UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN

FACULTAD DE CIENCIAS FÍSICO-MATEMÁTICAS

José Alfredo Gallegos Chavarría (1383375)

Iteración de Punto Fijo

SAN NICOLÁS DE LOS GARZA, 16 DE MARZO DE 2012

Análisis Numérico  
Iteración de Punto Fijo

**Introducción**

El método de Iteración de Punto Fijo (también conocido como el *método de aproximaciones sucesivas*) es un método usable para obtener una raíz para una determinada ecuación f(x) = 0 no necesariamente lineal, donde f(x) es una función real sobre x y siempre y cuando se cumplan algunas reglas de convergencia. El método de Punto Fijo realiza cálculos iterativos para encontrar un “punto fijo” en la gráfica de la ecuación (básicamente una intersección entre gráficas, como veremos más a detalle dentro de poco), para lo cual utiliza una ligera transformación del polinomio original (pasar de f(x) = 0 a g(x) = x) y una aproximación inicial dada o “tanteada” del valor del polinomio al evaluarlo en el punto fijo.

Definimos, para este método, un punto fijo, para una cierta función g, como un valor determinado a, para el cual se cumple que **a = g(a)** (de ahí la necesidad de transformar f(x) = 0 a g(x) = x).

Por ejemplo, en la imagen anterior tenemos trazadas las gráficas de **y = x** y **y = x2**. Dese cuenta como en los puntos (0, 0), y (1, 1) tenemos casos en que se cumple, para ambas gráficas que x = f(x), si consideramos cada miembro por separado como una función. Estos pueden ser considerados puntos fijos, y el análisis del mismo nos puede ayudar a encontrar raíces (como en el caso de (0, 0), en donde tenemos una intersección con el eje X) para polinomios de grados distintos a meramente primer y segundo grado.

La existencia del dicho para una función en particular se determina por medio del Teorema del Punto Fijo:

* Si…
  + g es continua en [a, b], con a, b ∈ R, y
  + g(x) ∈ [a, b] ∀ x ∈ [a, b]
* entonces **g tiene por lo menos un punto fijo en [a, b]**, y…
* Si además g’(x) existe para todo x ∈ [a, b], y |g’(x)| ≤ K < 1 para todo x ∈ [a, b] con K constante…
* entonces **g tiene un *único* punto fijo x** ∈ **[a, b]**

En este tratado explicaré algunos datos y haré observaciones sobre el método de iteración de punto fijo, e igualmente presentaré un programa en C++ que permite encontrar raíces de polinomios por medio del método, del cual usted debió haber recibido una copia física. Se explican también las decisiones de diseño y lógica del programa.

**Notas Históricas**

El método ya era conocido por varias civilizaciones antiguas, como la babilónica (año 1700 a. C.) o la arábiga (circa 1400 d. C.). En este último caso, por ejemplo, el eminente Jamshid al-Kashi elaboró una tabla trigonométrica con iteraciones de punto fijo, así como la cultura de Babilonia ya conocía una aproximación adecuadamente exacta al valor de la raíz cuadrada de 2 por medio del método.

**Explicación y algoritmo del método**

El método de Punto Fijo tiene el siguiente algoritmo:

1. Dada una ecuación no necesariamente lineal (grado n) f(x) = 0, para encontrar su raíz real, transforme a la forma:

(Es posible lograr esta transformación de varias maneras: despeje, sumar x en cada miembro de la ecuación, u otras operaciones elementales)

1. Después, dada una aproximación de la raíz inicial x0, obtenida por tanteo, obtenemos g(x0). **x0 debe cumplir el criterio de convergencia**:

El resultado de dicha operación es x1:

para i = 0, 1, 2…

1. Realice cálculos sucesivos como sea deseado para obtener más aproximaciones del punto fijo. Dichos cálculos pueden resultar en dos casos:
   1. xi+1 = xi, en cuyo caso ya hemos encontrado la raíz de la ecuación, o
   2. xi+1 ≠ xi en cuyo caso se realiza una nueva iteración del proceso usando el resultado de xi+1 hasta que sea el primer caso.

Podemos introducir la noción de un error aproximado entre iteraciones del método, que corresponde con la forma tradicional del error que hemos usado para otros métodos numéricos en casos anteriores:

El cual es un porcentaje el cual podemos usar como un margen de error; considerar que si la evaluación de un xi determinado se encuentra en el rango de la tolerancia deseada de error en el cálculo se ha llegado o no a una solución aceptable.

**Análisis programático**

He programado una aplicación pequeña que permite ver el método en acción, lo cual realicé usando el lenguaje de programación C/C++ debido a su versatilidad, increíble madurez y soporte de terceros tanto en el lenguaje, su sintaxis y sus recursos y librerías, gran portabilidad y por el hecho de que es el lenguaje que me resulta favorito. Aparte, pretendo llevar uniformidad con los trabajos presentados anteriormente en el curso en lo que consta a estilo de programación y lenguaje elegido.

El producto presentado es completamente de mi diseño y edición; no obstante un documento disponible en el internet sobre el método de iteración de Punto Fijo incluye una implementación en Javascript, que me resultó de inspiración al hacer mi programa, si bien no es código abierto. El documento es de la autoría del maestro Tomeu Barceló del Departamento de Matemáticas de la Universidad Autónoma de Madrid, y se incluye como referencia al final de mi tratado.

Se ha limitado otra vez el grado máximo del polinomio posible de tratar en el programa a 5 debido a que casos superiores pueden resultar de excesiva complejidad computacional, debido a las operaciones recursivas hechas y al número de datos manejados. Por una similar razón, se ha limitado el número máximo de iteraciones del programa a 75, para evitar procesos demasiado largos. Igualmente, el programa se ha diseñado alrededor de la solución de raíces de polinomios en términos de potencias de x, debido a la complejidad alrededor de implementar un parseador de texto de la entrada estándar en nuestro programa, que está en una aplicación de consola, para manejar términos exponenciales/logarítmicos/trigonométricos de x y el hecho de que requeriría la implementación de funcionalidades de librerías de terceros.

El algoritmo utilizado en el programa, descrito en lenguaje natural, pero con orden obviamente lógico (no siguiendo completamente el esquema de seudocódigo, y no a completa profundidad excepto en las partes relevantes al cálculo matemático, son):

* Captura el grado del polinomio a entrar
* Captura el error deseado
* Captura los coeficientes de la ecuación a resolver en forma canónica f(x) = 0
* Después de obtener esta ecuación calcula la derivada y guárdala
* Pide una aproximación inicial…
  + Si la derivada evaluada en la aproximación inicial es ≥ 1, pide otro dato, no pases mientras no sea menor a 1.
  + Si la derivada evaluada en la aproximación inicial es < 1, procede al siguiente paso…
* Al obtener una aproximación inicial adecuada, empieza con iteraciones de punto fijo… haciendo ciclos, de una hasta setenta y cinco veces… (Primer valor: aprox inicial. Guardar este valor como previo)
  + - Siguiente valor: evaluar la función en el valor previo. Guardar este valor Siguiente.
    - Calcular el error como valor absoluto de: (siguiente – previo / siguiente)
    - Comparar: ¿es igual el valor siguiente con el valor previo? o

¿el valor está en el margen de error?

* + - Sí: esta es la raíz, romper el ciclo, presentarla y terminar
    - No: simplemente continua los ciclos guardando el siguiente como el previo
  + Imprime una columna de la tabla: valor previo, valor siguiente, error

**Corrida de prueba del programa**

Intentaremos dos casos de prueba del programa, tanto convergiendo como sin hacerlo…

Resolucion de raices de ecuaciones

Metodo de Iteracion de Punto Fijo

Jose Alfredo Gallegos Chavarria

(Marzo 16, 2012)

Grado Maximo del Polinomio: 3

Entrar margen de tolerancia (+-valor): 0.05

Entre la ecuacion a resolver en forma canonica f(x) = 0,

por ejemplo, para la ecuacion...

2x^3 - 2x^2 + x = 1

...ud. entraria los siguientes datos con tabulacion entre coeficientes

y un salto de linea al final...

2 -2 1 -1

>> 2 -2 1 -1

f(x) = 0 2.0000 -2.0000 1.0000 -1.0000  
 g(x) = x 2.0000 -2.0000 2.0000 -1.0000  
 g'(x) 0.0000 6.0000 -4.0000 2.0000

Entrar aproximacion inicial x0 para obtener g'(x0): 0

g'(0.0000) = 2.0000

Valor inadecuado, vuelva a entrar un valor...

Entrar aproximacion inicial x0 para obtener g'(x0): 1

g'(1.0000) = 4.0000

Valor inadecuado, vuelva a entrar un valor...

Entrar aproximacion inicial x0 para obtener g'(x0): -1

g'(-1.0000) = 12.0000

Valor inadecuado, vuelva a entrar un valor...

Presione una tecla para continuar. . .

Resolucion de raices de ecuaciones

Metodo de Iteracion de Punto Fijo

Jose Alfredo Gallegos Chavarria

(Marzo 16, 2012)

Grado Maximo del Polinomio: 3

Entrar error (porcentaje): 0.05

Entre la ecuacion a resolver en forma canonica f(x) = 0,

por ejemplo, para la ecuacion...

2x^3 - 2x^2 + x = 1

...ud. entraria los siguientes datos con tabulacion entre coeficientes

y un salto de linea al final...

2 -2 1 -1

>> 1 -5 7 -3

f(x) = 0 1.0000 -5.0000 7.0000 -3.0000  
 g(x) = x 1.0000 -5.0000 8.0000 -3.0000  
 g'(x) 0.0000 3.0000 -10.0000 8.0000

Entrar aproximacion inicial x0 para obtener g'(x0): 0

g'(0.0000) = 8.0000

Valor inadecuado, vuelva a entrar un valor...

Entrar aproximacion inicial x0 para obtener g'(x0): 1

g'(1.0000) = 1.0000

Valor adecuado para resolucion por el metodo: g'(1.0000) < 1

Solucion por Metodo de Punto fijo con e = 0.0500

i p f(p) e  
 ------------------------------  
 1 1.0000 0.0000

Raiz de la ecuacion: 1.0000

Press any key to continue . . .

**Notas sobre el producto entregado**

Junto con este tratado, usted debió haber recibido un CD con una versión del programa el cual contiene los siguientes archivos:

* **puntofijo.docx**
  + *MD5:* *e5216158f02be1bb2a0ba51d80ee8d3b*
  + copia digital de este documento, en formato .docx (MS Office Word 2007 en adelante).
* **puntofijo.cbp**
  + *MD5:* *456bc8399f93b2987fb2c24971c8d1a5*
  + archivo de projecto (estructura de qué sources y qué configuración del compilador/linker deben ser usadas para compilar el source, entre otras configuraciones) compatible con la IDE Code::Blocks (la cual utiliza el port de Windows del compilador MinGW para compilar programas en C junto con la vinculación necesaria. MinGW es un port de compiladores de C para GNU/Linux como g++ o gcc, aún cuando mi código incluye la llamada a función system(“PAUSE”); la cual no es portable para esos ambientes de cómputo. no obstante, el código debe ser compatible con otros compiladores en la misma plataforma, y con otros compiladores con ligeros ajustes a las llamadas de funciones).
* **main.cpp** 
  + *MD5:* *3a0de8b9950ce27af82994db56886860*
  + código fuente del programa en lenguaje C++
* **puntofijo.exe**
  + *MD5:* *9703f22841391f3a0e0c9eb172b82832*
  + programa compilado bajo MinGW en formato ejecutable de Windows, compatible con cualquier implementación del kernel NT para arquitecturas de procesador x86 o x86 – 64.
* **main.o**
  + *MD5:* *d0c56e8dad5c5fffc2053c1663653beb*
  + código objeto proveniente de la compilación de main.cpp, antes de la vinculación.

(Por favor use su herramienta favorita de comparación/verificación de sumas MD5 para validar la integridad de estos archivos en la copia recibida).

El programa ha sido probado exitosamente en un sistema bajo la siguiente configuración:

* SO: Windows 7 Ultimate SP1 (build 7601) 64 bit
* Procesador: Intel Celeron E3400 (doble núcleo) @ 2.6 GHz
* Memoria Principal: 2 GB DDR2
* Memoria Secundaria: disco duro magnético SATA-II
* Unidad de lectura de medios ópticos

No debería haber problemas ejecutando el programa en configuraciones similares.

**Conclusión**

Esto concluye mi tratado sobre el método de iteración de Punto Fijo. La actividad práctica que me ha permitido este trabajo me ha ayudado a ver más de cerca la relación entre métodos cerrados y abiertos, como fue explicado en clase, debido a que entre varios de los ejemplos que intentaba hacer de prueba había muchas más funciones que divergían antes que convergir. No obstante, cuando funciona, o cuando se está en las condiciones adecuadas, el método de punto fijo, así como otros métodos similares, resultaron increíblemente rápidos al encontrar soluciones de las ecuaciones.

**Recursos**

* http://www.uam.es/bartolome.barcelo