Método de Punto Fijo

Al igual que otros métodos numéricos para encontrar las raíces de una ecuación, el método de punto fijo es un método iterativo: parte de una solución inicial, obtenida de alguna manera, y mediante una sucesión de valores calculados de forma recurrente, los cuales, con suerte, convergen hacia la raíz.

En análisis numérico, el método de punto fijo es un método que basado en el concepto de punto fijo, permite obtener una sucesión aplicando la fórmula siguiente:

$$x_{n+1} = f(x_n),\ n = 0, 1, 2, \ldots$$

donde f es una función definida en los números reales con valores reales y x_0 es un punto en el dominio de f.

Específicamente, en matemáticas, un punto fijo de una función es un valor que la función asocia a sí mismo. Formalmente, c es un punto fijo de una función f si c pertenece tanto al dominio como al codominio de f, y $f(c) = c$. Un punto fijo, también conocido como punto invariante, es un valor que no cambia bajo una transformación dada.

```
Por ejemplo, si $f$ es una función definida en los números reales como
$f(x)=x^{2}-3x+4$, entonces $2$ es un punto fijo de $f$, porque $f(2) = 2$.
```

No todas las funciones tienen puntos fijos: por ejemplo, $f(x) = x + 1$, no tiene puntos fijos, ya que x nunca es igual a $x + 1$ para ningún número real.

Para aplicar el método de punto fijo a una función $f(x)$, podemos definir una nueva función $g(x)$ tal que $g(x) = x - f(x)$. Si podemos demostrar que $g(x) \neq 0$ para todo valor de x, entonces no podemos utilizar este método.

En el ejemplo $f(x) = x + 1$, la función

$$g(x) = x - f(x) = x - (x + 1) = 1$$

con lo cual sabemos que $g(x) \neq 0$ para todo valor de x y, por tanto, nunc se cumple que $x = f(x)$ que es la condición de punto fijo.

Si tomamos la función $f(x) = x^2 - 3x + 4$, tendríamos que:

$$g(x) = f(x) - x = x^2 - 3x + 4 - x = x^2 - 4x + 4$$

utilizando la ecuación para obtener las raíces de una ecuación de segundo grado

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

como

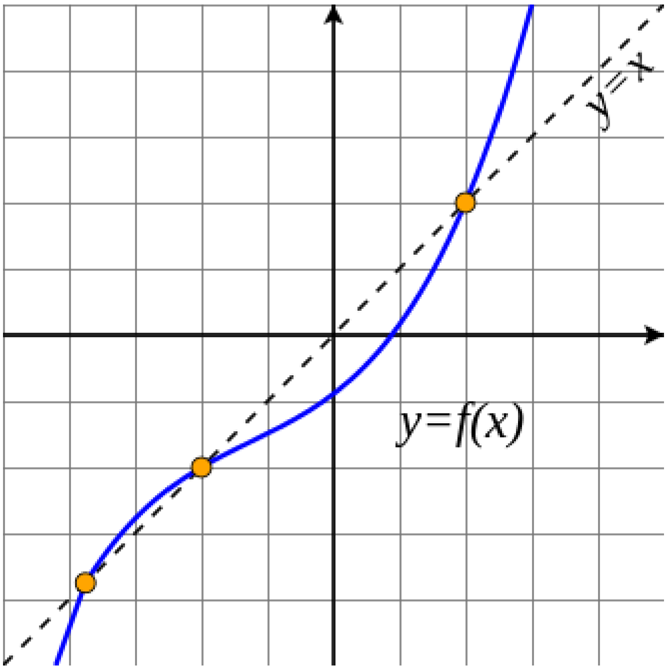
$$b^2 - 4ac = (-4)^2 - (4 \times 1 \times 4) = 16 - 16 = 0$$

Luego, si el término subradical se anula, estamos en presencia de una ecuación de segundo grado con raíz multiple

$$x = \frac{-(-4)}{2 \times 1} = 2$$

que es el punto fijo.

En términos gráficos, un punto fijo x significa que el punto $(x, f(x))$ forma parte de la recta $y = x$. Dicho en otras palabras, la gráfica de f tiene un punto en común con dicha recta.



La sucesión de punto fijo $x_0, x_1, x_2 \ldots$ se espera que converja a un punto x_{fix}. Si f es continua, entonces se puede probar que el punto x_{fix} obtenido es un punto fijo de f, es decir, $f(x_{fix}) = x_{fix}$. Más generalmente, la función f se puede definir en cualquier espacio métrico con valores en ese mismo espacio.

Algoritmo

Dada una ecuación $f(x)$, queremos encontrar un valor $x \mid f(x) = 0$, i.e., que sea raíz de la función f. Entonces, definimos $g(x) = f(x) - x$. Conviene comprobar que $g(x)$ tiene al menos un punto fijo.

Aplicando los criterios de convergencia:

- C1. El número de iteraciones n no debe sobrepasar el valor N.
- C2. la condición $|x_{i+1} - g(x_i)| < \epsilon$ (donde i es el número de iteración) y ϵ representa tolerancia del error al considerar x_{i+1} como punto fijo de g.

A partir de una suposición inicial x_0

While not(C1 and C2):

$x_{i+1} = g(x_i)$

$i++$

Ejemplo: se desea calcular una raíz de $f(x) = 0.5sin(x) - x + 1$. Por tanto, $g(x) = 0.5sin(x) + 1$

In [14]:

```
%matplotlib inline

import matplotlib.pyplot as plt
import numpy as np
import math
from math import sin
```

In [20]:

```
def g(x):
    return 0.5*sin(x)+1

def f(x):
    return 0.5*sin(x)-x+1
```

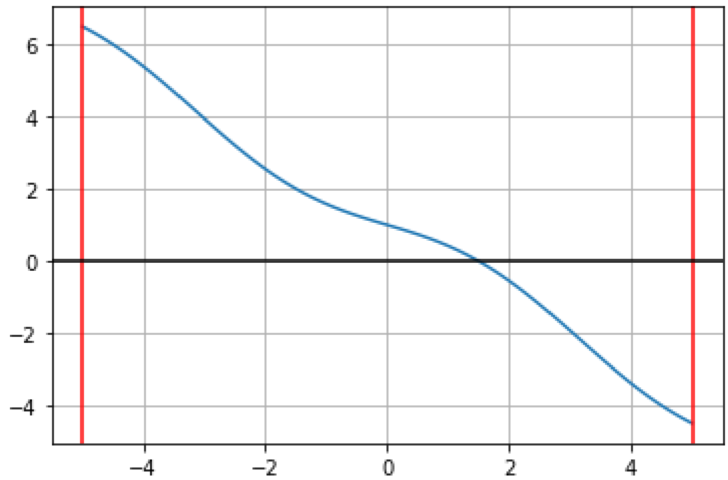
In [21]:

```
x_low = -5
x_high = 5

def graf0():
    x = np.linspace(x_low, x_high, num = 100)
    f_x = [f(x[i]) for i in range(len(x))]

    plt.plot(x, f_x)
    plt.grid()
    plt.axvline(x_low, color = 'r')
    plt.axvline(x_high, color = 'r')
    plt.axhline(color = 'k')

graf0()
plt.show()
```



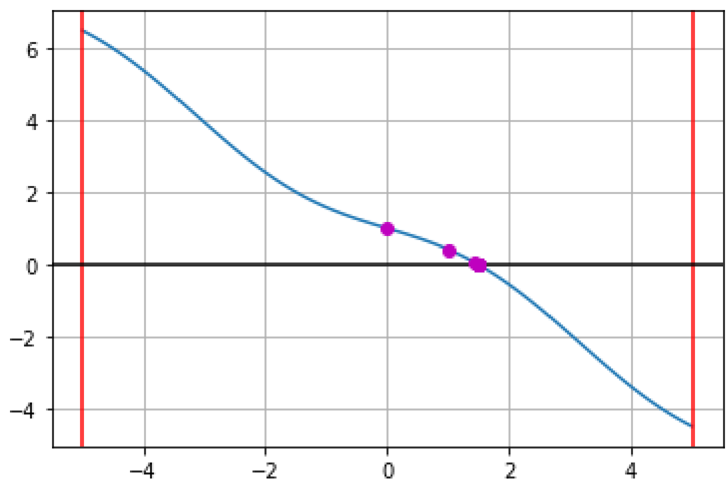
In [23]:

```
x0 = 0.0
n = 40
eps = 0.000000001
print('g(x0)=', g(x0))

def pto_fijo():
    global x0
    print('Iteración\tAproximación\tf(x)')
    for i in range(n):
        print('\t', i, '\t', x0, '\t', f(x0))
        x1 = g(x0)
        plt.plot(x1, f(x1), 'mo')
        if abs(x1-x0)<eps:
            return x1
        x0 = x1
    return x1

graf0()
plt.plot(x0, f(x0), 'mo')
pto_fijo()
plt.show()
```

```
g(x0)= 1.0
Iteración       Aproximación    f(x)
         0       0.0     1.0
         1       1.0     0.42073549240394825
         2       1.4207354924039484      0.07364550016037219
         3       1.4943809925643206      0.004159891834847995
         4       1.4985408843991685      0.00015447112272926056
         5       1.4986953555218978      5.569885144218745e-06
         6       1.498700925407042       2.006153981826131e-07
         7       1.49870112602244        7.2254506822133635e-09
         8       1.4987011332478908      2.602347226599022e-10
```



¿Y qué pasa si, en lugar de medir la diferencia entre dos puntos sucesivos de la sucesión, medimos qué tan cerca se encuentra $f(x)$ de cero?

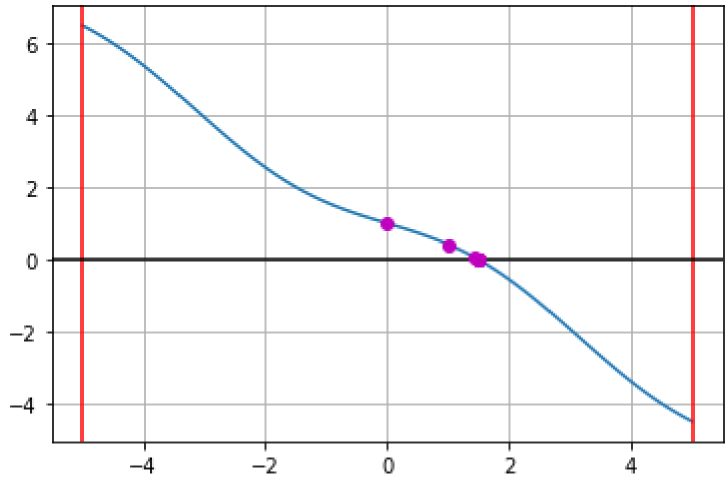
In [24]:

```
x0 = 0.0
n = 40
eps = 0.000000001
print('g(x0)=', g(x0))

def pto_fijo():
    global x0
    print('Iteración\tAproximación\tf(x)')
    for i in range(n):
        print('\t', i, '\t', x0, '\t', f(x0))
        x1 = g(x0)
        plt.plot(x1, f(x1), 'mo')
        if abs(f(x1))<eps:
            return x1
        x0 = x1
    return x1

graf0()
plt.plot(x0, f(x0), 'mo')
pto_fijo()
plt.show()
```

```
g(x0)= 1.0
Iteración        Aproximación     f(x)
         0        0.0      1.0
         1        1.0      0.42073549240394825
         2        1.4207354924039484       0.07364550016037219
         3        1.4943809925643206       0.004159891834847995
         4        1.4985408843991685       0.00015447112272926056
         5        1.4986953555218978       5.569885144218745e-06
         6        1.498700925407042        2.006153981826131e-07
         7        1.49870112602244         7.2254506822133635e-09
```



Como vemos, los reultados son equivalentes, teniendo en cuenta que si

$$abs(x1 - x0) < eps \implies abs(g(x0) - x0) < eps \implies$$

$$abs(f(x0) - x0 - x0) < eps \implies abs(f(x0)) < eps$$

ya que $x1 = g(x0) = f(x0) - x0.$

Bibliografía

- Fixed-point iteration (https://en.wikipedia.org/wiki/Fixed-point_iteration)
- Chapter 1 Numerical Methods for the Root Finding Problem (https://uomustansiriyah.edu.iq/media/lectures/9/9_2018_12_28!10_26_35_PM.pdf)
- FIXED POINT ITERATION METHOD (https://math.iitm.ac.in/public_html/sryedida/caimna/transcendental/iteration%20methods/fixed-point/iteration.html)
- Fixed point (mathematics) (https://en.wikipedia.org/wiki/Fixed_point_(mathematics))
- Métodos Numéricos (https://www.uv.mx/anmarin/slides/MetNum2.2.pdf)