

# Lista01Q02a

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:

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

```
[2]: #funcao que gera a sequencia numerica de termo geral a(n) 2(a)
def a(n):
    return (n - 1) / (n + 1)
```

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

nmin: 10, nmax: 200

```
[4]: #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: b

```
[5]: valoresN = []
valoresAn = []
print(f"\n{'n':>5} {'a(n)':>10}")
print("-" * 15)

for n in range( nmin, nmax +1):
    an = a(n)
```

```
print(f"{n:>5} {an:>10.6f}")
valoresN.append(n)
valoresAn.append(an)
```

n	a(n)
10	0.818182
11	0.833333
12	0.846154
13	0.857143
14	0.866667
15	0.875000
16	0.882353
17	0.888889
18	0.894737
19	0.900000
20	0.904762
21	0.909091
22	0.913043
23	0.916667
24	0.920000
25	0.923077
26	0.925926
27	0.928571
28	0.931034
29	0.933333
30	0.935484
31	0.937500
32	0.939394
33	0.941176
34	0.942857
35	0.944444
36	0.945946
37	0.947368
38	0.948718
39	0.950000
40	0.951220
41	0.952381
42	0.953488
43	0.954545
44	0.955556
45	0.956522
46	0.957447
47	0.958333
48	0.959184
49	0.960000
50	0.960784

51	0.961538
52	0.962264
53	0.962963
54	0.963636
55	0.964286
56	0.964912
57	0.965517
58	0.966102
59	0.966667
60	0.967213
61	0.967742
62	0.968254
63	0.968750
64	0.969231
65	0.969697
66	0.970149
67	0.970588
68	0.971014
69	0.971429
70	0.971831
71	0.972222
72	0.972603
73	0.972973
74	0.973333
75	0.973684
76	0.974026
77	0.974359
78	0.974684
79	0.975000
80	0.975309
81	0.975610
82	0.975904
83	0.976190
84	0.976471
85	0.976744
86	0.977011
87	0.977273
88	0.977528
89	0.977778
90	0.978022
91	0.978261
92	0.978495
93	0.978723
94	0.978947
95	0.979167
96	0.979381
97	0.979592
98	0.979798

99	0.980000
100	0.980198
101	0.980392
102	0.980583
103	0.980769
104	0.980952
105	0.981132
106	0.981308
107	0.981481
108	0.981651
109	0.981818
110	0.981982
111	0.982143
112	0.982301
113	0.982456
114	0.982609
115	0.982759
116	0.982906
117	0.983051
118	0.983193
119	0.983333
120	0.983471
121	0.983607
122	0.983740
123	0.983871
124	0.984000
125	0.984127
126	0.984252
127	0.984375
128	0.984496
129	0.984615
130	0.984733
131	0.984848
132	0.984962
133	0.985075
134	0.985185
135	0.985294
136	0.985401
137	0.985507
138	0.985612
139	0.985714
140	0.985816
141	0.985915
142	0.986014
143	0.986111
144	0.986207
145	0.986301
146	0.986395

147	0.986486
148	0.986577
149	0.986667
150	0.986755
151	0.986842
152	0.986928
153	0.987013
154	0.987097
155	0.987179
156	0.987261
157	0.987342
158	0.987421
159	0.987500
160	0.987578
161	0.987654
162	0.987730
163	0.987805
164	0.987879
165	0.987952
166	0.988024
167	0.988095
168	0.988166
169	0.988235
170	0.988304
171	0.988372
172	0.988439
173	0.988506
174	0.988571
175	0.988636
176	0.988701
177	0.988764
178	0.988827
179	0.988889
180	0.988950
181	0.989011
182	0.989071
183	0.989130
184	0.989189
185	0.989247
186	0.989305
187	0.989362
188	0.989418
189	0.989474
190	0.989529
191	0.989583
192	0.989637
193	0.989691
194	0.989744

```

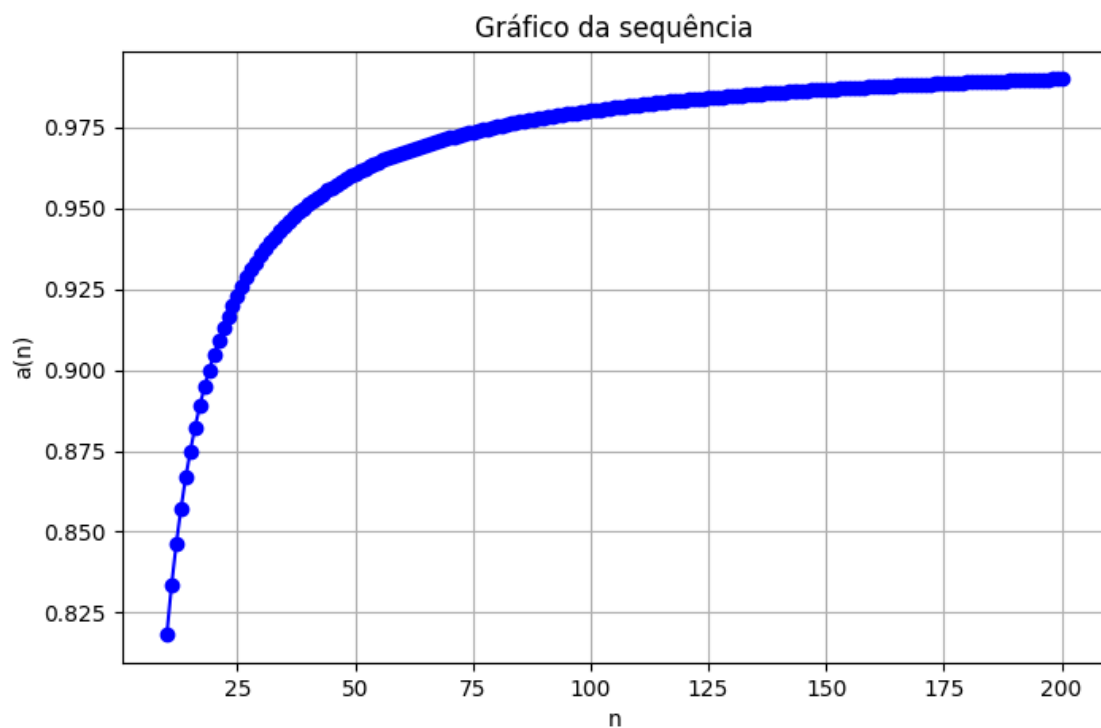
195  0.989796
196  0.989848
197  0.989899
198  0.989950
199  0.990000
200  0.990050

```

```

[6]: 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:

Como verificado, a sequencia possui limite e ele é igual a 1.

```

[7]: #funcao que verifica se o modulo da diferença 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
```

```
[8]: 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()
```

