# Método de Broyden

Leonard David Vivas Dallos
Tomás Escobar Rivera
Grupo 13

28 de noviembre de 2023

# Índice

# 1. Introducción

En el universo del análisis numérico y la optimización, el método de Broyden surge como un ingenio en términos de adaptabilidad. Un método por el cual su creador, Charles George Broyden deja una marca en la resolución de ecuaciones no lineales, y por tanto, una huella en el estudio de estos mismos. Así pues, en el presente artículo, se abordará el método de Broyden como una solución para la resolución de sistemas de ecuaciones no lineales. Se procederá a examinar tanto su motivación como su contexto histórico, seguido de una descripción acompañada de ejemplificaciones para una mayor claridad y, finalmente, un análisis detenido y detallado del mismo. Esto, con el fin de evidenciar cómo se logra llegar a un método que evade el cálculo del Jacobiano, cálculo costoso y así reduce la cantidad de operaciones aritméticas, evidenciando su utilidad y adaptabilidad a distintas situaciones.

# 2. Historia

En 1965, Broyden presentó una categoría de métodos dirigidos a la resolución de ecuaciones simultáneas no lineales, con el propósito de mejorar el método de Newton. El autor identifica algunas limitaciones inherentes a dicho método, destacando, principalmente, que el jacobiano es difícil de computar, incluso cuando se utiliza una aproximación, y la posibilidad de divergencia. No obstante, se aprecia un aspecto favorable en la simplicidad del algoritmo y su eficiencia. ¿Qué ocurriría si pudiéramos minimizar el número de evaluaciones del jacobiano, proponer un método para actualizarlo y prevenir la divergencia? Broyden menciona el método de Newton y posteriormente introduce una variable denominada $B$, la cual representa una aproximación del jacobiano. Asimismo, emplea la variable $t$ para evitar la divergencia.

Así es como el método de Broyden, se deriva del método generalizado de Newton. Algunos argumentan que proviene del método de la secante, basándose en la reutilización del punto anterior y la similitud de notación. Se podría sostener que, debido a la reutilización del punto anterior y la no-

tación similar, este método representa una generalización del método de la secante. En la literatura, es común encontrar afirmaciones que indican que Broyden describe una categoría de métodos secantes basados en la secante y la ecuación. No obstante, algunos, como Broyden, señalan que en su artículo sobre la mejora del método de Newton, nunca empleó la palabra *secante*, además de que su método requiere condiciones iniciales que parten de un solo punto y del jacobiano, a diferencia de dicho método. Si hubiera intentado generalizar el método secante, habría dado lugar a pasos colineales en lugar de la orientación ortogonal que finalmente adoptó.

# 3. Motivación

En un cuidadoso análisis del método de Newton, Broyden nota dos desventajas de dicho método en términos de practicidad y aplicabilidad. El primero de estos radica en la dificultad en la computación de la matriz Jacobiana, ya que sin importar de la complejidad de las funciones $f_j$, la cantidad de evaluaciones de las $n^2$ derivadas parecía un poco excesiva. Más aún, en la práctica, cuando la evaluación exacta de las derivadas parciales es complicada y se usan aproximaciones de diferencia finita a dichas derivadas parciales, como por ejemplo:

$$\frac{\partial f_j}{\partial x_k}(x^{(i)}) \approx \frac{f_j(x^{(i)} + e_k h) - f_j(x^{(i)})}{h}$$

donde $h$ es pequeño en valor absoluto y $e_k$ es el vector cuyo único elemento diferente de cero es 1 en la k-ésima coordenada.

Este tipo de aproximaciones aún requieren de la realización de por lo menos $n^2$ evaluaciones funcionales escalares para realizar la aproximación del Jacobiano y por ende, no reduce la cantidad de cálculos necesarios, el cual por lo general es $O(n^3)$ para resolver el sistema lineal que contiene al Jacobiano aproximado. Así pues, el esfuerzo total necesario para una sola iteración en el método de Newton es de por lo menos, $(n^2 + n)$ evaluaciones funcionales escalares $(n^2)$ para la evaluación de la matriz Jacobiana y n para la evaluación de $(F)$ junto con $O(n^3)$ operaciones aritméticas para resolver el sistema lineal. Es fácil evidenciar que la cantidad de operaciones necesarias genera un problema de esfuerzo computacional bastante grande excepto para valores relativamente pequeños de n con funciones escalares fáciles de evaluar.

La segunda desventaja notada radica en el hecho de que sin algunas modificaciones es bastante frecuente que el método falle en términos de convergencia, esto dado que la certeza en que la estimación inicial de la solución sea lo suficientemente buena es un requerimiento que en la mayoría de los casos prácticos es imposible de asegurar, por lo que el método cae en una poca confianza y utilidad práctica. Con estas motivaciones en mente y centrándonos en la primer desventaja del método surge el Método de Broyden, que será descrito y estudiado en el presente artículo.

# 4. Descripción matemática

Un fenómeno de interés del cual se apalanca el método radica en que el jacobiano no experimenta notables cambios después de los primeros pasos en el método de Newton. Broyden nota que al multiplicar el jacobiano por la variación en $X$ obtenemos un resultado que se acerca bastante al cambio en $F$. Es decir,

$$J(X_n)(X_n - X_{n-1}) \approx F(X_n) - F(X_{n-1})$$

donde $J(X_n)$ es una matriz $nxn$ y $X_n$ es el vector de $n$ elementos. Notemos que es un sistema indeterminado. Por otro lado, así lo expuso y formuló Broyden.

$$f_{i+1} - f_i = t_i B_{i+1} p_i$$

donde $p_i = x_{i+1} - x_i$, $x_{i+1} = x_i + t_i p_i$. Además, cabe resaltar que el valor $t$ se introduce para evitar la divergencia[1] y se mantendrá en 1. La variable $p$ corresponde, en realidad, a nuestro cambio en $X$. Si se intenta emplear esta aproximación para resolver $J$, una matriz de dimensiones $nxn$ el problema radica en que el lado derecho de la ecuación es un vector compuesto por $n$ elementos, lo que origina un sistema indeterminado. Broyden establece ciertas condiciones, notando que la ecuación no definía una matriz sóla pues no describía cómo actuaba $B_{i+1}$ en vectores ortogonales a $x_{i+1} - x_i$, y por tanto permitiendo esencialmente que la variación en la dirección perpendicular sea ortogonal a la actual, introduciendo estas dos restricciones.

[1] Broyden explica que la divergencia puede ser evitada asegurándonos que alguna norma de $f_i$ es una función no creciente de $i$. En nuestro método, $t_i$ es escogido con el fin de minimizar la norma de $f_{i+1}$

$$B_{i+1}q_i = B_i q_i \quad ; \quad q_i^T p_i = 0$$

Es decir, en el método, se escoge $B_{i+1}$ de tal manera que el cambio en $f$ predicho por $B_{i+1}$ en una dirección $q_i$ ortogonal a $p_i$ es el mismo que el que debería ser predicho por $B_i$. Esto permite la solución de $J$, pues permite la unicidad de $B_{i+1}$. Broyden emplea esta notación.

$$B_{i+1} = B_i + \frac{(f_{i+1} - f(t_i - s_i) - s_i B_i p_i)p_i^T}{s_i p_i^T p_i}$$

Sin embargo, es posible simplificarla haciendo que $s$ sea igual a $t$, que se convierte en 1 para obtener

$$B_{i+1} = B_i + \frac{(f_{i+1} - f_i - B_i p_i)p_i^T}{p_i^T p_i}$$

De igual manera, haciendo cambio de variable $y_i = f_{i+1} - f_i$ tenemos

$$B_{i+1} = B_i + \frac{(y_i - B_i p_i)p_i^T}{p_i^T p_i}$$

En términos algo más contemporáneos, esta ecuación es posible representarla como

$$J(X_n) = J(X_{n-1}) + \frac{F(X_n) - F(X_{n-1} - J(X_{x-1})(X_n - X_{n-1})}{||X_n - X_{n-1}||^2}(X_n - X_{n-1})^T$$

A pesar de ello, resulta extensa. Es factible simplificar con el propósito de eliminar algunas de las $X$ y haciendo uso de la función delta.

$$J_n = J_{n-1} + \frac{\Delta F_n - J_{n-1}\Delta X_n}{||\Delta X_n||^2}\Delta X_n^T$$

Posteriormente, si se utiliza el método de Newton y se toma la inversa de $J$, se obtiene una versión del método de Broyden.

$$J_n = J_{n-1} + \frac{\Delta F_n - J_{n-1}\Delta X_n}{||\Delta X_n||^2}\Delta X_n^T$$
$$X_{n+1} = X_n - J_n^{-1}F(X_n)$$

No obstante, si el método se lleva acabo como se indica anteriormente el número de evaluaciones funcionales escalares se reduce de $(n^2+n)$ a $n$, pero el método aún requiere de $O(n^3)$ cálculos para resolver el sistema lineal $n$x$n$ asociado. Por tanto, es importante destacar que esta versión no corresponde exactamente al método de Broyden, aunque sigue siendo valiosa. Broyden, de hecho, da un paso adicional en su método, al notar la existencia de $J$ inversa.

$$X_{n+1} = X_n - \boldsymbol{J_n^{-1}} F(X_n)$$

¿Qué sucedería si fuéramos capaces de resolver la inversa de $J$ en lugar de $J$? De esta forma, establece la ecuación para $H$, su aproximación a la inversa de $J$.

$$H_i = -B_i^{-1}$$
$$H_{i+1} = H_i - \frac{(s_i p_i + H_i y_i) p_i^T H_i}{p_i^T H_i y_i}$$

En realidad, es posible utilizar lo que se conoce como la fórmula de inversión matricial de Sherman-Morrison. Este resultado dice que si $A$ es una matriz no singular y $x$ e $y$ son vectores, entonces $A + xy^t$ es no singular siempre que $y^t A^{-1} x \neq -1$, y

$$(A + xy^t)^{-1} = A^{-1} - \frac{A^{-1} x y^t A^{-1}}{1 + y^t A^{-1} x}$$

No obstante, Broyden desconocía esta fórmula y usa una modificación de la formula de Householder, bastante similar de por si a la fórmula de Sherman-Morrison para llegar al resultado. En términos contemporáneos, la aproximación de la inversa de J se expresaría de la siguiente manera.

$$J_n^{-1} = J_{n-1}^{-1} + \frac{\Delta X_n - J_{n-1}^{-1} \Delta F_n}{\Delta X_n^T J_{n-1}^{-1} \Delta F_n} \Delta X_n^T J_{n-1}^{-1}$$

A continuación, se aplica el método de Newton.

$$X_{n+1} = X_n - J_n^{-1} F(X_n)$$

Y se obtiene el método de Broyden. Sin embargo, es necesario destacar que se trata de una versión del método de Broyden.

Esta versión del método de Broyden se denomina el *método de Broyden bueno*. Existe otra versión que Broyden desarrolló, conocida como el *método de Broyden malo*. Además, existe un método que ha sido pasado por alto en la historia. En términos actuales, nuestros métodos se representan de la siguiente manera.

- Método 1: *Bueno*

$$J_n^{-1} = J_{n-1}^{-1} + \frac{(\Delta X_n - J_{n-1}^{-1}\Delta F_n)\Delta X_n^T J_{n-1}^{-1}}{\Delta X_n^T J_{n-1}^{-1}\Delta F_n}$$

- Método 2: *Malo*

$$J_n^{-1} = J_{n-1}^{-1} + \frac{(\Delta X_n - J_{n-1}^{-1}\Delta F_n)\Delta F_n^T}{\Delta F_n^T \Delta F_n}$$

- Método 3: Supongamos $f_j$ *son las primeras derivadas parciales de una función convexa*

$$J_n^{-1} = J_{n-1}^{-1} + \frac{J_{n-1}^{-1}\Delta F_n \Delta F_n^T J_{n-1}^{-1}}{\Delta F_n^T J_{n-1}^{-1}\Delta F_n} - \frac{\Delta X_n \Delta X_n^T}{\Delta X_n^T \Delta F_n}$$

En su artículo, Broyden señala que el segundo método parecía insatisfactorio en la práctica.

# 5. Análisis del método y análisis de error

El método de Broyden resulta ser, un método que solo requiere de n evaluaciones funcionales escalares por iteración y reduce el número de cálculos aritméticos a $O(n^2)$. Método de la familia de los algoritmos llamados cuasi-Newton, los cuales reemplazan la matriz Jacobiana por una matriz de aproximación que se actualiza en cada iteración. Empero, como ya veremos y como es bien sabido, *la mejora en un aspecto implica la pérdida en otro aspecto*, y en este caso, una mejora en la cantidad de evaluaciones implica la pérdida en la convergencia cuadrática de Newton, siendo reemplazada por una convergencia **super-lineal**. Lo que implica que $\lim_{i\to\infty} \frac{||x^{(i+1)}-p||}{||x^{(i)}-p||} = 0$, donde $p$ denota la solución a $F(x) = 0$ y $x^{(i)}$ y $x^{(i+1)}$ son aproximaciones consecutivas.

Un cambio en teoría aceptable por la reducción de cómputo. Además de esto, el método resulta no ser autocorregible a diferencia del método de Newton, que generalmente corrige errores de redondeo mediante iteraciones sucesivas.

Finalmente, se aborda el tema del orden, que resulta ser bastante complejo. En 1971, Broyden pudo demostrar que, como mínimo, el método era lineal. Posteriormente, él y otros investigadores demostraron que, como mínimo, era superlineal dos años después. No fue sino hasta 1979 que Gay logró demostrar que el orden era cuadrático.

## 5.1. Teorema

Sea $F : R^n \to R^n$ continuamente diferenciable en un conjunto convexo abierto $D$, y suponga que $F(x^*) = 0$ y $F'(x^*)$ es no singular para algún $x^* \in D$. Más aún, suponga que $F'$ es Lipschitz continua en $x^*$ y que $F$ es afín con la matriz de coeficientes no singulares; esto es,

$$F(x) = Ax - b, \quad A \in L(R^n)\text{no singular.}$$

y considere el método de Broyden. Luego el método de Broyden es globalmente y Q-superlinealmente convergente a $x^*$.
**Nota:** Asumimos que $t_k$ es escogido para satisfacer

$$B_{k+1} \text{ no singular}, \quad |t_k - 1| \leq \hat{t} < 1$$

### 5.1.1. Demostración

El resultado sigue de una estimación cuidados de la diferencia entre $||E_{k+1}||_F^2$ y $||E_k||_F^2$ donde $E_k = B_k - A$ y $||\cdot||_F$ es la norma de Frobenius. Para esto note que

$$E_{k+1} = E_k \left( \frac{I - t_k s_k s_k^T}{||s_k||^2} \right),$$

y por tanto, un cálculo directo con $||E||_f^2 = trace(E^T E)^2$ produce

$$||E_{k+1}||_F^2 = ||E_k||_F^2 - t_k(2 - t_k) \left( \frac{||E_k s_k||}{||s_k||} \right)^2$$

[2] *trace* se refiere a la traza de una matriz

Esto implica que

$$(1-\hat{t})^2 \sum_{k=0}^{\infty} \left( \frac{||E_k s_k||}{||s_k||} \right)^2 \leq ||E_0||_F^2,$$

y en particular,

$$\lim_{k\to\infty} \frac{||E_k s_k||}{||s_k||} = 0$$

Ahora note que $(B_k - A)s_k = -F(x_{k+1}) = -A(x_{k+1} - x^*)$ donde $x^* = A^{-1}b$. Por tanto, si $\varepsilon_k$ esta definido por

$$||A^{-1}||||E_k s_k|| = \varepsilon_k ||s_k||,$$

entonces

$$||x_{k+1} - x^*|| \leq \varepsilon_k ||s_k|| \leq \varepsilon_k [||x_{k+1} - x^*|| + ||x_k - x^*||],$$

y el límite descrito anteriormente claramente muestra que $\{\varepsilon_k\}$ converge a cero. La inecuación de arriba entonces implica que

$$||x_{k+1} - x^*|| \leq \varepsilon_k \frac{||x_k - x^*||}{1 - \varepsilon_k}$$

para $k$ suficientemente largo, y esto prueba que $\{x_k\}$ converge Q-superlinealmente a $x^*$.

# 6. Ejemplos

## 6.1. Ejemplo 1

Considere el sistema de ecuaciones $g(x) = 0$, donde

$$g(x,y,z) = \begin{bmatrix} x^2 + y^2 + z^2 - 3 \\ x^2 + y^2 - z - 1 \\ x + y + z - 3 \end{bmatrix}$$

Iniciamos con un paso del Método de Newton para resolver sistemas de ecuaciones, con aproximación inicial $\mathrm{x}^{(0)} = (x^{(0)}, y^{(0)}, z^{(0)}) = (1, 0, 1)$. Computando el Jacobiano, tenemos

$$J_g(x,y,z) = \begin{bmatrix} 2x & 2y & 2z \\ 2x & 2y & -1 \\ 1 & 1 & 1 \end{bmatrix}$$

Luego, la iteración $\mathrm{x}^{(1)}$ de Newton es obtenida resolviendo el sistema de ecuaciones

$$J_g(\mathrm{x}^{(0)})(\mathrm{x}^{(1)} - \mathrm{x}^{(0)}) = -g(\mathrm{x}^{(0)})$$

o, equivalentemente, $D_0 d^{(0)} = -g(\mathrm{x}^{(0)})$ donde

$$D_0 = J_g(\mathrm{x}^{(0)}) = \begin{bmatrix} 2x^{(0)} & 2y^{(0)} & 2z^{(0)} \\ 2x^{(0)} & 2y^{(0)} & -1 \\ 1 & 1 & 1 \end{bmatrix}$$

$$d_0 = \begin{bmatrix} x^{(1)} - x^{(0)} \\ y^{(1)} - y^{(0)} \\ z^{(1)} - z^{(0)} \end{bmatrix}$$

$$g(\mathrm{x}^{(0)}) = \begin{bmatrix} (x^{(0)})^2 + (y^{(0)})^2 + (z^{(0)})^2 - 3 \\ (x^{(0)})^2 + (y^{(0)})^2 - (z^{(0)}) - 1 \\ (x^{(0)}) + (y^{(0)}) + (z^{(0)}) - 3 \end{bmatrix}$$

Sustituyendo $(x^{(0)}, y^{(0)}, z^{(0)}) = (1, 0, 1)$ nos queda el sistema

$$\begin{bmatrix} 2 & 0 & 2 \\ 2 & 0 & -1 \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} x^{(1)} - 1 \\ y^{(1)} \\ z^{(1)} - 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

El sistema tiene solución $d^{(0)} = \left(\frac{1}{2}, \frac{1}{2}, 0\right)$ lo que genera $\mathrm{x}^{(1)} = \left(\frac{3}{2}, \frac{1}{2}, 1\right)$. Ahora, en vez de computar $J_g(\mathrm{x}^{(1)})$, computamos

$$
\begin{aligned}
D_1 &= D_0 + \frac{1}{d^{(0)} \cdot d^{(0)}} g(\mathrm{x}^{(1)}) \otimes d^{(0)} \\
&= \begin{bmatrix} 2 & 0 & 2 \\ 2 & 0 & -1 \\ 1 & 1 & 1 \end{bmatrix} + \frac{1}{1/2} \begin{bmatrix} \frac{1}{2} \\ \frac{1}{2} \\ 0 \end{bmatrix} \otimes \begin{bmatrix} \frac{1}{2} \\ \frac{1}{2} \\ 0 \end{bmatrix} \\
&= \begin{bmatrix} 2 & 0 & 2 \\ 2 & 0 & -1 \\ 1 & 1 & 1 \end{bmatrix} + 2 \begin{bmatrix} \frac{1}{4} & \frac{1}{4} & 0 \\ \frac{1}{4} & \frac{1}{4} & 0 \\ 0 & 0 & 0 \end{bmatrix} \\
&= \begin{bmatrix} \frac{5}{2} & \frac{1}{2} & 2 \\ \frac{5}{1} & \frac{1}{2} & -1 \\ 1 & 1 & 1 \end{bmatrix}
\end{aligned}
$$

Esto genera el sistema

$$
\begin{bmatrix} \frac{5}{2} & \frac{1}{2} & 2 \\ \frac{5}{1} & \frac{1}{2} & -1 \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} x^{(2)} - \frac{3}{2} \\ y^{(2)} - \frac{1}{2} \\ z^{(2)} - 1 \end{bmatrix} = \begin{bmatrix} -\frac{1}{2} \\ -\frac{1}{2} \\ 0 \end{bmatrix}
$$

el cual tiene solución $\mathrm{x}^{(2)} = \left(\frac{5}{4}, \frac{3}{4}, 1\right)$. Esta resulta ser la misma segunda iteración computada con el Método de Newton, aunque no es el caso de una función general $g(x)$.

## 6.2. Ejemplo 2

Consideremos el sistema no lineal dado por

$$
\begin{aligned}
3x_1 - \cos(x_2 x_3) - \frac{1}{2} &= 0, \\
x_1^2 - 81(x_2 + 0{,}1)^2 + \sin x_3 + 1{,}06 &= 0, \\
\epsilon^{-x_1 x_2} + 20x_3 + \frac{10\pi - 3}{3} &= 0.
\end{aligned}
$$

La matriz Jacobiana para este sistema es

$$
J(x_1, x_2, x_3) = \begin{bmatrix} 3 & x_3 \sin(x_2 x_3) & x_2 \sin(x_2 x_3) \\ 2x_1 & -162(x_2 + 0{,}1) & \cos x_3 \\ -x_2 \epsilon^{-x_1 x_2} & -x_1 \epsilon^{-x_1 x_2} & 20 \end{bmatrix}
$$

Con $\mathrm{x}^{(0)} = (0{,}1, 0{,}1, -0{,}1)^t$, tenemos $F(\mathrm{x}^{(0)}) = (-1{,}199949, -2{,}269832, 8{,}462026)^t$.

$$A_0 = J(\mathrm{x}^{(0)}) = \begin{bmatrix} 3 & 9{,}999836 \times 10^{-4} & -9{,}999836 \times 10^{-4} \\ 0{,}2 & -323{,}9999 & 0{,}9950041 \\ -9{,}900498 \times 10^{-2} & -9{,}900498 \times 10^{-2} & 20 \end{bmatrix}$$

$$A_0^{-1} = J(\mathrm{x}^{(0)})^{-1} = \begin{bmatrix} 0{,}3333331 & 1{,}023852 \times 10^{-5} & 1{,}615703 \times 10^{-5} \\ 2{,}108606 \times 10^{-3} & -3{,}086882 \times 10^{-2} & 1{,}535838 \times 10^{-3} \\ 1{,}660522 \times 10^{-3} & -1{,}527579 \times 10^{-4} & 5{,}000774 \times 10^{-2} \end{bmatrix}$$

$$\mathrm{x}^{(1)} = \mathrm{x}^{(0)} - A_0^{-1} F(\mathrm{x}^{(0)}) = (0{,}4998693, 1{,}946693 \times 10^{-2}, -0{,}5215209)^t$$

$$F(\mathrm{x}^{(1)}) = (-3{,}404021 \times 10^{-4}, -0{,}3443899, 3{,}18737 \times 10^{-2})^t$$

$$y^{(1)} = F(\mathrm{x}^{(1)}) - F(\mathrm{x}^{(0)}) = (1{,}199608, 1{,}925442, -8{,}430152)^t$$

$$s_1 = (0{,}3998693, -8{,}053307 \times 10^{-2}, -0{,}4215209)^t$$

$$s_1^t A_0^{-1} y_1 = 0{,}3424604$$

$$A_1^{-1} = A_0^{-1} + \frac{1{,}0}{0{,}3424604}[(s_1 - A_0^{-1} y_1) s_1^t A_0^{-1}] =$$

$$= \begin{bmatrix} 0{,}3333781 & 1{,}11077 \times 10^{-5} & 8{,}944584 \times 10^{-6} \\ 2{,}021271 \times 10^{-3} & -3{,}094847 \times 10^{-2} & 2{,}196909 \times 10^{-3} \\ 1{,}022381 \times 10^{-3} & -1{,}650679 \times 10^{-4} & 5{,}010987 \times 10^{-2} \end{bmatrix}$$

y

$$\mathrm{x}^{(2)} = \mathrm{x}^{(1)} - A_1^{-1} F(\mathrm{x}^{(1)}) = (0{,}4999863, 8{,}737888 \times 10^{-3}, -0{,}5231746)^t$$

Los resultados de las otras iteraciones se muestran en la tabla siguiente:

| $k$ | $x_1^{(k)}$ | $x_2^{(k)}$ | $x_3^{(k)}$ | $\|\|\mathrm{x}^{(k)} - \mathrm{x}^{(k-1)}\|\|_\infty$ |
|---|---|---|---|---|
| 3 | 0,5000066 | $8{,}672215 \times 10^{-4}$ | $-0{,}5236918$ | $7{,}88 \times 10^{-3}$ |
| 4 | 0,5000005 | $6{,}087473 \times 10^{-5}$ | $-0{,}5235954$ | $8{,}12 \times 10^{-4}$ |
| 5 | 0,5000002 | $-1{,}445223 \times 10^{-6}$ | $-0{,}5235989$ | $6{,}24 \times 10^{-5}$ |

# 7. Algoritmo de Aplicación

Para aproximar una solución del sistema no lineal $F(x) = 0$, dada una aproximación inicial x:
**Entrada:** número $n$ de ecuaciones e incógnitas; aproximación inicial $\mathrm{x} = (x_1, x_2, \ldots, x_n)^t$; tolerancia $TOL$; número máximo de iteraciones $N_0$
**Salida:** solución aproximada $\mathrm{x} = (x_1, x_2, \ldots, x_n)^t$ o mensaje de que el número de iteraciones fue excedido.

**Paso 1:** tomar $A_0 = J(\mathrm{x})$ donde $J(\mathrm{x})_{ij} = \frac{\partial f_i(\mathrm{x})}{\partial x_j}$ para $1 \leq i, j \leq n$;

$$v = F(\mathrm{x}); \quad (\text{Nota: } v = F(\mathrm{x}^{(0)}))$$

**Paso 2:** tomar $A = A^{-1}$;
**Paso 3:** tomar $k = 1$;

$$s = -Av; \quad (\text{Nota: } s = s_1)$$
$$\mathrm{x} = \mathrm{x} + s; \quad (\text{Nota: } \mathrm{x} = \mathrm{x}^{(1)}))$$

**Paso 4:** mientras que $k \leq N_0$ seguir pasos $5 - 13$;
**Paso 5:** tomar

$$w = v; \quad (\text{Guardar } v)$$
$$v = F(\mathrm{x}); \quad (\text{Nota: } v = F(\mathrm{x}^{(k)}))$$
$$y = v - w; \quad (\text{Nota: } y = y^{(k)})$$

**Paso 6:** tomar

$$z = -Ay; \quad (\text{Nota: } z = -A_{k-1}^{-1} y_k)$$

**Paso 7:** tomar

$$p = -s^t z; \quad (\text{Nota: } p = s_k^t A_{k-1}^{-1} y_k)$$

**Paso 8:** tomar

$$C = pI + (s + z)s^t; \quad (\text{Nota: } C = s_k^t A_{k-1}^{-1} y_k I + (s_k + A_{k-1}^{-1} y_k) s_k^t)$$

**Paso 9:** tomar

$$A = CA/p; \quad \text{(Nota: } A = A_k^{-1}\text{)}$$

**Paso 10:** tomar

$$s = -Av; \quad \text{(Nota: } s = -A_k^{-1}F(\mathrm{x}^{(k)})$$

**Paso 11:** tomar

$$\mathrm{x} = \mathrm{x} + s; \quad \text{(Nota: } \mathrm{x} = \mathrm{x}^{(k)})\text{)}$$

**Paso 12:** si $||s|| < TOL$ entonces SALIDA ($\mathrm{x} = (x_1, x_2, \ldots, x_n)^t$); (*procedimiento completado satisfactoriamente*) PARAR

**Paso 13:** tomar $k = k + 1$

**Paso 14:** SALIDA ('*Número máximo* $N_0$ *de iteraciones excedido,* $N_0 =$', $N_0$); PARAR.

# 8. Programa en Python

- **Librerías necesarias:**
  - numpy
  - scipy.optimize (Para usar approx_fprime)
- **Función *jacobian_fd***: Función que devuelve el Jacobiano inverso de una función evaluada en un punto.

```
def jacobian_fd(f, x, h=1e-5):
    n = len(x)
    jacobian_matrix = np.zeros((n, n))
    for i in range(n):
        jacobian_matrix[i, :] = approx_fprime(x, lambda y: np.array(f(*y)).flatten()[i], h)
    return np.linalg.inv(np.matrix(jacobian_matrix))
```

- **Función** $f(*x)$: Método que toma las funciones ingresadas por consolas a funciones para usar en el programa.

```
def f(*x):
    global functions
    x_dict = {f'x{i}': x[i] for i in range(len(x))}
    results = [eval(func, {**x_dict, 'np': np}) for func in functions]
    return np.matrix(results).T
```

- **Función *broyden***: Función que desarrolla el método de Broyden.

```
def broyden(x0, N=10, tol=1e-5):
    n = len(x0)
    header = "k \t " + "\t\t ".join([f"x{i + 1}" for i in range(n)]) + "\t\t ||x(k) - x(k-1)||"
    print(header)

    A = jacobian_fd(f, x0)
    v = f(*x0)
    k = 2
    s = -A * v
    x0 = np.matrix(x0).T + s

    while k < N:
        w = v
        v = f(*[x0[i, 0].item() for i in range(n)])
        y = v - w
        z = -A * y
        p = (s.T * A * y).item()
        A = A + 1 / p * (s + z) * s.T * A
        s = -A * v
        x0 = x0 + s
        print("{0:1d} \t {1} \t {2:1.6f}".format( *args: k, "\t ".join("{:1.6f}".format(x0[i, 0].item()) for i in range(n)),
                                                  np.linalg.norm(s)))
        if np.linalg.norm(s) < tol:
            return x0
        k += 1
```

- **Interfaz**: Ingreso de información necesaria para el desarrollo del método

```
print("Método de Broyden\n")

n = int(input("Ingrese el número de funciones: "))
functions = []
for i in range(n):
    functions.append(input(f"Ingrese la función {i + 1}: "))

x = []

print("Ingrese los valores del vector inicial ('initial guess')")
for i in range(len(functions)):
    valor = float(input(f"Componente {i + 1}:"))
    x.append(valor)

broyden(x)
```

## 8.1. Caso de Prueba 1

- Número de funciones: 2
- f1: x0**2 - x1 - 1
- f2: x0 - x1**2 + 1

- Vector inicial:
  - componente 1: 1.5
  - componente 2: 2.0
- **Entrada:**

```
Ingrese el número de funciones: 2
Ingrese la función 1: x0**2 - x1 - 1
Ingrese la función 2: x0 - x1**2 + 1
Ingrese los valores del vector inicial ('initial guess')
Componente 1:1.5
Componente 2:2.0
```

- **Salida:**

| k | x1 | x2 | \|\|x(k) - x(k-1)\|\| |
|---|---|---|---|
| 2 | 1.617794 | 1.623311 | 0.040312 |
| 3 | 1.618255 | 1.618243 | 0.005089 |
| 4 | 1.618020 | 1.618024 | 0.000321 |
| 5 | 1.618034 | 1.618034 | 0.000017 |
| 6 | 1.618034 | 1.618034 | 0.000000 |

## 8.2. Caso de Prueba 2

- Número de funciones: 3
- f1: 3*x0 - np.cos(x0*x2) - 0.5
- f2: x0**2 - 81*(x1+0.1)**2 + np.sin(x2) + 1.06
- f3: np.exp(-x0*x1) + 20*x2 + (10*np.pi - 3)/3
- Vector inicial:
  - componente 1: 0.1

- componente 2: 0.1
- componente 3: -0.1

- **Entrada:**

```
Ingrese el número de funciones: 3
Ingrese la función 1: 3*x0 - np.cos(x0*x2) - 0.5
Ingrese la función 2: x0**2 - 81*(x1+0.1)**2 + np.sin(x2) + 1.06
Ingrese la función 3: np.exp(-x0*x1) + 20*x2 + (10*np.pi - 3)/3
Ingrese los valores del vector inicial ('initial guess')
Componente 1:0.1
Componente 2:0.1
Componente 3:-0.1
```

- **Salida:**

| k | x1 | x2 | x3 | ‖x(k) - x(k-1)‖ |
|---|---|---|---|---|
| 2 | 0.488836 | 0.008800 | -0.523208 | 0.015326 |
| 3 | 0.489219 | 0.002521 | -0.523464 | 0.006296 |
| 4 | 0.489124 | -0.000623 | -0.523617 | 0.003149 |
| 5 | 0.489127 | -0.000664 | -0.523615 | 0.000041 |
| 6 | 0.489127 | -0.000667 | -0.523615 | 0.000003 |

# 9. Conclusiones

En síntesis, el método de Broyden se presenta como una herramienta eficaz para resolver sistemas de ecuaciones no lineales, con una convergencia de orden cuadrático. Únicamente se requiere evaluar el Jacobiano una vez o aproximarlo mediante diferencias finitas. Además, se aprovecha el uso de los valores previos de $X$ y $F$ para actualizar la inversa del Jacobiano, sin necesidad de invertir una matriz después de la primera iteración, reduciendo la cantidad de cálculos aritméticos considerablemente. Cabe mencionar que existen diversas versiones del método de Broyden, de las cuales se puede escoger aquella que nos convenga más, para efectos de seguimiento de las iteraciones o convergencia, decidiendo también si dicha versión hace valer la pena el perder velocidad de convergencia con respecto a Newton. Un método sin duda alguna útil que ofrece un equilibrio entre eficiencia computacional

y convergencia, aunque en su camino sacrifica la convergencia cuadrática del mismo.

# 10. Apéndice: Método de Continuación

La convergencia en el método de Newton (y de la secante) no siempre está garantizada. En la práctica es difícil encontrar un buen punto inicial que lleve a la convergencia de las iteraciones.

Dados $f(x)$ y $g(x)$ donde el cero $r$ para $f(x) = 0$ es difícil de resolver, mientras que $g(x) = 0$ es conocido, digamos $g(u) = 0$, construimos una nueva función:

$$\Phi(x; s) = sf(x) + (1 - s)g(x), \quad 0 < s < 1$$

Si $s = 0$ tenemos $\Phi(x; 0) = g(x)$, y si $s = 1$ tenemos $\Phi(x; 1) = f(x)$. Como $s$ oscila entre 0 y 1, el 0 de $\Phi(x; s)$ va a moverse desde $u$ hasta $r$ de modo continuo. Por lo tanto, tomaremos pasos pequeños, empezando desde la raíz de $g(x)$ y continuamente acercándonos a la raíz para $f(x)$. Específicamente sea $N$ el número de pasos, y sea:

$$\Delta s = \frac{1}{N}, \quad sk = k\Delta s, \quad k = 1, 2, \ldots, N$$

Sea $r_0 = u$. Para $k = 1, 2, \ldots, N$ resolvemos $\Phi(x; s_k) = 0$ con el método de newton, usando $r_{k-1}$ como el punto inicial. Si $s$ es suficientemente pequeño, entonces $r_{k-1}$ está suficientemente cerca de $rx$ y el método de Newton converge rápidamente. La raíz final $r_N$ será la raíz de $f(x)$.

# Referencias


Broyden, C. G. (1965). A Class of Methods for Solving Nonlinear Simultaneous Equations. *Mathematics of Computation*, 19(92), 577 − 593. https://doi.org/10.2307/2003941

Muto, V. CAPITULO XXIII. METODOS CUASI-NEWTON. *Métodos Cuasi-Newton*, 307 − 311.

Moré, J. J., & Trangenstein, J. A. (1976). On the Global Convergence of Broyden's Method. *Mathematics of Computation*, 30(135), 523 − 540. https://doi.org/10.2307/2005323

Dennis, J. E. (1971). On the Convergence of Broyden's Method for Nonlinear Systems of Equations. *Mathematics of Computation*, 25(115), 559 − 567. https://doi.org/10.2307/2005218

Lambers, J. (2012). Broyden's Method. Lecture 11 Notes *Summer Session 2011-12*, $MAT419/519$, 1 − 5. https://www.math.usm.edu/lambers/mat419/lecture11.pdf

Gay, D. M. (1979). Some Convergence Properties of Broyden's Method. *SIAM Journal on Numerical Analysis*, 16(4), 623 − 630. http://www.jstor.org/stable/2156533

Martínez, J. M. (1978). On the order of convergence of Broyden-Gay-Schnabel's method. *Commentationes Mathematicae Universitatis Carolinae*, 19(1978), 107 − 118. http://dml.cz/dmlcz/105837