

$$P_n(x) = \sum_{i=0}^{n-1} a_i \cdot p_i(x)$$

$$p_i(x) = \begin{cases} 1 & \text{se } i = 0 \\ \prod_{k=0}^{i-1} (x - x_k) & \text{se } i > 0 \end{cases}$$

$$a_i = f[x_0, x_1, \dots, x_i] \\ = \frac{f[x_1, x_2, \dots, x_n] - f[x_0, x_1, \dots, x_{n-1}]}{x_n - x_0}$$

Para um caso onde  $n = 4$ , temos uma matriz quadrada de ordem  $m = 3$ :

$$\text{tabela} = \begin{matrix} & \begin{matrix} 0 & 1 & 2 & 3 \end{matrix} \\ \begin{matrix} 0 \\ 1 \\ 2 \\ 3 \end{matrix} & \begin{pmatrix} f[x_0] & f[x_0, x_1] & f[x_0, x_1, x_2] & f[x_0, \dots, x_3] \\ f[x_1] & f[x_1, x_2] & f[x_1, x_2, x_3] & \\ f[x_2] & f[x_2, x_3] & & \\ f[x_3] & & & \end{pmatrix} \end{matrix}$$

ou seja, cada elemento  $a_{ij}$  é calculado apenas com  $i = 0$  até  $i = m - j$ , Para

<i>passo 1</i>	<i>passo 2</i>	<i>passo 3</i>	<i>...</i>
$f(x_0)$	$f(x_0, x_1)$	$f(x_0, x_1, x_2)$	$f(x_0, x_1, x_2, x_3)$
$f(x_1)$	$f(x_1, x_2)$	$f(x_1, x_2, x_3)$	
$f(x_2)$	$f(x_2, x_3)$		
$f(x_3)$			

cada linha  $i$  e coluna  $j$ , para  $i = 1$  até  $m - 1$ , os termos  $a_{ij}$  da matriz serão:

$$a_{ij} \leftarrow \begin{cases} f[x_i, x_\alpha] = \frac{f[x_\alpha] - f[x_i]}{x_\alpha - x_i} & \text{somente em } j = 1 \\ f[x_0, \dots, x_j] = \frac{a_{v\varepsilon} - a_{i\varepsilon}}{x_\alpha - x_i} & \text{desde } j = 2 \end{cases} \quad (1)$$

onde:

$$\begin{aligned} \alpha &= i+j \\ v &= i+1 \\ \varepsilon &= j-1 \end{aligned}$$

E finalmente, com  $j$  representando o número da coluna da matriz criada,

$$P_n(x) = \sum_{j=0}^m a_{0j} \cdot \prod_{\substack{k=0 \\ j \neq 0}}^{j-1} (x - x_k) \quad (2)$$

# 1 USO DO PROGRAMA

Existem dois tipos de entradas para o programa:

1. usando um arquivo texto devidamente formatado.
2. passando os dados “manualmente” via teclado.

Para demonstrar o caso (1), consideremos o seguinte arquivo de entrada como exemplo:

```
1 4
2 0 2
3 2 3
4 5 4
5 9 1
```

A primeira linha deve ser um número  $n$  natural que indique a quantidade de pontos (a serem lidos a seguir). A as  $n$ -ésimas linhas restantes deve ter a forma  $x f(x)$ , que descrevem a tabela das coordenadas.

Para compilar e executar o programa usando esse arquivo **arqEntradas** digitamos esses dois comandos na linha de comandos:

```
$ gcc -D AUTO -o polinomioNewton main.c
$ ./polinomioNewton < arqEntradas
```

Essa opção apenas desativa os *printfs* para não poluir a tela quando um arquivo é passado.

Para demonstrar o caso (2), consideremos as mesmas entradas do exemplo anterior só que agora elas serão inseridas/digitadas pelo usuário.

Para compilar o comando é o mesmo só que agora sem a opção **-D**:

```
$ gcc -o polinomioNewton main.c
```

Após compilar e digitar o comando seguinte, teremos (neste caso) a execução do programa que espera o usuário digitar os valores indicados. A figura (1) mostra esta execução.

```
$ ./polinomioNewton # para executar o programa
```

```
~ polinomioNewton $ ./polinomioNewton
>> Digite a quantidade de pontos (numero natural): 4
>> Digite no formato "x f(x)" (sem aspas) seguido de um ENTER a cada Xi
i
0: 0 2
1: 2 3
2: 5 4
3: 9
```

Nos dois tipos de entradas demonstrados, a precisão dos coeficientes do polinômio interpolador é (por padrão) de 3 casas decimais. Para alterar isso, independente do modo de entrada dos dados, basta acrescentar **-D PRECISAO=N** ao executar o comando que compilará o código, por exemplo:

```
$ gcc -D PRECISAO -o polinomioNewton main.c    # o mesmo que N=1
$ gcc -D PRECISAO=4 -o polinomioNewton main.c # N deve ser um número natural em [0..6]
```