t\_steps = [1, .5, .25, .125]  
results = []  
c\_temp = T\_init  
c\_time = 0  
  
def Num(K, T, O, dt):  
 # Regular old dT  
 return -K \* (T - O) \* dt  
  
# Warming  
for current\_step in t\_steps:  
 while c\_time <= time:  
 c\_temp += Num(K\_, c\_temp, T\_out, current\_step)  
 c\_time += current\_step  
 results.append(c\_temp)  
 c\_temp = T\_init  
 c\_time = 0  
  
print("Numerical solutions for each time step:", results)  
print("\nConverge to analytical solution", a\_sol)  
print("Accuracy:", a\_sol/results)

Console:

Analytical solution: 18.04753455623894

Numerical solutions for each time step: [18.076069576879544, 18.06180286283447, 18.054668908543196, 18.05110178213491]

Converge to analytical solution 18.04753455623894

Accuracy: [0.99842139 0.99921003 0.99960485 0.99980238]