Interpolação Polinomial de Newton

Interpolação Linear: $n = 1$ $\begin{cases} x_0 \to p_1(x_0) = f(x_0) \\ x_1 \to p_1(x_1) = f(x_1) \end{cases}$, tem-se:

$p_0(x) = C_0 = f(x_0)$ e $p_1(x) = p_0(x) + C_1 \cdot (x - x_0)$,note que $p_0(x)$ já satisfaz a primeira condição e para que $p_1(x_1) = f(x_1)$ deve-se ter:

$$p_1(x_1) = f(x_1) = p_0(x_1) + C_1 \cdot (x_1 - x_0) \Rightarrow C_1 = \frac{f(x_1) - p_0(x_1)}{x_1 - x_0}$$

Interpolação Parabólica: $n = 2$ $\begin{cases} x_0 \to p_2(x_0) = f(x_0) \\ x_1 \to p_2(x_1) = f(x_1) \\ x_2 \to p_2(x_2) = f(x_2) \end{cases}$, tem-se:

$p_2(x) = p_1(x) + C_2 \cdot (x - x_0) \cdot (x - x_1)$, note que $p_1(x)$ já satisfaz as duas primeiras condições e para que $p_2(x_2) = f(x_2)$ deve-se ter:

$$p_2(x_2) = f(x_2) = p_1(x_2) + C_2 \cdot (x_2 - x_0) \cdot (x_2 - x_1) \Rightarrow C_2 = \frac{f(x_2) - p_1(x_2)}{(x_2 - x_0) \cdot (x_2 - x_1)}$$

Interpolação Cúbica: $n = 3$ $\begin{cases} x_0 \to p_3(x_0) = f(x_0) \\ x_1 \to p_3(x_1) = f(x_1) \\ x_2 \to p_3(x_2) = f(x_2) \\ x_3 \to p_3(x_3) = f(x_3) \end{cases}$, tem-se:

$p_3(x) = p_2(x) + C_3 \cdot (x - x_0) \cdot (x - x_1) \cdot (x - x_2)$, note que $p_2(x)$ já satisfaz as três primeiras condições e para que $p_3(x_3) = f(x_3)$ deve-se ter:

$$p_3(x_3) = f(x_3) = p_2(x_3) + C_3 \cdot (x_3 - x_0) \cdot (x_3 - x_1) \cdot (x_3 - x_2) \Rightarrow C_3 = \frac{f(x_3) - p_2(x_3)}{(x_3 - x_0) \cdot (x_3 - x_1) \cdot (x_3 - x_2)}$$

Recursivamente:

$$\begin{cases} p_m(x) = p_{m-1}(x) + C_m \cdot (x - x_0)\cdot(x - x_1)\cdot\cdots\cdot(x - x_{m-1}) \\ C_m = \dfrac{f(x_m) - p_{m-1}(x_m)}{(x_m - x_0)\cdot(x_m - x_1)\cdot\cdots\cdot(x_m - x_{m-1})} \end{cases} \quad \text{para } m = 1, \ \cdots, n$$

inicializando com: $p_0(x) = C_0 = f(x_0)$.

Tal procedimento recursivo foi implementado em MATHCAD conforme apresentado a seguir.

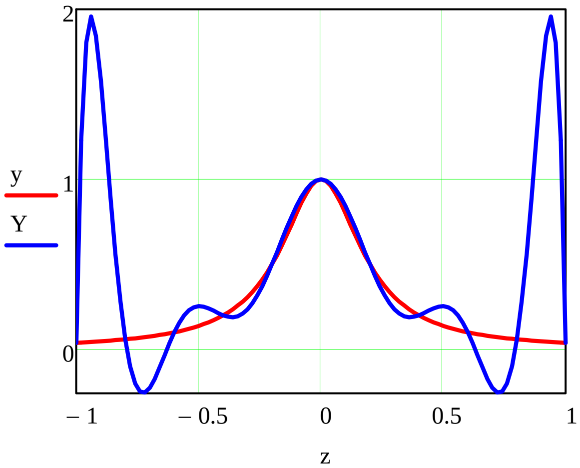
```
P_Newton(n, c, x, X) := | y ← c_n
                        | I ← n − 1
                        | for  i ∈ I.. 0          if  n > 0
                        |   y ← c_i + y·(X − x_i)
                        | y

Coef(n, x, f) := | c_0 ← f(x_0)
                 | m ← 0
                 | for  i ∈ 1.. n
                 |   | c_i ← (f(x_i) − P_Newton(m, c, x, x_i)) / (∏_{k=0}^{m} (x_i − x_k))
                 |   | m ← i
                 | c
```

Para exemplificar o uso desse procedimento, o aplicamos à interpolação polinomial de décimo grau da função: $f(x) = \dfrac{1}{1 + 25 \cdot x^2}$ efetuado no intervalo: [–1,+1] adotando onze pontos igualmente espaçados (incluindo as duas extremidades!).

$f(x) := \dfrac{1}{1 + 25 \cdot x^2}$ $\quad n := 10 \quad k := 0..\, n \quad x_k := \dfrac{2 \cdot k}{n} - 1 \quad c := \mathrm{Coef}(n, x, f) \quad f_{ap}(X) := P_{Newton}(n, c, x, X)$

$j := 0..\, 100 \quad z_j := \dfrac{j}{50} - 1 \quad y_j := f(z_j) \quad Y_j := f_{ap}(z_j) \quad \mathrm{erro} := y - Y$

A representação gráfica da função e de sua interpolação polinomial, assim como a representação gráfica do errosão mostradas a seguir.

Gráfico de $f(x)=\dfrac{1}{1+25\cdot x^2}$ [curva vermelha] e correspondente interpolação polinomial de décimo grau (onze pontos de interpolação igualmente espaçados) [curva azul] 	Erro da interpolação polinomial de décimo grau da função $f(x)=\dfrac{1}{1+25\cdot x^2}$