Análisis de diversos metodos numéricos para hallar raices

Julián Arturo Calle

06 de Febrero de 2019

1. Descripción de los Algoritmos

Para la realizacion del trabajo se selecionaron los algoritmos de biseccion, punto fijo secante para hallar las raices de una funcion y estimar su error.

1.1. Algoritmo de bisección

El algortimo de bisección consiste en tomar un intervalo $[a, b]$ de una funcion $f(x)$ y dividir la funcion en dos, a partis de este punto se evalua la funcion en el punto medio y la siguiente iteracion se realiza sobre el sentido de la funcion que mas se aproxime a 0. partiendo en dos la funcion por cada iteracion y asi aproximar el punto de la raiz. existen condiciones para que se pueda hallar la raiz en un punto de la función:

- En el intervalo: $[a, b]$ de la funcion $f(x)$ solo debe contener una de las raices.
- Debe existir una raiz en el intervalo: $[a, b]$ de la funcion $f(x)$ de lo contrario el algoritmo no funcionará

Algorithm 1 Algoritmo de bisección.

```
1: procedure BISECCION(a, b)
2:     Fx ← funcion dada
3:     x ←secuencia de numeros dentro del intervalo
4:     d ← (a+b)/2
5:     i ← 0
6:     error ← |(a−b)/2|
7:     while error>1.e − 4 do
8:         i+1
9:         if Fx(x) = 0 then
10:            break
11:        end if
12:        if Fx(x) ∗ Fx(a)>0 then
13:            b ← x
14:            else: a ← x
15:        end if
16:        d
17:        x ← (a+b)/2
18:        error ← |(a−b)/2|
19:    end while
20: end procedure
```

1.2. Algoritmo de punto fijo

El algoritmo de punto fijo consite en tomar un intervalo $[a, b]$ de una funcion $f(x)$ hallar una funcion $g(x)$ donde se despeje x de la funcion original, y evalauar para intervalos en el segmento si $g(x) = x$, a los cuales se les considera puntos fijos de la funcion $g(x)$ y raices en $f(x)$. Existen condiciones para que se pueda hallar la raiz en un punto de la función:

- La funcion $f(x)$ debe ser continua en el intervalo $[a, b]$
- Debe existir una raiz en el intervalo: $[a, b]$ de la funcion $f(x)$ de lo contrario el algoritmo no funcionará

Algorithm 2 Algoritmo de punto fijo.

```
1: procedure PUNTOFIJO(a, b)
2:     Fx ← funcion dada
3:     Gx ← funcion despejada en x
4:     x ← a
5:     i ← 0
6:     while x<b do
7:         i ← i + 1
8:         resultado ← Gx(x)
9:         if x<resultado  x + 0,1>resultado then
10:            break
11:        end if
12:        x ← x + 0,1
13:    end while
14:    error←—x+0.1-x/|x+0,1|
15:    while x<b do
16:        i ← i + 1
17:        resultado ← Gx(x)
18:        if x<resultado  x + 0,01>resultado then
19:            break
20:        end if
21:        x ← x + 0,01
22:    end while
23: end procedure
```

1.3. Algoritmo de Secante

El algoritmo de punto fijo consite en tomar un intervalo $[a, b]$ de una funcion $F(x)$ hallar su derivada $f(x)$ donde se evalua que: $x_{n+1} = x_n - \frac{x_n - x_{n-1}}{F(x_n)+F(x_{n-1}))}$ donde $F(x_{n+1})$ se aproxima a 0 y el error se puede estimar como: $error = |\frac{F(x)}{f(x)}|$. Existen condiciones para que se pueda hallar la raiz en un punto de la función:

- La funcion $F(x)$ debe tener una raiz en el intervalo $[a, b]$
- la función $F(x)$ debe ser derivable.

Algorithm 3 Algoritmo de bisección.

1: **procedure** SECANTE(x_0, x_1)
2: $Fx \leftarrow$ funcion dada
3: $fx \leftarrow$derivada de la funcion
4: $x \leftarrow \frac{F(x_1)*x_0-F(x_0)*x_1}{F(x_1)-F(x_0}$
5: $i \leftarrow 0$
6: $error \leftarrow 1$
7: **while** $error > 1.e-4$ **do**
8: i+1
9: $x_0 \leftarrow x_1$
10: $x_1 \leftarrow x$
11: $x \leftarrow \frac{F(x_1)*x_0-F(x_0)*x_1}{F(x_1)-F(x_0}$
12: $error \leftarrow |\frac{F(x)}{f(x)}|$
13: **end while**
14: **end procedure**

2. Analisis de complejidad

En vista de que todas las funciones que se muestran hace uso de los valores dentro de un intervalo consideraremos que en el peor de los casos se demoraran la cantidad de datos que recorren por ciclo, es decir que si la funcion es recurrente o iterativa, hara la iteracion la cantidad de datos que deba recorrer: n datos. Dado que ninguna de estas funciones tiene ciclos anidados y se desprecia el resto de pasos a los cuales se accede una unica ves las funciones tienen una complegidad de $O(n)$