Método de las secantes

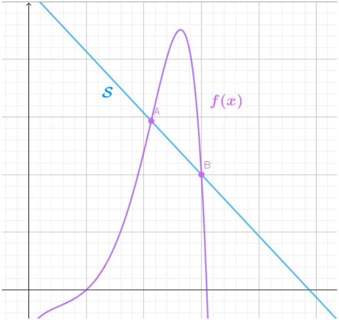
Dr. Carlos H.D. Alliera

Recta secante a una función f

Definición

Dada una función f *definida en cierto intervalo* $[a, b]$, *la recta que pasa por los puntos de la gráfica* $A = (a, f(a))$ *y* $B = (b, f(b))$ *es una recta secante a* f *y se define como:*

$$\mathcal{S}(x) = \frac{f(b) - f(a)}{b - a}(x - a) + f(a)$$



Recta secante definida por $\mathcal{S}(x)$ a la función f en los puntos $A = (a, f(a))$, $B = (b, f(b))$

Este método es como Newton-Raphson, pero usamos diferencias divididas backward en vez de la derivada. Esto implica que en este método x_{n+1} es función de x_n y de x_{n-1}.
La iteración viene dada por:

$$x_{n+1} = x_n - f(x_n) \cdot \frac{x_n - x_{n-1}}{f(x_n) - f(x_{n-1})} = \frac{x_{n-1} f(x_n) - x_n f(x_{n-1})}{f(x_n) - f(x_{n-1})}$$

Para este método necesitamos conocer x_0 y x_1.
Ventajas:

♡ Generalmente es más rápido que bisección y regula falsi, pero no tan rápido como Newton-Raphson.

♡♡ No necesitamos evaluar en f'.

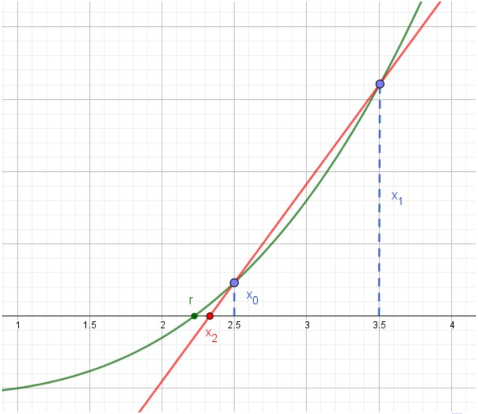
Desventajas:
★ No siempre converge.

Ejemplo

Emplear el método de la secante para hallar la raíz de $f(x) = x^3 - 11$

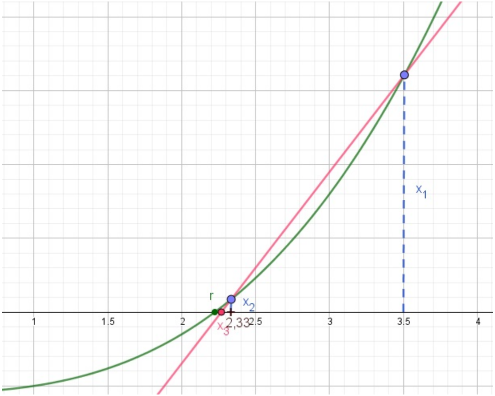
Este problema equivale a plantearse una aproximación de $\sqrt[3]{11}$.

$$x_0 = 2{,}5, \quad x_1 = 3,5, \quad x_2 = \frac{x_0 f(x_1) - x_1 f(x_0)}{f(x_1) - f(x_0)} = 2{,}33028$$



Ejemplo

$$x_1 = 3{,}5, \quad x_2 = 2{,}33028, \quad x_3 = \frac{x_1 f(x_2) - x_2 f(x_1)}{f(x_2) - f(x_1)} = 2{,}266265215$$



Calculando raíces con Python

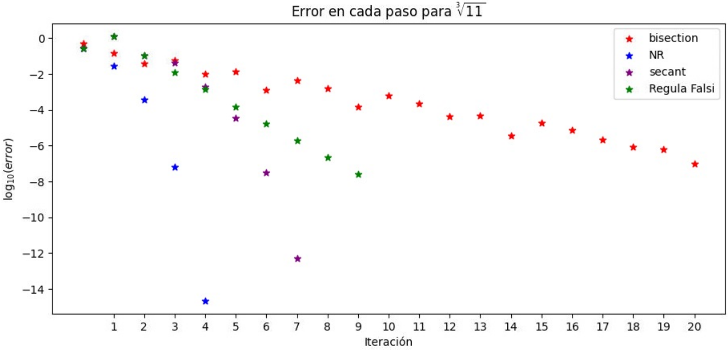
SciPy es una biblioteca open source de herramientas y algoritmos matemáticos para Python. SciPy contiene módulos para optimización, álgebra lineal, integración, interpolación, funciones especiales, FFT, procesamiento de señales y de imagen, resolución de ODEs y otras tareas para la ciencia e ingeniería.
Las funciónes que calculan raíces están incluídas dentro del módulo de optimización (scipy.optimize).
Si solo queremos usar las funciones de un módulo, pordemos cargar únicamente ese módulo.
import scipy.optimize as op.

Comparando errores para hallar una raíz cúbica

Vamos a usar todos los métodos que vimos para calcular $\sqrt[3]{11}$ y en graficamos el error para cada paso n de la iteración:
El error que vamos a calcular es la resta de $|p_n - r|$ donde la raiz r es la calculada con Python y p_n es el punto obtenido en el paso n con cada uno de los métodos.

```
r = 11**(1/3)
print('el valor calculado por python es : ', r)
#el valor calculado por python es :  2.2239800905693152
```

Gráfico para comparar errores



Para pensar

La función $f(x) = \tan(\pi x) - 6$ tiene una raíz en $r = \dfrac{\arctan(6)}{\pi} \approx 0,447431543$.
Sean $x_0 = 0,\ x_1 = 0,48$ use 10 iteraciones de los siguientes métodos para aproximar dicha raíz:

- Regula Falsi
- Bisección
- Secante

Diga cuál le parece más eficaz y porqué.