Binômio de Newton

Mário Leite

...

Sir. **Issac Newton** foi um dos maiores cientistas de todos os tempos; matemático, físico, astrônomo, teólogo e autor. É considerado por muitos o pai da Física Clássica, ampliando e consolidando formalmente as leis do movimento; foi influenciador e figura-chave na Revolução Científica. Quem nunca ouviu falar na “Lei da Gravidade” ou no “Disco de Newton” nos cursos fundamentais na composição da cor branca ao girar um disco pintado com várias cores!?
Como matemático desenvolveu o Cálculo com o conceito de “limite”, possibilitando a computação de áreas não regulares e problemas complexos. Na Álgebra existe uma situação em que é preciso expandir uma soma algébrica elevada a uma potência a qual chamamos de “Binômio de Newton”, que já havia sido estudo muito antes por Chu Shi-Kié, em 1303 e Blaise Pascal em 1623. Entretanto foi Newton, em 1643, que o formalizou de modo mais compacto, baseado na série mostrada abaixo...

$$(a+b)^n = \binom{n}{0} a^n + \binom{n}{1} a^{n-1}b + \binom{n}{2} a^{n-2}b^2 + \ldots + \binom{n}{n-1} ab^{n-1} + \binom{n}{n} b^n$$

Então, o objetivo é calcular a potência de uma soma algébrica onde **n** é o expoente da potenciação do binômio. Este é um caso muito comum no Curso Médio quando **n=2**, cuja solução é: $\mathbf{a^2 + 2ab + b^2}$; mas... e quando n=3,4,5,...?! Para **n=3** ainda é fácil: $\mathbf{(a^2 + 2ab + b^2)(a+b)}$; mas, para **n>4** a coisa fica complicada e muito trabalhosa. Assim, a fórmula criada por Newton é uma solução muito boa e nos auxilia bastante. Por outro lado, devido aos fatores combinatórios na fórmula, nos obriga a criar um algoritmo computacional, pois fazer manualmente é bastante trabalhoso e sujeito a erros. Então, para resolver definitivamente o problema de maneira computacional o programa **“BinomioNewton”**, codificado em Python”, é uma boa solução. As **figuras 1**, **2**, **3** e **4** mostram as saídas para **n=2,3,4,5**, respectivamente. Os argumentos **a** e **b**, embora sejam passados com seus respectivos valores numéricos para a função **CalcularBinomioNewton()**, a saída do programa os deixou propositalmente, como letras para mostrar a forma geral da solução. Portanto, para saber os valores resultantes de multiplicações e potenciações basta substituir **a** e **b** por seus valores numéricos mostrados na expressão.

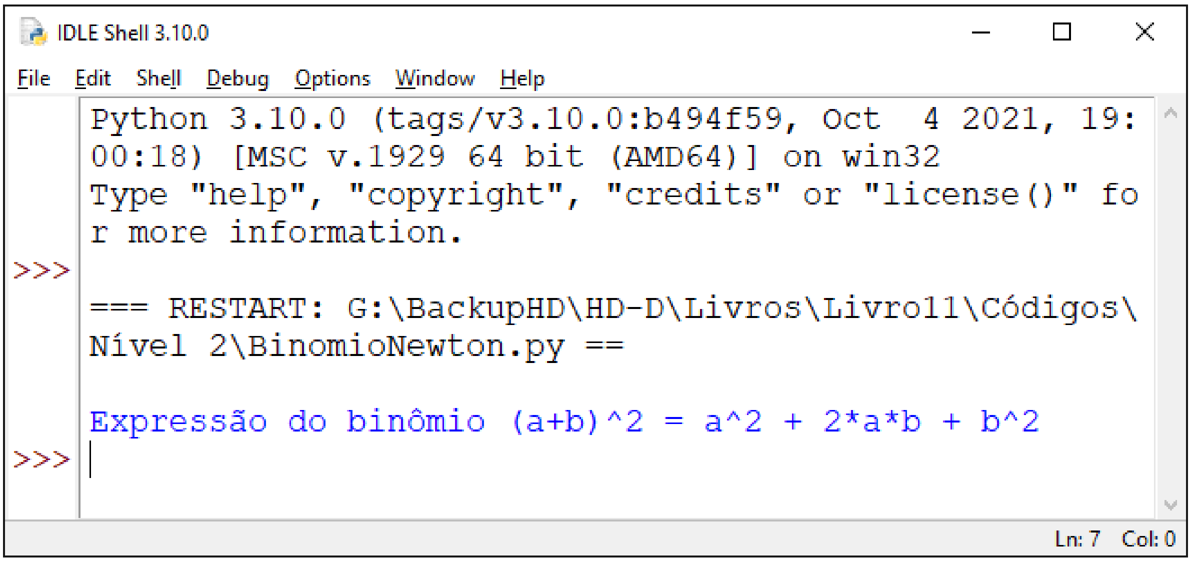

Figura 1- Solução do Binômio de Newton para n=2

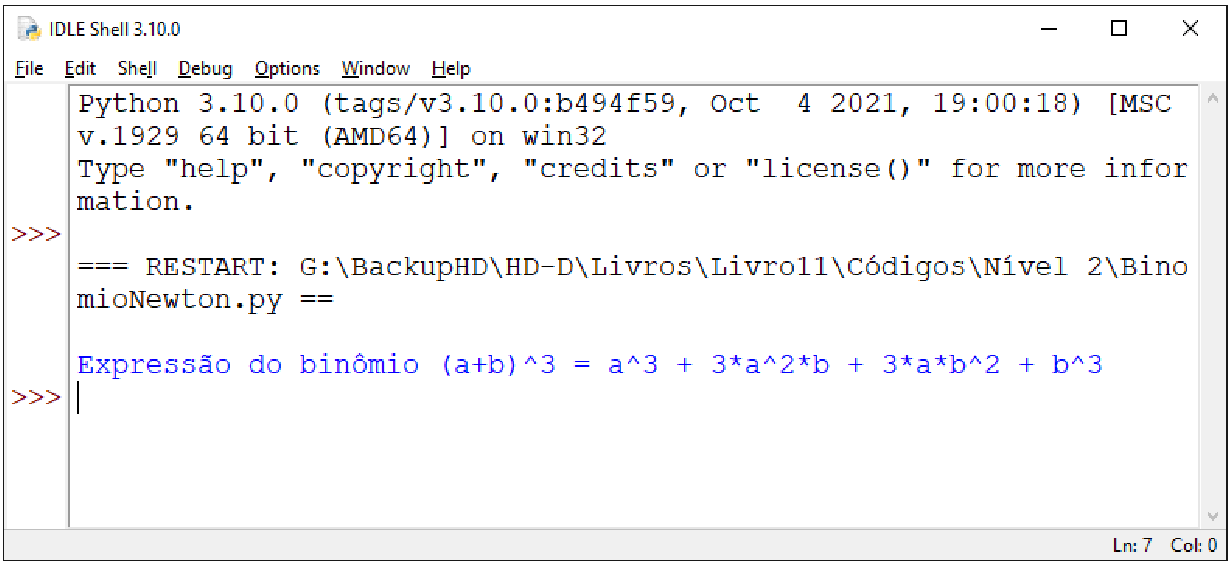

Figura 2 - Solução do Binômio de Newton para n=3

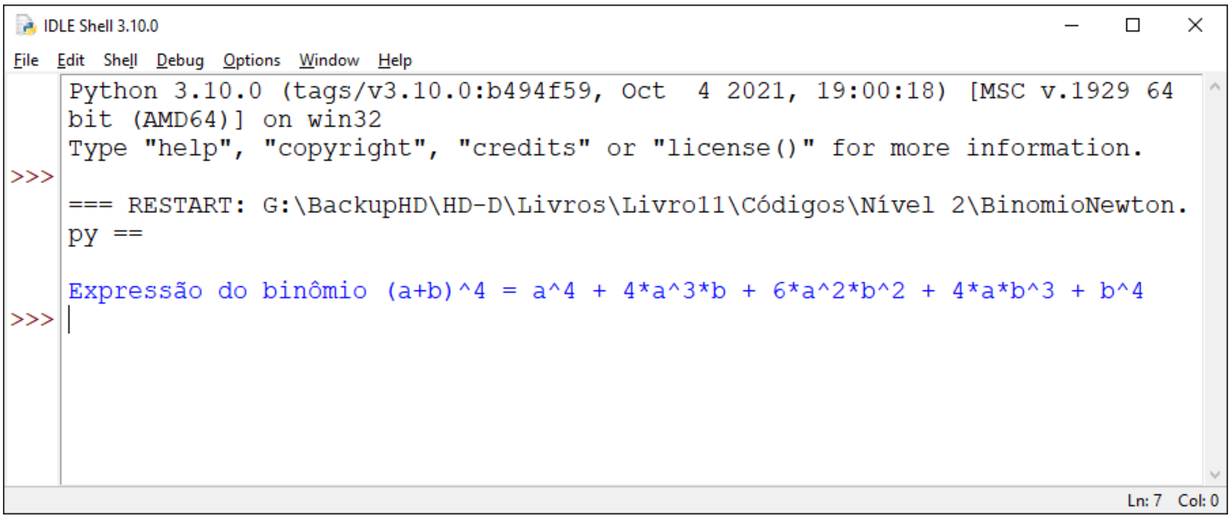

Figura 3 - Solução do Binômio de Newton para n=4

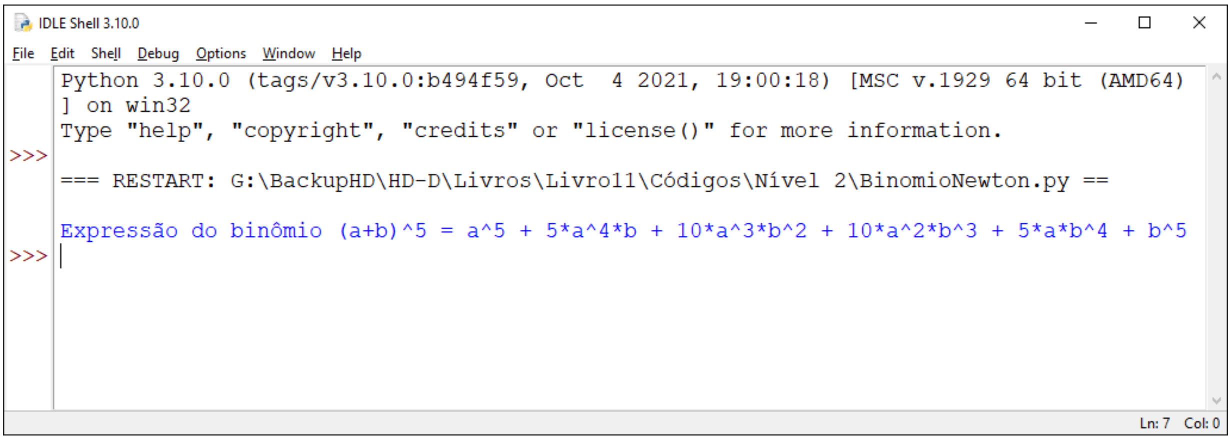

Figura 4 - Solução do Binômio de Newton para n=5

```python
'''
BinomioNewton.py
----------------------------------------------------------------------
Contém duas funções que calculam e mostram a fórmula geral do
"Binômio de Newton" do tipo (a+b)^n.
----------------------------------------------------------------------
Autor: Mário Leite
Data: 03/09/2023
----------------------------------------------------------------------
'''
import re

def CalcularCombinacoes(n,p):
    fat1 = 1
    for i in range(1,(p+1)):  #calcula [An,p]
        fat1 = fat1*(n-i+1)
    fat2 = 1
    for i in range(1,(p+1)):  #calcula fatorial de [p]
        fat2 = fat2*i

    comb = int(fat1/fat2)  #calcula [Cn,p]

    return comb
#----------------------------------------------------------------------

def CalcularBinomioNewton(n, a, b):
    #n: expoente do binômio
    #a,b: valores do binômio (interios positivos)
    termo = ""
    for k in range(0, (n+1)):
        if(k < n):
            termo = termo + str(CalcularCombinacoes(n, k)) + "*a^" +
                    str(n-k) + "*" + "b^" + str(k) + " + "
        else:
            termo = termo + str(CalcularCombinacoes(n, k)) + "*a^" +
                    str(n-k) + "*" + "b^" + str(k)
    return termo

#===============================================================-------===
#Calcula o termo usando a função "CalcularBinobioNewton"
n = 2  #um exemplo para (a+b)^2
termo = CalcularBinomioNewton(n, 3, 4)

#Remove os termos inúteis (com resultado igual a 1)
expresFinal = re.sub(r'\b1\*', '', termo)
expresFinal = re.sub(r'\b\w\^0\b|\b\w\^1\b', '', expresFinal)

#Mostra a expressão final do "Binômio de Newton"
print(f"Expressão do binâomio (a+b)^{n} = {expresFinal}")
#Fim do programa "BinomioNewton " ----------------------------------
```