Laboratorio de Métodos Numéricos

Departamento de Computación
Facultad de Ciencias exactas, Físicas y Naturales
Universidad Nacional de Córdoba

Alumno: Aguilar Mauricio
Matricula: 34,496,071
Carrera: Ingeniería en Computación
Fecha: 30/04/2011
E-mail: aguilarmauri@gmail.com
Docente de Laboratorio: Ing. Javier Jorge
Docente de Teórico: Ing. Beatriz Pedrotti

Índice de Contenidos

1) Problema Planteado

“Nivel de Iluminación en una Habitación”

La iluminancia “E” producida por una fuente de luz de intensidad “I” a una distancia “r” de la fuente está dada por

$$E(r)=\frac{I}{r^2}$$

La iluminancia total originada por dos lámparas de intensidad I_1 = 125 e I_2 = 216 es la suma de las dos luminancias parciales. Si los focos luminosos están separados a 10 mt., encuentre el punto “P” entre ellos donde la iluminancia total sea mínima (ver Fig. 1).

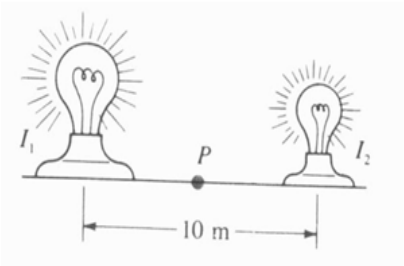


Figura 1: Esquema del problema planteado [1].

Desarrolle un algoritmo capaz de calcular el punto de iluminancia mínima, para valores de intensidad “ I_1 ” e “ I_2 ”, y una distancia “d”, dados por el usuario.

2) Modelo Matemático

Matemáticamente el problema puede modelarse:

$$E(r)=f(r,I_1,I_2,d)$$

Teniendo en cuenta que la luminancia en un cierto punto es la suma de las luminancias parciales en dicho punto, se plantea la ecuación para valores cualesquiera:
La función en forma genérica es:

$$E(r)=E_1+E_2=\frac{I_1}{r^2}+\frac{I_2}{(d-r^2)}$$

La derivada primera genérica es:

$$E'(r)=\frac{-2.\mathrm{I}_1}{r^3}+\frac{2.\mathrm{I}_2}{(d-r)^3}$$

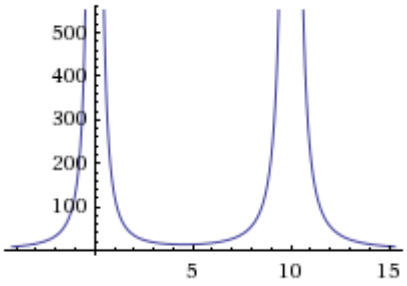
Y la derivada segunda genérica es:

$$E''(r)=\frac{6.\mathrm{I}_1}{r^4}+\frac{6.\mathrm{I}_2}{(d-r)^4}$$

Para los valores $I_1=125$, $I_2=216$ y $d=10\mathrm{m}$ del problema planteado:

$$E(r)=E_1+E_2=\frac{125}{r^2}+\frac{216}{(10-r^2)}$$

Cuya gráfica es:

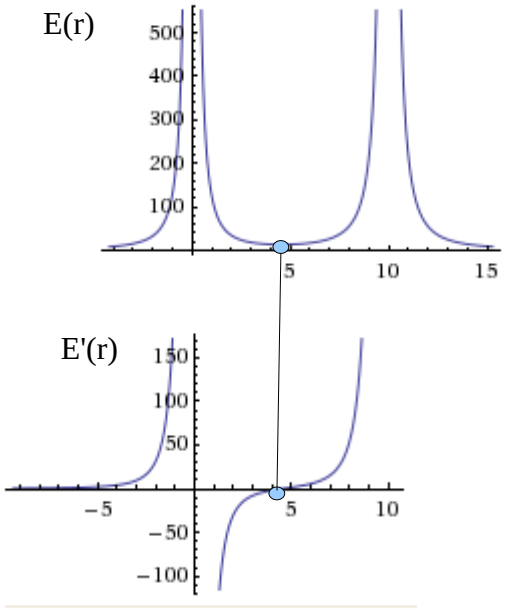


La derivada primera para los valores dados, es:

$$E'(r)=\frac{-250}{r^3}+\frac{432}{(10-r)^3}$$

Resulta muy complicado hallar analíticamente los puntos críticos, para determinar el mínimo de la función con el criterio de la derivada segunda.
Se procede entonces, en busca de un método numérico adecuado que encuentre dicho mínimo.
Dicho método buscará la raíz de la derivada primera de la función en el intervalo (0,d); punto en el cual la función original presenta el mínimo buscado.

Gráficamente esto es:



3) Criterios de selección del Método Numérico

> Como se tiene dos valores iniciales, estos son, r y (d-r), y la derivada de la función E(r) cambia de signo a ambos lados de la raíz, se tiene lo necesario para aplicar un método cerrado.

Entre los métodos cerrados contamos con el Método de la Bisección y el Método de la Falsa Posición o Regula Falsi.

No conviene utilizar el Método de la Falsa posición dado que conociendo el tipo de función con el que tratamos, este método tiene prácticamente el estancamiento asegurado. A mayor distancia entre las fuentes, mas estancamiento.

El Método de la Bisección es totalmente aplicable y tiene convergencia segura. Tiene como desventaja un mayor numero de iteraciones que Newton-Raphson y Secante, y un mayor error.

> Analizando entre los métodos abiertos, contamos con el método de Iteración del punto fijo, Newton-Raphson y el método de la Secante.

El método del punto fijo parece no cumplir con los requisitos necesarios para su convergencia ($|g'(x)|<1$), dada la forma de la función E(r) y sus derivadas, por lo que no será utilizado.

El método de Newton-Raphson parece ser apto para esta función. Tiene como ventaja que si converge, lo hace más rápido que el de Bisección. Como desventaja, que si no se da un valor correcto, diverge.

El método de la Secante mas óptimo que Newton-Raphson ya que, si bien el algoritmo es bastante similar, no requiere el cálculo de la derivada de la función a la cual se le quiere calcular su raíz (en este caso, no requiere E''(r)).

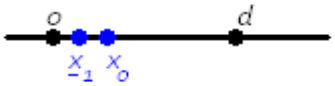
> Se utiliza entonces el método de la Secante, ya que tomando dos valores dentro del intervalo (0,d) converge seguro a la única raíz en dicho intervalo. Además, el número de iteraciones por este método, es significativamente menor al número generado por el método de la Bisección.

4) Algoritmo para resolver el problema planteado

1. Se elige el intervalo donde se encuentre la raíz. Esto es, (0,d).

2. Se toman dos valores iniciales para aplicar M. Secante.

$$x_{-1}=d/22 \qquad x_0=(d/21)$$



3. Se aproxima a la raíz con la ecuación dada por este método.
Esto es,

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

4. Se verifica si Ea<Es o si número de iteraciones es mayor al máximo permitido.

a. En caso de ocurrir una de estas dos cosas, el programa finalizará.

b. En caso de no ocurrir ninguna de las dos condiciones, se regresa al paso 3. donde ahora x_0 y x_{i+1} para el nuevo calculo de la aproximación.

Algoritmo del Método de la Secante:

FUNCION Secante (x_{-1},x_0,es,imax,iter,ea)	Declara la función con sus argumentos datos
xrold=x_{-1}	Guarda valor de X_{-1} en Xr
xr=x_0	Guarda valor de X_0 en Xr
iter=0	Define al iterador en 0 al comenzar
DO	INICIA UN CICLO

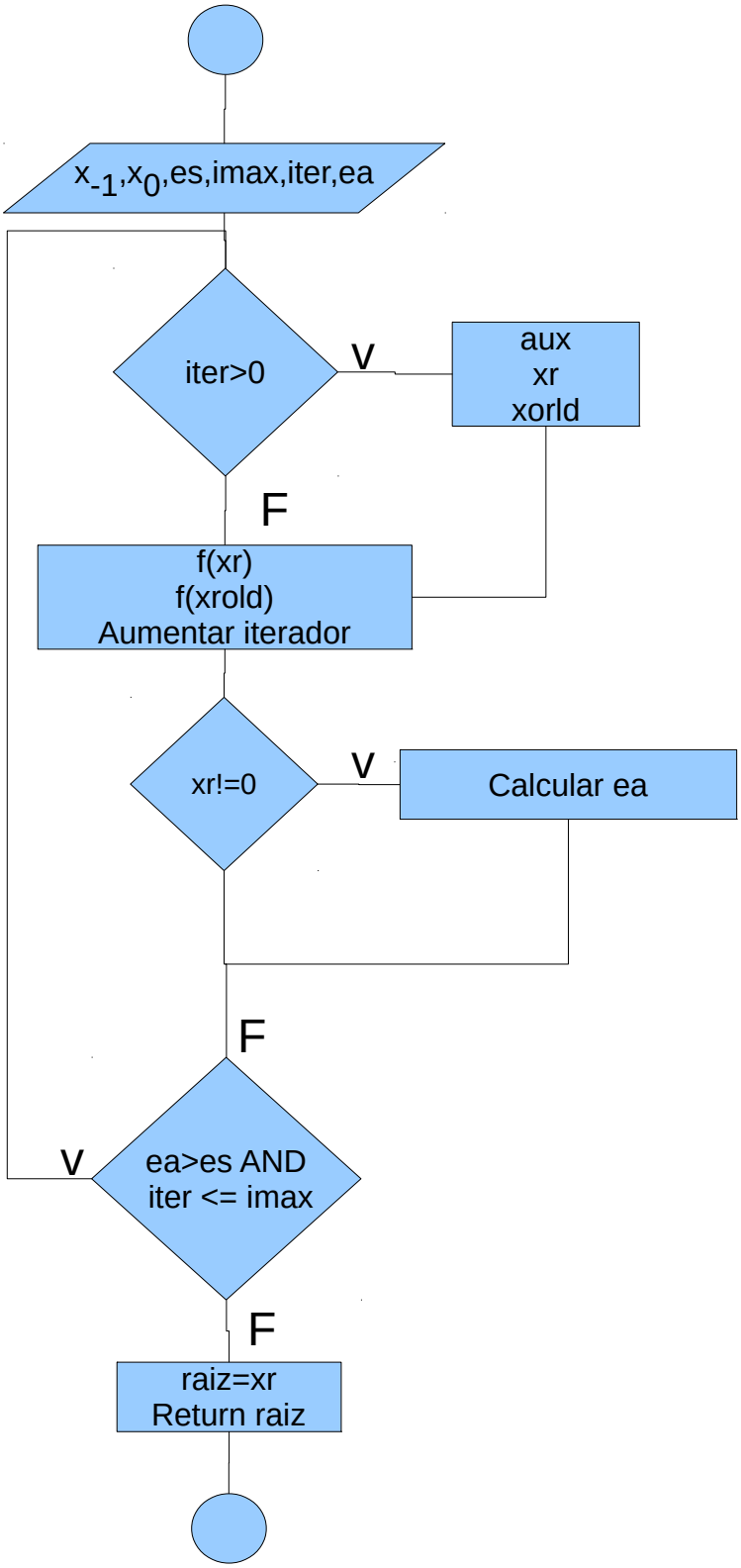
IF iter>0 THEN
aux=xr
$xr=x_i - f(x_i).(x_{i-1} - x_i) / (f(x_{i-1})-f(x_i))$
xrold=aux
END IF
	Si ya uso los valores X_{-1} y X_0 entonces:
	Guardar el xr en una variable auxiliar
	Calcula Xr; la aproximación a la raíz
	Guarda valor de aux en Xrold
	Caso contrario no hace nada. Termina el IF.

fdexrold=-(2*i1)/pow(xrold,3) + (2*i2)/pow((d-xrold),3)	Calcula f(xrold)
fdexr=-(2*i1)/pow(xr,3) + (2*i2)/pow((d-xr),3);	Calcula f(xr)
iter+=1	incrementa en 1 al iterador (contador de it.)

IF xr!=0 THEN
ea=|(xr-xrold)/xr|.100
END IF
	Verifica Si Xr no es cero
	Si No es cero, calcula el Ea
	Caso contrario no hace nada y termina el IF

IF ea<es OR iter >= imax EXIT END DO	Si se cumple algun criteriode finaliz,Termina Ciclos FIN DEL CICLO
raiz=xr END Secante	Guarda el valor xr obtenido en la variable raiz FIN DE FUNCION

Diagrama de Flujos del Método de la Secante:



5) **Prueba de escritorio**

Método de la secante

Iteración	x_(i-1)	x_i	f_(i-1)	f_i	εa %		
0	3,000000	4,000000	-7,999784	-1,906250	-		
1	4,000000	4,312832	-1,906250	-0,767868	7,2535%		
2	4,312832	4,523845	-0,767868	-0,069720	4,6645%		
3	4,523845	4,544918	-0,069720	-0,001730	0,4637%		
4	4,544918	4,545454	-0,001730	-0,000003	0,0118%		
5	4,545454	4,545455	-0,000003	0,000000	0,0000%	< 0.5 x 10^(-2) =	0,01

x = 4,5455

Se observa en la 4º iteración que $|E_a| < E_s$, esto es, 0,0118% < 0,01% para 4 cifras significativas. Luego el programa terminará en esta iteración.

La raíz buscada para los valores del problema, es x=4,54.

5) Código fuente

PYTHON

Laboratorio MN en Python.py

```
def Secante (i1,i2,imax,d):

 #valores iniciales
 es=0.01
 xant=3
 x0=4
 xrold=xant
 xr=x0
 iter=0
 ea=1
 while (ea>es and iter<=imax):
 #Calcular datos de la tabla
  if iter>0:
   #Hago una copia del xr.
   aux=xr
   #Calculo el nuevo xr. Aproximacion a la raiz
   xr=xr-(fdexr*(xrold-xr))/(fdexrold-fdexr)
   #Asigno a xrold, el anteultimo valor que tuvo xr
   xrold=aux
  fdexrold=-(2*i1)/pow(xrold,3) + (2*i2)/pow((d-xrold),3)
  fdexr=-(2*i1)/pow(xr,3) + (2*i2)/pow((d-xr),3)
  if (xr!=0):
   ea=abs(float(xr-xrold)/xr)*100
  #Aumento iterador. Numero de iteraciones = Numero de xr calculados
  iter+=1
 #devuelvo valor a la funcion principal
 return xr
```

C++

Laboratorio MN en c++.cpp

```
float Secante (float i1,float i2,int imax,float d)
{
 float es,ea,xrold,xrold2,fdexr,fdexrold,xr,xant,x0,ea2;
 int iter;

//valores iniciales
 es=0.01;
 ant=3;
 x0=4;
 xrold=xant;
 xr=x0;
 iter=0;
 do{
        //Calcular datos de la tabla
        if (iter>0)
        {
              //Hago una copia del xr.
             aux=xr;
             //Calculo el nuevo xr. Aproximacion a la raiz
             xr=xr-(fdexr*(xrold-xr))/(fdexrold-fdexr);
            //Asigno a xrold, el anteultimo valor que tuvo xr
             xrold=aux;
        }
        fdexrold=-(2*i1)/pow(xrold,3) + (2*i2)/pow((d-xrold),3);
        fdexr=-(2*i1)/pow(xr,3) + (2*i2)/pow((d-xr),3);
        if (xr!=0)
         ea=abs((xr-xrold)/xr)*100;

        //Aumento iterador. Numero de iteraciones = Numero de xr calculados
        iter+=1;
}
while (ea>es && iter<=imax);
//fin del ciclo
```

En los archivos se incluyen los codigos fuentes de C++ y phyton, los cuales permiten elegir entre los métodos de Bisección, Newton-Raphson y Secante, para poder contrastar los resultados. Tambien se expone una Prueba de escritorio para cada uno, así como una gráfica aproximada de la función con su raíz calculada.

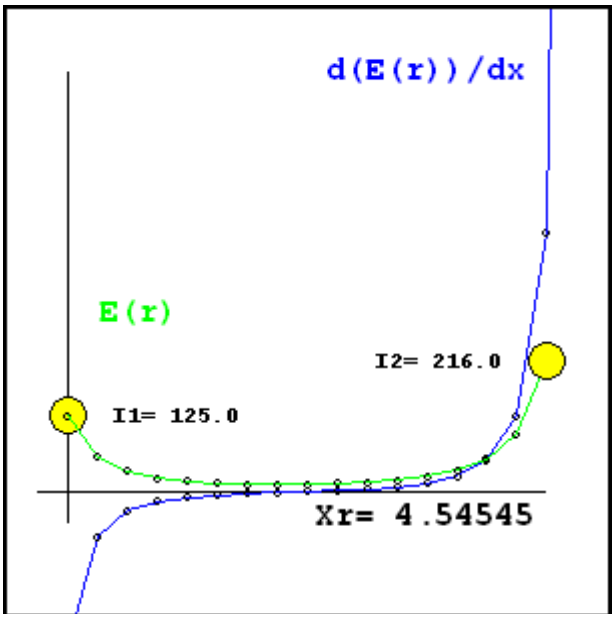
6) Evidencia de Resultados de Ejecución contrastada con prueba de escritorio.

Ingresando los datos del problema y eligiendo el método de la secante se obtuvieron los siguientes resultados:

PYTHON

Laboratorio MN en Python.py

```
Programa para calcular el punto de mínima luminancia entre dos fuentes de luz
============================
Ingrese I1: 125
Ingrese I2: 216
Ingrese Distancia: 10
Que método desea utilizar?:
 Opción 1: M. de la Bisección
 Opción 2: M. de Newton-Rapshon
 Opción 3: M. de la Secante
Deseo utilizar la opción: 3
PRUEBA DE ESCRITORIO    //es= 0.01
====================
iter  |  xrold   |    xr     |  fdexrold  |  fdexr     |  ea    |
-----------------------------------------------------------
0.0   |   3.0   |   4.0     |   -8.0     |   -1.906  |   25.0   |
1.0   |   4.0   |   4.313  |   -1.906   |   -0.768  |   7.25351      |
2.0   |   4.313 |   4.524 |   -0.768  |   -0.07    |   4.66447      |
3.0   |   4.524 |   4.545 |   -0.07    |   -0.002  |   0.46366      |
4.0   |   4.545 |   4.545 |   -0.002  |   0.0      |   0.01179      |
5.0   |   4.545 |   4.545 |   0.0      |   0.0      |   2e-005       |
El punto buscado es:  4.545
El número de iteraciones es: 5
```



C++

Laboratorio MN en c++.cpp

```
Software para el calculo del punto minimo de luminancia:
=========================================================

Ingrese I1: 125
Ahora I2: 216
Distancia de separacion en metros de las fuentes: 10

Que metodo desea utilizar?:
 Opcion 1: M. de la Biseccion
 Opcion 2: M. de Newton-Rapshon
 Opcion 3: M. de la Secante
Deseo utilizar la opcion: 3


PRUEBA DE ESCRITORIO //es=0.01
======================
iter|  xrold  |    xr    |  fdexrold |  fdexr    |   ea         |
-----------------------------------------------------------------------
0   | 3.0000  |  4.0000  |  -7.9998  |  -1.9063  |  25.0000     |
1   | 4.0000  |  4.3128  |  -1.9063  |  -0.7679  |  7.2535      |
2   | 4.3128  |  4.5238  |  -0.7679  |  -0.0697  |  4.6645      |
3   | 4.5238  |  4.5449  |  -0.0697  |  -0.0017  |  0.4637      |
4   | 4.5449  |  4.5455  |  -0.0017  |  -0.0000  |  0.0118      |
```

```
5   | 4.5455  |  4.5455  |  -0.0000  |  -0.0000  |  0.0000      |


---------------------------------
Numero de iteraciones: 5
---------------------------------

---------------------------------
El punto buscado es: 4.5455
---------------------------------

GRAFICA DE E(r)
==============================
 E(r)||             |
     | \            |
     |  \           |
     |   \_________/
    _|_____________________r
     |       xr=4.5455


GRAFICA DE LA DERIVADA DE E(r)
==============================
E'(r)|              |
     |              |
     |              |
     |         ____/
    _|________/_______r
     |    ___/xr=4.5455
         /
        /
Presione una tecla para continuar . . .
```

Tanto en la prueba de escritorio del punto 5) como las obtenidas en los resultados de ambos lenguajes, coinciden en que en la 5ta. Iteración el proceso finaliza, debido a que Ea<Es.

Comparándolo con los resultados de otros métodos expuestos en estos programas, se nota claramente la diferencia en el número de iteraciones entre estos. El método de la Biseccion emplea casi mas del triple (16) de iteraciones que Secante, en tanto que Newton-Raphson emplea la mitad (3) que Secante.

7) Conclusiones.

Colocar conclusiones sobre el la resolución del problema explicando como todo el infome mantiene coherencia.
Se ha logrado...

Y una opinion personal sobre el practico.
Práctico: Motivador.