Sistemas de Ecuaciones No Lineales -SENL-

Ing. Diego Ezcurra

1. Introducción

Los métodos para resolver sistemas de ecuaciones no lineales (SENL) funcionan de manera análoga a los métodos para resolver ENL y también en forma similar a los métodos iterativos para resolver SEL. Puede pensarse a estos métodos como una generalización de los anteriores, en el primer caso, aumentando la dimensión, y en el segundo, incorporando no linealidades.

Veremos un ejemplo que lo resolveremos por 3 esquemas: simil Jacobi o Gauss Seidel, Newton Raphson y Cuasi Newton.

2. Ejemplo

Vamos a resolver el siguiente SENL

$$\frac{x^2}{25} = 1 - \frac{y^2}{4}$$

$$2x + 5y = 10$$

Para ello, aplicamos el mismo concepto que en 1 variable, con un vector incógnita X = (x ; y)

Si despejamos la variable *i* de la *i-ésima* ecuación, obtenemos una función de iteración G(X), siendo G = (g_1, g_2)

$$\boldsymbol{g1(X)} = \sqrt[2]{\frac{\boldsymbol{1 - y^2}}{\boldsymbol{4}} * \boldsymbol{25}}$$

$$\boldsymbol{g2(X)} = \frac{\boldsymbol{10 - 2x}}{\boldsymbol{5}}$$

Si aplicamos los conceptos de los métodos de Jacobi y Gauss Seidel, obtenemos nuestras expresiones iterativas:

Simil Jacobi:

$$\boldsymbol{x_{k+1}} = \sqrt[2]{\frac{\boldsymbol{1 - y_k^2}}{\boldsymbol{4}} * \boldsymbol{25}}$$

$$\boldsymbol{y_{k+1}} = \frac{\boldsymbol{10 - 2x_k}}{\boldsymbol{5}}$$

Simil Gauss Seidel:

$$\boldsymbol{x_{k+1}} = \sqrt[2]{\frac{\boldsymbol{1 - y_k^2}}{\boldsymbol{4}} * \boldsymbol{25}}$$

$$\boldsymbol{y_{k+1}} = \frac{\boldsymbol{10 - 2x_{k+1}}}{\boldsymbol{5}}$$

En ambos casos se deben verificar las condiciones de convergencia de punto fijo ya vistas para ENL (existencia y unicidad). En este caso, dichas condiciones se refieren a una dimensión 2, es decir:

-Existencia: g_1 y g_2 $\in$ [a;b]x[a;b] para todo X:(x;y) $\in$ [a;b]x[a;b]

-Unicidad: $|g_1'|$ y $|g_2'|<1$ para todo X:(x;y) $\in$ [a;b]x[a;b]
Notar que aparecen las derivadas parciales:

$$\frac{\partial g_1}{\partial x}\ ;\ \frac{\partial g_1}{\partial y}\ ;\ \frac{\partial g_2}{\partial x}\ ;\ \frac{\partial g_2}{\partial y}$$

En este caso planteado, no hay un intervalo de convergencia válido (pueden probar experimentalmente con distintas semillas).

Método Newton

Este método intenta acelerar la convergencia, utilizando la idea del máximo descenso.

$$\bar{G}(\bar{x}) = \bar{x} - \frac{\bar{F}(\bar{x})}{\bar{F}'(\bar{x})} = \bar{x} - \bar{\bar{J}}(\bar{x})^{-1}\bar{F}(\bar{x})$$

Matriz Jacobiana

$$J_{ij} = \left|\frac{\partial f_i}{\partial x_j}\right| J_{ij} =$$

La forma iterativa queda de la siguiente manera:

$$\bar{X}_{k+1} = \bar{X}_k - \bar{\bar{J}}(\bar{X}_k)^{-1}\bar{F}(\bar{x}_k)$$

Para resolverlo, puede verse que se necesita calcular la inversa de la matriz Jacobiana, lo que no resulta ágil computacionalmente. Por ello, se propone resolverlo mediante la siguiente expresión:

$$\boldsymbol{\bar{\bar{J}}(\bar{X}_k) * (\bar{X}_{k+1} - \bar{X}_k) = -\bar{F}(\bar{x}_k)}$$

Esto resulta mas sencillo pero de todas maneras se debe resolver un SEL en cada iteración.

Apliquemos la expresión al problema anterior:

$$f_1(x,y) = \frac{x^2}{25} + \frac{y^2}{4} - 1$$

$$f_2(x,y) = 2x + 5y - 10$$

Por lo tanto, obtenemos la matriz jacobiana:

$$\bar{\bar{J}}(x,y) = \begin{bmatrix} \frac{\partial f_1}{\partial x} & \frac{\partial f_1}{\partial y} \\ \frac{\partial f_2}{\partial x} & \frac{\partial f_2}{\partial y} \end{bmatrix} = \begin{bmatrix} \frac{2}{25}x & \frac{1}{2}y \\ 2 & 5 \end{bmatrix}$$

Armamos el esquema iterativo:

$$\begin{bmatrix} \frac{2x_k}{25} & \frac{y_k}{2} \\ 2 & 5 \end{bmatrix} * \left(\begin{bmatrix} x_{k+1} \\ y_{k+1} \end{bmatrix} - \begin{bmatrix} x_k \\ y_k \end{bmatrix} \right) = - \begin{bmatrix} f_1(x_k; y_k) \\ f_2(x_k; y_k) \end{bmatrix}$$

Aplicamos para k = 0

$$\begin{bmatrix} \frac{2x_0}{25} & \frac{y_0}{2} \\ 2 & 5 \end{bmatrix} * \left(\begin{bmatrix} x_1 \\ y_1 \end{bmatrix} - \begin{bmatrix} x_0 \\ y_0 \end{bmatrix} \right) = - \begin{bmatrix} f_1(x_0; y_0) \\ f_2(x_0; y_0) \end{bmatrix}$$

Si utilizamos una semilla (2 ; 2)

$$\begin{bmatrix} \frac{4}{25} & \frac{1}{2} \\ 2 & 5 \end{bmatrix} * \Delta X_1 = - \begin{bmatrix} 0{,}16 \\ 4 \end{bmatrix}$$

Queda armado el SEL para resolver en la primera iteración:

$\begin{bmatrix} 0{,}16 & 0{,}5 \\ 2 & 5 \end{bmatrix} * \Delta X_1 = - \begin{bmatrix} 0{,}16 \\ 4 \end{bmatrix}$ => resolviendo: (Δx; Δy) = (-2,66 ; 0,266)

Los valores obtenidos corresponden al vector DELTA. Debemos hallar el vector (x;y)

Por lo tanto, $X_{k+1} = \Delta X_{k+1} + X_k$

=> $$\begin{bmatrix} x_1 \\ y_1 \end{bmatrix} = \begin{bmatrix} -2{,}66 \\ 0{,}266 \end{bmatrix} + \begin{bmatrix} 2 \\ 2 \end{bmatrix} = \begin{bmatrix} -0{,}66 \\ 2{,}26 \end{bmatrix}$$

Si repetimos el procedimiento iterativamente, obtenemos lo siguiente

k	x_k	y_k	$\Delta x=\lvert x_k-x_{k-1}\rvert$	$\Delta y=\lvert y_k-y_{k-1}\rvert$	$\Delta x/x_k$	$\Delta y/y_k$	p_1	p_2
0	2	2						
1	-0,66667	2,26667	2,67	0,267	38	0,13		
2	-0,07018	2,02807	0,597	0,239	623	0,12		
3	-0,00096	2,00038	0,0692	0,0277	377222	0,014	1,42	19,4
4	-1,8E-07	2	0,00096	0,00038	1,3E+11	0,00019	1,98	1,98
5	-7E-15	2	1,8E-07	7,3E-08	49,9E+7	3,7E-08	1,99	1,99
6	-3,7E-16	2	6,7E-15	2,7E-15	18,12	1,3E-15	2,00	2,00
7	-3,7E-16	2	0	0	0	0		

Podemos ver que el método logra convergencia cuadrática, es decir, es de orden 2.

Método Cuasi-Newton

Este método relaja la necesidad de contar con la derivada y la reemplaza por su expresión en diferencias, similar a lo que ocurría con el método de la secante.

$$\frac{\partial F_i(x_k)}{\partial x_j} \approx \frac{F_i\left(x_k + he_j\right) - F_i(x_k)}{h}$$

Donde $e_j = (0;\ldots..0;1;0\ldots\ldots0)$, indicando la posición a perturbar dentro del vector.

Asi, las cuatro derivadas parciales del ejemplo, quedan aproximadas de la siguiente manera:

$$\frac{\partial f_1(x_k, y_k)}{\partial x} \approx \frac{f_1(x_k + h\,;\, y_k) - f_1(x_k; y_k)}{h}$$

$$\frac{\partial f_1(x_k, y_k)}{\partial y} \approx \frac{f_1(x_k\,;\, y_k + h) - f_1(x_k; y_k)}{h}$$

$$\frac{\partial f_2(x_k, y_k)}{\partial x} \approx \frac{f_2(x_k + h\,;\, y_k) - f_2(x_k; y_k)}{h}$$

$$\frac{\partial f_2(x_k, y_k)}{\partial y} \approx \frac{f_2(x_k\,;\, y_k + h) - f_2(x_k; y_k)}{h}$$

En el ejemplo, para f1:

$$\frac{\partial f_1(x_k, y_k)}{\partial x} \approx \frac{\left[\frac{(x+h)^2}{25} + \frac{y^2}{4} - 1\right] - \left[\frac{x^2}{25} + \frac{y^2}{4} - 1\right]}{h}$$

$$\frac{\partial f_1(x_k, y_k)}{\partial y} \approx \frac{\left[\frac{x^2}{25} + \frac{(y+h)^2}{4} - 1\right] - \left[\frac{x^2}{25} + \frac{y^2}{4} - 1\right]}{h}$$

En el ejemplo, para f_2:

$$\frac{\partial f_2(x_k, y_k)}{\partial x} \approx \frac{[2(x+h) + 5y - 10] - [2x + 5y - 10]}{h}$$

$$\frac{\partial f_2(x_k, y_k)}{\partial y} \approx \frac{[2x + 5(y+h) - 10] - [2x + 5y - 10]}{h}$$

La consecuencia directa es que se pierde el orden 2.

Otra alternativa para reducir el esfuerzo de cómputo del método de Newton consiste en “fijar” la matriz jacobiana durante una determinada cantidad de iteraciones y luego actualizarla. De esta manera, en lugar de generar un nuevo SEL en cada iteración, la matriz del sistema se mantiene inalterada y solo cambia el vector de términos independientes. Luego de una serie de iteraciones, se actualiza dicha matriz y se vuelve a iterar con esa matriz actualizada, sin modificarla, por otra cantidad determinada de iteraciones. Claramente se pierde la convergencia cuadrática en este caso también.