

Lista01Q02c

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 c(n) 2(c)
def c(n):
    return math.log(n) / math.exp(n)
```

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

nmin: 20, nmax: 280

```
[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} {'c(n)':>10}")
print("-" * 15)

for n in range( nmin, nmax +1):
    cn = c(n)
```

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

n	c(n)
20	0.000000
21	0.000000
22	0.000000
23	0.000000
24	0.000000
25	0.000000
26	0.000000
27	0.000000
28	0.000000
29	0.000000
30	0.000000
31	0.000000
32	0.000000
33	0.000000
34	0.000000
35	0.000000
36	0.000000
37	0.000000
38	0.000000
39	0.000000
40	0.000000
41	0.000000
42	0.000000
43	0.000000
44	0.000000
45	0.000000
46	0.000000
47	0.000000
48	0.000000
49	0.000000
50	0.000000
51	0.000000
52	0.000000
53	0.000000
54	0.000000
55	0.000000
56	0.000000
57	0.000000
58	0.000000
59	0.000000
60	0.000000

61	0.000000
62	0.000000
63	0.000000
64	0.000000
65	0.000000
66	0.000000
67	0.000000
68	0.000000
69	0.000000
70	0.000000
71	0.000000
72	0.000000
73	0.000000
74	0.000000
75	0.000000
76	0.000000
77	0.000000
78	0.000000
79	0.000000
80	0.000000
81	0.000000
82	0.000000
83	0.000000
84	0.000000
85	0.000000
86	0.000000
87	0.000000
88	0.000000
89	0.000000
90	0.000000
91	0.000000
92	0.000000
93	0.000000
94	0.000000
95	0.000000
96	0.000000
97	0.000000
98	0.000000
99	0.000000
100	0.000000
101	0.000000
102	0.000000
103	0.000000
104	0.000000
105	0.000000
106	0.000000
107	0.000000
108	0.000000

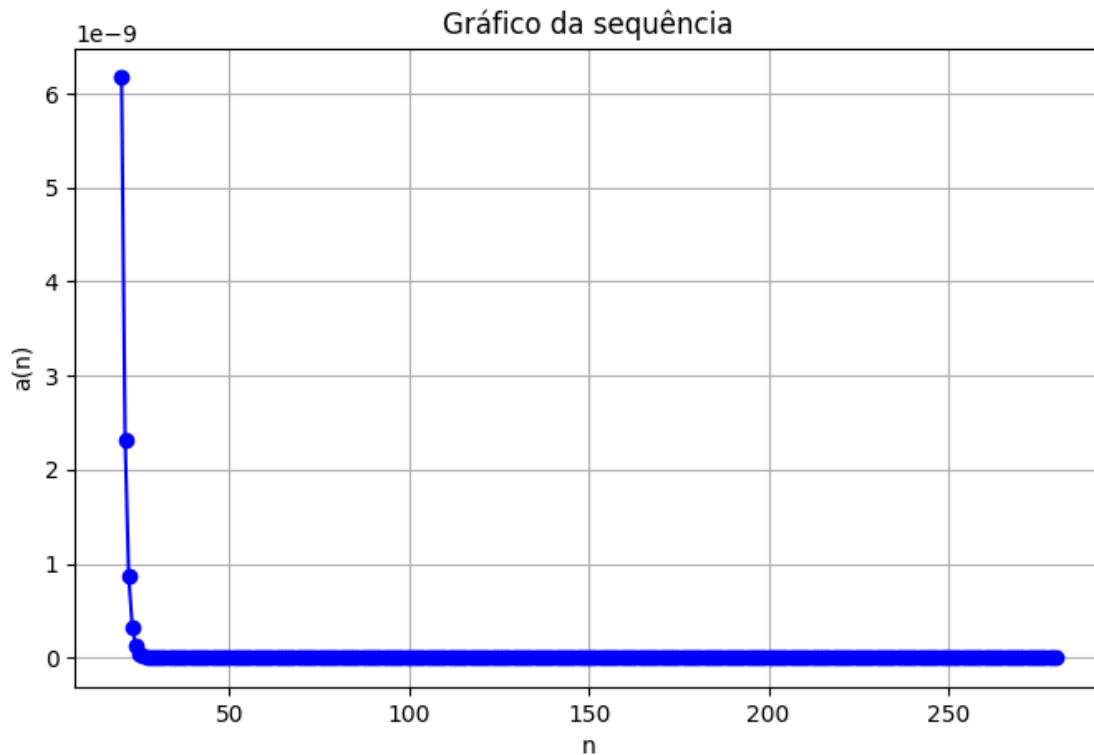
109	0.000000
110	0.000000
111	0.000000
112	0.000000
113	0.000000
114	0.000000
115	0.000000
116	0.000000
117	0.000000
118	0.000000
119	0.000000
120	0.000000
121	0.000000
122	0.000000
123	0.000000
124	0.000000
125	0.000000
126	0.000000
127	0.000000
128	0.000000
129	0.000000
130	0.000000
131	0.000000
132	0.000000
133	0.000000
134	0.000000
135	0.000000
136	0.000000
137	0.000000
138	0.000000
139	0.000000
140	0.000000
141	0.000000
142	0.000000
143	0.000000
144	0.000000
145	0.000000
146	0.000000
147	0.000000
148	0.000000
149	0.000000
150	0.000000
151	0.000000
152	0.000000
153	0.000000
154	0.000000
155	0.000000
156	0.000000

157	0.000000
158	0.000000
159	0.000000
160	0.000000
161	0.000000
162	0.000000
163	0.000000
164	0.000000
165	0.000000
166	0.000000
167	0.000000
168	0.000000
169	0.000000
170	0.000000
171	0.000000
172	0.000000
173	0.000000
174	0.000000
175	0.000000
176	0.000000
177	0.000000
178	0.000000
179	0.000000
180	0.000000
181	0.000000
182	0.000000
183	0.000000
184	0.000000
185	0.000000
186	0.000000
187	0.000000
188	0.000000
189	0.000000
190	0.000000
191	0.000000
192	0.000000
193	0.000000
194	0.000000
195	0.000000
196	0.000000
197	0.000000
198	0.000000
199	0.000000
200	0.000000
201	0.000000
202	0.000000
203	0.000000
204	0.000000

205	0.000000
206	0.000000
207	0.000000
208	0.000000
209	0.000000
210	0.000000
211	0.000000
212	0.000000
213	0.000000
214	0.000000
215	0.000000
216	0.000000
217	0.000000
218	0.000000
219	0.000000
220	0.000000
221	0.000000
222	0.000000
223	0.000000
224	0.000000
225	0.000000
226	0.000000
227	0.000000
228	0.000000
229	0.000000
230	0.000000
231	0.000000
232	0.000000
233	0.000000
234	0.000000
235	0.000000
236	0.000000
237	0.000000
238	0.000000
239	0.000000
240	0.000000
241	0.000000
242	0.000000
243	0.000000
244	0.000000
245	0.000000
246	0.000000
247	0.000000
248	0.000000
249	0.000000
250	0.000000
251	0.000000
252	0.000000

```
253 0.000000
254 0.000000
255 0.000000
256 0.000000
257 0.000000
258 0.000000
259 0.000000
260 0.000000
261 0.000000
262 0.000000
263 0.000000
264 0.000000
265 0.000000
266 0.000000
267 0.000000
268 0.000000
269 0.000000
270 0.000000
271 0.000000
272 0.000000
273 0.000000
274 0.000000
275 0.000000
276 0.000000
277 0.000000
278 0.000000
279 0.000000
280 0.000000
```

```
[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 visto no grafico, o limite existe e é 0.

```
[7]: #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
```

```
[ ]: 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(c(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): "))
```



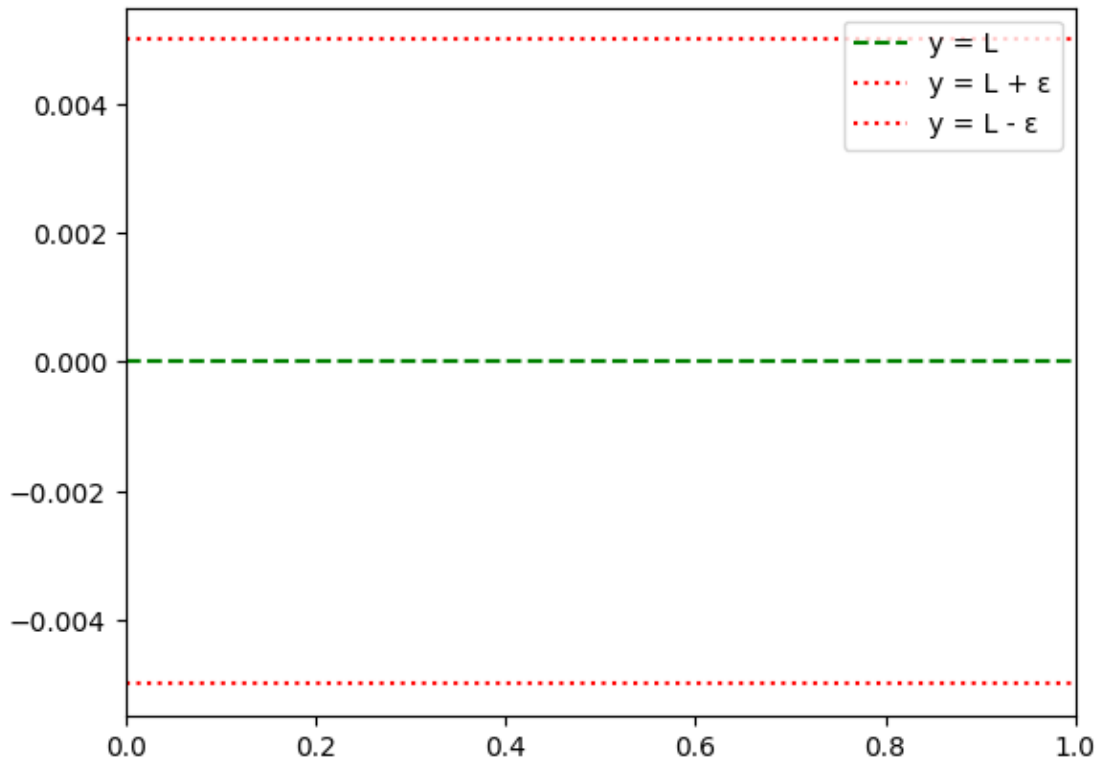
```

Ne = N_epsilon_is_true(c(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()

```



The Kernel crashed while executing code in the current cell or a previous cell.

Please review the code in the cell(s) to identify a possible cause of the failure.

Click [here](https://aka.ms/vscodeJupyterKernelCrash) for more info.

View Jupyter [log](command:jupyter.viewOutput) for further details.