Método del punto fijo

José Ramón Pérez Navarro

September, 2019

0.1. Definición

El método del punto fijo es fácil de usar y se aplica a una amplia variedad de problemas. En su forma más simple, la ecuación que se va a iterar se obtiene reagrupando la ecuación que contiene x en el lado izquierdo de la ecuación. Una aproximación a x se inserta en el lado derecho; así se calcula un nuevo valor de x. El nuevo valor de x se usa en el cálculo para dar más valores de x, y el proceso se repite en forma iterativa. Si el punto inicial y la reordenación de la ecuación son adecuados, los valores se aproximan cada vez más a la solución verdadera. En este caso se puede decir que el método es convergente.

Teorema

Sea g una función continua en el intervalo [a,b] con a ≤g(x)≤b.Suponiendo que g' es continua en [a,b] y que existe una constante 1>k>0 tal que:

$|g'(x)|$ ≤K para toda x en $[a, b]$

Si $g'(P) \neq 0$, entonces para cualquier $p_0 \in [a, b]$,la sucesión generada por

$$X_{n+1} = g(x_n).n = 1, 2...$$

El método de punto fijo consta basicamente en despejar la variable de la función, de esa forma la variable queda en función de sí misma. Debido a esto se necesita una prueba de convergencia para garantizar que desde el punto inicial se converge a la solución deseada. A continuación mostrare un código en Matlab de este método.

```
%  Programa principal para determinar los cruces por cero de una función continua cle
%  fx = 1 + 2.*x - 3.*x.^2.*exp(-x) + 2.*x.^3.*sin(x).*exp(-x./5) dentro del
%  intervalo [4,20]. Utilizar un error relativo de 1e-6. r=0;
% Inicia contador de cruces por cero. % Ciclo para calcular todos los cruces por cero
% Evaluación de la función en el punto k.    gk = 1 + 2.*k - 3.*k.^2.*exp(-k) + 2.*k.
% Evaluación de la función en el punto l=k+1.    gl = 1 + 2.*l - 3.*l.^2.*exp(-l) + 2
```

```
% Condicional que marca el subintervalo donde se encuentra un cruce por cero.    if g
% Método de punto fijo.         [Cero,Mat] = PuntoFijo(k);         r = r + 1;
% Contador de cruces por cero.        Cruce(r) = Cero;
% Almacena los cruces por cero.        dr = length(Mat(:,1));  M(r,1:dr,1:dc) = Mat;
% Matriz que almacena todas las iteraciones de                                  % tod
% Iteraciones del primer cruce por cero. M2(:,:) = M(2,:,:);
% Iteraciones del segundo cruce por cero. M3(:,:) = M(3,:,:);
% Iteraciones del tercer cruce por cero. M4(:,:) = M(4,:,:);
% Iteraciones del cuarto cruce por cero. M5(:,:) = M(5,:,:);
% Iteraciones del quinto cruce por cero.
```

Función llamada punto fijo

```
%  Función del método Punto fijo para cálculo de cruces por cero de una función no %
% %  El método calcula un cruce por cero. % %  La función se llama de la siguiente ma
% %               [Cero,Mat]=Punto fijo(a).
% %  Entradas:
%              a  -- Punto inicial del método.
% %  Salida:
%         Cero -- Valor de la variable para la cual la magnitud de la función es cero
%         o menor a una tolerancia especificada previamente.
%         Mat  -- Vector que contiene todas las iteraciones hasta convergencia.
% function[Cero,Mat] = PuntoFijo(a); Err = 1;
% Inicializa el error para ingresar al ciclo iterativo. tol = 1e-6;
% Tolerancia especificada para la convergencia. d = round(a/pi);
% Numero de pi enteros de acuerdo a la condición inicial. e = (-1)^d;
% Signo de acuerdo si es par o impar en # de pi.
% Valor de la función en el punto inicial. fa = 1 + 2.*a - 3.*a.^2.*exp(-a) + 2.*a.^3
% Contador de iteraciones. c   = 1;
% Inicia la matriz que almacena cada iteración. Mat(c,:) = [a fa]; while Err > tol &
% Cálculo del punto de aproximación que determina el método de punto fijo.   gx = d*p
% Valor de la función en el nuevo punto.   fgx = 1 + 2.*gx - 3.*gx.^2.*exp(-gx) + 2.*
% Contador de iteraciones para no dejar en un ciclo el programa en caso de alguna % i
% Matriz que almacena los resultados de cada iteración.   Mat(c,:) = [gx fgx];   % El
% Criterio de error.   Err = abs(fgx); end
% Cruce por cero que determina el método de la secante. Cero = gx;
```