



UNIVERSIDAD DE BUENOS AIRES
FACULTAD DE INGENIERÍA
<75.12> ANÁLISIS NUMÉRICO

DATOS DEL TRABAJO PRÁCTICO

1	2	0	1	8				
	AÑO				Ecuaciones no lineales en sistemas			
	1°				mecánicos			
TP NRO	CUAT				TEMA			

INTEGRANTES DEL GRUPO

8	B	r	a	c	c	e	l	a	r	g	h	e		A	i	l	i	n		Y		9	9	3	6	6
	APELLIDO Y NOMBRE																				PADRÓN					
	J	a	m	i	l	i	s		N	e	t	a	n	e	l							9	9	0	9	3
GRUPO	APELLIDO Y NOMBRE																				PADRÓN					

DATOS DE LA ENTREGA

P	T	0	8	.	t	x	t		-					0	2	/	0	5	/	1	8	2	9	/	0	4	/	1	8
ARCHIVO								NRO CONTROL					FECHA VENC					FECHA ENTR											

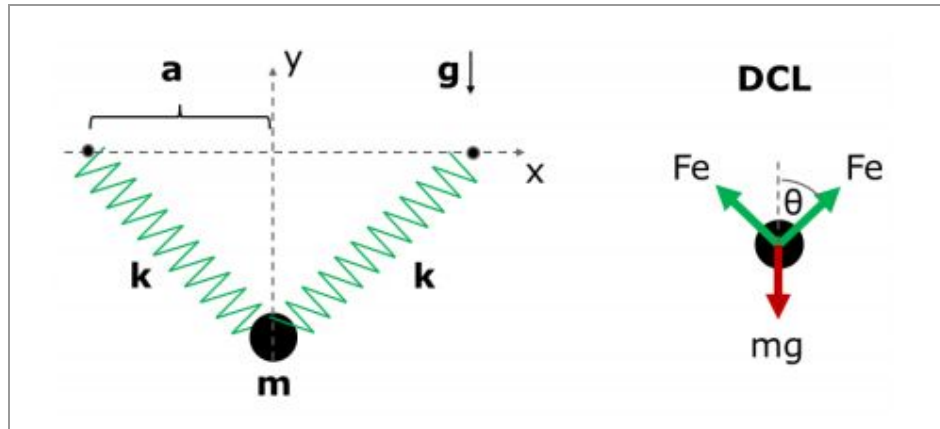
CORRECCIONES

FECHA	NOTA	OBSERVACIONES
DOCENTE	FIRMA	

Objetivo

Resolver ecuaciones no lineales de un problema mecánico con la experimentación de métodos numéricos.

Sistema mecánico:



$$F_{resultante}(y) = 2 \cdot F_{elastica} \cdot \cos\theta - mg$$

donde $F_{elastica} = -k \cdot \text{estiramiento}$ y $\text{estiramiento} = \sqrt{y^2 + a^2} - L_0$

Los puntos de equilibrio del sistema serán aquellos donde la resultante de fuerzas conservativas sea nula: $F_{resultante}(y) = 0$.

Utilizando los métodos numéricos aprendidos en clase, se procederá a resolver las ecuaciones lineales y no lineales del sistema (dependiendo del valor de la masa m), y así entender si existen puntos de equilibrio en el sistema, cuáles son y si éstos son estables o inestables.

Desarrollo del trabajo práctico

1. Punto de equilibrio positivo, sin tener en cuenta el efecto de la gravedad.

Sin efecto de gravedad la constante g es nula y nos queda la ecuación:

$$f(y) = -2ky \left(1 - \frac{Lo}{\sqrt{y^2 + a^2}}\right) = 0$$

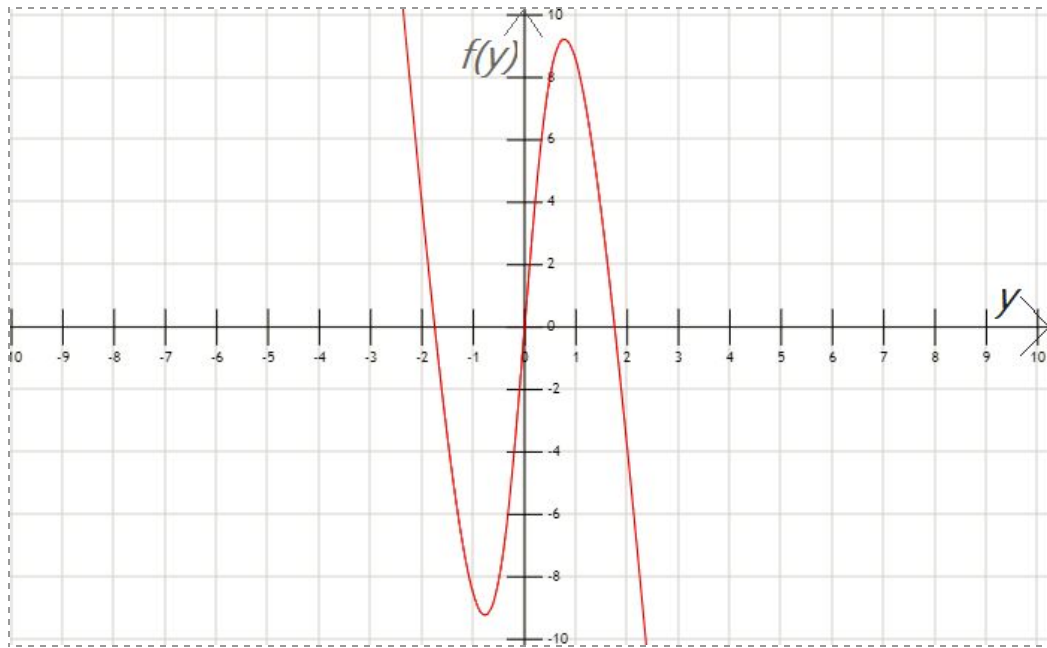


gráfico $f(y)$

Constante	Lo	k	a
Valor	2,02	10	1

Cuyas raíces son: $y_1 = 0$, $y_2 = 1,7532$, $y_3 = -1,7532$

Entonces el punto de equilibrio positivo es $y = 1,7532$ e intentaremos hallarlo mediante los siguientes métodos numéricos:

a. Regula-Falsi

Tablas de datos que resultan del algoritmo utilizado para este método:

k	a_k	b_k	$f(a_k)$	$f(b_k)$
0	1	2	8.5671139	-3.865141
1	1.68910375907	2	0.982278	-3.865141
2	1.75210359838	2	0.045315	-3.865141
3	1.75497631505	2	0.001970	-3.865141
4	1.75510117367	2	8.546774e-05	-3.865141
5	1.75510658887	2	3.706445e-06	-3.865141
6	1.75510682370	2	1.607351e-07	-3.865141
7	1.75510683389	2	6.970505e-09	-3.865141
8	1.75510683433	2	3.022840e-10	-3.865141
9	1.75510683435	2	1.310991e-11	-3.865141

k	r_{k+1}	Δr_{k+1}	$\Delta r/r$	λ	p
0	1.689103759076				
1	1.752103598381	6.29998393e-02	3.59566862e-02		
2	1.754976315052	2.87271667e-03	1.63689768e-03	0.047599	1.015530
3	1.755101173678	1.24858626e-04	7.11404150e-05	0.043637	1.000683
4	1.755106588871	5.41519287e-06	3.08539259e-06	0.043382	1.000029
5	1.755106823709	2.34838288e-07	1.33802846e-07	0.043367	1.000001
6	1.755106833893	1.01840875e-08	5.80254567e-09	0.043366	0.999999
7	1.755106834335	4.41647385e-10	2.51635613e-10	0.043369	1.000003
8	1.755106834354	1.91524573e-11	1.09124168e-11	0.043328	1.000045
9	1.755106834355	8.30446822e-13	4.73160269e-13		

b. Punto Fijo

Con la función que se pide en el enunciado: $g(y) = y - f(y)$ obtenemos en las primeras 10 iteraciones, con la semilla $y=1.5$:

k	y_k	$\Delta y = y_k - y_{k+1} $	$\Delta y / y_k$	λ	p
0	1.5000				
1	-2.1148	3.6148	1.7092762342		
2	-7.8886	5.77386	0.7319160898	0.001474	6.437708
3	-125.5833	117.6946	0.9371835782	20.035649	1.009832
4	-2596.8506	2471.26737	0.9516401517	20.993036	1.000042
5	-54493.4643	51896.6136	0.9523456488	20.999998	1.000000
6	-1144322.3519	1089828.88	0.9523792712	20.999999	1.000000
7	-24030728.991	22886406.6	0.9523808723	20.999999	1.000000
8	-504645268.41	480614539.4	0.9523809485	20.999999	1.000000
9	-10597550596	10092905327	0.9523809521	21.000000	1.000000
10	-222548562480	211951011884	0.9523809523		

Con estos datos se puede observar que este algoritmo no converge a ninguna raíz; es por eso que resulta en '*overflow*'.

A partir de esto analizaremos si esta función cumple con las condiciones del teorema de punto fijo:

- No cumple con la primer condición del teorema (el valor de la función ' g ' para todos los puntos del intervalo pertenecen a ese mismo intervalo):

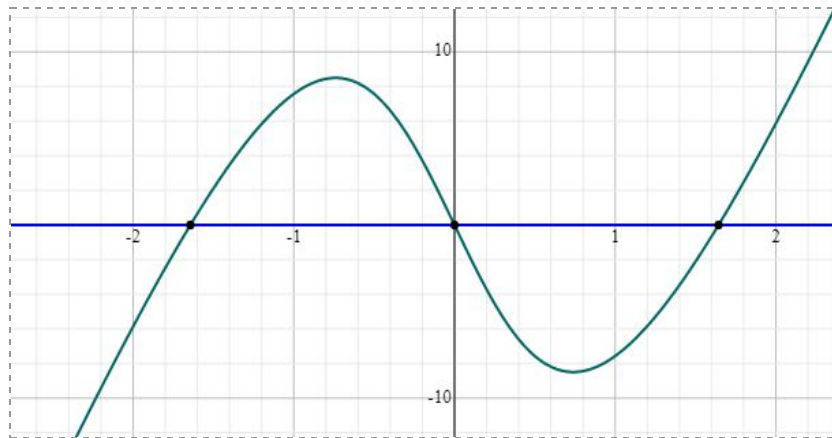


gráfico: $g(y) = y - f(y)$

- Tampoco cumple con la segunda condición del teorema (el módulo de la derivada de la función ' g ' es menor a 1 en todo el intervalo):

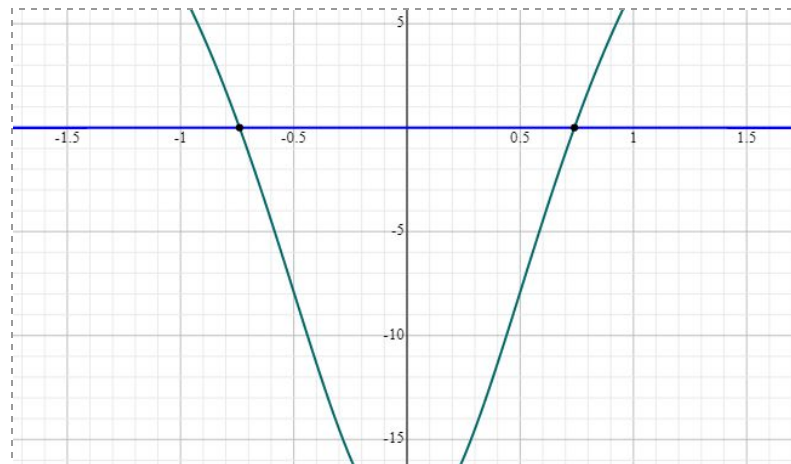


gráfico: $g'(y)$

Para poder aplicar el método de punto fijo a partir de la función original, se puede desarrollar la expresión hasta llegar a la forma: $\frac{yLo}{\sqrt{y^2+a^2}} - \frac{mg}{2k} = y$, y a esta última la llamamos: $g(y) = \frac{yLo}{\sqrt{y^2+a^2}} - \frac{mg}{2k}$.

Como en este caso no tenemos en cuenta los efectos de la gravedad, la función nos queda: $g(y) = \frac{yLo}{\sqrt{y^2+a^2}}$, cuya derivada es: $g'(y) = \frac{Lo\sqrt{y^2+a^2} - \frac{y^2Lo}{\sqrt{y^2+a^2}}}{y^2+a^2}$.

A continuación veremos qué esta nueva función sí cumple las condiciones del teorema de punto fijo en el intervalo $[1;2]$:

- Se puede observar que cumple con la primer condición del teorema:

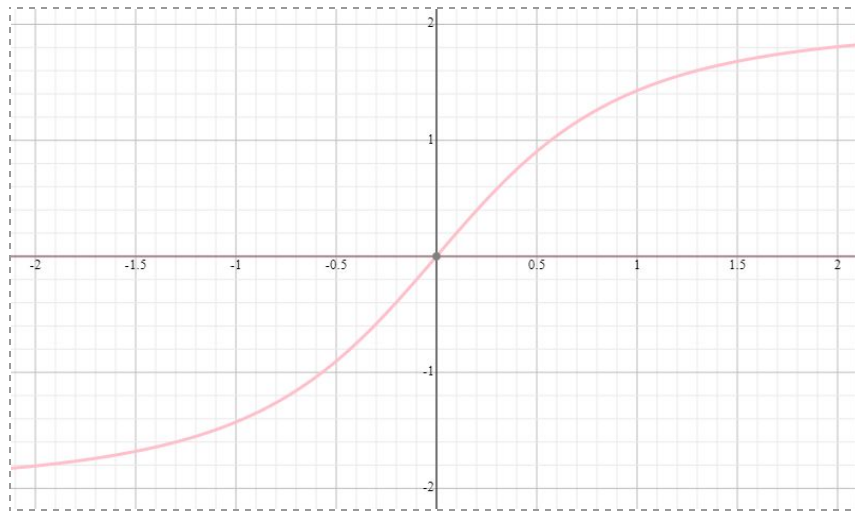


gráfico $g(y)$

- También cumple con la segunda condición del teorema en ese intervalo:

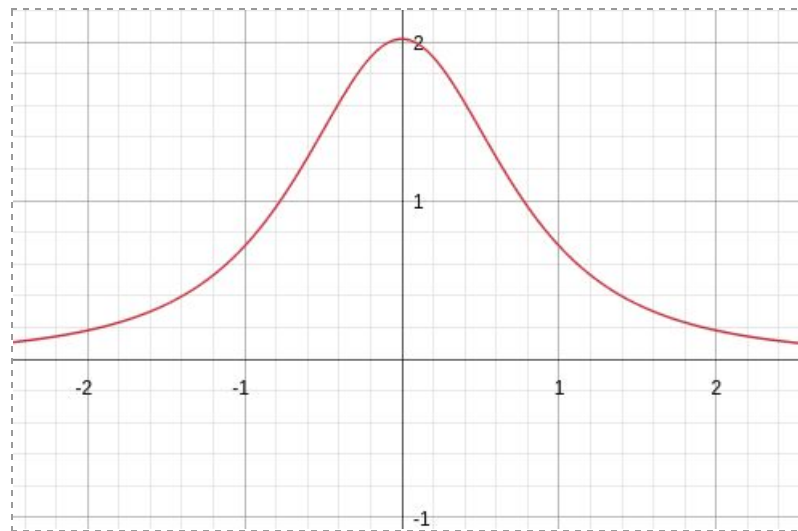


gráfico $g'(y)$

Busco el punto de equilibrio utilizando como semilla $y=2$:

k	y_k	$\Delta y = y_k - y_{k-1} $	$\Delta y/y_k$	λ	p
0	2.000000000000				
1	1.806742925819	0.193257	0.106964		
2	1.767351117204	0.039392	0.022289	0.175759	0.909854
3	1.758084036135	9.2671e-3	5.2711e-3	0.219461	0.909854
4	1.755835069795	2.2490e-3	1.2809e-3	0.236808	0.994765
5	1.755285222112	5.4985e-4	3.1325e-4	0.242586	0.998719
6	1.755150547527	1.3467e-4	7.6731e-5	0.244354	0.999686
7	1.755117547015	3.3001e-5	1.8802e-5	0.244871	0.999923
8	1.755109459731	8.0873e-6	4.6078e-6	0.245018	0.999981
9	1.755107477765	1.9820e-6	1.1292e-6	0.245059	0.999995
10	1.755106992038	4.8573e-7	2.7675e-7	0.245070	0.999999
11	1.755106872999	1.1904e-7	6.7824e-8	0.245073	0.999999
12	1.755106843825	2.9173e-8	1.6622e-8	0.245074	0.999999
13	1.755106836676	7.1496e-9	4.0736e-9	0.245073	0.999999
14	1.755106834923	1.7522e-9	9.983e-10	0.245075	1.000000
15	1.755106834494	4.294e-10	2.447e-10	0.245073	1.000000
16	1.755106834389	1.052e-10	5.996e-11	0.245075	0.999991
17	1.755106834363	2.579e-11	1.470e-11	0.245075	1.000023
18	1.755106834357	6.320e-12	3.601e-12	0.245762	1.000115
19	1.755106834355	1.549e-12	8.824e-13		

c. *Newton-Raphson*

Busco el punto de equilibrio utilizando como semilla el $y=1.5$:

k	y_k	$\Delta y = y_k - y_{k+1} $	$\Delta y / y_k$	λ	p
0	1.5000				
1	1.77584345	0.2758	0.1553		
2	1.7551942120	2.0649e-02	1.1764e-02	0.31202	2.10836
3	1.755106835954	8.7376e-05	4.9783e-5	0.20177	1.99601
4	1.7551068343551053	1.5989e-09	9.1101e-10	0.00066	1.38391
5	1.7551068343551057	4.4408e-16	2.5302e-16		

d. Gráfico representativo del error relativo en todos los métodos empleados

Con la primer función de punto fijo $g(y) = y - f(y)$:

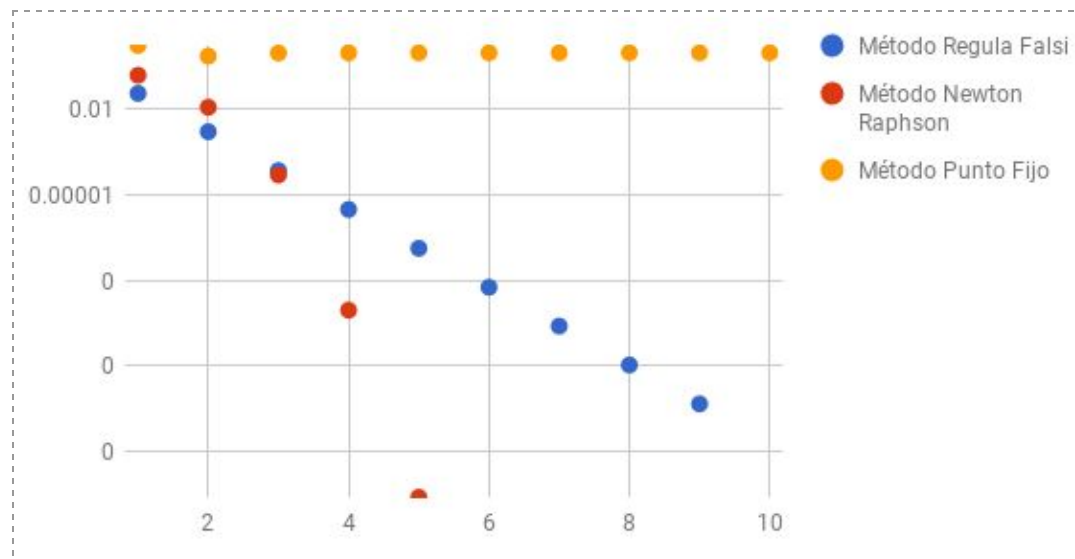


gráfico (1) del error relativo en función de las iteraciones.

Con la nueva función de punto fijo $g(y) = \frac{yL_0}{\sqrt{y^2+a^2}}$:

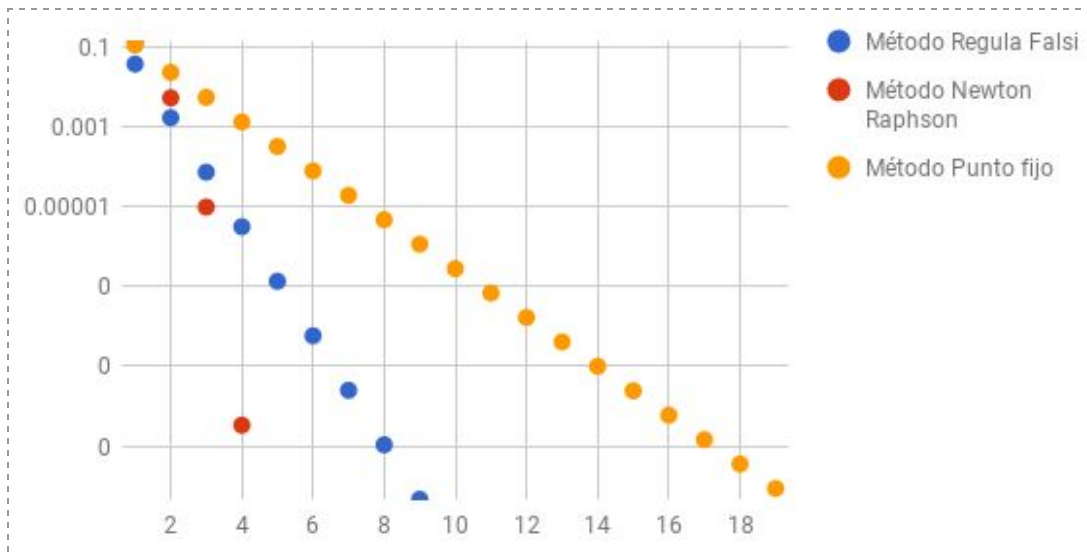


gráfico (2) del error relativo en función de las iteraciones.

2. Todos los puntos de equilibrio del sistema, teniendo en cuenta el efecto de la gravedad.

Teniendo en cuenta el efecto de la gravedad nos queda la ecuación:

$$f(y) = -2ky \left(1 - \frac{L_0}{\sqrt{y^2+a^2}}\right) - mg = 0$$

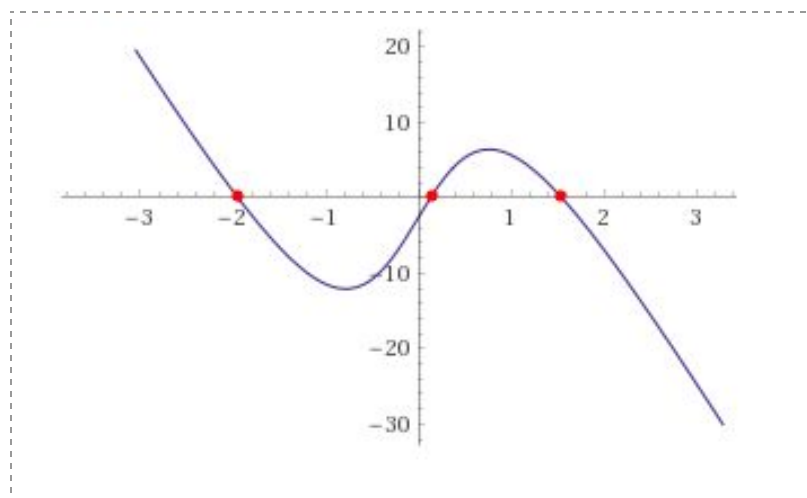


gráfico f(y)

Y sus raíces son: $y_1 \approx -1.94511$, $y_2 \approx 0,148924$, $y_3 \approx 1,54820$

<i>Constante</i>	<i>Lo</i>	<i>k</i>	<i>a</i>	<i>m</i>	<i>g</i>
<i>Valor</i>	<i>2,02</i>	<i>10</i>	<i>1</i>	<i>0,303</i>	<i>9,81</i>

Buscamos las tres raíces de la función utilizando los mismo métodos numéricos usados en el punto 1, completando las mismas tablas:

a. Regula-Falsi

Primero busco el punto de equilibrio $y_3 \approx 1,54820$ en el intervalo $[1,2]$:

k	a_k	b_k	$f(a_k)$	$f(b_k)$
0	1.0000000000	2	5.594684	-6.837571
1	1.4500135945	2	1.285195	-6.837571
2	1.5370331899	2	0.150678	-6.837571
3	1.5470155378	2	0.015994	-6.837571
4	1.5480726288	2	0.001679	-6.837571
5	1.5481835844	2	0.000176	-6.837571
6	1.5481952197	2	1.846E-5	-6.837571
7	1.5481964397	2	1.936E-6	-6.837571
8	1.5481965677	2	2.03E-7	-6.837571
9	1.5481965811	2	2.128E-8	-6.837571
10	1.5481965825	2	2.232E-9	-6.837571
11	1.5481965826	2	2.340E-10	-6.837571
12	1.5481965827	2	2.453E-11	-6.837571

k	r_{k+1}	Δr_{k+1}	$\Delta r/r$	λ	p
0	1.4500135945				
1	1.5370331899	0.0870196	0.0566153		
2	1.5470155378	0.0099823	0.0064526	0.125540	1.036938
3	1.5480726288	0.0010571	0.0006828	0.107836	1.003941
4	1.5481835844	0.0001110	7.1668e-5	0.105261	1.000414
5	1.5481952197	1.1636e-5	7.5154e-6	0.104907	1.000043
6	1.5481964397	1.2200e-6	7.8803e-7	0.104860	1.000005
7	1.5481965677	1.2792e-7	8.2628e-8	0.104854	1.000000
8	1.5481965811	1.3413e-8	8.6638e-9	0.104853	0.999999
9	1.5481965825	1.4064e-9	9.084e-10	0.104854	1.000000
10	1.5481965826	1.4747e-10	9.525e-11	0.104842	0.999994
11	1.548196582704	1.5463e-11	9.988e-12	0.105143	1.000121
12	1.548196582706	1.6209e-12	1.047e-12		

Luego busco el punto de equilibrio $y_2 \approx 0,148924$ en el intervalo $[0,1]$:

k	a_k	b_k	$f(a_k)$	$f(b_k)$
0	0	1.00000000000000	-2.97243	5.5946839
1	0	0.3469581487885	-2.97243	3.3310842
2	0	0.1636085491665	-2.97243	0.2784571
3	0	0.1495945386617	-2.97243	0.0127892
4	0	0.1489536510251	-2.97243	5.5698e-4
5	0	0.1489257448453	-2.97243	2.4197e-5
6	0	0.1489245325076	-2.97243	1.0511e-6
7	0	0.1489244798449	-2.97243	4.5659e-8
8	0	0.1489244775573	-2.97243	1.9833e-9
9	0	0.1489244774579	-2.97243	8.6155e-11
10	0	0.1489244774536	-2.97243	3.7418e-12

k	r_{k+1}	Δr_{k+1}	$\Delta r/r$	λ	p
0	0.3469581487885				
1	0.1636085491665	0.1833496	1.1206602		
2	0.1495945386617	0.0140140	0.0936799	0.107261	1.199749
3	0.1489536510251	6.4088e-4	4.3026e-3	0.0489426	1.015899
4	0.1489257448453	2.7906e-5	1.8738e-4	0.043778	1.000731
5	0.1489245325076	1.2123e-6	8.1406e-6	0.043457	1.000032
6	0.1489244798449	5.2662e-8	3.5362e-7	0.043440	1.000001
7	0.1489244775573	2.2876e-9	1.5360e-8	0.043439	1.000000
8	0.1489244774579	9.937e-11	6.673e-10	0.043439	1.000001
9	0.1489244774536	4.316e-12	2.898e-11	0.043466	1.000028
10	0.1489244774534	1.875e-13	1.259e-12		

Y por último, busco el punto de equilibrio $y_3 \approx 1,54820$ en el intervalo $[-2,0]$:

k	a_k	b_k	$f(a_k)$	$f(b_k)$
0	-2	0	0.892711	-2.97243
1	-2	-1.5380704756136	0.892711	-6.081584
2	-2	-1.9408729155165	0.892711	-0.068365
3	-2	-1.9450788881169	0.892711	-5.3047e-4
4	-2	-1.9451115041864	0.892711	-4.1023e-6
5	-2	-1.9451117564186	0.892711	-3.1724e-8
6	-2	-1.9451117583692	0.892711	-2.453e-10
7	-2	-1.9451117583842	0.892711	-1.899e-12

k	r_{k+1}	Δr_{k+1}	$\Delta r/r$	λ	p
0	-1.5380704756136				
1	-1.9408729155165	0.402802	0.207536		
2	-1.9450788881169	4.2059e-3	2.1624e-3	0.011079	1.065218
3	-1.9451115041864	3.2616e-5	1.6768e-5	0.007779	1.000567
4	-1.9451117564186	2.5223e-7	1.2967e-7	0.007733	1.000004
5	-1.9451117583692	1.9506e-9	1.0028e-9	0.007733	0.999999
6	-1.9451117583842	1.508e-11	7.755e-12	0.007693	0.999742
7	-1.9451117583844	1.1679e-13	6.0046e-14		

b. Punto Fijo

Con la función que se pide en el enunciado: $g(y) = y - f(y)$ busco el punto de equilibrio $y_3 \approx 1,54820$ con la semilla $y=2$ y obtengo en las primeras 10 iteraciones:

k	y_k	$\Delta y = y_k - y_{k+1} $	$\Delta y/y_k$	λ	p
0	2.00				
1	8.83	6.83	0.77369348		
2	148.41	139.58	0.94045469	20.049671	1.009359
3	30.79e02	2930.92	0.95180218	20.995205	1.000028
4	64.62e03	61549.43	0.95235337	20.999998	1.000000
5	13.57e05	1292538.12	0.95237963	20.999999	1.000000
6	28.50e06	27143300.65	0.95238088	20.999999	1.000000
7	59.85e07	570009313.84	0.95238094	21.000000	1.000000
8	12.56e08	11970195590.81	0.95238095	21.000000	1.000000
9	26.39e10	251374107407.1	0.95238095	21.000000	1.000000
10	55.42e11	5278856255549	0.95238095		

Con estos datos y como el algoritmo resulta en 'overflow' se entiende que no converge a ninguna raíz. Se probó el algoritmo con esta semilla y con otras dentro del intervalo al que pertenecen las tres raíces, y en todos los casos se observó el mismo comportamiento, entonces no se puede hallar las raíces con esta función $g(y)$.

Analizamos esa función gráficamente y pudimos observar que:

- La función $g(y)$ no cumple con la primer condición del teorema en los intervalos donde se encuentran las raíces:

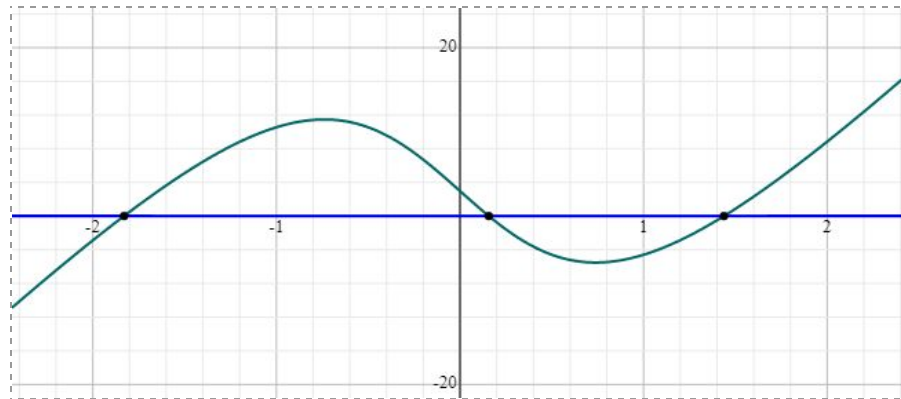


gráfico $g(y)$

- Luego observamos que el término $M \cdot G$ no influye en la derivada, por lo que la función derivada y su gráfico coinciden con el del punto 1.b de punto fijo, que no cumplía con la segunda condición del teorema, entonces no se cumple aquí tampoco.

Utilizando la función que habíamos hallado para poder aplicar el método en

el punto 1: $g(y) = \frac{yLo}{\sqrt{y^2+a^2}} - \frac{mg}{2k}$. cuya derivada es: $g'(y) = \frac{Lo\sqrt{y^2+a^2} - \frac{y^2Lo}{\sqrt{y^2+a^2}}}{y^2+a^2}$

analizamos si cumple con las condiciones del teorema de punto fijo:

- Se puede observar que la primera condición del teorema se cumple en el intervalo $[-4;2]$:

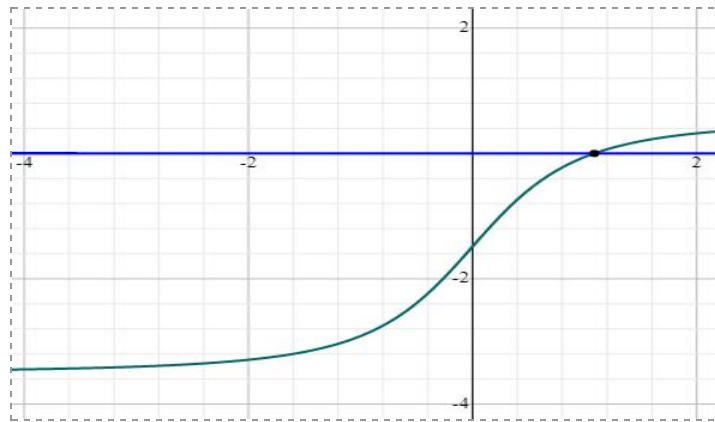


gráfico $g(y)$

- Pero al ver el gráfico de la derivada, vemos que dentro de ese intervalo, existe un subintervalo $[-0.8;0.8]$ donde la segunda condición no se cumple:

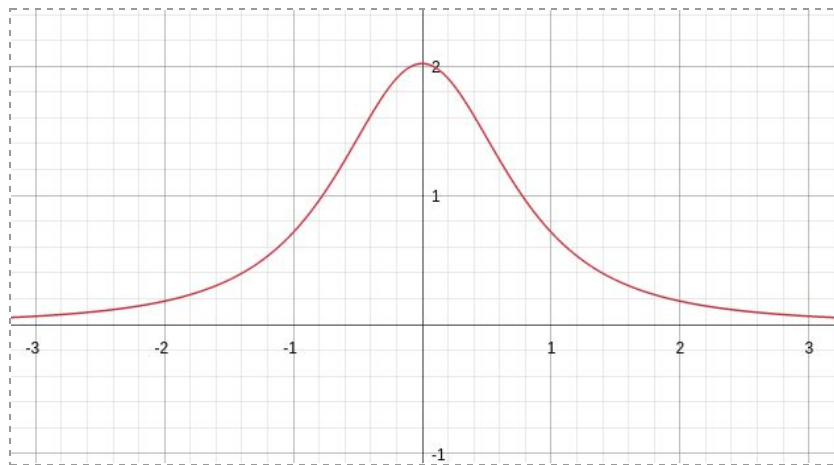


gráfico $g'(y)$

Sin embargo se observó, al utilizar esta función en el algoritmo, que pueden encontrarse las raíces $y_1 \approx -1.94511$ e $y_3 \approx 1.54820$, pero no $y_2 \approx 0.148924$ la cual se encuentra dentro del intervalo donde la segunda condición no se cumple. Entonces con la nueva función $g(y)$ se obtienen los siguientes datos:

Busco el punto de equilibrio $y_3 \approx 1,54820$ utilizando como semilla el $y=2$:

k	y_k	$\Delta y = y_k - y_{k+1} $	$\Delta y / y_k$	λ	p
0	2.000000000000000				
1	1.65812142581983	0.341879	0.206184		
2	1.58115002596302	0.076971	0.048681	0.186228	0.823202
3	1.55859354840737	0.022556	0.014472	0.254928	0.945654
4	1.55152740138758	0.007066	0.004554	0.293325	0.982655
5	1.54926881163704	0.002259	0.001458	0.310925	0.994421
6	1.54854227845543	7.2653e-4	4.6917e-4	0.318170	0.998201
7	1.54830809332029	2.3418e-4	1.5125e-4	0.320984	0.999419
8	1.54823255829427	7.5535e-5	4.8788e-5	0.322039	0.999813
9	1.54820818976082	2.4369e-5	1.5740e-5	0.322427	0.999939
10	1.54820032763364	7.8621e-6	5.0782e-6	0.322567	0.999980
11	1.54819779098506	2.5366e-6	1.6384e-6	0.322617	0.999994
12	1.54819697255112	8.1843e-7	5.2864e-7	0.322635	0.999998
13	1.54819670848787	2.6406e-7	1.7056e-7	0.322641	0.999999
14	1.54819662328925	8.5198e-8	5.5031e-8	0.322643	0.999999
15	1.54819659580035	2.7488e-8	1.7755e-8	0.322644	0.999999
16	1.54819658693119	8.8691e-9	5.7287e-9	0.322644	0.999999
17	1.54819658406961	2.8615e-9	1.8483e-9	0.322646	1.000000
18	1.54819658314633	9.2328e-10	5.964e-10	0.322637	0.999998
19	1.54819658284844	2.9789e-10	1.924e-10	0.322649	1.000001
20	1.54819658275233	9.6113e-11	6.208e-11	0.322696	1.000007
21	1.54819658272132	3.1010e-11	2.003e-11	0.322387	0.999966
22	1.54819658271131	1.0006e-11	6.463e-12	0.323675	1.000131

23	1.54819658270808	3.2279e-12	2.085e-12	0.319145	0.999574
24	1.54819658270704	1.0418e-12	6.729e-13		

Busco el punto de equilibrio $y_1 \approx -1.94511$ utilizando como semilla el $y=-2$:

k	y_k	$\Delta y = y_k - y_{k+1} $	$\Delta y / y_k$	λ	p
0	2.00000000000000				
1	-1.955364425819	0.044635	0.022827		
2	-1.947079031893	0.008285	0.004255	0.174991	0.981029
3	-1.945491145001	0.001587	8.161e-4	0.188335	0.996361
4	-1.945184993555	3.061e-4	1.573e-4	0.191934	0.999298
5	-1.945125898028	5.909e-5	3.038e-5	0.192815	0.999864
6	-1.945114488448	1.140e-5	5.865e-6	0.193020	0.999973
7	-1.945112285505	2.202e-6	1.132e-6	0.193067	0.999994
8	-1.945111860161	4.253e-7	2.186e-7	0.193077	0.999999
9	-1.945111778035	8.212e-8	4.222e-8	0.193079	0.999999
10	-1.945111762178	1.585e-8	8.152e-9	0.193080	1.000000
11	-1.945111759117	3.061e-9	1.574e-9	0.193079	0.999999
12	-1.945111758525	5.911e-10	3.039e-10	0.193082	1.000000
13	-1.945111758411	1.141e-10	5.867e-11	0.193071	0.999997
14	-1.945111758389	2.203e-11	1.132e-11	0.193121	1.000009
15	-1.945111758385	4.255e-12	2.187e-12	0.193823	1.000157
16	-1.945111758384	8.213e-13	4.222e-13		

c. Newton-Raphson

Busco el punto de equilibrio $y_3 \approx 1.54820$ utilizando como semilla el $y=2$:

k	y_k	$\Delta y = y_k - y_{k+1} $	$\Delta y / y_k$	λ	p
0	2.000				
1	1.582731786535113	0.417268	0.263638		
2	1.548562687612221	0.034169	0.022065	0.166622	1.812764
3	1.548196626325567	3.666e-4	2.364e-4	0.304907	1.991735
4	1.548196582706551	4.362e-8	2.817e-8	0.305392	1.991936
5	1.548196582706550	6.66e-16	4.30e-16		

Luego busco el punto de equilibrio $y_2 \approx 0.148924$ con la semilla $y=0.5$:

k	y_k	$\Delta y = y_k - y_{k+1} $	$\Delta y / y_k$	λ	p
0	0.500				
1	-0.0719649749810105	0.571964	7.947824		
2	0.1487129144359895	0.220678	1.483919	13.01298	7.297588
3	0.1489244574458927	2.115e-4	0.001420	0.001586	1.333242
4	0.1489244774534454	2.001e-8	1.343e-7	0.414793	1.991138
5	0.1489244774534455	1.94e-16	1.30e-15		

Y por último, busco el punto de equilibrio $y_1 \approx -1.94511$ con la semilla $y=-2$:

k	y_k	$\Delta y = y_k - y_{k+1} $	$\Delta y / y_k$	λ	p
0	-2.0000000000000000				
1	-1.9455215749047685	0.054478	0.028002		
2	-1.9451117828834088	4.0979e-4	2.1068e-4	0.133625	1.988743
3	-1.9451117583844084	2.4499e-8	1.2595e-8	0.069165	1.904314
4	-1.9451117583844082	2.220e-16	1.141e-16		

d. Gráfico representativo del error relativo en función de las iteraciones

Punto de equilibrio $y_3 \approx 1.54820$, con función de punto fijo $g(y) = y - f(y)$:

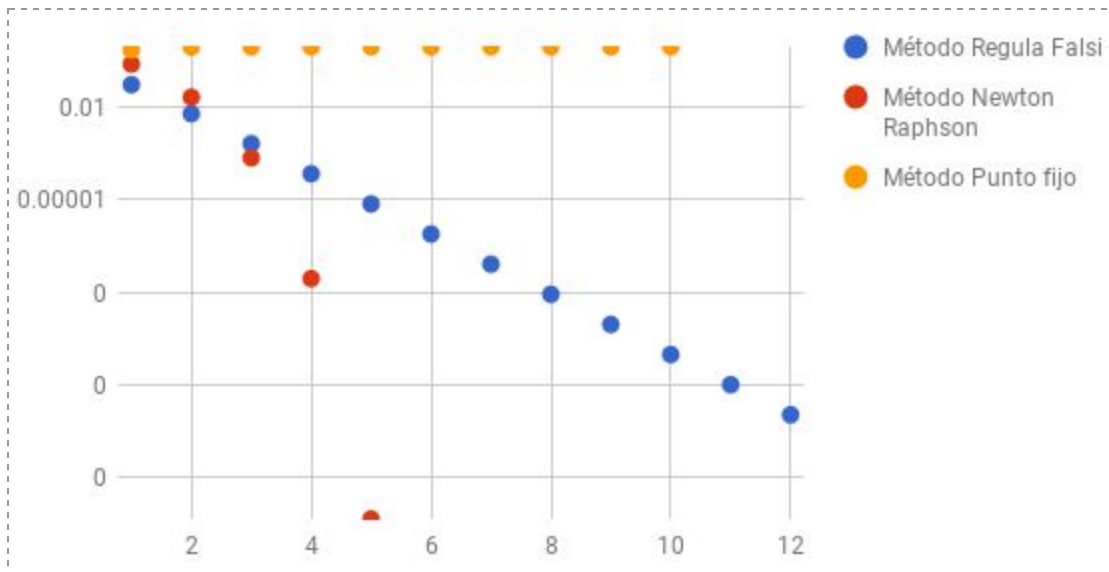


gráfico (3) del error relativo en función de las iteraciones.

Con función de punto fijo $g(y) = \frac{yL_0}{\sqrt{y^2+a^2}} - \frac{mg}{2k}$:

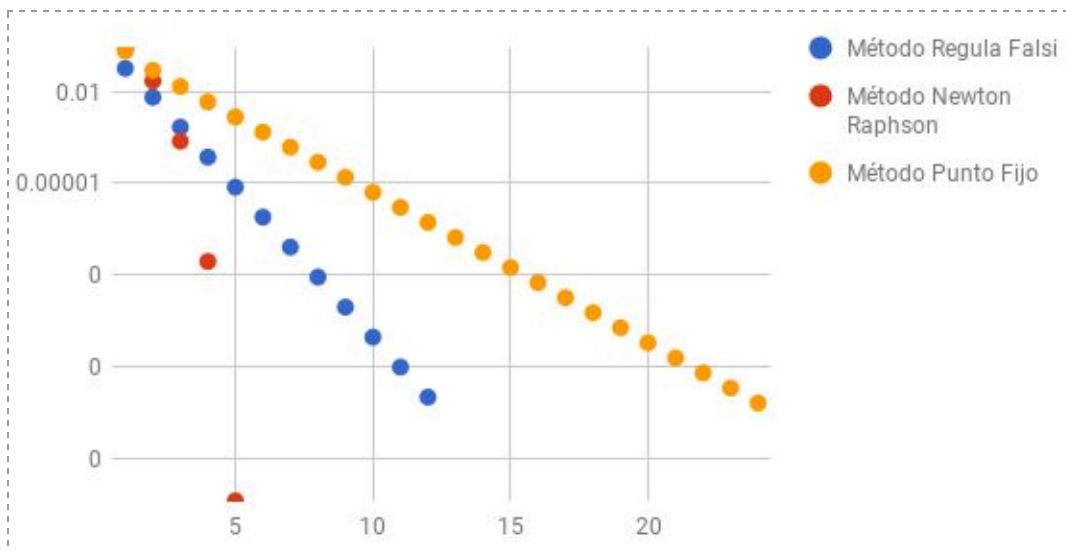


gráfico (4) del error relativo en función de las iteraciones.

Punto de equilibrio $y_2 \approx 0.14892$, con función de punto fijo $g(y) = y - f(y)$:

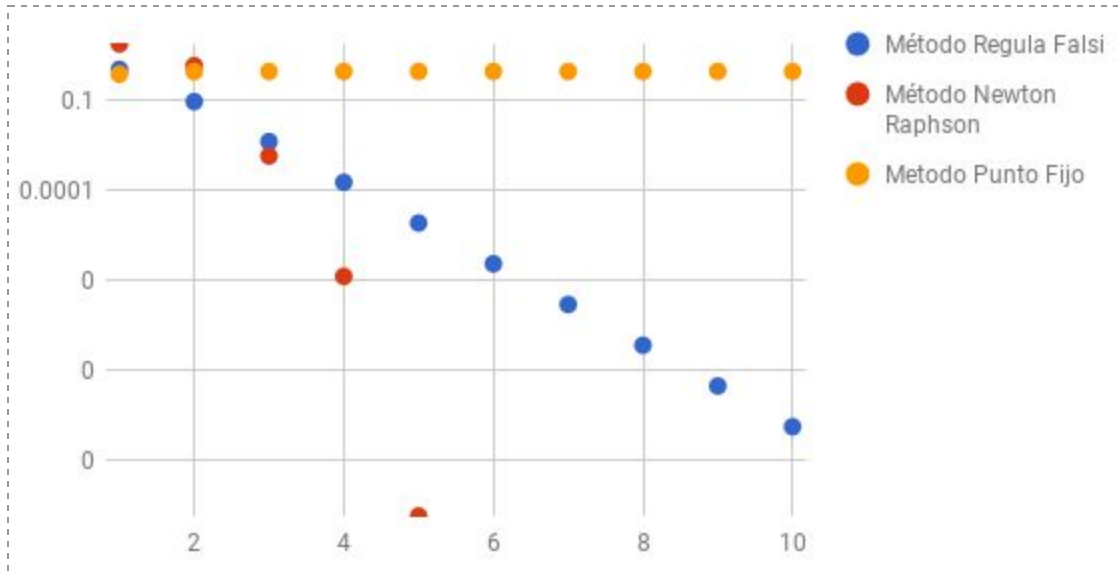


gráfico (5) del error relativo en función de las iteraciones.

Punto de equilibrio $y_1 \approx -1.94511$, con función de punto fijo $g(y) = \frac{yLo}{\sqrt{y^2+a^2}} - \frac{mg}{2k}$:

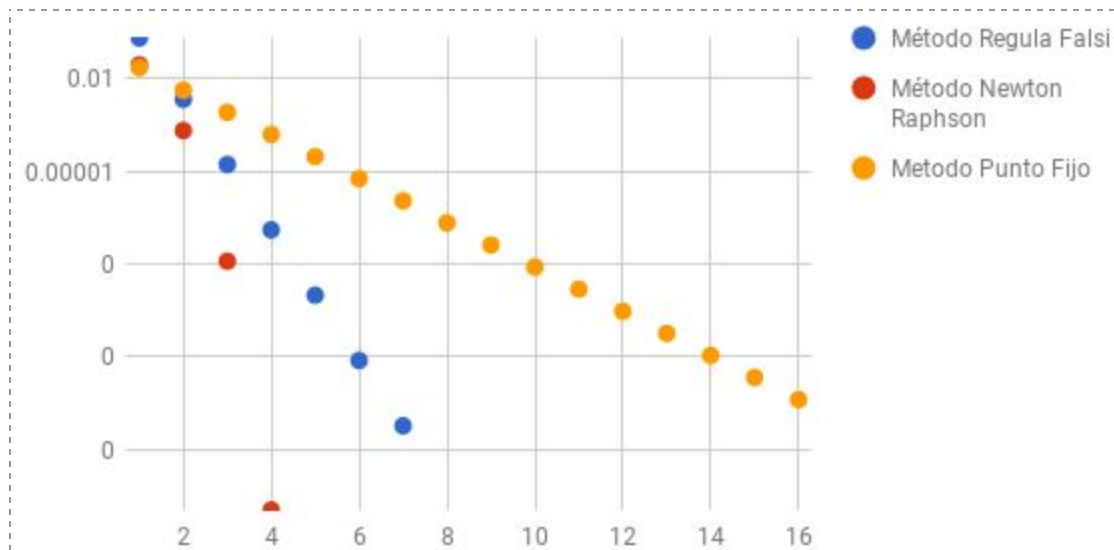


gráfico (6) del error relativo en función de las iteraciones.

e. Comparaciones entre punto 1 y 2 :

El comportamiento de los tres métodos fue el mismo en ambos casos, y esto se puede apreciar en los gráficos. El método Regula-Falsi siempre fue acercándose al resultado, lento pero seguro de manera lineal, en cambio Newton-Raphson comenzaba las primeras iteraciones con un error mayor que el Regula-Falsi, pero rápidamente lo corregía y terminaba en muchas menos iteraciones y un menor error de manera cuadrática. En el caso de Punto Fijo, cuando se utilizó la función $g(y) = y - f(y)$ se puede apreciar fácilmente que el método no converge a ninguna raíz, ya que el error relativo es siempre el mismo entonces en el gráfico se ve algo parecido a una función constante, en cambio cuando se utilizó la función alternativa que sí converge, en ambos casos de manera lineal pero mucho menos eficiente que el método Regula-Falsi.

3. Máximo intervalo de convergencia.

Se busco de manera práctica el máximo intervalo de convergencia, proponiendo semillas cada vez más grandes que converjan a una raíz determinada, hasta llegar al límite (con la próxima semilla el método no converge a esa raíz o ninguna) para el extremo derecho, y semillas cada vez más chicas para el izquierdo, y los resultados fueron:

raiz	extremo izquierdo	extremo derecho
1.548197	0.773	3.638e16
0.148924	-0.467	0.600
-1.945112	-3.638e16	-0.773

4. Todos los puntos de equilibrio del sistema, con método Newton-Raphson en cada caso:

m	0	$0.3 * m_0$	$0.6 * m_0$	$0.9 * m_0$	$1.2 * m_0$	$1.5 * m_0$
raiz 1	1.755107	1.548197	1.307084	0.928938	-	-
raiz 2	-6.162e-33	0.148924	0.322146	0.626155	-	-
raiz 3	-1.755106	-1.945112	-2.124972	-2.298102	-2.466479	-2.631349

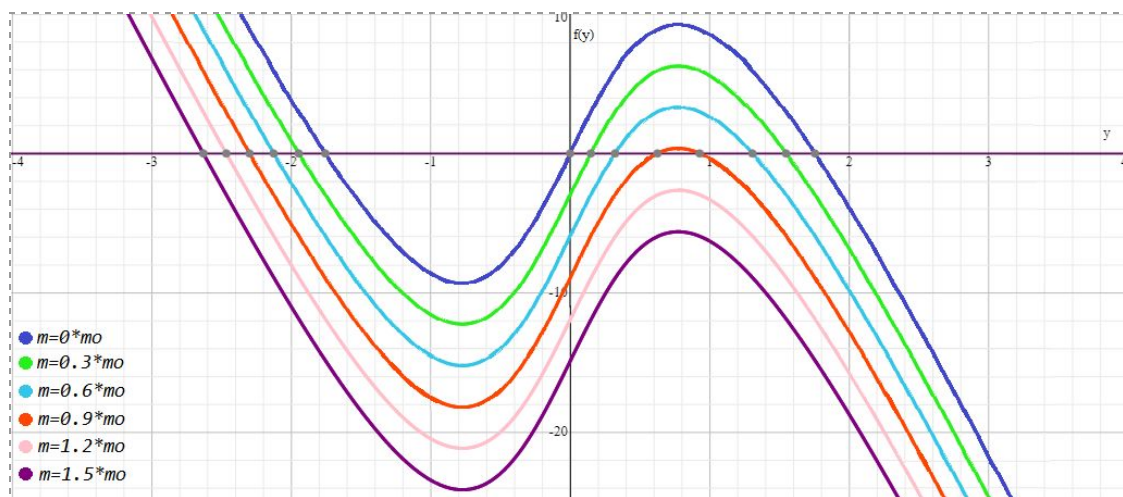
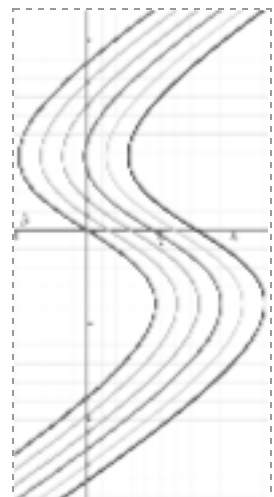


gráfico de las diferentes funciones $f(y)$ debido a los diferentes valores de masa.

Si rotamos la imagen 90° , y recordamos que los cortes con el eje 'y' son los puntos de equilibrio del sistema, es decir donde la fuerza resultante en la masa es cero, podemos observar que a medida que la masa aumenta, ésta tiende a quedar estable por debajo del eje 'y'.

Cuando el valor de la masa es cero, el sistema oscila continuamente, entre sus puntos de equilibrio, entonces éstos son inestables. A medida que la masa aumenta, este movimiento oscilatorio continuo desaparece; el punto de equilibrio negativo va alejándose más del origen, y los puntos de equilibrio positivo van acercándose más entre ellos, esto se debe a que la fuerza resultante negativa en eje 'y', aumenta por el efecto de la gravedad,



entonces la masa quedará estática sostenida por los resortes, pero cuanto más chica sea la masa, va a oscilar por más tiempo entre sus puntos de equilibrio, entonces el punto de equilibrio negativo es estable, los demás no. Cuando la masa es tal que la fuerza ejercida por la gravedad es mayor a la fuerza elástica resultante, existe un solo punto de equilibrio, negativo y más alejado del origen que los anteriores, entonces la masa cae, y la fuerza elástica no es la suficiente para poder volver a subirla, entonces queda estática en ese punto; cuanto mayor sea la masa, mayor será la velocidad con la que ésta cae, y quede estable sostenida por los resortes en el punto de equilibrio, el cual es estable.

5. Conclusiones

El método Regula-Falsi, siempre converge de manera lineal y el método Newton-Raphson siempre converge de manera cuadrática, en menos iteraciones que el anterior, ambos como se esperaba que lo hicieran. Por otro lado, en el método Punto Fijo, cuando se utilizó una función $g(y)$ que cumpliera el teorema, la cantidad de iteraciones depende mucho de esta función, y con la propuesta por nosotros en este práctico, el número de iteraciones fue mayor que en el método Regula-Falsi, pero tampoco supera el orden lineal como se vio en la teórica.

Además se observó que la precisión con la que se pueden obtener los datos siempre va a depender de las limitaciones de la máquina, ya que no se pudo trabajar con un error relativo menor a $0,5 \cdot 10^{-16}$ porque en este punto, las raíces son tomadas como 0.0 y no se pueden calcular los valores necesarios para completar las tablas.

- Anexo I

#BRACCELARGHE AILIN - 99366 - TARELA - 29/04/18

#JAMILIS NETANEL - 99093

Código fuente: Regula-Falsi

```
import math
```



```

ERROR = math.pow(5, -17)
VACIO = None
# PADRON = 99093
# L0 = 2*100000/PADRON
L0 = 2.02
G = 9.81
K = 10
# M0 = 100000/PADRON
M0 = 1.01
M = 0*M0
A = 1

```

```

def regula_falsi (inicio,fin):
    a = []
    b = []
    raiz = []
    absError = []
    relError = []

    k = 0

    if regula_falsi_rec(k, inicio, fin, raiz, a, b, absError, relError):
        print ("HAY RAIZ")
    else:
        print ("NO HAY RAIZ")

    print a
    print calcular_f_en_arreglo(a)
    print b
    print calcular_f_en_arreglo(b)
    print raiz
    print absError
    print relError
    p = calcular_exp_p(absError)
    print p
    print calcular_lambda(absError, p)

```

```

def regula_falsi_rec(k, inicio, fin, raiz, a, b, absE, relE):

```

```

        medio = calcular_medio(inicio, fin)
        raiz.append(medio)

    a.append(inicio)
    b.append(fin)

    if k == 0:
        absE.append(VACIO)
        relE.append(VACIO)

    else:
        absE.append(abs(raiz[k]-raiz[k-1]))
        relE.append(abs(absE[k]/(raiz[k])))

        if relE[k] < ERROR:
            return True

    k += 1

    if f(inicio)*f(medio) < 0:
        regula_falsi_rec(k, inicio, medio, raiz, a, b, absE, relE)

    if f(medio)*f(fin) < 0:
        regula_falsi_rec(k, medio, fin, raiz, a, b, absE, relE)

    else:
        return False

    return True

def calcular_medio( inicio, fin):
    return inicio-((fin-inicio)/(f(fin)-f(inicio)))*f(inicio)

def calcular_f_en_arreglo(arreglo):
    f_en_arreglo = []

```

```

        for a in arreglo:
            f_en_arreglo.append(f(a))

        return f_en_arreglo

def f(y):
    return -2*K*y*(1 - L0/math.sqrt(math.pow(y, 2) + math.pow(A, 2))) - M*G

def calcular_exp_p(absE):
    p = []

    for k in range(len(absE)-1):
        if k > 1:
            p.append(math.log((absE[k+1]/absE[k]), math.e) /
math.log((absE[k]/absE[k-1]), math.e))

        else:
            p.append(VACIO)

    return p

def calcular_lambda(absE, p):
    y = []

    for k in range(len(absE)-1):
        if k > 1:
            y.append(absE[k+1] / math.pow(absE[k], p[k]))

        else:
            y.append(VACIO)

    return y

def main():
    regula_falsi(1, 2)

```

```
main()
```

Código fuente: Punto Fijo

```
import math
```

```
ERROR = math.pow(5, -17)
```

```
VACIO = None
```

```
# PADRON = 99093
```

```
# L0 = 2*100000/PADRON
```

```
L0 = 2.02
```

```
G = 9.81
```

```
K = 10
```

```
# M0 = 100000/PADRON
```

```
M0 = 1.01
```

```
M = 0.3*M0
```

```
A = 1
```

```
def punto_fijo(semilla):
```

```
    raiz = []
```

```
    absError = []
```

```
    relError = []
```

```
    raiz.append(semilla)
```

```
    absError.append(VACIO)
```

```
    relError.append(VACIO)
```

```
    k = 1
```

```
    punto_fijo_rec(k, semilla, raiz, absError, relError)
```

```
    print raiz
```

```
    print absError
```

```
    print relError
```

```
    p = calcular_exp_p(absError)
```

```

    print p
    print calcular_lambda(absError, p)

def punto_fijo_rec(k, pto_anterior, raiz, absE, relE):

    punto = g(pto_anterior)
    raiz.append(punto)

    absE.append(abs(raiz[k]-raiz[k-1]))
    relE.append(abs(absE[k]/raiz[k]))

# Para observar las primeras 10 iteraciones cuando el metodo no converge
    if k == 10:
        p = calcular_exp_p(absE)
        y = calcular_lambda(absE, p)
        print raiz, "\n", absE, "\n", relE, "\n", p, "\n", y
        return
    if relE[k] < ERROR:
        return

    k+=1
    punto_fijo_rec(k, punto, raiz, absE, relE)

def g(y):
    return y - (-2*K*y*(1 - L0/math.sqrt(math.pow(y, 2) + math.pow(A, 2))) -
M*G)

#Funcion punto fijo alternativa que converge
# return y*L0/math.sqrt(math.pow(y,2)+math.pow(A,2))-M*G/(2*K)

def calcular_exp_p(absE):
    p = []

    for k in range(len(absE)-1):
        if k > 1:

```

```

        p.append(math.log((absE[k+1]/absE[k]), math.e) /
math.log((absE[k]/absE[k-1]), math.e))

    else:
        p.append(VACIO)

    return p

def calcular_lambda(absE, p):
    y = []

    for k in range(len(absE)-1):
        if k > 1:
            y.append(absE[k] / math.pow(absE[k - 1], p[k]))

        else:
            y.append(VACIO)

    return y

def main():
    punto_fijo(2) # el parametro es la semilla

main()

```

Código fuente: Newton-Raphson

```

import math

ERROR = math.pow(5, -17)
VACIO = None
# PADRON = 99093
# L0 = 2*100000/PADRON
L0 = 2.02

```

```

G = 9.81
K = 10
# M0 = 100000/PADRON
M0 = 1.01
M = 1.5*M0
A = 1

def newton_raphson(semilla):

    raiz = []
    absError = []
    relError = []

    raiz.append(semilla)
    absError.append(VACIO)
    relError.append(VACIO)

    k = 1
    newton_raphson_rec(semilla, raiz, absError, relError, k)

    print raiz
    print absError
    print relError
    p = calcular_exp_p(absError)
    print p
    print calcular_lambda(absError, p)

def newton_raphson_rec(pto_anterior, raiz, errorAbs, errorRel, k):

    punto = calcular_proximo_punto(pto_anterior)
    raiz.append(punto)

    errorAbs.append(abs(raiz[k]-raiz[k-1]))
    errorRel.append(abs(errorAbs[k]/raiz[k]))

    if errorRel[k] < ERROR:
        print errorRel[k], ERROR
        return

```

```

        k+=1
        newton_raphson_rec(punto, raiz, errorAbs, errorRel, k)

def calcular_proximo_punto(xn):
    return xn-f(xn)/derivada(xn)

def f(y):
    return -2*K*y*(1 - L0/math.sqrt(math.pow(y, 2)+math.pow(A, 2))) - M*G

def derivada(y):
    return -2*K*(1 - L0/math.sqrt(math.pow(y,2)+math.pow(A,2)))*(1 -
math.pow(y,2)/(math.pow(y,2)+math.pow(A,2)))

def calcular_exp_p(absE):
    p = []

    for k in range(len(absE)-1):
        if k > 1:
            p.append(math.log((absE[k+1]/absE[k]), math.e) /
math.log((absE[k]/absE[k-1]), math.e))

        else:
            p.append(VACIO)

    return p

def calcular_lambda(absE, p):
    y = []

    for k in range(len(absE)-1):
        if k > 1:
            y.append(absE[k+1] / math.pow(absE[k], p[k]))

        else:

```



```

        y.append(VACIO)

    return y

def main():
    newton_raphson(0) # el parametro es la semilla

main()

```

- Anexo II

#BRACCELARGHE AILIN - 99366 - TARELA - 29/04/18

#JAMILIS NETANEL - 99093

Ejercicio 1:

#Regula Falsi [1,2]

[1, 1.6891037590760076, 1.7521035983812254, 1.7549763150521807, 1.7551011736781876, 1.7551065888710613, 1.75510682370935, 1.7551068338934375, 1.7551068343350849, 1.7551068343542373]

[8.567113959936519, 0.9822784468385077, 0.045315849273563226, 0.0019705943969325285, 8.546774093034366e-05, 3.7064455000459508e-06, 1.607351482822961e-07, 6.970505808793923e-09, 3.022840127365427e-10, 1.3109911802295158e-11]

[2, 2, 2, 2, 2, 2, 2, 2, 2, 2]

[-3.8651414836034004, -3.8651414836034004, -3.8651414836034004, -3.8651414836034004, -3.8651414836034004, -3.8651414836034004, -3.8651414836034004, -3.8651414836034004, -3.8651414836034004, -3.8651414836034004]

[1.6891037590760076, 1.7521035983812254, 1.7549763150521807, 1.7551011736781876, 1.7551065888710613, 1.75510682370935, 1.7551068338934375, 1.7551068343350849, 1.7551068343542373, 1.755106834355068]

[None, 0.0629998393052178, 0.002872716670955233, 0.00012485862600697395, 5.415192873670094e-06, 2.3483828859305333e-07, 1.0184087573250622e-08, 4.4164738532970205e-10, 1.9152457397808575e-11, 8.306688670245421e-13]

[None, 0.035956686216170985, 0.0016368976870607048, 7.114041508234317e-05, 3.085392595530801e-06, 1.3380284631150356e-07, 5.802545677894019e-09, 2.516356136787641e-10, 1.0912416852878023e-11, 4.732867827557517e-13]

[None, None, 1.0155308514239914, 1.0006831605011643, 1.0000296454064428, 1.000001290515331, 0.999997651597362, 1.0000034465588927, 0.999959946211525]

[None, None, 0.0475993074262518, 0.0436377299895276, 0.04338215296614661, 0.043367239011931076, 0.04336622927866549, 0.043369167298185476, 0.043328548569987066]

#Punto Fijo diverge (1.5)

[1.5, -2.1148318912488886, -7.888693843718579, -125.58330494295761,
-2596.8506845595894, -54493.464378746794, -1144322.3519536895, -24030728.991027478,
-504645268.411577, -10597550596.243118, -222548562480.7055]

[None, 3.6148318912488886, 5.773861952469691, 117.69461109923903, 2471.267379616632,
51896.6136941872, 1089828.8875749428, 22886406.63907379, 480614539.4205495,
10092905327.831541, 211951011884.4624]

[None, 1.7092762342987902, 0.7319160898945475, 0.9371835782845357,
0.9516401517847547, 0.9523456488926698, 0.952379271202987, 0.9523808723247242,
0.9523809485687508, 0.952380952199419, 0.952380952372308]

[None, None, 6.437708645012134, 1.009832026908522, 1.000042449866469,
1.0000000080861802, 1.0000000000009006, 1.0000000000000002, 1.0, 1.0]

[None, None, 0.0014745441385488835, 20.03564942518405, 20.99303621873272,
20.99999815631617, 20.999999999737078, 20.9999999999993, 20.99999999999996,
21.000000000000004]

[1.5, -2.1148318912488886, -7.888693843718579, -125.58330494295761,
-2596.8506845595894, -54493.464378746794, -1144322.3519536895, -24030728.991027478,
-504645268.411577, -10597550596.243118, -222548562480.7055]

[None, 3.6148318912488886, 5.773861952469691, 117.69461109923903, 2471.267379616632,
51896.6136941872, 1089828.8875749428, 22886406.63907379, 480614539.4205495,
10092905327.831541, 211951011884.4624]

[None, 1.7092762342987902, 0.7319160898945475, 0.9371835782845357,
0.9516401517847547, 0.9523456488926698, 0.952379271202987, 0.9523808723247242,
0.9523809485687508, 0.952380952199419, 0.952380952372308]

[None, None, 6.437708645012134, 1.009832026908522, 1.000042449866469,
1.0000000080861802, 1.0000000000009006, 1.0000000000000002, 1.0, 1.0]

[None, None, 0.0014745441385488835, 20.03564942518405, 20.99303621873272,
20.99999815631617, 20.999999999737078, 20.9999999999993, 20.99999999999996,
21.000000000000004]

#Punto Fijo converge (1.5)

[1.5, 1.6807415945624444, 1.7359719820717703, 1.7503588957757894, 1.7539396645525198,
1.7548205758301194, 1.7550366668708108, 1.7550896373496592, 1.7551026197692177,
1.7551058014668226, 1.755106581220861, 1.7551067723184706, 1.755106819151538,
1.7551068306291064, 1.7551068334419602, 1.7551068341313174, 1.755106834300261,
1.7551068343416647, 1.7551068343518117, 1.7551068343542984, 1.7551068343549077]

[None, 0.18074159456244443, 0.05523038750932585, 0.014386913704019122,
0.003580768776730414, 0.0008809112775995942, 0.00021609104069142404,
5.297047884833539e-05, 1.2982419558493774e-05, 3.1816976049370282e-06,
7.797540384490986e-07, 1.9109760951430133e-07, 4.6833067424145725e-08,
1.1477568451212505e-08, 2.812853727718334e-09, 6.89357237959598e-10,
1.6894352583562977e-10, 4.140376930195089e-11, 1.014699435586408e-11,
2.4866775305554256e-12, 6.092903959142859e-13]

[None, 0.10753681300396314, 0.03181525282649548, 0.00821940788185762,
0.002041557557023474, 0.0005019950698850669, 0.00012312622566268617,
3.0181067519904857e-05, 7.3969575409789204e-06, 1.812823820807804e-06,
4.4427731443334783e-07, 1.0888090259139287e-07, 2.6683884372796173e-08,
6.539526968337561e-09, 1.6026680963927502e-09, 3.9277223730986863e-10,
9.62582576364848e-11, 2.359045528842768e-11, 5.781411226520307e-12,
1.4168240256840383e-12, 3.4715288208551223e-13]

[None, None, 1.1346535698652511, 1.0338592178278625, 1.0083655953593345,
1.0020545324620156, 1.0005037777265897, 1.0001234788011846, 1.0000302624832043,

1.0000074162165442, 1.0000018175456153, 1.0000004460436394, 1.0000001006047903,
1.000000050029969, 1.000000108531073, 0.9999999627480559, 0.999998178040932,
1.0000012898458164, 1.0000253600059945, 1.0001263904663362]

[None, None, 0.38473328319775546, 0.28732830725727293, 0.2578804009051593,
0.2488750310065132, 0.24617484108493273, 0.2453859724435193, 0.24516086371262044,
0.24509784972163973, 0.24508048061715076, 0.24507575325687045, 0.2450744435497327,
0.24507423437023443, 0.24507449649397087, 0.24507379304993568, 0.2450645664738487,
0.24508172653051344, 0.24522280208044497, 0.24585075450321228]

#Newton Raphson (1.5)

[1.5, 1.7758434546159454, 1.7551942120879342, 1.7551068359540396, 1.7551068343551053,
1.7551068343551057]

[None, 0.2758434546159454, 0.020649242528011147, 8.737613389464549e-05,
1.5989343182809534e-09, 4.440892098500626e-16]

[None, 0.15533095211683565, 0.011764648257042355, 4.97839402734419e-05,
9.110182280547408e-10, 2.530268819864964e-16]

[None, None, 2.1083662064589177, 1.9960141752167906, 1.3839101472550601]

[None, None, 0.3120268895820852, 0.20177533522385527, 0.0006615737693348575]

Ejercicio 2:

#Regula Falsi [1,2]:

[1, 1.4500135945037746, 1.5370331899905978, 1.5470155378389647, 1.5480726288921058,
1.5481835844072, 1.5481952197730047, 1.54819643979781, 1.5481965677220424,
1.5481965811353693, 1.5481965825418063, 1.5481965826892763, 1.5481965827047393]

[5.594683959936519, 1.2851951624854525, 0.15067849493508545, 0.015993577956587846,
0.0016791471929189505, 0.00017608854290429576, 1.8463791686240683e-05,
1.9359992000644866e-06, 2.029966532646199e-07, 2.1284950690869664e-08,
2.2318036307922284e-09, 2.3401680593337915e-10, 2.453104386290761e-11]

[2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2]

[-6.8375714836034005, -6.8375714836034005, -6.8375714836034005, -6.8375714836034005,
-6.8375714836034005, -6.8375714836034005, -6.8375714836034005, -6.8375714836034005,
-6.8375714836034005, -6.8375714836034005, -6.8375714836034005, -6.8375714836034005,
-6.8375714836034005]

[1.4500135945037746, 1.5370331899905978, 1.5470155378389647, 1.5480726288921058,
1.5481835844072, 1.5481952197730047, 1.54819643979781, 1.5481965677220424,
1.5481965811353693, 1.5481965825418063, 1.5481965826892763, 1.5481965827047393,
1.5481965827063602]

[None, 0.08701959548682314, 0.009982347848366935, 0.0010570910531411126,
0.0001109555150942132, 1.1635365804618658e-05, 1.220024805403952e-06,
1.279242323182217e-07, 1.3413326893640942e-08, 1.4064369668886911e-09,
1.4747003618253984e-10, 1.5462964242374255e-11, 1.6209256159527285e-12]

[None, 0.05661530021180314, 0.006452648731835839, 0.0006828433197592485,
7.166818987859124e-05, 7.515438399509222e-06, 7.880297189956615e-07,
8.262790073643204e-08, 8.663839629334593e-09, 9.084356487724728e-10,
9.525278497022567e-11, 9.98772663311274e-12, 1.0469766139899578e-12]

[None, None, 1.0369380254905185, 1.003941258278814, 1.000414059420806,
1.0000434244674417, 1.000004554289576, 1.00000047909091, 0.9999999237517996,
1.0000002554406469, 0.9999944345536467, 1.0001211246202468]

[None, None, 0.125540568591622, 0.10783636460467029, 0.10526129072643366,
0.1049066176726125, 0.10486029882841756, 0.10485448023833555, 0.10485355602915568,
0.10485418646613927, 0.10484174703919148, 0.10514285774470315]

#Regula Falsi [0,1]:

[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]

[-2.97243, -2.97243, -2.97243, -2.97243, -2.97243, -2.97243, -2.97243, -2.97243,
-2.97243, -2.97243, -2.97243]

[1, 0.34695814878853615, 0.16360854916654227, 0.1495945386617274, 0.1489536510251409,
0.1489257448453728, 0.14892453250767593, 0.14892447984493595, 0.14892447755732902,
0.1489244774579581, 0.14892447745364154]

[5.594683959936519, 3.331084183441032, 0.27845712562422653, 0.012789170486980694,
0.000556983388025234, 2.4197416501170466e-05, 1.0511120018819042e-06,
4.565905875608678e-08, 1.983375241820795e-09, 8.61555271569614e-11,
3.741895682196628e-12]

[0.34695814878853615, 0.16360854916654227, 0.1495945386617274, 0.1489536510251409,
0.1489257448453728, 0.14892453250767593, 0.14892447984493595, 0.14892447755732902,
0.1489244774579581, 0.14892447745364154, 0.14892447745345405]

[None, 0.18334959962199388, 0.014014010504814878, 0.0006408876365864846,
2.7906179768111272e-05, 1.212337696870902e-06, 5.266273997439619e-08,
2.2876069372124164e-09, 9.937092815270887e-11, 4.316547119742609e-12,
1.874889132835733e-13]

[None, 1.1206602622907962, 0.0936799607137012, 0.004302597701873809,
0.00018738318077298057, 8.14061777772185e-06, 3.536204392268484e-07,
1.5360852525615165e-08, 6.672571886697513e-10, 2.8984806215528266e-11,
1.2589529705898915e-12]

[None, None, 1.1997488284850228, 1.0158988433695368, 1.0007312361571394,
1.0000318486105015, 1.0000013860099481, 1.0000000090696164, 1.0000005884931873,
1.000027922601966]

[None, None, 0.10726089994866222, 0.04894259228993285, 0.04377775540672734,
0.0434578533477724, 0.0434398224804036, 0.04343882004702104, 0.04343932081575506,
0.043466677241713665]

#Regula Falsi [-2,0]:

[-2, -2, -2, -2, -2, -2, -2, -2, -2]

[0.8927114836034002, 0.8927114836034002, 0.8927114836034002, 0.8927114836034002,
0.8927114836034002, 0.8927114836034002, 0.8927114836034002, 0.8927114836034002]

[0, -1.5380704756136678, -1.940872915516565, -1.9450788881169874, -1.945111504186467,
-1.945111756418643, -1.9451117583692066, -1.9451117583842907]

[-2.97243, -6.08158455412301, -0.06836569602080012, -0.0005304707178233059,
-4.1023458838829185e-06, -3.172428897357804e-08, -2.4533219900035874e-10,
-1.8989254613188677e-12]

[-1.5380704756136678, -1.940872915516565, -1.9450788881169874, -1.945111504186467,
-1.945111756418643, -1.9451117583692066, -1.9451117583842907, -1.9451117583844075]

[None, 0.4028024399028971, 0.004205972600422436, 3.2616069479685095e-05,
2.522321760345392e-07, 1.9505634885064183e-09, 1.5084156146372152e-11,
1.1679546219056647e-13]

[None, 0.20753674116560636, 0.0021623660747735526, 1.6768226093715177e-05,
1.296749018158994e-07, 1.0028027850398593e-09, 7.754904612217152e-12,
6.004563063645029e-14]

[None, None, 1.0652167823398684, 1.0005668601523208, 1.0000043847551852, 0.9999993726401615, 0.9997423700821245]

[None, None, 0.01107972470348771, 0.00777879130089579, 0.007733721623516349, 0.007733132733541022, 0.007693376976572559]

#Punto fijo diverge(2):

[2, 8.837571483603401, 148.4176076905635, 3079.3431084942176, 64628.77771050884, 1357166.9043506903, 28500467.563794497, 598509781.4121145, 12568705372.226835, 263942812779.336, 5542799068328.628]

[None, 6.837571483603401, 139.5800362069601, 2930.9255008036544, 61549.43460201462, 1292538.1266401815, 27143300.659443807, 570009313.84832, 11970195590.81472, 251374107407.10916, 5278856255549.292]

[None, 0.7736934853980353, 0.9404546965746214, 0.9518021855761508, 0.9523533754222074, 0.9523796391561514, 0.9523808898463559, 0.9523809494031142, 0.9523809522391505, 0.9523809523742, 0.9523809523806308]

[None, None, 1.0093593750820478, 1.0000286007492094, 1.0000000048816242, 1.0000000000005402, 1.0, 1.0, 1.0, 1.0]

[None, None, 20.04967176252373, 20.995205475704218, 20.999998869481328, 20.9999999984036, 20.99999999999996, 21.000000000000004, 21.0, 21.000000000000004]

[2, 8.837571483603401, 148.4176076905635, 3079.3431084942176, 64628.77771050884, 1357166.9043506903, 28500467.563794497, 598509781.4121145, 12568705372.226835, 263942812779.336, 5542799068328.628]

[None, 6.837571483603401, 139.5800362069601, 2930.9255008036544, 61549.43460201462, 1292538.1266401815, 27143300.659443807, 570009313.84832, 11970195590.81472, 251374107407.10916, 5278856255549.292]

[None, 0.7736934853980353, 0.9404546965746214, 0.9518021855761508, 0.9523533754222074, 0.9523796391561514, 0.9523808898463559, 0.9523809494031142, 0.9523809522391505, 0.9523809523742, 0.9523809523806308]

[None, None, 1.0093593750820478, 1.0000286007492094, 1.0000000048816242, 1.0000000000005402, 1.0, 1.0, 1.0, 1.0]

[None, None, 20.04967176252373, 20.995205475704218, 20.999998869481328, 20.9999999984036, 20.99999999999996, 21.000000000000004, 21.0, 21.000000000000004]

#Punto fijo converge(2):

[2, 1.65812142581983, 1.5811500259630278, 1.5585935484073743, 1.5515274013875835, 1.549268811637049, 1.5485422784554332, 1.5483080933202966, 1.5482325582942742, 1.5482081897608233, 1.548200327633648, 1.5481977909850693, 1.5481969725511249, 1.548196708487879, 1.5481966232892506, 1.548196595800351, 1.5481965869311982, 1.5481965840696112, 1.5481965831463351, 1.5481965828484445, 1.5481965827523316, 1.5481965827213215, 1.548196582711316, 1.5481965827080881, 1.5481965827070463]

[None, 0.34187857418017, 0.07697139985680224, 0.022556477555653487, 0.007066147019790803, 0.002258589750534501, 0.0007265331816157516, 0.00023418513513662198, 7.55350260224219e-05, 2.4368533450891405e-05, 7.862127175295441e-06, 2.5366485787436943e-06, 8.184339443939592e-07, 2.640632459360859e-07, 8.519862837452763e-08, 2.748889960102474e-08, 8.869152745560882e-09, 2.8615869673842553e-09, 9.232761222222052e-10, 2.978906010753235e-10, 9.61128954202195e-11, 3.10100833900151e-11, 1.0005551942526836e-11, 3.22786242179518e-12, 1.0418332863082469e-12]

[None, 0.20618428111266576, 0.048680642945264745, 0.01447232832363671, 0.004554316613081605, 0.0014578423922107756, 0.00046917232530481484,

0.00015125228379735426, 4.878790697028145e-05, 1.5739829831707584e-05,
5.078236346398619e-06, 1.6384525242925874e-06, 5.286368329769697e-07,
1.7056181846168372e-07, 5.5030883734597785e-08, 1.775543214317311e-08,
5.7286993269641065e-09, 1.8483356679823228e-09, 5.963558712588486e-10,
1.9241135420106047e-10, 6.208055003541814e-11, 2.0029809997066098e-11,
6.462714137376776e-12, 2.0849176763774014e-12, 6.729334620326993e-13]

[None, None, 0.8232017282579935, 0.9456541555229155, 0.9826552414594637,
0.9944212310443855, 0.9982017808977818, 0.9994199927778471, 0.9998128821954211,
0.9999396293098188, 0.9999805218929636, 0.9999937156355629, 0.9999979726508801,
0.9999993455325574, 0.9999997833495966, 0.9999999444410941, 0.9999999278292574,
1.000000325765625, 0.9999988967153318, 1.0000006856831174, 1.0000073002985543,
0.9999657153381558, 1.0001305856665574, 0.9995741158195164]

[None, None, 0.18622880465025216, 0.2549285792730075, 0.29332513315914027,
0.3109250915038273, 0.31817036093234596, 0.3209839938837237, 0.3220399408746392,
0.3224275706269687, 0.3225676538892072, 0.32261767899140703, 0.32263537512866297,
0.3226415833819772, 0.322643723034641, 0.3226445691019919, 0.3226444757920675,
0.3226468562761352, 0.32263778611666905, 0.3226497936482151, 0.3226966093107255,
0.3223872342818242, 0.3236759086580976, 0.319145953043651]

#Punto fijo converge(-2):

[-2, -1.95536442581983, -1.947079031893054, -1.945491145001699, -1.9451849935553254,
-1.9451258980286645, -1.9451144884489895, -1.9451122855054839, -1.9451118601611224,
-1.9451117780354956, -1.9451117621786478, -1.9451117591170015, -1.9451117585258575,
-1.9451117584117192, -1.9451117583896813, -1.9451117583854263, -1.945111758384605]

[None, 0.04463557418017006, 0.008285393926775964, 0.0015878868913550281,
0.0003061514463735815, 5.909552666083506e-05, 1.1409579675047965e-05,
2.202943505613675e-06, 4.2534436150987176e-07, 8.212562674358992e-08,
1.585684783478314e-08, 3.0616462698418445e-09, 5.911440226213927e-10,
1.1413825440342862e-10, 2.2037927038809357e-11, 4.255040764178375e-12,
8.213429936176908e-13]

[None, 0.0228272405853224, 0.0042552941052015035, 0.0008161881874582583,
0.0001573893729326027, 3.0381337640266304e-05, 5.865762525961042e-06,
1.1325533862643756e-06, 2.1867347077645117e-07, 4.222154617074723e-08,
8.15215256167207e-09, 1.5740207499602504e-09, 3.039126261153257e-10,
5.867953546105144e-11, 1.1329902738881242e-11, 2.187555931341624e-12,
4.22260052707618e-13]

[None, None, 0.9810295684126165, 0.9963618486493313, 0.999298431169111,
0.9998645736470166, 0.999973853058881, 0.9999949516922586, 0.9999990246544223,
0.9999997971772744, 1.0000000108275886, 0.9999998097149294, 1.0000005126641403,
0.999997772696765, 1.0000091429544578, 1.000157017537695]

[None, None, 0.1749910769889204, 0.18833581250773582, 0.1919344508008999,
0.192815700744852, 0.1930209625862431, 0.19306731726263285, 0.1930775603827968,
0.1930797485805586, 0.1930804216014016, 0.19307972421242325, 0.1930823850308491,
0.193071143819525, 0.19312140798720173, 0.1938234382347125]

#Newton Raphson(2):

[2, 1.5827317865351138, 1.5485626876122218, 1.5481966263255678, 1.548196582706551,
1.5481965827065503]

[None, 0.4172682134648862, 0.034169098922891994, 0.00036606128665406246,
4.3619016842910696e-08, 6.661338147750939e-16]

```
[None, 0.26363798150434686, 0.022065040825424004, 0.00023644366641132573,  
2.8174081592827256e-08, 4.3026436191362847e-16]  
[None, None, 1.8127647769167732, 1.991735603912653, 1.9919361746319624]  
[None, None, 0.16662159817858158, 0.30490750643177794, 0.3053917964933347]
```

#Newton Rapshon(0.5):

```
[0.5, -0.07196497498101051, 0.1487129144359895, 0.1489244574458927,  
0.1489244774534454, 0.14892447745344556]  
[None, 0.5719649749810105, 0.220677889417, 0.00021154300990319785,  
2.000755269793153e-08, 1.6653345369377348e-16]  
[None, 7.947824273293163, 1.483918799210854, 0.0014204719193290046,  
1.3434697264045126e-07, 1.1182409805388275e-15]  
[None, None, 7.297588710103577, 1.33324206527038, 2.007773725013937]  
[None, None, 13.012983927951328, 0.001586087064670444, 0.477487438609874]
```

#Newton Raphson(-2):

```
[-2, -1.9455215749047685, -1.9451117828834088, -1.9451117583844084,  
-1.9451117583844082]  
[None, 0.054478425095231486, 0.0004097920213597117, 2.4499000383926273e-08,  
2.220446049250313e-16]  
[None, 0.02800196399667177, 0.00021067787721291845, 1.2595163377283222e-08,  
1.1415519132404995e-16]  
[None, None, 1.9887425738008269, 1.9043138985781163]  
[None, None, 0.13362494860955762, 0.06916569973518458]
```