

## SISTEMAS NO LINEALES

### Metodo Iterativo

Sean  $\mathbf{F}: f_1, f_2, \dots, f_n$  sistema de ecuaciones no lineales con variables  $\mathbf{X}: x_1, x_2, \dots, x_n$ . Se requiere calcular un vector real que satisfaga al sistema  $\mathbf{F}$

En el caso de que  $\mathbf{F}$  contenga una sola ecuación  $f$  con una variable  $x$ , la conocida fórmula iterativa de Newton puede escribirse de la siguiente manera:

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - \left( \frac{df^{(k)}}{dx} \right)^{-1} f^{(k)}, \quad k=0, 1, 2, \dots \quad (\text{iteraciones})$$

Si  $\mathbf{F}$  contiene  $n$  ecuaciones, la fórmula se puede extender, siempre que las derivadas existan:

$$\mathbf{X}^{(k+1)} = \mathbf{X}^{(k)} - \left( \frac{\partial \mathbf{F}^{(k)}}{\partial \mathbf{X}} \right)^{-1} \mathbf{F}^{(k)} = \mathbf{X}^{(k)} - (\mathbf{J}^{(k)})^{-1} \mathbf{F}^{(k)}$$

En donde:

$$\mathbf{X}^{(k+1)} = \begin{bmatrix} x_1^{(k+1)} \\ x_2^{(k+1)} \\ \dots \\ x_n^{(k+1)} \end{bmatrix}, \quad \mathbf{X}^{(k)} = \begin{bmatrix} x_1^{(k)} \\ x_2^{(k)} \\ \dots \\ x_n^{(k)} \end{bmatrix}, \quad \mathbf{F}^{(k)} = \begin{bmatrix} f_1^{(k)} \\ f_2^{(k)} \\ \dots \\ f_n^{(k)} \end{bmatrix}, \quad \mathbf{J}^{(k)} = \begin{bmatrix} \frac{\partial f_1^{(k)}}{\partial x_1} & \frac{\partial f_1^{(k)}}{\partial x_2} & \dots & \frac{\partial f_1^{(k)}}{\partial x_n} \\ \frac{\partial f_2^{(k)}}{\partial x_1} & \frac{\partial f_2^{(k)}}{\partial x_2} & \dots & \frac{\partial f_2^{(k)}}{\partial x_n} \\ \dots & \dots & \dots & \dots \\ \frac{\partial f_n^{(k)}}{\partial x_1} & \frac{\partial f_n^{(k)}}{\partial x_2} & \dots & \frac{\partial f_n^{(k)}}{\partial x_n} \end{bmatrix}$$

$\mathbf{J}$  es la matriz jacobiana. Esta ecuación de recurrencia se puede usar iterativamente con  $k = 0, 1, 2, \dots$  partiendo de un vector inicial  $\mathbf{X}^{(0)}$  generando vectores de aproximación:  $\mathbf{X}^{(1)}, \mathbf{X}^{(2)}, \mathbf{X}^{(3)}, \dots$

### ALGORITMO

Sea  $\mathbf{F}: f_1, f_2, \dots, f_n$  ecuaciones con variables independientes  $\mathbf{X}: x_1, x_2, \dots, x_n$ .

Ecuación de recurrencia:

$$\mathbf{X}^{(k+1)} = \mathbf{X}^{(k)} - (\mathbf{J}^{(k)})^{-1} \mathbf{F}^{(k)}, \quad k=0, 1, 2, \dots$$

En donde  $\mathbf{J}$  es la matriz jacobiana del sistema

Entrada

- $\mathbf{F}$ : Vector con las ecuaciones
- $\mathbf{X}$ : Vector con las variables independientes
- $\mathbf{U}$ : Vector con valores iniciales para las variables

Salida

- $\mathbf{U}$ : Vector con los nuevos valores calculados para las variables

## DEDUCCION DE LA FORMULA

Sean  $f_1(x_1, x_2) = 0$ ,  $f_2(x_1, x_2) = 0$  dos ecuaciones no-lineales con variables  $x_1, x_2$ .

Sean  $r_1, r_2$  valores reales tales que  $f_1(r_1, r_2) = 0$ ,  $f_2(r_1, r_2) = 0$ , entonces  $(r_1, r_2)$  constituye una raíz real del sistema y es de interés calcularla.

Suponer que  $f_1, f_2$  son funciones diferenciables en alguna región cercana al punto  $(r_1, r_2)$

Con el desarrollo de la serie de Taylor expandimos  $f_1, f_2$  desde el punto  $(x_1^{(k)}, x_2^{(k)})$  al punto  $(x_1^{(k+1)}, x_2^{(k+1)})$

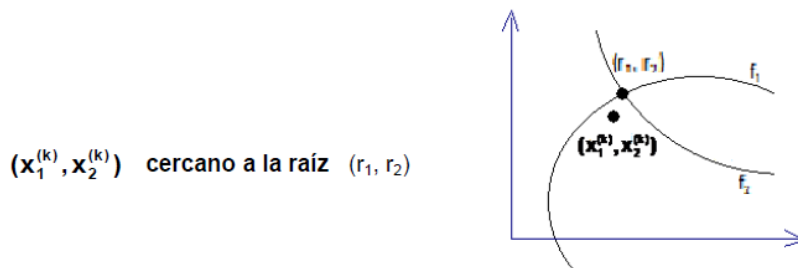
$$f_1^{(k+1)} = f_1^{(k)} + (x_1^{(k+1)} - x_1^{(k)}) \frac{\partial f_1^{(k)}}{\partial x_1} + (x_2^{(k+1)} - x_2^{(k)}) \frac{\partial f_1^{(k)}}{\partial x_2} + O(x_1^{(k+1)} - x_1^{(k)})^2 + O(x_2^{(k+1)} - x_2^{(k)})^2$$

$$f_2^{(k+1)} = f_2^{(k)} + (x_1^{(k+1)} - x_1^{(k)}) \frac{\partial f_2^{(k)}}{\partial x_1} + (x_2^{(k+1)} - x_2^{(k)}) \frac{\partial f_2^{(k)}}{\partial x_2} + O(x_1^{(k+1)} - x_1^{(k)})^2 + O(x_2^{(k+1)} - x_2^{(k)})^2$$

Por simplicidad se ha usado la notación:  $f_1^{(k)} = f_1(x_1^{(k)}, x_2^{(k)})$ ,  $f_1^{(k+1)} = f_1(x_1^{(k+1)}, x_2^{(k+1)})$ , etc.

En los últimos términos de ambos desarrollos se han escrito únicamente los componentes de interés, usando la notación  $O(\quad)$ .

## SUPUESTOS



Si el método converge cuadráticamente entonces  $(x_1^{(k+1)}, x_2^{(k+1)})$  estará muy cercano a  $(r_1, r_2)$

Por lo tanto se puede aproximar:

$$f_1(x_1^{(k+1)}, x_2^{(k+1)}) \approx 0$$

$$f_2(x_1^{(k+1)}, x_2^{(k+1)}) \approx 0$$

Por otra parte, si  $(x_1^{(k)}, x_2^{(k)})$  es cercano a  $(x_1^{(k+1)}, x_2^{(k+1)})$ , las diferencias serán pequeñas y al elevarse al cuadrado se obtendrán valores más pequeños y se los omite.

## ENTONCES

$$0 = f_1^{(k)} + (x_1^{(k+1)} - x_1^{(k)}) \frac{\partial f_1^{(k)}}{\partial x_1} + (x_2^{(k+1)} - x_2^{(k)}) \frac{\partial f_1^{(k)}}{\partial x_2}$$
$$0 = f_2^{(k)} + (x_1^{(k+1)} - x_1^{(k)}) \frac{\partial f_2^{(k)}}{\partial x_1} + (x_2^{(k+1)} - x_2^{(k)}) \frac{\partial f_2^{(k)}}{\partial x_2}$$

En notación matricial:

$$-\mathbf{F}^{(k)} = \mathbf{J}^{(k)} (\mathbf{X}^{(k+1)} - \mathbf{X}^{(k)})$$

Siendo

$$\mathbf{F}^{(k)} = \begin{bmatrix} f_1^{(k)} \\ f_2^{(k)} \end{bmatrix}, \quad \mathbf{X}^{(k)} = \begin{bmatrix} x_1^{(k)} \\ x_2^{(k)} \end{bmatrix}, \quad \mathbf{X}^{(k+1)} = \begin{bmatrix} x_1^{(k+1)} \\ x_2^{(k+1)} \end{bmatrix}, \quad \mathbf{J}^{(k)} = \begin{bmatrix} \frac{\partial f_1^{(k)}}{\partial x_1} & \frac{\partial f_1^{(k)}}{\partial x_2} \\ \frac{\partial f_2^{(k)}}{\partial x_1} & \frac{\partial f_2^{(k)}}{\partial x_2} \end{bmatrix}$$
$$\mathbf{J}^{(k)} \mathbf{X}^{(k+1)} = \mathbf{J}^{(k)} \mathbf{X}^{(k)} - \mathbf{F}^{(k)}$$
$$\mathbf{X}^{(k+1)} = \mathbf{X}^{(k)} - (\mathbf{J}^{(k)})^{-1} \mathbf{F}^{(k)}, \quad |\mathbf{J}^{(k)}| \neq 0$$

Es la ecuación de recurrencia que se puede usar iterativamente con  $k=0, 1, 2, \dots$  partiendo de un vector inicial  $\mathbf{X}^{(0)}$  generando vectores de aproximación:  $\mathbf{X}^{(1)}, \mathbf{X}^{(2)}, \mathbf{X}^{(3)}, \dots$

## CONDICIONES DE CONVERGENCIA

En forma general la convergencia de este método para sistemas no lineales requiere que:

- $f_1, f_2, \dots, f_n$  así como sus derivadas sean continuas en la región de aplicación.
- El determinante del Jacobiano no se anule en esta región
- El valor inicial y los valores calculados pertenezcan a esta región, la cual incluye a la raíz que se intenta calcular

**Nota:** La convergencia será controlada desde la ventana interactiva llamando iterativamente a la función. Por las propiedades de este método, la convergencia o divergencia será muy rápida.

## MEJORAMIENTO

Alternativamente, se puede mejorar la instrumentación incorporando un ciclo con un máximo de iteraciones para que las iteraciones se realicen dentro de la función.

## METODO EN PYTHON

Las derivadas parciales se obtienen con la función **diff** y la sustitución de los valores de **U** en las variables se realiza con la función **subs**. La solución se la obtiene con la inversa de la matriz de las derivadas parciales **J**. Estas funciones están en librerías de Python que deben cargarse.

En su estado actual, la librería SymPy de Python no tiene instrumentado algún método para resolver sistemas no lineales como se puede encontrar en otros lenguajes como MATLAB, sin embargo MATLAB solamente pudo calcular una de las dos raíces del ejemplo anterior.

## METODO EN R

En R hay varios paquetes para resolver sistema no lineales del tipo  $F(x) = 0$  donde  $F: \mathbb{R}^p \rightarrow \mathbb{R}^p$  es una función no lineal con derivadas parciales continuas. Por ejemplo `multiroot`, `rootsolve`, `nleqslv` y `BB`

## EJERCICIOS

1. Encuentre las soluciones del sistema de ecuaciones dado:

$$\sin(x) + e^y - xy = 5$$

$$x^2 + y^3 - 3xy = 7$$

- a) Grafique las ecuaciones en el intervalo  $[-4, 4, -4, 4]$  y observe que hay dos raíces reales. Elija del gráfico, valores aproximados para cada raíz.
- b) Use iterativamente la función `snewton`
- c) Compruebe que las soluciones calculadas satisfacen a las ecuaciones

2. Encuentre las soluciones del sistema de ecuaciones dado:

$$\cos(x+y) + xy = 3$$

$$3(x-2)^2 - 2(y-3)^2 = 5xy$$

- a) Grafique las ecuaciones en el intervalo  $[-6, 6, -6, 6]$  y observe que hay dos raíces reales. Elija del gráfico, valores aproximados para cada raíz.
- b) Use iterativamente la función `snewton`:
- c) Compruebe que las soluciones calculadas satisfacen a las ecuaciones

3. Dada la función  $f(x, y) = (x^2 - y) \sin(2x+y)$  se desea localizar y calcular las coordenadas de un máximo local.

Plantee un sistema de dos ecuaciones no lineales con las derivadas parciales de  $f$  igualadas a cero. Grafique las ecuaciones en el plano  $X, Y$ . Elija un punto inicial y use iterativamente la función `snewton` para calcular una solución.