

El método de Ridder

Material de apoyo Búsqueda de raíces

M. en C. Gustavo Contreras Mayén. `gux7avo@ciencias.unam.mx`

1. Definición.

El **método de Ridder** es una modificación inteligente del *método de la falsa posición*, veamos cómo se construye.

Suponiendo que la raíz está entre paréntesis (x_1, x_2) , primero calculamos $f_3 = f(x_3)$, donde x_3 es el punto medio del paréntesis, como se indica en la figura (1).

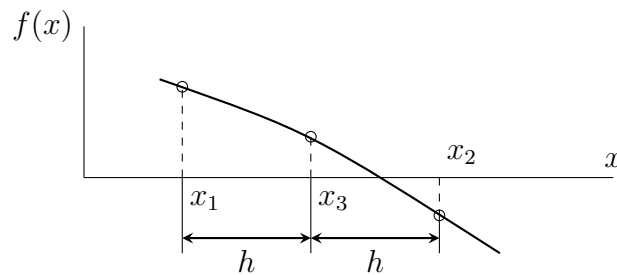


Figura 1: Construyendo el método de Ridder

A continuación introducimos la función:

$$g(x) = f(x) \exp((x - x_1) Q) \quad (1)$$

donde la constante Q se determina haciendo que los puntos (x_1, g_1) , (x_2, g_2) y (x_3, g_3) estén en línea recta, como se muestra en la figura ().

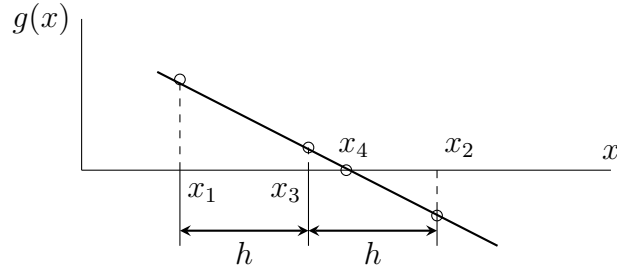


Figura 2: Construyendo el método de Ridder

Como antes, la notación que usamos es $g_i = g(x_i)$. El valor mejorado de la raíz se obtiene luego por interpolación lineal de $g(x)$ en lugar de $f(x)$.

Veamos ahora los detalles. De la ecuación. (1) obtenemos:

$$g_1 = f_1 \quad g_2 = f_2 \exp(2 h Q) \quad g_3 = f_3 \exp(h Q) \quad (2)$$

donde $h = (x_2 - x_1)/2$.

El requisito de que los tres puntos de la figura (2) estén sobre una línea recta es $g_3 = (g_1 + g_2)/2$, o:

$$f_3 \exp(h Q) = \frac{1}{2} \left[f_1 + f_2 \exp(2 h Q) \right]$$

que es una ecuación cuadrática en $\exp(h Q)$. La solución es:

$$\exp(h Q) = \frac{f_3 \pm \sqrt{f_3^2 - f_1 f_2}}{f_2} \quad (3)$$

La interpolación lineal basada en los puntos (x_1, g_1) y (x_3, g_3) ahora da como resultado la raíz mejorada:

$$\begin{aligned} x_4 &= x_3 - g_3 \frac{x_3 - x_1}{g_3 - g_1} = \\ &= x_3 - f_3 \exp(h Q) \frac{x_3 - x_1}{f_3 \exp(h Q) - f_1} \end{aligned}$$

donde en el último paso usamos las ecs. (2). Como paso final, sustituimos $\exp(h Q)$ de la ec. (3), y obtener después de un poco de álgebra:

$$x_4 = x_3 \pm (x_3 - x_1) \frac{f_3}{\sqrt{f_3^2 - f_1 f_2}} \quad (4)$$

Se puede demostrar que el resultado correcto se obtiene eligiendo el signo más si $f_1 - f_2 > 0$, y el signo menos si $f_1 - f_2 < 0$. Después del cálculo de x_4 , se determinan nuevos corchetes para la raíz y la ec. (4) se aplica de nuevo. El procedimiento se repite hasta que la diferencia entre dos valores sucesivos de x_4 se vuelve despreciable. La fórmula iterativa de Ridder en la ec. (4) tiene una propiedad muy útil: si x_1 y x_2 están a ambos lados de la raíz, entonces x_4 siempre está dentro del intervalo (x_1, x_2) . En otras palabras, una vez que la raíz está entre corchetes, permanece entre corchetes, lo que hace que el método sea muy confiable. La desventaja es que cada iteración requiere dos evaluaciones de función. Hay métodos competitivos que se las arreglan con una sola evaluación de función por iteración, pero son más complejos y requieren un elaborado procedimiento.

Se puede demostrar que el método de Ridder converge cuadráticamente, haciéndolo más rápido que el método de la secante o el de la falsa posición. Es el método a utilizar si la derivada de $f(x)$ es imposible o difícil de calcular.

2. Ejercicios.

2.1. Ejercicio 1.

Determina la raíz de:

$$f(x) = x^3 - 10x^2 + 5 = 0$$

que se encuentra en el intervalo $(0.6, 0.8)$ con el método de Ridder.

Solución: Para tener una idea de la función con la que estamos trabajando, hagamos un bosquejo que se presentamos en la gráfica (3):

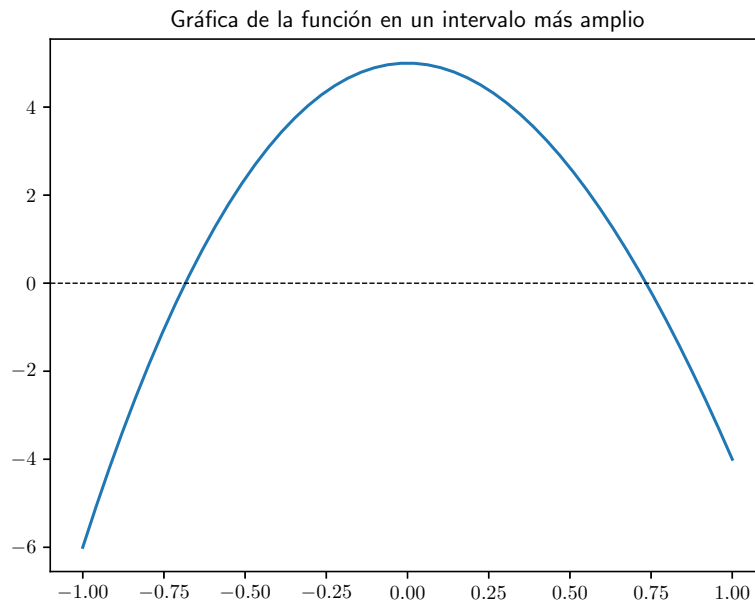


Figura 3: ¿Tenemos un caso de raíz de duplicidad 2?.

Como el problema nos pide que calculemos la raíz en el intervalo $[0.6, 0.8]$, así que si hacemos un acercamiento a ese intervalo como se muestra en la figura (4):

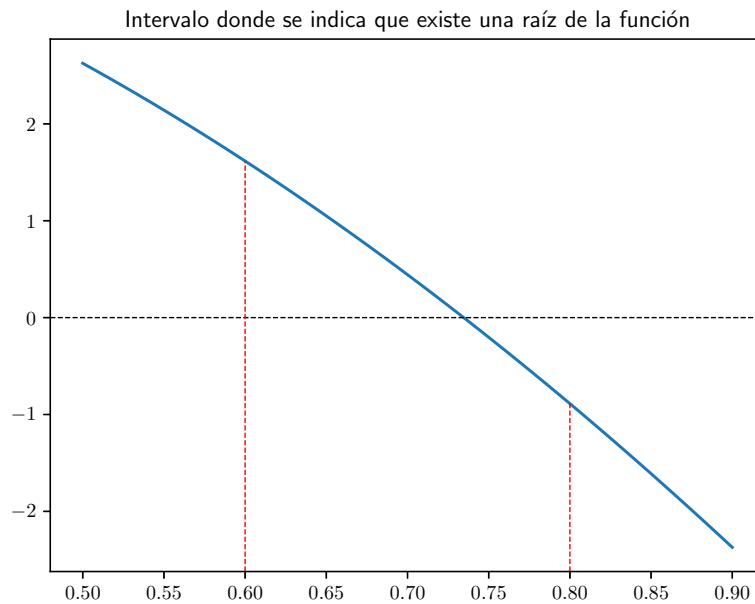


Figura 4: Acercamiento al intervalo de interés.

Procedemos a ocupar la función `ridder` del módulo `metodoRidder`:

```
from metodoRidder import ridder

x1 = 0.6
x2 = 0.8

raiz = ridder(f, x1, x2)
print('La raíz es: {0:1.4f}'.format(raiz))
```

El resultado que se muestra en la terminal como:

La raíz es: 0.7346

En la figura (5) se muestra la raíz de la función $f(x)$ en el intervalo $[0.6, 0.8]$.

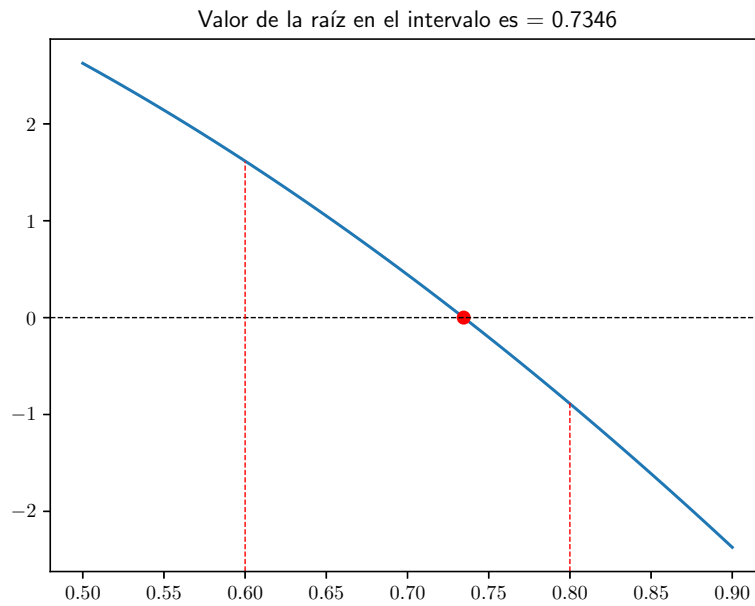


Figura 5: La raíz encontrada en el intervalo de interés.

2.2. Ejercicio 2.

Calcula la raíz de la función:

$$f(x) = \frac{1}{(x - 0.3)^2 + 0.01} - \frac{1}{(x - 0.8)^2 + 0.04}$$

En este ejercicio no se nos señala el intervalo en donde se encuentra la raíz, por lo que ocupamos nuestras habilidades de trabajo, para elaborar una gráfica de la función, que es la que se muestra como figura (6):

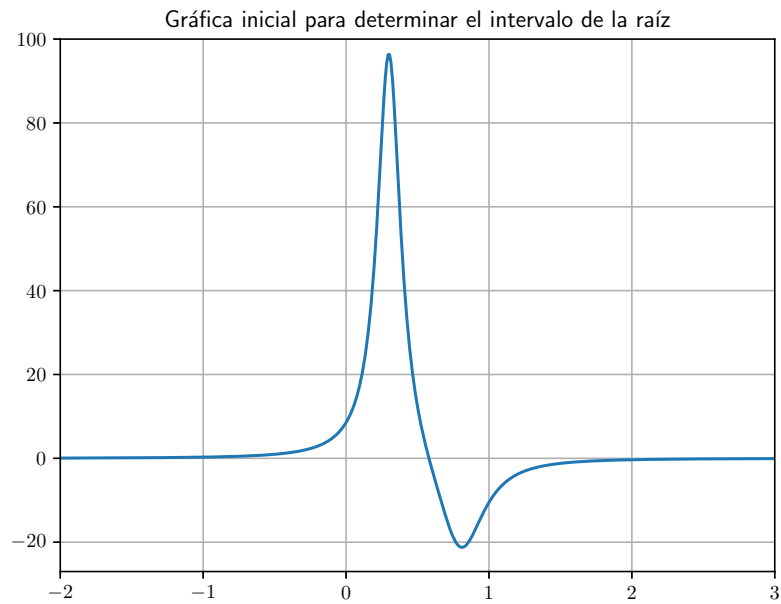


Figura 6: Gráfica de la función donde identificamos el intervalo de interés.

De la figura anterior, vemos que de manera aproximada la raíz se encuentra entre $x = 0$ y 1 , por lo que ocupamos la función `ridder` en el siguiente código:

```
from metodoRidder import ridder

def f(x):
    a = (x - 0.3)**2 + 0.01
    b = (x - 0.8)**2 + 0.04

    return 1./a - 1./b

raiz = ridder(f, 0., 1.)
print('La raíz es: {0:}'.format(raiz))
```

El resultado de la raíz se presenta con la siguiente gráfica:

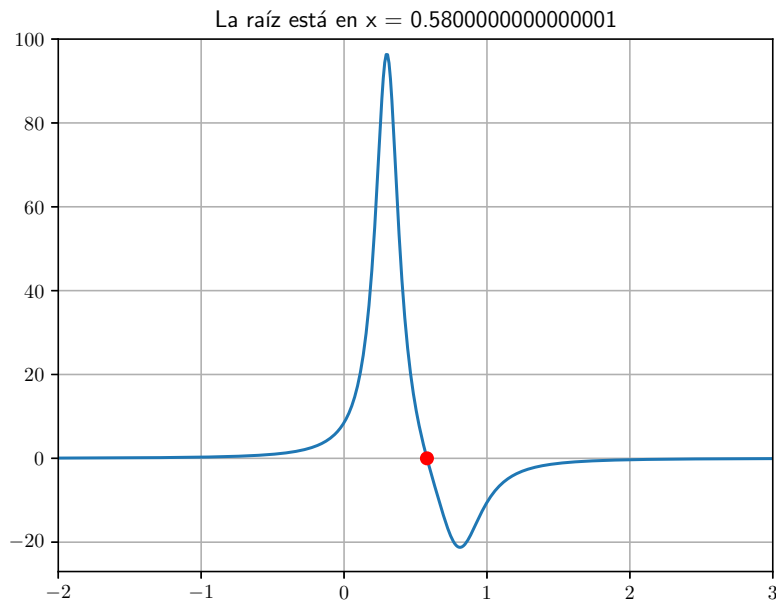


Figura 7: Gráfica con la raíz encontrada.

Como habrás notado falta incluir las rutinas que generan las gráficas, pero que sin problema alguno puedes implementar y contar con el material completo de cada ejercicio.

3. Resolviendo con funciones de python.

En el módulo `scipy.optimize` se dispone de la función `ridder`, que ocupa el mencionado método de Ridder para calcular una raíz de la función f entre los argumentos a y b . Los argumentos mínimos son:

`ridder(f, a, b)`

donde:

- `f`: Es una función continua y se requiere que $f(a)$ y $f(b)$ tengan signos contrarios.
- `a`: Intervalo inicial para calcular la raíz.
- `b`: Intervalo final para calcular la raíz.

La función `ridder` devuelve x_0 que es el valor de la raíz de la función `f` definida en el intervalo $[a, b]$.

En el siguiente ejemplo veremos la implementación de la función:

```
from scipy.optimize import ridder

def f(x):
    return (x**2 - 1)

raiz1 = ridder(f, 0., 2.)
raiz2 = ridder(f, -2., 0.)
print('La raíz es: {0:}'.format(raiz1))
print('\nLa raíz es: {0:}'.format(raiz1))
```

En la siguiente gráfica se muestra el resultado obtenido:

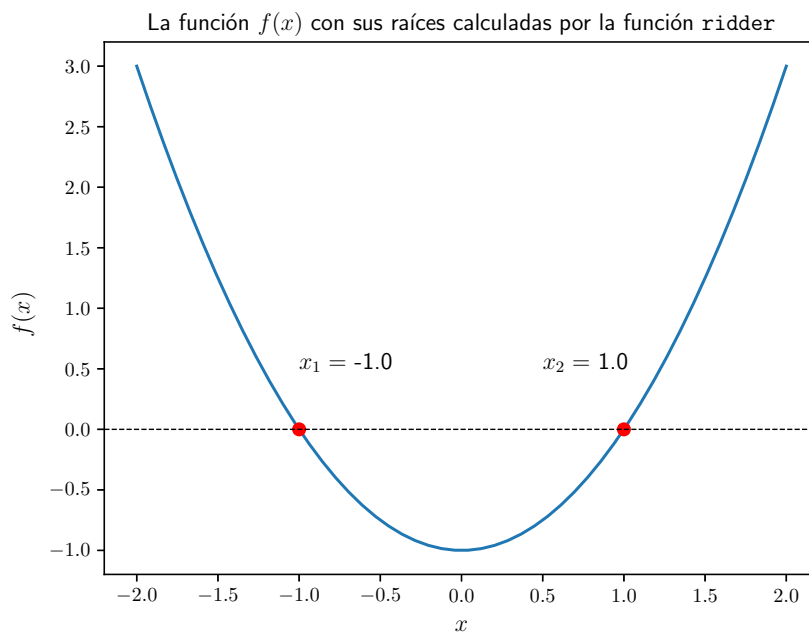


Figura 8: Gráfica con la raíz encontrada con la función `scipy.optimize.ridder`.