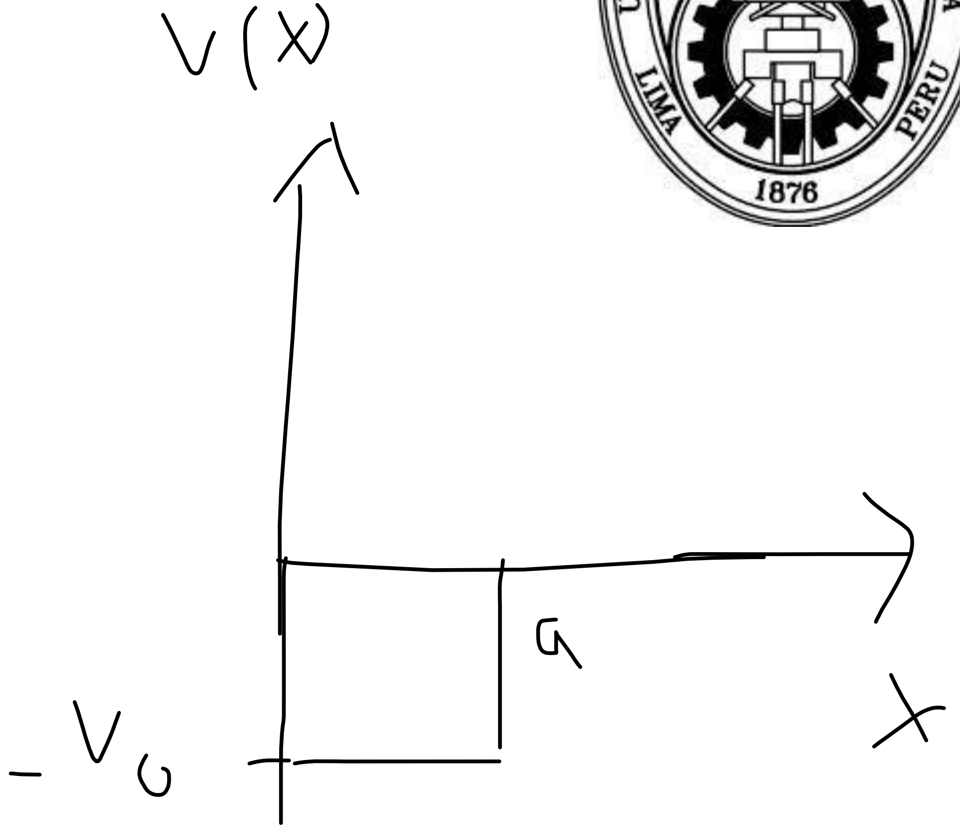
Solución de ecuaciones no lineales

PhD. Alejandro Paredes

Ecuación de Shrödinger 1D estacionaria

$$-\frac{\hbar^2}{2m}\frac{d^2u}{dx^2}+V(x)u(x)=Eu(x),$$

$$V(r)=\begin{cases}-V_0 & 0\leq x<a\\ 0 & x>a\end{cases}$$



U(x): función de onda

Consideramos sólo estados ligados para E< 0

Ecuación de Shrödinger 1D estacionaria

Condiciones de frontera:
u(x) y u'(x) se anulan en el infinito

Condiciones de continuidad:
u(x) y u'(x) son continuas en x=a

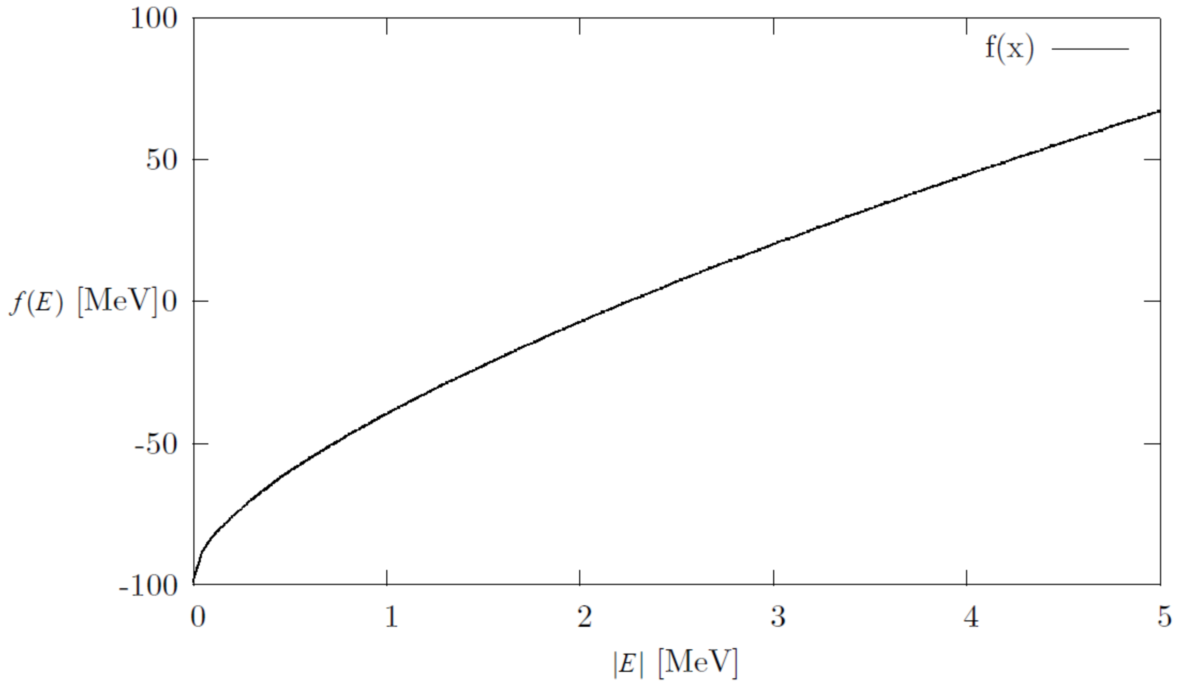
$$\frac{d^2u(x)}{dx^2} + \frac{2m}{\hbar^2}(V_0 + E)u(x) = 0 \qquad x < a, \qquad u(\mathrm{x}) = A sin(\sqrt{2m(V_0 - |E|)}\mathrm{x}/\hbar)$$

$$\frac{d^2u(x)}{dx^2} + \frac{2m}{\hbar^2}Eu(x) = 0 \qquad x > a. \qquad u(\mathrm{x}) = B\exp(-\sqrt{2m|E|}\mathrm{x}/\hbar)$$

$$\sqrt{2m(V_0 - |E|)}\cot(\sqrt{2ma^2(V_0 - |E|)}/\hbar) = -\sqrt{2m|E|}.$$

Ecuación de Shrödinger 1D estacionaria

$$f(E) = \sqrt{2m(V_0 - |E|)}\cot(\sqrt{2ma^2(V_0 - |E|)}/\hbar) + \sqrt{2m|E|}.$$

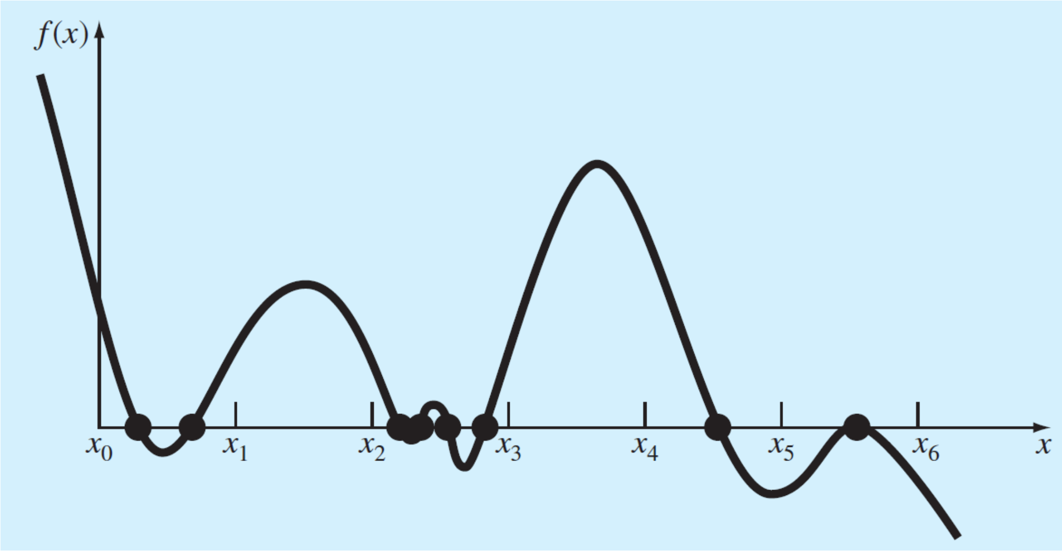


El problema se reduce a encontrar la raíz de una función.

V_0= 20 MeV, a= 2fm, m=938 Mev

E≈2,2 MeV

Métodos iterativos



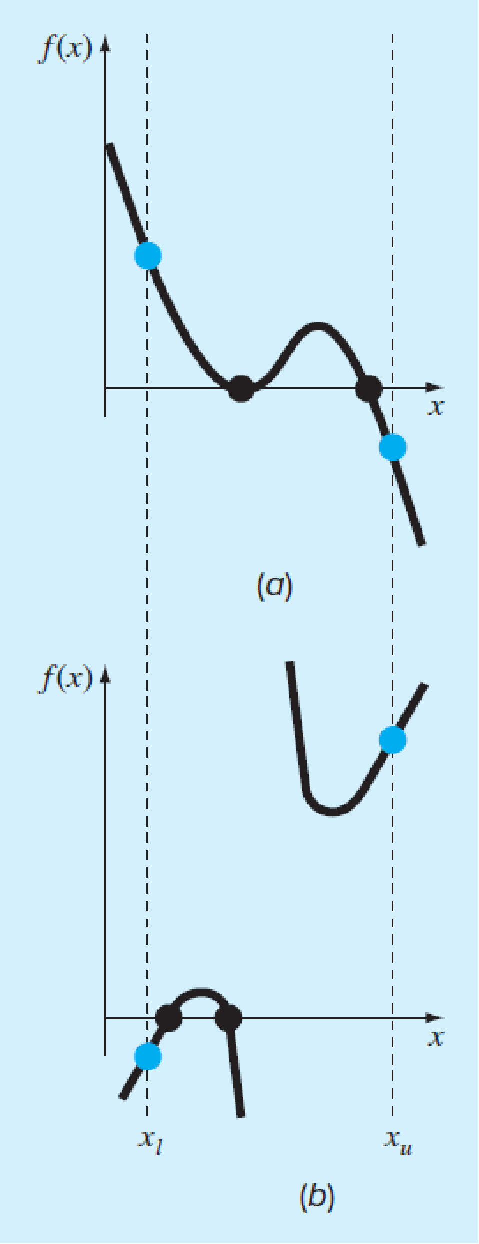
Bracketing methods:

- Acotan la raíz.
- Necesitan dos estimaciones iniciales a los lados de la raíz buscada.
- Ejemplos: búsqueda incremental, bisección, falsa posición, etc.

Open methods:

- Pueden necesitar dos o más estimaciones.
- No acotan la raíz.
- Ejemplos: Newton-Raphson, secante, Broydens,etc

Bracketing Methods



Búsqueda incremental

Si temenos una función continua *f(x),* entonces podemos afirmar que

$$f(x_l) \times f(x_u) < 0$$

Búsqueda incremental

Descripción del método

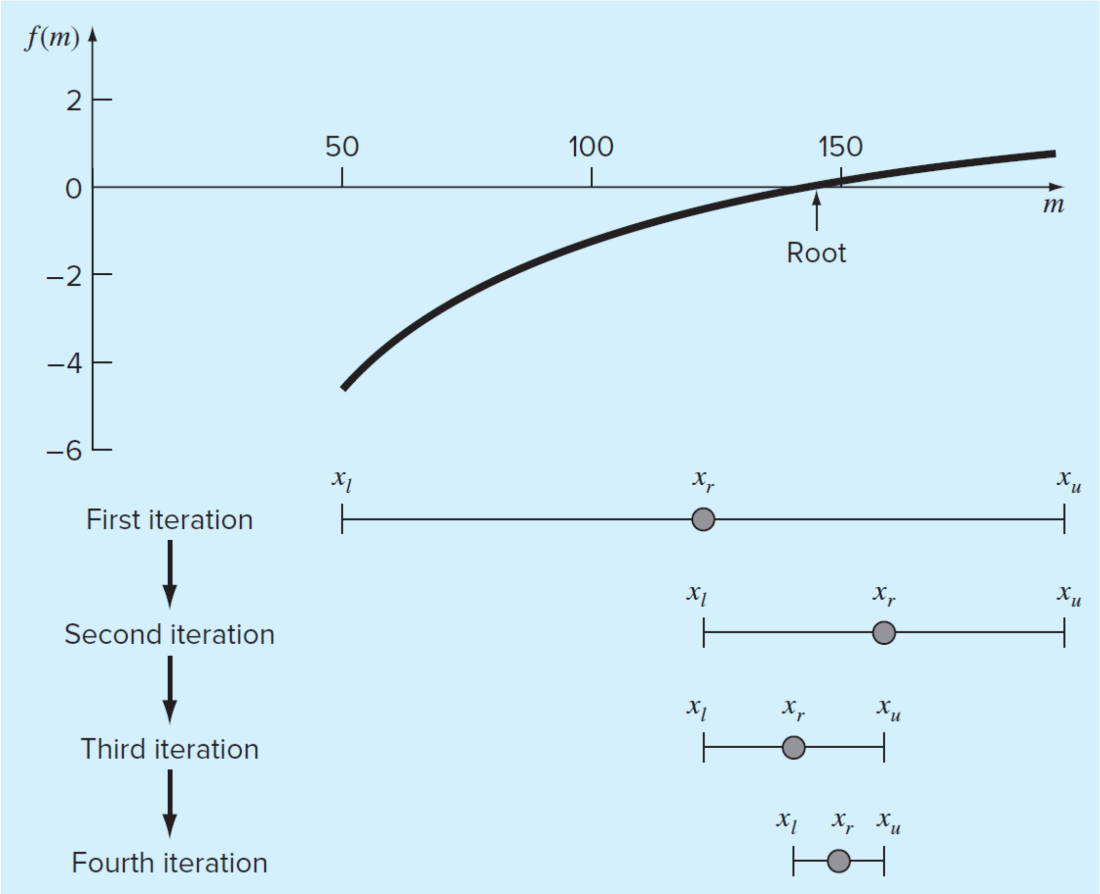
- Elija un punto inicial x_0 y un tamaño de paso Δx. Use un Δx positivo si desea buscar a la derecha, y uno negativo si desea buscar a la izquierda.
- Hacemos $\mathrm{x}_1 = \mathrm{x}_0 + \Delta x$ y calcule $\mathrm{f}(x_0)$ y $\mathrm{f}(x_1)$.
- Si $\mathrm{f}(x_0) \times \mathrm{f}(x_1) < 0$, se supone que existe al menos una raíz en el intervalo $(\mathrm{x}_0, \mathrm{x}_1)$.
- Si el signo de f no cambia entre x_0 y x_0, hacemos $x_2 = x_1 + \Delta x$ y repita el proceso.

Método de la bisección:

- Variante del método de búsqueda incremental.
- En cada iteración el intervalo se divide por la mitad.
- Convergencia lenta
- Dado un intervalo inicial $\epsilon_0 = x_{0u} - x_{0l}$ y fijado un intervalo de tolerancia (error deseado) ϵ_n, luego de

$$\frac{\epsilon_0}{2^n} = \epsilon_n \qquad n = \log_2 \frac{\epsilon_0}{\epsilon_n}$$

iteraciones el método converge.



Método de la bisección: Algoritmo

Step 1: Choose lower x_l and upper x_u guesses for the root such that the function changes sign over the interval. This can be checked by ensuring that $f(x_l)f(x_u) < 0$.

Step 2: An estimate of the root x_r is determined by

$$x_r = \frac{x_l + x_u}{2}$$

Step 3: Make the following evaluations to determine in which subinterval the root lies:

(*a*) If $f(x_l)f(x_r) < 0$, the root lies in the lower subinterval. Therefore, set $x_u = x_r$ and return to step 2.

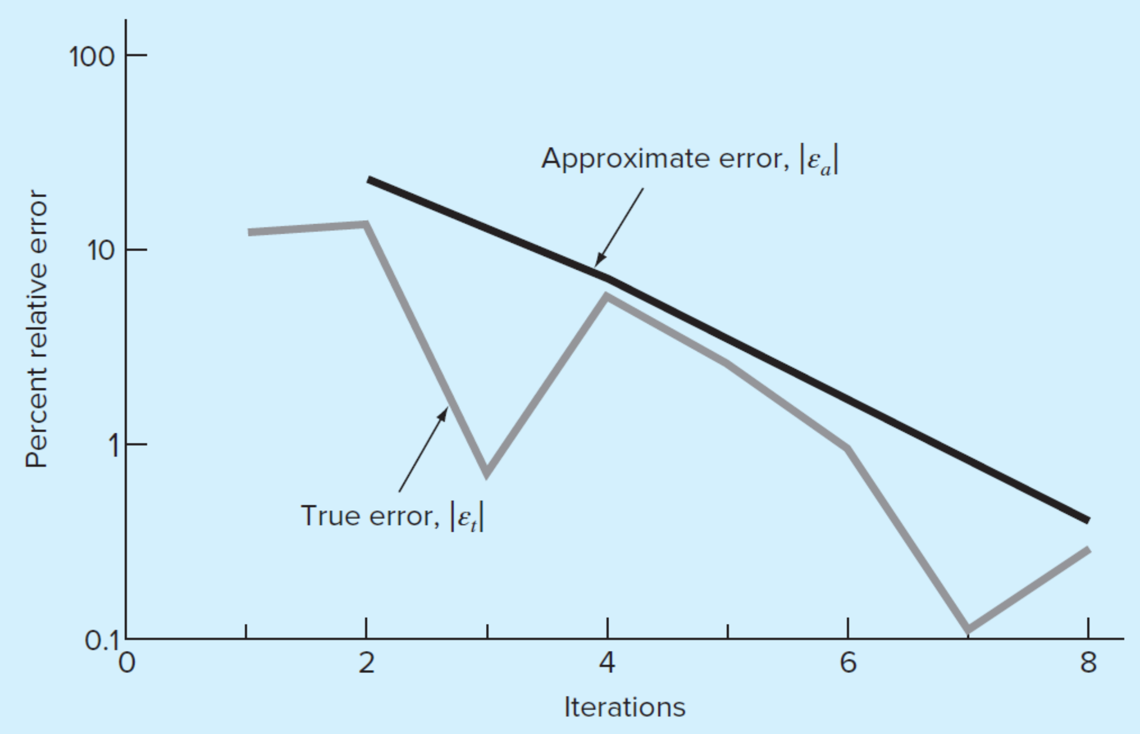
(*b*) If $f(x_l)f(x_r) > 0$, the root lies in the upper subinterval. Therefore, set $x_l = x_r$ and return to step 2.

(*c*) If $f(x_l)f(x_r) = 0$, the root equals x_r; terminate the computation.

Método de la bisección: Pseudo código

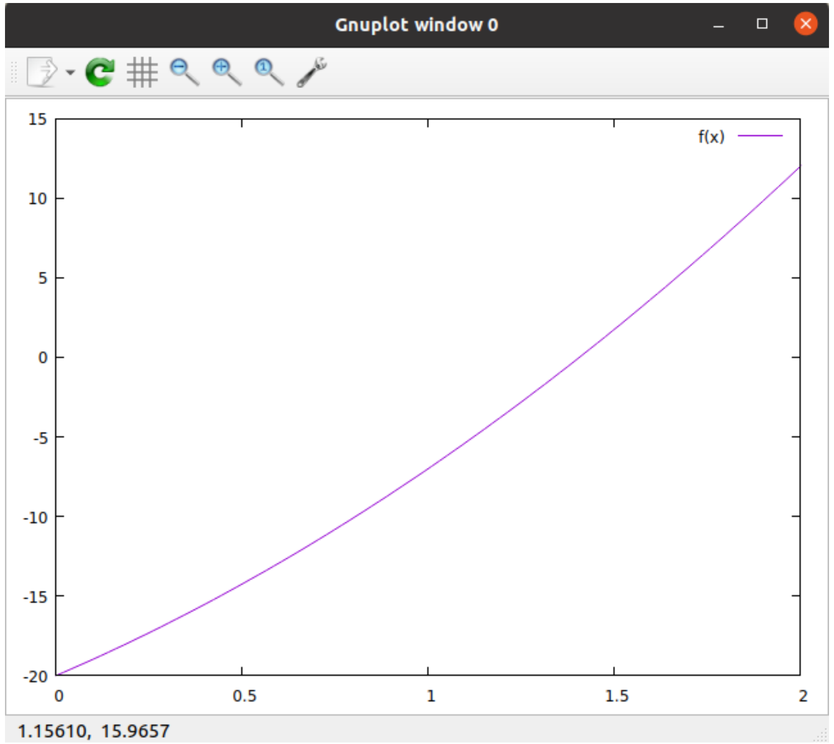
```
FUNCTION Bisect(xl, xu, es, imax, xr, iter, ea)
  iter = 0
  DO
    xrold = xr
    xr = (xl + xu) / 2
    iter = iter + 1
    IF xr ≠ 0 THEN
       ea = ABS((xr − xrold) / xr) * 100
    END IF
    test = f(xl) * f(xr)
    IF test < 0 THEN
       xu = xr
    ELSE IF test > 0 THEN
       xl = xr
    ELSE
       ea = 0
    END IF
    IF ea < es OR iter ≥ imax EXIT
  END DO
  Bisect = xr
END Bisect
```

$$|\varepsilon_a| = \left|\frac{x_r^{\text{new}} - x_r^{\text{old}}}{x_r^{\text{new}}}\right| 100\% \qquad \varepsilon_t = \frac{\text{true value} - \text{approximation}}{\text{true value}}\, 100\%$$



Método de la bisección:

Ejemplo: Encontrar la raiz del polinomio $f(x)=x^2+2x^2+10x-20$
Considerando x_{0l}=1 y x_{0u}=2 para una tolerancia de 10^{-3}.



$$n=\log_2\frac{1}{10^{-3}}=9.965$$

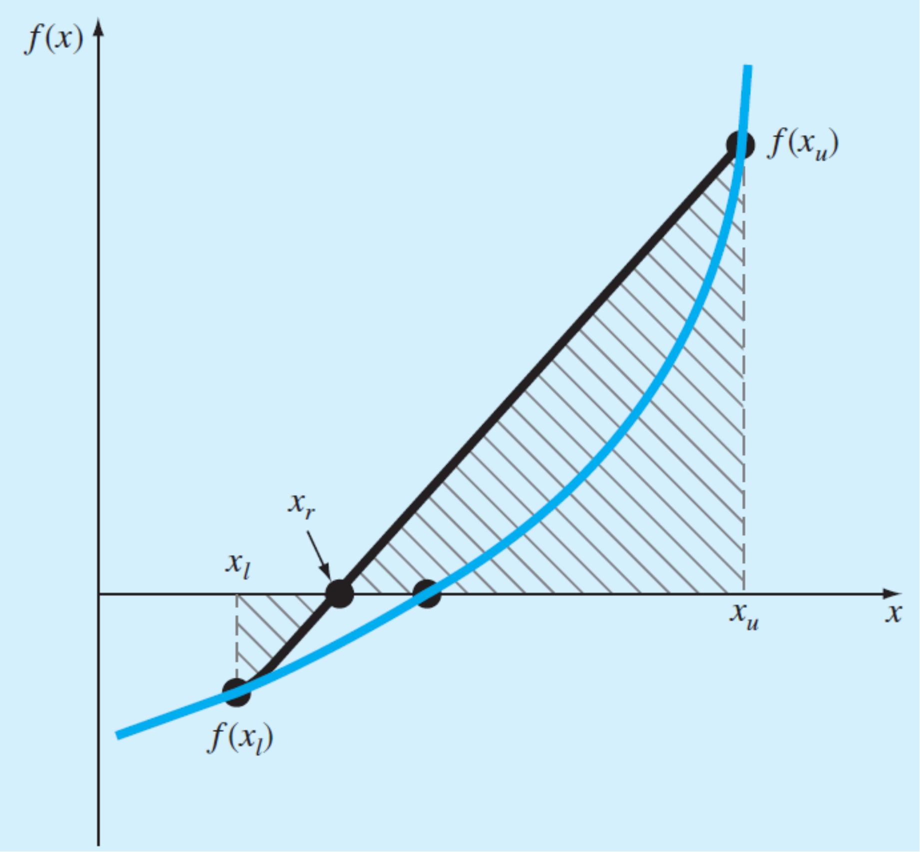
i	x_l	x_u	f(x)
0	1.000	2.000	
1	1.000	2.000	2.875
2	1.000	1.500	2.421
3	1.250	1.500	0.130
4	1.250	1.375	1.168
5	1.312	1.375	0.524
6	1.343	1.375	0.198
7	1.359	1.375	0.034
8	1.367	1.375	0.048
9	1.367	1.371	0.007
10	1.367	1.369	0.013
11	1.368	1.369	0.003

Método falsa posición

- Llamado método de interpolación lineal.
- Para funciones suaves cerca de la raíz, converge más rápido que el método de la bisección.
- **Las pendientes de los triángulos son iguales**.
- La intersección con el eje x es una estimación de la raíz buscada.
- Mismo algoritmo que método de la bisección.

$$x_r = x_u - \frac{f(x_u)(x_l - x_u)}{f(x_l) - f(x_u)}$$

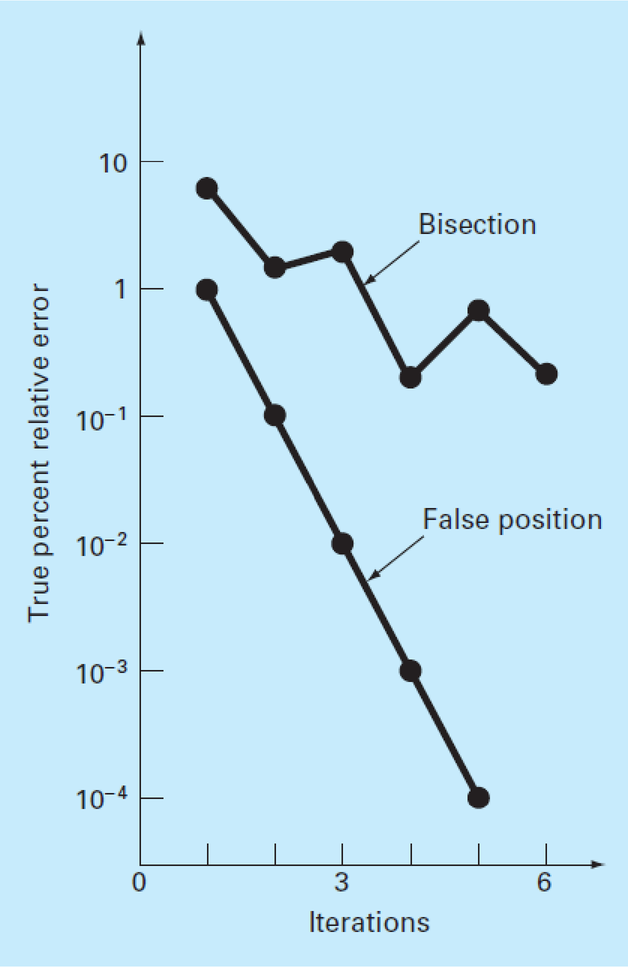
solo el valor x_r es modificado en el algoritmo.



Falsa posición vs bisección

- Error en método de la falsa posición decrece más rápido que para el caso del método de la bisección.

- En cada iteración el intervalo para el método de la bisección se reduce a la mitad.



Desventajas

- No se puede utilizar si se tiene una raiz doble (multiple) o si las raices están muy cerca. Uno de los valores inciales debe estar entre las dos raices de otra manera no se detectarán.

Open Methods

Método punto fijo

- Sustitución sucesiva.
- Se reordena la ecuación de tal manera que la variable x siempre aparezca a la izquierda

$$x = g(x) \text{ -------> } x_{i+1} = g(x_i)$$

Ejemplo:

x^2-2x +3 =0 -------> x=(x^2 +3)/2 ------> $x_{i+1} = x^2_{i+1}$

Sin(x) = 0 -------> x = Sin(x) +x ------> x_{i+1} = Sin(x_i) + x_i

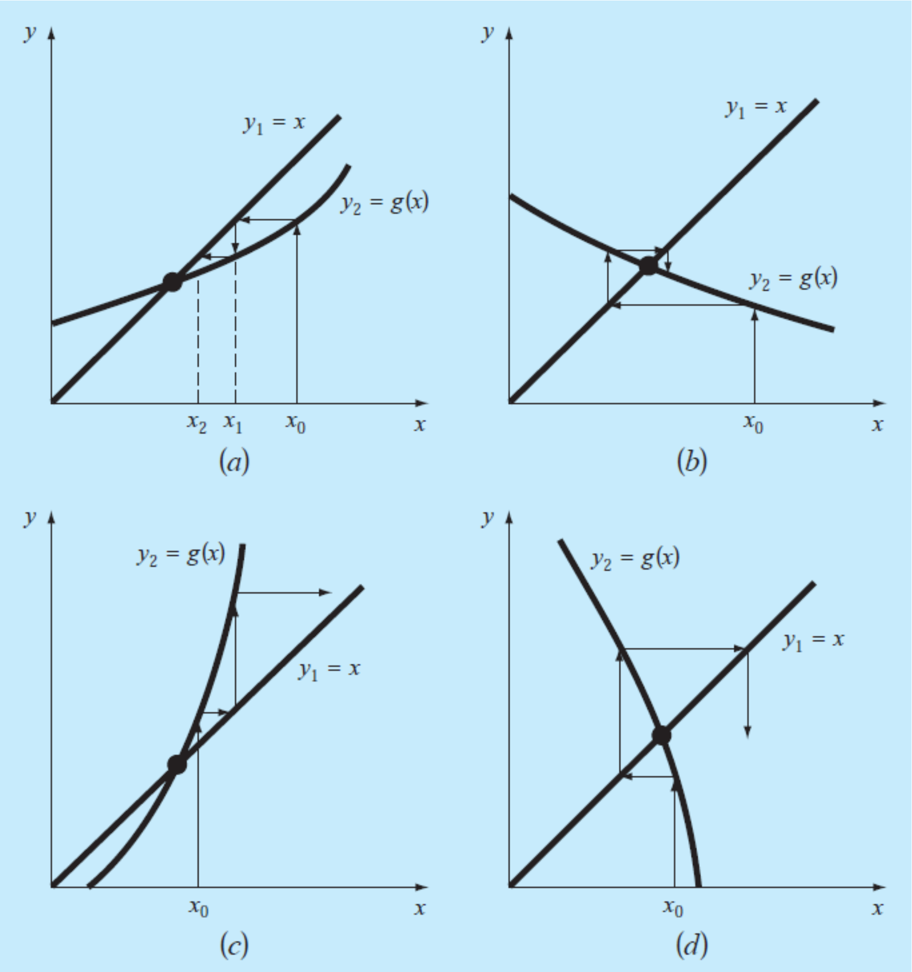
Método punto fijo

Criterio de convergencia: $|g'(x)|<1$

$g(x_i)-g(x_{i-1}) = g'(\varepsilon_i)(x_i - x_{i-1})$, $\varepsilon_i \in [x_i,x_{i-1}]$

$|x_{i+1} - x_i| = |g'(\varepsilon_i)|\ |x_i - x_{i-1}|$, $\varepsilon_i \in [x_i,x_{i-1}]$
para todo $i=0,\ldots,n$

Suponiendo que $|g'(x)| \leq M$
$|x_{i+1} - x_i| \leq M\ |x_i - x_{i-1}|$ para todo $i=0,\ldots,n$
$|x_{i+1} - x_i| \leq M^i\ |x_1 - x_0|$ para todo $i=0,\ldots,n$



M<1 Es condición suficiente, pero no necesaria para la convergencia.

Método punto fijo

$$|\varepsilon_a| = \left|\frac{x_r^{\text{new}} - x_r^{\text{old}}}{x_r^{\text{new}}}\right| 100\%$$

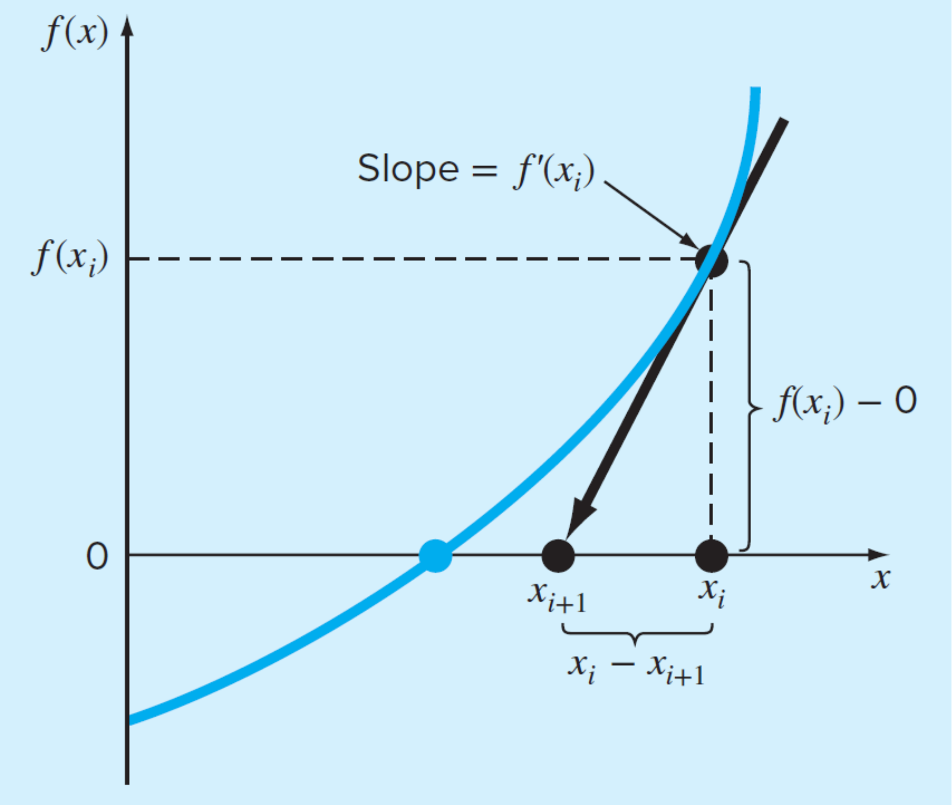
Convergencia lineal

$$\varepsilon_i = a\varepsilon_{i-1}$$

```
FUNCTION Fixpt(x0, es, imax, iter, ea)
  xr = x0
  iter = 0
  DO
    xrold = xr
    xr = g(xrold)
    iter = iter + 1
    IF xr ≠ 0 THEN
      ea = |(xr − xrold) / xr| · 100
    END IF
    IF ea < es OR iter ≥ imax EXIT
  END DO
  Fixpt = xr
END Fixpt
```

Método Newton Raphson

- Necesita evaluación de la función f y de su derivada en puntos arbitrarios.
- Recomendable cuando sea fácil obtener la derivada de f.
- Si la función no es suave se desaconseja su uso.
- Estimación inicial : x_i
- Se prolonga la tangente que pasa por el punto $(x_i, F(x_i))$ y se obtiene x_{i+1}.
- x_{i+1} es una mejor estimación de la raíz.



$$f'(x_i) = \frac{f(x_i) - 0}{x_i - x_{i+1}} \qquad x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)}$$

Método Newton Raphson: Convergencia

Usamos una aproximación de Taylor

$$f(x_{i+1}) = f(x_i) + f'(x_i)(x_{i+1} - x_i) + \frac{f''(\xi)}{2!}(x_{i+1} - x_i)^2$$

$\xi \in [x_{i+1}, x_i]$

Entonces tenemos que $x_{i+1} = x_r$ y $f(x_r) = 0$

$$0 = f(x_i) + f'(x_i)(x_r - x_i) + \frac{f''(\xi)}{2!}(x_r - x_i)^2$$

usando: $x_{i+1} = x_i - \dfrac{f(x_i)}{f'(x_i)}$

tenemos $$0 = f'(x_i)(x_r - x_{i+1}) + \frac{f''(\xi)}{2!}(x_r - x_i)^2$$

haciendo $E_{t,i+1} = x_r - x_{i+1}$

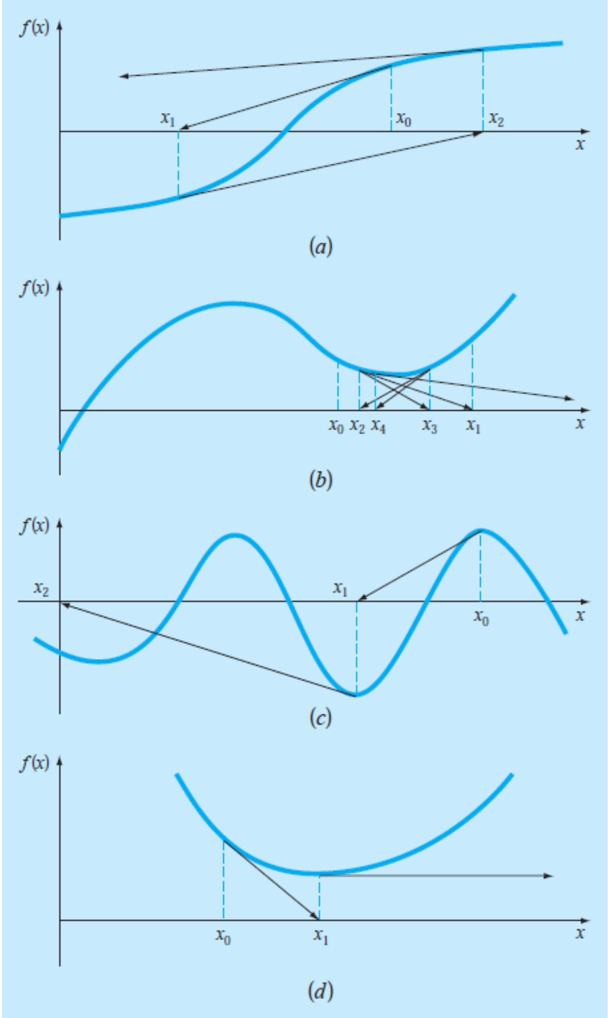
$$0 = f'(x_i)E_{t,i+1} + \frac{f''(\xi)}{2!}E_{t,i}^2$$

$$\Rightarrow \quad E_{t,i+1} = \frac{-f''(x_r)}{2f'(x_r)}E_{t,i}^2$$

Método Newton Raphson: Convergencia

- Convergencia cuadrática:
  Error proporcional al cuadrado del error precedente
- No recomendable con problemas de raíces múltiples.

$$E_{t,i+1} = \frac{-f''(x_r)}{2f'(x_r)} E_{t,i}^2$$

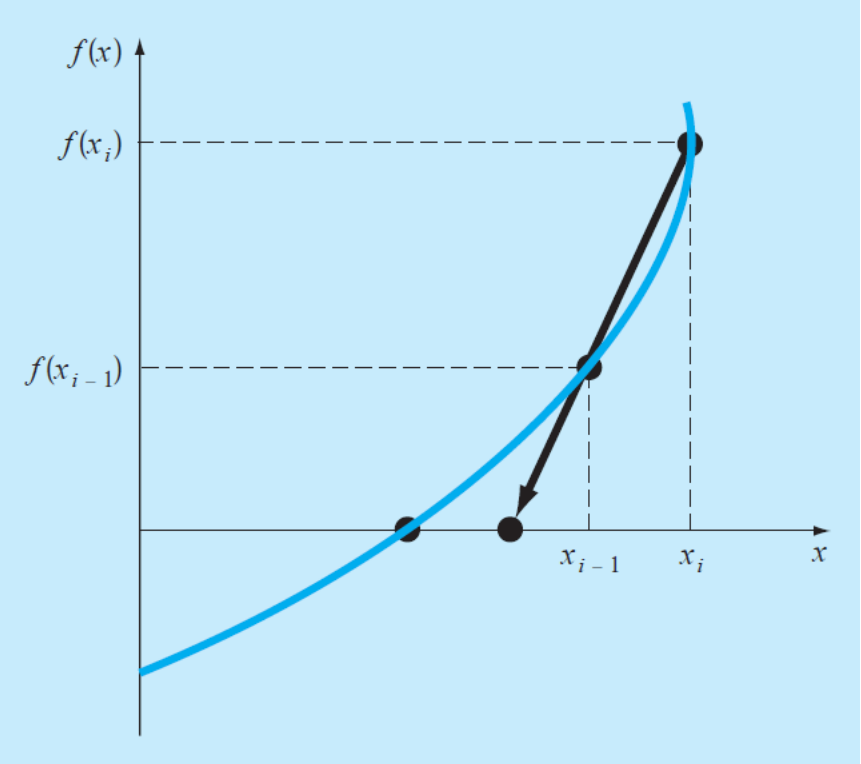


Método de la secante

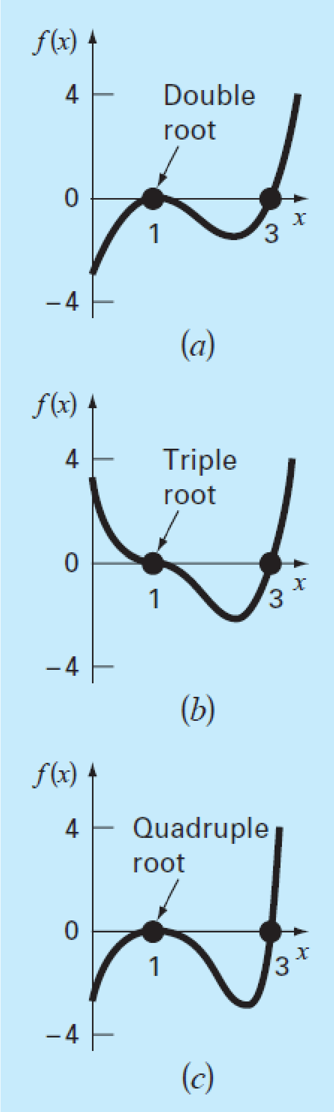
Método de Newton-Raphson con aproximación backward para la derivada

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

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



Raíces múltiples



- Raíces de multiplicidad impar: cruzan el eje x.
- Raíces de multiplicidad par: la función no cambia de signo.
- La derivada de f(x) puede ser cero en la raíz.
- Newton Raphson y la Secante convergen linealmente.
- f(x) se anula antes que f’(x)

$$u(x) = \frac{f(x)}{f'(x)}$$

$$u'(x) = \frac{f'(x)f'(x) - f(x)f''(x)}{[f'(x)]^2}$$

Newton-Raphson modificado

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

$$x_{i+1} = x_i - \frac{f(x_i)f'(x_i)}{[f'(x_i)]^2 - f(x_i)f''(x_i)}$$

Secante modificado

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

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

Conclusiones

- Todos los métodos estudiados solo pueden encontrar raices reales.
- Algunos métodos no pueden encontar la raices de funciones que son tangentes al eje x.

Constante de desplazamiento Wien

La ley de radiación de Planck nos dice que la intensidad de radiación
Por unidad de área por unidad de longitud de onda de un cuerpo
Negro a una temperatura T es:

$$I(\lambda) = \frac{2\,\pi h c^2\,\lambda^{-5}}{e^{hc/\lambda k_B T}\ - 1}$$

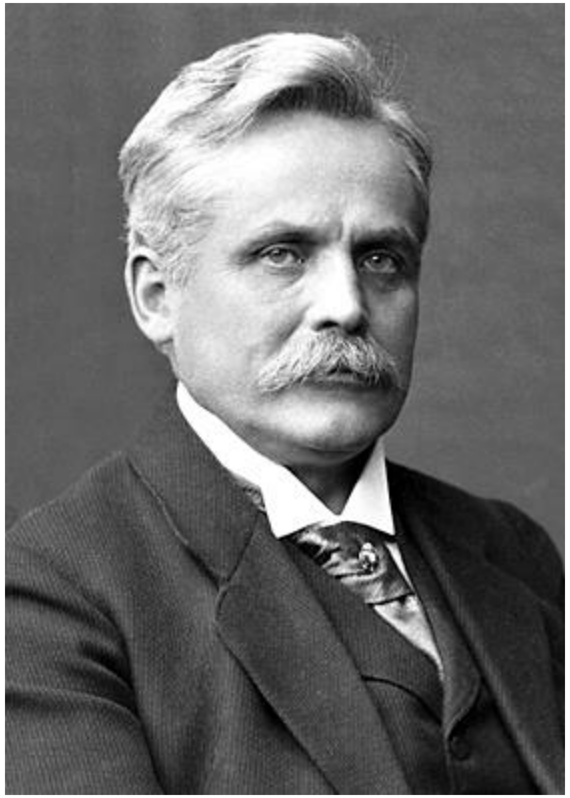
A) Muestre que la longitud de onda λ para la cual $\mathrm{I}(\lambda)$ es máxima verifica la ecuación

$$5e^{-x} + x - 5 = 0, \qquad x = {}^{hc}/_{\lambda k_B T}$$

donde se puede identificar la ley de desplazamiento de Wien

$$\lambda = \frac{b}{T}; \quad b = {}^{hc}/_{k_B x}$$

b es llamada la constante de desplazamiento de Wien



Constante de desplazamiento Wien

B) Escirba un programa que determine el valor de b con una precisión de $\epsilon = 10^{-6}$. Use diferentes métodos.

C) Haga un gráfico en el que se observe la evolución del error (disminución) con respect al número de iteraciones. Decida a partir de este gráfico cuál es el major método.