

Lista01Q02g

June 19, 2025

Lista 01

Victor Paiva Paulo Neto

Questões : 1, 2(e), 2(a), 2(g), 2(c), 2(i)

Escreva um programa e envie ao arquivo fonte ou um link do mesmo, o qual tenha como ponto de partida uma funcao $a(n)$ que define o termo geral de uma sequência numerica $a_n = a(n)$ (use n como a variavel do programa), que sejam fornecidos dois números naturais n_{min} e n_{max} que definem um intervalo $[n_{min}, n_{max}]$ para os indices “plotados” e que tenha duas opcoes exclusivas de execucao:

```
[5]: import matplotlib.pyplot as plt
import math
```

```
[1]: #funcao que gera a sequencia numerica de termo geral g(n) 2(g)
def g(n):
    numerador = math.sqrt(math.factorial(n)) + math.exp(2 * n)
    demoninador = 5 * math.sqrt(math.factorial(n)) - math.exp(n)
    return numerador/demoninador
```

```
[8]: #diferença de nmin e nmax
nmin = int(input("Digite o nmin: "))
nmax = int(input("Digite nmax: "))
print(f"nmin: {nmin}, nmax: {nmax}")
```

nmin: 20, nmax: 100

```
[34]: print(f"nmin: {nmin}, nmax: {nmax}")
```

nmin: 10, nmax: 100

```
[3]: #separa nos casos em que o limite é e não é conhecido
existeLimite = (input("eh conhecido se a sequencia converge para um limite L?_
↪(a se nao, b se sim): "))
print(f"opcao escolhida: {existeLimite}")
```

opcao escolhida: a

```
[9]: valoresN = []
valoresAn = []
```

```

print(f"\n{'n':>5} {'a(n)':>10}")
print("-" * 15)

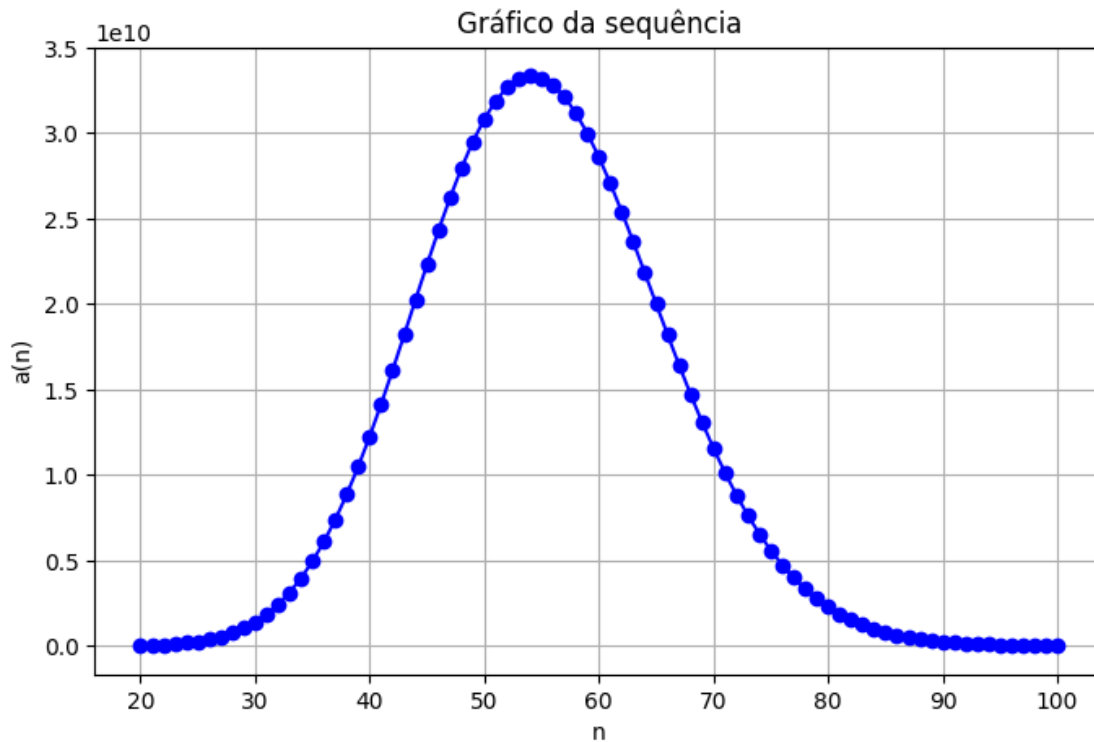
for n in range( nmin, nmax +1):
    gn = g(n)
    print(f"{'n':>5} {'gn:>10.6f}")
    valoresN.append(n)
    valoresAn.append(gn)

```

n	a(n)
20	32184085.852138
21	50530722.025275
22	78341590.922567
23	119570875.715020
24	179367247.113463
25	264254367.057711
26	382279434.252652
27	543104894.484086
28	758015585.119650
29	1039813350.117289
30	1402574274.064473
31	1861250446.925596
32	2431108460.989340
33	3127010105.526352
34	3962555997.664251
35	4949128840.603758
36	6094888038.131446
37	7403779810.604958
38	8874635126.166012
39	10500430345.172428
40	12267781620.784266
41	14156733576.483585
42	16140886065.093180
43	18187881065.335220
44	20260246742.103313
45	22316569549.979851
46	24312940344.619171
47	26204599061.622955
48	27947686588.948586
49	29501003433.709141
50	30827673435.474411
51	31896617139.356300
52	32683752830.296234
53	33172862341.279663
54	33356081815.806297
55	33234002596.662304

56 32815392243.928234
57 32116568419.838951
58 31160477401.389072
59 29975543121.580833
60 28594361235.536919
61 27052315617.301620
62 25386192254.399124
63 23632858452.674652
64 21828064610.627823
65 20005412748.455559
66 18195521699.699722
67 16425404540.969439
68 14718060440.525738
69 13092271427.886221
70 11562585157.082754
71 10139457839.514311
72 8829527211.271261
73 7635983531.386476
74 6559006888.560182
75 5596241131.185262
76 4743278084.367329
77 3994129925.912086
78 3341672236.823346
79 2778044950.763353
80 2295002901.678827
81 1884211687.559519
82 1537487986.384535
83 1246986203.727812
84 1005335386.432867
85 805731736.895369
86 641992875.583285
87 508580314.597106
88 400596521.523460
89 313762570.599612
90 244381791.171929
91 189294114.580301
92 145825059.269974
93 111732534.776664
94 85153928.506509
95 64555292.026407
96 48683881.628115
97 36524837.721812
98 27262408.553266
99 20245830.167464
100 14959757.539900

```
[10]: plt.figure(figsize=(8, 5))
plt.plot(valoresN, valoresAn, 'bo-', label='a(n)')
plt.xlabel('n')
plt.ylabel('a(n)')
plt.title('Gráfico da sequência')
plt.grid(True)
```



Limite:

esta sequencia não possui limite

```
[11]: #funcao que verifica se o modulo da diferenca entre o termo geral e o limite é
      <- menor ou igual a epsilon
def N_epsilon_is_true(x, L, epsilon):
    if abs(x - L) <= epsilon:
        return True
    else:
        return False
```

```
[12]: if existeLimite == "b":
    L = float(input("Digite o valor do limite L: "))
    epsilon = float(input("Digite a tolerância epsilon: "))
    N_epsilon = float(input("Digite o valor de N(epsilon): "))
```

```

Ne = N_epsilon_is_true(a(n), L, epsilon)

while Ne == False:
    L = float(input("Digite o valor do limite L: "))
    epsilon = float(input("Digite a tolerância epsilon: "))
    N_epsilon = int(input("Digite o valor de N(epsilon): "))
    Ne = N_epsilon_is_true(a(n), L, epsilon)

plt.axhline(y=L, color='green', linestyle='--', label='y = L')
plt.axhline(y=L+epsilon, color='red', linestyle=':', label='y = L + ')
plt.axhline(y=L-epsilon, color='red', linestyle=':', label='y = L - ')

plt.legend()
plt.show()

```