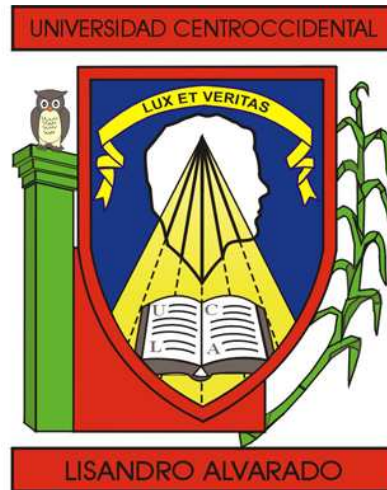


UNIVERSIDAD CENTROCCIDENTAL
“LISANDRO ALVARADO”

Decanato de Ciencias y Tecnología
Licenciatura en Ciencias Matemáticas



“CONVERGENCIA GLOBAL DEL MÉTODO BFGS
MODIFICADO PARA LA MINIMIZACIÓN NO-CONVEXA
IRRESTRICTA”

TRABAJO ESPECIAL DE GRADO PRESENTADO POR

BR. ORLANDO ROMERO A.

COMO REQUISITO FINAL

PARA OBTENER EL TÍTULO DE LICENCIADO

EN CIENCIAS MATEMÁTICAS

ÁREA DE CONOCIMIENTO: MATEMÁTICA APLICADA.

TUTOR: DR. JAVIER HERNÁNDEZ B.

Barquisimeto, Venezuela. Octubre de 2012



Universidad Centroccidental
 "Lisandro Alvarado"
 Decanato de Ciencias y Tecnología
 Licenciatura en Ciencias Matemáticas



ACTA
 TRABAJO ESPECIAL DE GRADO

Los suscritos miembros del Jurado designado por el Jefe del Departamento de Matemáticas del Decanato de Ciencias y Tecnología de la Universidad Centroccidental "Lisandro Alvarado", para examinar y dictar el veredicto sobre el Trabajo Especial de Grado titulado:

“CONVERGENCIA GLOBAL DEL MÉTODO BFGS MODIFICADO PARA LA MINIMIZACIÓN NO-CONVEXA IRRESTRICTA”

presentado por el ciudadano BR. ORLANDO ROMERO A. titular de la Cédula de Identidad No. 18.684.019, con el propósito de cumplir con el requisito académico final para el otorgamiento del título de Licenciado en Ciencias Matemáticas.

Luego de realizada la Defensa y en los términos que imponen los Lineamientos para el Trabajo Especial de Grado de la Licenciatura en Ciencias Matemáticas, se procedió a discutirlo con el interesado habiéndose emitido el veredicto que a continuación se expresa:

¹ _____

Con una calificación de _____ puntos.

En fe de lo expuesto firmamos la presente Acta en la Ciudad de Barquisimeto a los _____ días del mes de _____ de _____.

 TUTOR

 FIRMA

 PRINCIPAL

 FIRMA

 PRINCIPAL

 FIRMA

OBSERVACIONES:

¹ Aprobado ó Reprobado

Dedico este trabajo a la mujer que me crió y me dio todo su amor de madre mi querida abuela Carmen Marina, a mi querido tío Chano gracias por ser como un padre y un hermano para mí, siempre te llevaré en mi corazón, cuando yo muera espero verlos a ambos, y a mi querida Mayarin.

AGRADECIMIENTOS

A dios todo poderoso por todas las cosas que me ha dado.

A mi amada madre por el apoyo incondicional brindado en estos años tan difíciles, a mi querida hermana Isabel.

A mi amada hermana Maria gracias por darme tu tierno y dulce apoyo en los momentos mas fuertes de mi vida te quiero mucho hija.

A mi hermano Aldemar "Leon S. Kennedy" Gomez gracias por tu gran apoyo incondicional y tu gran amistad.

A mi compadre Ricardo "bocono" hernandez por su consejos y su amistad sincera, mi compañero de parrandas.

A mi apreciada compañera Vianney Barrese gracias por brindarme tu apoyo en todo el transcurso de la carrera.

A mi buena amiga y gran compañera de clases Emily, gracias por tu amistad y tu apoyo incondicional y sincero.

A mi querida y adorada amiga yogeydi "la pequeña" gracias por ser la luz que me guió todo este tiempo, y sobre todo en los momentos mas difíciles de mi carrera, y vida personal, por creer siempre en mi, brindándome tu pura y noble amistad vivirás eternamente en mi corazón.

A los muchachos de ASOEM, Joelviz, Emely, Uvencio, Pegao, cleiver, y los muchachos de Unuma gracias a todos ustedes.

A mis compadres de chivacoa y grandes amigos: mi compadre y hermano Josue, mi gran amigo y hermano Miguel, mi compadre Luiggi, mi compadre Luis eduardo, y mi apreciada y buena amiga Valery López los quiero mucho .

A mis grandes y apreciados amigos Fernando y Mayra, gracias por su consejos y siempre desearme lo mejor en mi vida.

Y por ultimo a mi tutor y gran amigo el profesor Javier hernández, gracias por todo el apoyo brindado durante toda la carrera, por sus consejos, sobre todos los del primer semestres que fueron de gran inspiración para no abandonar la carrera, por siempre haber creído en mis cualidades académicas, esta etapa final no lo hubiese logrado sin su incondicional ayuda, estaré eternamente agradecido.

RESUMEN

Convergencia Global del Método BFGS Modificado para la Minimización No-convexa Irrestringida.

En este trabajo, se propone una nueva búsqueda lineal no monótona del tipo armijo y probar que el método MBGFS propuesto por Li y Fukushima con esta nueva búsqueda lineal converge globalmente para la minimización no-convexa. Algunos experimentos numéricos muestran que el método MBFGS no monótono es eficiente para dar resultados de prueba.

Agradecimientos	i
Resumen	iii
Introducción	1
1. Estrategias de búsqueda lineal	3
2. Métodos Cuasi-Newton	16
3. Método BFGS-Modificado	25
4. Pruebas numéricas	37
Conclusiones	42
4.1. funciones.m	43
4.2. gradiente	46
4.3. Armijo	47
4.4. Algoritmos	50
Referencias	54

ÍNDICE DE FIGURAS

1.1. La ideal longitud del paso es un minimizador global.	5
1.2. Reducción insuficiente en f	6
1.3. Condición de disminución suficiente.	7
1.4. Condición de curvatura.	8
1.5. La longitud de paso satisfaciendo las condiciones de Wolfe.	9
1.6. Condiciones de Goldstein.	12
1.7. Condición de Armijo no-monótona	13
1.8. Condición de Goldstein no-monótona	14
1.9. Condición de Wolfe no-monótona	15

INTRODUCCIÓN

En este trabajo, consideramos el siguiente problema de optimización sin restricciones:

$$\begin{aligned} \text{mín } f(x) \\ x \in \mathbb{R}^n \end{aligned} \tag{1}$$

donde $f : \mathbb{R}^n \rightarrow \mathbb{R}$ es una función continuamente diferenciable cuyo gradiente será denotado por g . Este problema lo estudió Zhou y Zhang en [18] y el cual se basa esta monografía.

Los métodos monótonos generan una sucesión $\{x_k\}$ que converge a la solución de (1), requieren que $f(x_k) \leq f(x_{k-1})$ se cumpla en cada iteración. Los métodos que no cumplen esta condición son los llamados métodos no-monótonos. La técnica de búsqueda lineal no monótona fue introducida por primera vez por Grippo y otros en 1986 [4]. Ellos consideraron la siguiente búsqueda lineal no-monótona de manera general como sigue: dado una constante $a > 0$, $\delta, \rho \in (0, 1)$ y un entero no negativo M , seleccionar el tamaño del paso $\alpha_k = \max\{a\rho^0, a\rho^1, \dots\}$ satisfaciendo

$$f(x_k + \rho^m a d_k) \leq \max_{0 \leq j \leq M} f(x_{k-j}) + \delta \rho^m a g_k^T d_k. \tag{2}$$

Cuando $M = 0$ estamos en el caso de la búsqueda lineal de Armijo estándar. Una ventaja de la búsqueda lineal no monótona es que el tamaño del paso α_k puede ser seleccionado tan libremente como sea posible. Experimentos numéricos han mostrado que la técnica de búsqueda lineal no monótonas es muy eficiente (ver [5, 13, 14, 15]). Esta técnica fue originalmente aplicada a métodos tipo Newton [4] y ha sido aplicada a métodos de gradiente conjugado y métodos Cuasi-Newton (ver [3, 2, 6, 8, 9, 10]). En particular Han y Liu [6] propusieron una búsqueda lineal no-monótona del tipo Wolfe para el método BFGS: calculando el tamaño del paso α_k tal que

$$\begin{aligned} f(x_k + \alpha_k d_k) &\leq \max_{0 \leq j \leq M} f(x_{k-j}) + \epsilon_1 \alpha_k g_k^T d_k, \\ d_k^T g(x_k + \alpha_k d_k) &\geq \max\{\epsilon_2, 1 - (\alpha_k \|d_k\|)^p\} g_k^T d_k. \end{aligned}$$

donde $\epsilon_1 \in (0, 1)$, $\epsilon_2 \in (0, \frac{1}{2})$, $p \in (-\infty, 1)$. Bajo ciertas condiciones, Han y Liu [6] probaron que el método BFGS con esta búsqueda lineal no-monótona tipo Wolfe convergía globalmente cuando f es una función convexa .

Dai [3] construyó un ejemplo donde muestra que el método BFGS con búsqueda lineal no-monótona tipo Wolfe puede divergir. Más aun, Mascarenhas [11] también mostró que el método BFGS incluso con búsqueda lineal exacta puede no converger para funciones no convexas.

Afortunadamente, Li y Fukushima [7] propusieron una modificación al método BFGS (MBFGS) y probaron que el método MBFGS con búsqueda lineal de Armijo o Wolfe convergía globalmente incluso para la minimización no-convexa. Pero no se sabe si el método MBFGS con búsqueda lineal no monótona como (2) converge para funciones objetivo no-convexa.

El propósito de este trabajo es el análisis y estudio de los trabajos realizados en [18]. La estructura del trabajo es la siguiente: En el capítulo 1 se estudiarán las estrategias de búsqueda lineal tanto monótonas como no-monótonas. Luego, en el capítulo 2 presentamos algunas propiedades de los métodos Cuasi-Newton, en particular la fórmula BFGS. En el capítulo 3 se presenta una alternativa del método BFGS, el método BFGS modificado propuesto en [7]. En el capítulo 4 presentamos resultados numéricos obtenidos por nuestra implementación del algoritmo propuesto en [18] y la comparación con el BFGS original.

CAPÍTULO 1

ESTRATEGIAS DE BÚSQUEDA LINEAL

Cada iteración de un método de búsqueda lineal calcula una dirección de búsqueda y luego decide que tan lejos puede moverse a lo largo de esa dirección. La iteración viene dada por

$$x_{k+1} = x_k + \alpha_k d_k \quad (1.1)$$

donde el escalar positivo α_k se denomina longitud o tamaño de paso. El éxito de un método de búsqueda lineal depende de las opciones eficaces de la dirección d_k y la longitud del paso α_k .

La mayoría de los algoritmos de búsqueda línea requieren que d_k sea una dirección descenso es decir que el vector d_k cumpla la siguiente proposición: $\exists \bar{\delta} > 0$ tal que

$$f(x_k + \alpha d_k) < f(x_k) \quad \forall \alpha \in (0, \bar{\delta}).$$

porque esta propiedad garantiza que la función f pueda ir disminuyendo a medida que avanza a lo largo de esta dirección. Esta dirección de búsqueda, a menudo tiene la forma

$$d_k = -B_k^{-1} f_k, \quad (1.2)$$

donde B_k es una matriz simétrica y definida positiva. Cuando d_k es definida por (1.2) y B_k es definida positiva, tenemos

$$d_k^T \nabla f_k = -\nabla f_k^T B_k^{-1} \nabla f_k < 0,$$

y por lo tanto d_k es una dirección de descenso. En el método de descenso más rápido, B_k no es más que la matriz identidad I_n , mientras que en el método de Newton, B_k es la exactamente el Hessiano $\nabla^2 f_k$.

En los métodos Cuasi-Newton, B_k es una aproximación a la Hessiana que se actualiza en cada iteración a través de una fórmula de bajo rango.

En el cálculo de la longitud del paso, nos encontramos ante una disyuntiva. nos gustaría escoger α_k que de una reducción sustancial de la función f , pero al mismo tiempo, no queremos gastar demasiado tiempo en tomar la decisión. La opción ideal sería el minimizador global de la función univariante $\phi(\cdot)$, definida por

$$\phi(\alpha) = f(x_k + \alpha d_k), \alpha > 0, \quad (1.3)$$

pero, en general, es demasiado caro para identificar a este valor (ver figura 1.1). Para encontrar incluso un minimizador local de f con precisión moderada generalmente se requiere demasiadas evaluaciones de la función objetivo f y, posiblemente, el gradiente ∇f . (ver [16]).

Estrategias más prácticas realizan una búsqueda lineal inexacta para identificar una longitud de paso que logre reducciones adecuadas en f a un costo mínimo. Típicamente los algoritmos de búsqueda lineal prueban con una sucesión de valores candidatos para α , deteniéndose para aceptar uno de estos valores cuando se cumplen ciertas condiciones. La búsqueda lineal realiza dos etapas: Una fase de soporte donde encuentra un intervalo que contiene longitudes de paso deseables, y una bisección o fase de la interpolación donde calcula una longitud del paso buena dentro de ese intervalo (ver[16]).

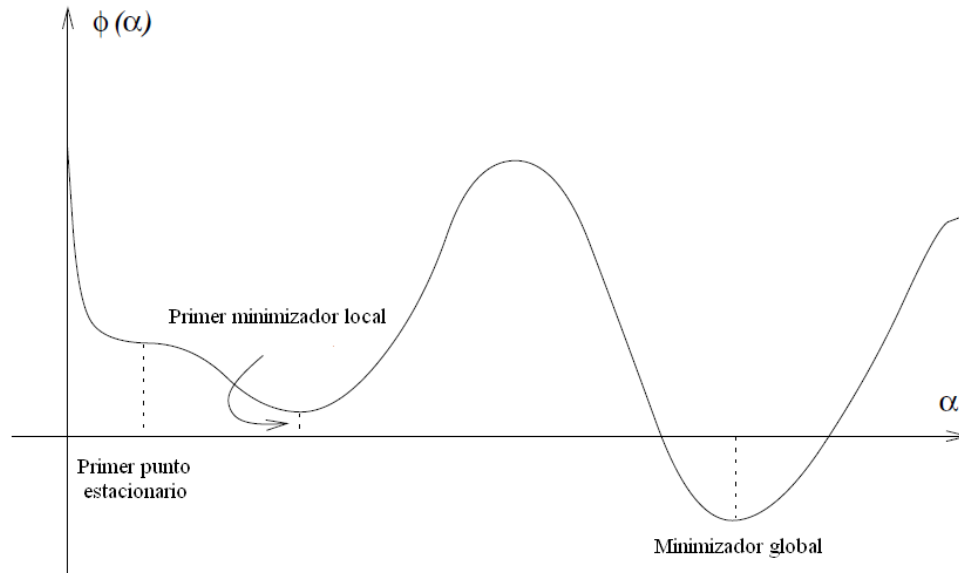


Figura 1.1: La ideal longitud del paso es un minimizador global.

Ahora analizaremos las diversas condiciones de terminación de los algoritmos de búsqueda lineal y mostrar que la longitud de paso efectiva no necesariamente se encuentran cerca de los minimizadores de la función univariante $\phi(\alpha)$ definida en (1.3). Una condición simple que podría imponerse a α_k es exigir una reducción de la función f es decir $f(x_k + \alpha_k d_k) < f(x_k)$. Este requisito no es suficiente para producir la convergencia a x^* ilustrado en la Figura (1.2), para la cual el valor mínimo de la función $f = (x - c)^2 - 1$ con $c \geq 4$ es $f^* = -1$, pero una sucesión de iterados x_k para la cual $f(x_k) = \frac{5}{k}$, $k = 1, \dots$ produce una disminución en cada iteración, pero tiene un valor de la función límite de cero. La reducción insuficiente en f en cada paso hace que no convergen al minimizador de esta función convexa. Para evitar este comportamiento es necesario hacer cumplir una condición de disminución suficiente, un concepto que discutiremos a continuación.

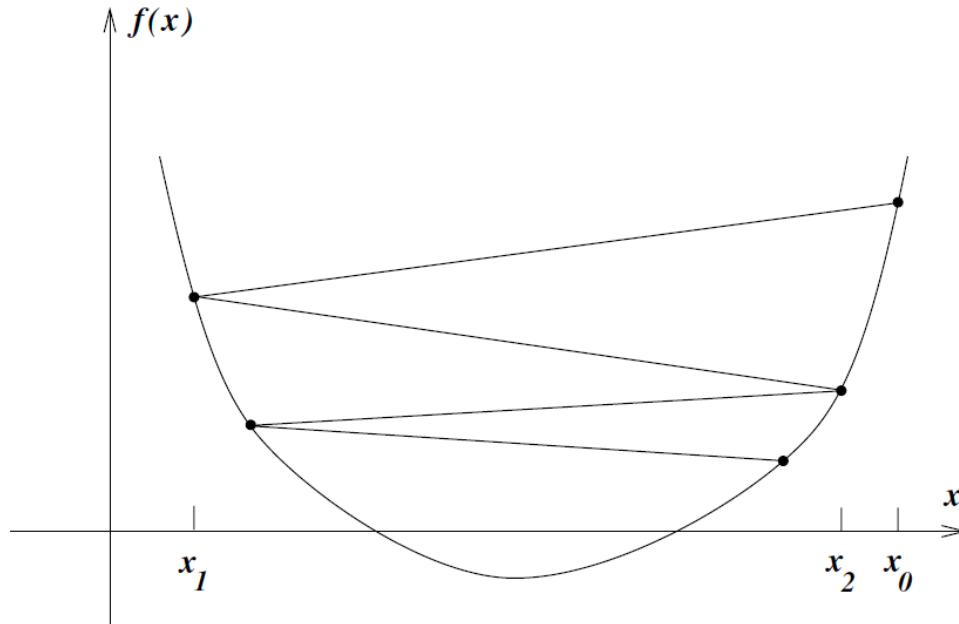


Figura 1.2: Reducción insuficiente en f .

Una popular búsqueda lineal inexacta establece la condición que α_k debe dar en primer lugar una disminución suficiente en la función objetivo f , según medida por la siguiente desigualdad:

$$f(x_k + \alpha d_k) \leq f(x_k) + c_1 \alpha \nabla f_k^T d_k \quad (1.4)$$

para alguna constante $c_1 \in (0, 1)$. En otras palabras, la reducción en f debe ser proporcional tanto a la longitud de paso α_k como a la derivada direccional $\nabla f_k^T d_k$. La desigualdad (1.4) se conoce en la literatura como la condición de Armijo.

La condición de disminución suficiente se ilustra en la Figura (1.3). El lado derecho de (1.4), que es una función lineal, la denotaremos por $l(\alpha)$. La función $l(\cdot)$ tiene pendiente negativa $c_1 \nabla f_k^T d_k$, pero debido a $c_1 \in (0, 1)$, se encuentra por encima de la gráfica de ϕ para pequeños valores positivos de α . La condición de suficiente disminución establece que α es aceptable solo si $\phi(\alpha) \leq l(\alpha)$. Los intervalos en los que se cumple esta condición se muestra en la Figura (1.3). En la práctica, c_1 se elige pequeño, digamos $c_1 = 10^{-4}$.

La condición suficiente de disminución no es suficiente por sí sola para asegurar que el algoritmo haga un progreso razonable, porque como se ve en la Figura (1.3), se cumple para todos los valores suficientemente pequeños de α . Para descartar los pasos inaceptablemente cortos introducimos un segundo requisito, llamado la condición de curvatura que requiere que α_k satisfaga

$$\nabla f(x_k + \alpha_k d_k)^T d_k \geq c_2 \nabla f_k^T d_k \quad (1.5)$$

para alguna constante $c_2 \in (c_1, 1)$, donde c_1 es la constante de la condición de Armijo. Note que el lado izquierdo es simplemente la derivada $\phi'(\alpha_k)$, así la condición de curvatura asegura que la pendiente de ϕ en α_k es mayor que c_2 veces la pendiente inicial $\phi'(0)$. Esto tiene sentido porque si la pendiente $\phi'(\alpha)$ es muy negativa,

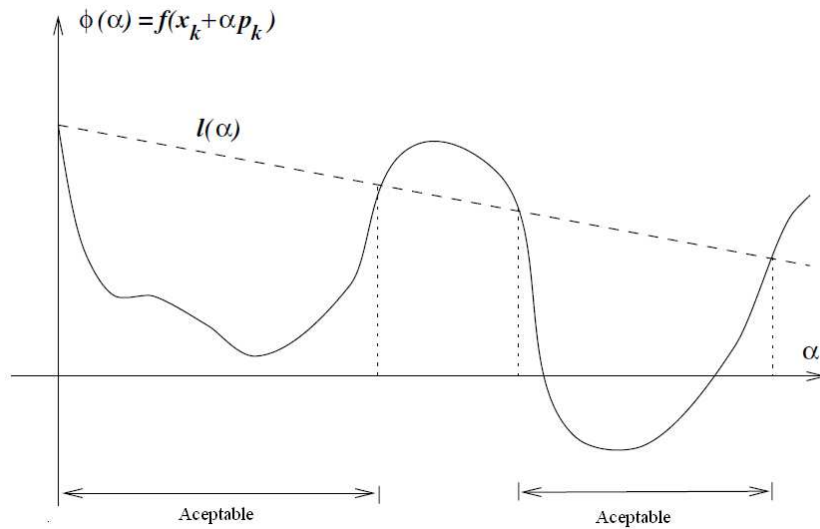


Figura 1.3: Condición de disminución suficiente.

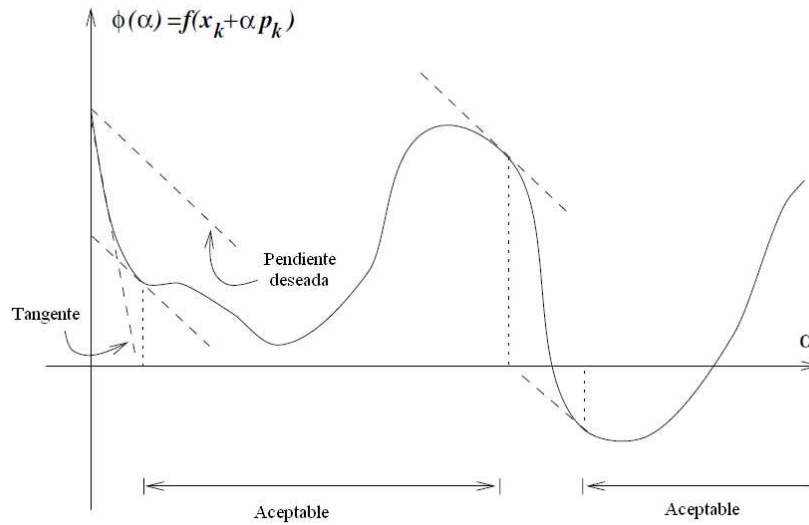


Figura 1.4: Condición de curvatura.

tenemos indicación de que podemos reducir significativamente a f moviéndonos más a lo largo de la dirección elegida.

Por otro lado, si $\phi'(\alpha_k)$ es ligeramente negativa o incluso positiva, es una señal de que no podemos esperar que la función disminuya mucho más en esta dirección, así que tiene sentido terminar la búsqueda lineal. La condición de curvatura es ilustrada en la Figura (1.4). Los valores típicos de c_2 son de 0.9 cuando la dirección de búsqueda d_k es escogida por el método de Newton o por el método Cuasi-Newton, y 0.1 cuando d_k es obtenida del método no lineal de gradiente conjugado (ver [16]).

La condición de Armijo y la condición de curvatura son colectivamente conocidas como las condiciones de Wolfe. Vamos a ilustrarlas en la Figura 3.5 y reunir las aquí para referencias futuras:

$$f(x_k + \alpha d_k) \leq f(x_k) + c_1 \alpha \nabla f_k^T d_k \quad (1.6a)$$

$$\nabla f(x_k + \alpha_k d_k)^T d_k \geq c_2 \nabla f_k^T d_k \quad (1.6b)$$

con $0 < c_1 < c_2 < 1$.

Una longitud de paso puede satisfacer las condiciones de Wolfe sin estar particularmente cerca de un minimizador de ϕ , como se muestra en la Figura (1.5). Podemos sin embargo

modificar la condición de curvatura para forzar a α_k a colocarse en al menos una amplia vecindad de un minimizador local o punto estacionario de ϕ . Las condiciones fuertes de Wolfe requieren que α_k satisfagan

$$f(x_k + \alpha d_k) \leq f(x_k) + c_1 \alpha \nabla f_k^T d_k \quad (1.7a)$$

$$|\nabla f(x_k + \alpha_k d_k)^T d_k| \geq c_2 |\nabla f_k^T d_k| \quad (1.7b)$$

con $0 < c_1 < c_2 < 1$. La única diferencia con las condiciones de Wolfe es que ya no permitir que el derivada $\phi'(\alpha_k)$ sea demasiado positiva. Por lo tanto se excluyen los puntos que están lejos de los puntos estacionarios de ϕ .

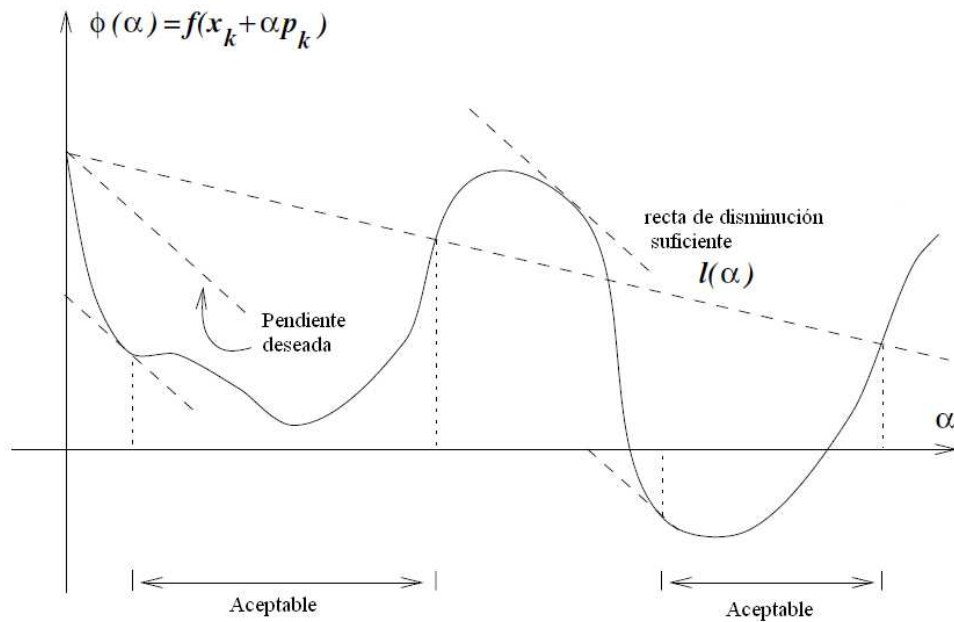


Figura 1.5: La longitud de paso satisfaciendo las condiciones de Wolfe.

No es difícil probar que existe una longitud de paso que satisface las condiciones de Wolfe para toda función f que es suave y acotada inferiormente.

Lema 1.1. (ver[19]) *Supongamos que $f : \mathbb{R}^n \rightarrow \mathbb{R}$ es continuamente diferenciable. Sea d_k una dirección de descenso en x_k , y asumamos que f es acotada inferiormente a lo largo de $\{x_k + \alpha_k d_k | \alpha > 0\}$. Entonces si $0 < c_1 < c_2 < 1$, existen intervalos de longitud de pasos que satisfacen la condiciones de Wolfe (1.6a, 1.6b) y las condiciones fuertes de Wolfe (1.7a, 1.7b).*

Demostración. Note que $\phi(\alpha) = f(x_k + \alpha d_k)$ es acotada inferiormente para todo $\alpha > 0$. Dado $0 < c_1 < 1$, la recta $l(\alpha) = f(x_k) + \alpha c_1 \nabla f_k^T d_k$ no es acotada inferiormente y debe por lo tanto intersectar al gráfico de ϕ al menos una vez. Sea $\alpha' > 0$ el valor mas pequeño de intersección de α , es decir,

$$f(x_k + \alpha' d_k) = f(x_k) + \alpha' c_1 \nabla f_k^T d_k \quad (1.8)$$

ahora las condiciones de disminución suficiente (1.6a) se cumplen claramente para toda longitud de paso menor que α' .

Luego por el teorema del valor medio, existe $\alpha'' \in (0, \alpha')$ tal que

$$f(x_k + \alpha' d_k) - f(x_k) = \alpha' \nabla f(x_k + \alpha'' d_k)^T d_k \quad (1.9)$$

de allí combinando (1.8) y (1.9), obtenemos

$$\nabla f(x_k + \alpha'' d_k)^T d_k = c_1 \nabla f_k^T d_k > c_2 \nabla f_k^T d_k, \quad (1.10)$$

donde $c_1 < c_2$ y $\nabla f_k^T d_k < 0$. Por tanto, α'' satisface las condiciones de Wolfe (1.6a, 1.6b), y se cumplen ambas desigualdades estrictas. Por lo tanto, por lo supuesto de la suavidad de f , hay un intervalo de alrededor de α'' para el que las condiciones Wolfe se cumplen. Además, dado que el termino del lado izquierdo de (1.10) es negativo, las condiciones fuertes de Wolfe (1.7a, 1.7b) se mantienen en el mismo intervalo. ■

Las condiciones de Wolfe son invariante en escala en un sentido amplio: se multiplica la función objetivo por una constante o se hace un cambio de variables afín para no modificarlas. Se pueden utilizar en la mayoría de los métodos de búsqueda lineal, y son

particularmente importantes en la aplicación de los métodos Cuasi-Newton.

Al igual que las condiciones de Wolfe, las condiciones Goldstein aseguran que la longitud de paso α alcance una disminución suficiente, pero no demasiado corta. Las condiciones Goldstein pueden establecerse como un par de desigualdades, de la siguiente manera:

$$f(x_k) + (1 - c)\alpha_k \nabla f_k^T d_k \leq f(x_k + \alpha_k d_k) \leq f(x_k) + c\alpha_k \nabla f_k^T d_k, \quad (1.11)$$

con $0 < c < \frac{1}{2}$. La segunda desigualdad es la condición de disminución suficiente (1.4), mientras que la primera desigualdad se introduce para controlar la longitud del paso desde abajo; como se ve en la Figura (1.6).

Una desventaja de las condiciones Goldstein en comparación con las condiciones Wolfe es que la primera desigualdad en (1.11) puede excluir a todos los minimizadores de ϕ . Sin embargo, las condiciones de Goldstein y Wolfe tienen mucho en común, y sus teorías de convergencia son muy similares.

Las condiciones de Goldstein se utilizan a menudo en los métodos de tipo Newton pero no son muy adecuadas para métodos Cuasi-Newton que mantienen una aproximación del hessiano definido positivo.

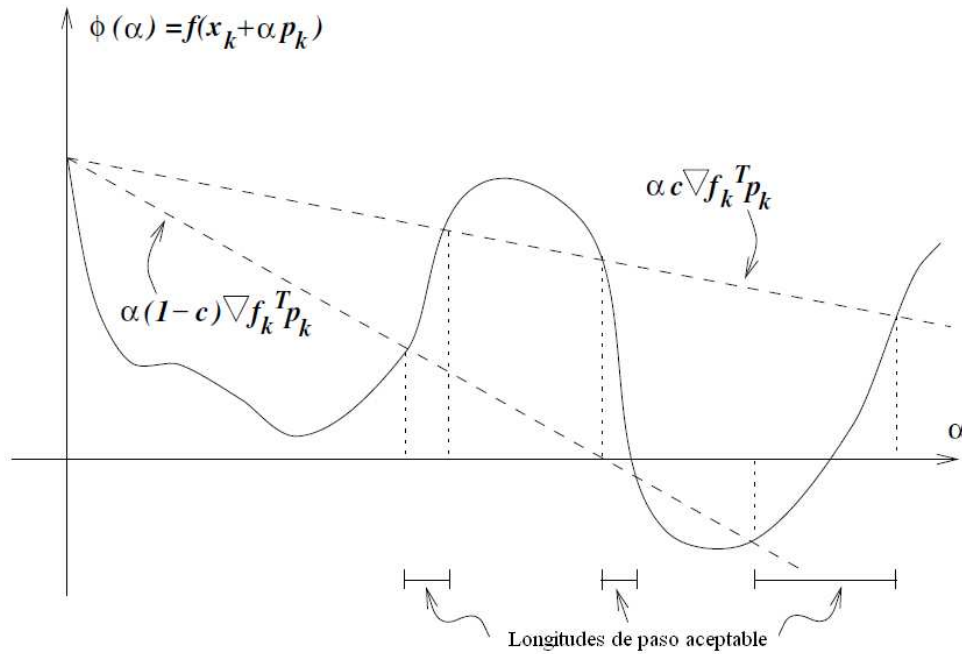


Figura 1.6: Condiciones de Goldstein.

El método de búsqueda lineal no-monótona es una nueva técnica para resolver problema de optimización. Esta relaja el rango de la búsqueda lineal y encuentra un mayor tamaño de paso en cada iteración, a fin de evitar posibles minimizadores locales y salir de la región que los contenga. También es muy útil para encontrar el minimizador global de problemas de optimización.

Los métodos de búsqueda lineal han sido investigado por numerosos autores (ver [17]). De hecho el método Barzilai-Borwein (ver [17]) es el método de descenso no-monótono. La búsqueda lineal no-monótona no necesariamente satisface la condición $f(x_{k+1}) < f(x_k)$, $k = 0, 1, 2, \dots$ en cada iteración; como resultado, es muy utilizado en el caso que la sucesión de iterados recorre inferiormente la región curva donde se encuentren los posibles minimizadores, algo común en la práctica de problemas de programación no-lineal.

Sin embargo autores como Sun y otros (ver [17]) estudiaron los métodos generales de la búsqueda lineal no-monótona mediante el uso inverso de módulos continuos, lo cual

se describe a continuación.

- **Búsqueda lineal no-monótona Armijo.** Dado una constante $a > 0$, $\delta, \rho \in (0, 1)$ y un entero no negativo M , para cada k $h(k)$ cumple $h(0) = 0$, $0 \leq h(k) \leq \min[m(k-1), M]$, $\forall k \geq 1$. seleccionar el tamaño del paso $\alpha_k = \max\{a\rho^0, a\rho^1, \dots\}$ satisfaciendo

$$f(x_k + \rho^m a d_k) \leq \max_{0 \leq j \leq M} f(x_{k-j}) + \delta \rho^m a g_k^T d_k.$$

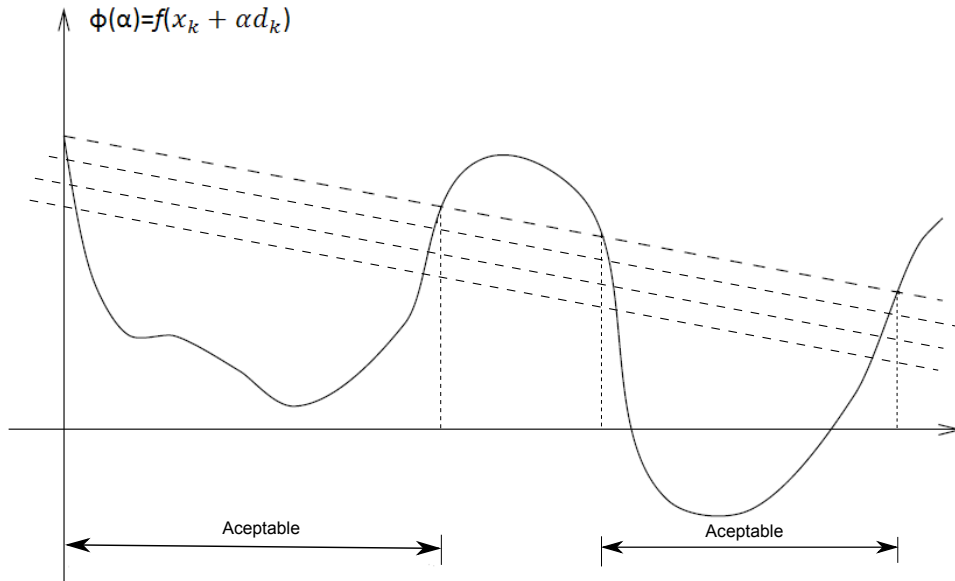


Figura 1.7: Condición de Armijo no-monótona

- **Búsqueda lineal no-monótona Goldstein.** Dado $\delta \in (0, \frac{1}{2})$ se escoge α_k satisfaciendo

$$\begin{aligned} f(x_k + \rho^m a d_k) &\leq \max_{0 \leq j \leq M} f(x_{k-j}) + \delta \rho^m a g_k^T d_k, \\ f(x_k + \rho^m a d_k) &\geq \max_{0 \leq j \leq M} f(x_{k-j}) + (1 - \delta) \rho^m a g_k^T d_k. \end{aligned}$$

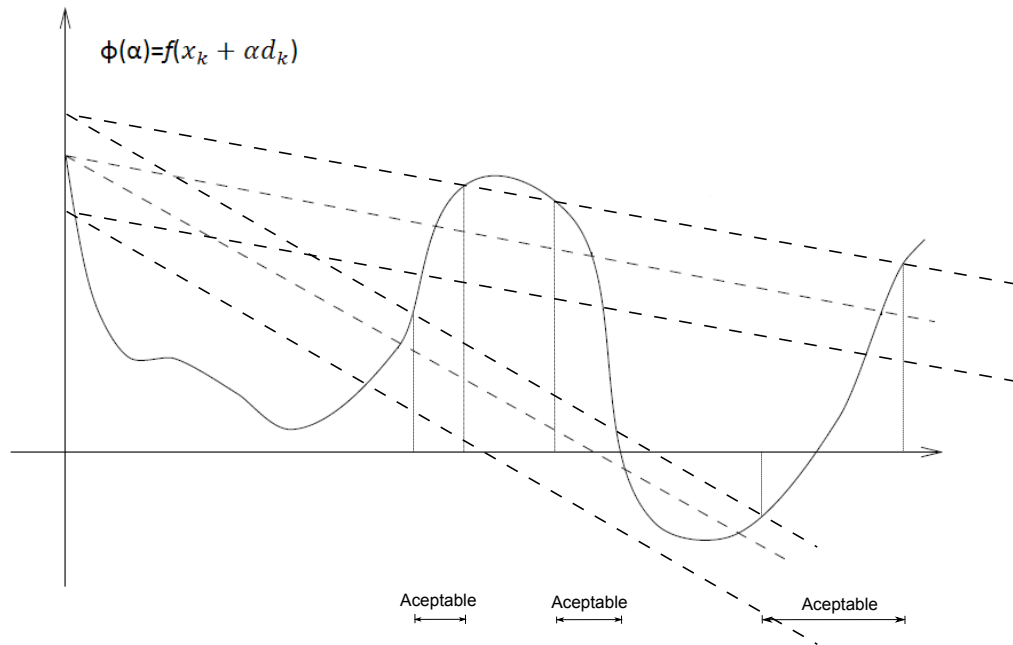


Figura 1.8: Condición de Goldstein no-monótona

- Búsqueda lineal no-monótona Wolfe. Dado $\delta \in (0, \frac{1}{2})$, y $\beta \in (0, 1)$, se escoge α_k que cumpla

$$f(x_k + \rho^m a d_k) \leq \max_{0 \leq j \leq M} f(x_{k-j}) + \delta \rho^m a g_k^T d_k,$$

$$\nabla f(x_k + \alpha_k d_k)^T d_k \geq \beta \nabla f_k^T d_k.$$

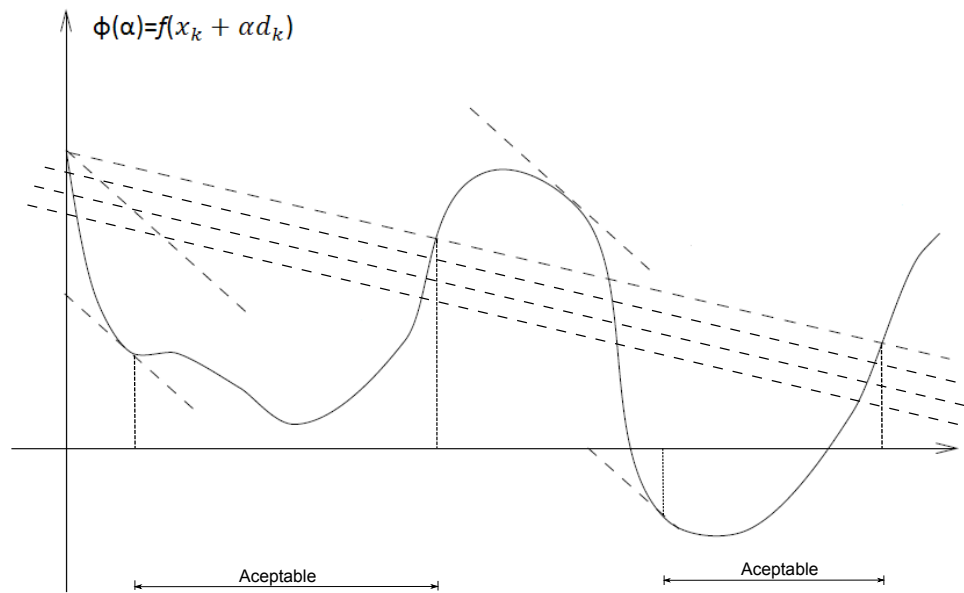


Figura 1.9: Condición de Wolfe no-monótona

Sun (ver [17]) estableció la convergencia de los tres métodos de búsqueda lineal no-monótona.

CAPÍTULO 2

MÉTODOS CUASI-NEWTON

A mediados de 1950, W.C Davidon un físico que trabajaba en el laboratorio nacional de argonne, estaba usando el método de descenso de coordenadas (ver [16]) para llevar a cabo un cálculo de optimización largo. En esa época las computadoras no eran muy estables, y era muy frustrante para Davidon, ver como el sistema de computadoras se colgaba antes de terminar los cálculos. Así Davidon decidió buscar la manera acelerar los cálculos en cada iteración. El algoritmo que desarrolló el primer algoritmo Cuasi-Newton resultó ser una de las ideas más creativas en optimización no lineal. Pronto se demostró por Fletcher y Powell que el nuevo algoritmo era mucho más rápido y más fiable que los otros métodos existentes, y este dramático avance transformo la optimización no lineal de la noche a la mañana. Durante los siguientes veinte años, numerosas variantes fueron propuestos y cientos de artículos se dedicaron a su estudio. Una interesante historia irónica es que el artículo de Davidon no fue aceptado para su publicación, sino que se mantuvo como un informe técnico por más de treinta años hasta que apareció en la primera edición de la revista SIAM optimización en el año 1991.

Los métodos Cuasi-Newton, como el descenso más rápido (ver[16]) toma el gradiente de la función objetivo en cada iteración. Al medir los cambios en los gradientes, construyen un modelo a la función objetivo que es bueno para generar la convergencia superlineal. Las ventajas del método descenso más rápido son considerables, sobre todo

en problemas de gran dificultad (ver[16]). Además, dado que las segundas derivadas no son necesarias, los métodos Cuasi-Newton son a veces más eficiente que el método de Newton. Hoy en día, las librerías de software de optimización contienen una variedad de algoritmos Cuasi-Newton para la resolución de problemas de optimización sin restricciones, con restricciones, y de gran escala. El desarrollo de las técnicas de diferenciación automática ha hecho posible el uso del método de Newton sin que los usuarios deban proporcionar segundas derivadas; sin embargo, la herramienta de diferenciación automática pueden no ser aplicable en muchas situaciones, y puede ser mucho más costoso que trabajar con derivadas segundas en diferenciación automática que con el gradiente. Por estas razones, los métodos cuasi-Newton son muy utilizados.

El más popular de los algoritmos Cuasi-Newton es el método BFGS, conocido así por sus descubridores Broyden, Fletcher, Goldfarb, y Shanno. Obtendremos ahora este algoritmo (y su pariente cercano el algoritmo DFP) y describir sus propiedades teóricas. Comenzamos para obtener el algoritmo se establece el siguiente modelo cuadrático de la función objetivo en la iteración actual de x_k :

$$m_k(p) = f_k + \nabla f_k^T p + \frac{1}{2} p^T B_k p. \quad (2.1)$$

Aquí B_k es una matriz $n \times n$ simétrica definida positiva, que se actualiza en cada iteración. Notese que el valor de la función y el gradiente de este modelo partiendo en $p = 0$ son f_k y ∇f_k^T respectivamente. El minimizador p_k de este modelo cuadrático convexo, el cual puede ser escrito explícitamente como

$$p_k = -B_k^{-1} \nabla f_k, \quad (2.2)$$

se utiliza como la dirección de búsqueda, y es la nueva iteración

$$x_{k+1} = x_k + \alpha_k p_k, \quad (2.3)$$

donde la longitud de paso α_k es escogido de tal manera que cumple las condiciones de Wolfe(1.6a,1.6b). Esta iteración es muy similar al método de búsqueda lineal Newton; pero la diferencia clave está en que la aproximación al Hessiano B_k es usada en lugar de el Hessiano verdadero.

En lugar de calcular nuevamente B_k en cada iteración, Davidon propuso actualizarla de una manera sencilla para tener en cuenta la curvatura medida durante la etapa más reciente. Supongamos que hemos generado una nueva iteración x_{k+1} y deseamos de construir un nuevo modelo cuadrático, de la forma

$$m_{k+1}(p) = f_{k+1} + \nabla f_{k+1}^T p + \frac{1}{2} p^T B_{k+1} p.$$

¿Qué requisitos debemos imponer a B_{k+1} , basándose en los conocimientos adquiridos durante el último paso? Un requisito razonable es que el gradiente de m_{k+1} debe coincidir con el gradiente de la función objetivo f en las últimas dos iteraciones x_k y x_{k+1} . Dado $\nabla m_{k+1}(0)$ es precisamente ∇f_{k+1} , la segunda de esas condiciones es satisfecha automáticamente. La primera condición puede ser escrita matemáticamente como

$$\nabla m_{k+1}(-\alpha_k p_k) = \nabla f_{k+1} - \alpha_k B_{k+1} p_k = \nabla f_k.$$

Reordenando, se obtiene

$$B_{k+1} \alpha_k p_k = \nabla f_{k+1} - \nabla f_k. \quad (2.4)$$

Para simplificar la notación definimos los vectores

$$s_k = x_{k+1} - x_k = \alpha_k p_k, y_k = \nabla f_{k+1} - \nabla f_k \quad (2.5)$$

así de (2.4) se tiene que

$$B_{k+1} s_k = y_k. \quad (2.6)$$

La expresión anterior es conocida como la ecuación de la secante.

Dado el desplazamiento α_k y el cambio de los gradientes y_k , la ecuación de la secante requiere que la matriz B_{k+1} sea definida positiva vía s_k en y_k . Esto solo sería posible si s_k y y_k satisfacen la condición de curvatura

$$s_k^T y_k > 0 \quad (2.7)$$

como es fácil de ver al multiplicar (2.6) por s_k^T . Cuando f es fuertemente convexa, la desigualdad (2.7) será satisfecha por cualquier par de puntos x_k y x_{k+1} . Sin embargo esta condición no siempre se cumple para funciones no-convexas, y en este caso se necesita para hacer cumplir explícitamente (2.7) mediante la imposición de restricciones en el procedimiento de elección de la longitud de paso α en la búsqueda lineal. De hecho la condición (2.7) esta garantizada que se cumpla si imponemos las condiciones de Wolfe (1.6a),(1.6b) o las condiciones fuertes de Wolfe (1.7a),(1.7b) en la búsqueda lineal. Para verificar esta afirmación, notemos de (2.5) y (1.6b) que $\nabla f_{k+1}^T s_k \geq c_2 \nabla f_k^T s_k$, y por tanto

$$y_k^T s_k \geq (c_2 - 1)\alpha_k \nabla f_k^T p_k. \quad (2.8)$$

Dado que $c_2 < 1$ y p_k es una dirección de descenso, el termino del lado derecho es positivo, y la condición de curvatura (2.7) se cumple.

Cuando la condición de curvatura es satisfecha, la ecuación de la secante (2.6) siempre tiene una solución B_{k+1} . De hecho ésta admite un numero infinito de soluciones, dado los $n(n+1)/2$ grados de libertad en una matriz simétrica definida positiva excediendo las n condiciones impuesta por la ecuación de la secante. El requisito de definible positivo impone n desigualdades adicionales-todos los menores principales debe ser positivo-pero estas condiciones no absorben los restantes grados de libertad. Para determinar B_{k+1} únicamente, imponemos la condición adicional de que, entre todas las matrices simétricas que satisface la ecuación secante, B_{k+1} es, en cierto sentido, más cercana a la actual matriz B_k . En otras palabras, podemos resolver el problema

$$\min_B \|B - B_k\| \quad (2.9a)$$

$$\text{sujeeto a } B = B^T, B s_k = y_k, \quad (2.9b)$$

donde s_k y y_k satisfacen (2.7) y B_k es simétrica y definida positiva. Se pueden utilizar diferentes normas en (2.9a), y cada norma da lugar a un método diferente Cuasi-Newton. Una norma que permite una solución fácil del problema de minimización (2.9a) y da lugar a una escala invariante en el método de optimización, es la norma de Frobenius ponderado

$$\|A\|_W \equiv \|W^{\frac{1}{2}} A W^{\frac{1}{2}}\|_F \quad (2.10)$$

donde $\|\cdot\|_F$ es definida por $\|C\|_F^2 = \sum_{j=1}^n \sum_{j=1}^n c_{ij}^2$. La matriz de poderación W puede ser elegida como cualquier matriz que satisface la relación $W y_k = s_k$. Para ser concretos,

el lector puede suponer que $W = \bar{G}_k^{-1}$ donde \bar{G}_k es el hessiano promedio definido por

$$\bar{G}_k = \left[\int_0^1 \nabla^2 f(x_k + \tau \alpha_k p_k) d\tau \right]. \quad (2.11)$$

La propiedad

$$y_k = \bar{G}_k \alpha_k p_k = \bar{G}_k s_k \quad (2.12)$$

se sigue del teorema de Taylor's, (ver [16]) el cual establece que:

Teorema 2.1. *supongamos que $f : \mathbb{R}^n \rightarrow \mathbb{R}$ es continuamente diferenciable y que $p \in \mathbb{R}^n$. Entonces tenemos que*

$$f(x + p) = f(x) + \nabla f(x + tp)^T p,$$

para algún $t \in (0, 1)$. Mas aun, si es dos veces diferenciable, tenemos que

$$\nabla f(x + t) = \nabla f(x) + \int_0^1 \nabla^2 f(x + tp) p dt,$$

y que

$$f(x + p) = f(x) + \nabla f(x)^T p + \frac{1}{2} p^T \nabla^2 f(x + tp) p,$$

para algún $t \in (0, 1)$.

Con esta elección de la matriz de ponderación W , la norma (2.10) es no dimensional, que es una propiedad deseable, ya que no deseamos que la solución de (2.9a) dependa de las unidades del problema.

Con esta matriz de ponderación y esta norma, la única solución de (2.9a,2.9b) es

$$(DFP) \quad B_{k+1} = (I - p_k y_k s_k^T) B_k (I - p_k s_k y_k^T) + p_k y_k y_k^T, \quad (2.13)$$

con

$$p_k = \frac{1}{y_k^T s_k}. \quad (2.14)$$

La formula es llamada la formula de actualización DFP ya que es originalmente propuesta por Davidon en 1959, y estudios subsecuentes, implementado, y popularizado

por Fletcher y Powell.

La inversa de B_k , la cual denotaremos por

$$H_k = B_k^{-1},$$

es útil en la implementación del método, ya que permite que la dirección de búsqueda (2.2) se calcule por medio de una simple multiplicación matriz-vector. Usando la fórmula de Sherman-Morrison-Woodbury (ver [16]) la cual establece que:

Lema 2.1. *Si la matriz A es no singular, cuadrada y si A se somete a una actualización de rango uno se convierte en $\bar{A} = A + ab^T$, donde $a, b \in R^T$, entonces si \bar{A} es no singular, obtenemos*

$$\bar{A}^{-1} = A^{-1} - \frac{A^{-1}ab^T A^{-1}}{1 + b^T A^{-1}a}.$$

Se obtiene la siguiente expresión para la actualización de la aproximación de la inversa del hessiano H_k que corresponde a la actualización DFP de B_k en (2.13):

$$(DFP) \quad H_{k+1} = H_k - \frac{H_k y_k y_k^T H_k}{y_k^T H_k y_k} + \frac{s_k s_k^T}{y_k^T s_k}. \quad (2.15)$$

Note que los dos últimos términos en el lado derecho de (2.15) son matrices de rango 1, así que H_k se somete a una modificación de rango 2. Es fácil ver que (2.13) es también una modificación de rango 2 de B_k . Esta es la idea fundamental de la actualización Cuasi-Newton: En vez de recalcular la aproximación hessiana (o la inversa hessiana) a partir de cero en cada iteración, aplicamos una simple modificación que combina la información más recientemente observado sobre la función objetivo con el conocimiento existente incorporada en nuestra aproximación hessiana actual.

La fórmula de actualización DFP es bastante eficaz, pero pronto fue sustituida por la fórmula BFGS, la cual se considera actualmente que es de todas las fórmulas de actualización Cuasi-Newton la más efectiva (ver [16]). La actualización BFGS puede obtenerse de hacer un simple cambio en el argumento que dio lugar a (2.13).

En lugar de imponer condiciones en la aproximación Hessiana B_k , imponemos condiciones similares en su inversa H_k . La aproximación a la actualización H_{k+1} debe ser

simétrica y definida positiva, y debe satisfacer la ecuación de la secante (2.6), ahora la escribimos como

$$H_{k+1}y_k = s_k.$$

La condición de proximidad a H_k ahora se especifica por el siguiente análogo de (2.9a,2.9b):

$$\min_H \|H - H_k\| \quad (2.16a)$$

$$\text{sujeeto a } H = H^T, Hy_k = s_k. \quad (2.16b)$$

La norma usada es la de Frobenius ponderada descrita anteriormente, donde la matriz de ponderación W es ahora cualquier matriz que satisfice $W s_k = y_k$. Para ser concretos, supongamos de nuevo que W es dada por la hessiana promedio definida en (2.11), la única solución H_{k+1} a (2.16a,2.16b) es dada por

$$(BFGS) \quad H_{k+1} = (I - p_k s_k y_k^T) H_k (I - p_k y_k s_k^T) + p_k s_k s_k^T \quad (2.17)$$

con p_k definida por (2.14). Sólo un problema tiene que ser resuelto antes de que podamos definir un algoritmo BFGS completo: ¿como debe elegirse la aproximación inicial H_0 ? Desafortunadamente, no existe una fórmula mágica que funciona bien en todos los casos. Podemos utilizar la información específica sobre el problema, por ejemplo, si se establece en el inverso del aproximado del Hessiano calculado por diferencias finitas en x_0 . De lo contrario, simplemente se puede configurar para ser la matriz identidad, o un múltiplo de la matriz de identidad, donde se elija el múltiplo para reflejar la escala de las variables.

Algoritmo 2.1. [método BFGS]

- PASO 1: Tome un punto inicial x_0 , con tolerancia de convergencia $\epsilon > 0$, la aproximación inversa al Hessiano H_0 ;
- PASO 2: $k \rightarrow 0$;
- PASO 3: haga mientras $\|\nabla f_k\| > \epsilon$;
- PASO 4: calcule la dirección de búsqueda $p_k = -H_k \nabla f_k$;

- PASO 5: Sea $x_{k+1} = x_k + \alpha_k p_k$ donde α_k se calcula a partir del procedimiento de búsqueda lineal que satisface las condiciones de Wolfe ; (1.6a),(1.6b)
- PASO 6: defina $s_k = x_{k+1} - x_k$ y $y_k = \nabla f_{k+1} - \nabla f_k$;
- PASO 7: calcule H_{k+1} mediante (2.15);
- PASO 8: $k \leftarrow k + 1$;
- PASO 9: fin(haga mientras)

Cada iteración se puede realizar a un costo de $O(n^2)$ operaciones aritméticas (más el costo de evaluaciones de la función y el gradiente); no hay $O(n^3)$ operaciones aritméticas tales como el resolver sistemas lineales o operaciones de la matriz-matriz. El algoritmo es robusto, y su velocidad de convergencia es superlineal, que es lo suficientemente rápido como para la mayoría de los propósitos prácticos. A pesar de que el método de Newton converge más rápidamente (es decir, en forma cuadrática), su coste por iteración generalmente es más alta, porque tiene la necesidad de segundas derivadas y la solución de un sistema lineal. Podemos obtener una versión del algoritmo BFGS que trabaja con la aproximación de Hessiana B_k en lugar de H_k . La fórmula de actualización para B_k se obtiene mediante la simple aplicación de la formula Sherman-Morrison-Woodbury a (2.17) para obtener

$$(BFGS) \quad B_{k+1} = B_k - \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k} + \frac{y_k y_k^T}{y_k^T s_k} \quad (2.18)$$

Una aplicación sencilla de esta variante no es eficiente para la minimización sin restricciones, debido a que se requiere que el sistema $B_k p_k = -\nabla f_k$ sea resuelto para el paso p_k , lo que aumenta el coste computacional del paso a $O(n^3)$. Sin embargo, se tiene aplicaciones menos costosas de esta variante mediante la actualización de los factores de Cholesky (ver [16]) de B_k .

Para cualquier vector z no-nulo, tenemos

$$z^T H_{k+1} z = w^T H_k w + p_k (z^T s_k)^2 \geq 0,$$

donde definimos $w = z - p_k y_k (s_k^T z)$. El lado derecho puede ser cero sólo si $s_k^T z = 0$, pero en este caso $w = z \neq 0$, lo cual implica que el primer termino es mayor que cero.

Por tanto, H_{k+1} es definida positiva.

Para hacer la fórmula de actualización Cuasi-Newton invariante a las transformaciones en las variables (tales como transformaciones de escala), es necesario para los objetivos (2.9a) y (2.16a) ser invariantes bajo esas mismas transformaciones. La elección de la matriz de ponderación W utilizada para definir las normas en (2.9a) y (2.16a) asegura que se cumple esta condición. Muchas otras opciones de la matriz de ponderación W son posibles, cada uno de ellos dando una fórmula de actualización diferente. Sin embargo, a pesar de intensas investigaciones, no hay una fórmula que se halla encontrado que sea significativamente más eficaz que BFGS. Es razonable preguntarse si hay situaciones en las que la fórmula de actualización tales como (2.17) puede producir malos resultados. Si en alguna iteración la matriz H_k se convierte en una pobre aproximación a la verdadera Hessiana inversa, ¿hay alguna posibilidad de corregirlo? Por ejemplo cuando el producto interno $y_k^T s_k$ es pequeño (pero positivo), entonces se sigue de (2.14) y (2.17) que H_{k+1} contiene elementos muy grandes. ¿Es este comportamiento razonable? Una cuestión relacionada se refiere a los errores de redondeo que se producen en la aplicación de la precisión finita de estos métodos. ¿Pueden estos errores crecer hasta el punto de borrar toda la información útil para el cuasi-Newton Hessiano aproximado?

Estas cuestiones han sido estudiados analíticamente y experimentalmente, y se sabe ahora que la fórmula BFGS tiene propiedades muy eficaces de auto-corrección. Si la matriz H_k calcula incorrectamente la curvatura en la función objetivo, y si esta mala estimación retrasa la iteración, entonces la aproximación al Hessiano tenderá a corregirse dentro de unos pocos pasos. También se sabe que el método DFP es menos efectivo para corregir malas aproximaciones del Hessiano; esta propiedad se cree que es la razón de su poco rendimiento en la práctica. Las propiedades de autocorrección del método BFGS solo se cumplen cuando se realiza una búsqueda lineal adecuada. En particular, las condiciones de la búsqueda lineal Wolfe asegurar que los gradientes se evalúen en puntos que permiten al modelo (2.1) capturar información apropiada de la curvatura. Es interesante observar que las fórmulas de actualización DFP y BFGS son duales cada una de la otra, en el sentido de que se puede obtener de la otra por los intercambios $s \leftrightarrow y$, $B \leftrightarrow H$. Esta simetría no es sorprendente, dada la manera en que se obtuvieron los métodos anteriormente.

CAPÍTULO 3

MÉTODO BFGS-MODIFICADO

Las propiedades de convergencia del método BFGS han sido bien estudiadas en problemas de minimización convexa (ver [7]). Sin embargo, no se sabe si este método converge globalmente cuando es aplicado a los problemas de minimización no-convexa incluso si se utiliza búsqueda lineal exacta. Además, dado que $G(x)$ (donde $G(x)$ denota la matriz hessiana de f en x_k) generalmente no es definida positiva cuando f es no-convexa, parece que no hay razón para creer que la matriz definida positivo B_k aun permita una buena aproximación de $G(x)$. Por lo tanto, sería razonable esperar que una modificación adecuada del método es eficaz para los problemas no convexos.

Recordando que en el caso donde f es no-convexa la dirección Newton d_k dada por $d_k G_k + g_k = 0$ puede no ser una dirección de descenso de f en x_k dado que G_k no necesariamente es definida positiva. Por tanto la búsqueda lineal (3.1) en la cual se escoge un $\lambda_k > 0$ satisfaciendo

$$f(x_k + \lambda_k d_k) \leq f(x_k) + \sigma \lambda_k g(x_k)^T d_k, \quad (3.1)$$

donde $\sigma \in (0, 1)$ es una constante, $g(x_k)$ el gradiente de f en x_k , puede no estar bien definida. Para superar esta dificultad, el método de Newton se ha modificado de alguna manera o de otra. Por ejemplo, que pueda generar una dirección d_k en la cual G_k es sustituida por la matriz

$$\bar{G}_k \triangleq G_k + r_{k-1} I, \quad (3.2)$$

donde I es la matriz identidad de orden n y r_{k-1} es una constante positiva que se escoge tal que \bar{G}_k es definida positiva. Por esta modificación del método Newton, tenemos los siguientes resultados.

Teorema 3.1. *Dado el conjunto de nivel*

$$\Omega = \{x | f(x) \leq f(x_0)\}$$

sea acotado y f dos veces continuamente diferenciable en un conjunto convexo contenido en Ω . Asumamos que $\{x_k\}$ es generada por el método de Newton modificado usando la matriz \bar{G}_k dada por (3.2) y búsqueda lineal tipo Armijo satisfaciendo (3.1). Supongamos que $\{r_k\}$ es acotada superiormente y que existe un punto de acumulación \bar{x} de $\{x_k\}$ con $G(\bar{x})$ definida positiva. Entonces las siguientes afirmaciones se cumple:

(i) *tenemos*

$$\liminf_{k \rightarrow \infty} \|g_k\| = 0. \quad (3.3)$$

(ii) *Si suponemos mas aun que $\{x_k\} \rightarrow \bar{x}$, $r_k \rightarrow 0$ y $\sigma \in (0, \frac{1}{2})$, entonces la convergencia en rata es al menos superlineal.*

(iii) *Si además, $G(x)$ es Lipschitz alrededor de \bar{x} y existe una constante $C > 0$ tal que $r_{k-1} \leq C\|g_k\|$ para todo k , entonces la convergencia en rata es cuadrática.*

Demostración. (i) Sumando las desigualdades (3.1) obtenemos

$$-\sum_{k=0}^{\infty} \sigma \lambda_k g_k^T d_k \leq f(x_0) - \lim_{k \rightarrow \infty} f(x_k) < \infty$$

donde el limite existe ya que $\{f(x_k)\}$ es decreciente y por el acotamiento de Ω . En particular, $\lambda_k g_k^T d_k \rightarrow 0$. Dado $\{x_k\}_{k \in K} \rightarrow \bar{x}$ alguna subsucesión. Denotemos $\bar{\lambda} = \liminf_{k \rightarrow \infty, k \in K} \lambda_k \geq 0$. Si $\bar{\lambda} > 0$, entonces $g_k^T d_k \rightarrow 0$ o equivalentemente $-g_k \bar{G}_k^{-1} g_k \rightarrow 0$ cuando $k \rightarrow \infty$ con $k \in K$. Dado que $G(\bar{x})$ es definida positiva y $r_k \geq 0$ para cada k , $\{\bar{G}(x_k)\}_{k \in K}$ es uniformemente definida positiva cuando k es suficientemente grande.

Entonces se deduce que $\|g(\bar{x})\| = \liminf_{k \in K, k \rightarrow \infty} \|g_k\| = 0$.

Si $\bar{\lambda} > 0$, supongamos sin pérdida de generalidad que $\liminf_{k \rightarrow \infty, k \in K} \lambda_k = 0$ y

$\bar{G}_k \rightarrow \bar{G}(\bar{x})$ cuando $k \rightarrow \infty$ con $k \in K$, donde este última sobre todo depende de la

acotación de $\{r_k\}$. Por el criterio de la búsqueda lineal, cuando $k \in K$ es suficientemente grande, $\lambda'_k \triangleq \frac{\lambda_k}{\rho}$ no satisface (3.1). En otras palabras, tenemos

$$f(x_k + \lambda'_k d_k) - f(x_k) > \sigma \lambda'_k g(x_k)^T d_k.$$

Dividiendo por λ'_k y tomando límites en ambos lados cuando $k \rightarrow \infty$ obtenemos

$$g(\bar{x})^T \bar{d} \geq \sigma g(\bar{x})^T \bar{d}, \quad (3.4)$$

donde $\bar{d} = -\bar{G}(\bar{x})^{-1}g(\bar{x})$. Como $G(\bar{x})$ es definida positiva y r_k es no-negativa, se tiene que $\bar{G}(\bar{x})$ y $\bar{G}(\bar{x})^{-1}$ son definidos positivo. Dado que $\sigma \in (0, 1)$, (3.4) implica que $g(\bar{x})^T \bar{d} = -g(\bar{x})^T \bar{G}(\bar{x})^{-1}g(\bar{x}) \geq 0$. De esto se deduce que $g(\bar{x}) = 0$, es decir (3.3) se cumple.

(ii) Por suposición, tenemos que $d_k = -\bar{G}(x_k)^{-1}g_k \rightarrow 0$. Como $r_k \rightarrow 0$, tenemos

$$\begin{aligned} f(x_k + d_k) - f(x_k) - \sigma g_k^T d_k &= (1 - \sigma)g_k^T d_k + \frac{1}{2}d_k^T G(x_k)d_k + o(\|d_k\|^2) \\ &= -(1 - \sigma)d_k^T \bar{G}(x_k)d_k + \frac{1}{2}d_k^T G(x_k)d_k + o(\|d_k\|^2) \\ &= -(1 - \sigma - \frac{1}{2})d_k^T \bar{G}(x_k)d_k - \frac{1}{2}d_k^T (G(x_k) \\ &\quad - G(x_k))d_k + o(\|d_k\|^2) \\ &= -(\frac{1}{2} - \sigma)d_k^T \bar{G}(x_k)d_k + o(\|d_k\|^2), \end{aligned}$$

donde la ultima igualdad se sigue de suponer que $r_k \rightarrow 0$. Dado que $\sigma \in (0, \frac{1}{2})$ y $\bar{G}(x_k)$ es definida positiva, se sigue que cuando k es suficientemente grande, el tamaño del paso unitario $\lambda_k \equiv 1$ es aceptado por el criterio de búsqueda lineal. De allí $x_{k+1} = x_k + d_k$ para todo k suficientemente grande. Por otra parte, tenemos

$$\begin{aligned} 0 &= \bar{G}_k d_k + g_k \\ &= \bar{G}_k(x_k + d_k - \bar{x}) + g_k - \bar{G}_k(x_k - \bar{x}) \\ &= \bar{G}_k(x_{k+1} - \bar{x}) + g(x_k) - g(\bar{x}) - G_k(x_k - \bar{x}) - r_{k-1}(x_k - \bar{x}) \\ &= \bar{G}_k(x_{k+1} - \bar{x}) + \left[\int_0^1 G(\bar{x} + t(x_k - \bar{x}))dt - G_k \right] (x_k - \bar{x}) - r_{k-1}(x_k - \bar{x}). \end{aligned}$$

Es decir,

$$x_{k+1} - \bar{x} = -\bar{G}_k^{-1} \left[\int_0^1 G(\bar{x} + t(x_k - \bar{x}))dt - G_k \right] (x_k - \bar{x}) - r_{k-1} \bar{G}_k^{-1}(x_k - \bar{x}).$$

Dado que $G(\bar{x})$ es definida positiva y $r_k \geq 0$, \bar{G}_k es uniformemente definida positiva para un k suficientemente grande.

De allí, se deduce de la igualdad de arriba que existe una constante positiva M tal que

$$\|x_{k+1} - \bar{x}\| \leq M \left\{ \int_0^1 \|G(\bar{x} + t(x_k - \bar{x})) - G_k\| dt + r_{k-1} \right\} \|x_k - \bar{x}\|, \quad (3.5)$$

lo cual prueba (ii).

(iii) De (3.5) tenemos que

$$\begin{aligned} \|x_{k+1} - \bar{x}\| &\leq M \left\{ \int_0^1 \|G(\bar{x} + t(x_k - \bar{x})) - G_k\| dt + r_{k-1} \right\} \|x_k - \bar{x}\| \\ &\leq M \left\{ \int_0^1 \|G(\bar{x} + t(x_k - \bar{x})) - G_k\| dt + C\|g_k\| \right\} \|x_k - \bar{x}\| \\ &\leq M \left\{ \int_0^1 \|G(\bar{x} + t(x_k - \bar{x})) - G_k\| dt + C\|g(x_k) - g(\bar{x})\| \right\} \|x_k - \bar{x}\| \\ &\leq M(L_1 + CL)\|x_k - \bar{x}\|^2, \end{aligned}$$

donde $L > 0$ y $L_1 > 0$ son constante Lipschitz de g y G , respectivamente. ■

El teorema anterior muestra que si elegimos r_k de una manera adecuada (ver[7]), entonces el método Newton modificado sigue conservando la rapidez en la convergencia local como también en la convergencia global. Este hecho lleva a considerar la siguiente pregunta: ¿es posible una modificación del método Cuasi-Newton de tal forma que B_k aproxime a \bar{G}_k , y de este modo asegurar la convergencia global y superlineal sin hipótesis de convexidad en el problema?

Para responder a esta pregunta, primero observe que dada G_{k+1} esta satisface

$$G_{k+1}(x_{k+1} - x_k) \approx g_{k+1} - g_k,$$

la matriz \bar{G}_{k+1} satisface la relación

$$\bar{G}_{k+1}(x_{k+1} - x_k) = (G_{k+1} + r_k I)(x_{k+1} - x_k) \approx (g_{k+1} - g_k) + r_k(x_{k+1} - x_k)$$

o equivalentemente

$$\bar{G}_{k+1}s_k \approx \gamma_k + r_k s_k, \quad (3.6)$$

donde $\gamma_k = g_{k+1} - g_k$. Por tanto, sería razonable exigirle a la matriz B_{k+1} que cumpla (3.6), es decir,

$$B_{k+1}s_k = y_k, \quad (3.7)$$

donde $y_k = \gamma_k + r_k s_k$. Si r_k es pequeño, (3.7) puede considerarse como una aproximación ordinaria de la ecuación secante $B_{k+1}s_k = \gamma_k$. Ahora, se propondrá una modificación del método BFGS basado en las consideraciones anteriores.

Se observa que la convergencia global de un algoritmo con búsqueda lineal no-monótona como (2) requiere a menudo de suposiciones fuertes. Por ejemplo se necesita la condición suficiente de descenso $g_k^T d_k \leq -c_1 \|g_k\|^2$, donde c_1 es una constante positiva. Más aun la fórmula relación $\lim_{k \rightarrow \infty} \|s_k\| = 0$ juega un papel importante en el análisis de convergencia.

Pero la condición suficiente de descenso es difícil de ser satisfecha (ver [18]) por el método BFGS. Con el fin de asegurar $\lim_{k \rightarrow \infty} \|s_k\| = 0$ para el método BFGS, construiremos una nueva búsqueda lineal no monótona tipo Armijo. Ahora planteamos nuestro método que llamamos el método MBFGS como sigue:

Algoritmo 3.1. [*Método MBFGS con búsqueda lineal no monótona*]

- PASO 1: Dado $x_0 \in R^n$, $B_0 = I$, $\delta_1, \rho \in (0, 1)$, $\delta_2 > 0$ y un entero no negativo M . tome $k:=0$. En este artículo por simplicidad, estableceremos $M:=\min(k, M)$.
- PASO 2: calcule d_k por la siguiente ecuación lineal

$$B_k d_k + g_k = 0. \quad (3.8)$$

- PASO 3: calcule el tamaño del paso $\alpha_k = \max\{\rho^0, \rho^1, \dots\}$ satisfaciendo

$$f(x_k + \rho^m d_k) \leq \max_{0 \leq j \leq M} f(x_{k-j}) + \delta_1 \rho^m g_k^T d_k - \delta_2 \|\rho^m d_k\|^2 \quad (3.9)$$

- PASO 4: haga $x_{k+1} = x_k + \alpha_k d_k$
- PASO 5: actualice B_k para obtener B_{k+1} por la siguiente fórmula MBFGS propuesta por Li y Fukushima [7]

$$B_{k+1} = B_k - \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k} + \frac{z_k z_k^T}{z_k^T s_k} \quad (3.10)$$

donde $s_k = x_{k+1} - x_k$, $y_k = g_{k+1} - g_k$,

$$z_k = y_k + C \|g(x_k)\|^r s_k + \text{máx} \left\{ 0, -\frac{y_k^T s_k}{\|s_k\|^2} \right\} s_k, \quad (3.11)$$

donde $C, r > 0$ son constantes dadas.

- PASO 6: haga $k := k+1$ y vuelva al PASO 1

Observación. La formula de actualización MBFGS (3.10) tiene una atractiva propiedad que para cualquier k , siempre se cumple que

$$z_k^T s_k \geq C \|g(x_k)\|^r \|s_k\|^2 > 0, \quad (3.12)$$

la cual asegura que B_{k+1} herede la propiedad de definida positiva de B_k . Esta propiedad es independiente de la convexidad de f así como la búsqueda lineal utilizada. Por la tanto la búsqueda de la dirección definida por (3.8) es siempre una dirección de descenso de la función objetivo, a saber, $g_k^T d_k < 0$. El siguiente resultado muestra que la búsqueda lineal no monótona de Armijo (3.9) esta bien definida.

Proposición 3.1. *El algoritmo (3.1) esta bien definido.*

Demostración. De hecho, sólo tenemos que demostrar que la longitud del paso α_k se puede obtener en un numero finito de pasos. Si no es cierto, entonces para todo m positivo suficientemente grande, tenemos que:

$$\begin{aligned} f(x_k + \rho^m d_k) &\geq \text{máx}_{0 \leq j \leq M} f(x_{k-j}) + \delta_1 \rho^m g_k^T d_k - \delta_2 \rho^2 m \|d_k\|^2 \\ &\geq f(x_k) + \delta_1 \rho^m g_k^T d_k - \delta_2 \rho^2 m \|d_k\|^2 \end{aligned}$$

$$\begin{aligned} f(x_k + \rho^m d_k) - f(x_k) &\geq \delta_1 \rho^m g_k^T d_k - \delta_2 \rho^2 m \|d_k\|^2 \\ &\geq \rho^m (\delta_1 g_k^T d_k - \delta_2 \rho^m \|d_k\|^2) \end{aligned}$$

$$\frac{f(x_k + \rho^m d_k) - f(x_k)}{\rho^m} \geq \delta_1 g_k^T d_k - \delta_2 \rho^m \|d_k\|^2 \quad (3.13)$$

ahora haciendo $m \rightarrow \infty$ en (3.13) y por la definición de derivada direccional se tiene que

$g_k^T d_k \geq \delta_1 g_k^T d_k$. Por otro lado como $g_k^T d_k < 0$ ya que d_k es dirección de descenso y $\delta_1 \in (0, 1)$ se tiene que

$$\begin{aligned} g_k^T d_k < 0 &\Rightarrow \delta_1 < 0 \\ &\Rightarrow \delta_1 g_k^T d_k > g_k^T d_k > \delta_1 g_k^T d_k \text{ lo cual es una contradicción} \end{aligned}$$

así el Algoritmo (3.1) está bien definido. ■

En esta sección, probaremos la convergencia global del algoritmo 3.1 bajo las siguientes suposiciones.

Suposición (A).

1. el conjunto de nivel $\Omega = \{x \in \mathbb{R}^n | f(x) \leq f(x_0)\}$ es acotado.
2. En alguna vecindad N de Ω , f es continuamente diferenciable y su gradiente es Lipschitz continuo, esto es existe una constante $L > 0$ tal que

$$\|g(x) - g(y)\| \leq L\|x - y\|, \forall x, y \in N \quad (3.14)$$

Es claro que la secuencia $\{x_k\}$ generada por el Algoritmo (3.1) está contenida en Ω . Además, obtenemos de la suposición A que existe una constante $\gamma_1 > 0$, tal que

$$\|g(x)\| \leq \gamma_1, \forall x \in \Omega. \quad (3.15)$$

Desde ahora siempre supondremos que las condiciones en la suposición A se cumplen. Sin especificación, tomaremos $\{x_k\}$ y $\{d_k\}$ como la secuencia iterativa y la secuencia de dirección generada por el algoritmo (3.1), respectivamente.

El siguiente lema viene del teorema 2.1 (ver [1]) y es muy usado para probar la convergencia global de el método *BFGS*.

Lema 3.1. *sea \mathbf{B}_k la actualización por la formula MBFGS (3.10). Supongamos que \mathbf{B}_0 es simétrica y definida positiva. Si existen constantes positivas $m \leq M_1$ tal que para todo $k \geq 0$, z_k y s_k satisface*

$$\frac{z_k^T s_k}{\|s_k\|^2} \geq m, \frac{\|z_k\|^2}{z_k^T s_k} \leq M_1, \quad (3.16)$$

entonces para cualquier $\kappa \in (0, 1)$, existen constantes $\beta_1, \beta_2, \beta_3 > 0$ tal que para cualquier $k \geq 1$, las desigualdades

$$\begin{aligned} \|\mathbf{B}_j s_j\| &\leq \beta_1 \|s_j\|, \\ \beta_2 \|s_j\|^2 &\leq s_j^T \mathbf{B}_j s_j \leq \beta_3 \|s_j\|^2 \end{aligned} \quad (3.17)$$

se mantiene para el menor $\lceil \kappa k \rceil$ valores de $j \in [1, k]$.

Para cualquier k , definimos los conjuntos índices K_k y K como sigue:

$$K_k = \{i \leq k \mid (3.17) \text{ se mantiene}\}, K = \bigcup_{k=0}^{\infty} K_k. \quad (3.18)$$

Dado que $s_k = \alpha_k d_k$, si reemplazamos s_k por d_k , entonces la desigualdades (3.17) se cumplen también. Además tenemos de (3.8) y las desigualdades (3.17) que

$$\|g_k\| = \|\mathbf{B}_k d_k\| \leq \beta_1 \|d_k\|, \text{ y}$$

$\beta_2 \|d_k\|^2 \leq \|d_k^T \mathbf{B}_k d_k\| = \|d_k^T g_k\| \leq \|d_k\| \|g_k\|$. Entonces se sigue de las desigualdades de arriba que

$$\beta_2 \|d_k\| \leq \|g_k\| \leq \beta_1 \|d_k\|, \forall k \in K \quad (3.19)$$

donde β_1, β_2 son los mismos que de (3.17).

El siguiente lema muestra que $\lim_{k \rightarrow \infty} \|s_k\| = 0$.

Lema 3.2. Dado que la **suposición A** se cumple, si $\alpha_k > 0$ es calculada por la búsqueda lineal no monótona (3.9), entonces tenemos

$$\lim_{k \rightarrow \infty} \alpha_k d_k = 0, \lim_{k \rightarrow \infty} \alpha_k g_k^T d_k = 0. \quad (3.20)$$

Demostración. Sea $h(k)$ un entero que satisface

$$k - M \leq h(k) \leq k, f(x_{h(k)}) = \max_{0 \leq j \leq M} f(x_{k-j}) \quad (3.21)$$

ahora se sigue de (3.9) que la secuencia $\{f(x_{h(k)})\}$ es decreciente. En efecto, note que $g_k^T d_k < 0$, y tenemos de (3.9) que

$$f(x_{k+1}) \leq \max_{0 \leq j \leq M} f(x_{k-j}) = f(x_{h(k)})$$

Entonces obtenemos de las desigualdades de arriba que

$$\begin{aligned}
f(x_{h(k+1)}) &= \max_{0 \leq j \leq M} f(x_{k+1-j}) \\
&= \max \left(\max_{0 \leq j \leq M-1} f(x_{k-j}), f(x_{k+1}) \right) \\
&\leq \max \left(\max_{0 \leq j \leq M-1} f(x_{k-j}), f(x_{k-M}), f(x_{k+1}) \right) \\
&= \max \left(\max_{0 \leq j \leq M} f(x_{k-j}), f(x_{k+1}) \right) \\
&= \max (f(x_{h(k)}), f(x_{k+1})) \\
&= f(x_{h(k)})
\end{aligned}$$

Así hemos probado que la secuencia $\{f(x_{h(k)})\}$ es decreciente. Por otra parte tenemos de la ultima desigualdad, (3.9) y (3.21) que

$$\begin{aligned}
f(x_{h(k)}) &= f(x_{h(k)-1} + \alpha_{h(k)-1} d_{h(k)-1}) \\
&\leq \max_{0 \leq j \leq M} f(x_{h(k)-1-j}) + \delta_1 \alpha_{h(k)-1} g_{h(k)-1}^T d_{h(k)-1} - \delta_2 \|\alpha_{h(k)-1} d_{h(k)-1}\|^2 \\
&= f(x_{h(h(k)-1)}) + \delta_1 \alpha_{h(k)-1} g_{h(k)-1}^T d_{h(k)-1} - \delta_2 \|\alpha_{h(k)-1} d_{h(k)-1}\|^2
\end{aligned}$$

Dado que $\{f(x_{h(k)})\}$ es decreciente y acotada en virtud de la suposición A, y al tomar $k \rightarrow \infty$ en la desigualdad de arriba, tenemos que

$$\lim_{k \rightarrow \infty} \alpha_{h(k)-1} d_{h(k)-1} = 0 \quad (3.22)$$

denotemos $\widehat{h}(k) = h(k + M + 2)$. Ahora por inducción, probaremos que para cualquier $j \geq 1$, se cumple:

$$\lim_{k \rightarrow \infty} \alpha_{\widehat{h}(k)-j} d_{\widehat{h}(k)-j} = 0. \quad (3.23)$$

$$\lim_{k \rightarrow \infty} f(x_{\widehat{h}(k)-j}) = \lim_{k \rightarrow \infty} f(x_{h(k)}). \quad (3.24)$$

veamos para $j = 1$, dado que $\{\widehat{h}(k)\} \subset \{h(k)\}$ luego se sigue de (3.22) que (3.23) se cumple, por lo cual $\|x_{\widehat{h}(k)} - x_{\widehat{h}(k)-1}\| \rightarrow 0$

por otro lado como $f(x)$ es uniformemente continua en el conjunto de nivel Ω , (3.24)

se cumple para $j = 1$.

Ahora supongamos que (3.23) y (3.24) se cumple para cualquier j . luego de (3.9) se sigue que

$$f(x_{\widehat{h}(k)-j}) \leq f(x_{h(\widehat{h}(k)-j-1)}) + \delta_1 \alpha_{\widehat{h}(k)-j-1} g_{\widehat{h}(k)-j-1}^T d_{\widehat{h}(k)-j-1} - \delta_2 \|\alpha_{\widehat{h}(k)-j-1} d_{\widehat{h}(k)-j-1}\|^2$$

y cuando $\rightarrow \infty$, obtenemos de (3.24) que

$$\lim_{k \rightarrow \infty} \alpha_{\widehat{h}(k)-(j+1)} d_{\widehat{h}(k)-(j+1)} = 0,$$

lo cual implica que $\|x_{\widehat{h}(k)-j} - x_{\widehat{h}(k)-(j+1)}\| \rightarrow 0$. Y dado que $f(x)$ es uniformemente continua en el conjunto de nivel Ω , se sigue que

$$\lim_{k \rightarrow \infty} f(x_{\widehat{h}(k)-(j+1)}) = \lim_{k \rightarrow \infty} f(x_{\widehat{h}(k)-j}) = \lim_{k \rightarrow \infty} f(x_{h(k)})$$

luego (3.23) y (3.24) se cumplen para cualquier $j \geq 1$.

Ahora para cualquier k , se cumple que (ver [4])

$$x_{k+1} = x_{\widehat{h}(k)} - \sum_{j=1}^{\widehat{h}(k)-k-1} \alpha_{\widehat{h}(k)-j} d_{\widehat{h}(k)-j} \quad (3.25)$$

y como $\widehat{h}(k) - k - 1 = h(k + M + 2) - k - 1 \leq k + M + 2 - k - 1 = M + 1$, tenemos de (3.23) y (3.25) que

$$\lim_{k \rightarrow \infty} \|x_{k+1} - x_{\widehat{h}(k)}\| = 0$$

y de la continuidad uniforme de $f(x)$ obtenemos que

$$\lim_{k \rightarrow \infty} f(x_k) = \lim_{k \rightarrow \infty} f(x_{\widehat{h}(k)})$$

Se sigue de (3.9) que $f(x_{k+1}) \leq f(x_{h(k)}) + \delta_1 \alpha_k g_k^T d_k - \delta_2 \|\alpha_k d_k\|^2$

luego tomando $k \rightarrow \infty$, tenemos que

$$\lim_{k \rightarrow \infty} \alpha_k d_k = 0, \lim_{k \rightarrow \infty} \alpha_k g_k^T d_k = 0$$

■

Teorema 3.2. *se tiene que*

$$\liminf_{k \rightarrow \infty} \|g_k\| = 0$$

Demostración. supongamos que la conclusión no es cierta. Luego por definición de limite inferior, existe una constante $\varepsilon > 0$ tal que para cualquier $k \geq 0$ se cumple que

$$\|g_k\| \geq \varepsilon \quad (3.26)$$

así de la desigualdad de arriba, (3.11), (3.12), (3.14) y (3.15), se tiene que existe una constante $C_2 > 0$ tal que

$$z_k^T s_k \geq C \|g_k\|^r s_k^T s_k \geq C \varepsilon^r s_k^T s_k = C_2 s_k^T s_k, \quad (3.27)$$

donde $C_2 = C \varepsilon^r$

$$\|z_k\| \leq \|y_k\| + C \|g_k\|^r \|s_k\| + \frac{\|y_k\| \|s_k\|^2}{\|s_k\|^2} \leq (2L + C \gamma_1^r) \|s_k\| \quad (3.28)$$

luego de (3.27) y (3.28), tenemos

$$z_k^T s_k \leq (2L + C \gamma_1^r)^2 \|s_k\|^2 \leq \frac{(2L + C \gamma_1^r)^2}{C_2} z_k^T s_k = C_3 z_k^T s_k, \text{ donde } C_3 = \frac{(2L + C \gamma_1^r)^2}{C_2}.$$

Por tanto las condiciones (3.17) y (3.19) en el lema (3.1) son satisfechas, y así las conclusiones del lema (3.1) se cumplen.

Ahora si $\liminf_{k \in K, k \rightarrow \infty} \alpha_k > 0$, se sigue de (3.19) y (3.20) que

$$\liminf_{k \rightarrow \infty} \|g_k\| = 0$$

lo cual es una contradicción.

Si $\liminf_{k \in K, k \rightarrow \infty} \alpha_k = 0$, entonces para un $k \in K$ suficientemente grande, se cumple que

$$f\left(x_k + \frac{\alpha_k d_k}{\rho}\right) \geq \max_{0 \leq j \leq M} f(x_{k-j}) + \delta_1 \frac{\alpha_k}{\rho} g_k^T d_k - \delta_2 \left(\frac{\alpha_k}{\rho}\right)^2 \|d_k\|^2$$

lo cual implica que

$$f\left(x_k + \frac{\alpha_k d_k}{\rho}\right) \geq f(x_k) + \delta_1 \frac{\alpha_k}{\rho} g_k^T d_k - \delta_2 \left(\frac{\alpha_k}{\rho}\right)^2 \|d_k\|^2$$

por otro lado por el teorema del valor medio, existe $t_k \in (0, 1)$ tal que

$$(g(u_k))^T d_k \geq \delta_1 g_k^T d_k - \delta_2 \left(\frac{\alpha_k}{\rho} \right)^2 \|d_k\|^2 \quad (3.29)$$

donde $u_k \in [x_k, x_k + \frac{\alpha_k d_k}{\rho}]$. Sea $\{x_k\}_K$ una subsecuencia la cual converge a x^* tal que

$$\liminf_{k \in K, k \rightarrow \infty} x_k = x^*, \quad \liminf_{k \in K, k \rightarrow \infty} \frac{d_k}{\|d_k\|} = d^*$$

Entonces también se cumplen que

$$\liminf_{k \in K, k \rightarrow \infty} u_k = x^*$$

Dado $k \in K$ y $k \rightarrow \infty$ aplicando en (3.29), tenemos

$$g^T(x^*)d^* \geq \delta_1 g^T(x^*)d^*.$$

Y como $1 - \delta_1 > 0$, se tiene que

$$g^T(x^*)d^* \geq 0 \quad (3.30)$$

pero (3.8), (3.17), (3.19) y (3.26) implica

$$g_k^T d_k = -d_k^T B_k d_k \leq -\beta_2 \|d_k\|^2 \leq -\frac{\beta_2}{\beta_1} \|g_k\| \|d_k\| \leq -\frac{\beta_2}{\beta_1} \varepsilon \|d_k\| \quad (3.31)$$

Dado $k \in K$ y $k \rightarrow \infty$ a (3.31) tenemos que

$$(g(x^*))^T d^* < 0,$$

lo cual es una contradicción con (3.30). ■

CAPÍTULO 4

PRUEBAS NUMÉRICAS

Ahora reportaremos algunas pruebas numéricas. Probaremos el algoritmo 3.1 con diferentes valores de M y el método BFGS estándar para algunos problemas tomados de [12].

Todos los códigos fueron escritos en código Matlab. Detendremos la iteración si el número total de iteraciones excede 4×10^3 o $\|g(x_k)\| \leq 10^{-5}$. También escogeremos los parámetros del algoritmo 3.1 como sigue: $\delta_1 = \delta_2 = 0,1$, $C = 10^{-6}$, $r = 2$, $\rho = 0,4$.

Los problemas de prueba incluyen algunas funciones no-convexas tales como las funciones Band y Jensam. A continuación mostramos todas las funciones a las cuales se le realizaron los experimentos numéricos:

- función Band ($n=10$):

$$f_1(x) = \sum_{j=1}^n \left[x_i(2 + 15x_i^2) - \sum_{j \in J_i} x_j(1 + x_j) \right]^2, \quad x_0 = [0, \dots, 0]$$

donde

$$J_i = \{j : j \neq i, \max\{1, i - m_l\} \leq j \leq \min\{n, i + m_u\}\}$$
$$y \quad m_l = 5, m_u = 1.$$

Si se escogen dos puntos $y = (1, \dots, 1)^T$ y $x = (0, \dots, 0)^T$, entonces tenemos $f(y) - f(x) - \nabla f(x)^T(y - x) = -10 < 0$. Esto demuestra que la función Band no es convexa.

- función Jensam (n=2)

$$f_2(x) = (4 - e^{x_1} - e^{x_2})^2 + (6 - e^{2x_1} - e^{2x_2})^2, \quad x_0 = [0, \dots, 0].$$

Al obtener la matriz Hessiana de $f_2(x)$ en $x = (0, 0)$

$$\nabla^2 f(0, 0) = \begin{pmatrix} -18 & 10 \\ 10 & -18 \end{pmatrix}, \text{ el cual no es una matriz semi-definida positiva.}$$

Así la función Jensam es no convexa.

- función Rosenbrock (n=4)

$$f_3(x) = \sum_{i=1}^{n-1} 100(x_{i+1} - x_i^2)^2 + (1 - x_i)^2, \quad x_0 = [-1, 2, 1, \dots, -1, 2, 1].$$

- función extendida de white y Holst (n=8)

$$f_4(x) = \sum_{i=1}^{n/2} 100(x_{2i} - x_{2i-1}^3)^2 + (1 - x_{2i-1}^2)^2, \quad x_0 = [-1, 2, 1, \dots, -1, 2, 1].$$

- función de Raydan 1 (n=4)

$$f_5(x) = \sum_{i=1}^n \frac{i}{10} (\exp(x_i) - x_i), \quad x_0 = [1, 1, \dots, 1, 1]$$

- función Hager (n=6)

$$f_6(x) = \sum_{i=1}^n (\exp(x_i) - \sqrt{i}x_i), \quad x_0 = [1, 1, \dots, 1, 1]$$

- función Cuadrática QF1 (n=5)

$$f_7(x) = \frac{1}{2} \sum_{i=1}^n ix_i^2 - x_n, \quad x_0 = [1, 1, \dots, 1, 1]$$

- función FLETCHCR(CUTE) (n=8)

$$f_8(x) = \sum_{i=1}^{n-1} 100(x_{i+1} - x_i + 1 - x_i^2)^2, \quad x_0 = [0, 0, \dots, 0, 0]$$

- función DQDRTIC(CUTE) (n=10)

$$f_9(x) = \sum_{i=1}^{n-2} (x_i^2 + 100x_{i+1}^2 + 100x_{i+2}^2), \quad x_0 = [3, 3, \dots, 3, 3]$$

- función POWER(CUTE) (n=12)

$$f_{10}(x) = \sum_{i=1}^n (ix_i)^2, \quad x_0 = [1, 1, \dots, 1, 1].$$

- función Boot (n=2)

$$f_{11}(x) = (x_1 + 2x_2 - 7)^2 + (2x_1 + x_2 - 5)^2, \quad x_0 = [0, 0]$$

- función Beale (n=2)

$$\begin{aligned} f_{12}(x) &= (1,5 - x_1 + x_1x_2)^2 + (2,25 - x_1 + x_1x_2^2)^2 + (2,625 - x_1 + x_1x_2^3)^2 \\ x_0 &= [0, 0] \end{aligned}$$

- función Griewank (n=50)

$$f_{13}(x) = \sum_{i=1}^n \frac{x_i^2}{4000} - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right), \quad x_0 = [-1, 2, \dots, -1, 2]$$

- función Trid (n=15)

$$f_{14}(x) = \sum_{i=1}^n (x_i - 1)^2 - \sum_{i=2}^n x_i x_{i-1}, \quad x_0 = [0, 5, \dots, 0, 5]$$

- función Rastrigin (n=7)

$$f_{15}(x) = 10n + \sum_{i=1}^n (x_i^2 - 10\cos(2\pi x_i)), \quad x_0 = [2, \dots, 2]$$

La siguiente tabla donde n : es la dimensión de los problemas de las pruebas numéricas, $iter$: el número de iteraciones, $f(x^*)$: el valor de la función en el punto óptimo, tiempo: cuanto tarda en ejecutar los algoritmos medidos en segundos, muestra que el método MBFGS funciona de manera similar como el método BFGS estándar cuando $M=0$, sin embargo en las pruebas numéricas que realizaron W.Zhou, L.Zhang estos no lograron resolver los problemas de la función Band y la función bv con $n=50$. El método MBFGS con $M=3$ o $M=5$ resuelve todos los problemas dados con éxito, incluso para funciones no convexas como función band. Esto muestra que los métodos no monótonos son más estables que los métodos monótonos correspondientes. También podemos ver que el método MBFGS con $M=5$ funciona mejor, ya que puede resolver el 55% (8 de 15) de todos los problemas de prueba con el menor tiempo de ejecución y evaluaciones de las funciones. Sin embargo, para algunos problemas tales como función Griewank, su rendimiento es peor que los de otros métodos. Esto también muestra que la forma de elegir un M adecuado es muy importante, pero este es un problema relativamente difícil.

<i>funciones</i>	<i>n</i>	<i>BFGS</i> (<i>M</i> = 0) <i>iter</i> / <i>f</i> (<i>x</i> [*])/ <i>tiempo</i>	<i>MBFGS</i> (<i>M</i> = 0) <i>iter</i> / <i>f</i> (<i>x</i> [*])/ <i>tiempo</i>	<i>MBFGS</i> (<i>M</i> = 3) <i>iter</i> / <i>f</i> (<i>x</i> [*])/ <i>tiempo</i>	<i>MBFGS</i> (<i>M</i> = 5) <i>iter</i> / <i>f</i> (<i>x</i> [*])/ <i>tiempo</i>
<i>f</i> ₁	10	10/4.4513e-014/ 0.9844	10/4.4467e-014/ 0.7884	25/8.1466e-015/ 0.0581	36/ 7.8242e-014/ 0.0524
<i>f</i> ₂	2	12/ 0.2653/1.7915	12/0.2653/ 0.1501	30/0.2653/0.0825	48/0.2653/0.0373
<i>f</i> ₃	4	42/7.2218e-015/ 0.3532	42/2.4702e-014/0.2635	64/ 8.1917e-014/0.0855	100/ 7.4640e-015/ 0.0901
<i>f</i> ₄	8	87/5.7625e-013/0.1406	NaN	120/5.4118e-013/2.4210	149/4.7458e-013/ 3.1544
<i>f</i> ₅	4	14/1.0000/ 2.7363	16/ 1.0000/ 1.0782	20/1.0000/0.0483	35/1.0000/0.0330
<i>f</i> ₆	6	8/3.3184/ 0.0468	8/3.3184/0.0144	31/ 3.3184/0.0818	51/3.3184/0.0755
<i>f</i> ₇	5	11/-2.5000/0.2208	10/-2.5000/7.1735	31/-2.5000/3.5884	50/-2.5000/2.4498
<i>f</i> ₈	8	28/5.7808e-015/1.2890	29/4.9983e-014/0.3597	47/ 8.1534e-015/0.0968	77/1.2877e-014/0.1549
<i>f</i> ₉	10	18/1.0647e-014/0.0517	18/1.0530e-014/0.0484	54/3.3132e-014/0.1079	80/2.2831e-014/0.1386
<i>f</i> ₁₀	12	22/9.4186e-014/0.5844	22/9.4238e-014/0.4057	44/1.6134e-013/0.1143	68/6.3419e-014/ 0.1168
<i>f</i> ₁₁	2	10/3.5268e-012/ 0.0130	10/3.5268e-012/0.0099	28/3.8878e-012/0.0225	46/3.8868e-012/0.0341
<i>f</i> ₁₂	2	15/1.1951e-012/0.1674	14/1.4121e-011/0.0126	26/1.2782e-012/ 0.0221	63/ 8.3297e-013/0.0531
<i>f</i> ₁₃	50	169/6.1883e-010/4.0261	1000/ 9.2826e-007/18.2719	505/1.0873e-009/8.8416	445/3.4036e-010/7.7948
<i>f</i> ₁₄	15	16/-665.0000/0.0565	38/-665.0000/0.1062	41/665.0000/0.0800	62/-665.0000/0.1544
<i>f</i> ₁₅	7	10/27.8588/15.3456	10/27.8588/0.4219	42/27.8588/ 0.1935	58/27.8588/0.0714

CONCLUSIONES

En las búsquedas lineales monótonas Se busca un valor de α aproximado que reduzca significativamente a $f(x_k + \alpha d_k)$ este valor aproximado debe ser muy fácil y económico de calcular. En la práctica, basta con encontrar un α que satisfaga criterios como el de Armijo, Wolfe, Goldstein fáciles de comprobar que tratan de evitar valores muy pequeños o muy grandes de α .

La técnicas de búsqueda lineal no-monótona a pesar de no tener la propiedad de descenso en cada iteración de la función f , ofrecen gran estabilidad en su implementación. Los métodos Cuasi-Newton ofrecen muchas ventajas, ya que su convergencia es tanto superlineal, como global cuando se utiliza búsqueda lineal exacta, termina en $n + 1$ iteraciones, $O(n^2)$ operaciones por paso y genera direcciones conjugadas.

Se puede observar que el método MBFGS con búsqueda lineal no-monótona tipo Armijo ofrece mayor estabilidad en comparación con el método BFGS estándar con búsqueda lineal monótona, las cuales se pueden ver en la parte de pruebas numéricas donde muestra mayor eficacia a la hora de resolver los problemas de funciones convexas y no-convexas.

Para estudiar el método MBFGS, se realizaron un conjunto de algoritmos desarrollados en MATLAB versión R2011a.

4.1. funciones.m

presentamos las distintas funciones que se utilizaron para realizar pruebas al algoritmo MBFGS.

función Rosenbrock #1

```
n=length(x);
y=0;
for i=1:n-1
    y= y +100*(x(i+1)-x(i).^2).^2 + (1-x(i)).^2;
end
```

función #2 funcion jensam

```
y=(4-exp(x(1))-exp(x(2))).^2+(6-exp(2*x(1))-exp(2*x(2))).^2;
```

función #3 Band **function** (n=10)

```
n=length(x);
s=5*(x(1)*(1+x(1))+ x(6)*(1+x(6)));
t=6*(x(2)*(1+x(2))+ x(3)*(1+x(3))+x(4)*(1+x(4))+x(5)*(1+x(5)));
u=4*x(7)*(1+x(7));
v=3*x(8)*(1+x(8));
```

```

w=2*x(9)*(1+x(9));
z=x(10)*(1+x(10));
for i=1:n
    y= (x(i)*(2+15*(x(i)).^2) + 1-(s+t+u+v+w+z)).^2;
end
función #4 extendida de white y Holst (n=8)
n=length(x);
    y=0;
    for i=1:n/2
        y= y +100*(x(2*i)-x(2*i-1).^3).^2 + (1-x(2*i-1).^2).^2;
    end
función #5 de Raydan 1 (n=4)
n=length(x);
    y=0;
    for i=1:n
        y= y+ (i/10)*(exp(x(i))-x(i));
    end
función #6 Hager (n=6)
n=length(x);
    y=0;
    for i=1:n
        y= y+(exp(x(i))-sqrt(i)*x(i));
    end
función #7 Cuadratica QF1 (n=5)
n=length(x);
    y=0;
    for i=1:n
        y= y+(0.5)*(i*x(i).^2)-x(n);
    end
función # 8 FLETCHCR(CUTE) (n=8)
n=length(x);
    y=0;

```

```

    for i=1:n-1
        y= y+100*(x(i+1)-x(i)+1-x(i).^2).^2;
    end
función #9 DQDRTIC(CUTE) (n=10)
n=length(x);
    y=0;
    for i=1:n-2
        y= y+(x(i).^2+100*(x(i+1)).^2+100*(x(i+2).^2));
    end
función # 10 funcin POWER(CUTE) (n=12)
n=length(x);
    y=0;
    for i=1:n
        y= y+(i*x(i)).^2;
    end
función # 11 booth
y = (x(1)+2*x(2)-7)^2+(2*x(1)+x(2)-5)^2;
función # 12 beale
y = (1.5-x(1)*(1-x(2)))^2+(2.25-x(1)*(1-x(2)^2))^2+
(2.625-x(1)*(1-x(2)^3))^2;
función # 13 funcion Griewank
n =length(x);
fr = 4000;
s = 0;
p = 1;
for j = 1:n; s = s+x(j)^2; end
for j = 1:n; p = p*cos(x(j)/sqrt(j)); end
y = s/fr-p+1;
función #14 funcion Trid
n =length(x);
s1 = 0;
s2 = 0;

```

```

for j = 1:n;
    s1 = s1+(x(j)-1)^2;
end
for j = 2:n;
    s2 = s2+x(j)*x(j-1);
end
y = s1-s2;
función #15 Rastrigin function
n=length(x);
s = 0;
for j = 1:n
    s = s+(x(j)^2-10*cos(2*pi*x(j)));
end
y = 10*n+s;

```

4.2. gradiente

Este algoritmo permite calcular el valor numérico del gradiente de las funciones.

```

function Dy = grad(x)
    n=length(x);
    E=eye(n);
    h=10^(-2);
    hm=h/2;
    hh=2*h;
    for i=1:n
        xM=f(x+h*E(:,i));
        xm=f(x-h*E(:,i));
        xM2=f(x+hm*E(:,i));
        xm2=f(x-hm*E(:,i));
        Dy(i,1)=(4*(xM2-xm2)/h - (xM-xm)/hh)/3;
    end

```

4.3. Armijo

Permite calcular el tamaño de paso mediante el criterio de Armijo estándar.

```
function alpha=armijo2(x0,d)
ro=0.4;
a=2;
delta=0.1;
%delta2=0.1;
fx=f(x0);
%mf=max(ff);
g=grad(x0);
sw=0;
j=0;
while sw==0
    alpha=a*ro.^j;
    fj= f(x0+alpha*d);
    Fj= fx+delta*alpha*g'*d;
    if Fj>=fj
        sw=1;
        %alpha=alpha(j);
    else
        alpha=ro*alpha;
        j=j+1;
    end
end
```

Permite calcular el tamaño del paso mediante la modificación del criterio de Armijo.

```
function alpha=armijo(x0,d)
ro=0.4;
a=2;
delta=0.1;
delta2=0.1;
fx=f(x0);
%mf=max(ff);
g=grad(x0);
sw=0;
j=0;
while sw==0
    alpha=a*ro.^j;
    fj= f(x0+alpha*d);
    Fj= fx+delta*alpha*g'*d -delta2*(norm(alpha*d))^2;
    if Fj>=fj
        sw=1;
        %alpha=alpha(j);
    else
        alpha=ro*alpha;
        j=j+1;
    end
end
```

Permite calcular el tamaño de paso, mediante una búsqueda lineal no-monótona tipo Armijo propuesta L.Grippio

```
function alpha=grippolampm(x,d,ff)
rho=0.4;
a=2;
delta1=0.1;
delta2=0.1;
mf=max(ff);
g=grad(x);
sw=0;
j=0;
while sw==0
    alpha=a*rho.^j;
    fj= f(x+alpha*d);
    Fj= mf+delta1*alpha*g'*d-delta2*(norm(alpha*d))^2;
    if Fj>=fj
        sw=1;
        %alpha=alpha(j);
    else
        alpha=rho*alpha;
        j=j+1;
    end
end
```


4.4. Algoritmos

Algoritmo BFGS estándar

```

function [x, iter, tiempo, H0, g0, fun]=BFGS(x0)
tic
N=length(x0);
B0=eye([N N]);
H0=eye([N N]);
g0=grad(x0);
iter=0;
d=-B0*g0;
N0=norm(g0);
while N0>=(10)^-5 && iter <1000
    alpha=armijo2(x0,d);
    x=x0+alpha*d;
    g1=grad(x);
    s=x-x0;
    y=g1-g0;
    p=(1)/(y'*s);
    H0=(B0-p*s*y')*H0*(B0-p*y*s')+p*(s)*s';
    iter= iter +1;
    x0=x;
    g0=g1;
    fun=f(x0);
    d=-H0*g0;
    N0=norm(g0);

end
tiempo=toc;

```

Algoritmo MBFGS para el caso $M=0$

```

function [x, iter, tiempo, B0, g0, fun]=MBFGSA(x0)
tic
N=length(x0);
B0=eye([N N]);
g0=grad(x0);
C=10.^-6;
r=2;
iter=0;
d=-B0\g0;
N0=norm(g0);
while N0>=(10)^-5 && iter <1000
    alpha=armijo(x0,d);
    x=x0+alpha*d;
    g1=grad(x);
    s=x-x0;
    y=g1-g0;
    norma=((-y'*s)/(norm(s)^2));
    mf=max([norma, 0]);
    z= y+C*((norm(g1))^r)*s + mf*s;
    B0= B0+ -((B0*s)*(s'*B0))/(s'*B0*s)+(z*z')/(z'*s);
    iter= iter+1;
    x0=x;
    g0=g1;
    d=-B0\g0;
    N0=norm(g0);
    fun=f(x0);
end
tiempo=toc;

```

Algoritmo MBFGS para el caso $M > 0$

```

function [x, iter , tiempo , B0 , g0 , fun]=MBFGS(x0 ,M)
tic
N=length(x0);
B0=eye([N N]);
g0=grad(x0);
C=10.^-6;
r=2;
ff=f(x0)*ones(M,1);
iter=0;
d=-B0\g0;
N0=norm(g0);
while N0>=(10)^-5 && iter <1000
    alpha=grippolampm(x0 ,d , ff );
    x=x0+alpha*d;
    g1=grad(x);
    s=x-x0;
    y=g1-g0;
    norma=((-y'*s)/(norm(s)^2));
    mf=max([norma , 0]);
    z= y+C*((norm(g1))^r)*s + mf*s;
    B0= B0+ -((B0*s)*(s'*B0))/(s'*B0*s)+(z*z')/(z'*s);
    iter= iter+1;
    x0=x;
    g0=g1;
    d=-B0\g0;
    N0=norm(g0);
    for k=M-1:-1:1
        ff(k+1)=ff(k);
    end
    ff(1)=f(x0);
    fun=f(x0);

```

end

tiempo=`toc`;

donde x_0 es un punto cualquiera de \mathbb{R}^n , d es una dirección de descenso, M un entero no negativo.

REFERENCIAS

- [1] R. H. Byrd and J. Nocedal. A tool for the analysis of quasi-newton methods with application to unconstrained optimization. *SIAM Numer. Anal.*, 26:727–739, 1989.
- [2] Y. H. Dai. A nonmonotone conjugate gradient algorithm for unconstrained optimization. *J. Syst. Sci. Complex.*, 115:139–145, 2002.
- [3] Y. H. Dai. On the nonmonotone line search. *J. Optim. Theory Appl.*, 112:315–330, 2002.
- [4] L. Grippo, F. Lampariello, and S. Lucidi. A nonmonotone line search technique for Newton’s method. *SIAM J. Numer. Anal.*, 23:707–716, 1986.
- [5] L. Grippo, F. Lampariello, and S. Lucidi. A truncated Newton method with nonmonotone line search for unconstrained optimization. *J. Optim. Theory Appl.*, 60:401–419, 1989.
- [6] J. Y. Han and G.vH. Liu. Global convergence analysis of a new nonmonotone BFGS algorithm on convex objective functions. *Comput. Optim. Appl.*, 7:277–289, 1997.
- [7] D. H. Li and M. Fukushima. A modified BFGS method and its global convergence in nonconvex minimization. *J. Comput. Appl. Math.*, 129:15–35, 2001.
- [8] G. H. Liu, D. Han, and D. F. Sun. Global convergence of the BFGS algorithm with nonmonotone line search. *Optimization*, 34:147–159, 1995.

-
- [9] G. H. Liu and L. L. Jing. Convergence of the nonmonotone Perry and Shanno method for optimization. *Comp. Optim. Appl.*, 16:159–172, 2000.
- [10] G. H. Liu, L. L. Jing, L. X. Han, and D. Han. A class of nonmonotone conjugate gradient methods for unconstrained optimization. *J. Optim. Theory Appl.*, 101:127–140, 1999.
- [11] W. F. Mascarenhas. The BFGS method with exact line searches fails for non-convex objective functions. *Math. Program.*, 99:69–94, 2004.
- [12] A. Neculai. An unconstrained optimization test functions collection. *Adv. Model. Optim.*, 10(1):147–161, 2008.
- [13] M. Raydan. The Barzilai and Borwein gradient method for the large scale unconstrained minimization problems. *SIAM J. Optim.*, 7:26–33, 1997.
- [14] P. L. Toint. An assessment of nonmonotone line search techniques for unconstrained optimization. *SIAM J. Sci. Comput.*, 17:725–739, 1996.
- [15] P. L. Toint. Non-monotone trust-region algorithms for nonlinear optimization subject to convex constraints. *Math. Program.*, 77:69–94, 1997.
- [16] S. Wright and J. Nocedal. *Numerical optimization*. Springer series in operations research and financial engineering. Springer, 2nd edition, 2006.
- [17] S. Zhenjun and W. Shengquan. Modified nonmonotone Armijo line search for descent method. *Numer. Algor.*, 57:1–25, 2011.
- [18] W. Zhou and L. Zhang. Global convergence of the nonmonotone MBFGS method for nonconvex unconstrained minimization. *J. Comput. Appl. Math.*, 223:40–47, 2009.