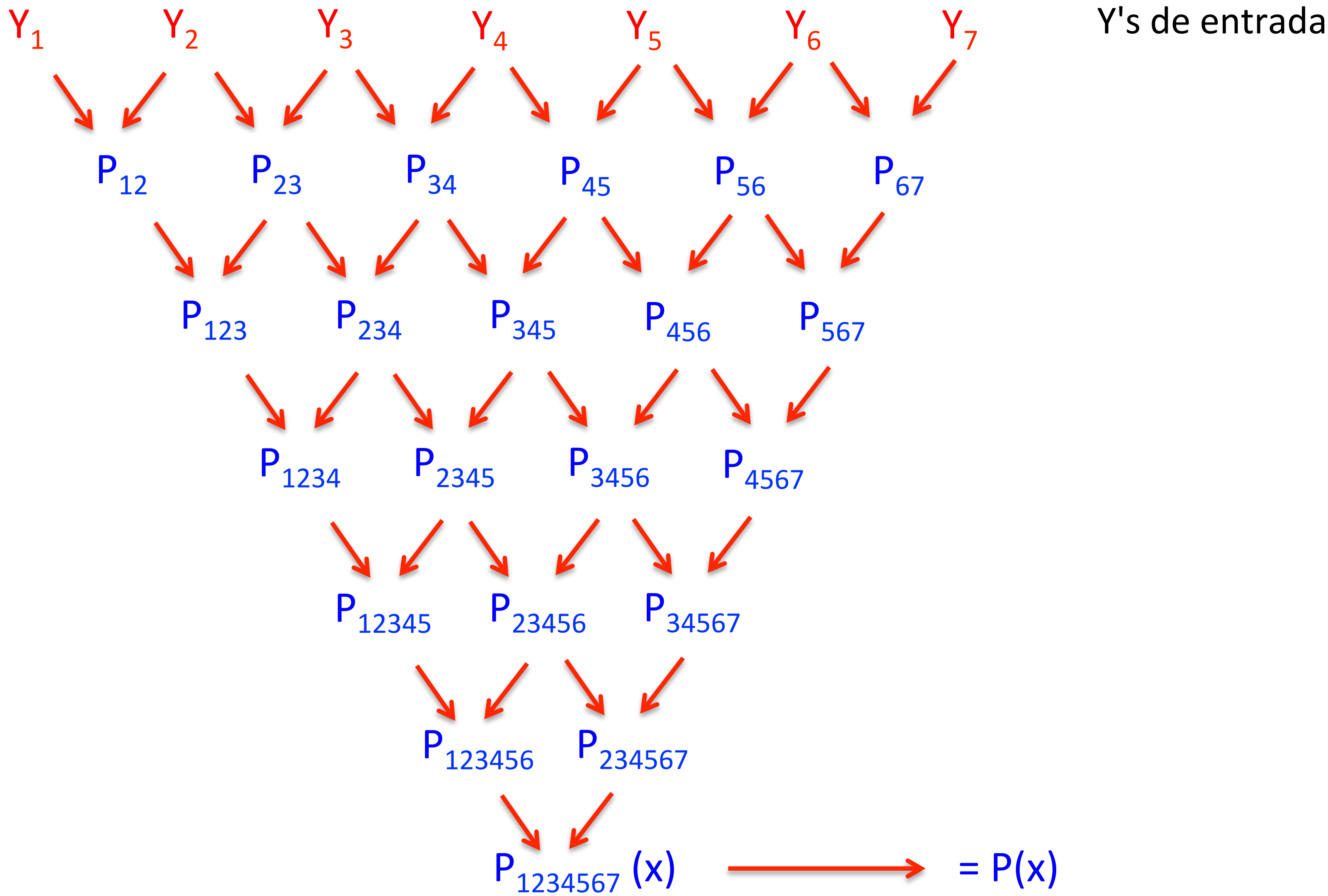
$$P_{123}(x) = \frac{x - X_3}{X_1 - X_3} P_{12}(x) - \frac{x - X_1}{X_1 - X_3} P_{23}(x)$$

Onde sabemos que

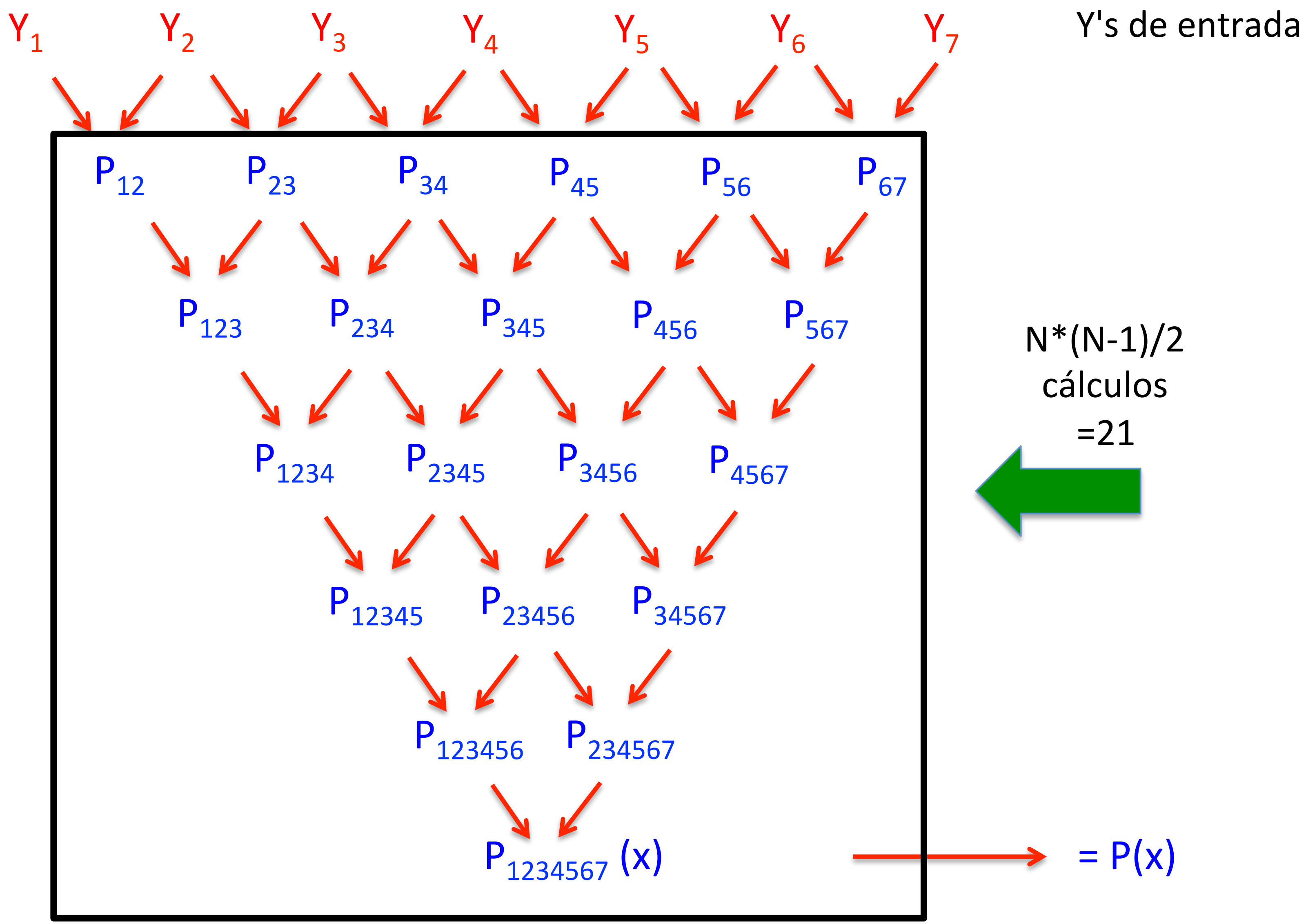
$$P_{12}(X_1) = Y_1 \qquad P_{23}(X_2) = Y_2$$
$$P_{12}(X_2) = Y_2 \qquad P_{23}(X_3) = Y_3$$



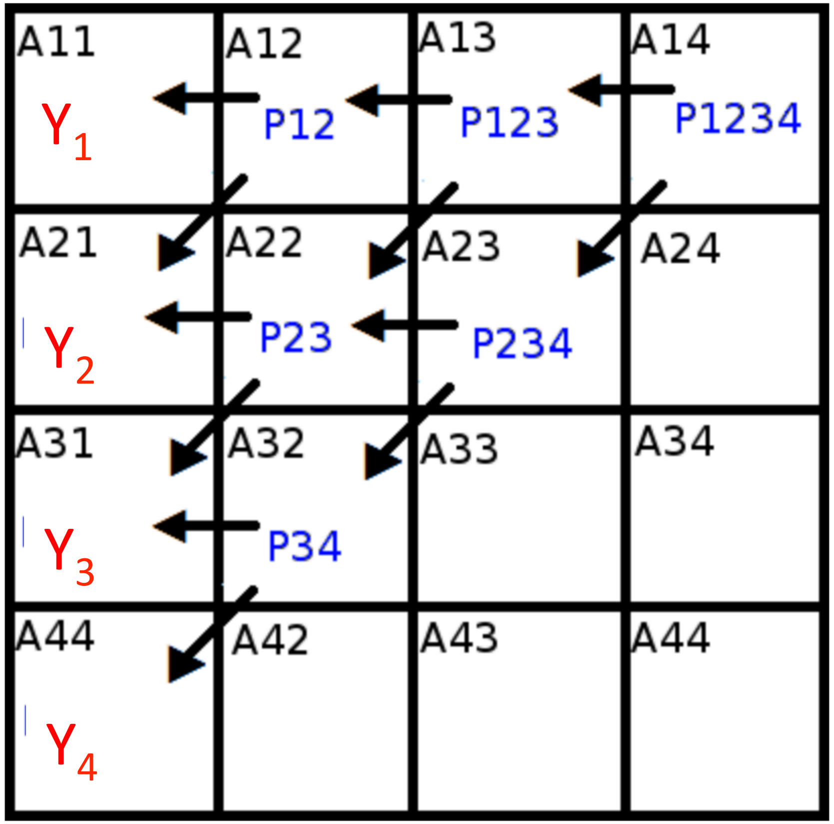
Base do Algorítmo de Neville: Fórmula de recorrência



Base do Algorítmo de Neville: Fórmula de recorrência



A base do algorítmo é ter uma matriz quadrada N x N onde N é o número de pontos a serem interpolados



$$A_{ij} = \frac{1}{X[i] - X[k]}[(x - X[k])A_{i,j-1} - (x - X[i])A_{i+1,j-1}]$$

onde

$$k = j + i - 1$$

No final do processo, o polinômio interpolador é dado por

$$P(x) = A14.$$

A leitura dos pontos é dado pelos valores de X e da primeira coluna da matriz A.

$$X, Y \to X[i], A[i][1]$$

A ordem do preenchimento é fundamental:

$$j = 1;\ i = 1, 2, 3, 4$$
$$j = 2;\ i = 1, 2, 3$$
$$j = 3;\ i = 1, 2$$
$$j = 4;\ i = 1$$
$$j = 1 \ldots N;\ i = 1 \ldots (N - j + 1)$$

Esta implementação usa índices de matrizes de 1 a N , no entanto você pode adaptar para índices de 0 a N-1 que são mais convenientes em C.

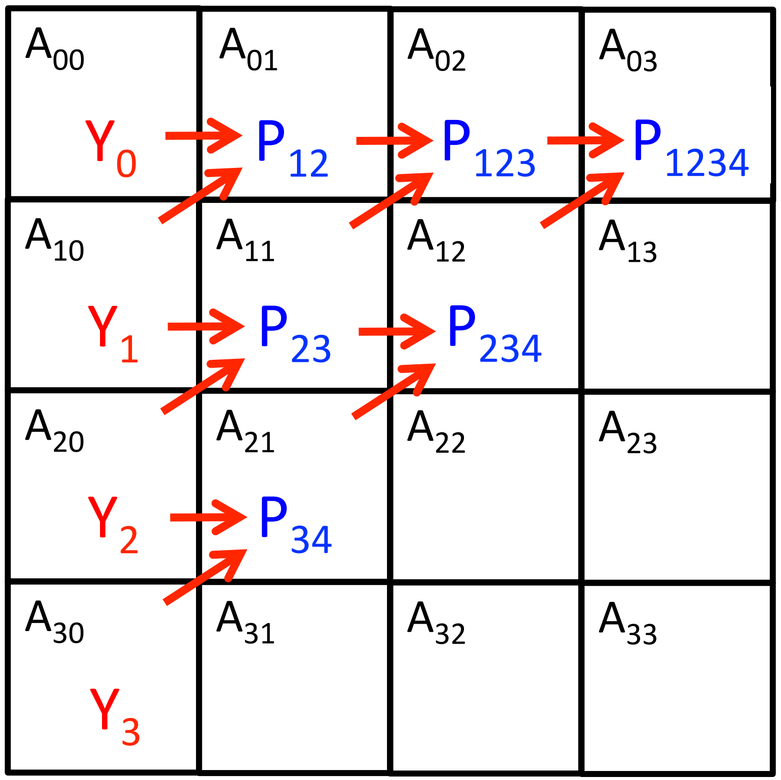
Exemplo de uma função que implementa o Algorítmo de Neville (Usando índices de 1 a n para a Matriz A)

```c
float  neville_func( float x, int n, float point[n][2])
{
int i,j,k;
float A[n+1][n+1];

for(i=1;i<=n;i++)
   {
   A[i][1] = point[i-1][1];
   }

for(j=2; j<= n;j++)
   {
   for(i=1; i<= n-j+1; i++)
      {
      k = j+i-1;
      A[i][j] =  ( (x - point[k-1][0])*A[i][j-1] - (x-point[i-1][0])*A[i+1][j-1] )/(point[i-1][0]-point[k-1][0]);
      }
   }
return  A[1][n];
}
```

Implementação do Algoritmo de Neville usando índices de 0 até N-1



$$A_{01} = \frac{1}{X_0 - X_1}[(x - X_1)A_{00} - (x - X_0)A_{10}]$$

$$A_{11} = \frac{1}{X_1 - X_2}[(x - X_2)A_{10} - (x - X_1)A_{20}]$$

$$\vdots$$

$$A_{02} = \frac{1}{X_0 - X_2}[(x - X_2)A_{01} - (x - X_0)A_{11}]$$

Podemos claramente generalizar isto
Para a fórmula de recorrência

$$A_{ij} = \frac{1}{X_i - X_{i+j}}[(x - X_{i+j})A_{i,j-1} - (x - X_0)A_{i+1,j-1}]$$

Exemplo de uma função que implementa o Algorítmo de Neville (Usando índices de 0 a n-1 para a Matriz A)

```c
 float  neville_func( float x, int n, float point[n][2])
{
int i,j,k;
float A[n][n];

for(i=0;i<n;i++)
   {
   A[i][0] = point[i][1];
   }

for(j=1; j< n; j++)
   {
   for(i=0; i< n-j; i++)
      {
      k = j+i;
      A[i][j] =  ( (x - point[k][0])*A[i][j-1] - (x-point[i][0])*A[i+1][j-1] )/(point[i][0]-point[k][0]);
      }
   }
return A[0][n-1];
}
```