3 Raíces de ecuaciones

3.1 Ejemplo de motivación

Los planetas al girar alrededor del sol se mueven en órbitas elípticas no uniformes. Su movimiento cumple las leyes de Kepler, de las cuales la segunda ley específica que el momento angular es constante.

La ecuación de Kepler para el movimiento planetario es

$$M = E - \varepsilon \operatorname{sen} E,$$

donde M es la anomalía media, ε es la excentricidad de la elipse y E es la anomalía excéntrica. Puede ser de interés resolver la ecuación para encontrar la anomalía excéntrica (E) para lo cual nos encontramos ante una ecuación no lineal muy difícil de resolver mediante métodos algebraicos.

Observemos en la siguiente imagen el movimiento que describe un objeto celeste al trasladarse en el cielo. Para conocer con exactitud estos movimientos debemos ser capaces de resolver ecuaciones como la de Kepler.

Figura 5: Ejemplo de raíces de ecuaciones.

Nota: Órbitas planetarias, ©2007–08 Tunç Tezel.

En esta sección estudiaremos tres métodos que nos permiten resolver ecuaciones no lineales de forma numérica. Varios fenómenos pueden ser descritos mediante el uso de ecuaciones o sistemas de ecuaciones no lineales, es muy importante entonces conocer formas de hallar soluciones de forma eficiente.

Puede encontrar información interesante sobre este tema en el siguiente enlace `https://earthobservatory.nasa.gov/features/OrbitsHistory`

3.2 Método de la bisección

Si tenemos una función, por ejemplo, $f(x) = x^2 + 4x - 5$, puede ser de nuestro interés encontrar los valores en los cuales la función se anula. Estos valores son comúnmente conocidos como raíces de la ecuación. Para nuestro ejemplo, es claro que las raíces de f son $x_1 = 1$ y $x_2 = -5$ (para un polinomio de segundo grado es muy fácil encontrar estos valores factorando la expresión). Ahora bien, si tenemos la función $g(z) = z^3 - \cos(z)$ ya no es sencillo encontrar el valor de z que anule nuestra función g.

Estas raíces nos pueden dar información muy importante sobre el fenómeno que se está modelando con la ecuación, por lo tanto, es importante tener una forma sencilla y eficiente para encontrar estos valores.

Recordemos ciertas definiciones de funciones que nos son de utilidad.

Definición 8: Función continua en un punto

Si x_0 es un punto del dominio de una función f, entonces f es continua en x_0 si,

$$\lim_{x \to x_0} f(x) = f(x_0).$$

La función es continua en un conjunto $A \subseteq \mathrm{Dom}(f)$, si es continua en todo punto del conjunto A.

Teorema 1: Teorema del valor medio

Si la función f es continua en el intervalo cerrado $[a, b]$, y si $f(a) \leq y \leq f(b)$ o $f(b) \leq y \leq f(a)$, existe entonces un punto c tal que $a \leq c \leq b$ y $f(c) = y$, (Cheney y Kincaid, 2008).

Si una función continua en un intervalo $I = [a, b]$ tiene valores de signo opuesto en cada extremo del intervalo, entonces f tiene una raíz en I. En otras palabras, existe un valor $c \in [a, b]$ tal que $f(c) = 0$. Esto es una conclusión del teorema del valor medio.

Piense detenidamente sobre este fenómeno, grafique algunas funciones y observe como el teorema del valor medio se cumple. No olvide que la función debe ser continua.

Utilizando la idea del teorema del valor medio, podemos desarrollar un algoritmo que nos permita encontrar la raíz de una función dentro de un intervalo. Si tenemos una función f y buscamos una raíz en el intervalo $[a, b]$, comprobamos primero que $f(a)$ y $f(b)$ tengan signos contrarios, es decir, si $f(a) \cdot f(b) < 0$. Luego hacemos $c = \frac{a+b}{2}$ y calculamos $f(c)$. Comprobamos el signo de $f(a) \cdot f(c)$, si este es menor que cero entonces la raíz está en el intervalo $[a, c]$, caso contrario sabemos que se encuentra en $[c, b]$. Reasignamos según el caso a las variables $a = c$ o $b = c$. Notemos que el nuevo intervalo mide la mitad que el anterior. Seguimos este procedimiento hasta alcanzar el máximo de iteraciones o la tolerancia deseada.

Observemos a continuación un seudocódigo del algoritmo de la bisección.

Seudocódigo 1: Método de la bisección (Cheney y Kincaid, 2008)

```
Entradas: f, a, b, nmax, ε
entero n, nmax; real a, b, c, fa, fb, fc, error
función f
fa ← f(a)
fb ← f(b)
if sign(fa)=sign(fb) then
    output: a, b, fa, fb
    output: "f(a) y f(b) tienen el mismo signo"
error ← b-a
for n=0 to nmax do
    error ← error/2
    c ← a+error
    fc ← f(c)
    output: n, c, fc, error
    if |error| < ε then
        output: "convergencia"
        return
    if sign(fa) ≠ sign(fc) then
        b ← c
        fb ← fc
    else
        a ← c
        fa ← fc
```

Notemos que evaluar una función en un valor es una operación "cara", por lo que es mejor realizarla la menor cantidad de veces. Observemos también que mientras avanza el algoritmo la diferencia entre $f(a)$ y $f(b)$ puede ser muy pequeña, esto puede provocar un subdesbordamiento aritmético.

Programe el método y busque raíces para varias funciones. Experimente con diferentes valores para los criterios de parada. Busque formas de evitar un subdesbordamiento aritmético al hacer $f(a) \cdot f(b)$.

Si revisamos brevemente la convergencia del método de la bisección tenemos que su convergencia es global, es decir, si se cumplen las condiciones iniciales el método siempre converge. Un posible inconveniente es que su convergencia es lineal. Para calcular el número de iteraciones n necesario para alcanzar una tolerancia ε tenemos la siguiente fórmula,

$$n > \frac{\log(b-a) - \log(2\varepsilon)}{\log 2},$$

puede encontrar más sobre este tema en los libros referidos en la bibliografía del curso.

Investigue sobre el método conocido como *Regula falsi* (regla falsa). ¿Cuáles son las similitudes y las diferencias con el método de la bisección? ¿Qué ventajas e inconvenientes presenta este método?

3.3 Método de Newton

Si desarrollamos la fórmula de Taylor alrededor de un punto x_i obtenemos,

$$f(x) = f(x_i) + f'(x_i)(x - x_i) + \frac{f''(x_i)}{2!}(x - x_i)^2 + \dots$$

recordemos que la fórmula de Taylor aproxima la función en las cercanías del punto mediante un polinomio. Por facilidad truncamos el polinomio a partir del término de segundo orden y obtenemos la siguiente linealización de f en los valores cercanos a x_i

$$f_l(x) = f(x_i) + f'(x_i)(x - x_i)$$

entonces, el cero de f se puede aproximar con el cero de la linealización f_l que se encuentra fácilmente igualando a cero y despejando la ecuación anterior. Obtenemos entonces la iteración para el método de Newton,

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

Definición 9: Método de Newton

Sea una función f derivable podemos aproximar c, el cero de la función f, por medio de la iteración

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

Para obtener x_{i+1} tomamos la recta tangente de f en el punto x_i. Donde la recta tangente corte con el eje de las x, será x_{i+1}, (Rojas y Serrano, 2019).

Uno de los problemas del método de Newton es que no podemos garantizar su convergencia. La convergencia depende en su mayoría de que la elección del punto x_0 haya sido buena, es decir, que haya elegido x_0 cerca de la raíz. Si la raíz coincide con un punto de inflexión o la función tiene pendientes grandes cerca de la raíz seguramente las iteraciones comenzarán a oscilar y el algoritmo no logrará converger.

Revisemos la iteración del método de Newton con un ejercicio corto. Mientras revisa el ejercicio medite sobre los posibles inconvenientes en la implementación del algoritmo.

Ejercicio: Mediante el método de Newton encuentre el punto de intersección aproximado de las gráficas para las funciones f y g de ecuaciones $f(x) = e^x$ y $g(x) = -x$, respectivamente. Tome un valor inicial de $x_0 = 0.75$ y 2 iteraciones. Ejercicio tomado de (Rojas y Serrano, 2019).

Para encontrar el punto de intersección de las gráficas de las dos funciones dadas tenemos que igualarlas, es decir resolver la ecuación

$$e^x = -x, \qquad \text{o} \qquad e^x + x = 0.$$

Si definimos $h(x) := e^x + x$ y encontramos el cero de h mediante el método de Newton podemos encontrar la intersección de las gŕaficas de las dos funciones. Sabemos que, para el método de Newton, las aproximaciones del cero de h para $n \in \mathbb{N}$ están dadas por:

$$x_{n+1} = x_n - \frac{h(x_n)}{h'(x_n)} = x_n - \frac{e^{x_n} + x_n}{e^{x_n} + 1}.$$

Entonces la iteración inicial con $x_0 = 0.75$ es

$$\begin{aligned} x_1 &= x_0 - \frac{e^{x_0} + x_0}{e^{x_0} + 1} \\ &= 0.75 - \frac{e^{0.75} + 0.75}{e^{0.75} + 1} \\ &\approx -0.169. \end{aligned}$$

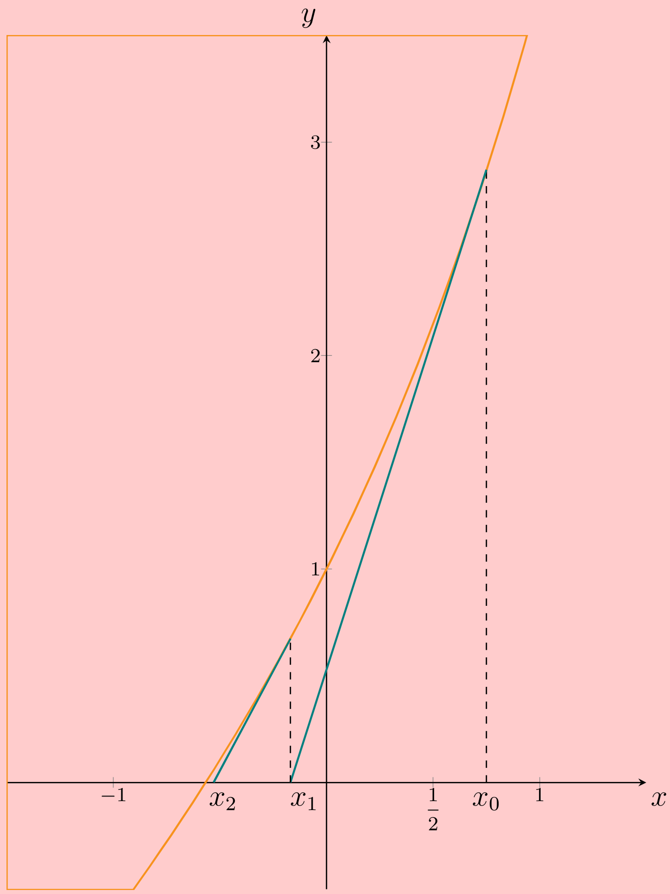
La segunda iteración es

$$\begin{aligned} x_2 &= x_1 - \frac{e^{x_1} + x_1}{e^{x_1} + 1} \\ &= -0.169 - \frac{e^{-0.169} - 0.169}{e^{-0.169} + 1} \\ &\approx -0.534, \end{aligned}$$

la aproximación del punto de intersección $\overline{x}$ es -0.534.

Podemos ver gráficamente como se aplicó el método de Newton para la función h.

Figura 6: Raíces por el método de Newton.



Nota: Iteraciones en el cálculo de raíces utilizando el método de Newton, elaboración propia.

Luego de observar el ejercicio sobre el método de Newton, intente encontrar la fórmula de la iteración de forma geométrica basándose únicamente en el gráfico de una función, los puntos x_0 y x_1, y la tangente en $(x_0, f(x_0))$.

A continuación, tenemos el seudocódigo para el método de Newton.

Seudocódigo 2: Método de Newton (Cheney y Kincaid, 2008)

```
Entradas: f, a, b, nmax, ε
entero n, nmax; real a, b, fa, fb, ε, d
función f, f'
fx ← f(x)
output: 0, x, fx
for n=1 to nmax do
    df ← f'(x)
    if df< δ then
        output: "valor de la derivada muy pequeño"
        return
    d ← fx/df
    x ← x-d
    fx ← f(x)
    output: n, x, fx
    if |d| < ε then
        output: "convergencia"
        return
```

¿Para qué se utilizan los valores ε y δ dentro del algoritmo? ¿Guardan estos valores alguna relación con el número máximo de iteraciones?

El método de Newton es generalmente muy rápido para converger, tiene un orden de convergencia cuadrático. Su velocidad de convergencia depende de la multiplicidad de las raíces buscadas y de la cercanía del punto inicial a la raíz buscada.

Investigue sobre los inconvenientes del método de Newton, ¿en qué casos el método no converge? Programe el método y busque casos de convergencia cuadrática y lineal. Lea en la bibliografía sobre la aplicación del método de Newton para resolver sistemas de ecuaciones no lineales.

3.4 Método de la secante

El método de la secante es un método bastante práctico, porque presenta una convergencia casi igual de rápida que la del método de Newton, pero no necesita del cálculo de la derivada de la función. Esta ventaja es muy importante, puesto que el cálculo de la derivada de una función puede algunas veces ser complicado.

La idea principal en el método de la secante es utilizar la fórmula del método de Newton, pero prescindir de la derivada. Podemos aproximar la derivada utilizando la definición,

$$f'(x) = \lim_{h \to 0} \frac{f(x+h) - f(x)}{h},$$

sabemos que, si h es lo suficientemente pequeño, tenemos que

$$f'(x) \approx \frac{f(x+h) - f(x)}{h},$$

para no depender del valor h, utilizamos una aproximación llamada *diferencia finita* que consiste en aproximar la primera derivada con,

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

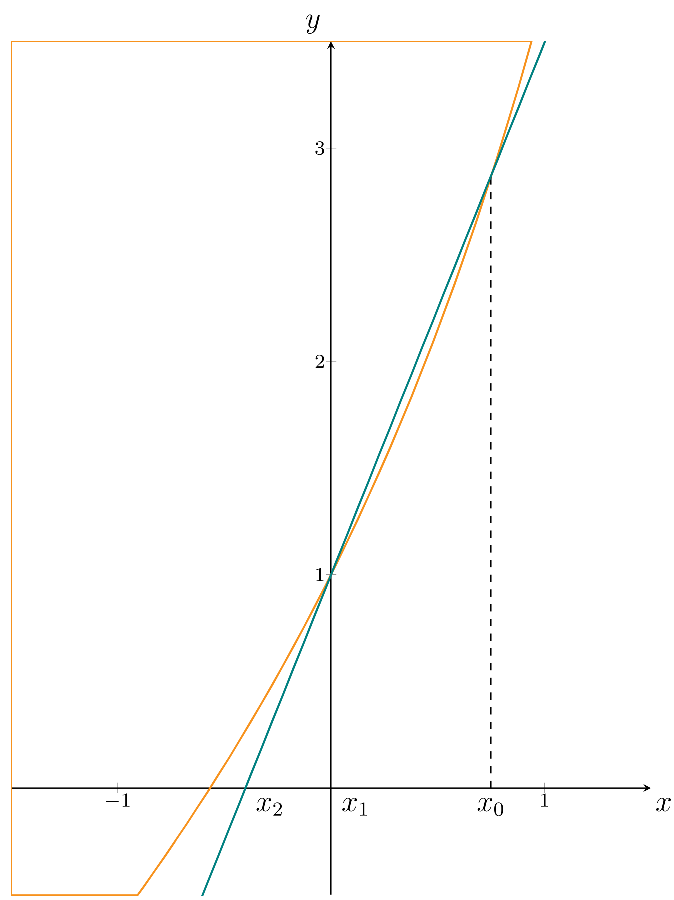
Reemplazamos esta aproximación en la fórmula del método de Newton y obtenemos que

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

Note que el valor de x_{i+1} depende de dos valores anteriores de la iteración. Es por esto que para inicializar el método se requiere x_0 y x_1.

Observe en el siguiente gráfico, para el mismo ejemplo del método de Newton, como funciona la primera iteración del método de la secante. Tomando $x_0 = 0.75$ y $x_1 = 0$, se obtiene una buena aproximación en la primera iteración del método.

Figura 7: Raíces por el método de la secante.



Nota: Primera iteración utilizando el método de la secante, elaboración propia.

¿Cómo la aproximación por diferencias finitas aproxima la derivada? Realice un pequeño gráfico y observe como funciona esta aproximación.

Note que se debe tener cuidado al hacer la división para $f(x_i) - f(x_{i-1})$, puesto que al acercarse a la raíz este valor será muy pequeño y podemos producir un desbordamiento aritmético. Tenemos a continuación el seudocódigo del método de la secante.

Seudocódigo 3: Método de la secante (Cheney y Kincaid, 2008)

```
Entradas: f, f', x, nmax, ε, δ
entero n, nmax; real x, fx, df, d, ε, δ
función f
fa ← f(a)
fb ← f(b)
if |fa| > |fb| then
    a⟷b
    fa⟷fb
output: 0, a, fa
output: 1, b, fb
for n=2 to nmax do
    if |fa| > |fb| then
        a⟷b
        fa⟷fb
    d ← (b-a)/(fb-fa)
    b ← a
    fb ← fa
    d ← d· fa
    if |d| < ε then
        output: "convergencia"
        return
    a ← a-d
    fa ← f(a)
    output: n, a, fa
```

El método de la secante tiene una convergencia super lineal si se parte cerca de la raíz. Es entonces un poco más lento que el método de Newton, pero más rápido en converger que la bisección.

¿Se podría crear un método híbrido que utilice las bondades de dos de los métodos estudiados? ¿Cómo sería este método?

Recursos complementarios

- Vídeo sobre el método de Newton `https://youtu.be/-5e2cULI3H8`

- Vídeo sobre el método de la secante `https://youtu.be/ZfMibcIr5YA`
- Vídeo sobre el método de la bisección en Python `https://youtu.be/QcuVPbN4_Vk`