

Sistemas de ecuaciones no lineales

Método de Newton-Raphson y método de Broyden

Apuntes de clase: formulación, interpretación, algoritmos y ejemplo

1 El problema

Consideremos un sistema de M ecuaciones no lineales con M incógnitas,

$$f_1(x_1, \dots, x_M) = 0, \quad \dots, \quad f_M(x_1, \dots, x_M) = 0.$$

Es conveniente escribirlo en forma vectorial:

$$\mathbf{f}(\mathbf{x}) = \mathbf{0}, \quad \mathbf{x} = \begin{pmatrix} x_1 \\ \vdots \\ x_M \end{pmatrix}, \quad \mathbf{f}(\mathbf{x}) = \begin{pmatrix} f_1(\mathbf{x}) \\ \vdots \\ f_M(\mathbf{x}) \end{pmatrix}.$$

El objetivo es construir una sucesión $\mathbf{x}^{(0)}, \mathbf{x}^{(1)}, \dots$ que se acerque a una raíz $\mathbf{x}^*$ del sistema.

Idea común. Tanto Newton como Broyden reemplazan localmente el problema no lineal por uno lineal. La diferencia esencial es cómo obtienen la matriz que describe esa linealización: Newton calcula la Jacobiana; Broyden la aproxima y la actualiza de iteración en iteración.

2 Método de Newton-Raphson en varias dimensiones

2.1 De la linealización a la iteración

Cerca de un punto $\mathbf{x}^{(n)}$, el desarrollo de Taylor de primer orden es

$$\mathbf{f}(\mathbf{x}^{(n)} + \Delta\mathbf{x}) \approx \mathbf{f}(\mathbf{x}^{(n)}) + J(\mathbf{x}^{(n)}) \Delta\mathbf{x},$$

donde J es la **matriz Jacobiana**,

$$J(\mathbf{x}) = \left(\frac{\partial f_i}{\partial x_j} \right)_{i,j=1}^M = \begin{pmatrix} \frac{\partial f_1}{\partial x_1} & \dots & \frac{\partial f_1}{\partial x_M} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_M}{\partial x_1} & \dots & \frac{\partial f_M}{\partial x_M} \end{pmatrix}.$$

Pedimos que la aproximación lineal se anule. Así,

$$J(\mathbf{x}^{(n)}) \Delta\mathbf{x}^{(n)} = -\mathbf{f}(\mathbf{x}^{(n)}), \quad \mathbf{x}^{(n+1)} = \mathbf{x}^{(n)} + \Delta\mathbf{x}^{(n)}.$$

Formalmente, si J es invertible,

$$\boxed{\mathbf{x}^{(n+1)} = \mathbf{x}^{(n)} - J(\mathbf{x}^{(n)})^{-1} \mathbf{f}(\mathbf{x}^{(n)})}.$$

Importante en la práctica. No se debe calcular explícitamente J^{-1} . En cada iteración se resuelve el sistema lineal $J \Delta\mathbf{x} = -\mathbf{f}$, por ejemplo mediante una factorización LU, y después se actualiza $\mathbf{x}$.

2.2 Algoritmo

1. Elegir un punto inicial $\mathbf{x}^{(0)}$ y una tolerancia ε .
2. Evaluar $\mathbf{f}(\mathbf{x}^{(n)})$ y $J(\mathbf{x}^{(n)})$.
3. Resolver $J(\mathbf{x}^{(n)})\Delta\mathbf{x}^{(n)} = -\mathbf{f}(\mathbf{x}^{(n)})$.
4. Actualizar $\mathbf{x}^{(n+1)} = \mathbf{x}^{(n)} + \Delta\mathbf{x}^{(n)}$.
5. Detener el proceso cuando el paso y/o el residuo sean suficientemente pequeños, por ejemplo

$$\|\Delta\mathbf{x}^{(n)}\| \leq \varepsilon_x(1 + \|\mathbf{x}^{(n+1)}\|), \quad \|\mathbf{f}(\mathbf{x}^{(n+1)})\| \leq \varepsilon_f.$$

Usar ambos criterios es más seguro que vigilar únicamente el cambio entre iteraciones: un paso pequeño no garantiza por sí solo que el residuo sea pequeño.

2.3 Convergencia y dificultades

Si $\mathbf{f}$ es suficientemente regular, la raíz es simple y el punto inicial está suficientemente cerca de ella, Newton presenta típicamente **convergencia cuadrática**: una vez en el régimen asintótico, el error decrece aproximadamente como el cuadrado del error anterior. Sin embargo, el método puede fallar o comportarse mal si el punto inicial es inadecuado, si la Jacobiana es singular o está mal condicionada, o si el sistema conduce la iteración lejos de la raíz deseada.

3 Ejemplo: un sistema de dos ecuaciones

Consideremos

$$\begin{cases} x_1^2 + 4x_2^2 - 9 = 0, \\ -25x_1^2 + 10x_2 + 25 = 0, \end{cases} \quad \mathbf{x}^{(0)} = (1, 1)^T.$$

Definimos

$$\mathbf{f}(\mathbf{x}) = \begin{pmatrix} x_1^2 + 4x_2^2 - 9 \\ -25x_1^2 + 10x_2 + 25 \end{pmatrix}, \quad J(\mathbf{x}) = \begin{pmatrix} 2x_1 & 8x_2 \\ -50x_1 & 10 \end{pmatrix}.$$

En $\mathbf{x}^{(0)} = (1, 1)^T$,

$$\mathbf{f}(1, 1) = \begin{pmatrix} -4 \\ 10 \end{pmatrix}, \quad J(1, 1) = \begin{pmatrix} 2 & 8 \\ -50 & 10 \end{pmatrix}.$$

El primer paso se obtiene resolviendo

$$\begin{pmatrix} 2 & 8 \\ -50 & 10 \end{pmatrix} \begin{pmatrix} \Delta x_1 \\ \Delta x_2 \end{pmatrix} = \begin{pmatrix} 4 \\ -10 \end{pmatrix},$$

de donde

$$\Delta\mathbf{x}^{(0)} = \begin{pmatrix} 2/17 \\ 8/17 \end{pmatrix}, \quad \mathbf{x}^{(1)} = \begin{pmatrix} 19/17 \\ 25/17 \end{pmatrix} \approx \begin{pmatrix} 1.11765 \\ 1.47059 \end{pmatrix}.$$

Repetimos exactamente el mismo proceso: nueva función, nueva Jacobiana, nuevo sistema lineal. La tabla muestra la convergencia desde $(1, 1)$:

n	$x_1^{(n)}$	$x_2^{(n)}$	$\ \mathbf{f}(\mathbf{x}^{(n)})\ _2$
0	1.000000	1.000000	10.770330
1	1.117647	1.470588	3.081306
2	1.058636	1.301483	0.301585
3	1.057145	1.293968	0.000290
4	1.057144	1.293965	2.73×10^{-10}

Por tanto, esta elección inicial converge rápidamente a la raíz aproximada

$$\mathbf{x}^* \approx (1.057144, 1.293965)^T.$$

4 Métodos cuasi-Newton: motivación

Para sistemas grandes o funciones costosas, construir la Jacobiana en cada paso puede resultar caro. Los métodos **cuasi-Newton** sustituyen $J(\mathbf{x}^{(n)})$ por una aproximación $A^{(n)}$ que se mejora mediante la información adquirida durante las iteraciones. Broyden es la extensión multidimensional natural de la idea de la secante.

5 Método de Broyden

5.1 Recordatorio: secante en una dimensión

En una dimensión, Newton utiliza $f'(x^{(n)})$. La secante reemplaza esta derivada por

$$A^{(n)} = \frac{f(x^{(n)}) - f(x^{(n-1)})}{x^{(n)} - x^{(n-1)}}$$

y calcula

$$x^{(n+1)} = x^{(n)} - [A^{(n)}]^{-1} f(x^{(n)}).$$

La información de dos puntos permite, por tanto, evitar la evaluación explícita de la derivada.

5.2 La condición de secante multidimensional

Definamos

$$\mathbf{s}^{(n)} = \mathbf{x}^{(n)} - \mathbf{x}^{(n-1)}, \quad \mathbf{y}^{(n)} = \mathbf{f}(\mathbf{x}^{(n)}) - \mathbf{f}(\mathbf{x}^{(n-1)}).$$

Una aproximación razonable $A^{(n)}$ a la Jacobiana debe reproducir el cambio observado de la función en la dirección del último desplazamiento:

$$A^{(n)} \mathbf{s}^{(n)} = \mathbf{y}^{(n)}.$$

Esta es la **condición de secante**. En varias dimensiones no determina de forma única todos los elementos de $A^{(n)}$, por lo que necesitamos además una regla que altere lo menos posible la estimación anterior.

5.3 Actualización de Broyden

La actualización clásica, de rango uno, es

$$A^{(n)} = A^{(n-1)} + \frac{(\mathbf{y}^{(n)} - A^{(n-1)} \mathbf{s}^{(n)})[\mathbf{s}^{(n)}]^T}{[\mathbf{s}^{(n)}]^T \mathbf{s}^{(n)}}.$$

El vector $\mathbf{y}^{(n)} - A^{(n-1)} \mathbf{s}^{(n)}$ mide cuánto se equivocó la antigua aproximación al predecir el cambio real de $\mathbf{f}$. El producto exterior corrige A precisamente en la dirección para la que acabamos de obtener nueva información.

Es fácil comprobar la condición de secante multiplicando por $\mathbf{s}^{(n)}$:

$$A^{(n)} \mathbf{s}^{(n)} = A^{(n-1)} \mathbf{s}^{(n)} + (\mathbf{y}^{(n)} - A^{(n-1)} \mathbf{s}^{(n)}) = \mathbf{y}^{(n)}.$$

Para $M = 1$, la fórmula se reduce exactamente a la pendiente de la secante.

5.4 Algoritmo práctico

Una forma clara de organizar el método es:

1. Elegir $\mathbf{x}^{(0)}$ y una matriz inicial $A^{(0)}$. Si está disponible y su coste inicial es aceptable, una elección habitual es $A^{(0)} = J(\mathbf{x}^{(0)})$.
2. Resolver $A^{(n)}\Delta\mathbf{x}^{(n)} = -\mathbf{f}(\mathbf{x}^{(n)})$.
3. Actualizar $\mathbf{x}^{(n+1)} = \mathbf{x}^{(n)} + \Delta\mathbf{x}^{(n)}$.
4. Calcular $\mathbf{s}^{(n+1)} = \mathbf{x}^{(n+1)} - \mathbf{x}^{(n)}$ y $\mathbf{y}^{(n+1)} = \mathbf{f}(\mathbf{x}^{(n+1)}) - \mathbf{f}(\mathbf{x}^{(n)})$.
5. Actualizar $A^{(n+1)}$ con la fórmula de Broyden.
6. Comprobar la convergencia y repetir si es necesario.

Si $[\mathbf{s}^{(n)}]^T \mathbf{s}^{(n)}$ es extremadamente pequeño, la actualización requiere cuidado numérico; en una implementación robusta puede ser conveniente reiniciar A mediante una nueva Jacobiana.

6 Newton frente a Broyden

	Newton	Broyden
Matriz usada	Jacobiana exacta $J(\mathbf{x}^{(n)})$	Aproximación $A^{(n)}$ a la Jacobiana
Derivadas	Necesarias en cada iteración	No necesariamente después de inicializar
Coste por iteración	Puede ser elevado si formar J es costoso	Reduce el coste de actualizar la matriz
Convergencia local	Típicamente cuadrática bajo hipótesis estándar	Típicamente superlineal bajo hipótesis apropiadas
Ventaja principal	Rapidez cerca de una raíz simple	Evita recalcular derivadas completas
Riesgo práctico	Jacobiana singular/mal condicionada o costosa	La aproximación puede deteriorarse y requerir reinicio

7 Resumen operativo

Newton: linealizar con la Jacobiana exacta, resolver el sistema lineal y corregir la estimación.

Broyden: hacer el mismo tipo de paso, pero sustituir la Jacobiana por una matriz aproximada que aprende del último desplazamiento y del cambio observado en la función.

En ambos casos: resolver sistemas lineales en lugar de formar inversas explícitas, controlar tanto el tamaño del paso como el residuo y vigilar el condicionamiento de la matriz usada.