

Algoritmo de Newton e Newton-Gregory

Tabela de diferenças divididas

Seja o exemplo	x	-2	-1	0	1	2
	$f(x)$	-2	29	30	31	62

$x_i \ f(x_i)$

$$\begin{array}{cccccc}
 & -2 & -1 & 0 & 1 & 2 \\
 -2 & -2 & \frac{29 - (-2)}{(-1) - (-2)} = 31 & \frac{1 - 31}{0 - (-2)} = -15 & \frac{0 - (-15)}{1 - (-2)} = 5 & \frac{5 - 5}{2 - (-2)} = 0 \\
 -1 & 29 & \frac{30 - 29}{0 - (-1)} = 1 & \frac{1 - 1}{1 - (-1)} = 0 & \frac{15 - 0}{2 - (-1)} = 5 & \\
 0 & 30 & \frac{31 - 30}{1 - 0} = 1 & \frac{31 - 1}{2 - 0} = 15 & & \\
 1 & 31 & \frac{62 - 31}{2 - 1} = 31 & & & \\
 2 & 62 & & & &
 \end{array}$$

$$tb[i][j] = \frac{tb[i+1][j-1] - tb[i][j-1]}{tb[i+j-1][0] - tb[i][0]}$$

$$\begin{aligned}
 & \text{coluna anterior : linha de baixo - linha atual} \\
 & \text{coluna } \phi : \text{leque} - \text{linha atual}
 \end{aligned}$$

Polinômio gerado a partir da tabela

$$\begin{aligned}
 P(x) = & -2 + [(x - (-2))] \times 31 + [(x - (-2)) \times (x - (-1))] \times -15 \\
 & + [(x - (-2)) \times (x - (-1)) \times (x - 0)] \times 5 \\
 & + [(x - (-2)) \times (x - (-1)) \times (x - 0) \times (x - 1)] \times 0
 \end{aligned}$$

Algoritmo em Python para a tabela e o polinômio

```

# Polinomios de Newton (747)
import numpy as np
def polnew(a,b):
    t=len(a)
    tb=np.zeros((t,t+1),int)
    for i in range(t):
        tb[i][0]=a[i]
        tb[i][1]=b[i]
    j=2
    while j<t+1:
        for i in range(t-j+1):
            tb[i][j]=(tb[i+1][j-1]-tb[i][j-1])/(tb[i+j-1][0]-tb[i][0])
        j=j+1
    print(tb)
    print('----- o polinomio -----')
    print(str(tb[0][1]),' ',end='')
    for i in range(1,t):

```

```

aa=tb[0][i+1]
if aa<0:
    bb='('+str(aa)+')'
else:
    bb=str(aa)
print('+',bb,['',end=''])
for j in range(i):
    aa=tb[j,0]
    if aa<0:
        bb='('+str(aa)+')'
    else:
        bb=str(aa)
    print('x-',bb,' ',end='')
print(']',end='')

return tb

#polnew([-1,0,1,2,3],[1,1,0,-1,-2])
#polnew([0.9,1,1.1,1.2],[0.621,0.540,0.453,0.362]) - coseno
polnew([-2.0,-1,0,1,2],[-2.0,29,30,31,62])

#polnew([-3,-2,-1,2,4],[333,47,-7,83,1313])

```

Programa Python para o algoritmo de Newton-Gregory

```

# newton-gregory
import numpy as np
def ngreg(a,b):
    t=len(a)
    r=np.zeros((t,t+1),float)
    for i in range(t):
        r[i][0]=a[i]
        r[i][1]=b[i]
    j=2
    while j<=t:
        i=t-j
        while i>=0:
            r[i][j]=r[i+1][j-1]-r[i][j-1]
            i=i-1
        j=j+1
    return r
print/ngreg([3,4,5,6,7,8],[2, 5, 9, 15,14,20]))
```