Tema 2 - Operaciones matemáticas básicas

Cálculo de raíces II

M. en C. Gustavo Contreras Mayén

24 de septiembre de 2014

Contenido

Contenido

Contenido

Contenido

Contenido

Método de Newton-Raphson

El método de Newton-Raphson es el algoritmo más conocido para encontrar raíces por una buena razón: es simple y rápido.

El único detalle es que utiliza la derivada $f'(x)$ así como la función $f(x)$. Por tanto, en los problemas a resolver con este algoritmo, deberá de contemplarse que la derivada sea fácil de calcularse.

El método de N-R se obtiene de la expansión en series de Taylor de $f(x)$ alrededor de x:

$$f(x_{i+1}) = f(x_i) + f'(x_i)(x_{i+1} - x_i) + O(x_{i+1} - x_i)^2$$

Si x_{i+1} es una raíz de $f(x) = 0$, tenemos que:

$$0 = f(x_i) + f'(x_i)(x_{i+1} - x_i) + O(x_{i+1} - x_i)^2$$

Suponiendo que x_i está cerca de x_{i+1}, podemos eliminar el último término de la ecuación y resolver para x_{i+1}, por lo que la fórmula de Newton-Raphson es:

$$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)}$$

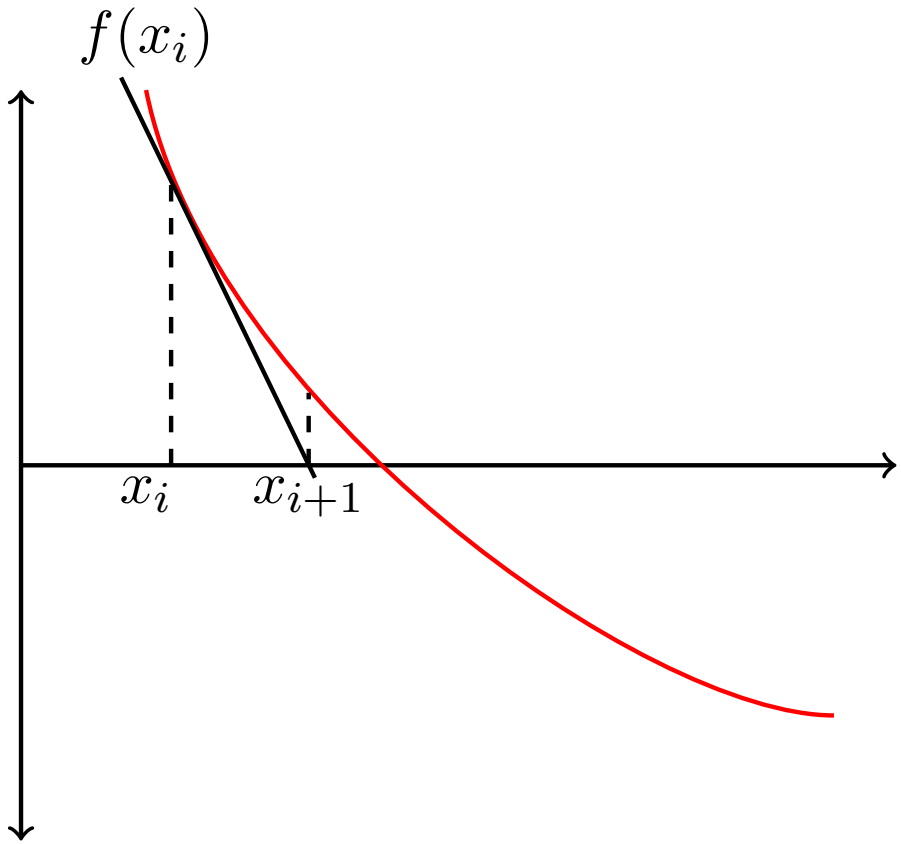
Si x representa el valor verdadero de la raíz, el error en x_i es $E_i = x - x_i$. Se puede demostrar que si x_{i+1} se calcula de la expresión de N-R, el error es:

$$E_{i+1} = -\frac{f''(x_i)}{2f'(x_i)}E_i^2$$

Lo que nos dice que el método de N-R converge de manera cuadrática, es decir, el error es el cuadrado del error del punto previo.

Representación gráfica

Podemos interpretar que x_{i+1} es el punto en donde la tangente de $f(x_i)$ cruza el eje de las x:



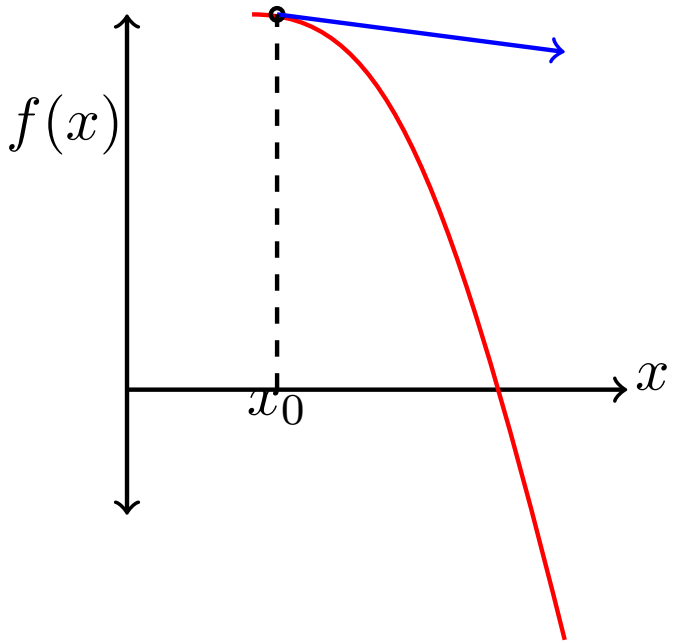
El método de N-R es sencillo: se aplica la expresión para x_{i+1} iniciando con un valor x_0, hasta alcanzar un criterio de convergencia:

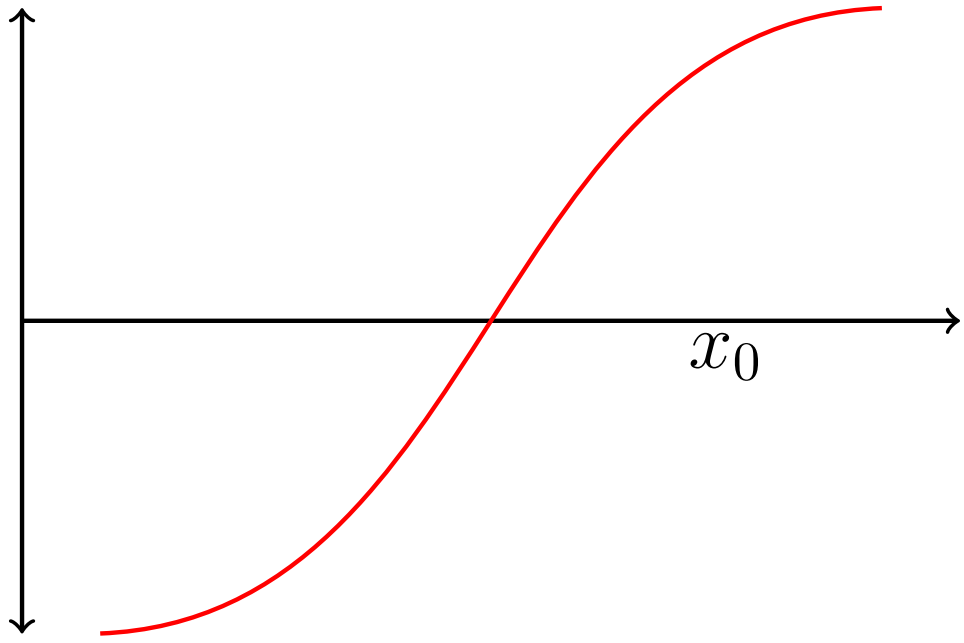
$$|x_{i+1} - x_i| < \epsilon$$

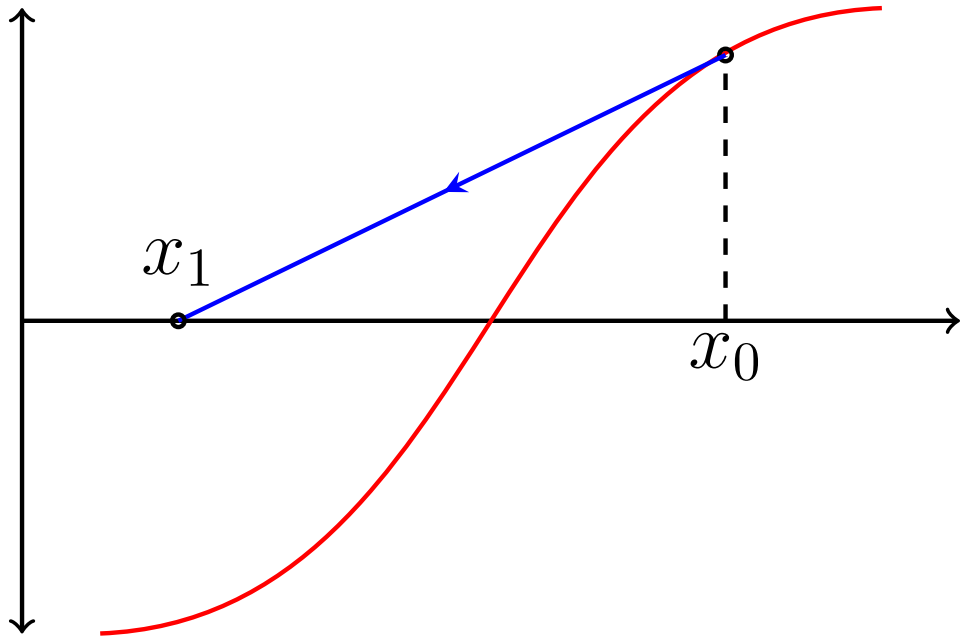
El algoritmo es el siguiente:

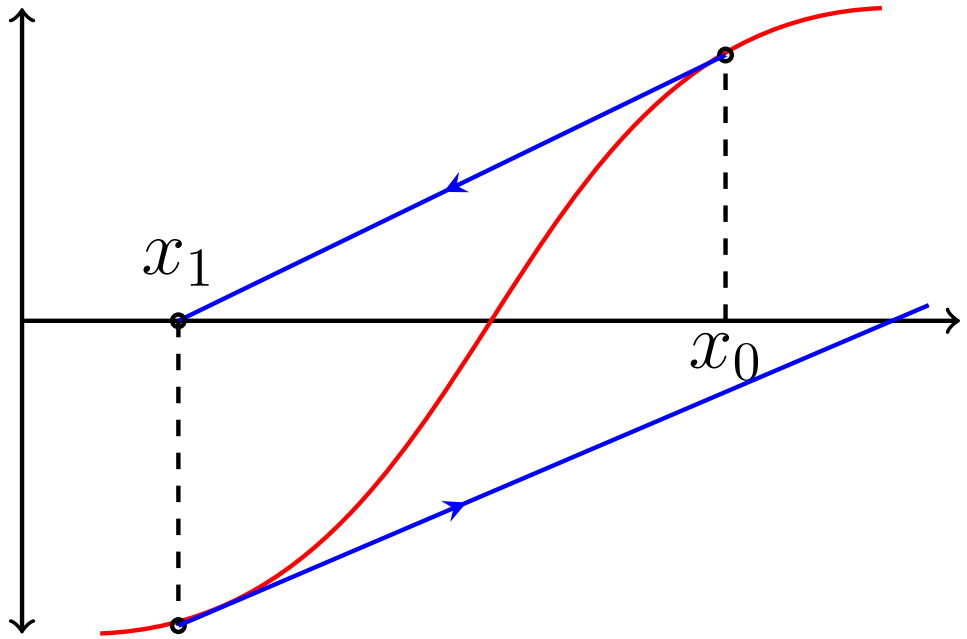
1. Sea x un valor inicial para la raíz de $f(x) = 0$.
2. Calcular $\Delta x = -f(x)/f'(x)$.
3. Asignar $x \leftarrow x + \Delta x$ y se repiten los pasos 2-3, hasta alcanzar $|\Delta x| < \epsilon$.

Aunque el método de Newton-Raphson converge rápidamente cerca de la raíz, sus características globales de convergencia son pobres. La razón es que la línea tangente no es siempre una aproximación aceptable de la función.


$f(x)$
x
x_0


x_0


x_1
x_0


x_1
x_0

Algoritmo para el método Newton-Raphson

La siguiente algoritmo para el método de Newton-Raphson supone que la raíz a calcularse inicialmente está en el intervalo (a, b).

El punto medio del intervalo se utiliza como aproximación inicial de la raíz. Los extremos del intervalo se actualizan luego de cada iteración. Si una iteración del método Newton-Raphson no se mantiene dentro del intervalo, se descarta y se reemplaza con el método de bisección.

Ya que el método `newtonRaphson` utiliza la función $f(x)$, así como su derivada, (denotadas por f y df) deben ser proporcionadas por el usuario.

Algoritmo de Newton-Raphson, versión larga

```
1 def newtonRaphson(f,df,a,b,tol=1.0e-9):
2     fa = (f(a)
3     if fa == 0.0: return a
4     fb = f(b)
5     if f(b) == 0.0: return b
6     if fa*fb > 0.0: print 'La raiz no esta en
          el intervalo'
7     x = 0.5 * (a + b)
```

```
1        for i in range(30):
2            fx = f(x)
3            if abs(fx) < tol: return x
4
5            if fa*fx < 0.0:
6                b = x
7            else:
8                a = x; fa = fx
```

```
1           dfx = df(x)
2
3           try: dx = -fx/dfx
4           except ZeroDivisionError: dx = b - a
5           x =  x + dx
6
7           if(b - x)*(x - a) < 0.0:
8               dx = 0.5*(b-a)
9               x = a + dx
10
11          if abs(dx) < tol*max(abs(b),1.0):
                return x
12
13      print 'Son demasiadas iteraciones'
```

Ejercicio

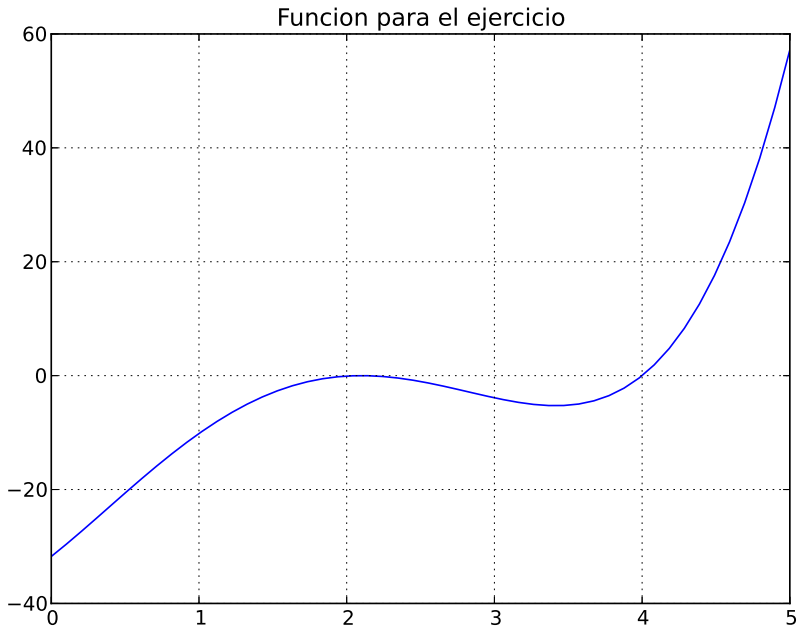
Encontrar la raíz positiva más pequeña de

$$f(x) = x^4 - 6.4x^3 + 6.45x^2 + 20.538x - 31.752$$

Ejercicio

Encontrar la raíz positiva más pequeña de

$$f(x) = x^4 - 6.4x^3 + 6.45x^2 + 20.538x - 31.752$$



Versión corta del algoritmo

Revisa que esta versión del algoritmo, sólo requiere de dos argumentos: x la aproximación inicial y tol la tolerancia de corte.

```
1  def f(x): return x**4 - 6.4*x**3 + 6.45*x**2 +
       20.538*x - 31.752
2  def df(x): return 4.0*x**3 - 19.2*x**2 + 12.9*
     x + 20.538
3
4  def newtonRaphson(x,tol=1e-05):
5      for i in range(30):
6          dx = -f(x)/df(x)
7          x = x + dx
8          if abs(dx) < tol: return x,i
9      print 'Son demasiadas iteraciones\n'
10
11 raiz,numIter = newtonRaphson(2.0)
12
```

Ejercicios a cuenta.

1. Encuentra las raíces de $x \sin x + 3 \cos x - x = 0$ en el intervalo $(-6, 6)$
2. Calcula todas las raíces reales de $x^4 + 0.9x^3 - 2.3x^2 + 3.6x - 25.2 = 0$.
3. Calcula todas las raíces reales de $x^4 + 2x^3 - 7x^2 + 3 = 0$.
4. Calcula todas las raíces de $\sin x - 0.1x = 0$.

Metodología a seguir

Para la solución de cada uno de los ejercicios, un proceso de solución podría ser el siguiente:

1. Graficar la función para conocer el comportamiento. Si conocemos el intervalo de solución, podremos acotar la gráfica, pero en caso contrario, debemos explorar para revisar y considerar un intervalo de operación más adecuado.
2. Conocer los intervalos donde hay una raíz. El método de incrementos sucesivos nos permite conocer un intervalo donde se encuentra una raíz, debemos de ajustar el código para que nos devuelva el(los) intervalo(s) donde hay una raíz, para ocupar éstos intervalos posteriomente en el algoritmo de Newton-Raphson.

Metodología a seguir

1. Iterar el algoritmo en cada intervalo donde hay una raíz. Como ya conocemos los intervalos con raíz, nuestro siguiente paso es usar el algoritmo para que de manera automática, nos devuelva el valor de la raíz, básicamente hay que recorrer el objeto en donde almacenamos los pares de puntos del intervalo con raíz.
2. Graficar la raíz. Como ya conocemos el valor de la raíz, podemos agregarlo a la gráfica de la función y obtener de manera autómatica la representación de la raíz.

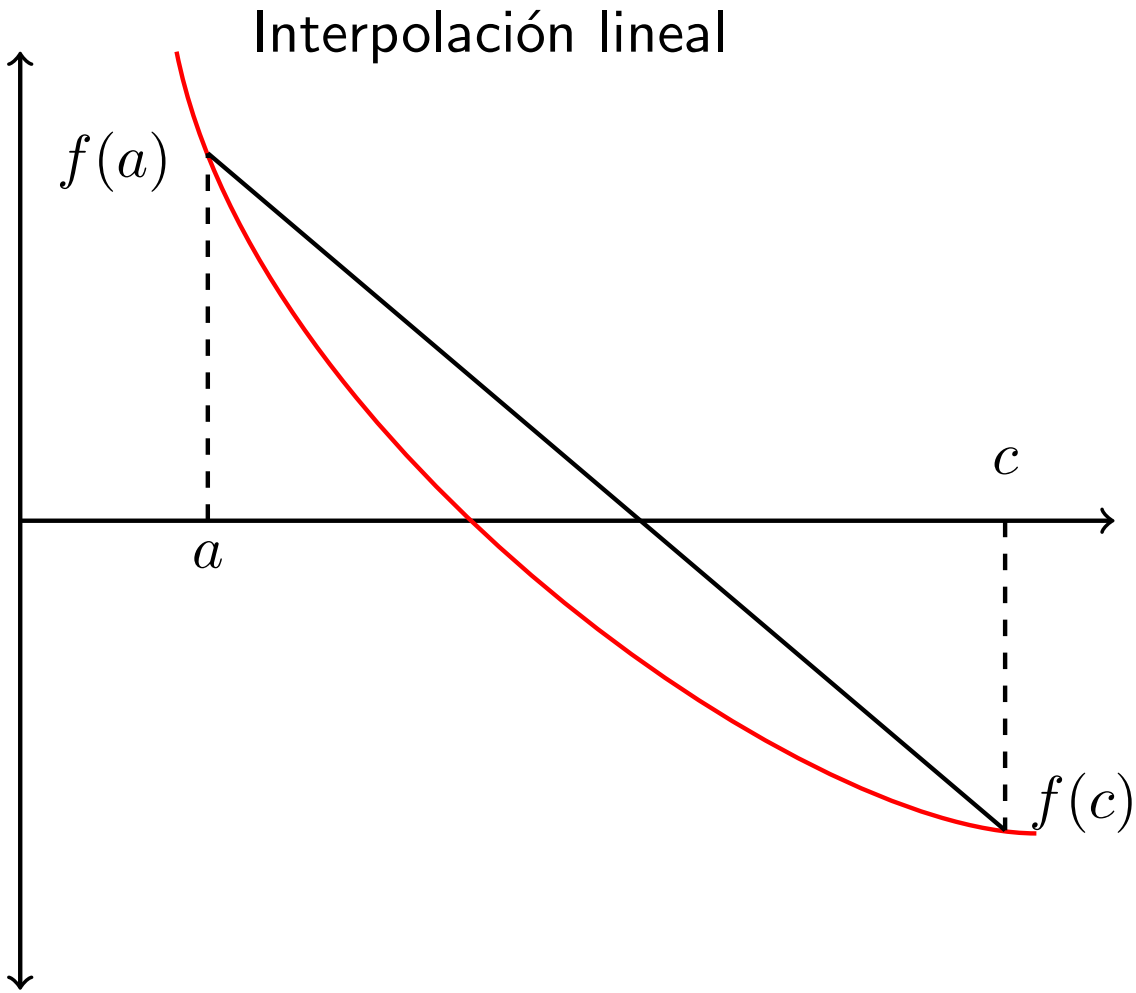
Contenido

Método de la falsa posición

Este método es parecido al de bisección, ya que el intervalo que contiene a la raíz se va reduciendo.

En vez de bisectar de manera monótona el intervalo, se utiliza una interpolación lineal ajustada a dos puntos extremos para encontrar la aproximación de la raíz.

Método de la falsa posición



La función está bien aproximada por la interpolación lineal, con lo que las raíces tendrán una buena precisión; la iteración convergerá más rápido que como ocurre con el método de bisección.

Dado un intervalo $[a, c]$ que contenga a la raíz, la función lineal que pasa por $(a, f(a))$ y $(c, f(c))$ se escribe como:

$$y = f(a) + \frac{f(c) - f(a)}{c - a}(x - a)$$

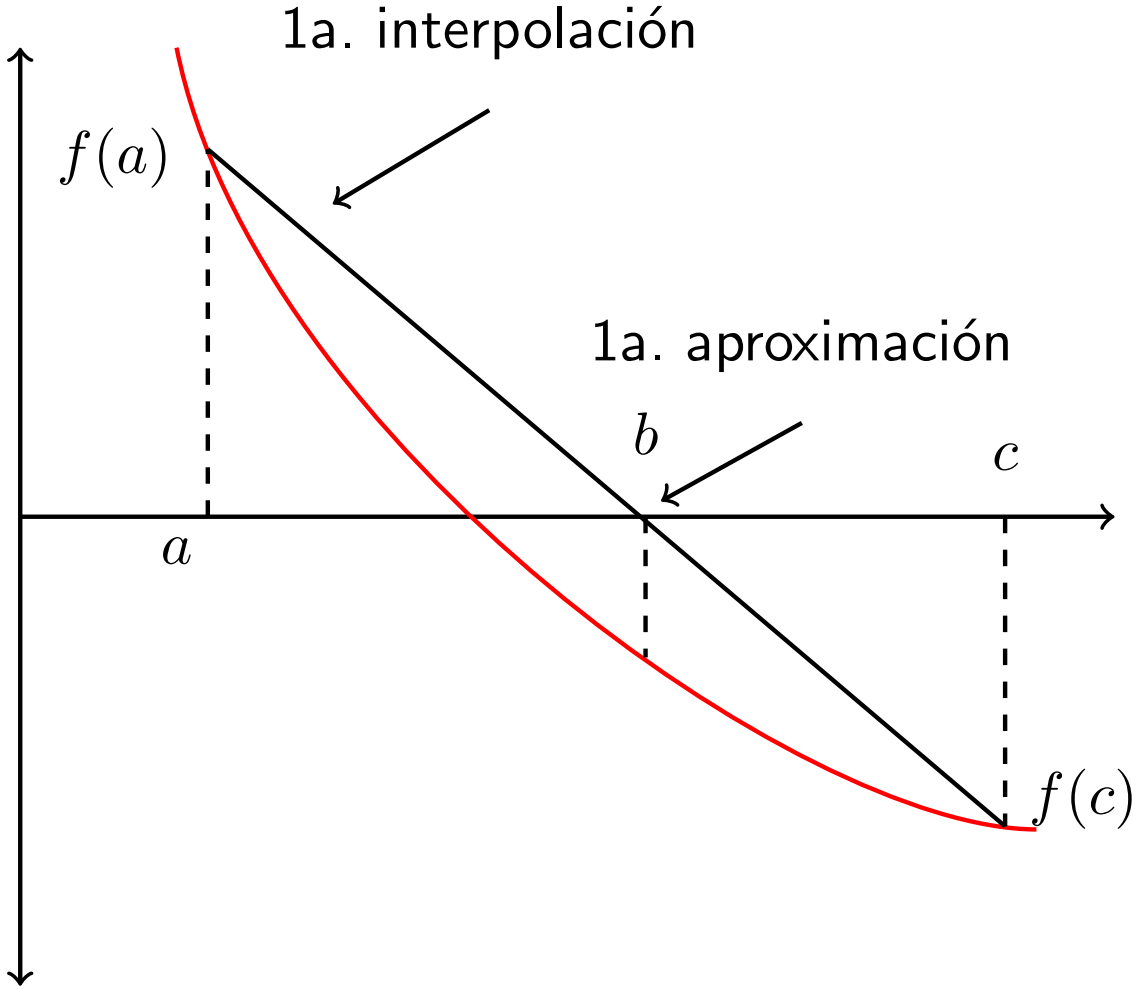
de donde se despeja x:

$$x = a + \frac{c - a}{f(c) - f(a)}(y - f(a))$$

La coordenada x en donde la línea intersecta al eje x se determina al hacer $y = 0$ en la ecuación anterior, por tanto:

$$b = a - \frac{c - a}{f(c) - f(a)} f(a) = \frac{af(c) - cf(a)}{f(c) - f(a)}$$

Método de la falsa posición

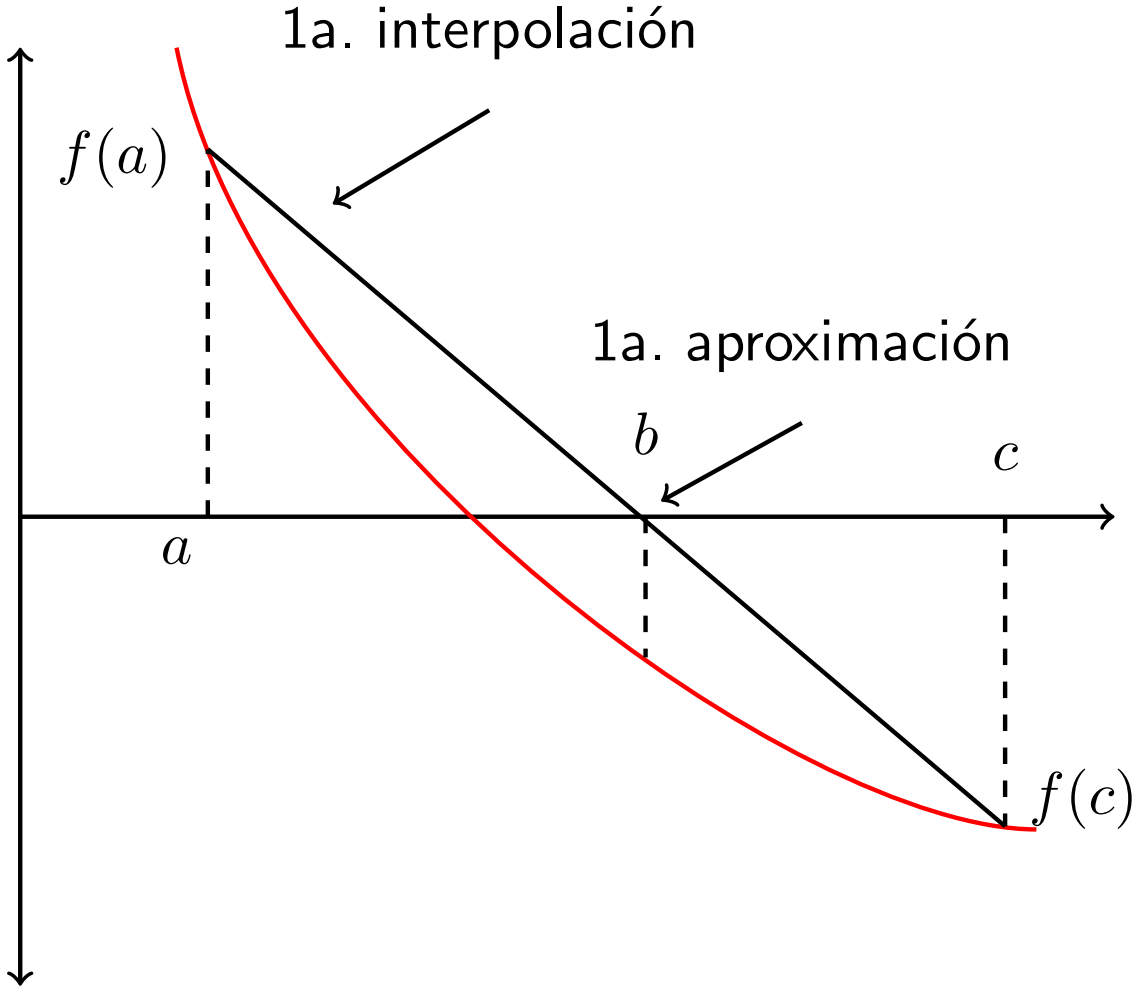


Después de encontrar b, el intervalo $[a, c]$ se divide en $[a, b]$ y $[b, c]$.

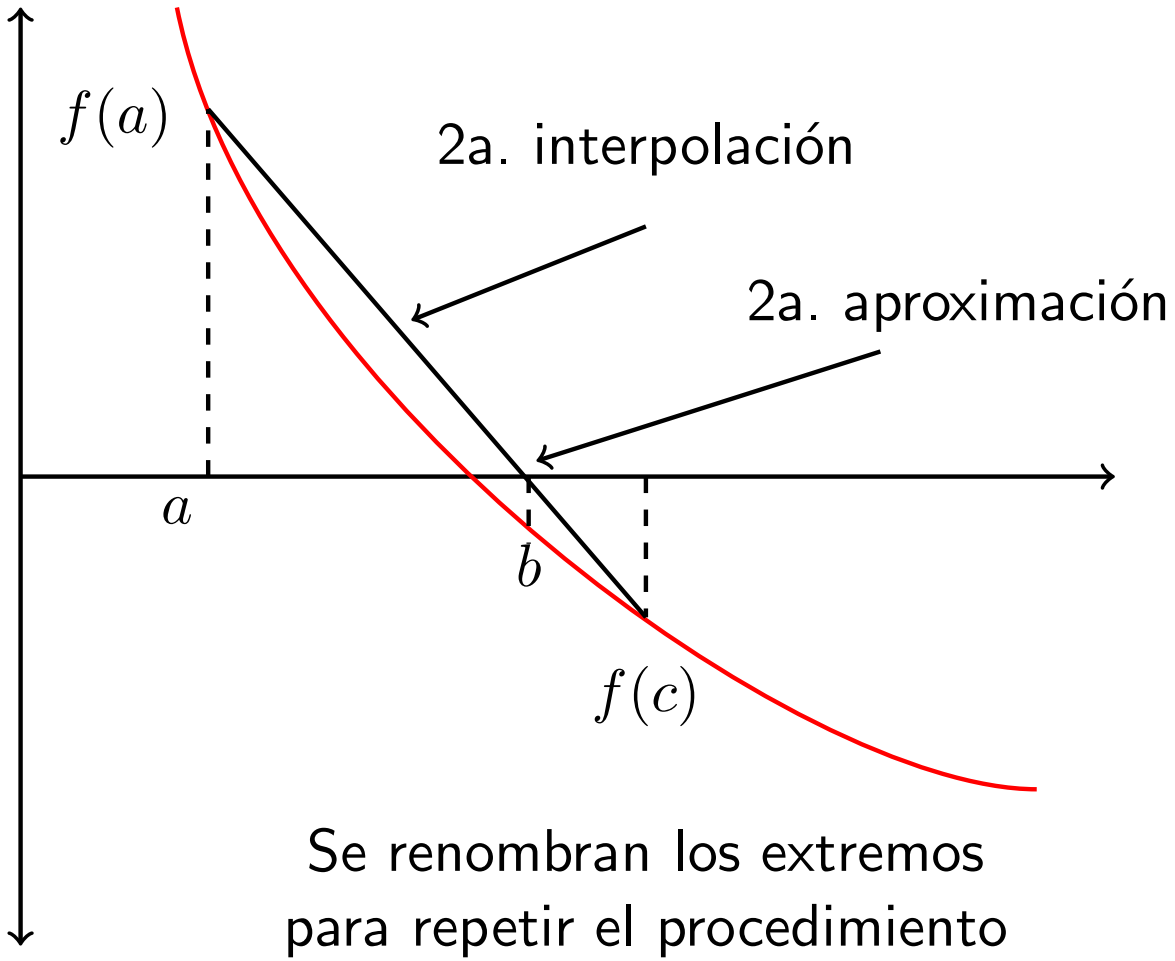
Si $f(a)f(b) \leq 0$, la raíz se encuentra en $[a, b]$; en caso contrario, está en $[b, c]$. Los extremos del nuevo intervalo que contiene a la raíz se renombran para el siguiente paso como a y c.

El procedimiento de interpolación se repite hasta que las raíces estimadas convergen.

Método de la falsa posición



Método de la falsa posición



Se renombran los extremos
para repetir el procedimiento

Consideraciones importantes del método.

La longitud de los nuevos intervalos, para el método de falsa posición, no decrece en cada nueva iteración como en el método de bisección, es decir, no siempre se garantiza que el nuevo intervalo sea la mitad (o menor) del intervalo anterior.

Por esta razón, aunque el método de la falsa posición normalmente tiene una mejor convergencia que el método de bisección, no siempre será el caso.

La desventaja de este método es que aparecen extremos fijos: uno de los extremos de la sucesión de intervalos no se mueve del punto original, por lo que las aproximaciones a la raíz, denotadas por b_1, b_2, b_3, etc. convergen a la raíz exacta solamente por un lado.

Los extremos fijos no son deseables debido a que hacen más lenta la convergencia, en particular cuando el intervalo es grande o cuando la función se desvía de manera significativa de una línea recta en el intervalo.

¿Qué podemos hacer al respecto?

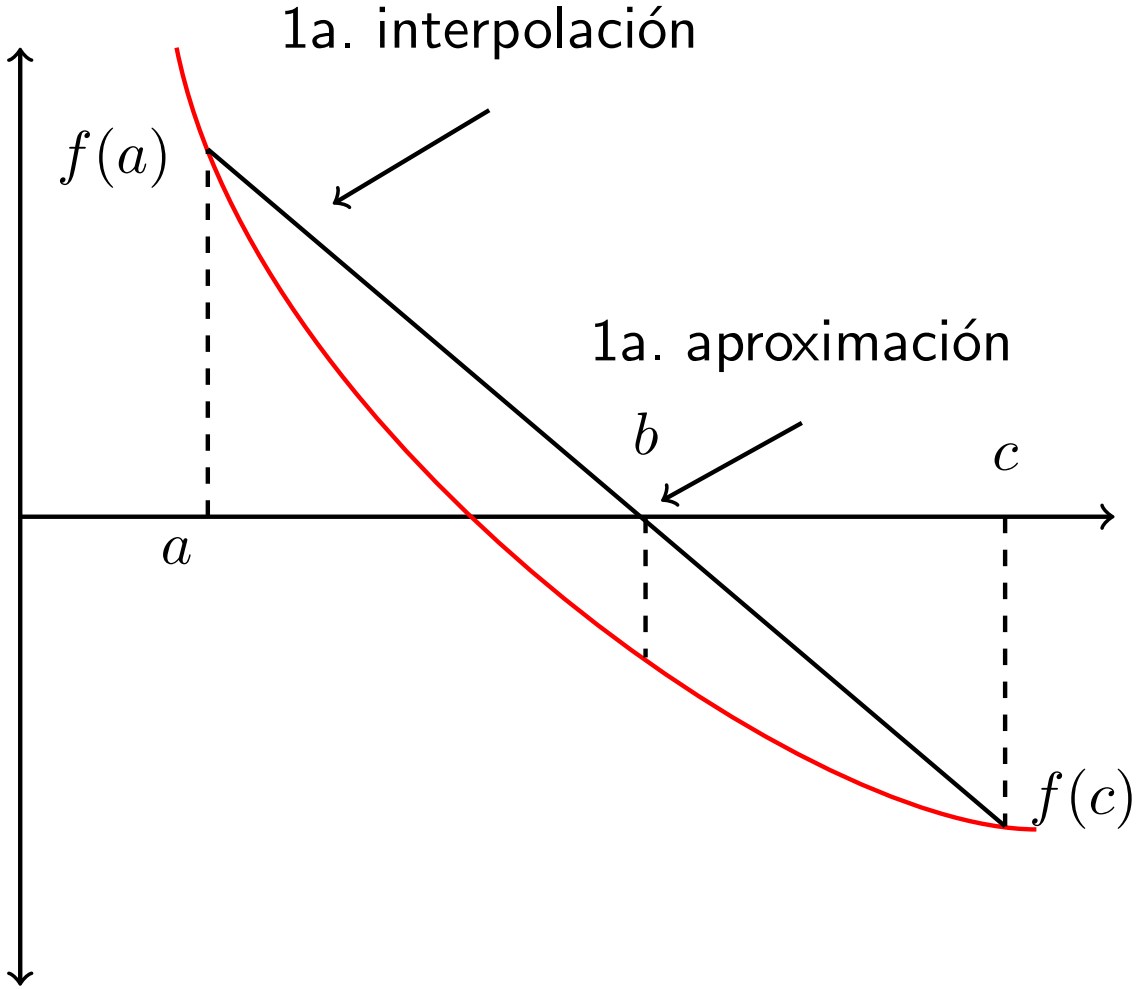
Contenido

Método de la falsa posición modificado

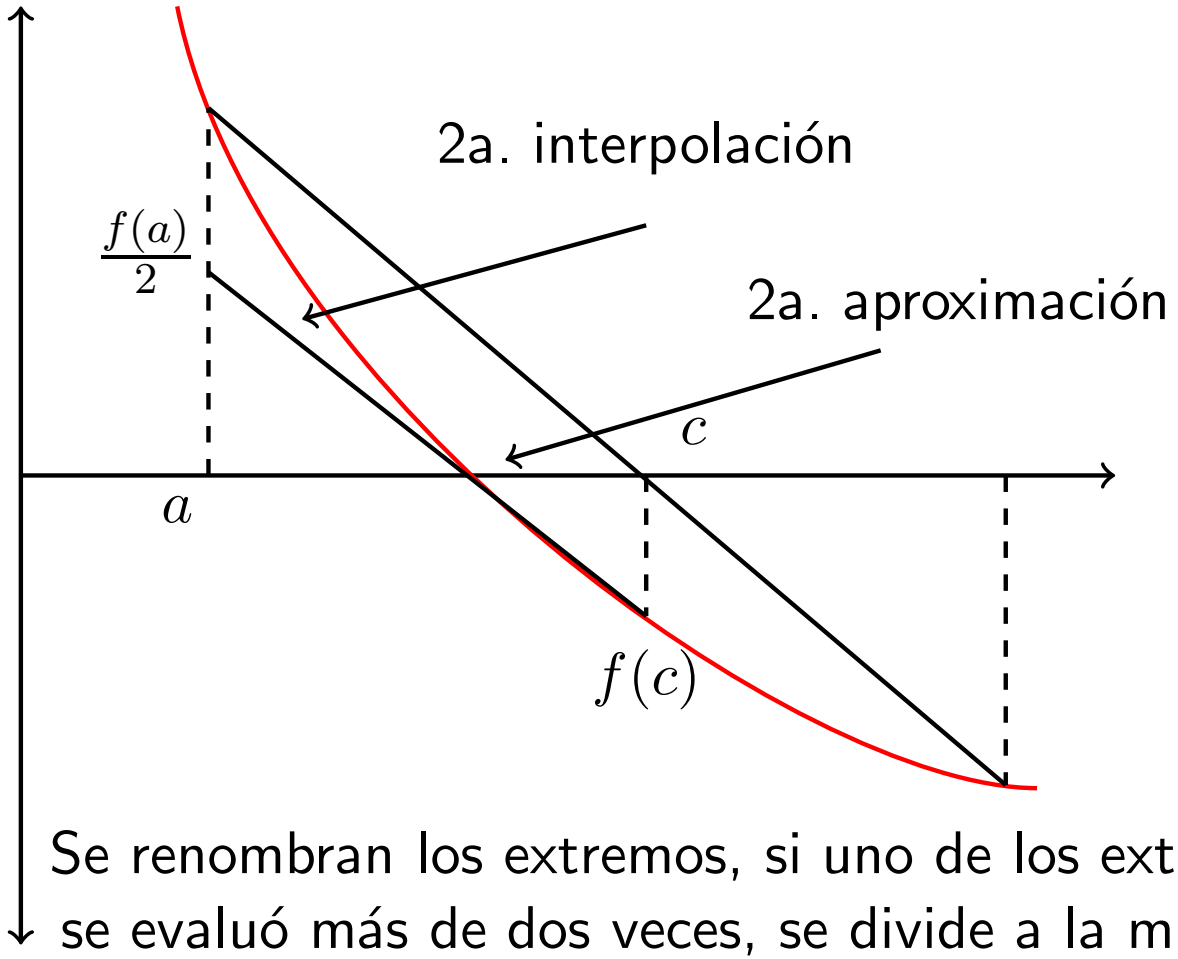
En este método, el valor de f en un punto fijo se divide a la mitad si este punto se ha repetido más de dos veces.

El extremo que se repite se llama extremo fijo. La excepción para esta regla es que para $i = 2$, el valor de f en un extremo se divide entre 2 de inmediato si no se mueve.

Método de falsa posición modificado



Método de la falsa posición modificado



Se renombran los extremos, si uno de los extremos ya se evaluó más de dos veces, se divide a la mitad y se realiza una nueva interpolación

Contenido

Método de la secante

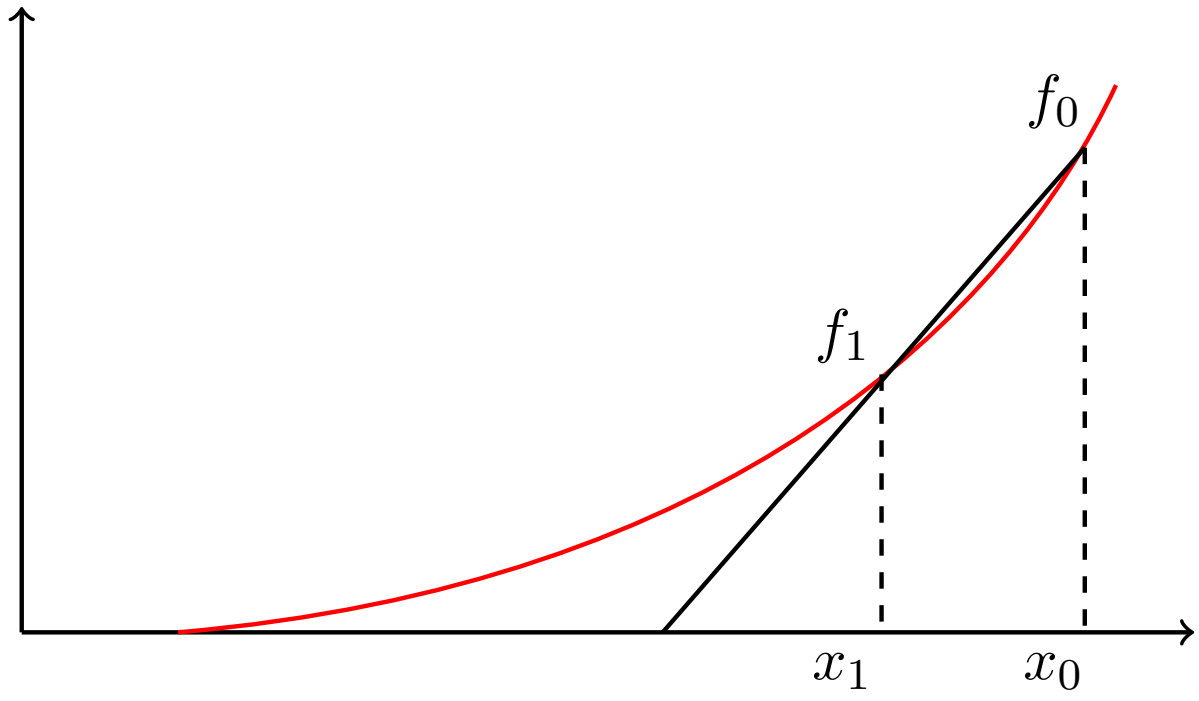
A diferencia del método de Newton, el valor de f' se aproxima utilizando dos valores de iteraciones consecutivas de f. Con lo que se elimina la necesidad de evaluar tanto a f como a f' en cada iteración.

Las aproximaciones sucesivas para la raíz en el método de la secante están dadas por:

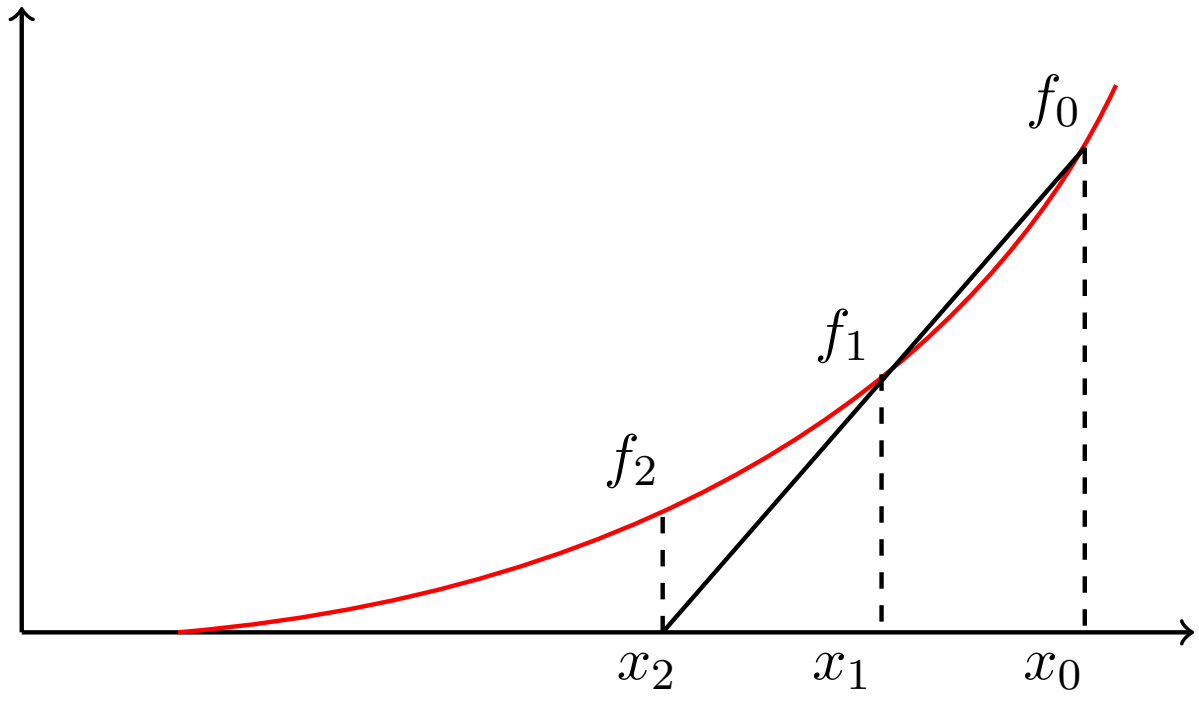
$$x_n = x_{n-1} - y_{n-1}\frac{x_{n-1} - x_{n-2}}{y_{n-1} - y_{n-2}}, \qquad n = 2, 3, \ldots$$

donde x_0 y x_1 son dos suposiciones iniciales para comenzar la iteración.

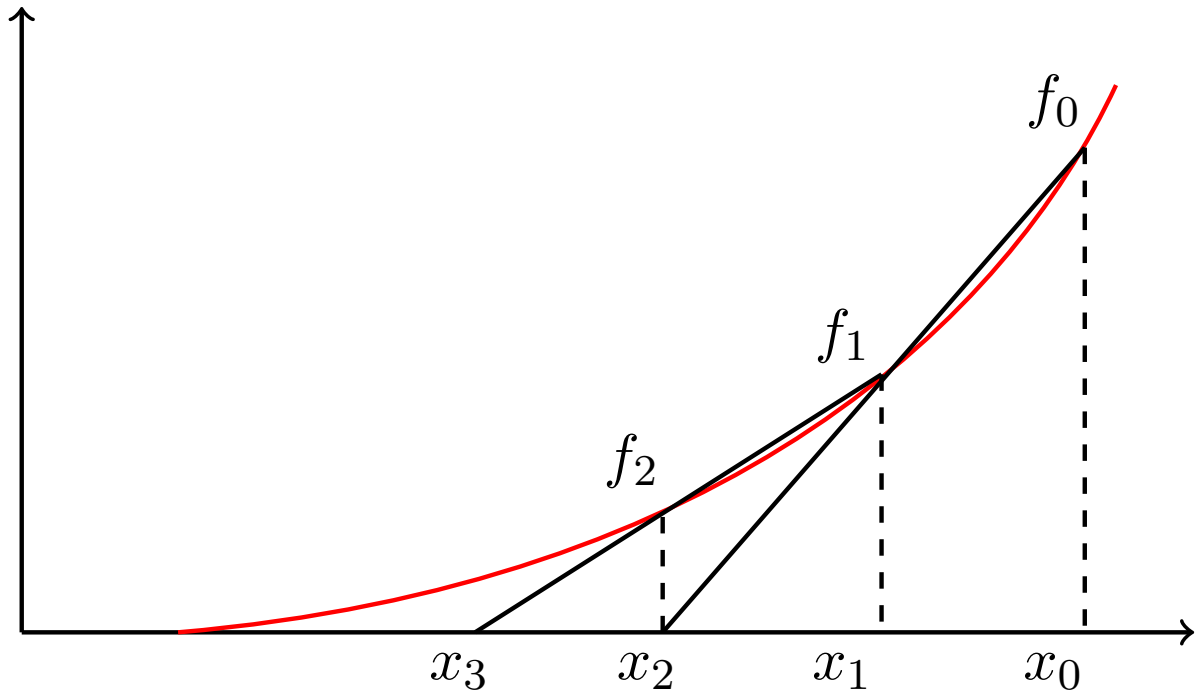
Método de la secante



Método de la secante



Método de la secante



Atento aviso

En las técnicas de falsa posición, falsa posición modificada y el método de la secante, no hemos presentado como tal un código en `Python` que nos devuelva las raíces, por lo que tendrás que proponer un código para cada una de las técnicas.

Ese código lo vas a utilizar par resolver los problmeas y ejercicios del examen.