

Conclusiones.

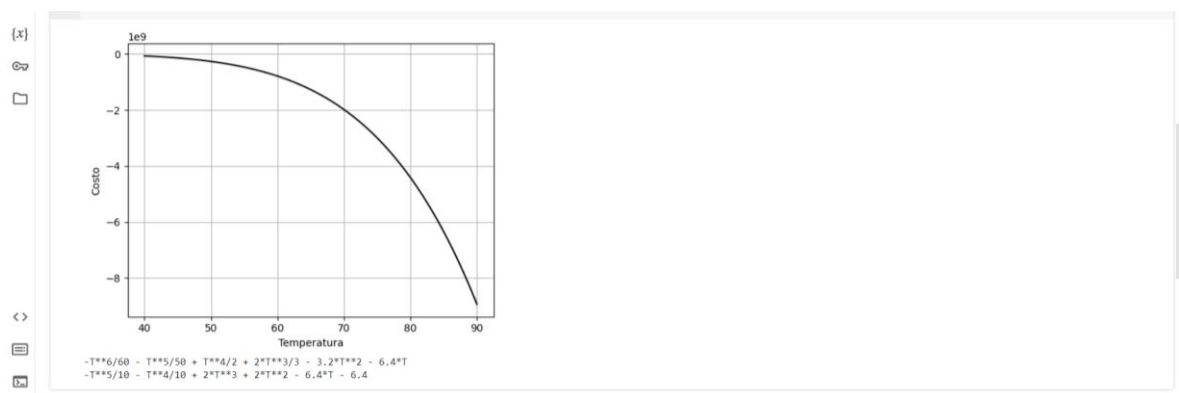
Antes de dar la conclusión mostraremos por cada método, la prueba de las funciones.

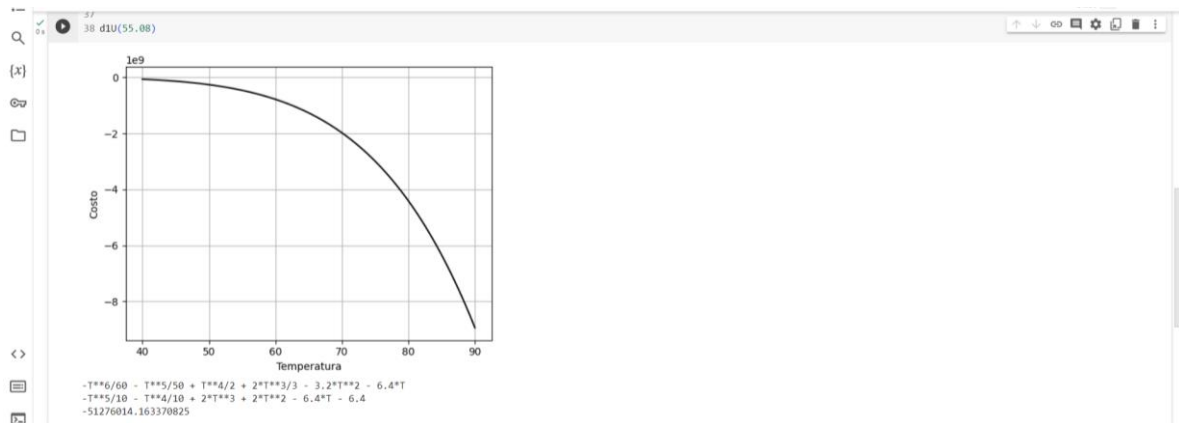
A continuación, mostraremos los métodos de cada función.

$$(-T^{**6}/60) - (T^{**5}/50) + (T^{**4}/2) + (2*T^{**3}/3) - 3.2*T^{**2} - 6.4*T$$

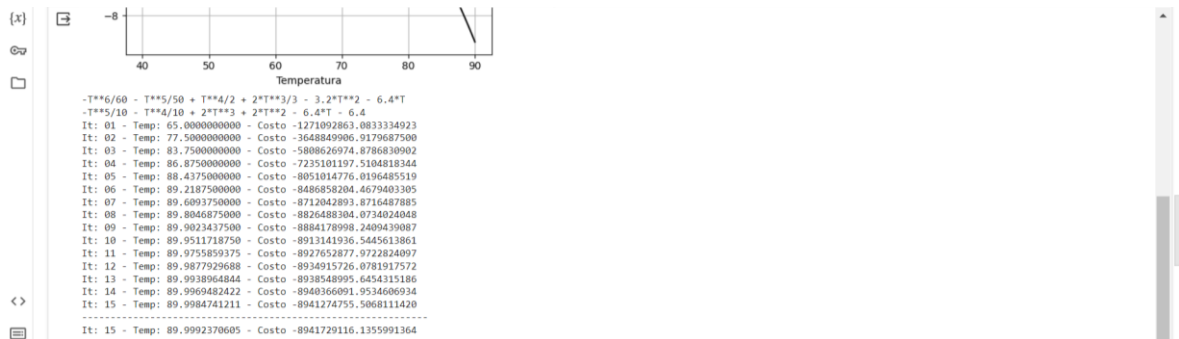
Gradiente.

```
1 import matplotlib.pyplot as plt
2 import numpy as np
3 import sympy
4
5 a = 40
6 b = 90
7 T = np.linspace(a, b, 100)
8 U = (-T**6/60) - (T**5/50) + (T**4/2) + (2*T**3/3) - 3.2*T**2 - 6.4*T
9
10 plt.figure()
11 plt.plot(T, U, 'k')
12 plt.xlabel('Temperatura')
13 plt.ylabel('Costo')
14 plt.grid()
15 plt.show()
16
17 def calcularGradiente():
18     T = sympy.Symbol('T')
19     fU = (-T**6/60) - (T**5/50) + (T**4/2) + (2*T**3/3) - 3.2*T**2 - 6.4*T
20
21     difU = sympy.diff(fU)
22
23     print(fU)
24     print(difU)
25
26     return None
27
28 calcularGradiente()
29
30
```

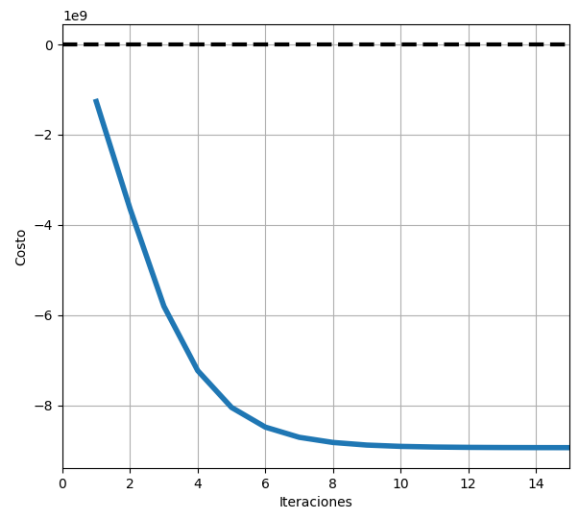
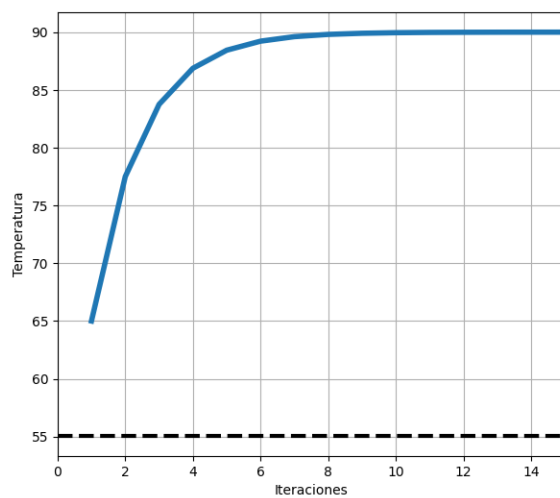




Método de la bisección con la evaluación del método.



Analisis de convergencia



Método de Newton – Raphson.



It: 49	-	Temp: 41.0000000000	-	Costo: -80032341.8033333421
It: 50	-	Temp: 40.0000000000	-	Costo: -68997376.0000000000
It: 51	-	Temp: 39.0000000000	-	Costo: -59259063.6300000027
It: 52	-	Temp: 38.0000000000	-	Costo: -50692691.0933333337
It: 53	-	Temp: 37.0000000000	-	Costo: -43182754.3900000006
It: 54	-	Temp: 36.0000000000	-	Costo: -36622494.7200000063
It: 55	-	Temp: 35.0000000000	-	Costo: -30913446.0033333358
It: 56	-	Temp: 34.0000000000	-	Costo: -25064994.8799999990
It: 57	-	Temp: 33.0000000000	-	Costo: -21693951.5099999979
It: 58	-	Temp: 32.0000000000	-	Costo: -18024133.9733333364
It: 59	-	Temp: 31.0000000000	-	Costo: -14885963.4700000007
It: 60	-	Temp: 30.0000000000	-	Costo: -12216072.0000000000
It: 61	-	Temp: 29.0000000000	-	Costo: -9956921.9633333329
It: 62	-	Temp: 28.0000000000	-	Costo: -8056437.7599999998
It: 63	-	Temp: 27.0000000000	-	Costo: -6467649.3899999997
It: 64	-	Temp: 26.0000000000	-	Costo: -5148348.0533333337
It: 65	-	Temp: 25.0000000000	-	Costo: -4060753.7499999995
It: 66	-	Temp: 24.0000000000	-	Costo: -3171194.8000000004
It: 67	-	Temp: 23.0000000000	-	Costo: -2449799.8433333333
It: 68	-	Temp: 22.0000000000	-	Costo: -1870200.6399999999
It: 69	-	Temp: 21.0000000000	-	Costo: -1409248.4700000000
It: 70	-	Temp: 20.0000000000	-	Costo: -1046741.3333333334
It: 71	-	Temp: 19.0000000000	-	Costo: -765163.6300000000
It: 72	-	Temp: 18.0000000000	-	Costo: -549437.7600000000
It: 73	-	Temp: 17.0000000000	-	Costo: -386687.7233333333
It: 74	-	Temp: 16.0000000000	-	Costo: -266014.7200000000
It: 75	-	Temp: 15.0000000000	-	Costo: -178284.7500000000
It: 76	-	Temp: 14.0000000000	-	Costo: -115928.2133333333
It: 77	-	Temp: 13.0000000000	-	Costo: -72751.5100000000
It: 78	-	Temp: 12.0000000000	-	Costo: -43760.6400000000
It: 79	-	Temp: 11.0000000000	-	Costo: -24996.8033333333
It: 80	-	Temp: 10.0000000000	-	Costo: -13384.0000000000
It: 81	-	Temp: 9.0000000000	-	Costo: -6588.6300000000

Método de la secante.

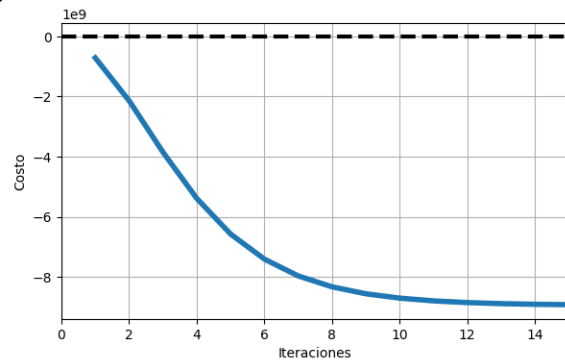
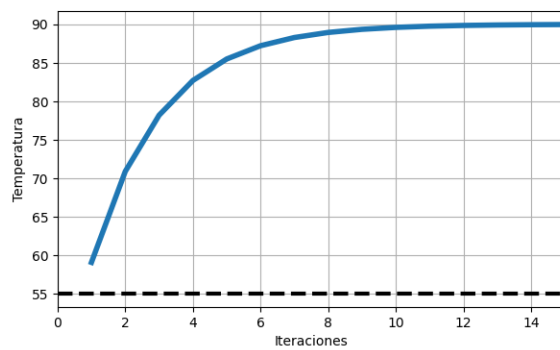


It: 735593	-	Temp: 4.00041595430	-	Costo: 5.1195830132
It: 735594	-	Temp: 4.00041595169	-	Costo: 5.1195836190
It: 735595	-	Temp: 4.00041594879	-	Costo: 5.1195836148
It: 735596	-	Temp: 4.00041594590	-	Costo: 5.1195836306
It: 735597	-	Temp: 4.00041594300	-	Costo: 5.1195836364
It: 735598	-	Temp: 4.00041594011	-	Costo: 5.1195836422
It: 735599	-	Temp: 4.00041593721	-	Costo: 5.1195836481
It: 735600	-	Temp: 4.00041593432	-	Costo: 5.1195836539
It: 735601	-	Temp: 4.00041593142	-	Costo: 5.1195836597
It: 735602	-	Temp: 4.00041592853	-	Costo: 5.1195836655
It: 735603	-	Temp: 4.00041592563	-	Costo: 5.1195836713
It: 735604	-	Temp: 4.00041592274	-	Costo: 5.1195836771
It: 735605	-	Temp: 4.00041591985	-	Costo: 5.1195836829
It: 735606	-	Temp: 4.00041591695	-	Costo: 5.1195836887
It: 735607	-	Temp: 4.00041591406	-	Costo: 5.1195836945
It: 735608	-	Temp: 4.00041591116	-	Costo: 5.1195837003
It: 735609	-	Temp: 4.00041590827	-	Costo: 5.1195837061
It: 735610	-	Temp: 4.00041590537	-	Costo: 5.1195837119
It: 735611	-	Temp: 4.00041590248	-	Costo: 5.1195837177
It: 735612	-	Temp: 4.00041589958	-	Costo: 5.1195837235
It: 735613	-	Temp: 4.00041589669	-	Costo: 5.1195837293
It: 735614	-	Temp: 4.00041589380	-	Costo: 5.1195837351
It: 735615	-	Temp: 4.00041589090	-	Costo: 5.1195837409
It: 735616	-	Temp: 4.00041588801	-	Costo: 5.1195837467
It: 735617	-	Temp: 4.00041588511	-	Costo: 5.1195837525
It: 735618	-	Temp: 4.00041588222	-	Costo: 5.1195837583
It: 735619	-	Temp: 4.00041587932	-	Costo: 5.1195837641
It: 735620	-	Temp: 4.00041587643	-	Costo: 5.1195837699
It: 735621	-	Temp: 4.00041587354	-	Costo: 5.1195837757
It: 735622	-	Temp: 4.00041587064	-	Costo: 5.1195837815
It: 735623	-	Temp: 4.00041586775	-	Costo: 5.1195837873
It: 735624	-	Temp: 4.00041586485	-	Costo: 5.1195837931
It: 735625	-	Temp: 4.00041586196	-	Costo: 5.1195837989

Sección dorada.

It: 44	- Temp: 89.9999999681	- Costo: -8942183477.0131931305
It: 45	- Temp: 89.9999999803	- Costo: -8942183484.2655105591
It: 46	- Temp: 89.9999999878	- Costo: -8942183488.7476882935
It: 47	- Temp: 89.9999999925	- Costo: -8942183491.5178318024
It: 48	- Temp: 89.9999999953	- Costo: -8942183493.229698425
It: 49	- Temp: 89.9999999971	- Costo: -8942183494.2879676819
It: 50	- Temp: 89.9999999982	- Costo: -8942183494.9419059753
It: 51	- Temp: 89.9999999989	- Costo: -8942183495.3460617065
It: 52	- Temp: 89.9999999993	- Costo: -8942183495.5958442688
It: 53	- Temp: 89.9999999996	- Costo: -8942183495.7502212524
It: 54	- Temp: 89.9999999997	- Costo: -8942183495.8456306458
It: 55	- Temp: 89.9999999998	- Costo: -8942183495.9045963287
It: 56	- Temp: 89.9999999999	- Costo: -8942183495.9410419464
It: 57	- Temp: 89.9999999999	- Costo: -8942183495.9635658264
It: 58	- Temp: 90.0000000000	- Costo: -8942183495.9774875641
It: 59	- Temp: 90.0000000000	- Costo: -8942183495.9860858917
It: 60	- Temp: 90.0000000000	- Costo: -8942183495.9913997650
It: 61	- Temp: 90.0000000000	- Costo: -8942183495.9946842194
It: 62	- Temp: 90.0000000000	- Costo: -8942183495.9967155457
It: 63	- Temp: 90.0000000000	- Costo: -8942183495.9976686737
It: 64	- Temp: 90.0000000000	- Costo: -8942183495.9987487793
It: 65	- Temp: 90.0000000000	- Costo: -8942183495.9992198944
It: 66	- Temp: 90.0000000000	- Costo: -8942183495.9995269775
It: 67	- Temp: 90.0000000000	- Costo: -8942183495.9997119904
It: 68	- Temp: 90.0000000000	- Costo: -8942183495.9998226166
It: 69	- Temp: 90.0000000000	- Costo: -8942183495.9998893738
It: 70	- Temp: 90.0000000000	- Costo: -8942183495.9999332428
It: 71	- Temp: 90.0000000000	- Costo: -8942183495.9999580383
It: 72	- Temp: 90.0000000000	- Costo: -8942183495.9999752045
It: 73	- Temp: 90.0000000000	- Costo: -8942183495.9999828339
It: 73	- Temp: 90.0000000000	- Costo: -8942183495.9999828339

Convergencia

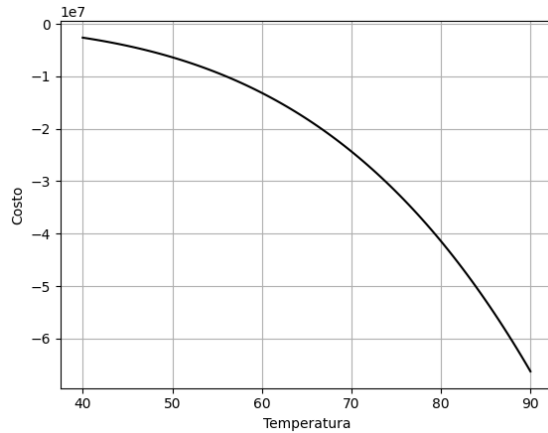


Método de Fibonacci.

It: 31	- Temp: 139.9999770464	- Costo: -126373944744.5605163574
It: 32	- Temp: 120.9016764839	- Costo: -52461313406.8552474976
It: 33	- Temp: 109.0982776089	- Costo: -28340675740.7671775818
It: 34	- Temp: 101.8033759214	- Costo: -18717649257.777674817
It: 35	- Temp: 97.2948787339	- Costo: -14267006261.7457040699
It: 36	- Temp: 94.5084742339	- Costo: -11986455339.8435935974
It: 37	- Temp: 92.7863815464	- Costo: -10735391116.4061050415
It: 38	- Temp: 91.7220697339	- Costo: -10018077410.2232456207
It: 39	- Temp: 91.0642888589	- Costo: -9595063882.8000958387
It: 40	- Temp: 90.6577579214	- Costo: -9341148317.8631136749
It: 41	- Temp: 90.4065079839	- Costo: -9187037473.8083686829
It: 42	- Temp: 90.2512269839	- Costo: -9092855390.4844398499
It: 43	- Temp: 90.1552580465	- Costo: -9035050900.6912918091
It: 44	- Temp: 90.0950459838	- Costo: -8999479082.1806716919
It: 45	- Temp: 90.0592891091	- Costo: -897552883.9023704529
It: 46	- Temp: 90.0366339211	- Costo: -8964024018.5475654602
It: 47	- Temp: 90.0226322345	- Costo: -8955671209.5881271362
It: 48	- Temp: 90.0139787330	- Costo: -8950512142.7411308289
It: 49	- Temp: 90.0086305478	- Costo: -8947324886.0382270813
It: 50	- Temp: 90.0053252316	- Costo: -8945355552.1395130157
It: 51	- Temp: 90.0032823626	- Costo: -8944138574.9620780945
It: 52	- Temp: 90.0020199154	- Costo: -8943386579.3754730225
It: 53	- Temp: 90.0012394936	- Costo: -8942921735.7220461670
It: 54	- Temp: 90.0007574883	- Costo: -8942634636.4648197754
It: 55	- Temp: 90.0004590717	- Costo: -8942456912.1994724274
It: 56	- Temp: 90.0002754430	- Costo: -8942347544.8839111328
It: 57	- Temp: 90.0001606751	- Costo: -8942279190.8775730133
It: 58	- Temp: 90.0000918143	- Costo: -8942238178.6827106476
It: 59	- Temp: 90.0000459072	- Costo: -8942197166.6445579529
It: 60	- Temp: 90.0000129536	- Costo: -8942183495.9999923706
It: 61	- Temp: 90.0000000000	- Costo: -8942183495.9999923706
It: 62	- Temp: 90.0000000000	- Costo: -8942183495.9999923706

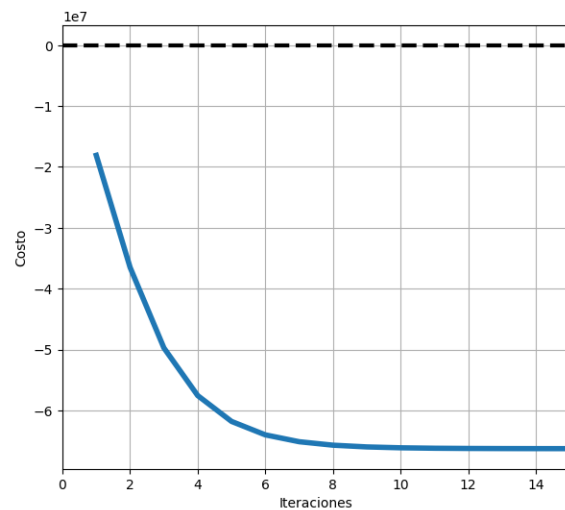
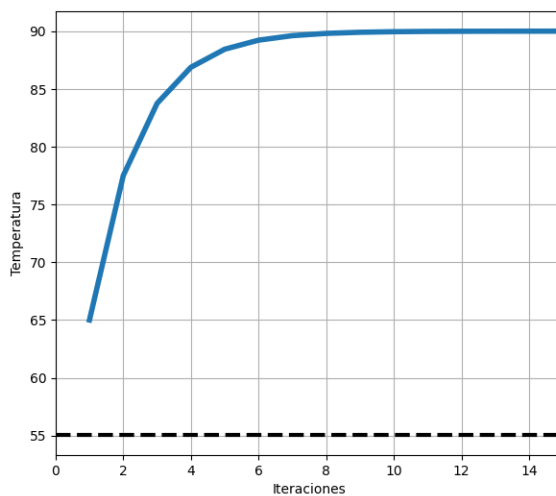
$$-(T^{**6}+T^{**5}-10*T-12)/(T^{**2}+6)$$

Método del gradiente y método de la bisección.



```
(-T**6 - T**5 + 10*T + 12)/(T**2 + 6)
-2*T*(-T**6 - T**5 + 10*T + 12)/(T**2 + 6)**2 + (-6*T**5 - 5*T**4 + 10)/(T**2 + 6)
It: 01 - Temp: 65.0000000000 - Costo -18099546.3455447890
It: 02 - Temp: 77.5000000000 - Costo -36504057.2347264960
It: 03 - Temp: 83.7500000000 - Costo -49741949.8909051324
It: 04 - Temp: 86.8750000000 - Costo -57571118.5405857340
It: 05 - Temp: 88.4375000000 - Costo -61815300.1193899438
It: 06 - Temp: 89.2187500000 - Costo -64023287.7781422958
It: 07 - Temp: 89.6093750000 - Costo -65149200.353024896
It: 08 - Temp: 89.8046875000 - Costo -65717692.3521447256
It: 09 - Temp: 89.9023437500 - Costo -66003329.3109137192
It: 10 - Temp: 89.9511718750 - Costo -66146496.4109141901
It: 11 - Temp: 89.975859375 - Costo -66218167.2262745649
It: 12 - Temp: 89.987929688 - Costo -66254024.4640775546
It: 13 - Temp: 89.993964844 - Costo -66271958.5422330275
It: 14 - Temp: 89.9969482422 - Costo -66280926.9463397041
It: 15 - Temp: 89.9984741211 - Costo -66285411.4896772057
.....
It: 15 - Temp: 89.9992370605 - Costo -66287653.8466703668
```

Analisis de convergencia



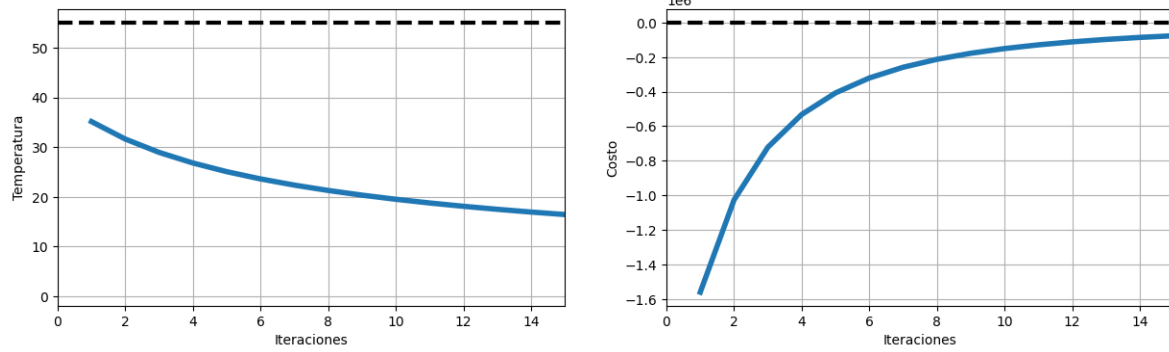
Método de Newton – Raphson.

```
It: 345942 - Temp: -345852.0000000000 - Costo: -14307373348034915074048.0000000000
It: 345943 - Temp: -345853.0000000000 - Costo: -14307538822813068558336.0000000000
It: 345944 - Temp: -345854.0000000000 - Costo: -14307704289025590466048.0000000000
It: 345945 - Temp: -345855.0000000000 - Costo: -14307869776575493380096.0000000000
It: 345946 - Temp: -345856.0000000000 - Costo: -14308035255759779397632.0000000000
It: 345947 - Temp: -345857.0000000000 - Costo: -14308200736279461101568.0000000000
It: 345948 - Temp: -345858.0000000000 - Costo: -14308366218234544783360.0000000000
It: 345949 - Temp: -345859.0000000000 - Costo: -14308531701625038811616.0000000000
It: 345950 - Temp: -345860.0000000000 - Costo: -14308697186450953732096.0000000000
It: 345951 - Temp: -345861.0000000000 - Costo: -14308862672712295776256.0000000000
It: 345952 - Temp: -345862.0000000000 - Costo: -14309028160409071255552.0000000000
It: 345953 - Temp: -345863.0000000000 - Costo: -14309193649541294850048.0000000000
It: 345954 - Temp: -345864.0000000000 - Costo: -14309359140108968656096.0000000000
It: 345955 - Temp: -345865.0000000000 - Costo: -14309524632112105259008.0000000000
It: 345956 - Temp: -345866.0000000000 - Costo: -14309690125550706753536.0000000000
It: 345957 - Temp: -345867.0000000000 - Costo: -14309855620424792014848.0000000000
It: 345958 - Temp: -345868.0000000000 - Costo: -14310021116734356848640.0000000000
It: 345959 - Temp: -345869.0000000000 - Costo: -1431018661447942226432.0000000000
It: 345960 - Temp: -345870.0000000000 - Costo: -14310352113659981856768.0000000000
It: 345961 - Temp: -345871.0000000000 - Costo: -14310517614276058808320.0000000000
It: 345962 - Temp: -345872.0000000000 - Costo: -14310683116327650983936.0000000000
It: 345963 - Temp: -345873.0000000000 - Costo: -14310848619814770966528.0000000000
It: 345964 - Temp: -345874.0000000000 - Costo: -14311014124737427144704.0000000000
It: 345965 - Temp: -345875.0000000000 - Costo: -1431117961095630084224.0000000000
It: 345966 - Temp: -345876.0000000000 - Costo: -14311345138889379545088.0000000000
It: 345967 - Temp: -345877.0000000000 - Costo: -14311510648118692544512.0000000000
It: 345968 - Temp: -345878.0000000000 - Costo: -14311676158783571099648.0000000000
It: 345969 - Temp: -345879.0000000000 - Costo: -14311841670884020090560.0000000000
It: 345970 - Temp: -345880.0000000000 - Costo: -143120071844420073111552.0000000000
It: 345971 - Temp: -345881.0000000000 - Costo: -14312172699391709151232.0000000000
It: 345972 - Temp: -345882.0000000000 - Costo: -14312338215798950592512.0000000000
```

Método de la secante.

```
+ Código + Texto
It: 36384 - Temp: 0.8445766126 - Costo: 2.9274825671
It: 36385 - Temp: 0.8445763081 - Costo: 2.9274825701
It: 36386 - Temp: 0.8445760036 - Costo: 2.9274825732
It: 36387 - Temp: 0.8445756992 - Costo: 2.9274825762
It: 36388 - Temp: 0.8445753948 - Costo: 2.9274825793
It: 36389 - Temp: 0.8445750904 - Costo: 2.9274825824
It: 36390 - Temp: 0.8445747861 - Costo: 2.9274825854
It: 36391 - Temp: 0.8445744819 - Costo: 2.9274825885
It: 36392 - Temp: 0.8445741777 - Costo: 2.9274825915
It: 36393 - Temp: 0.8445738735 - Costo: 2.9274825946
It: 36394 - Temp: 0.8445735694 - Costo: 2.9274825976
It: 36395 - Temp: 0.8445732653 - Costo: 2.9274826007
It: 36396 - Temp: 0.8445729613 - Costo: 2.9274826037
It: 36397 - Temp: 0.8445726573 - Costo: 2.9274826067
It: 36398 - Temp: 0.8445723534 - Costo: 2.9274826098
It: 36399 - Temp: 0.8445720495 - Costo: 2.9274826128
It: 36400 - Temp: 0.8445717457 - Costo: 2.9274826159
It: 36401 - Temp: 0.8445714419 - Costo: 2.9274826189
It: 36402 - Temp: 0.8445711381 - Costo: 2.9274826220
It: 36403 - Temp: 0.8445708344 - Costo: 2.9274826250
It: 36404 - Temp: 0.8445705308 - Costo: 2.9274826280
It: 36405 - Temp: 0.8445702271 - Costo: 2.9274826311
It: 36406 - Temp: 0.8445699236 - Costo: 2.9274826341
It: 36407 - Temp: 0.8445696201 - Costo: 2.9274826372
It: 36408 - Temp: 0.8445693166 - Costo: 2.9274826402
It: 36409 - Temp: 0.8445690131 - Costo: 2.9274826432
It: 36410 - Temp: 0.8445687098 - Costo: 2.9274826463
It: 36411 - Temp: 0.8445684064 - Costo: 2.9274826493
-----
It: 36411 - Temp: 0.8445684064 - Costo: 2.9274826493
```

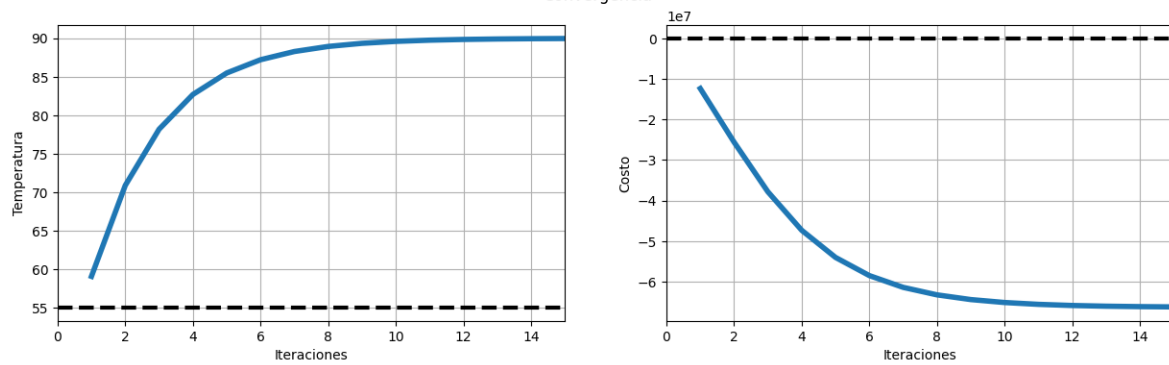
Convergencia



Método de la sección dorada.

It:	38	-	Temp:	89.9999994279	-	Costo:	-66289894.5791484714
It:	39	-	Temp:	89.9999996464	-	Costo:	-66289895.2213858664
It:	40	-	Temp:	89.9999997115	-	Costo:	-66289895.6183103845
It:	41	-	Temp:	89.9999998650	-	Costo:	-66289895.8636232242
It:	42	-	Temp:	89.9999999165	-	Costo:	-66289896.0152349100
It:	43	-	Temp:	89.9999999484	-	Costo:	-66289896.1089360565
It:	44	-	Temp:	89.9999999681	-	Costo:	-66289896.1668465808
It:	45	-	Temp:	89.9999999803	-	Costo:	-66289896.2026372552
It:	46	-	Temp:	89.9999999878	-	Costo:	-66289896.2247571126
It:	47	-	Temp:	89.9999999925	-	Costo:	-66289896.2384279370
It:	48	-	Temp:	89.9999999953	-	Costo:	-66289896.2468769625
It:	49	-	Temp:	89.9999999971	-	Costo:	-66289896.2520987540
It:	50	-	Temp:	89.9999999982	-	Costo:	-66289896.2553259879
It:	51	-	Temp:	89.9999999989	-	Costo:	-66289896.2573205233
It:	52	-	Temp:	89.9999999993	-	Costo:	-66289896.2585531995
It:	53	-	Temp:	89.9999999996	-	Costo:	-66289896.2593150586
It:	54	-	Temp:	89.9999999997	-	Costo:	-66289896.2597859278
It:	55	-	Temp:	89.9999999998	-	Costo:	-66289896.2600769252
It:	56	-	Temp:	89.9999999999	-	Costo:	-66289896.2602567822
It:	57	-	Temp:	89.9999999999	-	Costo:	-66289896.2603679299
It:	58	-	Temp:	90.0000000000	-	Costo:	-66289896.2604366392
It:	59	-	Temp:	90.0000000000	-	Costo:	-66289896.2604790777
It:	60	-	Temp:	90.0000000000	-	Costo:	-66289896.2605052963
It:	61	-	Temp:	90.0000000000	-	Costo:	-66289896.2605215162
It:	62	-	Temp:	90.0000000000	-	Costo:	-66289896.2605315298
It:	63	-	Temp:	90.0000000000	-	Costo:	-66289896.2605377212
It:	64	-	Temp:	90.0000000000	-	Costo:	-66289896.2605415583
It:	65	-	Temp:	90.0000000000	-	Costo:	-66289896.2605438977
It:	66	-	Temp:	90.0000000000	-	Costo:	-66289896.2605454102

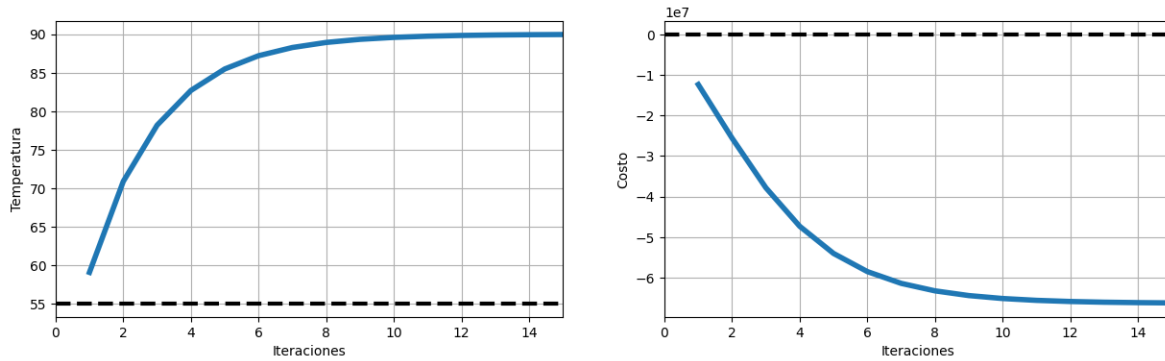
Convergencia



Método de Fibonacci.

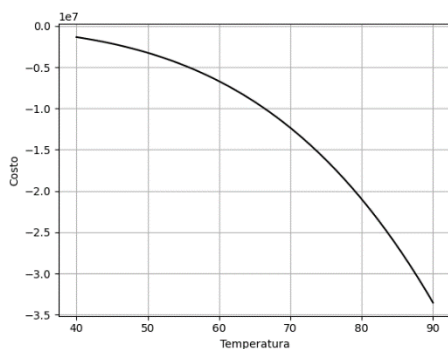
Temperatura		
It: 01	- Temp: 59.0983005625	- Costo: -12383451.9241154157
It: 02	- Temp: 70.9016994375	- Costo: -25597116.6496775001
It: 03	- Temp: 78.1966011250	- Costo: -37830687.4340184182
It: 04	- Temp: 82.7050983125	- Costo: -47311636.5287258774
It: 05	- Temp: 85.4015028125	- Costo: -53999024.9451837397
It: 06	- Temp: 87.2135955000	- Costo: -58471600.1773572258
It: 07	- Temp: 88.2779073125	- Costo: -61371373.7166927606
It: 08	- Temp: 88.9356881875	- Costo: -63216573.3085189983
It: 09	- Temp: 89.3422191250	- Costo: -64377577.2144985497
It: 10	- Temp: 89.5934690625	- Costo: -65183066.2380217314
It: 11	- Temp: 89.7487500625	- Costo: -65554497.8206858411
It: 12	- Temp: 89.8447189999	- Costo: -65834669.0385244265
It: 13	- Temp: 89.9040310626	- Costo: -66008272.7367537171
It: 14	- Temp: 89.9406879373	- Costo: -66115737.2399967611
It: 15	- Temp: 89.9633431253	- Costo: -66182219.5034932643
It: 16	- Temp: 89.9773448120	- Costo: -66223332.8903712854
It: 17	- Temp: 89.9859983134	- Costo: -66248751.8858799338
It: 18	- Temp: 89.9913464986	- Costo: -66264465.4269338921
It: 19	- Temp: 89.9946518148	- Costo: -66274178.1972920299
It: 20	- Temp: 89.9966046838	- Costo: -66280181.7627740577
It: 21	- Temp: 89.9979571310	- Costo: -66283892.0352133140
It: 22	- Temp: 89.9987375528	- Costo: -66286185.7360758185
It: 23	- Temp: 89.9992195781	- Costo: -66287602.4634017497
It: 24	- Temp: 89.9995179747	- Costo: -66288479.4964587167
It: 25	- Temp: 89.9997016834	- Costo: -66289019.2134342194
It: 26	- Temp: 89.9998163713	- Costo: -66289356.5382173285
It: 27	- Temp: 89.9998852321	- Costo: -66289558.9337050542
It: 28	- Temp: 89.9999311392	- Costo: -66289693.8642876670
It: 29	- Temp: 89.9999540928	- Costo: -66289761.3296561986
It: 29	- Temp: 89.9999770464	- Costo: -66289828.7950761989

Convergencia



$$5 - (T^{**4} + 2 * T^{**3} + 13) * 1/2$$

Método del gradiente y bisección.

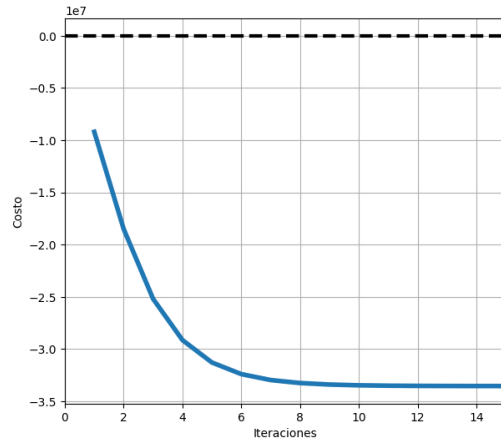
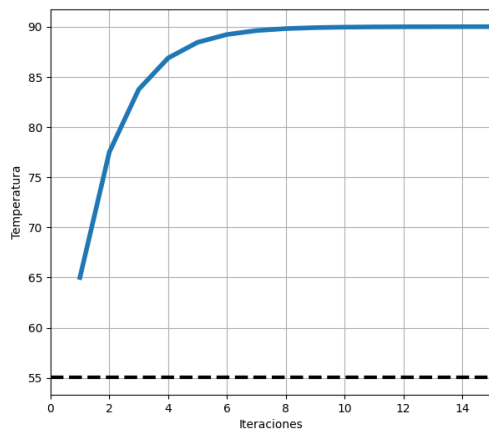



```

-T**4/2 - T**3 - 3/2
-2*T**3 - 3*T**2
It: 01 - Temp: 65.0000000000 - Costo -9199939.0000000000
It: 02 - Temp: 77.5000000000 - Costo -18503005.4062500000
It: 03 - Temp: 83.7500000000 - Costo -25185965.6113281250
It: 04 - Temp: 86.8750000000 - Costo -29136279.4083251953
It: 05 - Temp: 88.4375000000 - Costo -31277205.6215896606
It: 06 - Temp: 89.2187500000 - Costo -32390864.8416295052
It: 07 - Temp: 89.6093750000 - Costo -32958716.3701734841
It: 08 - Temp: 89.8046875000 - Costo -33245425.7045406699
It: 09 - Temp: 89.9023437500 - Costo -33389479.7998086296
It: 10 - Temp: 89.9511718750 - Costo -33461682.1286890544
It: 11 - Temp: 89.9755859375 - Costo -33497827.1773837544
It: 12 - Temp: 89.9877929688 - Costo -33515910.6784642041
It: 13 - Temp: 89.9938964844 - Costo -33524955.1740517356
It: 14 - Temp: 89.9969482422 - Costo -33529478.1082153432
It: 15 - Temp: 89.9984741211 - Costo -33531739.7466831103
-----
It: 15 - Temp: 89.9992370605 - Costo -33532870.6091501750

```

Analisis de convergencia



Método de Newton – Raphson.

```

+ Código + Texto
It: 259830 - Temp: -0.5800246896 - Costo: -1.3614551973
It: 259831 - Temp: -2.7794247145 - Costo: -9.8677543393
It: 259832 - Temp: 2.2802371386 - Costo: -3.1612528485
It: 259833 - Temp: -1.8906159293 - Costo: -1.1303982457
It: 259834 - Temp: -1.4858118280 - Costo: -0.6566972425
It: 259835 - Temp: -11.9687042782 - Costo: -8547.2488839138
It: 259836 - Temp: -9.1189386148 - Costo: -2700.5947415744
It: 259837 - Temp: -6.9876298886 - Costo: -852.3517059115
It: 259838 - Temp: -5.3970910839 - Costo: -268.5277757607
It: 259839 - Temp: -4.2143241693 - Costo: -84.3695577674
It: 259840 - Temp: -3.3392634573 - Costo: -26.4337773980
It: 259841 - Temp: -2.6948213327 - Costo: -8.2988127188
It: 259842 - Temp: -2.2166064951 - Costo: -2.6795253633
It: 259843 - Temp: -1.8360931143 - Costo: -0.9927158317
It: 259844 - Temp: -1.3980206724 - Costo: -0.6775824643
It: 259845 - Temp: -3.0978011288 - Costo: -17.8175246858
It: 259846 - Temp: -2.5167858106 - Costo: -5.6192614295
It: 259847 - Temp: 2.0805440359 - Costo: -1.8626887869
It: 259848 - Temp: -1.7099304701 - Costo: -0.7748840314
It: 259849 - Temp: -1.0787197956 - Costo: -0.9217872159
It: 259850 - Temp: -2.0189026685 - Costo: -1.5777748653
It: 259851 - Temp: -1.6459116541 - Costo: -0.7105925906
It: 259852 - Temp: -0.7470607541 - Costo: -1.2388032415
It: 259853 - Temp: -2.2210711920 - Costo: -2.7111267457
It: 259854 - Temp: -1.839906957 - Costo: -1.0016182599
It: 259855 - Temp: -1.4049058550 - Costo: -0.6749175509
It: 259856 - Temp: -3.2028347142 - Costo: -21.2596632437
It: 259857 - Temp: 2.5943007310 - Costo: -6.688450657
It: 259858 - Temp: -2.1402359231 - Costo: -2.1874074424
It: 259859 - Temp: -1.7672981676 - Costo: -0.8577571338

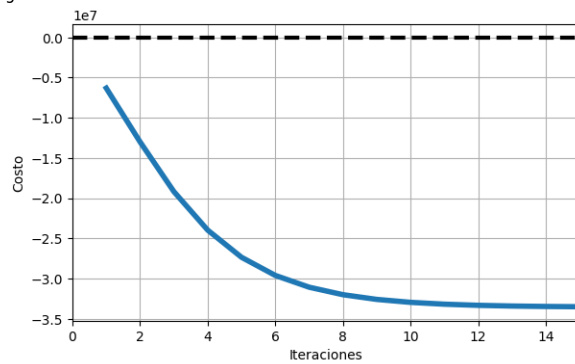
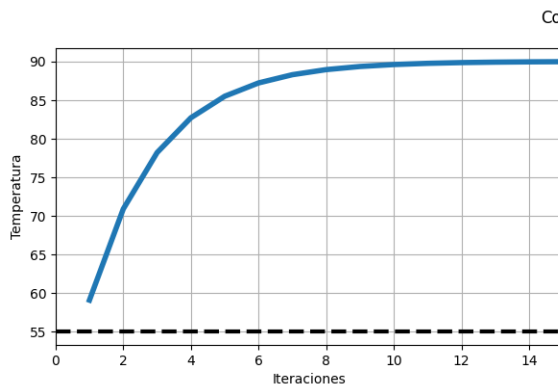
```

Método de la secante.

```
+ Código + Texto
It: 186623 - Temp: 0.4118513911 - Costo: -1.5842445977
It: 186624 - Temp: 0.4118471287 - Costo: -1.5842418526
It: 186625 - Temp: 0.4118428963 - Costo: -1.5842391077
It: 186626 - Temp: 0.4118386639 - Costo: -1.5842363628
It: 186627 - Temp: 0.4118344315 - Costo: -1.5842336179
It: 186628 - Temp: 0.4118301992 - Costo: -1.5842308732
It: 186629 - Temp: 0.4118259668 - Costo: -1.5842281285
It: 186630 - Temp: 0.4118217344 - Costo: -1.5842253838
It: 186631 - Temp: 0.4118175021 - Costo: -1.5842226393
It: 186632 - Temp: 0.4118132697 - Costo: -1.5842198948
It: 186633 - Temp: 0.4118090374 - Costo: -1.5842171504
It: 186634 - Temp: 0.4118048050 - Costo: -1.5842144060
It: 186635 - Temp: 0.4118005727 - Costo: -1.5842116617
It: 186636 - Temp: 0.4117963404 - Costo: -1.5842089175
It: 186637 - Temp: 0.4117921081 - Costo: -1.5842061733
It: 186638 - Temp: 0.4117878758 - Costo: -1.5842034292
It: 186639 - Temp: 0.4117836435 - Costo: -1.5842006852
It: 186640 - Temp: 0.4117794112 - Costo: -1.5841979412
It: 186641 - Temp: 0.4117751789 - Costo: -1.5841951973
It: 186642 - Temp: 0.4117709466 - Costo: -1.5841924535
It: 186643 - Temp: 0.4117667143 - Costo: -1.5841897097
It: 186644 - Temp: 0.4117624821 - Costo: -1.5841869661
It: 186645 - Temp: 0.4117582498 - Costo: -1.5841842224
It: 186646 - Temp: 0.4117540176 - Costo: -1.5841814789
It: 186647 - Temp: 0.4117497853 - Costo: -1.5841787354
It: 186648 - Temp: 0.4117455531 - Costo: -1.5841759920
It: 186649 - Temp: 0.4117413209 - Costo: -1.5841732486
It: 186650 - Temp: 0.4117370886 - Costo: -1.5841705053
It: 186651 - Temp: 0.4117328564 - Costo: -1.5841677621
It: 186652 - Temp: 0.4117286242 - Costo: -1.5841650189
It: 186653 - Temp: 0.4117243920 - Costo: -1.5841622759
It: 186654 - Temp: 0.4117201598 - Costo: -1.5841595328
It: 186655 - Temp: 0.4117159276 - Costo: -1.5841567899
```

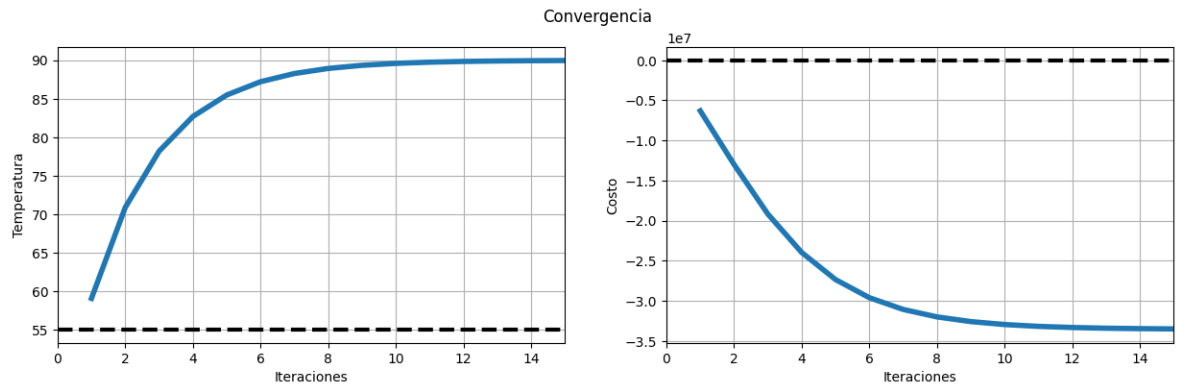
Método de la sección dorada.

```
+ Código + Texto
It: 38 - Temp: 89.9999994279 - Costo: -33534000.6520392373
It: 39 - Temp: 89.9999996464 - Costo: -33534000.9759314395
It: 40 - Temp: 89.9999997815 - Costo: -33534001.1761078164
It: 41 - Temp: 89.9999998650 - Costo: -33534001.2998236194
It: 42 - Temp: 89.9999999165 - Costo: -33534001.3762841970
It: 43 - Temp: 89.9999999484 - Costo: -33534001.4235394225
It: 44 - Temp: 89.9999999681 - Costo: -33534001.4527447745
It: 45 - Temp: 89.9999999803 - Costo: -33534001.4707946740
It: 46 - Temp: 89.9999999878 - Costo: -33534001.4819501229
It: 47 - Temp: 89.9999999925 - Costo: -33534001.4888445735
It: 48 - Temp: 89.9999999953 - Costo: -33534001.4931055680
It: 49 - Temp: 89.9999999971 - Costo: -33534001.4957390204
It: 50 - Temp: 89.9999999982 - Costo: -33534001.4973665662
It: 51 - Temp: 89.9999999989 - Costo: -33534001.4983724542
It: 52 - Temp: 89.9999999993 - Costo: -33534001.4989941195
It: 53 - Temp: 89.9999999996 - Costo: -33534001.4993783347
It: 54 - Temp: 89.9999999997 - Costo: -33534001.4996157996
It: 55 - Temp: 89.9999999998 - Costo: -33534001.4997625574
It: 56 - Temp: 89.9999999999 - Costo: -33534001.4998532608
It: 57 - Temp: 89.9999999999 - Costo: -33534001.4999093153
It: 58 - Temp: 90.0000000000 - Costo: -33534001.4999439679
It: 59 - Temp: 90.0000000000 - Costo: -33534001.4999653697
It: 60 - Temp: 90.0000000000 - Costo: -33534001.4999785982
It: 61 - Temp: 90.0000000000 - Costo: -33534001.4999867715
It: 62 - Temp: 90.0000000000 - Costo: -33534001.4999918267
It: 63 - Temp: 90.0000000000 - Costo: -33534001.4999949448
It: 64 - Temp: 90.0000000000 - Costo: -33534001.4999968819
It: 65 - Temp: 90.0000000000 - Costo: -33534001.4999980591
-----
It: 65 - Temp: 90.0000000000 - Costo: -33534001.4999980591
```



Método de Fibonacci.

It: 01	- Temp: 59.0983005625	- Costo: -6305568.0293742493
It: 02	- Temp: 70.9016994375	- Costo: -12992048.7513341252
It: 03	- Temp: 78.1966011250	- Costo: -19172980.4925862700
It: 04	- Temp: 82.7050983125	- Costo: -23959427.0871040188
It: 05	- Temp: 85.4915028125	- Costo: -27334098.7258196846
It: 06	- Temp: 87.2135955000	- Costo: -29590591.2444076203
It: 07	- Temp: 88.2779073125	- Costo: -31053288.4030555338
It: 08	- Temp: 88.9356881875	- Costo: -31983986.4307878688
It: 09	- Temp: 89.3422191250	- Costo: -32569552.3304167390
It: 10	- Temp: 89.5934690625	- Costo: -32935449.1643292755
It: 11	- Temp: 89.7487500625	- Costo: -33163121.8678193167
It: 12	- Temp: 89.8447189999	- Costo: -33304420.2326638699
It: 13	- Temp: 89.9040310625	- Costo: -33391972.8748002425
It: 14	- Temp: 89.9406879373	- Costo: -33446169.6270048246
It: 15	- Temp: 89.9633431253	- Costo: -33479698.0211249590
It: 16	- Temp: 89.9773448120	- Costo: -33500432.3234045208
It: 17	- Temp: 89.9859983134	- Costo: -33513251.6162928008
It: 18	- Temp: 89.9913464986	- Costo: -33521176.2546220757
It: 19	- Temp: 89.9946518148	- Costo: -33526074.5877973884
It: 20	- Temp: 89.9966046838	- Costo: -33529102.2982381604
It: 21	- Temp: 89.9979571310	- Costo: -33530973.4577694274
It: 22	- Temp: 89.9987375528	- Costo: -33532130.2137406096
It: 23	- Temp: 89.9992195781	- Costo: -33532844.6956151500
It: 24	- Temp: 89.9995179747	- Costo: -33533286.9996400774
It: 25	- Temp: 89.9997016034	- Costo: -33533559.1889070980
It: 26	- Temp: 89.9998163713	- Costo: -33533729.3080404103
It: 27	- Temp: 89.9998852321	- Costo: -33533831.3798310794
It: 28	- Temp: 89.9999311392	- Costo: -33533899.4278209805
It: 29	- Temp: 89.9999540928	- Costo: -33533933.4518547542
It: 29	- Temp: 89.9999770464	- Costo: -33533967.4759144224



```
- sin(3*T+45)**2+0.9*sin(9*T)**3-sin(15*T+50)*cos(2*T-30)
- sin(2*pi*T)/T - cos(3*pi*T)/T
```

Para estas funciones Python no me dejó compilar los métodos, me da errores de tamaños y que el objeto U no es iterable, se ingresó de las dos formas, pero no resultó.

```
+ Código + Texto
1 import matplotlib.pyplot as plt
2 import numpy as np
3 import sympy
4 import math
5 from math import sqrt, exp, sin, cos, tan, log
6
7 a = 40
8 b = 90
9 T = np.linspace(a, b, 100)
10 U = sin**2(3*T+45)+0.9*sin**3(9*T)-sin(15*T+50)*cos(2*T-30)
11
12 plt.figure()
13 plt.plot(T, U, 'k')
14 plt.xlabel('Temperatura')
15 plt.ylabel('Costo')
16 plt.grid()
17 plt.show()
18
19 def calcularGradiente():
20     T = sympy.Symbol('T')
21     fU = (sin(3*T+45)**3)+(0.9*sin(9*T)**3)-(sin(15*T+50)*cos(2*T-30))
22
23     d1fU = sympy.diff(fU)
24
25     print(fU)
26     print(d1fU)
27
28     return None
29
```

```
<>:10: SyntaxWarning: 'int' object is not callable; perhaps you missed a comma?
<>:10: SyntaxWarning: 'int' object is not callable; perhaps you missed a comma?
<>:10: SyntaxWarning: 'int' object is not callable; perhaps you missed a comma?
<>:10: SyntaxWarning: 'int' object is not callable; perhaps you missed a comma?
ipython-input-39-a3c24affb2d7:10: SyntaxWarning: 'int' object is not callable; perhaps you missed a comma?
U = sin**2(3*T+45)+0.9*sin**3(9*T)-sin(15*T+50)*cos(2*T-30)
ipython-input-39-a3c24affb2d7:10: SyntaxWarning: 'int' object is not callable; perhaps you missed a comma?
U = sin**2(3*T+45)+0.9*sin**3(9*T)-sin(15*T+50)*cos(2*T-30)
-----
TypeError                                 Traceback (most recent call last)
ipython-input-39-a3c24affb2d7 in <cell line: 10>()
      8 b = 90
      9 T = np.linspace(a, b, 100)
----> 10 U = sin**2(3*T+45)+0.9*sin**3(9*T)-sin(15*T+50)*cos(2*T-30)
      11
      12 plt.figure()

TypeError: 'int' object is not callable
BUSCAR EN STACK OVERFLOW
```

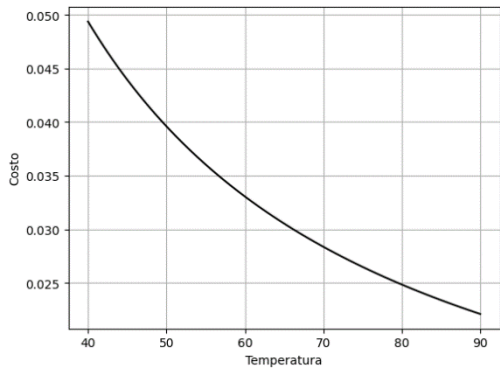
```
+ Código + Texto
2 import numpy as np
3 import sympy
4 import math
5 from math import sqrt, exp, sin, cos, tan, log, pi
6
7 a = 40
8 b = 90
9 T = np.linspace(a, b, 100)
10 U = sin(2*pi*T)/T - cos(3*pi*T)/T
11
12 plt.figure()
13 plt.plot(T, U, 'k')
14 plt.xlabel('Temperatura')
15 plt.ylabel('Costo')
16 plt.grid()
17 plt.show()
18
19 def calcularGradiente():
20     T = sympy.Symbol('T')
21     fU = sin(2*pi*T)/(T) - cos(3*pi*T)/T
22
23     d1fU = sympy.diff(fU)
24
25     print(fU)
26     print(d1fU)
27
28     return None
29
30 calcularGradiente()
31
```

```
-----
TypeError                                 Traceback (most recent call last)
ipython-input-43-fa0beb4bbe17 in <cell line: 10>()
      8 b = 90
      9 T = np.linspace(a, b, 100)
----> 10 U = sin(2*pi*T)/T - cos(3*pi*T)/T
      11
      12 plt.figure()

TypeError: only size-1 arrays can be converted to Python scalars
BUSCAR EN STACK OVERFLOW
```

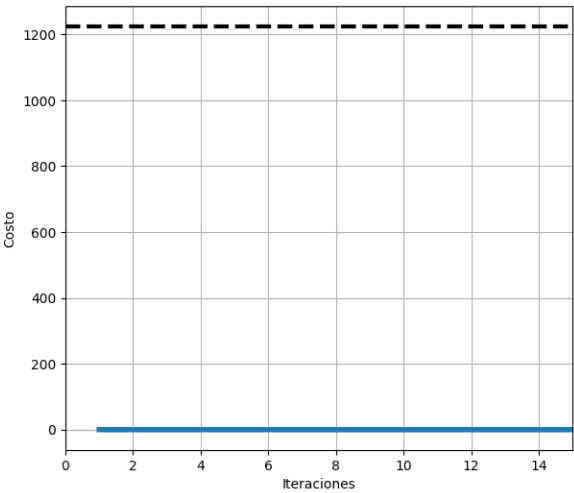
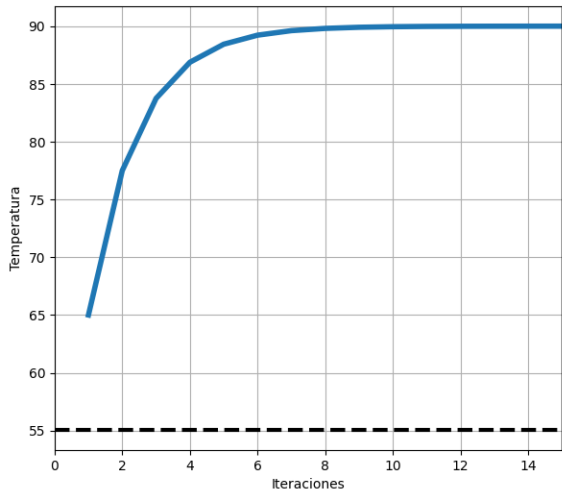
$$\frac{(2 \cdot T - 1)}{T^2}$$

Método del gradiente y Método de la bisección.



$\frac{(2 \cdot T - 1)}{T^2}$		
$2/T^2 - 2 \cdot (2 \cdot T - 1)/T^3$		
It: 01 - Temp:	65.0000000000	- Costo 0.0153846154
It: 02 - Temp:	77.5000000000	- Costo -0.0000000000
It: 03 - Temp:	83.7500000000	- Costo -0.0034972325
It: 04 - Temp:	86.8750000000	- Costo -0.0037343695
It: 05 - Temp:	88.4375000000	- Costo 0.0106092287
It: 06 - Temp:	89.2187500000	- Costo 0.0057094338
It: 07 - Temp:	89.6093750000	- Costo 0.0024923210
It: 08 - Temp:	89.8046875000	- Costo -0.0075144130
It: 09 - Temp:	89.9023437500	- Costo -0.0131400271
It: 10 - Temp:	89.9511718750	- Costo -0.0133180277
It: 11 - Temp:	89.9755859375	- Costo -0.0125194121
It: 12 - Temp:	89.9877929688	- Costo -0.0118906469
It: 13 - Temp:	89.9938964844	- Costo -0.0115195152
It: 14 - Temp:	89.9969482422	- Costo -0.0113199392
It: 15 - Temp:	89.9984741211	- Costo -0.0112166771
.....		
It: 15 - Temp:	89.9992370605	- Costo -0.0111641815

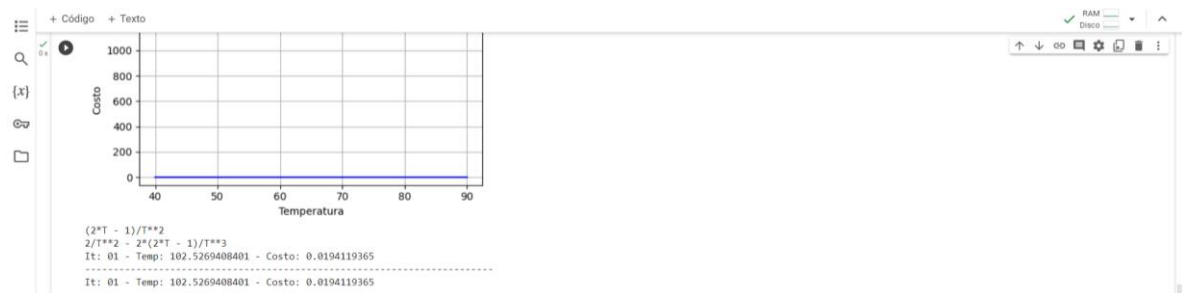
Analisis de convergencia



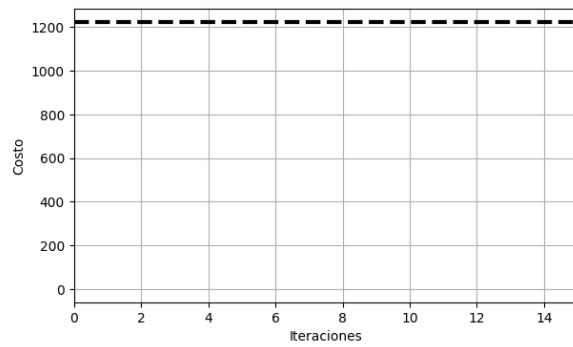
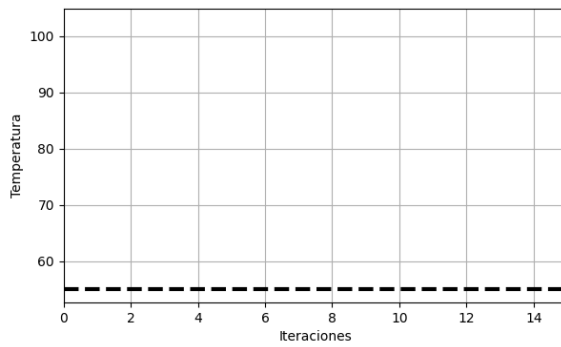
Método de Newton – Raphson.

It:	58	-	Temp:	32.0000000000	-	Costo:	0.0615234375
It:	59	-	Temp:	31.0000000000	-	Costo:	0.0634755463
It:	60	-	Temp:	30.0000000000	-	Costo:	0.0655555556
It:	61	-	Temp:	29.0000000000	-	Costo:	0.067764566
It:	62	-	Temp:	28.0000000000	-	Costo:	0.0701530612
It:	63	-	Temp:	27.0000000000	-	Costo:	0.0727023320
It:	64	-	Temp:	26.0000000000	-	Costo:	0.0754437070
It:	65	-	Temp:	25.0000000000	-	Costo:	0.0784000000
It:	66	-	Temp:	24.0000000000	-	Costo:	0.0815972222
It:	67	-	Temp:	23.0000000000	-	Costo:	0.0850661626
It:	68	-	Temp:	22.0000000000	-	Costo:	0.0888429752
It:	69	-	Temp:	21.0000000000	-	Costo:	0.0929705215
It:	70	-	Temp:	20.0000000000	-	Costo:	0.0975000000
It:	71	-	Temp:	19.0000000000	-	Costo:	0.1024930748
It:	72	-	Temp:	18.0000000000	-	Costo:	0.1080246914
It:	73	-	Temp:	17.0000000000	-	Costo:	0.1141868512
It:	74	-	Temp:	16.0000000000	-	Costo:	0.1210937500
It:	75	-	Temp:	15.0000000000	-	Costo:	0.1288888889
It:	76	-	Temp:	14.0000000000	-	Costo:	0.1377551020
It:	77	-	Temp:	13.0000000000	-	Costo:	0.1479289941
It:	78	-	Temp:	12.0000000000	-	Costo:	0.1597222222
It:	79	-	Temp:	11.0000000000	-	Costo:	0.1735537190
It:	80	-	Temp:	10.0000000000	-	Costo:	0.1900000000
It:	81	-	Temp:	9.0000000000	-	Costo:	0.2098765432
It:	82	-	Temp:	8.0000000000	-	Costo:	0.2343750000
It:	83	-	Temp:	7.0000000000	-	Costo:	0.2653061224
It:	84	-	Temp:	6.0000000000	-	Costo:	0.3055555556
It:	85	-	Temp:	5.0000000000	-	Costo:	0.3600000000
It:	86	-	Temp:	4.0000000000	-	Costo:	0.4375000000
It:	87	-	Temp:	3.0000000000	-	Costo:	0.5555555556
It:	88	-	Temp:	2.0000000000	-	Costo:	0.7500000000
It:	89	-	Temp:	1.0000000000	-	Costo:	1.0000000000

Método de la secante.



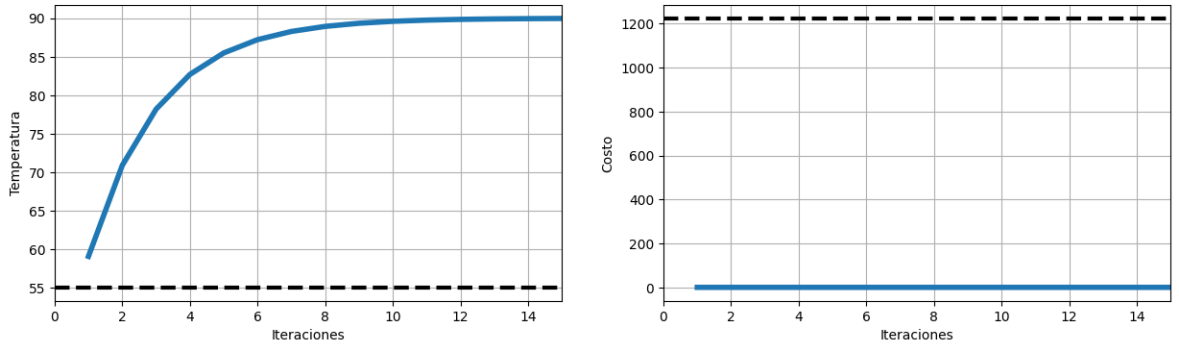
Convergencia



Método de la sección dorada.



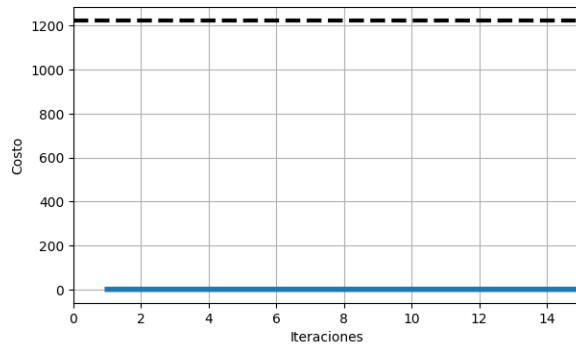
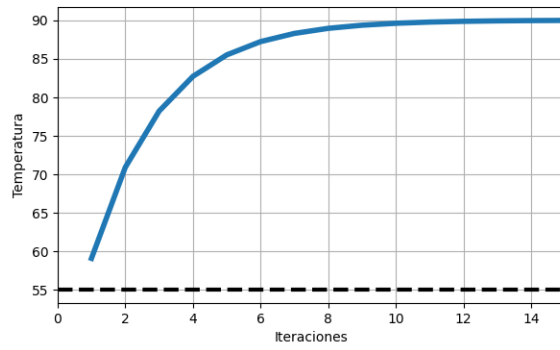
Convergencia



Método de Fibonacci.



Convergencia



Con las funciones $-T^{**4} \log(\text{abs}(T) - 1)$ y $\exp^{**T} \log(\text{abs}(T) + T/\text{abs}(T))$, python no me permitió hacer las comprobaciones con los métodos.

```
+ Código + Texto
RAM
Disco

1 import matplotlib.pyplot as plt
2 import numpy as np
3 import sympy
4 import math
5 from math import sqrt, exp, sin, cos, tan, log, pi
6
7 a = 40
8 b = 90
9 T = np.linspace(a, b, 100)
10 U = exp**T*log(abs(T)+T/abs(T))
11
12 plt.figure()
13 plt.plot(T, U, 'k')
14 plt.xlabel('Temperatura')
15 plt.ylabel('Costo')
16 plt.grid()
17 plt.show()
18
19 def calcularGradiente():
20     T = sympy.Symbol('T')
21     fu = exp**T*log(abs(T)+T/abs(T))
22
23     difu = sympy.diff(fu)
24
25     print(fu)
26     print(difu)
27
28     return None
29
30 calcularGradiente()
31
32 def U(T):
33     return exp**T*log(abs(T)+T/abs(T))
34
35 U(40,90)
36
```

```
+ Código + Texto
RAM
Disco

1 import matplotlib.pyplot as plt
2 import numpy as np
3 import sympy
4 import math
5 from math import sqrt, exp, sin, cos, tan, log, pi
6
7 a = 40
8 b = 90
9 T = np.linspace(a, b, 100)
10 U = -T**4*log(abs(T)-1)
11
12 plt.figure()
13 plt.plot(T, U, 'k')
14 plt.xlabel('Temperatura')
15 plt.ylabel('Costo')
16 plt.grid()
17 plt.show()
18
19 def calcularGradiente():
20     T = sympy.Symbol('T')
21     fu = -T**4*log(abs(T)-1)
22
23     difu = sympy.diff(fu)
24
25     print(fu)
26     print(difu)
27
28     return None
29
30 calcularGradiente()
31
32 def U(T):
33     return -T**4*log(abs(T)-1)
34
35 U(40,90)
36
```

Mismo error en ambas funciones.

```
-----
TypeError                                Traceback (most recent call last)
<ipython-input-56-a53ac08fcaab> in <cell line: 10>()
      8 b = 90
      9 T = np.linspace(a, b, 100)
----> 10 U = -T**4*log(abs(T)-1)
      11
      12 plt.figure()

TypeError: only size-1 arrays can be converted to Python scalars

BUSCAR EN STACK OVERFLOW
```

A continuación, daremos una conclusión sobre el código y los test que hicimos con los mismos, sobre el código es muy eficiente y elegante, tanto en su programación, así como se muestra en la terminal, es muy amigable, ya que muestra las soluciones numéricas y graficas. Es una gran herramienta para seguir probando funciones, analizar su convergencia y resultados. Tiene algunas sutilezas, como cuando las funciones no son bien portadas, el programa tiene errores, así como no hay una alternativa cuando en algún método se requiere una segunda derivada y esta es cero, pero en términos generales están muy bien los programas.

Sobre los test a las funciones, pudimos notar que para algunas funciones, los resultados de convergencia son mas rápidos en un método de convergencia que en otro, en otros, no tenía un resultado concreto y en las funciones trigonométricas complejas, no había manera de ver su convergencia, pero en las demás funciones nos dimos cuenta de ello, mientras unos métodos para encontrar el resultado, necesitaban miles de iteraciones, en otro método eran menos de cien iteraciones, también nos dimos cuenta que el resultado no solo depende de la complejidad de la función, sino también del método que se usa, ya que unos son más óptimos que otros, un ejemplo es con el método Newton – Raphson, converge más rápido que otros métodos. Esas son mis conclusiones.