

# Pregunta 5 - Dirigida 6

June 6, 2019

## 1 Pregunta 5

5. Grafique las 10 primeras iteraciones del punto fijo para las siguientes ecuaciones y valores iniciales. **Analice visualmente la convergencia.**

```
In [1]: #!/usr/bin/env python3
        # -*- coding: utf-8 -*-
```

```
import __metodos_No_Lineales as mnl
import numpy as np
import pandas as pd
```

toolNick se ha importado correctamente.

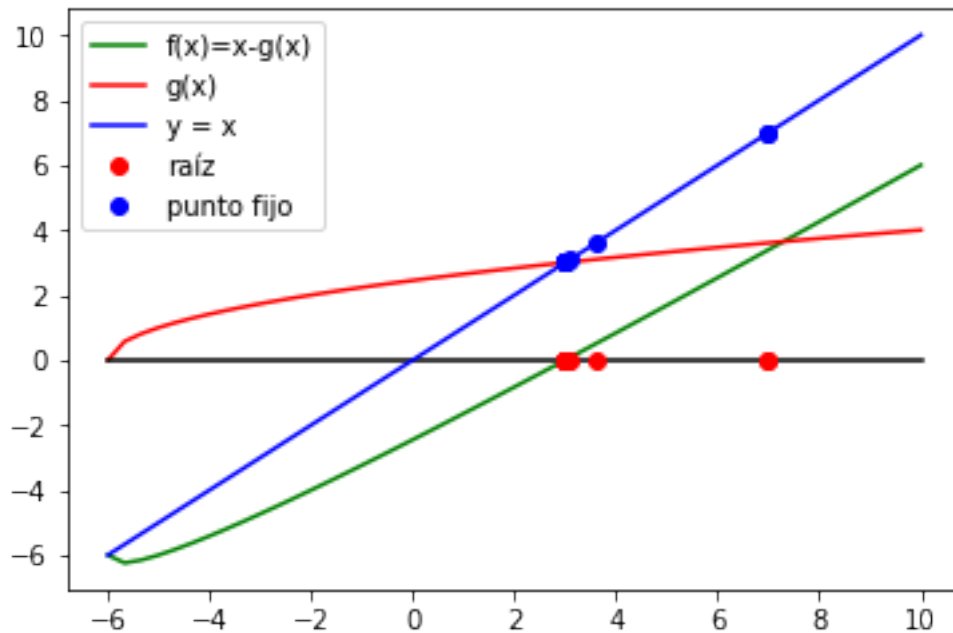
\_\_metodos\_No\_Lineales se ha importado correctamente.

```
In [6]: # Pregunta 5a)
        g = lambda x : np.sqrt(x+6)
        f = lambda x : x - np.sqrt(x+6)
        x_0 = 7
        x = mnl.solve_PuntoFijo(f, g, x_0, max_iter=10, tol=1e-6, v = True, graphic=True)
```

Data Frame - Metodo Punto Fijo

	x	g(x)
k_0	7.000000	3.605551
k_1	3.605551	3.099282
k_2	3.099282	3.016502
k_3	3.016502	3.002749
k_4	3.002749	3.000458
k_5	3.000458	3.000076
k_6	3.000076	3.000013
k_7	3.000013	3.000002
k_8	3.000002	3.000000
k_9	3.000000	3.000000

Converge en iter:9  
Resultado:3.0000000589156484



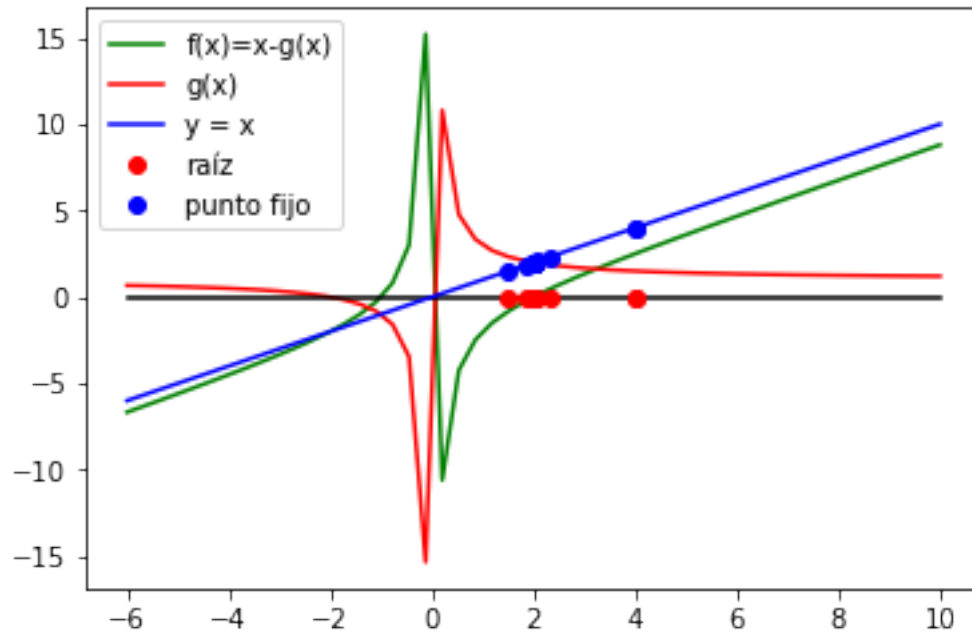
```
In [5]: # Pregunta 5b)
g = lambda x : 1 + (2/x)
f = lambda x : x - (1 + (2/x))
x_0 = 4
x = mnl.solve_PuntoFijo(f, g, x_0, max_iter=100, tol=1e-6, v = True, graphic=True)
```

Data Frame - Metodo Punto Fijo

	x	g(x)
k_0	4.000000	1.500000
k_1	1.500000	2.333333
k_2	2.333333	1.857143
k_3	1.857143	2.076923
k_4	2.076923	1.962963
k_5	1.962963	2.018868
k_6	2.018868	1.990654
k_7	1.990654	2.004695
k_8	2.004695	1.997658
k_9	1.997658	2.001172
k_10	2.001172	1.999414
k_11	1.999414	2.000293

k_12	2.000293	1.999854
k_13	1.999854	2.000073
k_14	2.000073	1.999963
k_15	1.999963	2.000018
k_16	2.000018	1.999991
k_17	1.999991	2.000005
k_18	2.000005	1.999998
k_19	1.999998	2.000001
k_20	2.000001	1.999999
k_21	1.999999	2.000000

Converge en iter:21  
 Resultado:2.0000002861023223



```
In [3]: # Pregunta 5c)
g = lambda x : x**3 / 2
f = lambda x : x - (x**3 / 2)
x_0 = 3.5
x = mn1.solve_PuntoFijo(f, g, x_0, tol=1e-6, v = True, graphic=True)
```

OverflowError

Traceback (most recent call last)

```

<ipython-input-3-f82a76b6f62c> in <module>()
      3 f = lambda x : x - (x**3 / 2)
      4 x_0 = 3.5
----> 5 x = mnl.solve_PuntoFijo(f, g, x_0, tol=1e-6, v = True, graphic=True)

~\Documents\UNI_Jupyter\Numerico_Jupyter\Library_metod\__metodos_No_Lineales.py in solve_PuntoFijo
    282             values.append(x_new)
    283             x = x_new
--> 284             x_new = g(x)
    285
    286             if v:

<ipython-input-3-f82a76b6f62c> in <lambda>(x)
      1 # Pregunta 5c)
----> 2 g = lambda x : x**3 / 2
      3 f = lambda x : x - (x**3 / 2)
      4 x_0 = 3.5
      5 x = mnl.solve_PuntoFijo(f, g, x_0, tol=1e-6, v = True, graphic=True)

OverflowError: (34, 'Result too large')

```

\_\_ El resultado se encuentra fuera de rango, por eso sale error \_\_

```

In [7]: # Pregunta 5d)
        g = lambda x : -x**2 + 2*x + 2
        f = lambda x : x - (-x**2 + 2*x + 2)
        x_0 = 2.5
        x = mnl.solve_PuntoFijo(f, g, x_0, tol=1e-6, v = True, graphic=True)

```

Max. iteraciones alcanzado.  
Resultado:2.678015883642905

```

In [4]: # Pregunta 5f)
        g = lambda x : np.cos(np.sin(x))
        f = lambda x : x - np.cos(np.sin(x))
        x_0 = 2.5
        x = mnl.solve_PuntoFijo(f, g, x_0, tol=1e-6, v = True, graphic=True)

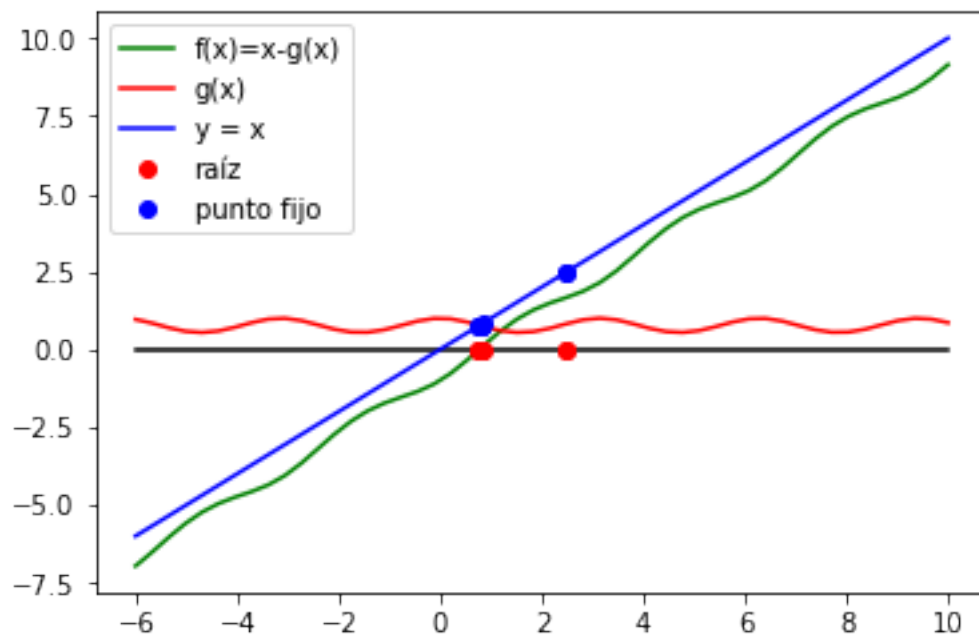
```

Data Frame - Metodo Punto Fijo

	x	g(x)
k_0	2.500000	0.826197
k_1	0.826197	0.741589

k_2	0.741589	0.780419
k_3	0.780419	0.762533
k_4	0.762533	0.770765
k_5	0.770765	0.766974
k_6	0.766974	0.768720
k_7	0.768720	0.767916
k_8	0.767916	0.768286
k_9	0.768286	0.768115
k_10	0.768115	0.768194
k_11	0.768194	0.768158
k_12	0.768158	0.768174
k_13	0.768174	0.768167
k_14	0.768167	0.768170
k_15	0.768170	0.768169
k_16	0.768169	0.768169

Converge en iter:16  
 Resultado:0.7681693925175375



```
In [15]: # Pregunta 5g)
g = lambda x : x**2 - np.sin(x+0.15)
f = lambda x : x - (x**2 - np.sin(x+0.15))
x_0 = 0
x = mn1.solve_PuntoFijo(f, g, x_0, tol=1e-6, v = True, graphic=True)
```

Max. iteraciones alcanzado.  
Resultado:-0.35132788484423716