

Universidad Central de Venezuela
Facultad de Ingeniería
Escuela de Ingeniería Eléctrica
“Melchor Centeno Vallenilla”



Algoritmos de Bajo Costo para Minimización Irrestringida y Sistemas de Ecuaciones no Lineales

Por

William René La Cruz Bastidas

Trabajo presentado ante la ilustre
Universidad Central de Venezuela como trabajo de
Ascenso para optar a la categoría de Profesor Agregado

Caracas, Noviembre 2005

*A Janetsi
y Victor Jehosmar*

Agradecimientos

Quisiera agradecer primeramente al Prof. Rafael Díaz quien me ha orientado sabiamente en el transcurso de mi formación como docente e investigador.

También debo extender mis agradecimientos a los profesores Arcángel Becerra y Marcos Raydan por sus acertadas orientaciones y correcciones.

Muy especialmente debo agradecer a mi esposa Janetsi por el apoyo incondicional que me ha brindado en todas las metas que he trazado.

A todos muchas gracias.

Este trabajo está soportado por el CDCH-UCV bajo el Proyecto Individual PI 08-14-5463-2004, titulado: *Métodos de Bajo Costo Computacional para la Resolución de Problemas en Ingeniería*.

William René La Cruz Bastidas

Universidad Central de Venezuela
Noviembre 2005

Resumen

El objetivo principal de esta investigación es el diseño y estudio de técnicas y algoritmos para problemas de minimización irrestricta y sistemas de ecuaciones no lineales. Específicamente, presentamos una búsqueda lineal no monótona para minimización irrestricta que consiste en perturbar la condición de Armijo con una sucesión de números positivos infinitamente sumable. Comprobamos teóricamente que la nueva técnica de globalización combinada con un método local, genera una sucesión que converge a un punto estacionario de la función mérito. Además, utilizando la nueva técnica de globalización diseñamos versiones globales de los métodos Newton y Barzilai-Borwein, que denominamos respectivamente Algoritmo de Newton Global y Nuevo Barzilai-Borwein Global (NGBB). Con problemas test de minimización irrestricta comprobamos numéricamente que los nuevos algoritmos globales compiten satisfactoriamente con los algoritmos de Newton y Barzilai-Borwein, que consideran respectivamente la condición de Armijo y la búsqueda lineal no monótona de Grippo, Lampariello y Lucidi. También comparamos la eficiencia computacional de NGBB con la función `fminunc` del toolbox de optimización de MATLAB 7.0 y el algoritmo GBB (Global Barzilai and Borwein), en la resolución del problema de Conformación Molecular. Nuestros resultados indican que NGBB compite satisfactoriamente con `fminunc` y GBB en la resolución del problema de Conformación Molecular.

Empleando la nueva búsqueda lineal redefinimos el algoritmo DF-SANE (Derivative free-Spectral Algorithm for Nonlinear Equations) para sistemas de ecuaciones no lineales. De esta forma, obtuvimos un nuevo algoritmo libre de derivadas para sistemas de ecuaciones no lineales, que denominamos NDF-SANE (New Derivative free-Spectral Algorithm for Nonlinear Equations). Comprobamos teóricamente la convergencia de NDF-SANE y realizamos pruebas numéricas con sistemas de ecuaciones no lineales de gran tamaño, donde constatamos que NDF-SANE compite satisfactoriamente con los algoritmos SANE y DF-SANE, y en algunos casos su comportamiento es mejor. También comparamos la eficiencia computacional de NDF-SANE con SANE y DF-SANE en la resolución del problema del Trazado de Rayos Sísmicos. Nuestros resultados indican que NDF-SANE compite satisfactoriamente con SANE y DF-SANE en la resolución del problema del Trazado de Rayos Sísmicos.

Índice

Agradecimientos	iii
Resumen	iv
Índice	v
Índice de Figuras	vii
Índice de Tablas	viii
Índice de Algoritmos	ix
Notación	x
Introducción	1
1 Conceptos y Métodos Básicos	5
1.1 Conceptos y Teoremas Básicos	5
1.1.1 Álgebra Lineal	5
1.1.2 Cálculo en Varias Variables	7
1.2 Sistemas de Ecuaciones no Lineales	8
1.2.1 Método de Newton	8
1.2.2 Métodos Casi-Newton	10
1.2.3 Métodos Newton Inexactos	15
1.3 Minimización Irrestric­ta	16
1.3.1 Métodos Tipo Newton	17
1.3.2 Métodos Tipo Gradiente	17
1.3.3 Búsqueda Lineal	19
2 Búsqueda Lineal No Monótona y Algoritmos para Minimización Irrestric­ta	21
2.1 Nueva Búsqueda Lineal no Monótona	21
2.2 Métodos Tipo Newton y Tipo Gradiente Globales	27
2.2.1 Algoritmo de Newton Global	27
2.2.2 Algoritmo de Barzilai-Borwein Global	29
2.3 Resultados Numéricos del Algoritmo de Newton Global	31

2.4	Resultados Numéricos del Algoritmo NGBB	35
2.4.1	Problemas Test	35
2.4.2	Problema de Conformación Molecular	39
3	Algoritmo Libre de Derivadas para Sistemas de Ecuaciones no Lineales	43
3.1	Algoritmo NDF-SANE	43
3.1.1	Análisis de Convergencia	45
3.2	Resultados Numéricos	56
3.2.1	Funciones test	56
3.2.2	Trazado de Rayos Sísmicos	62
	Conclusiones y Trabajos Futuros	68
	A Funciones Test	71
	Bibliografía	81

Índice de Figuras

2.1	Valores permitidos de λ bajo las condiciones de Armijo y (2.1) ($\gamma = 0.1$) . . .	22
2.2	Configuraciones óptimas de los problemas M19, M38 y M60	41
2.3	Mejor configuración obtenida por cada algoritmo para el problema M19 . . .	42
2.4	Mejor configuración obtenida por cada algoritmo para el problema M38 . . .	42
2.5	Mejor configuración obtenida por cada algoritmo para el problema M60 . . .	42
3.1	Medio Sintético	64
3.2	Trazado de Rayos para SANE y NDF-SANE en el caso (a)	65
3.3	Trazado de Rayos para SANE y NDF-SANE en el caso (b)	67

Índice de Tablas

1.1	Ventajas y desventajas del método de Newton	9
2.1	Resultados para los Problemas 1-8.	34
2.2	Resultados para el Problema 2	34
2.3	Resultados para el Problema 6	34
2.4	Resultados para el Problema 8	35
2.5	Problemas Test	36
2.6	Resultados numéricos para GBB y NGBB	38
2.7	Número de problemas en los que un algoritmo es ganador	39
2.8	Resultados para los problemas M19, M38 y M60	41
3.1	Resultados para los problemas 1-22	59
3.2	Resultados para los problemas 23-44	60
3.3	Número de problemas en los que un algoritmo es ganador	60
3.4	SANE vs. DF-SANE	61
3.5	SANE vs. NDF-SANE	61
3.6	DF-SANE vs. NDF-SANE	61
3.7	Resultados numéricos para SANE y NDF-SANE en el caso (a)	65
3.8	Resultados numéricos para SANE y NDF-SANE en el caso (b)	67

Índice de Algoritmos

1.1	Método de Newton para Sistemas de Ecuaciones No Lineales	9
1.2	Método Casi-Newton	10
1.3	Método de Newton Modificado	10
1.4	Método de Newton con Diferencia Finita	10
1.5	Método tipo Secante	11
1.6	Método de Broyden	11
1.7	Método Malo de Broyden	11
1.8	Método de Broyden Inverso	12
1.9	Método SANE	13
1.10	Método DF-SANE	14
1.11	Método de Newton para Minimización Irrestriccta	17
1.12	Método del Mínimo Descenso	17
1.13	Gradiente Espectral Globalizado	18
2.1	Algoritmo de Newton Global	28
2.2	Algoritmo NGBB	30
3.1	Algoritmo NDF-SANE	44

Notación

Los siguientes símbolos son usados a lo largo del texto.

\mathbb{N}	—	Conjunto de los números naturales
\mathbb{R}	—	Conjunto de los números reales
\mathbb{R}^n	—	espacio vectorial real de n dimensiones
$\mathbb{R}^{n \times n}$	—	espacio de matrices reales de orden n
$\ \cdot\ $	—	norma l_2 o Euclideana
$\langle v, w \rangle \equiv v^t w$	—	producto interno de v y w
$\kappa(A)$	—	número de condición 2 de la matriz A
$\nabla f(x)$	—	gradiente de f evaluado en x
$H(x)$	—	Matriz Hessiana de f evaluada en x
$J(x)$	—	Matriz Jacobiana de F evaluada en x
$C^k(D)$	—	Conjunto de funciones k -veces continuamente diferenciables en el conjunto D

Introducción

“El Análisis Numérico es la teoría de métodos constructivos en el análisis matemático”

Peter Henrici

Como parte de su resolución muchos problemas requieren encontrar el mínimo de una función y , muchos otros, requieren de la solución de un sistema de ecuaciones no lineales. Por ejemplo, problemas tan importantes como el problema de Conformación Molecular y el problema del Trazado de Rayos Sísmicos, se pueden modelar respectivamente como un problema de minimización irrestricta ([37], [72]) y un sistema de ecuaciones no lineales ([45],[46]).

Tradicionalmente los problemas de minimización irrestricta y sistemas de ecuaciones no lineales se resuelven empleando métodos tipo Newton combinados con una técnica de búsqueda lineal, que garantiza el decrecimiento monótono de la función objetivo ([26, 60, 73, 39, 44, 35, 49]). Estos métodos son atractivos porque convergen rápidamente para un iterado inicial suficientemente bueno.

El gran inconveniente de los métodos tipo Newton es que necesitan resolver un sistema de ecuaciones lineales en cada iteración usando la matriz Hessiana (en el caso de minimización irrestricta), y la matriz Jacobiana (en el caso de sistemas de ecuaciones no lineales), o aproximaciones de las mismas. Por ello, cuando la dimensión del problema es grande ($n \gg 10000$), el tiempo de cómputo de estos métodos es muy elevado, es decir, se tornan computacionalmente costosos.

Uno de los métodos para minimización irrestricta menos costosos y que se emplea con gran regularidad es el método Barzilai-Borwein [4]. Este método utiliza la dirección del gradiente negativo y la longitud de paso espectral. Raydan [62] combina una variante de la búsqueda lineal de Grippo, Lamparillo y Lucidi [40] con el método Barzilai-Borwein [4]

y obtiene el método Gradiente Espectral Globalizado, que ha mostrado ser muy robusto y efectivo en sus aplicaciones numéricas. En el año 2002, Grippo y Sciandrone [41] proponen una nueva estrategia de globalización para el método Barzilai-Borwein y reportan que este nuevo enfoque es eficiente en la solución de problemas de minimización irrestricta de gran dimensión.

Por otra parte, un procedimiento satisfactorio para la solución de un sistema de ecuaciones no lineales para valores grandes de n , es emplear los métodos Newton Inexactos que resuelven en forma aproximada el sistema lineal del método de Newton a través de procesos iterativos [5, 15, 16, 44]. La inexactitud ocurre ya que los métodos iterativos internos se detienen prematuramente en la resolución del sistema lineal, ocasionando, de esta forma, un bajo costo computacional por iteración. Modernas implementaciones usan métodos iterativos tipo Krylov (por ejemplo, TFQMR [31], GMRES [65] y Bi-CGSTAB [71]) combinados con el método de Newton y por ello se denominan métodos Newton-Krylov [6, 15, 16, 44]

En el año 2003, La Cruz y Raydan [48] desarrollan el algoritmo SANE (Spectral Algorithm for Nonlinear Equations), donde emplean la búsqueda lineal no monótona de Grippo, Lampariello y Lucidi [40], y reportan que SANE compite satisfactoriamente con métodos Newton-Krylov para sistemas de ecuaciones no lineales.

Recientemente, La Cruz, Martínez y Raydan [47] combinan y extienden las búsquedas lineales no monótonas de Grippo, Lampariello y Lucidi [40] y de Li y Fukushima [49], produciendo una nueva búsqueda lineal que se utiliza en el algoritmo DF-SANE (Derivative-free SANE) para sistemas de ecuaciones no lineales. La Cruz, Martínez y Raydan [47] reportan, además de resultados teóricos, que DF-SANE compite satisfactoriamente con SANE y en algunos casos su comportamiento computacional es mejor.

Las experiencias en la resolución de sistemas de ecuaciones no lineales nos permitieron desarrollar una nueva búsqueda lineal no monótona para problemas de minimización irrestricta. La nueva técnica que proponemos es una somera perturbación de la regla de Armijo. Específicamente, nosotros forzamos que se satisfaga la condición de Armijo adhiriéndole un término de una sucesión de números positivos. Con esta perturbación de la regla de Armijo pretendemos que se acepte la longitud del tamaño del paso con tanta frecuencia como sea

posible, garantizando, por su puesto, la convergencia global a un mínimo local.

Empleando la nueva búsqueda lineal diseñamos versiones globales de los métodos Newton y Barzilai-Borwein, que denominamos respectivamente Algoritmo de Newton Global y Nuevo Barzilai-Borwein Global (NGBB). Para estos algoritmos globales realizamos pruebas numéricas con problemas de minimización irrestricta, donde constatamos que la nueva búsqueda lineal mejora notablemente el comportamiento del método local, en comparación con el que se obtiene cuando se emplean las búsquedas lineales de Armijo o la de Grippo, Lampariello y Lucidi [40].

También comparamos la eficiencia computacional de NGBB con la función `fminunc` del toolbox de optimización de MATLAB 7.0 y el algoritmo GBB [62] (Global Barzilai and Borwein), en la resolución del problema de Conformación Molecular. Nuestros resultados indican que NGBB compite satisfactoriamente con `fminunc` y GBB en la resolución del problema de Conformación Molecular. Concretamente, NGBB emplea el menor número de evaluaciones de la función mérito y reduce sustancialmente el tiempo de CPU.

En el contexto de sistema de ecuaciones no lineales, el desarrollo de la nueva búsqueda lineal para minimización irrestricta, nos condujo a redefinir el algoritmo DF-SANE para sistemas de ecuaciones no lineales. Conservando el esquema fundamental de DF-SANE y empleando la nueva búsqueda lineal en lugar de la propuesta por La Cruz, Martínez y Raydan [47], obtuvimos un nuevo algoritmo libre de derivadas para sistemas de ecuaciones no lineales, que denominamos NDF-SANE (New Derivative free-Spectral Algorithm for Nonlinear Equations).

Realizamos un estudio teórico de convergencia de NDF-SANE y diseñamos pruebas numéricas donde comprobamos que compite satisfactoriamente con SANE y DF-SANE, y en algunos casos su comportamiento es mejor. También comparamos la eficiencia computacional de NDF-SANE con SANE y DF-SANE en la resolución del problema del Trazado de Rayos Sísmicos. Nuestros resultados indican que NDF-SANE compite satisfactoriamente con SANE y DF-SANE en la resolución del problema del Trazado de Rayos Sísmicos. Concretamente, NDF-SANE disminuye sustancialmente el tiempo de CPU y necesita de menos número de evaluaciones de la función mérito.

El resto del trabajo lo organizamos de la siguiente manera. En el Capítulo 1 presentamos los conceptos, teoremas y métodos que utilizamos en el análisis posterior. En el Capítulo 2 desarrollamos la nueva búsqueda lineal no monótona y las nuevas versiones globales de los métodos Newton y Barzilai-Borwein. En el Capítulo 3 introducimos un nuevo algoritmo libre de derivadas para sistemas de ecuaciones no lineales. Finalmente, presentamos las conclusiones y trabajos futuros.

Capítulo 1

Conceptos y Métodos Básicos

*“El Análisis Numérico es el análisis
de algoritmos continuos”*

Joseph Traub

Para comenzar nuestro estudio de la solución de problemas de minimización irrestricta y sistemas de ecuaciones no lineales, es necesario establecer las definiciones, teoremas y métodos básicos que permiten fundamentar las ideas que desarrollamos. En la Sección 1.1 se presentan los tópicos de álgebra lineal y cálculo en varias variables imprescindibles para implementar y analizar algoritmos para problemas de varias variables. Respectivamente, en las Secciones 1.2 y 1.3, se describen los métodos más relevantes para la solución numérica de problemas de sistemas de ecuaciones no lineales y minimización irrestricta. Excelentes referencias para este material son los libros: Dennis y Schnabel [25], Trefethen y Bau [69] y Ortega y Rheinboldt [60].

1.1 Conceptos y Teoremas Básicos

A continuación revisamos algunos conceptos y teoremas básicos del álgebra lineal y el cálculo en varias variables.

1.1.1 Álgebra Lineal

Definición 1.1 (Norma Euclideana). La norma l_2 o *Euclideana* de $w = (w_1, \dots, w_n)^t \in \mathbb{R}^n$, que denotamos con $\|\cdot\|$, está definida como:

$$\|w\| = \sqrt{w_1^2 + w_2^2 + \dots + w_n^2}.$$

La norma Euclídeana de una matriz $A \in \mathbb{R}^{n \times n}$ está definida como:

$$\|A\| = \max_{w \in \mathbb{R}^n, w \neq 0} \left\{ \frac{\|Aw\|}{\|w\|} \right\}.$$

Definición 1.2 (Producto interno). Sean $v, w \in \mathbb{R}^n$; el *producto interno* de v y w está definido por

$$\langle v, w \rangle \equiv v^t w = \sum_{i=1}^n v_i w_i = \|v\| \cdot \|w\| \cdot \cos \theta,$$

donde θ es el ángulo entre v y w , si $v \neq 0 \neq w$. Si $\langle v, w \rangle = 0$ se dice que v y w son ortogonales.

Definición 1.3 (Número de Condición). El *número de condición* de una matriz $A \in \mathbb{R}^{n \times n}$ está definido como:

$$\kappa(A) = \|A\| \cdot \|A^{-1}\|.$$

Definición 1.4 (Autovalores y Autovectores). Sea $A \in \mathbb{R}^{n \times n}$. Los *autovalores* y *autovectores* de A son escalares reales o complejos λ y vectores n -dimensionales v tales que $Av = \lambda v$, $v \neq 0$.

Definición 1.5. Sea $A \in \mathbb{R}^{n \times n}$ simétrica. Se dice que A es *positivo definida* si $v^t A v > 0$ para todo $v \neq 0$, $v \in \mathbb{R}^n$. Se dice que A es *positivo semidefinida* si $v^t A v \geq 0$ para todo $v \in \mathbb{R}^n$. Se dice que A es *negativo definida* o *negativo semidefinida* si $-A$ es positivo definida o positivo semidefinida, respectivamente. Se dice que A es *indefinida* si no es ni positivo semidefinida ni negativo semidefinida.

Teorema 1.1. Sea $A \in \mathbb{R}^{n \times n}$ simétrica. Entonces, A tiene n autovalores reales $\alpha_1, \dots, \alpha_n$ y un correspondiente conjunto de autovectores v_1, \dots, v_n que forman una base ortonormal de \mathbb{R}^n .

Teorema 1.2. Sea $A \in \mathbb{R}^{n \times n}$ simétrica. Entonces, A es positivo definida si, y sólo si, todos sus autovalores son positivos.

Teorema 1.3. Sea $A \in \mathbb{R}^{n \times n}$ simétrica con autovalores $\alpha_1, \dots, \alpha_n$. Entonces,

$$\|A\| = \max_{1 \leq i \leq n} |\alpha_i|.$$

Si A es no-singular, entonces

$$\kappa(A) = \frac{\max_{1 \leq i \leq n} |\alpha_i|}{\min_{1 \leq i \leq n} |\alpha_i|}.$$

1.1.2 Cálculo en Varias Variables

Definición 1.6. Una función continua $f : \mathbb{R}^n \rightarrow \mathbb{R}$ es *continuamente diferenciable* en $x \in \mathbb{R}^n$, si $(\partial f / \partial x_i)(x)$ existe y es continua, $i = 1, \dots, n$; entonces el *gradiente* de f en x está definido como

$$\nabla f(x) = \left(\frac{\partial f}{\partial x_1}(x), \dots, \frac{\partial f}{\partial x_n}(x) \right)^t.$$

La función f es continuamente diferenciable en un conjunto abierto $D \subset \mathbb{R}^n$, denotado por $f \in C^1(D)$, si ella es continuamente diferenciable en cada punto de D .

Definición 1.7. Una función continua $f : \mathbb{R}^n \rightarrow \mathbb{R}$ es *dos veces continuamente diferenciable* en $x \in \mathbb{R}^n$, si $(\partial^2 f / \partial x_i^2)(x)$ existe y es continua, $i = 1, \dots, n$; entonces la *Hessiana* de f en x , $H(x)$, está definida como una matriz $n \times n$ cuyo elemento i, j es

$$H(x)_{ij} = \frac{\partial^2 f}{\partial x_i^2}(x), \quad 1 \leq i, 1 \leq j.$$

La función f es dos veces continuamente diferenciable en un conjunto abierto $D \subset \mathbb{R}^n$, denotado por $f \in C^2(D)$, si ella es dos veces continuamente diferenciable en cada punto de D .

Teorema 1.4 (Teorema Fundamental del Cálculo). Sean D un conjunto abierto y $f \in C^1(D)$. Sean $x \in D$ y $p \in \mathbb{R}^n$ tal que $x + p \in D$. Entonces,

$$f(x + p) - f(x) = \int_0^1 \nabla f(x + \tau p)^t d\tau.$$

Teorema 1.5 (Teorema del Valor Medio). Sean D un conjunto abierto y $f \in C^1(D)$. Sean $x \in D$ y $p \in \mathbb{R}^n$ tal que $x + p \in D$. Entonces, para algún z perteneciente al segmento que conecta a x con p se satisface que

$$f(x + p) - f(x) = \nabla f(z)^t p.$$

Definición 1.8. Sea $F : \mathbb{R}^n \rightarrow \mathbb{R}^m$ una función continua tal que $F(x) = (f_1(x), \dots, f_m(x))^t$ y $f_i : \mathbb{R}^n \rightarrow \mathbb{R}$ es una función componente, $i = 1, \dots, m$. Se dice que F es continuamente diferenciable en $x \in \mathbb{R}^n$ si cada f_i , $i = 1, \dots, m$ es continuamente diferenciable en x . La derivada de F en x es llamada algunas veces *Jacobiano* de F en x , denotado por $J(x)$, y su elemento i, j es

$$J(x)_{ij} = \frac{\partial f_i}{\partial x_j}(x), \quad 1 \leq i, 1 \leq j.$$

F se dice continuamente diferenciable en un conjunto abierto $D \subset \mathbb{R}^n$, denotado por $F \in C^1(D)$, si F es continuamente diferenciable en cada punto de D .

1.2 Sistemas de Ecuaciones no Lineales

Sea $F : \mathbb{R}^n \rightarrow \mathbb{R}^m$ una función cualquiera. Estamos interesados en resolver la ecuación

$$F(x) = 0. \tag{1.1}$$

Es decir, encontrar un punto $x_* \in \mathbb{R}^n$ tal que $F(x_*) = 0$. El caso de interés para nosotros es $m = n$.

La ecuación (1.1) es llamada *sistema no lineal* y x es el vector de incógnitas. La solución numérica de (1.1) normalmente se obtiene mediante un esquema iterativo, es decir, moviéndose en cada iteración de una solución aproximada x_k a otra solución x_{k+1} .

Los métodos que expondremos a continuación encuentran una solución aproximada del problema (1.1) y los clasificamos en tres grupos: *Método de Newton*, *Métodos Casi-Newton* y *Métodos Newton Inexacto*.

1.2.1 Método de Newton

El método de Newton se muestra en el Algoritmo 1.1. En el contexto de sistemas de ecuaciones el método de Newton aparece en el siglo XVII ([59], [61]). Este método posee características excepcionales en un sentido local, es decir, posee convergencia cuadrática localmente, pero tiene un gran inconveniente. En cada iteración se debe resolver el sistema lineal $J(x_k)s_k = -F(x_k)$, que resulta muy costoso cuando la dimensión del problema es

grande ($n \gg 10^4$). En la Tabla 1.1 (Dennis y Schnabel [25]) se sumergen las ventajas y desventajas del método de Newton. Para un estudio más detallado del método de Newton se recomienda el libro de Dennis y Schnabel [25].

Algoritmo 1.1 Método de Newton para Sistemas de Ecuaciones No Lineales

Datos: $x_0 \in \mathbb{R}^n$ y $k \leftarrow 0$.

- 1: **mientras** $F(x_k) \neq 0$ **hacer**
 - 2: resolver el sistema $J(x_k)s_k = -F(x_k)$
 - 3: $x_{k+1} \leftarrow x_k + s_k$
 - 4: $k \leftarrow k + 1$
 - 5: **fin mientras**
-

Tabla 1.1: Ventajas y desventajas del método de Newton

Ventajas

1. Convergencia q-cuadrática para buenos iterados iniciales si $J(x_*)$ es no-singular.
2. Solución exacta en una iteración para una función afín F .

Desventajas

1. No es globalmente convergente para muchos problemas.
 2. Requiere de $J(x_k)$ en cada iteración.
 3. Cada iteración requiere la solución de un sistema de ecuaciones lineales que puede ser mal condicionado.
-

1.2.2 Métodos Casi-Newton

En el Algoritmo 1.2 se describe la forma general de los métodos Casi-Newton.

Algoritmo 1.2 Método Casi-Newton

Datos: $x_0 \in \mathbb{R}^n$, $H_0 \in \mathbb{R}^{n \times n}$, y $k \leftarrow 0$.

- 1: **mientras** $F(x_k) \neq 0$ **hacer**
 - 2: resolver el sistema $H_k s_k = -F(x_k)$
 - 3: $x_{k+1} \leftarrow x_k + s_k$
 - 4: $k \leftarrow k + 1$
 - 5: **fin mientras**
-

La matriz H_k aproxima a $J(x_k)$. Después del cálculo de x_{k+1} , H_{k+1} se obtiene utilizando H_k . La ventaja de estos métodos es que la resolución del sistema lineal $H_k s_k = -F(x_k)$ resulta a menudo más barata que la resolución del sistema $J(x_k) s_k = -F(x_k)$. A continuación mostramos el esquema general de los métodos Casi-Newton más utilizados, a saber: Newton Modificado, Newton con Diferencia Finita y tipo Secante.

Algoritmo 1.3 Método de Newton Modificado

Datos: $x_0 \in \mathbb{R}^n$ y $k \leftarrow 0$.

- 1: **mientras** $F(x_k) \neq 0$ **hacer**
 - 2: resolver el sistema $J(x_k) s_k = -F(x_k)$
 - 3: $x_{k+1} \leftarrow x_k + s_k$
 - 4: $k \leftarrow k + 1$
 - 5: **fin mientras**
-

Algoritmo 1.4 Método de Newton con Diferencia Finita

Datos: $x_0 \in \mathbb{R}^n$ y $k \leftarrow 0$.

- 1: **mientras** $F(x_k) \neq 0$ **hacer**
 - 2: $(H_k)_j \leftarrow \frac{1}{h_k} [F(x_k + h_k e_j) - F(x_k)]$
 - 3: resolver el sistema $H_k s_k = -F(x_k)$
 - 4: $x_{k+1} \leftarrow x_k + s_k$
 - 5: $k \leftarrow k + 1$
 - 6: **fin mientras**
-

En el Algoritmo 1.4 la matriz $H_k \approx J(x_k)$, $h_k > 0$ es pequeño y e_j es el vector cuya única componente distinta de cero e igual a uno es la j -ésima.

Algoritmo 1.5 Método tipo Secante

Datos: $x_0 \in \mathbb{R}^n$, $A_0 \in \mathbb{R}^{n \times n}$, y $k \leftarrow 0$.

- 1: **mientras** $F(x_k) \neq 0$ **hacer**
 - 2: resolver el sistema $A_k s_k = -F(x_k)$
 - 3: $x_{k+1} \leftarrow x_k + s_k$
 - 4: $k \leftarrow k + 1$
 - 5: **fin mientras**
-

En el Algoritmo 1.5 la matriz A_k satisface la ecuación de la secante,

$$A_k s_{k-1} = y_{k-1}, \quad (1.2)$$

donde $s_{k-1} = x_k - x_{k-1}$ y $y_{k-1} = F(x_k) - F(x_{k-1})$. Los métodos tipo secante más utilizados son: Broyden ([17]), Malo de Broyden y Broyden Inverso.

Algoritmo 1.6 Método de Broyden

Datos: $x_0 \in \mathbb{R}^n$, $A_0 \in \mathbb{R}^{n \times n}$, y $k \leftarrow 0$.

- 1: **mientras** $F(x_k) \neq 0$ **hacer**
 - 2: resolver el sistema $A_k s_k = -F(x_k)$
 - 3: $x_{k+1} \leftarrow x_k + s_k$
 - 4: $y_k \leftarrow F(x_{k+1}) - F(x_k)$
 - 5: $A_{k+1} \leftarrow A_k + \frac{(y_k - A_k s_k) s_k^t}{s_k^t s_k}$
 - 6: $k \leftarrow k + 1$
 - 7: **fin mientras**
-

El método de Broyden consiste en “escoger A_{k+1} lo más cercana posible A_k y que satisfaga la ecuación de la secante (1.2)”.

Algoritmo 1.7 Método Malo de Broyden

Datos: $x_0 \in \mathbb{R}^n$, $A_0 \in \mathbb{R}^{n \times n}$, y $k \leftarrow 0$.

- 1: **mientras** $F(x_k) \neq 0$ **hacer**
 - 2: resolver el sistema $A_k s_k = -F(x_k)$
 - 3: $x_{k+1} \leftarrow x_k + s_k$
 - 4: $y_k \leftarrow F(x_{k+1}) - F(x_k)$
 - 5: $A_{k+1} \leftarrow A_k + \frac{(y_k - A_k s_k) y_k^t}{y_k^t A_k s_k} A_k$
 - 6: $k \leftarrow k + 1$
 - 7: **fin mientras**
-

Algoritmo 1.8 Método de Broyden Inverso

Datos: $x_0 \in \mathbb{R}^n$, $A_0 \in \mathbb{R}^{n \times n}$, y $k \leftarrow 0$.

- 1: **mientras** $F(x_k) \neq 0$ **hacer**
 - 2: resolver el sistema $A_k s_k = -F(x_k)$
 - 3: $x_{k+1} \leftarrow x_k + s_k$
 - 4: $y_k \leftarrow F(x_{k+1}) - F(x_k)$
 - 5: $A_{k+1}^{-1} \leftarrow A_k^{-1} + \frac{(s_k - A_k^{-1} y_k) s_k^t A_k^{-1}}{s_k^t A_k^{-1} y_k}$
 - 6: $k \leftarrow k + 1$
 - 7: **fin mientras**
-

Para un estudio más detallado de los métodos Casi-Newton, se recomienda el libro de Dennis y Schnabel [25], y también el libro de Fletcher [30].

Métodos SANE y DF-SANE

Los métodos SANE y DF-SANE desarrollados respectivamente por La Cruz y Raydan [48] y La Cruz, Martínez y Raydan [47], se pueden considerar como métodos Casi-Newton que emplean sistemáticamente la dirección $\pm F(x)$. La iteración de cada uno de estos métodos puede definirse como

$$x_{k+1} = x_k - \lambda_k B_k^{-1} F(x_k),$$

donde B_k es una aproximación del Jacobiano que tiene la simple forma $B_k = (1/\alpha_k)I$, con $\alpha_k \in \mathbb{R}$ el coeficiente espectral.

Los métodos SANE y DF-SANE se muestran en los Algoritmos 1.9 y 1.10, respectivamente. En el Algoritmo 1.9 la función $f : \mathbb{R}^n \rightarrow \mathbb{R}$ está definida como

$$f(x) = \|F(x)\|^2,$$

y en el Algoritmo 1.10 está definida como

$$f(x) = \|F(x)\|^{nexp},$$

donde $nexp \in \{1, 2\}$.

Algoritmo 1.9 Método SANE

Datos: $x_0 \in \mathbb{R}^n$, α_0 , un entero $M > 0$, $0 < \gamma < 1$, $0 < \sigma_1 < \sigma_2 < 1$, $0 < \varepsilon < 1$, $\delta > 0$, y $k \leftarrow 0$.

- 1: **mientras** $F(x_k) \neq 0$ **hacer**
 - 2: **si** $|F(x_k)^t J(x_k) F(x_k)| / F(x_k)^t F(x_k) < \varepsilon$ **entonces**
 - 3: parar el proceso
 - 4: **fin si**
 - 5: **si** $\alpha_k \notin (\varepsilon, \varepsilon^{-1})$ **entonces**
 - 6: $\alpha_k \leftarrow \delta$
 - 7: **fin si**
 - 8: $\text{sgn}_k \leftarrow \text{sgn}(F(x_k)^t J(x_k) F(x_k))$
 - 9: $d_k \leftarrow -\text{sgn}_k F(x_k)$
 - 10: $\lambda \leftarrow 1/\alpha_k$
 - 11: **mientras** $f(x_k + \lambda d_k) > \max_{0 \leq j \leq \min(k, M)} [f(x_{k-j})] + 2\gamma \lambda F(x_k)^t J(x_k) d_k$ **hacer**
 - 12: escoger $\sigma \in [\sigma_1, \sigma_2]$
 - 13: $\lambda \leftarrow \sigma \lambda$
 - 14: **fin mientras**
 - 15: $\lambda_k \leftarrow \lambda$
 - 16: $x_{k+1} \leftarrow x_k + \lambda_k d_k$
 - 17: $y_k \leftarrow F(x_{k+1}) - F(x_k)$
 - 18: $\alpha_{k+1} \leftarrow \text{sgn}_k((d_k^t y_k) / (\lambda_k d_k^t d_k))$
 - 19: $k \leftarrow k + 1$
 - 20: **fin mientras**
-

Algoritmo 1.10 Método DF-SANE

Datos: $x_0 \in \mathbb{R}^n$, un entero $M > 0$, $0 < \gamma < 1$, $0 < \sigma_{min} < \sigma_{max} < \infty$, $0 < \tau_{min} < \tau_{max} < 1$, una sucesión $\{\eta_k\}_{k \in \mathbb{N}}$ tal que $\sum_{k=0}^{\infty} \eta_k < \eta < \infty$, y $k \leftarrow 0$.

1: **mientras** $F(x_k) \neq 0$ **hacer**
 2: escoger σ_k tal que $|\sigma_k| \in [\sigma_{min}, \sigma_{max}]$ (coeficiente espectral)
 3: $\bar{f}_k \leftarrow \max\{f(x_k), \dots, f(x_{\max\{0, k-M+1\}})\}$
 4: $d \leftarrow -\sigma_k F(x_k)$
 5: $\lambda_+ \leftarrow 1$
 6: $\lambda_- \leftarrow 1$
 7: *busqueda lineal* \leftarrow verdadero
 8: **mientras** *busqueda lineal* = verdadero **hacer**
 9: **si** $f(x_k + \lambda_+ d) \leq \bar{f}_k + \eta_k - \gamma \lambda_+^2 f(x_k)$ **entonces**
 10: $d_k \leftarrow d$
 11: $\lambda_k \leftarrow \lambda_+$
 12: *busqueda lineal* \leftarrow falso
 13: **de lo contrario**
 14: **si** $f(x_k - \lambda_- d) \leq \bar{f}_k + \eta_k - \gamma \lambda_-^2 f(x_k)$ **entonces**
 15: $d_k \leftarrow -d$
 16: $\lambda_k \leftarrow \lambda_-$
 17: *busqueda lineal* \leftarrow falso
 18: **de lo contrario**
 19: escoger $\lambda_{+new} \in [\tau_{min} \lambda_+, \tau_{max} \lambda_+]$ y $\lambda_{-new} \in [\tau_{min} \lambda_-, \tau_{max} \lambda_-]$
 20: $\lambda_+ \leftarrow \lambda_{+new}$
 21: $\lambda_- \leftarrow \lambda_{-new}$
 22: **fin si**
 23: **fin si**
 24: **fin mientras**
 25: $x_{k+1} \leftarrow x_k + \lambda_k d_k$
 26: $k \leftarrow k + 1$
 27: **fin mientras**

La diferencia primordial entre SANE y DF-SANE es la manera de como escogen el tamaño de paso. EL método SANE utiliza una modificación de la búsqueda lineal no monótona de Grippo, Lampariello y Lucidi [40]; en cambio, DF-SANE emplea una búsqueda lineal que combina la propuesta por Grippo, Lampariello y Lucidi [40] con la diseñada por Li y Fukushima [49]. Para un estudio más detallado de los métodos SANE y DF-SANE se recomiendan las referencias [45], [48] y [47].

1.2.3 Métodos Newton Inexactos

Los métodos tipo *Newton Inexactos* son muy apropiados para sistemas de ecuaciones no lineales de gran tamaño. Tales métodos emplean un proceso iterativo para encontrar una solución aproximada del sistema lineal, $J(x_k)s_k = -F(x_k)$, del método de Newton ([23, 28, 54, 66]). La utilización de los métodos iterativos para resolver el sistema lineal anterior origina una cierta inexactitud, pero con un bajo costo computacional por iteración. Formalmente, los métodos Newton inexactos según Dembo et al [23] y Kelley [44] consisten en: dado $x_0 \in \mathbb{R}^n$, generar $x_{k+1} \in \mathbb{R}^n$ como

$$x_{k+1} = x_k + s_k,$$

donde $s_k \in \mathbb{R}^n$ satisface la desigualdad

$$\|J(x_k)s_k + F(x_k)\|_2 \leq \beta_k \|F(x_k)\|_2,$$

con $\{\beta_k\}_{k \in \mathbb{N}}$ una sucesión de números reales positivos definida con anterioridad.

Implementaciones modernas de métodos Newton inexactos emplean métodos iterativos basados en procesos de proyección sobre subespacios de Krylov ([56]) para resolver el sistema lineal $J(x_k)s_k = -F(x_k)$ [5, 15, 16, 44]. Como ejemplo de estas implementaciones podemos nombrar:

- *Newton inexacto con GMRES (Newton-GMRES)*. Newton inexacto con la regla de Armijo y la fórmula de Eisenstat-Walker [28]. Utiliza GMRES para resolver sistemas de ecuaciones lineales. Newton-GMRES posee código en MATLAB dado por Kelley [44].

- *Newton Inexacto con Bi-CGSTAB (Newton-Bi-CGSTAB)*. Newton inexacto con la regla de Armijo y la fórmula de Eisenstat-Walker [28]. Utiliza Bi-CGSTAB para resolver sistemas de ecuaciones lineales. Newton-Bi-CGSTAB posee código en MATLAB dado por Kelley [44].
- *Newton Inexacto con TFQMR (Newton-TFQMR)*. Newton inexacto con la regla de Armijo y la fórmula de Eisenstat-Walker [28]. Utiliza TFQMR para resolver sistemas de ecuaciones lineales. Newton-TFQMR posee código en MATLAB dado por Kelley [44].

1.3 Minimización Irrestringida

Dada una función $f : \mathbb{R}^n \rightarrow \mathbb{R}$ diferenciable, el problema que consideramos es:

$$\min_{x \in \mathbb{R}^n} f(x), \quad (1.3)$$

que denominamos *minimización irrestringida*.

La idea básica de un método para el problema de minimización irrestringida es escoger una dirección d en el punto actual x_c en el que f disminuye inicialmente, y luego obtener un nuevo punto x_+ en esa dirección desde x_c tal que $f(x_+) < f(x_c)$. Semejante dirección se llama *dirección de descenso*. Matemáticamente, d es una dirección de descenso en x_c si la derivada direccional de f en x_c en la dirección d es negativa, esto es,

$$\nabla f(x_c)^t d < 0. \quad (1.4)$$

Si (1.4) es cierta, entonces esto es una garantía de que para $\varepsilon > 0$ pequeño,

$$f(x_c + \varepsilon d) < f(x_c).$$

Las direcciones de descenso son la base de los métodos para minimización y son importantes en todos ellos (Dennis y Schnabel [25]).

1.3.1 Métodos Tipo Newton

Generalmente un método para minimización recibe el nombre según el tipo de dirección de descenso que genera. Los *métodos tipo Newton* son aquellos que generan una dirección de descenso tipo Newton $d^N = A_c^{-1}\nabla f(x_c)$, donde A_c es una matriz simétrica y positivo definida tal que $A_c \approx H(x_c)$. Un ejemplo de estos métodos tipo Newton es el método de Newton para minimización irrestricta cuyo esquema es el siguiente.

Algoritmo 1.11 Método de Newton para Minimización Irrestricta

Datos: $x_0 \in \mathbb{R}^n$ y $k \leftarrow 0$.

- 1: **mientras** $\nabla f(x_k) \neq 0$ **hacer**
 - 2: resolver el sistema $H(x_k)s_k = -\nabla f(x_k)$
 - 3: $x_{k+1} \leftarrow x_k + s_k$
 - 4: $k \leftarrow k + 1$
 - 5: **fin mientras**
-

El Algoritmo 1.11 es simplemente la aplicación del Método de Newton al sistema de ecuaciones no lineales $\nabla f(x) = 0$. Bajo esta perspectiva, las ventajas y desventajas de este algoritmo para minimización se muestran claramente en la Tabla 1.1.

1.3.2 Métodos Tipo Gradiente

Los métodos tipo gradiente son aquellos que generan una dirección de descenso de la forma $d^G = -\nabla f(x_c)$. Un clásico método tipo gradiente es el método del mínimo descenso o método de Cauchy, cuyo esquema se muestra a continuación.

Algoritmo 1.12 Método del Mínimo Descenso

Datos: $x_0 \in \mathbb{R}^n$ y $k \leftarrow 0$.

- 1: **mientras** $\nabla f(x_k) \neq 0$ **hacer**
 - 2: $\lambda_k \leftarrow \operatorname{argmín}_{\lambda > 0} f(x_k - \lambda \nabla f(x_k))$
 - 3: $x_{k+1} \leftarrow x_k - \lambda_k \nabla f(x_k)$
 - 4: $k \leftarrow k + 1$
 - 5: **fin mientras**
-

Este método no es computacionalmente eficiente, porque cada paso contiene un problema de minimización en una dirección. Además, la convergencia es solamente lineal y algunas veces muy lenta.

Método de Barzilai-Borwein

Un método tipo gradiente muy particular es el método de Barzilai-Borwein [4] (BB) que está definido como:

$$x_{k+1} = x_k - \frac{1}{\alpha_k} \nabla f(x_k),$$

donde el escalar α_k está dado por

$$\alpha_k = \frac{s_{k-1}^t y_{k-1}}{s_{k-1}^t s_{k-1}}, \quad (1.5)$$

donde $s_{k-1} = x_k - x_{k-1}$ y $y_{k-1} = \nabla f(x_k) - \nabla f(x_{k-1})$.

Para el problema (1.3), Raydan [62] propone combinar una variante de la búsqueda lineal de Grippo, Lamparillo y Lucidi [40] con BB, obteniendo el método Gradiente Espectral Globalizado que se describe en el Algoritmo 1.13.

Algoritmo 1.13 Gradiente Espectral Globalizado

Datos: $x_0 \in \mathbb{R}^n$, α_0 , un entero $M > 0$, $\delta > 0$, $0 < \sigma_1 < \sigma_2 < 1$, $0 < \varepsilon < 1$, $0 < \gamma, < 1$, y $k \leftarrow 0$.

- 1: **mientras** $\nabla f(x_k) \neq 0$ **hacer**
 - 2: **si** $\alpha_k \notin (\varepsilon, \varepsilon^{-1})$ **entonces**
 - 3: $\alpha_k \leftarrow \delta$
 - 4: **fin si**
 - 5: $\lambda \leftarrow 1/\alpha_k$
 - 6: **mientras** $f(x_k - \lambda \nabla f(x_k)) > \max_{0 \leq j \leq \min(k, M)} [f(x_{k-j})] - \gamma \lambda \nabla f(x_k)^t \nabla f(x_k)$ **hacer**
 - 7: escoger $\sigma \in [\sigma_1, \sigma_2]$
 - 8: $\lambda \leftarrow \sigma \lambda$
 - 9: **fin mientras**
 - 10: $\lambda_k \leftarrow \lambda$
 - 11: $x_{k+1} \leftarrow x_k - \lambda_k \nabla f(x_k)$
 - 12: $y_k \leftarrow \nabla f(x_{k+1}) - \nabla f(x_k)$
 - 13: $\alpha_{k+1} \leftarrow -(\nabla f(x_k)^t y_k) / (\lambda_k \nabla f(x_k)^t \nabla f(x_k))$
 - 14: $k \leftarrow k + 1$
 - 15: **fin mientras**
-

El Gradiente Espectral ha mostrado ser muy robusto y efectivo en sus aplicaciones numéricas. Se ha utilizado en el problema químico de conformación molecular ([37] y [72]); en el trazado de rayos sísmicos ([22]) y en la tomografía de reflexión sísmica ([18]); en la estimación de constantes ópticas ([10]); en el problema de escalamiento multidimensional que

aparece en psicometría y en estadística ([14], [52] y [70]) y en la diferenciación automática de funciones ([11]). Versiones preconditionadas del método se han usado en la solución numérica de ecuaciones en derivadas parciales ([33], [52] y [55]). El método se ha extendido al problema de minimizar funciones sobre conjuntos convexos ([7], [34] y [12]). También se ha extendido al problema general de programación no lineal mediante técnicas Lagrangianas ([27]).

1.3.3 Búsqueda Lineal

La búsqueda lineal es una técnica de globalización que consiste en: dada una dirección de descenso d_k , escoger un tamaño de paso λ_k de tal manera que se garantice la convergencia global. Es decir, en cada iteración k :

- se calcula la dirección de descenso d_k ,
- se escoge $\lambda_k > 0$ de tal forma que $x_{k+1} = x_k + \lambda_k d_k$ satisface cierta condición que permita garantizar convergencia.

Las condiciones más empleadas que deben satisfacer λ_k y x_{k+1} son:

Condición α de Armijo [3]. Dados $\alpha \in (0, 1)$ y una dirección de descenso d , se escoge λ_c entre los $\lambda > 0$ tales que

$$f(x_c + \lambda d) \leq f(x_c) + \alpha \lambda \nabla f(x_c)^t d. \quad (1.6)$$

Condición β de Goldstein [38]. Dados $\alpha \in (0, 1)$, $\beta \in (\alpha, 1)$ y una dirección de descenso d , se escoge λ_c entre los $\lambda > 0$ tales que

$$\nabla f(x_c + \lambda d) \geq \beta \nabla f(x_c)^t d. \quad (1.7)$$

La condición $\beta > \alpha$ garantiza que (1.6) y (1.7) puedan satisfacerse simultáneamente. En la práctica, (1.7) generalmente no es necesaria porque el uso de la estrategia de backtracking arroja pasos excesivamente pequeños (Dennis y Schnabel [25]).

La condición de Armijo induce un decrecimiento monótono en la función f en cada iteración. Este tipo de comportamiento caracteriza a las búsquedas lineales monótonas. Exis-

ten otras búsquedas lineales que no necesariamente disminuyen a la función objetivo en cada iteración, pero garantizan convergencia. Un ejemplo de estas búsquedas lineales no-monótonas es la propuesta por Grippo, Lampariello y Lucidi [40]. La condición de Grippo, Lampariello y Lucidi [40] (GLL) es:

$$f(x_k + \lambda d_k) \leq \max_{0 \leq j \leq m(k)} [f(x_{k-j})] + \alpha \lambda \nabla f(x_k)^t d_k, \quad (1.8)$$

donde $m(0) = 0$ y $0 \leq m(k) \leq \min\{m(k-1) + 1, M\}$, $k \geq 0$, con M un entero positivo.

Una variante de GLL es definir $m(k)$ como:

$$m(k) = \begin{cases} 0, & \text{para } k < N, \\ \min[m(k-1) + 1, M], & \text{para } k \geq N, \end{cases}$$

donde N es un entero positivo.

Otro ejemplo de una búsqueda lineal no-monótona es la de Li y Fukushima [49] para sistemas de ecuaciones no lineales. La condición de Li y Fukushima [49] (LF) es:

$$\|F(x_k + \lambda d_k)\| \leq (1 + \eta_k) \|F(x_k)\| - \alpha \lambda^2 \|d_k\|^2, \quad (1.9)$$

donde $F : \mathbb{R}^n \rightarrow \mathbb{R}^n$, $\{\eta_k\}_{k \in \mathbb{N}}$ es una sucesión de números positivos tal que $\sum_k \eta_k < \eta < \infty$, y d_k es una dirección de descenso de la función mérito $f(x) = \frac{1}{2} \|F(x)\|^2$.

Capítulo 2

Búsqueda Lineal No Monótona y Algoritmos para Minimización Irrestric­ta

“El Análisis Numérico es el estudio de errores de redondeo”

En este capítulo presentamos una nueva búsqueda lineal no monótona para minimización irrestric­ta. También presentamos versiones globales de los métodos Newton y BB. En la Sección 2.1 se define una nueva búsqueda lineal. En la Sección 2.2 se construye las nuevas versiones globales de los métodos Newton y BB. En la Sección 2.3 se presentan pruebas numéricas de la versión global del método de Newton para problemas test. Finalmente, en la Sección 2.4 se muestran resultados numéricos del algoritmo NGBB para problemas test y el problema de Conformación Molecular.

2.1 Nueva Búsqueda Lineal no Monótona

Consideramos el problema de minimización irrestric­ta (1.3) con $f : \mathbb{R}^n \rightarrow \mathbb{R}$ una función diferenciable en \mathbb{R}^n .

La idea básica de un método global para minimización es la combinación de un método local con una estrategia de globalización que garantice la convergencia a un mínimo local. Normalmente se emplean búsquedas lineales, por ejemplo, la condición de Armijo (1.6) o la condición GLL (1.8), en conjunción con métodos tipo Newton o tipo gradiente.

La condición de Armijo es una búsqueda lineal monótona, es decir, exige decrecimiento de la función mérito en cada iteración. Por el contrario la condición GLL es una búsqueda li-

neal no monótona, o sea, admite crecimiento de la función mérito en algunas iteraciones pero garantiza la convergencia a un mínimo local. Ahora bien, por las características (monótono o no monótono) de un método local, es conveniente utilizar una búsqueda lineal que conserve tales propiedades. Por ejemplo, el método Barzilai-Borwein es no monótono para funciones estrictamente convexas y su extensión para cualquier función, emplea la condición GLL (Ver Raydan [62]).

La búsqueda lineal que proponemos y, que a continuación definimos, posee la característica no monótona de GLL y es una modificación de la regla de Armijo. Dados $\gamma \in (0, 1)$, $x_k \in \mathbb{R}^n$ y d una dirección de descenso, se escoge λ_c entre los $\lambda > 0$ tal que:

$$f(x_k + \lambda_c d) \leq f(x_k) + \eta_k - \gamma \lambda_c^2 \|d\|^2, \tag{2.1}$$

donde η_k es un término de una sucesión de números positivos $\{\eta_k\}_{k \in \mathbb{N}}$ tal que

$$\sum_{k=0}^{\infty} \eta_k \leq \eta < \infty. \tag{2.2}$$

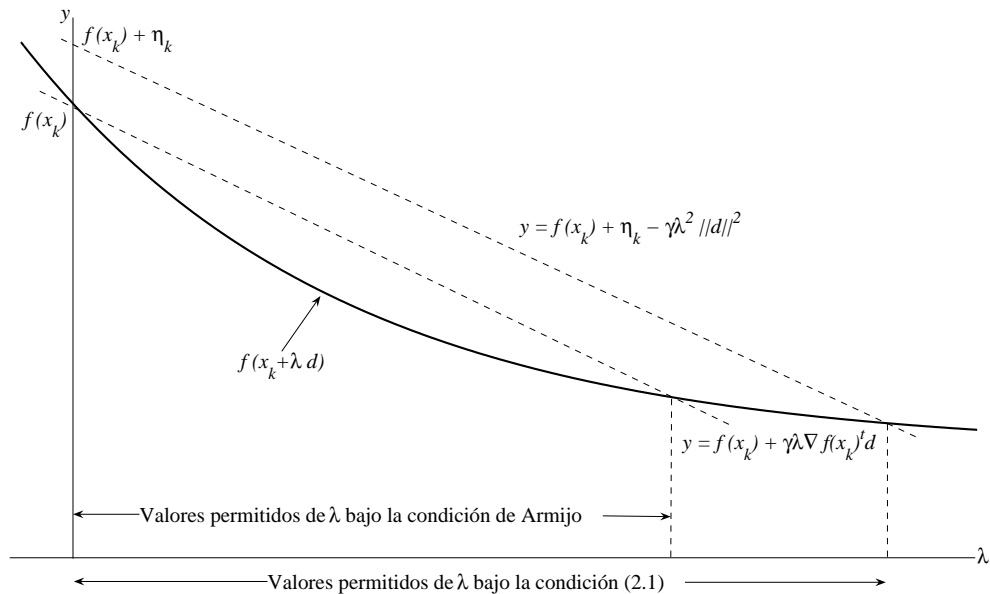


Figura 2.1: Valores permitidos de λ bajo las condiciones de Armijo y (2.1) ($\gamma = 0.1$)

Con esta búsqueda lineal tratamos que se acepte el tamaño de paso con tanta frecuencia como sea posible, cuando x_k se encuentra lejos de un punto estacionario. Si x_k está muy cerca de un punto estacionario, la condición (2.1) es muy parecida a la regla de Armijo y en este momento tiende a ser monótona. En efecto, bajo ciertas condiciones que satisfacen la dirección d y que se establecen en el Teorema 2.1, en la Figura 2.1 se muestran los valores permitidos de λ bajo las condiciones de Armijo y (2.1). En esta figura observamos que la longitud del intervalo de valores permitidos por la condición (2.1) es mayor a la del intervalo permitido por la condición de Armijo. Cuando $k \rightarrow \infty$, la sucesión $\{\eta\}_{k \in \mathbb{N}}$ tiende a cero; por lo tanto, la condición (2.1) tiende a ser similar a la de Armijo cuando k es relativamente grande.

El nuevo criterio de aceptación de la longitud del paso permite el crecimiento de f pero garantiza la convergencia global a un mínimo local. El Teorema 2.1 establece que la condición (2.1) garantiza que $\lim_{k \rightarrow \infty} \|\nabla f(x_k)\| = 0$ y, además, presenta las propiedades que deben satisfacer las direcciones de descenso para garantizar la convergencia global.

Teorema 2.1. Sean $\gamma \in (0, 1)$, $0 < \sigma_{\min} < \sigma_{\max} < 1$, y las sucesiones de números positivos $\{a_k\}_{k \in \mathbb{N}}$ y $\{\eta_k\}_{k \in \mathbb{N}}$ tales que $\{a_k\}_{k \in \mathbb{N}}$ es acotada y $\{\eta_k\}_{k \in \mathbb{N}}$ satisface (2.2). Sea $\{x_k\}_{k \in \mathbb{N}}$ una sucesión definida como

$$x_{k+1} = x_k + \lambda_k d_k, \quad d_k \neq 0.$$

Supongamos que:

(i) el conjunto $\Omega_0 = \{x : f(x) \leq f(x_0) + \eta\}$ es compacto;

(ii) existen constantes positivas c_1 y c_2 tales que

$$\nabla f(x_k)^t d_k \leq -c_1 \|\nabla f(x_k)\|^2, \quad (2.3)$$

$$\|\nabla f(x_k)\| \leq c_2 \|d_k\|, \quad (2.4)$$

(iii) existen un entero positivo i_k y un número $\rho_j \in [\sigma_{\min}, \sigma_{\max}]$, $1 \leq j \leq i_k$, tales que $\sigma_k = \prod_{j=1}^{i_k} \rho_j$ y $\lambda_k = \sigma_k a_k$, además, $x_{k+1} = x_k + \lambda_k d_k$ satisface la condición (2.1) para $k \geq 1$.

Entonces:

- (a) la sucesión $\{x_k\}_{k \in \mathbb{N}}$ está contenida en Ω_0 ;
- (b) $\lim_{k \rightarrow \infty} \|x_{k+1} - x_k\| = 0$;
- (c) todos los puntos límites \bar{x} de $\{x_k\}_{k \in \mathbb{N}}$ satisfacen que $\nabla f(\bar{x}) = 0$;
- (d) si existe un índice \bar{k} tal que:

$$\eta_k < \frac{\gamma}{2} \lambda_k^2 \|d_k\|^2, \text{ para } k \geq \bar{k}, \quad (2.5)$$

entonces la sucesión $\{f(x_k)\}_{k \in \mathbb{N}}$ converge y ningún punto de acumulación de $\{x_k\}_{k \in \mathbb{N}}$ es un máximo local de f en Ω_0 .

Demostración. Prueba de (a). Por (2.1) podemos escribir para $k \geq 0$:

$$\begin{aligned} f(x_{k+1}) &\leq f(x_k) + \eta_k \\ f(x_{k+2}) &\leq f(x_{k+1}) + \eta_{k+1} \leq f(x_k) + \eta_k + \eta_{k+1} \\ f(x_{k+3}) &\leq f(x_{k+2}) + \eta_{k+2} \leq f(x_k) + \eta_k + \eta_{k+1} + \eta_{k+2} \\ &\vdots \end{aligned}$$

Siguiendo con este proceso inductivo obtenemos:

$$f(x_{k+j}) \leq f(x_k) + \sum_{i=k}^{k+j-1} \eta_i, \text{ para } k \geq 0, j \geq 1. \quad (2.6)$$

Tomando $k = 0$ y $j \geq 1$ en la ecuación (2.6) obtenemos:

$$f(x_j) \leq f(x_0) + \sum_{i=0}^{j-1} \eta_i, \text{ para } j \geq 1. \quad (2.7)$$

Por (2.2) y (2.7) se deduce que $f(x_k) \leq f(x_0) + \eta$, para $k \in \mathbb{N}$. En otras palabras, la sucesión $\{x_k\}_{k \in \mathbb{N}}$ está contenida en Ω_0 .

Prueba de (b). Como $x_{k+1} = x_k + \lambda_k d_k$, entonces empleando (2.1) podemos escribir:

$$\|x_{k+1} - x_k\|^2 = \lambda_k^2 \|d_k\|^2 \leq \frac{\eta_k}{\gamma} + \frac{1}{\gamma} (f(x_k) - f(x_{k+1})).$$

Sumando y luego utilizando (2.2) obtenemos:

$$\begin{aligned} \sum_{k=0}^{\infty} \|x_{k+1} - x_k\|^2 &\leq \frac{1}{\gamma} \left[\sum_{k=0}^{\infty} \eta_k + \sum_{k=0}^{\infty} (f(x_k) - f(x_{k+1})) \right] \\ &\leq \frac{1}{\gamma} \left[\eta + \left| \sum_{k=0}^{\infty} (f(x_k) - f(x_{k+1})) \right| \right] \\ &= \frac{\eta + |f(x_0)|}{\gamma} < \infty, \end{aligned}$$

de donde se infiere que $\lim_{k \rightarrow \infty} \|x_{k+1} - x_k\| = 0$.

Prueba de (c). Como $\|x_{k+1} - x_k\| = \lambda_k \|d_k\|$, la aserción (b) implica que

$$\lim_{k \rightarrow \infty} \lambda_k \|d_k\| = 0. \quad (2.8)$$

Como $0 < \lambda_k \leq a_k$ para $k \in \mathbb{N}$ y $\{a_k\}_{k \in \mathbb{N}}$ es acotada, entonces por (2.4) y (2.8) obtenemos:

$$\lim_{k \rightarrow \infty} \lambda_k \|\nabla f(x_k)\| = 0. \quad (2.9)$$

Sea \bar{x} un punto de acumulación de $\{x_k\}_{k \in \mathbb{N}}$. Existe una subsucesión $\{x_k\}_{k \in M} \subset \{x_k\}_{k \in \mathbb{N}}$ tal que converge a \bar{x} . Entonces, por (2.9)

$$\lim_{k \rightarrow \infty, k \in M} \lambda_k \|\nabla f(x_k)\| = 0,$$

que es cierto si $\nabla f(\bar{x}) = 0$, con lo cual se demuestra (c). La ecuación anterior también es cierta si existe un conjunto infinito de índices $K = \{k_1, k_2, k_3, \dots\} \subseteq M$ tal que

$$\lim_{j \rightarrow \infty} \lambda_{k_j} = 0.$$

En este caso, por la suposición (iii), existe un índice \bar{j} suficientemente grande tal que para todo $j \geq \bar{j}$, existe $\sigma_{k_j} \in (0, \sigma_{max}]$ para el que $\lambda_{k_j} = \sigma_{k_j} a_{k_j}$ no satisface la condición (2.1), es decir,

$$\begin{aligned} f(x_{k_j} + (\sigma_{k_j} a_{k_j}) d_{k_j}) &> f(x_{k_j}) + \eta_{k_j} - \gamma (\sigma_{k_j}^2 a_{k_j}^2) \|d_{k_j}\|^2 \\ &\geq f(x_{k_j}) - \gamma (\sigma_{k_j}^2 a_{k_j}^2) \|d_{k_j}\|^2. \end{aligned}$$

Por lo tanto,

$$\frac{f(x_{k_j} + (\sigma_{k_j} a_{k_j}) d_{k_j}) - f(x_{k_j})}{(\sigma_{k_j} a_{k_j})} > -\gamma (\sigma_{k_j} a_{k_j}) \|d_{k_j}\|^2. \quad (2.10)$$

Como $\{a_k\}_{k \in \mathbb{N}}$ es acotada y $\lim_{j \rightarrow \infty} \lambda_{k_j} = 0$, entonces $\lim_{j \rightarrow \infty} \sigma_{k_j} = 0$. De esta forma, Mediante el Teorema del Valor Medio la relación (2.10) se puede expresar como

$$\nabla f(x_{k_j} + t_{k_j} d_{k_j})^t d_{k_j} > -\gamma(\sigma_{k_j} a_{k_j}) \|d_{k_j}\|^2, \text{ para } j \geq \bar{j}, \quad (2.11)$$

donde $t_{k_j} \in [0, \sigma_{k_j} a_{k_j}]$ tiende a cero cuando $j \rightarrow \infty$. Sin pérdida de generalidad, existe un conjunto infinito de índices $L = \{l_1, l_2, l_3, \dots\} \subset K$ tal que

$$\lim_{j \rightarrow \infty} x_{l_j} = \bar{x}, \quad \lim_{j \rightarrow \infty} d_{l_j} = \bar{d}.$$

Observando que $(x_{l_j} + t_{l_j} d_{l_j}) \rightarrow \bar{x}$ cuando $j \rightarrow \infty$, entonces tomando límite en (2.11) obtenemos:

$$\nabla f(\bar{x})^t \bar{d} \geq 0.$$

Como $\nabla f(\bar{x})^t \bar{d} < 0$, entonces $\nabla f(\bar{x})^t \bar{d} = 0$, lo cual implica, por (2.3), $\nabla f(\bar{x}) = 0$, y esto completa la prueba de (c).

Prueba de (d). Por (2.1) y (2.5) podemos escribir:

$$\begin{aligned} f(x_{k+1}) &\leq f(x_k) + \eta_k - \gamma \lambda_k^2 \|d_k\|^2 \\ &< f(x_k) + \frac{\gamma}{2} \lambda_k^2 \|d_k\|^2 - \gamma \lambda_k^2 \|d_k\|^2 \\ &= f(x_k) - \frac{\gamma}{2} \lambda_k^2 \|d_k\|^2 \\ &< f(x_k), \text{ para } k \geq \bar{k}. \end{aligned} \quad (2.12)$$

En consecuencia, la subsucesión $\{f(x_k)\}_{k \geq \bar{k}} \subset \{f(x_k)\}_{k \in \mathbb{N}}$ es no creciente. De esta forma, $\{f(x_k)\}_{k \geq \bar{k}}$ admite un límite para $k \rightarrow \infty$. Se sigue de la continuidad uniforme de f en Ω_0 que la sucesión $\{f(x_k)\}_{k \in \mathbb{N}}$ converge.

Por otra parte, supongamos que existe un punto de acumulación \hat{x} de $\{x_k\}_{k \in \mathbb{N}}$ que es un máximo local de f en Ω_0 . Entonces, existe una subsucesión $\{x_k\}_{k \in K} \subset \{x_k\}_{k \in \mathbb{N}}$ que converge a \hat{x} . Por (2.12) para $k \geq \bar{k}$, $k \in K$ suficientemente grande podemos encontrar un punto x_k en alguna vecindad de \hat{x} que satisface $f(x_k) > f(\hat{x})$. Esto contradice la suposición de que \hat{x} es un máximo local de f en Ω_0 . Con esto se completa la prueba de (d). \square

El Teorema 2.1 presenta un metodología para la generación de soluciones aproximadas del problema (1.3). En particular, la condición (iii) describe una búsqueda lineal donde la sucesión $\{a_k\}_{k \in \mathbb{N}}$ representa la longitud de paso inicial en cada iteración k .

La convergencia del esquema propuesto en el Teorema 2.1 depende esencialmente de las condiciones impuestas al tamaño de paso λ_k y a la dirección d_k . El tamaño de paso λ_k debe satisfacer la ecuación (2.1). La condición (iii) garantiza que d_k es una dirección de descenso que está uniformemente relacionada con el gradiente de f en x_k . Muchos métodos locales generan direcciones que satisfacen la condición (iii), por ejemplo, los métodos tipo Newton y tipo gradiente.

2.2 Métodos Tipo Newton y Tipo Gradiente Globales

A continuación mostramos dos algoritmos donde se emplea la búsqueda lineal no monótona definida en la Sección 2.1. El primer algoritmo es una versión modificada del método de Newton y el segundo es una versión globalizada de BB.

2.2.1 Algoritmo de Newton Global

A las direcciones tipo Newton se les puede exigir que satisfagan la condición (ii) del Teorema 2.1. En efecto, una dirección tipo Newton se define como:

$$d_k = -B_k^{-1} \nabla f(x_k).$$

Si asumimos que B_k es una matriz positiva definida y, además, que existen constantes positivas Φ y T , tales que para todo k : $\|B_k\| \leq T$ y los autovalores $\phi_i(B_k)$ satisfacen $0 < \phi_i(B_k) \leq \Phi$. Entonces,

$$\nabla f(x_k)^t d_k \leq -\frac{1}{\Phi} \|\nabla f(x_k)\|^2$$

y

$$\|\nabla f(x_k)\| \leq T \|d_k\|.$$

En otras palabras, las direcciones tipo Newton satisfacen la condición (ii) del Teorema 2.1.

El Algoritmo 2.1 es la versión modificada del método de Newton y posee una estructura muy similar a la del Algoritmo Modelo en [40, p. 712], con algunos cambios necesarios para que la dirección de Newton y el tamaño de paso satisfagan las condiciones (ii) y (iii) del Teorema 2.1.

Algoritmo 2.1 Algoritmo de Newton Global

Datos: $x_0 \in \mathbb{R}^n$, $\gamma \in (0, 1)$, $0 < \sigma_{min} < \sigma_{max} < 1$, $c_1 > 0$, $c_2 > 0$, $\{\eta_k\}_{k \in \mathbb{N}}$ que satisface (2.2) y $k \leftarrow 0$.

```

1: mientras  $\nabla f(x_k) \neq 0$  hacer
2:   si  $H_k$  es singular entonces
3:      $d_k \leftarrow -\nabla f(x_k)$ 
4:   de lo contrario
5:      $d_k \leftarrow -H_k^{-1} \nabla f(x_k)$ 
6:   si  $|\nabla f(x_k)^t d_k| < c_1 \|\nabla f(x_k)\|^2$  o  $\|\nabla f(x_k)\| > c_2 \|d_k\|$  entonces
7:      $d_k \leftarrow -\nabla f(x_k)$ 
8:   de lo contrario
9:     si  $\nabla f(x_k)^t d_k > 0$  entonces
10:       $d_k \leftarrow -d_k$ 
11:    fin si
12:  fin si
13:  fin si
14:   $\lambda \leftarrow 1$ 
15:  mientras  $f(x_k + \lambda d_k) > f(x_k) + \eta_k - \gamma \lambda^2 \|d_k\|^2$  hacer
16:    escoger  $\sigma \in [\sigma_{min}, \sigma_{max}]$ 
17:     $\lambda \leftarrow \sigma \lambda$ 
18:  fin mientras
19:   $\lambda_k \leftarrow \lambda$ 
20:   $x_{k+1} \leftarrow x_k + \lambda_k d_k$ 
21:   $k \leftarrow k + 1$ 
22: fin mientras

```

Comentarios.

- (a) El algoritmo no puede realizar infinitas reducciones de λ . En efecto, por la continuidad de f y dado que $\eta_k > 0$, entonces con un número finito de reducciones de λ se cumple la condición $f(x_k + \lambda d_k) \leq f(x_k) + \eta_k - \gamma \lambda^2 \|d_k\|^2$.
- (b) En el algoritmo $a_k = 1$ para todo $k \in \mathbb{N}$.
- (c) Las líneas 16 y 17 del algoritmo consisten en un proceso de backtraking donde se generan el entero i_k y los números $\rho_j \in [\sigma_{min}, \sigma_{max}]$, $1 \leq j \leq i_k$, tales que $\sigma_k = \prod_{j=1}^{i_k} \rho_j$ y $\lambda_k = \sigma_k a_k$.
- (d) La sucesión $\{x_k\}_{k \in \mathbb{N}}$ generada por el Algoritmo 2.1 está contenida en el conjunto Ω_0 .

En el siguiente teorema establecemos la convergencia global del Algoritmo 2.1.

Teorema 2.2. *Supongamos que Ω_0 es compacto. Sea $f : \mathbb{R}^n \rightarrow \mathbb{R}$ dos veces continuamente diferenciable en algún abierto que contiene a Ω_0 . Sea $\{x_k\}_{k \in \mathbb{N}}$ la sucesión generada por el Algoritmo 2.1. Entonces,*

$$\lim_{k \rightarrow \infty} \|\nabla f(x_k)\| = 0.$$

Demostración. Las sucesiones $\{x_k\}_{k \in \mathbb{N}}$, $\{d_k\}_{k \in \mathbb{N}}$ y $\{\lambda_k\}_{k \in \mathbb{N}}$ satisfacen las condiciones (i), (ii) y (iii) del Teorema 2.1. Por lo tanto, de la asercción (c) del Teorema 2.1 se infiere que $\lim_{k \rightarrow \infty} \|\nabla f(x_k)\| = 0$. \square

2.2.2 Algoritmo de Barzilai-Borwein Global

Igual que las direcciones tipo Newton, las direcciones tipo gradiente también satisfacen las condiciones (ii) y (iii) del Teorema 2.1. En efecto, la dirección tipo gradiente definida por $d_k = -\nabla f(x_k)$ satisface:

$$\nabla f(x_k)^t d_k = \|\nabla f(x_k)\|^2 \quad \text{y} \quad \|d_k\| = \|\nabla f(x_k)\|.$$

A la dirección $d_k = -\nabla f(x_k)$ y al tamaño de paso $\lambda = 1/\alpha_k$ del método BB se les puede exigir que satisfagan las condiciones (ii) y (iii) del Teorema 2.1. Para tal efecto, modificamos

el algoritmo Global Barzilai and Borwein (GBB) dado en [62, p. 28], donde se combina BB con una variante de GLL. Nuestra modificación consiste en reemplazar GLL con la búsqueda lineal no monótona descrita en el Teorema 2.1. De esta forma, obtuvimos una nueva versión globalizada de BB que denominamos NGBB (New Global Barzilai-Borwein) y que se muestra en el Algoritmo 2.2.

Algoritmo 2.2 Algoritmo NGBB

Datos: $\alpha_0 > 0$, $x_0 \in \mathbb{R}^n$, $\gamma \in (0, 1)$, $0 < \sigma_{min} < \sigma_{max} < 1$, $0 < \varepsilon < 1$, $\delta \in [\varepsilon, 1/\varepsilon]$, $\{\eta_k\}_{k \in \mathbb{N}}$ que satisface (2.2) y $k \leftarrow 0$.

- 1: **mientras** $\nabla f(x_k) \neq 0$ **hacer**
- 2: **si** $\alpha_k \leq \varepsilon$ o $\alpha_k \geq 1/\varepsilon$ **entonces**
- 3: $\alpha_k \leftarrow \delta$
- 4: **fin si**
- 5: $d_k \leftarrow -\nabla f(x_k)$
- 6: $\lambda \leftarrow 1/\alpha_k$
- 7: **mientras** $f(x_k + \lambda d_k) > f(x_k) + \eta_k - \gamma \lambda^2 \|d_k\|^2$ **hacer**
- 8: escoger $\sigma \in [\sigma_{min}, \sigma_{max}]$
- 9: $\lambda \leftarrow \sigma \lambda$
- 10: **fin mientras**
- 11: $\lambda_k \leftarrow \lambda$
- 12: $x_{k+1} \leftarrow x_k + \lambda_k d_k$
- 13: $y_k \leftarrow \nabla f(x_{k+1}) - \nabla f(x_k)$
- 14: $\alpha_{k+1} \leftarrow -(\nabla f(x_k)^t y_k) / (\lambda_k \nabla f(x_k)^t \nabla f(x_k))$
- 15: $k \leftarrow k + 1$
- 16: **fin mientras**

Comentarios.

(a) El algoritmo NGBB no puede realizar infinitas reducciones de λ . Como $\eta_k > 0$, por continuidad, para λ suficientemente pequeño la condición

$$f(x_k + \lambda d_k) \leq f(x_k) + \eta_k - \gamma \lambda^2 \|d_k\|^2$$

se satisface.

(b) La sucesión $\{a_k\}_{k \in \mathbb{N}}$ está definida como $a_k = 1/\alpha_k$ y está uniformemente acotada. En efecto, como $1/\alpha_k \in (\varepsilon, 1/\varepsilon)$ y $\delta \in [\varepsilon, 1/\varepsilon]$, entonces

$$\varepsilon < a_k < 1/\varepsilon, \text{ para } k \in \mathbb{N}.$$

(c) La sucesión $\{x_k\}_{k \in \mathbb{N}}$ generada por el algoritmo NGBB está contenida en Ω_0 .

Las propiedades de convergencia del algoritmo NGBB se establecen en el siguiente teorema.

Teorema 2.3. *Supongamos que Ω_0 es compacto. Sea $f : \mathbb{R}^n \rightarrow \mathbb{R}$ continuamente diferenciable en algún abierto que contiene a Ω_0 . Sea $\{x_k\}_{k \in \mathbb{N}}$ la sucesión generada por el algoritmo NGBB. Entonces, $\nabla f(x_j) = 0$ para algún j finito, o*

$$\lim_{k \rightarrow \infty} \|\nabla f(x_k)\| = 0.$$

Demostración. En el Algoritmo NGBB $d_k = -\nabla f(x_k)$ para $k \in \mathbb{N}$ y $c_1 = c_2 = 1$. La sucesión $\{a_k\}_{k \in \mathbb{N}}$ definida como $a_k = 1/\alpha_k$ está acotada. Luego, las sucesiones $\{x_k\}_{k \in \mathbb{N}}$, $\{d_k\}_{k \in \mathbb{N}}$ y $\{\lambda_k\}_{k \in \mathbb{N}}$ satisfacen las condiciones (i), (ii) y (iii) del Teorema 2.1. Por lo tanto, de la aserción (c) del Teorema 2.1 se infiere que $\lim_{k \rightarrow \infty} \|\nabla f(x_k)\| = 0$. \square

2.3 Resultados Numéricos del Algoritmo de Newton Global

En esta sección reportamos los resultados numéricos obtenidos para el Algoritmo 2.1 sobre el siguiente conjunto de problemas test.

Problema 1. *Rosenbrock function* [57].

$$\begin{aligned} f(x) &= 100(x_2 - x_1^2)^2 + (1 - x_1)^2, \\ x_0 &= (-1.2, 1)^t, \quad x_\star = (1, 1)^t, \quad f(x_\star) = 0. \end{aligned}$$

Problema 2. *Wood function* [57].

$$\begin{aligned} f(x) &= 100(x_1^2 - x_2)^2 + (x_1 - 1)^2 + (x_3 - 1)^2 + 90(x_3^2 - x_4)^2 \\ &\quad + 10.1((x_2 - 1)^2 + (x_4 - 1)^2) + 19.8(x_2 - 1)(x_4 - 1), \\ x_0 &= (-3, -1, -3, -1)^t, \quad x_\star = (1, 1, 1, 1)^t, \quad f(x_\star) = 0. \end{aligned}$$

Problema 3. *Powell singular function* [57].

$$\begin{aligned} f(x) &= (x_1 + 10x_2)^2 + 5(x_3 - x_4)^2 + (x_2 - 2x_3)^4 + 10(x_1 - x_4)^4, \\ x_0 &= (3, -1, 0, 1)^t, \quad x_\star = (0, 0, 0, 0)^t, \quad f(x_\star) = 0. \end{aligned}$$

Problema 4. *Cube* [40].

$$\begin{aligned} f(x) &= 100(x_2 - x_1^3)^2 + (1 - x_1)^2, \\ x_0 &= (-1.2, -1)^t, \quad x_\star = (1, 1)^t, \quad f(x_\star) = 0. \end{aligned}$$

Problema 5. *Freudenstein and Roth function* [57].

$$\begin{aligned} f(x) &= (-13 + x_1 + ((5 - x_2)x_2 - 2)x_2)^2 \\ &\quad + (-29 + x_1 + ((x_2 + 1)x_2 - 14)x_2)^2, \\ x_0 &= (0.5, -2)^t, \quad x_\star = (5, 4)^t, \quad f(x_\star) = 0. \end{aligned}$$

Problema 6. *Box three-dimensional function* [57].

$$\begin{aligned} f(x) &= \sum_{j=1}^{10} (\exp(-t_j x_1) - \exp(-t_j x_2) - x_3 (\exp(-t_j) - \exp(-10t_j)))^2, \\ &\text{donde } t_j = (0.1)j. \\ x_0 &= (0, 10, 20)^t, \\ f(x_\star) &= 0 \text{ en } x_\star = (1, 10, 1)^t, (10, 1, -1)^t, \\ &\text{y donde } (x_1 = x_2 \text{ y } x_3 = 0). \end{aligned}$$

Problema 7. *Strictly convex 2* (modificada) [62].

$$\begin{aligned} f(x) &= \sum_{i=1}^n \frac{i}{10} (e^{x_i} - x_i) - c, \\ &\text{donde } c = \begin{cases} 1, & \text{si } n = 4, \\ 21, & \text{si } n = 20, \\ 183, & \text{si } n = 60, \end{cases} \\ x_0 &= (1, 1, \dots, 1)^t, \quad x_\star = (0, 0, \dots, 0)^t, \quad f(x_\star) = 0. \end{aligned}$$

Problema 8. *Penalty function I* [57].

$$\begin{aligned} f(x) &= a \sum_{j=1}^n (x_j - 1)^2 + \left(\sum_{j=1}^n x_j^2 - 0.25 \right)^2, \\ &\text{donde } a = 10^{-5}, \\ x_0 &= (1, 2, \dots, n)^t. \end{aligned}$$

Todos los experimentos de esta parte se corrieron en un Computador Pentium IV de 3.0 GHz en MATLAB 7.0. Implementamos el Algoritmo 2.1 con los parámetros: $\gamma = 10^{-4}$, $\sigma_{min} = 0.1$, $\sigma_{max} = 0.5$, $c_1 = 10^{-5}$, y $c_2 = 10^5$. Tomamos $\sigma = 0.5$ en cada iteración.

Estudiamos el comportamiento del Algoritmo 2.1 para diferentes sucesiones $\{\eta_k\}_{k \in \mathbb{N}}$. En la Tabla 2.1 se muestran los resultados numéricos para los Problemas 1-8. En esta tabla se observan los resultados del Algoritmo 2.1 con $\eta_k = 500(1 - 10^{-6})^k$ (NLS). También se observa en esta tabla los resultados para el Algoritmo Modelo en [40, p. 712] con la regla de Armijo (ALS) y los parámetros que allí se especifican. Reportamos número del problema (P), dimensión (n), número de iteraciones (IT), número de evaluaciones de f (EF), número de ocasiones donde la búsqueda lineal es usada (BL) y el valor $f(\hat{x})$ de la función objetivo en la solución encontrada \hat{x} , que es menor que la tolerancia exigida.

En las Tablas 2.2 y 2.3 mostramos respectivamente los resultados numéricos para los Problemas 2 y 6 para diferentes sucesiones $\{\eta_k\}_{k \in \mathbb{N}}$. Reportamos número de iteraciones, número de evaluaciones de f y número de ocasiones donde la búsqueda lineal es usada. En la Tabla 2.4 mostramos los resultados numéricos para el Problema 8 para diferentes sucesiones $\{\eta_k\}_{k \in \mathbb{N}}$. Reportamos número de iteraciones, número de evaluaciones de f , número de ocasiones donde la búsqueda lineal es usada y el valor $f(\hat{x})$ de la función objetivo en la solución encontrada \hat{x} .

Observamos que en casi todos los problemas considerados el número de iteraciones y el número de evaluaciones de f para $\eta_k = 500(1 - 10^{-6})^k$, son considerablemente menor que el requerido por ALS. Para el Problema 1 el ahorro es del orden del 66%. En el Problema 2 el ahorro es de 41%. En el Problema 4 el ahorro es de 69%. En el Problema 6 el ahorro es de 1.4%. En el Problema 8 el ahorro es de 99.8% para $n = 4$, 65% para $n = 10$ y 99.6% para $n = 50$.

Los resultados de la Tabla 2.1 pueden variar drásticamente si se considera otra sucesión $\{\eta_k\}_{k \in \mathbb{N}}$, lo cual se aprecia en las Tablas 2.2, 2.3 y 2.4. Entonces, es evidente que la manera de definir la sucesión $\{\eta_k\}_{k \in \mathbb{N}}$ influye considerablemente en la eficiencia del Algoritmo 2.1. También observamos que la búsqueda lineal se activa al principio o en el intermedio de la ejecución del Algoritmo 2.1.

Tabla 2.1: Resultados para los Problemas 1-8.

P	n	NLS				ALS			
		IT	EF	BL	$f(\hat{x})$	IT	EF	BL	$f(\hat{x})$
1	2	8	10	1	$< 10^{-38}$	22	30	4	$< 10^{-38}$
2	4	28	34	3	$< 10^{-38}$	40	70	8	$< 10^{-38}$
3	4	43	44	0	$< 10^{-28}$	43	44	0	$< 10^{-28}$
4	2	9	12	1	$< 10^{-38}$	29	41	6	$< 10^{-38}$
5	2	11	12	0	$< 10^{-38}$	11	12	0	$< 10^{-38}$
6	3	266	267	0	$< 10^{-38}$	268	273	4	$< 10^{-38}$
7	4	5	6	0	$< 10^{-38}$	5	6	0	$< 10^{-38}$
	20	5	6	0	$< 10^{-38}$	5	6	0	$< 10^{-38}$
	60	5	6	0	$< 10^{-38}$	5	6	0	$< 10^{-38}$
8	4	15	17	1	$3.4 \cdot 10^{-5}$	4932	13762	3541	$3.4 \cdot 10^{-5}$
	10	862	863	0	$7.1 \cdot 10^{-5}$	1912	3455	1221	$7.1 \cdot 10^{-5}$
	50	21	39	1	$5.6 \cdot 10^{-4}$	34	106	17	$5.6 \cdot 10^{-4}$

Tabla 2.2: Resultados para el Problema 2

	ALS	η_k				
		$2^{-k}500$	$3^{-k}500$	$4^{-k}500$	$(1/1.1)^k500$	$(0.99)^k500$
IT	40	39	39	39	30	28
EF	70	67	53	53	40	34
BL	8	8	6	6	4	3

Tabla 2.3: Resultados para el Problema 6

	ALS	η_k				
		$2^{-k}500$	$3^{-k}500$	$4^{-k}500$	$(1/1.1)^k500$	$(0.99)^k500$
IT	268	268	268	268	266	266
EF	273	273	273	273	267	267
BL	4	4	4	4	0	0

Tabla 2.4: Resultados para el Problema 8

n		ALS	η_k				
			$2^{-k}500$	$3^{-k}500$	$4^{-k}500$	$(1/1.1)^k 500$	$(0.99)^k 500$
4	IT	4932	10000	4937	4932	15	15
	EF	13762	153368	13739	13762	17	17
	BL	3541	9770	3532	3541	1	1
	$f(\hat{x})$	$3.4 \cdot 10^{-4}$	$5.2 \cdot 10^{-4}$	$3.4 \cdot 10^{-5}$	$3.4 \cdot 10^{-5}$	$3.4 \cdot 10^{-5}$	$3.4 \cdot 10^{-5}$
10	IT	1912	1863	1911	1922	1531	862
	EF	3455	3325	3445	3473	2510	863
	BL	1221	1164	1220	1233	779	0
	$f(\hat{x})$	$7.1 \cdot 10^{-5}$	$7.1 \cdot 10^{-5}$	$7.1 \cdot 10^{-5}$	$7.1 \cdot 10^{-5}$	$7.1 \cdot 10^{-5}$	$7.1 \cdot 10^{-5}$
50	IT	34	21	26	49	21	21
	EF	116	39	58	312	39	39
	BL	17	1	7	29	1	1
	$f(\hat{x})$	$5.6 \cdot 10^{-4}$	$5.6 \cdot 10^{-4}$	$5.6 \cdot 10^{-4}$	$5.6 \cdot 10^{-4}$	$5.6 \cdot 10^{-4}$	$5.6 \cdot 10^{-4}$

2.4 Resultados Numéricos del Algoritmo NGBB

En esta sección reportamos los resultados numéricos obtenidos para el Algoritmo 2.2 sobre un extenso conjunto de problemas test y el problema de Conformación Molecular.

2.4.1 Problemas Test

La lista de problemas test que usamos para comparar numéricamente a NGBB con GBB se muestra en la Tabla 2.5. En esta tabla se observa el número del problema (P), nombre y referencia (NR) e iterado inicial (x_0). En este trabajo describimos dos nuevas funciones:

Exponencial 1:

$$f(x) = e^{x_1-1} - 10^6 x_1 + \sum_{i=2}^n i \left(e^{x_i-1} - \frac{x_i^2}{2} \right).$$

Casi-cuadrática:

$$f(x) = \frac{1}{2} x^t A x - b^t x + \frac{1}{4} h^2 \sum_{i=1}^n (x_i - 1)^4,$$

donde $A = \text{diag}(h, h+1, \dots, n-1+h)$, $b = A(1, 1, \dots, 1)^t$, y $h = 1/(n+1)$.

El único minimizador de Exponencial 1 es $x_* = (6 \ln(10) + 1, 1, \dots, 1)^t$. La Hessiana de Exponencial 1 en x_* es $\text{diag}(10^6, 0, \dots, 0)$. El único minimizador de Casi-cuadrática es $x_* = (1, 1, \dots, 1)^t$. La Hessiana de Casi-cuadrática en x_* es A , con $\kappa(A) = n^2$.

Tabla 2.5: Problemas Test

P	NR	x_0
1	Strictly Convex 1 [62]	$(\frac{1}{n}, \dots, \frac{i}{n}, \dots, 1)^t$
2	Strictly Convex 2 [62]	$(1, 1, \dots, 1)^t$
3	Extended Rosenbrock [57]	$(-1.2, 1, -1.2, 1, \dots)^t$
4	Extended Powell singular [57]	$(3, -1, 0, 1, 3, -1, 0, 1, \dots)^t$
5	Penalty 1 [57]	$(1, 2, \dots, n)^t$
6	Variably dimensioned [57]	$(0, 1 - \frac{1}{2}, \dots, 1 - \frac{1}{n})^t$
7	Extended Freudenstein and Roth [9]	$(90, 60, 90, 60, \dots)^t$
8	Extended Cragg and Levy [57]	$(1, 2, 2, 2, 1, 2, 2, 2, \dots)^t$
9	Extended Wood [43]	$(-3, -1, -3, -1, \dots)^t$
10	Brent [1]	$(0, 0, \dots, 0)^t$
11	Troesch [64]	$(0, 0, \dots, 0)^t$
12	Trigonometric [67]	$(\frac{1}{5n}, \frac{1}{5n}, \dots, \frac{1}{5n})^t$
13	Chained Rosenbrock [21]	$(1.2, 1, 1.2, 1, \dots)^t$
14	Generalized Broyden tridiagonal [57]	$(-1, -1, \dots, -1)^t$
15	Penalty 1 [36]	$(\frac{1}{n+1}, \frac{1}{n+1}, \dots, \frac{1}{n+1})^t$
16	Generalization of the Brown 1 [21]	$(3, 3, \dots, 3)^t$
17	Banded trigonometric [53]	$(\frac{1}{n}, \frac{1}{n}, \dots, \frac{1}{n})^t$
18	Tridiagonal system [50]	$(0, 0, \dots, 0)^t$
19	Problem 76 [53]	$(2, 2, \dots, 2)^t$
20	Problem 77 [53]	$(0, 0, \dots, 0)^t$
21	Problem 78 [53]	$(0, 0, \dots, 0)^t$
22	Problem 79 [53]	$(-1, -1, \dots, -1)^t$
23	Problem 81 [53]	$(0.5, 0.5, \dots, 0.5)^t$
24	Problem 82 [53]	$(0.5, 0.5, \dots, 0.5)^t$
25	Problem 87 [53]	$(1, 1, \dots, 1)^t$
26	Singular Broyden [53]	$(-1, -1, \dots, -1)^t$
27	Structured Jacobian [53]	$(-1, -1, \dots, -1)^t$
28	Discrete boundary value problem [53]	$(\frac{1-(n+1)^2}{(n+1)^2}, \frac{2(2-(n+1)^2)}{(n+1)^2}, \dots)^t$
29	Problema 7 [53]	$(-1, -1, \dots, -1)^t$
30	Chained serpentine [53]	$(-0.8, -0.8, \dots, -0.8)^t$
31	Attracting-Repelling [53]	$(1, 1, \dots, 1)^t$
32	Wright and Holt zero residual [53]	$(\sin(1)^2, \sin(2)^2, \dots, \sin(n)^2)^t$
33	Toint quadratic merging [53]	$(5, 5, \dots, 5)^t$
34	Chained exponential problem [53]	$(0.2, 0.2, \dots, 0.2)^t$
35	Countercurrent reactors 1 [53]	$(0.5, 0.5, \dots, 0.5)^t$
36	Trigonometric - exponential system [53]	$(0, 0, \dots, 0)^t$
37	Five-diagonal system [53]	$(-2, -2, \dots, -2)^t$
38	Exponencial 1	$(15, \frac{n}{n-1}, \dots, \frac{n}{n-1})^t$
39	Casi-cuadrática	$(0, 0, \dots, 0)^t$

Todos los experimentos de esta parte se corrieron en un Computador Pentium IV de 3.0 GHz en doble precisión FORTRAN. Implementamos a GBB con los parámetros: $\gamma = 10^{-4}$, $\varepsilon = 10^{-10}$, $\sigma_1 = 0.1$, $\sigma_2 = 0.5$, $\alpha_0 = 1$ y $M = 10$. El parámetro δ lo tomamos como en Raydan [62, p. 30]:

$$\delta = \begin{cases} 1, & \text{si } \|\nabla f(x_k)\| > 1, \\ \|\nabla f(x_k)\|, & \text{si } 10^{-5} \leq \|\nabla f(x_k)\| \leq 1, \\ 10^5 & \text{si } \|\nabla f(x_k)\| \leq 10^{-5}. \end{cases} \quad (2.13)$$

Implementamos a NGBB con los parámetros: $\gamma = 10^{-4}$, $\varepsilon = 10^{-10}$, $\sigma_{min} = 0.1$, $\sigma_{max} = 0.5$, $\alpha_0 = 1$, y $\eta_k = \theta(1 - 10^{-10})^k$, donde

$$\theta = \begin{cases} |f(x_0)|, & |f(x_0)| \leq 10^5, \\ 10^6, & |f(x_0)| > 10^5. \end{cases}$$

El parámetro δ lo tomamos como en (2.13).

Tanto para GBB y NGBB, el parámetro σ se escoge a través de la interpolación cuadrática descrita en [26, pp. 126–127]. Todas las corridas pararon cuando

$$\|\nabla f(x_k)\|_2 \leq 10^{-6}(1 + |f(x_k)|).$$

Los resultados numéricos se muestran en la Tabla 2.6. Reportamos el número y la dimensión del problema ($P(n)$), número de iteraciones (IT), número de evaluaciones de la función (f), número de ocasiones donde la búsqueda lineal es usada (BL), mejor valor de f conseguido (f_{best}) y tiempo de CPU en segundos (T). Los métodos GBB y NGBB realizan una evaluación del gradiente por iteración. Por lo tanto, el número de iteraciones representa el número de evaluaciones del gradiente para ambos métodos.

Los resultados numéricos de la Tabla 2.6 se resumen en la Tabla 2.7. En esta tabla reportamos el número de problemas donde un algoritmo es ganador en: número de iteraciones, número de evaluaciones de la función, número de ocasiones donde la búsqueda lineal es usada, mejor valor de f conseguido y tiempo de CPU.

Los resultados de la Tabla 2.1 pueden variar drásticamente si se considera otra sucesión $\{\eta_k\}_{k \in \mathbb{N}}$. También observamos que la búsqueda lineal generalmente se activa al principio o en el intermedio de la ejecución del algoritmo.

Tabla 2.6: Resultados numéricos para GBB y NGBB

P(n)	GBB					NGBB				
	IT	f	LS	fbest	T	IT	f	LS	fbest	T
1(1000)	7	7	0	1.000D+03	0.0000	7	7	0	1.000D+03	0.0000
1(50000)	6	6	0	5.000D+04	0.1250	6	6	0	5.000D+04	0.1250
2(1000)	92	137	13	5.005D+04	0.0469	72	79	2	5.005D+04	0.0313
2(10000)	59	83	7	5.001D+06	0.2969	64	74	2	5.000D+06	0.2813
3(1000)	18	33	5	1.233D-25	0.0000	18	33	5	1.233D-25	0.0000
3(10000)	18	33	5	1.333D-22	0.0313	18	33	5	1.333D-22	0.0156
4(1000)	14	21	1	8.212D+03	0.0156	14	21	1	8.212D+03	0.0000
4(10000)	13	20	1	8.248D+04	0.0313	15	21	1	8.144D+04	0.0469
5(1000)	102	149	10	1.556D-02	0.0000	102	193	10	2.689D-03	0.0156
5(10000)	102	3196	89	1.479D+10	0.7656	32	165	5	2.500D-02	0.0625
6(500)	102	5524	101	6.580D+11	0.1094	102	3639	101	6.156D+02	0.0625
6(1000)	102	6329	101	2.871D+15	0.2500	102	4108	101	7.432D+02	0.1719
7(1000)	13	38	1	4.201D+07	0.0000	13	38	1	4.201D+07	0.0000
7(10000)	12	37	1	4.203D+08	0.0313	12	37	1	4.203D+08	0.0313
8(1000)	47	61	7	2.711D-08	0.0313	75	79	3	1.181D-08	0.0469
8(10000)	55	69	7	1.530D-08	0.3906	77	81	3	1.208D-08	0.5000
9(1000)	13	32	1	6.308D+06	0.0156	13	32	1	6.308D+06	0.0000
9(10000)	12	31	1	6.308D+07	0.0313	12	31	1	6.308D+07	0.0313
10(1000)	102	201	21	1.022D-07	0.0156	48	59	1	1.979D-10	0.0000
10(10000)	102	201	21	1.022D-07	0.1875	48	59	1	1.979D-10	0.0781
11(100)	102	134	13	4.473D-04	0.0156	102	106	3	1.112D-03	0.0000
11(500)	102	143	18	1.900D-03	0.0313	102	108	5	1.191D-03	0.0156
12(10000)	6	6	0	4.504D-18	0.0625	6	6	0	4.504D-18	0.0625
12(50000)	5	5	0	4.868D-14	0.2813	5	5	0	4.868D-14	0.2656
13(1000)	102	122	5	2.818D-04	0.0156	102	112	1	5.599D-04	0.0156
13(5000)	102	127	9	4.787D-04	0.0625	102	112	1	2.444D-04	0.0469
14(1000)	102	134	13	1.120D-04	0.0625	102	109	2	1.569D-04	0.0469
14(5000)	102	133	15	1.530D-04	0.2813	102	108	2	8.731D-05	0.2500
15(1000)	8	9	1	2.891D+02	0.0000	8	9	1	2.891D+02	0.0000
15(10000)	10	14	1	5.671D+03	0.0156	10	13	1	5.671D+03	0.0000
16(10000)	30	51	6	9.989D+02	0.1250	26	44	4	9.989D+02	0.1094
16(50000)	34	66	8	4.995D+03	0.7344	30	53	7	4.995D+03	0.7031
17(1000)	102	143	15	-4.273D+02	0.0938	102	134	12	-4.273D+02	0.0781
17(10000)	102	170	24	-4.157D+03	0.9063	102	155	19	-4.159D+03	0.8906
18(100)	102	149	16	5.616D-01	0.0000	102	113	5	6.811D-01	0.0000
18(500)	102	144	17	4.779D-01	0.0000	102	114	5	6.206D-01	0.0156
19(1000)	8	8	0	3.139D+00	0.0000	8	8	0	3.139D+00	0.0000
19(10000)	8	8	0	3.139D+00	0.0000	8	8	0	3.139D+00	0.0156
20(1000)	7	7	0	4.160D+01	0.1250	2	2	0	4.160D+01	0.0313
20(5000)	2	2	0	2.083D+02	0.8438	2	2	0	2.083D+02	0.8594
21(100)	102	153	17	4.580D-07	0.0000	102	112	5	4.567D-07	0.0156
21(1000)	4	6	1	4.976D-10	0.0000	5	5	0	4.974D-10	0.0000
22(1000)	102	135	14	3.711D-06	0.0156	102	118	8	1.288D-02	0.0156
22(10000)	102	144	19	3.867D-02	0.1250	102	114	7	3.886D+00	0.1406
23(1000)	25	28	1	6.955D-14	0.0000	27	29	1	6.162D-14	0.0156
23(10000)	24	27	1	6.277D-14	0.0781	26	28	1	1.842D-13	0.0781
24(10000)	9	11	1	8.021D-14	0.0313	9	11	1	8.021D-14	0.0469
24(50000)	9	11	1	4.680D-13	0.1719	9	11	1	4.680D-13	0.1875
25(100)	102	136	12	1.983D-03	0.0000	102	107	4	2.057D-03	0.0000
25(500)	102	133	15	3.050D-03	0.0000	102	106	3	2.717D-03	0.0156
26(500)	83	97	6	1.080D-09	0.0156	92	101	2	3.485D-09	0.0000
26(1000)	83	97	6	4.343D-09	0.0313	102	113	4	9.694D-02	0.0313
27(500)	102	186	21	6.269D-01	0.0156	102	132	10	2.048D+00	0.0000
27(1000)	102	193	19	1.192D+00	0.0313	102	133	8	2.099D+00	0.0156
28(100)	102	136	12	1.281D+01	0.0000	102	109	4	8.221D+00	0.0000
28(250)	102	142	21	4.168D+02	0.0156	102	104	2	2.155D+02	0.0000
29(1000)	102	119	12	1.204D+03	0.1406	102	109	2	1.211D+03	0.1250
29(5000)	102	117	10	7.200D+03	0.7188	102	109	2	7.203D+03	0.6719
30(100)	102	154	17	1.827D+02	0.0000	102	113	2	1.935D+02	0.0000
30(500)	102	146	17	9.751D+02	0.0156	102	109	2	9.836D+02	0.0156
31(1000)	102	142	16	2.244D+03	0.0625	102	110	1	2.244D+03	0.0625
31(5000)	102	144	14	1.124D+04	0.2969	102	110	1	1.124D+04	0.2813
32(100)	102	151	18	1.694D-03	0.0469	102	111	5	8.389D-03	0.0469
32(500)	102	142	15	2.003D-03	0.2188	102	111	3	1.143D-03	0.2188
33(1000)	45	62	2	2.218D+03	0.0313	54	70	2	2.214D+03	0.0313
33(10000)	49	70	3	2.221D+04	0.3438	53	73	2	2.221D+04	0.3750
34(1000)	102	135	11	1.915D+02	0.3594	102	110	1	1.915D+02	0.3281
34(10000)	86	110	9	1.913D+03	2.9531	84	92	1	1.913D+03	2.7813
35(5000)	102	125	10	1.980D-02	0.0781	102	118	6	2.821D-01	0.0938
35(15000)	102	134	12	2.019D-02	0.2344	102	116	5	2.040D-02	0.2500
36(10000)	52	60	3	7.086D+00	0.5000	37	44	2	7.086D+00	0.3750
36(50000)	43	50	2	7.086D+00	2.1094	43	50	2	7.086D+00	2.2188
37(100)	102	146	14	1.059D-01	0.0000	102	114	1	1.103D-01	0.0000
37(500)	102	136	11	4.423D+00	0.0156	102	114	1	4.311D+00	0.0000
38(10000)	4	24	1	1.119D+07	0.0625	7	24	1	1.119D+07	0.0469
38(100000)	3	23	1	2.486D+09	0.5000	6	23	1	2.486D+09	0.5781
39(1000)	102	144	11	-2.498D+05	0.0156	102	201	32	-2.498D+05	0.0156
39(3000)	102	147	15	-2.249D+06	0.0469	102	194	33	-2.249D+06	0.0469

Tabla 2.7: Número de problemas en los que un algoritmo es ganador

Algoritmo	IT	f	LS	fbest	T
GBB	13	12	2	26	18
NGBB	9	46	47	34	34
Empates	56	20	29	18	26

Es claro que en la mayoría de los casos NGBB supera en eficiencia a GBB, ya que en el 59.97% de los problemas obtuvo el menor número de evaluaciones de la función; en el 60.26% de los problemas usó menos veces la búsqueda lineal; en el 43.59% de los problemas consiguió la mejor solución y en el 43.59% de los problemas empleó menos tiempo de CPU para satisfacer el criterio de parada.

2.4.2 Problema de Conformación Molecular

Un área importante de investigación en bioquímica computacional es la designación de moléculas para específicas aplicaciones. Ejemplos de tales aplicaciones son: el desarrollo de enzimas para la eliminación de basuras tóxicas, el desarrollo de nuevos catalizadores para el procesamiento materiales, y la designación de nuevos agentes anti-cancerígenos. La designación de estas drogas depende de la determinación exacta de la estructura de las macromoléculas biológicas. Este problema es conocido como el problema de Conformación Molecular, que consiste en hallar la configuración de una molécula cuya energía libre es la más baja.

Bajo el supuesto de que la estructura nativa de una molécula corresponde a una conformación para la que la energía es o está cerca del mínimo global, el problema de conformación molecular puede formularse como un problema de optimización.

A continuación describimos el modelo usado. Consideramos un polímero bidimensional que consiste en N átomos conectados por varas rígidas. El potencial E que empleamos para describir la energía de este sistema está dado por:

$$E = \sum_{j=1}^M \sum_{i=j+1}^N (\|a_i - a_j\|^2 - d_{ij}^2)^2, \quad (2.14)$$

donde $a_i = (x_i, y_i)^t \in \mathbb{R}^2$ es el átomo i , d_{ij} es la distancia conocida del átomo i al átomo

j , y $1 \leq M \leq N - 1$. Encontrar el mínimo de E se puede plantear como un problema de minimización irrestric­ta con $f : \mathbb{R}^{2N} \rightarrow \mathbb{R}$ definida como:

$$f(\mathbf{x}) = \sum_{j=1}^M \sum_{i=j+1}^N ((x_i - x_j)^2 + (y_i - y_j)^2 - d_{ij}^2)^2, \quad (2.15)$$

donde

$$\mathbf{x} = (x_1, x_2, \dots, x_N, y_1, y_2, \dots, y_N)^t \in \mathbb{R}^{2N}.$$

Para los experimentos numéricos tomamos $M = \lceil N/2 \rceil$, donde $\lceil r \rceil$ es el entero superior más cercano a $r \in \mathbb{R}$. En otras palabras, conocemos la distancia de los primeros M átomos a todos los demás átomos.

Todos los experimentos se corrieron en MATLAB 7.0 en un Computador Pentium IV de 3.0 GHz. Implementamos a GBB y a NGBB con los parámetros descritos en la Sección 2.4, salvo el parámetro η_k que lo tomamos como $\eta_k = |f(\mathbf{x}_0)|(0.9)^k$. Para escoger a σ en GBB y NGBB, consideramos el modelo parabólico descrito en [26, pp. 126–127].

Para la función `fminunc` del Toolbox de optimización de MATLAB 7.0 empleamos la opción:

```
options = optimset('largescale', 'on', 'gradobj', 'on', 'TolX', 5e-4,
'TolFun', 1e-15).
```

Empleamos el criterio de parada:

$$\|g_k\|_\infty \leq 2TolX,$$

donde $\|\cdot\|_\infty$ es la norma infinito y $TolX = 5 \times 10^{-4}$.

Comparamos los algoritmos GBB, MATLAB (*fminunc*) y NGBB para moléculas de diferentes números de átomos. Consideramos tres problemas test: M19, M38 y M60, que consisten en moléculas de 19, 38 y 60 átomos, respectivamente, cuya conformación óptima se muestra en la Figura 2.2. Para cada corrida generamos un iterado inicial aleatorio \mathbf{x}_0 , cuyas coordenadas están uniformemente distribuidas en el intervalo $[-10, 10]$.

En la Tabla 2.8 reportamos el número promedio de evaluaciones de la función f dada en (2.15) (EF), el promedio del tiempo de CPU en segundos (T), y el valor de f en la mejor solución encontrada (fbest), sobre 50 corridas de los algoritmos para cada problema test.

Para GBB y NGBB también reportamos en la Tabla 2.8 el promedio del número de ocasiones donde la búsqueda lineal es usada (BL). En las Figuras 2.3, 2.4 y 2.5 se observan las mejores conformaciones obtenidas por los algoritmos para los problemas M19, M38 y M60, respectivamente.

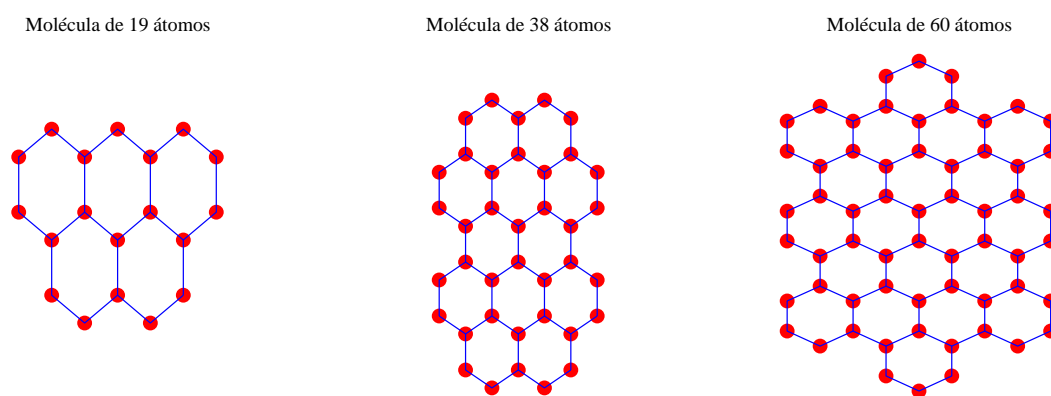


Figura 2.2: Configuraciones óptimas de los problemas M19, M38 y M60

Tabla 2.8: Resultados para los problemas M19, M38 y M60

Problema	MatLab			GBB				NGBB			
	EF	T	fbest	EF	BL	T	fbest	EF	BL	T	fbest
M19	20	0.59	1.96(-10)	4676	397	0.80	1.60(-9)	1372	109	0.29	1.02(-9)
M38	29	4.24	5.14(-10)	8287	600	8.64	1.36(-12)	2546	182	2.76	7.09(-10)
M60	35	18.07	2.12(-9)	14416	974	11.46	1.07(-12)	3898	259	3.60	8.18(-10)

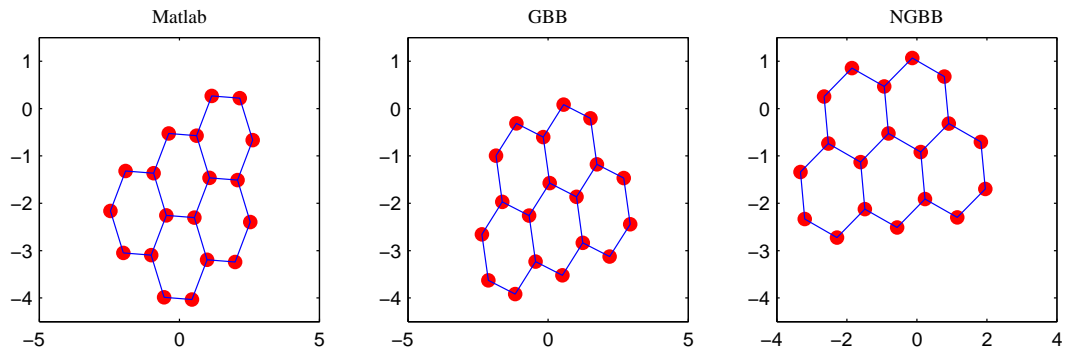


Figura 2.3: Mejor configuración obtenida por cada algoritmo para el problema M19

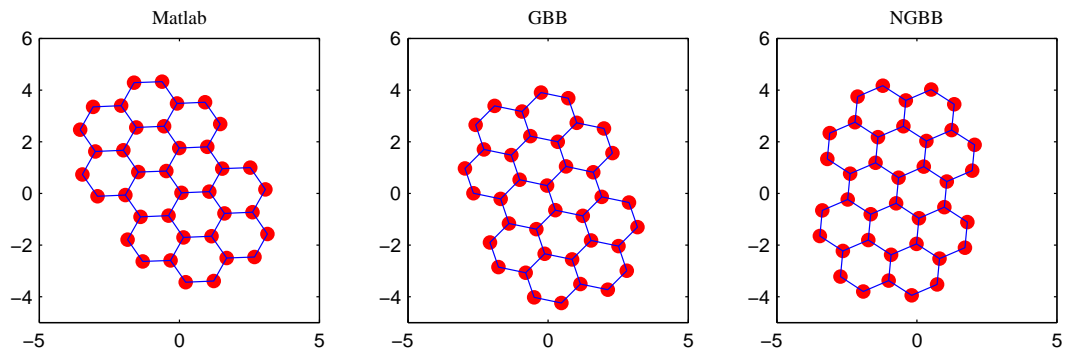


Figura 2.4: Mejor configuración obtenida por cada algoritmo para el problema M38

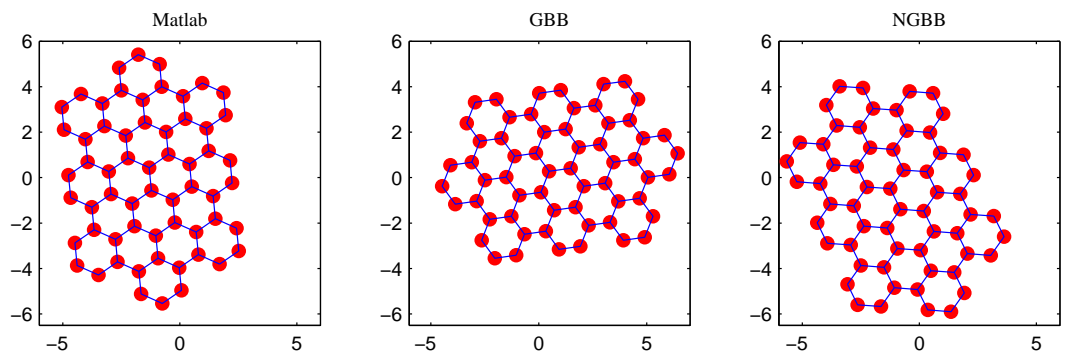


Figura 2.5: Mejor configuración obtenida por cada algoritmo para el problema M60

Capítulo 3

Algoritmo Libre de Derivadas para Sistemas de Ecuaciones no Lineales

“El Análisis Numérico es el estudio de algoritmos para problemas de matemática continua”

Lloyd Trefethen

En este capítulo presentamos un estudio teórico y experimental de un nuevo algoritmo libre de derivadas para sistemas de ecuaciones no lineales, que emplea sistemáticamente la dirección $\pm F(x)$ y, además, utiliza la búsqueda lineal propuesta en el Capítulo 2 con la función mérito $f(x) = \|F(x)\|^2$.

En la Sección 3.1 describimos el esquema del algoritmo NDF-SANE y realizamos un estudio de convergencia. En la Sección 3.2 realizamos experiencias numéricas con un extenso conjunto de funciones test y el problema del Trazado de Rayos Sísmicos.

3.1 Algoritmo NDF-SANE

Para describir el esquema del Algoritmo NDF-SANE (New Derivative-Free SANE) es necesario definir una función mérito y algunos parámetros.

Definimos la función mérito $f : \mathbb{R}^n \rightarrow \mathbb{R}$ como

$$f(x) = \|F(x)\|^2, \text{ para } x \in \mathbb{R}^n. \quad (3.1)$$

Dentro del esquema del Algoritmo NDF-SANE se consideran los siguientes parámetros:

- una sucesión de números positivos $\{\eta_k\}_{k \in \mathbb{N}}$ que satisfice

$$\sum_{k=0}^{\infty} \eta_k < \eta; \quad (3.2)$$

- $\alpha_0 > 0$;
- $0 < \gamma < 1$;
- $0 < \alpha_{min} < \alpha_{max} < \infty$;
- $0 < \sigma_{min} < \sigma_{max} < 1$.

En el Algoritmo 3.1 se muestra el esquema de NDF-SANE.

Algoritmo 3.1 Algoritmo NDF-SANE

Datos: $x_0 \in \mathbb{R}^n$, $0 < \gamma < 1$, $0 < \alpha_{min} < \alpha_{max} < \infty$, $0 < \sigma_{min} < \sigma_{max} < 1$, una sucesión $\{\eta_k\}_{k \in \mathbb{N}}$ tal que $\sum_{k=0}^{\infty} \eta_k < \eta < \infty$, y $k \leftarrow 0$.

```

1: mientras  $F(x_k) \neq 0$  hacer
2:     escoger  $\alpha_k$  tal que  $|\alpha_k| \in [\alpha_{min}, \alpha_{max}]$  (coeficiente espectral)
3:      $d \leftarrow -\alpha_k F(x_k)$ 
4:      $\lambda \leftarrow 1$ 
5:     busquedalineal  $\leftarrow$  verdadero
6:     mientras busquedalineal = verdadero hacer
7:         si  $f(x_k + \lambda d) \leq f(x_k) + \eta_k - \gamma \lambda^2 \|d\|^2$  entonces
8:              $d_k \leftarrow d$ 
9:             busquedalineal  $\leftarrow$  falso
10:        de lo contrario
11:            si  $f(x_k - \lambda d) \leq f(x_k) + \eta_k - \gamma \lambda^2 \|d\|^2$  entonces
12:                 $d_k \leftarrow -d$ 
13:                busquedalineal  $\leftarrow$  falso
14:            de lo contrario
15:                escoger  $\lambda_{new} \in [\sigma_1 \lambda, \sigma_2 \lambda]$ 
16:                 $\lambda \leftarrow \lambda_{new}$ 
17:            fin si
18:        fin si
19:    fin mientras
20:     $\lambda_k \leftarrow \lambda$ 
21:     $x_{k+1} \leftarrow x_k + \lambda_k d_k$ 
22:     $k \leftarrow k + 1$ 
23: fin mientras
    
```

Comentarios.

(i) Los iterados $x_{k+1} = x_k + \lambda_k d_k$ satisfacen la condición:

$$f(x_{k+1}) \leq f(x_k) + \eta_k - \gamma \lambda_k^2 \|d_k\|^2, \quad \forall k \geq 0. \quad (3.3)$$

(ii) La diferencia primordial entre los métodos SANE, DF-SANE y NDF-SANE es la manera de como escogen el tamaño de paso. El método NDF-SANE emplea la búsqueda lineal desarrollada en el Capítulo 2; en cambio, SANE utiliza GLL y DF-SANE emplea la combinación de GLL y LF.

3.1.1 Análisis de Convergencia

A continuación realizamos el análisis de convergencia del Algoritmo 3.1. El primer resultado que presentamos muestra que el algoritmo está bien definido.

Proposición 3.1. *El Algoritmo 3.1 está bien definido.*

Demostración. Como $\eta_k > 0$, entonces por la continuidad de $f(x)$ alguna de las condiciones

$$f(x_k + \lambda d) \leq f(x_k) + \eta_k - \gamma \lambda^2 \|d\|^2$$

o

$$f(x_k - \lambda d) \leq f(x_k) + \eta_k - \gamma \lambda^2 \|d\|^2$$

se cumple para λ suficientemente pequeño. \square

Las siguientes tres proposiciones son resultados técnicos necesarios para demostrar la convergencia del Algoritmo 3.1.

Proposición 3.2. *Si Ω_0 es acotado y $\{x_k\}_{k \in \mathbb{N}}$ es la sucesión generada por el Algoritmo 3.1, entonces*

$$\sum_{k=0}^{\infty} \|x_{k+1} - x_k\|^2 < \infty, \quad (3.4)$$

además,

$$\lim_{k \rightarrow \infty} \|x_{k+1} - x_k\| = 0 \quad (3.5)$$

y

$$\lim_{k \rightarrow \infty} \lambda_k \|F(x_k)\| = 0. \quad (3.6)$$

Demostración. Se tiene que $x_{k+1} - x_k = \lambda_k d_k$, $\|d_k\| = |\alpha_k| \|F(x_k)\|$ y $|\alpha_k| \in [\alpha_{min}, \alpha_{max}]$.

Luego,

$$\|x_{k+1} - x_k\|^2 = \lambda_k^2 \alpha_k^2 \|F(x_k)\|^2 \leq \lambda_k^2 \alpha_{max}^2 \|F(x_k)\|^2 = \alpha_{max}^2 \lambda_k^2 \|d_k\|^2. \quad (3.7)$$

Ahora, por (3.3)

$$\lambda_k^2 \|d_k\|^2 \leq \frac{\eta_k}{\gamma} + \frac{f(x_k) - f(x_{k+1})}{\gamma}. \quad (3.8)$$

De (3.7) y (3.8) obtenemos:

$$\|x_{k+1} - x_k\|^2 \leq \frac{\alpha_{max}^2}{\gamma} (\eta_k + (f(x_k) - f(x_{k+1}))). \quad (3.9)$$

Como η_k satisface (3.2) y Ω_0 es acotado, sumando en ambos lados de (3.9),

$$\begin{aligned} \sum_{k=0}^{\infty} \|x_{k+1} - x_k\|^2 &\leq \frac{\alpha_{max}^2}{\gamma} \sum_{k=0}^{\infty} \eta_k + \frac{\alpha_{max}^2}{\gamma} \sum_{k=0}^{\infty} (f(x_k) - f(x_{k+1})) \\ &\leq \frac{\alpha_{max}^2 (\eta + f(x_0))}{\gamma} < \infty, \end{aligned}$$

de donde deducimos

$$\lim_{k \rightarrow \infty} \|x_{k+1} - x_k\|^2 = 0. \quad (3.10)$$

Como $\lambda_k \|d_k\| = \|x_{k+1} - x_k\| \geq 0$, entonces por (3.10) podemos escribir:

$$\lim_{k \rightarrow \infty} \|x_{k+1} - x_k\| = 0$$

y

$$\lim_{k \rightarrow \infty} \lambda_k \|d_k\| = 0. \quad (3.11)$$

Como $\|d_k\| = |\alpha_k| \|F(x_k)\|$ y $|\alpha_k| \in [\alpha_{min}, \alpha_{max}]$, entonces

$$\|d_k\| \geq \alpha_{min} \|F(x_k)\|. \quad (3.12)$$

Por (3.11) y (3.12) tenemos que

$$\lim_{k \rightarrow \infty} \lambda_k \|F(x_k)\| = 0.$$

Esto completa la demostración. \square

La Proposición 3.3 corresponde al Lema 3.3 en [24].

Proposición 3.3. Sean $\{a_k\}_{k \in \mathbb{N}}$ y $\{b_k\}_{k \in \mathbb{N}}$ sucesiones de números positivos que satisfacen:

$$a_{k+1} \leq (1 + b_k)a_k + b_k \text{ y } \sum_{k=0}^{\infty} b_k < \infty.$$

Entonces, $\{a_k\}_{k \in \mathbb{N}}$ converge.

Proposición 3.4. Si Ω_0 es acotado y $\{x_k\}_{k \in \mathbb{N}}$ es la sucesión generada por el Algoritmo 3.1, entonces la sucesión $\{\|F(x_k)\|\}_{k \in \mathbb{N}}$ converge.

Demostración. Como $f(x_k) \geq 0$ y $(1 + \eta_k) \geq 1$, para todo $k \in \mathbb{N}$, entonces por la condición (3.3) podemos escribir:

$$f(x_{k+1}) \leq f(x_k) + \eta_k \leq (1 + \eta_k)f(x_k) + \eta_k.$$

Tomando $a_k = f(x_k)$ y $b_k = \eta_k$, entonces la desigualdad anterior adquiere la forma:

$$a_{k+1} \leq (1 + b_k)a_k + b_k,$$

además, $\sum_{k=0}^{\infty} b_k < \eta < \infty$. Por lo tanto, por la Proposición 3.3 la sucesión $\{a_k\}_{k \in \mathbb{N}}$ converge. En otras palabras, la sucesión $\{f(x_k)\}_{k \in \mathbb{N}}$ converge. Como $f(x) = \|F(x)\|^2$ y $\|F(x)\| \geq 0$, entonces la sucesión $\{\|F(x_k)\|\}_{k \in \mathbb{N}}$ converge. \square

Teorema 3.1. Asumamos que Ω_0 es acotado y que $\{x_k\}_{k \in \mathbb{N}}$ es la sucesión generada por el Algoritmo 3.1. Entonces, todo punto límite x_* de $\{x_k\}_{k \in \mathbb{N}}$ satisface

$$F(x_*)^t J(x_*) F(x_*) = 0. \quad (3.13)$$

Demostración. Sea x_* un punto límite de $\{x_k\}_{k \in \mathbb{N}}$. Sin pérdida de generalidad podemos suponer que $\{x_k\}_{k \in \mathbb{N}}$ converge a x_* . Por la Proposición 3.2 se tiene que

$$\lim_{k \rightarrow \infty} \lambda_k \|F(x_k)\| = 0$$

que es cierto si

$$\lim_{k \rightarrow \infty} \|F(x_k)\| = 0,$$

o si

$$\liminf_{k \rightarrow \infty} \lambda_k = 0. \quad (3.14)$$

Si $\lim_{k \rightarrow \infty} \|F(x_k)\| = 0$, la conclusión del teorema es inmediata.

Supongamos que (3.14) es cierta. Entonces, existe un conjunto infinito de índices $K = \{k_1, k_2, k_3, \dots\} \subseteq \mathbb{N}$ tal que

$$\lim_{j \rightarrow \infty} \lambda_{k_j} = 0.$$

Por la forma como λ_{k_j} es escogido en el Algoritmo 3.1, existe un índice \bar{j} suficientemente grande tal que para todo $j \geq \bar{j}$, existe ρ_{k_j} ($0 < \sigma_{min} \leq \rho_{k_j} \leq \sigma_{max}$) para el que $\lambda = \lambda_{k_j}/\rho_{k_j}$ no satisface la condición (3.3), es decir,

$$\begin{aligned} f\left(x_{k_j} + \frac{\lambda_{k_j}}{\rho_{k_j}} d_{k_j}\right) &> f(x_{k_j}) + \eta_{k_j} - \gamma \frac{\lambda_{k_j}^2}{\rho_{k_j}^2} \|d_{k_j}\|^2 \\ &\geq f(x_{k_j}) - \gamma \frac{\lambda_{k_j}^2}{\rho_{k_j}^2} \|d_{k_j}\|^2. \end{aligned}$$

Por lo tanto,

$$\frac{f(x_{k_j} + \frac{\lambda_{k_j}}{\rho_{k_j}} d_{k_j}) - f(x_{k_j})}{\lambda_{k_j}/\rho_{k_j}} > -\gamma \frac{\lambda_{k_j}}{\rho_{k_j}} \|d_{k_j}\|^2 \geq -\gamma \frac{\lambda_{k_j}}{\sigma_{min}} \|d_{k_j}\|^2.$$

Mediante el Teorema del Valor Medio esta relación se puede expresar como

$$\nabla f(x_{k_j} + t_{k_j} d_{k_j})^t d_{k_j} > -\gamma \frac{\lambda_{k_j}}{\sigma_{min}} \|d_{k_j}\|^2, \text{ para } j \geq \bar{j}, \quad (3.15)$$

donde $t_{k_j} \in [0, \lambda_{k_j}/\rho_{k_j}]$ tiende a cero cuando $j \rightarrow \infty$.

Como $\{\alpha_k\}_{k \in \mathbb{N}}$ es acotada, entonces ella tiene una subsucesión convergente. Supongamos, sin pérdida de generalidad, que $\{\alpha_{k_j}\}_{k_j \in K}$ converge a α_* . Por la Proposición 3.4 la sucesión $\{\|F(x_k)\|\}_{k \in \mathbb{N}}$ converge, por lo tanto, si observamos que $d_k = \pm \alpha_k F(x_k)$ y $(x_{k_j} + t_{k_j} d_{k_j}) \rightarrow x_*$ cuando $j \rightarrow \infty$, entonces tomando límite en (3.15) se deduce que

$$\nabla f(x_*)^t d_* \geq 0, \quad (3.16)$$

donde $d_* = \lim_{j \rightarrow \infty} d_{k_j}$.

Sean $K_1, K_2 \subset K$ tales que

$$d_{k_j} = -\alpha_{k_j} F(x_{k_j}), \text{ para todo } k_j \in K_1$$

y

$$d_{k_j} = \alpha_{k_j} F(x_{k_j}), \quad \text{para todo } k_j \in K_2.$$

Por (3.16) podemos escribir:

$$\begin{aligned} \lim_{j \rightarrow \infty, k_j \in K_1} \nabla f(x_{k_j})^t d_{k_j} &= \lim_{j \rightarrow \infty, k_j \in K_1} -\alpha_{k_j} F(x_{k_j})^t J(x_{k_j}) F(x_{k_j}) \\ &= -\alpha_* F(x_*)^t J(x_*) F(x_*) \geq 0 \end{aligned} \quad (3.17)$$

y

$$\begin{aligned} \lim_{j \rightarrow \infty, k_j \in K_2} \nabla f(x_{k_j})^t d_{k_j} &= \lim_{j \rightarrow \infty, k_j \in K_2} \alpha_{k_j} F(x_{k_j})^t J(x_{k_j}) F(x_{k_j}) \\ &= \alpha_* F(x_*)^t J(x_*) F(x_*) \geq 0. \end{aligned} \quad (3.18)$$

De (3.17) y (3.18) se deduce que

$$\alpha_* F(x_*)^t J(x_*) F(x_*) = 0.$$

Como $0 < \alpha_{min} \leq |\alpha_k| \leq \alpha_{max}$, para todo $k \in \mathbb{N}$, entonces

$$F(x_*)^t J(x_*) F(x_*) = 0.$$

Esto completa la demostración del teorema. \square

En todo el análisis previo no hemos considerado propiedades de la la matriz Jacobiana de F . Sólomente hemos visto algunas propiedades del Algoritmo 3.1 para cualquier función F . A continuación realizamos un análisis para funciones F muy particulares. El siguiente resultado muestra que si x_* es un punto límite de la sucesión $\{x_k\}_{k \in \mathbb{N}}$ generada por el Algoritmo 3.1 y 0 no es un autovalor de la matriz $J(x_*)$, entonces $F(x_*) = 0$.

Corolario 3.1. *Asumamos que Ω_0 es acotado, $\{x_k\}_{k \in \mathbb{N}}$ es la sucesión generada por el Algoritmo 3.1 y que x_* es un punto límite de $\{x_k\}_{k \in \mathbb{N}}$ tal que para todo $v \in \mathbb{R}^n$, $v \neq 0$,*

$$v^t J(x_*) v \neq 0.$$

Entonces, $F(x_) = 0$.*

Demostración. Por Teorema 3.1 tenemos que $F(x_*)^t J(x_*) F(x_*) = 0$ que implica, por la hipótesis del corolario, que $F(x_*) = 0$. \square

Para el siguiente resultado de convergencia necesitamos una definición.

Definición 3.1. Decimos que un mapeo $F : \mathbb{R}^n \rightarrow \mathbb{R}^n$ es *estrictamente monótono*, si $J(x)$ es positivo definida para todo $x \in \mathbb{R}^n$. Decimos que el mapeo F es *estricto*¹, si F o $-F$ son estrictamente monótonos. Si un mapeo es estricto y admite solución, esta debe ser única [60].

Corolario 3.2. *Asumamos que Ω_0 es acotado y que $\{x_k\}_{k \in \mathbb{N}}$ es la sucesión generada por el Algoritmo 3.1. Si el mapeo F es estricto, entonces $\{x_k\}_{k \in \mathbb{N}}$ converge a la solución de (1.1).*

Demostración. Como F es estricto, entonces $J(x)$ es positivo o negativo definida y el sistema (1.1) posee solución única $x_* \in \mathbb{R}^n$. Por el Teorema 3.1 todo punto límite \bar{x} de $\{x_k\}_{k \in \mathbb{N}}$ satisface que $F(\bar{x})^t J(\bar{x}) F(\bar{x}) = 0$. Pero $J(x)$ es positivo o negativo definida, luego $F(\bar{x}) = 0$. Como x_* es la solución única de (1.1) y todo punto límite \bar{x} de $\{x_k\}_{k \in \mathbb{N}}$ satisface que $F(\bar{x}) = 0$, entonces $\{x_k\}_{k \in \mathbb{N}}$ converge a x_* . \square

En el Teorema 3.1 demostramos que para todo punto límite x_* de $\{x_k\}_{k \in \mathbb{N}}$ el gradiente de $\|F(x_*)\|^2$ es ortogonal al vector residual $F(x_*)$. El caso en el que el punto límite x_* es tal que $F(x_*) = 0$ merece ser analizado.

Seguidamente mostramos nuestro primer resultado de convergencia local. El Teorema 3.2 expresa que si existe un punto límite de la sucesión $\{x_k\}_{k \in \mathbb{N}}$ que es solución de (1.1), entonces todos los puntos límites de esta sucesión son solución de (1.1). También este teorema expresa que si una solución aislada de (1.1) es un punto límite de $\{x_k\}_{k \in \mathbb{N}}$, entonces la sucesión generada por el Algoritmo 3.1 converge a esta solución.

Teorema 3.2. *Asumamos que Ω_0 es acotado, $\{x_k\}_{k \in \mathbb{N}}$ es la sucesión generada por el Algoritmo 3.1 y que existe un punto límite x_* de $\{x_k\}_{k \in \mathbb{N}}$ tal que $F(x_*) = 0$. Entonces se cumple:*

- (i) *La sucesión $\{F(x_k)\}_{k \in \mathbb{N}}$ converge a 0. Además, todo punto límite de $\{x_k\}_{k \in \mathbb{N}}$ es solución de (1.1).*

¹Traducción del inglés del término strict.

(ii) Si existe $\delta > 0$ tal que $F(x) \neq 0$ siempre que $0 < \|x - x_*\| \leq \delta$, entonces la sucesión $\{x_k\}_{k \in \mathbb{N}}$ converge a x_* .

Demostración. Prueba de (i). Sea \bar{x} un punto límite de $\{x_k\}_{k \in \mathbb{N}}$ distinto a x_* . Existe un conjunto infinito de índices $K = \{k_1, k_2, k_3, \dots\} \subseteq \mathbb{N}$ tal que $\{x_k\}_{k \in K}$ converge a \bar{x} . Por la continuidad de F se tiene que

$$\lim_{j \rightarrow \infty} \|F(x_{k_j})\| = \|F(\bar{x})\|. \quad (3.19)$$

Como x_* es un punto límite de $\{x_k\}_{k \in \mathbb{N}}$ tal que $F(x_*) = 0$, entonces por la Proposición 3.4 y la ecuación (3.19) podemos escribir:

$$\|F(\bar{x})\| = \|F(x_*)\| = 0,$$

que implica $F(\bar{x}) = 0$. Como \bar{x} fue escogido arbitrariamente, entonces

$$\lim_{k \rightarrow \infty} F(x_k) = 0.$$

En otras palabras, todo punto límite de $\{x_k\}_{k \in \mathbb{N}}$ es solución de (1.1).

Prueba de (ii). Por la Proposición 3.2 se tiene que

$$\lim_{k \rightarrow \infty} \|x_{k+1} - x_k\| = 0.$$

Luego, existe $k_1 \in \mathbb{N}$ tal que

$$\|x_{k+1} - x_k\| \leq \delta/2, \text{ para todo } k \geq k_1. \quad (3.20)$$

Consideremos el conjunto

$$A = \{x \in \mathbb{R}^n : \delta/2 \leq \|x - x_*\| \leq \delta\}.$$

Por hipótesis, el conjunto A no contiene soluciones de (1.1). Pero, por (i), todos los puntos límites de $\{x_k\}_{k \in \mathbb{N}}$ son solución de (1.1). Por lo tanto, A no contiene puntos límites de $\{x_k\}_{k \in \mathbb{N}}$. De esta forma, como A es compacto, no puede contener infinitos elementos de $\{x_k\}_{k \in \mathbb{N}}$. En consecuencia, existe $k_2 \in \mathbb{N}$ tal que

$$x_k \notin A, \text{ para todo } k \geq k_2. \quad (3.21)$$

Sea $k_3 \geq \max\{k_1, k_2\}$ tal que

$$\|x_{k_3} - x_*\| \leq \delta/2.$$

Por (3.20), tenemos:

$$\|x_{k_3+1} - x_*\| \leq \|x_{k_3} - x_*\| + \|x_{k_3+1} - x_{k_3}\| \leq \delta.$$

Pero, por (3.21), $x_{k_3+1} \notin A$, de esta forma, tenemos que

$$\|x_{k_3+1} - x_*\| \leq \delta/2.$$

Continuando con este argumento inductivo tenemos que

$$\|x_k - x_*\| \leq \delta/2, \text{ para todo } k \geq k_3. \quad (3.22)$$

Esto implica que todo punto límite \hat{x} de $\{x_k\}_{k \in \mathbb{N}}$ satisface que

$$\|\hat{x} - x_*\| \leq \delta/2.$$

Por (i), $F(\hat{x}) = 0$ para todo punto límite \hat{x} , y por hipótesis el conjunto

$$\{x \in \mathbb{R}^n : 0 < \|x - x_*\| \leq \delta/2\}$$

no contiene soluciones de (1.1). Por lo tanto, este conjunto no contiene puntos límites de $\{x_k\}_{k \in \mathbb{N}}$. Sólomente el punto límite \hat{x} que satisface $\|\hat{x} - x_*\| \leq \delta/2$ es x_* . Entonces, por (3.22) la sucesión $\{x_k\}_{k \in \mathbb{N}}$ converge a x_* , que era lo que queríamos demostrar. \square

Para probar nuestro segundo teorema de convergencia local necesitamos de una nueva definición y un lema técnico.

Definición 3.2. Decimos que $x_* \in \mathbb{R}^n$ es una *solución fuertemente aislada* de (1.1) si $F(x_*) = 0$ y existe $\varepsilon > 0$ tal que

$$0 < \|x - x_*\| \leq \varepsilon > 0 \Rightarrow F(x)^t J(x) F(x) \neq 0.$$

En otras palabras, en una vecindad de una solución fuertemente aislada el vector residual $F(x)$ no es ortogonal al vector gradiente de $f(x) = \|F(x)\|^2$.

Lema 3.1. *Asumamos que $F(x_*) = 0$, y $k_0 \in \mathbb{N}$. Entonces, para todo $\delta_1 > 0$ existe $\delta \in (0, \delta_1]$ tal que el k_0 -ésimo iterado calculado por el Algoritmo 3.1 satisface*

$$\|x_{k_0} - x_*\| < \delta_1,$$

siempre que x_0 satisfaga $\|x_0 - x_\| < \delta$. (La tesis del lema se cumple independientemente de la forma como se escojan α_k y λ_k , para $k = 0, 1, \dots, k_0 - 1$.)*

Demostración. Procederemos por inducción. Si $k_0 = 0$ el resultado es inmediato tomando $\delta = \delta_1$. Asumamos que, dado $\delta_1 > 0$, existe $\delta \in (0, \delta_1]$ tal que

$$\|x_0 - x_*\| < \delta \Rightarrow \|x_{k_0} - x_*\| < \delta_1. \quad (3.23)$$

Ahora probemos la tesis del lema para $k_0 + 1$. Sea $\delta_1 > 0$. Como $F(x_*) = 0$, entonces por la continuidad de F existe

$$\delta_2 \in (0, \delta_1/2] \quad (3.24)$$

tal que

$$\|x_0 - x_*\| < \delta_2 \Rightarrow \|F(x_{k_0})\| < \delta_1/c, \quad (3.25)$$

donde $c = 2 \max\{\alpha_{max}, 1\}$. Pero,

$$\|x_{k_0+1} - x_{k_0}\| = |\lambda_{k_0} \alpha_{k_0}| \|F(x_{k_0})\| \leq \alpha_{max} \|F(x_{k_0})\|.$$

Luego por (3.25) tenemos:

$$\|x_0 - x_*\| < \delta_2 \Rightarrow \|x_{k_0+1} - x_{k_0}\| < \delta_1/2. \quad (3.26)$$

Por hipótesis inductiva existe $\delta \in (0, \delta_2]$ tal que

$$\|x_0 - x_*\| < \delta \Rightarrow \|x_{k_0} - x_*\| < \delta_2. \quad (3.27)$$

Por (3.24), (3.26) y (3.27) podemos escribir:

$$\begin{aligned} \|x_{k_0+1} - x_*\| &= \|(x_{k_0+1} - x_{k_0}) + (x_{k_0} - x_*)\| \\ &\leq \|x_{k_0+1} - x_{k_0}\| + \|x_{k_0} - x_*\| \\ &\leq \frac{\delta_1}{2} + \frac{\delta_1}{2} = \delta_1. \end{aligned}$$

Esto completa la prueba del lema. □

El Teorema 3.3 dice que si x_0 se escoge lo suficientemente cerca de alguna solución fuertemente aislada x_* , entonces la sucesión $\{x_k\}_{k \in \mathbb{N}}$ converge a x_* .

Teorema 3.3. *Asumamos que Ω_0 es acotado y que x_* es una solución fuertemente aislada de (1.1). Entonces, existe $\delta > 0$ tal que*

$$\|x_0 - x_*\| < \delta \Rightarrow \lim_{k \rightarrow \infty} x_k = x_*,$$

donde $\{x_k\}_{k \in \mathbb{N}}$ es la sucesión generada por el Algoritmo 3.1.

Demostración. Sea $\varepsilon > 0$ tal que

$$0 < \|x - x_*\| \leq \varepsilon > 0 \Rightarrow F(x)^t J(x) F(x) \neq 0. \quad (3.28)$$

Sea $c = \max\{\alpha_{max}, 1\}$. Como $F(x_*) = 0$, entonces por continuidad de F existe

$$\varepsilon_1 \in (0, \varepsilon/2] \quad (3.29)$$

tal que

$$\|x - x_*\| \leq \varepsilon_1 \Rightarrow \|F(x_k)\| \leq \frac{\varepsilon}{2c}. \quad (3.30)$$

Pero, $|\lambda_k \alpha_k| \leq \alpha_{max}$ y $\|x_{k+1} - x_k\| \leq |\lambda_k \alpha_k| \|F(x_k)\|$, luego por (3.30)

$$\|x_k - x_*\| \leq \varepsilon_1 \Rightarrow \|x_{k+1} - x_k\| \leq \frac{\varepsilon}{2}. \quad (3.31)$$

Definamos

$$T_\varepsilon = \{x \in \mathbb{R}^n : \varepsilon_1 \leq \|x - x_*\| \leq \varepsilon\}.$$

Como T_ε es compacto y f es continua, existen $\hat{x} \in T_\varepsilon$ y $\beta > 0$, tales que

$$\beta \equiv f(\hat{x}) \leq f(x), \text{ para todo } x \in T_\varepsilon. \quad (3.32)$$

Ahora, como f es continua, el conjunto

$$\{x \in \mathbb{R}^n : \|x - x_*\| \leq \varepsilon_1/2 \text{ y } f(x) < \beta/2\}$$

es una vecindad abierta de x_* . Por lo tanto, existe $\delta_1 \in (0, \varepsilon_1/2)$ tal que

$$\|x - x_*\| \leq \delta_1 \Rightarrow f(x) < \beta/2. \quad (3.33)$$

Sea $k_0 \in \mathbb{N}$ tal que

$$\sum_{k=k_0}^{\infty} \eta_k < \beta/2. \quad (3.34)$$

Probemos por inducción que

$$f(x_k) \leq f(x_{k_0}) + \sum_{i=k_0}^{k-1} \eta_i, \text{ para } k \geq k_0 + 1. \quad (3.35)$$

Por (3.3), (3.35) es cierta para $k = k_0 + 1$. Supongamos que (3.35) se cumple para $k = k_0 + j$, para algún entero $j \geq 1$. Demostremos que (3.35) se cumple para $k = k_0 + j + 1$. Por (3.3) y la hipótesis inductiva podemos escribir:

$$f(x_{k_0+j+1}) \leq f(x_{k_0+j}) + \eta_{k_0+j} \leq f(x_{k_0}) + \sum_{i=k_0}^{k_0+j-1} \eta_i + \eta_{k_0+j} = f(x_{k_0}) + \sum_{i=k_0}^{k_0+j} \eta_i,$$

con lo cual culmina la prueba inductiva.

De esta forma, por (3.34) y (3.35),

$$f(x_k) \leq f(x_{k_0}) + \beta/2, \text{ para } k \geq k_0 + 1. \quad (3.36)$$

Aplicando el Lema 3.1 para k_0 deducimos que existe $\delta \in (0, \delta_1)$ tal que

$$\|x_0 - x_*\| \leq \delta \Rightarrow \|x_{k_0} - x_*\| \leq \delta_1 < \varepsilon_1. \quad (3.37)$$

Por (3.33) y (3.37), $f(x_{k_0}) < \beta/2$. Ahora, por (3.36),

$$f(x_k) < \beta, \text{ para } k \geq k_0 + 1. \quad (3.38)$$

Probemos por inducción que, tomando $x_0 \in \mathbb{R}^n$ tal que $\|x_0 - x_*\| \leq \delta$, entonces

$$\|x_{k_0+j} - x_*\| \leq \varepsilon_1, \text{ para todo } j \in \mathbb{N}. \quad (3.39)$$

Por (3.37), tenemos que (3.39) es cierta para $j = 0$. Asumamos como hipótesis inductiva que, para algún $j \geq 1$,

$$\|x_{k_0+j-1} - x_*\| < \varepsilon_1.$$

Como $\varepsilon_1 < \varepsilon/2$, entonces

$$\|x_{k_0+j-1} - x_*\| < \varepsilon/2.$$

Pero, por (3.31) podemos escribir:

$$\begin{aligned}
\|x_{k_0+j} - x_*\| &= \|(x_{k_0+j} - x_{k_0+j-1}) + (x_{k_0+j-1} - x_*)\| \\
&\leq \|x_{k_0+j} - x_{k_0+j-1}\| + \|x_{k_0+j-1} - x_*\| \\
&\leq \frac{\varepsilon}{2} + \frac{\varepsilon}{2} = \varepsilon.
\end{aligned} \tag{3.40}$$

Por (3.38), $f(x_{k_0+j}) < \beta$, luego (3.32) y (3.40) implican que $\|x_{k_0+j} - x_*\| \leq \varepsilon_1$. Esto completa la prueba inductiva.

Por lo tanto, $\{x_k\}_{k \geq k_0+1} \subset \{x \in \mathbb{R}^n : \|x - x_*\| \leq \varepsilon_1\}$. De esta forma, todos los puntos límites \bar{x} de $\{x_k\}_{k \in \mathbb{N}}$ son tales que $\|\bar{x} - x_*\| \leq \varepsilon_1 < \varepsilon$. Pero por (3.28) y el Teorema 3.1, el único punto límite que satisface lo anterior es x_* . De esta forma, $\lim_{k \rightarrow \infty} x_k = x_*$, que era lo que queríamos demostrar. \square

Corolario 3.3. *Asumamos que Ω_0 es acotado y que $\{x_k\}_{k \in \mathbb{N}}$ es la sucesión generada por el Algoritmo 3.1. Si $J(x_*)$ es positiva definida o negativa definida, entonces existe $\delta > 0$ tal que*

$$\|x_0 - x_*\| \leq \delta \Rightarrow \lim_{k \rightarrow \infty} x_k = x_*.$$

Demostración. Empleando la continuidad de J obtenemos que x_* es una solución fuertemente aislada de (1.1). Luego, por el Teorema 3.3 existe $\delta > 0$ tal que

$$\|x_0 - x_*\| \leq \delta \Rightarrow \lim_{k \rightarrow \infty} x_k = x_*,$$

que era lo que queríamos demostrar. \square

3.2 Resultados Numéricos

En esta sección reportamos los resultados numéricos obtenidos para el Algoritmo 3.1 sobre un extenso conjunto de funciones test y el problema del Trazado de Rayos Sísmicos.

3.2.1 Funciones test

Implementamos NDF-SANE (Algoritmo 3.1) con los siguientes parámetros: $\alpha_{min} = 10^{-10}$, $\alpha_{max} = 10^{10}$, $\alpha_0 = 1$, $\sigma_{min} = 0.1$, $\sigma_{max} = 0.5$, $\gamma = 10^{-4}$, $\eta_k = \theta(1 - 10^{-10})^k$,

donde

$$\theta = \begin{cases} \|F(x_0)\|^2, & \|F(x_0)\|^2 \leq 10^5, \\ 10^6, & \|F(x_0)\|^2 > 10^5. \end{cases}$$

En DF-SANE (Algoritmo 1.10) y NDF-SANE, el coeficiente espectral lo calculamos a través de la fórmula:

$$\alpha_k = \frac{s_k^t s_k}{s_k^t y_k},$$

donde $s_k = x_{k+1} - x_k$ y $y_k = F(x_{k+1}) - F(x_k)$. Escogemos α_k tal $|\alpha_k| \in [\alpha_{min}, \alpha_{max}]$. Sin embargo, si $|\alpha_k| \notin [\alpha_{min}, \alpha_{max}]$, reemplazamos el coeficiente espectral por

$$\alpha_k = \begin{cases} 1 & \text{si } \|F(x_k)\| > 1, \\ \|F(x_k)\|^{-1} & \text{si } 10^{-5} \leq \|F(x_k)\| \leq 1, \\ 10^5 & \text{si } \|F(x_k)\| < 10^{-5}. \end{cases}$$

Para escoger λ_{new} en el Algoritmo 3.1 procedimos de la siguiente forma. Dado $\lambda > 0$, tomamos $\lambda_{new} > 0$ como:

$$\lambda_{new} = \begin{cases} \sigma_{min} \lambda & \text{si } \lambda_c < \sigma_{min} \lambda, \\ \sigma_{max} \lambda & \text{si } \lambda_c > \sigma_{max} \lambda, \\ \lambda_c & \text{de lo contrario,} \end{cases}$$

donde

$$\lambda_c = \frac{\lambda^2 f_c}{f_c + (2\lambda - 1)f(x_k)},$$

y

$$f_c = \text{máx}\{f(x_k + \lambda d), f(x_k - \lambda d)\}.$$

Este modelo parabólico es similar al descrito en [44, pp. 142-143], en el cual la matriz Jacobiana en x_k es reemplazada por la matriz identidad.

También implementamos a SANE (Algoritmo 1.9) con los siguientes parámetros: $\gamma = 10^{-4}$, $\varepsilon = 10^{-8}$, $\sigma_1 = 0.1$, $\sigma_2 = 0.5$, $\alpha_0 = 1$, $M = 10$, y

$$\delta = \begin{cases} 1 & \text{si } \|F(x_k)\| > 1, \\ \|F(x_k)\|^{-1} & \text{si } 10^{-5} \leq \|F(x_k)\| \leq 1, \\ 10^5 & \text{si } \|F(x_k)\| < 10^{-5}. \end{cases}$$

Para DF-SANE empleamos los siguientes parámetros: $nexp = 2$, $\sigma_{min} = 10^{-10}$, $\sigma_{max} = 10^{10}$, $\sigma_0 = 1$, $\tau_{min} = 0.1$, $\tau_{max} = 0.5$, $\gamma = 10^{-4}$, $M = 10$, $\eta_k = \|F(x_0)\|/(1+k)^2$ para todo $k \in \mathbb{N}$.

En SANE, DF-SANE y NDF-SANE detuvimos el proceso cuando

$$\frac{\|F(x_k)\|}{\sqrt{n}} = e_a + e_r \frac{\|F(x_0)\|}{\sqrt{n}}, \quad (3.41)$$

donde $e_a = 10^{-5}$ y $e_r = 10^{-4}$.

Corrimos SANE, DF-SANE y NDF-SANE usando un conjunto de funciones test. Todas las funciones test se describen en el Apéndice A.

Los resultados numéricos se muestran en las Tablas 3.1-3.6. Reportamos solamente una falla de DF-SANE y usamos el símbolo (*), cuando corrimos la función 18 con $n = 100$. En este caso DF-SANE falló porque genera una sucesión que converge a un punto \bar{x} para el que $(J(\bar{x})^t F(\bar{x}))^t F(\bar{x}) = 0$, pero $F(\bar{x}) \neq 0$ y $J(\bar{x})^t F(\bar{x}) \neq 0$.

Los resultados de las Tablas 3.1 y 3.1 se resumen en las Tablas 3.3-3.6. En la Tabla 3.3 comparamos la eficiencia en la ejecución (números de problemas para los que un algoritmo posee menor número de iteraciones, menor número de evaluaciones de la función, menor número de veces que emplea la búsqueda lineal y menor tiempo de CPU) entre SANE, DF-SANE y NDF-SANE. En la Tabla 3.4 comparamos la eficiencia en la ejecución entre SANE y DF-SANE. En la Tabla 3.5 comparamos la eficiencia en la ejecución entre SANE y NDF-SANE. En la Tabla 3.4 comparamos la eficiencia en la ejecución entre DF-SANE y NDF-SANE. En las Tablas 3.1 y 3.1 reportamos el número de la función y la dimensión (Función(n)), el número de iteraciones (IT), el número de evaluaciones de la función (EF), número de ocasiones que se emplea la búsqueda lineal (BL), y tiempo de CPU en segundos (T).

En SANE es necesario evaluar la derivada direccional $(J(x_k)^t F(x_k))^t F(x_k)$ en cada iteración. Como asumimos que la matriz Jacobiana no es fácilmente disponible, usamos el factor

$$(J(x_k)^t F(x_k))^t F(x_k) = F(x_k)^t J(x_k) F(x_k)$$

y la aproximación

$$J(x_k) F(x_k) = \frac{F(x_k + hF(x_k)) - F(x_k)}{h},$$

donde $h > 0$ pequeño. De esta forma, calculamos la derivada direccional con una evaluación adicional de la función (incluida en EF) en cada iteración.

Nuestros resultados indican que el nuevo algoritmo libre de derivadas NDF-SANE compete favorablemente con los algoritmos SANE y DF-SANE.

Tabla 3.1: Resultados para los problemas 1-22

Función(n)	SANE				DF-SANE				NDF-SANE			
	IT	EF	BL	T	IT	EF	BL	T	IT	EF	BL	T
1(1000)	5	10	0	0.0000	5	5	0	0.0000	5	5	0	0.0156
1(10000)	2	4	0	0.0156	2	2	0	0.0156	2	2	0	0.0000
2(1000)	212	475	49	0.0938	210	218	3	0.0469	153	157	1	0.0313
2(10000)	168	353	14	0.7500	38	52	4	0.1094	597	603	1	1.4219
3(1000)	49	120	16	0.0156	14	21	2	0.0156	14	21	2	0.0000
3(10000)	28	73	10	0.1563	48	90	10	0.1875	81	132	15	0.3125
4(9999)	128	358	86	0.7031	102	319	76	0.6563	108	247	56	0.5156
4(69999)	154	404	86	5.6406	172	659	138	9.9063	126	283	64	4.3750
5(49)	289	1065	243	0.0000	524	2062	466	0.0156	442	962	201	0.0156
5(99)	3862	19095	3839	0.2969	2026	11086	1978	0.1719	1639	3761	789	0.0625
6(100)	2	5	1	0.0000	3	3	0	0.0000	2	5	1	0.0000
6(10000)	2	5	1	0.0000	3	3	0	0.0000	2	5	1	0.0000
7(100)	23	49	2	0.0000	23	29	2	0.0000	23	29	2	0.0000
7(10000)	23	49	2	0.0625	23	29	2	0.0469	23	29	2	0.0469
8(1000)	1	2	0	0.0000	1	1	0	0.0000	1	1	0	0.0000
8(10000)	1	2	0	0.0000	1	1	0	0.0000	1	1	0	0.0000
9(100)	6	12	0	0.0000	6	6	0	0.0000	6	6	0	0.0000
9(1000)	6	12	0	0.3594	6	6	0	0.1875	6	6	0	0.1875
10(100)	1	8	1	0.0000	2	12	1	0.0000	2	12	1	0.0000
10(500)	1	8	1	0.0000	2	12	1	0.0000	2	12	1	0.0000
11(99)	11	34	4	0.0000	17	49	7	0.0156	17	47	6	0.0000
11(399)	11	34	4	0.0000	17	49	7	0.0000	22	72	8	0.0156
12(1000)	6	14	2	0.0156	30	62	12	0.0313	6	6	0	0.0000
12(10000)	5	12	2	0.0625	23	59	11	0.3438	4	4	0	0.0313
13(100)	3	8	1	0.0000	3	7	1	0.0000	3	7	1	0.0000
13(1000)	4	10	1	0.0156	4	8	1	0.0000	4	8	1	0.0000
14(10000)	12	28	1	0.0625	12	20	1	0.0625	12	20	1	0.0469
14(100000)	12	29	1	0.7188	12	22	1	0.5938	12	22	1	0.5781
15(5000)	5	10	0	0.0156	5	5	0	0.0000	5	5	0	0.0000
15(15000)	5	10	0	0.0156	5	5	0	0.0156	5	5	0	0.0156
16(500)	14	29	1	0.0000	14	16	1	0.0000	14	16	1	0.0000
16(2000)	16	32	0	0.0000	16	16	0	0.0000	16	16	0	0.0156
17(100)	9	19	1	0.0000	9	11	1	0.0000	9	11	1	0.0000
17(1000)	7	15	1	0.0156	7	9	1	0.0000	7	9	1	0.0000
18(50)	24	50	2	0.0000	19	21	1	0.0000	302	637	132	0.0000
18(100)	24	49	1	0.0000	*	*	*	*	48	73	9	0.0000
19(1000)	5	10	0	0.0000	5	5	0	0.0000	5	5	0	0.0000
19(50000)	5	10	0	0.1094	5	5	0	0.0781	5	5	0	0.0781
20(100)	32	67	2	0.0000	40	42	1	0.0000	40	42	1	0.0000
20(1000)	51	117	9	0.0313	44	62	5	0.0156	50	54	1	0.0000
21(399)	4	9	1	0.0000	5	7	1	0.0000	5	7	1	0.0000
21(9999)	4	9	1	0.0156	5	7	1	0.0156	5	7	1	0.0156
22(1000)	1	2	0	0.0000	1	2	0	0.0000	1	2	0	0.0000
22(15000)	1	2	0	0.0000	1	2	0	0.0000	1	2	0	0.0000

Tabla 3.2: Resultados para los problemas 23-44

Función(n)	SANE				DF-SANE				NDF-SANE			
	IT	EF	BL	T	IT	EF	BL	T	IT	EF	BL	T
23(500)	1	10	1	0.0000	2	18	1	0.0000	2	18	1	0.0000
23(1000)	1	11	1	0.0000	2	20	1	0.0000	2	20	1	0.0000
24(500)	25	54	4	0.0000	59	99	14	0.0000	122	276	64	0.0156
24(1000)	263	929	161	0.0469	17	25	3	0.0000	17	25	3	0.0000
25(100)	2	6	1	0.0000	2	6	1	0.0000	2	6	1	0.0000
25(500)	3	9	1	0.0000	3	9	1	0.0000	3	9	1	0.0000
26(1000)	1	2	0	0.0000	1	1	0	0.0000	1	1	0	0.0000
26(10000)	1	2	0	0.0000	1	1	0	0.0000	1	1	0	0.0000
27(50)	10	20	0	0.0469	10	10	0	0.0156	10	10	0	0.0313
27(100)	11	22	0	0.2188	11	11	0	0.1250	11	11	0	0.1094
28(100)	1	2	0	0.0000	1	1	0	0.0000	1	1	0	0.0000
28(1000)	1	2	0	0.0000	1	1	0	0.0000	1	1	0	0.0000
29(100)	1	4	1	0.0000	1	5	1	0.0000	1	5	1	0.0000
29(1000)	1	4	1	0.0000	1	5	1	0.0000	1	5	1	0.0000
30(99)	18	39	3	0.0000	11	16	2	0.0000	11	16	2	0.0000
30(9999)	18	39	3	0.0938	11	16	2	0.0313	11	16	2	0.0469
31(1000)	4	9	0	0.0000	6	6	0	0.0156	6	6	0	0.0000
31(5000)	4	9	0	0.0156	6	6	0	0.0000	6	6	0	0.0156
32(500)	6	12	0	0.0000	6	7	0	0.0000	6	7	0	0.0000
32(1000)	6	12	0	0.0000	6	7	0	0.0000	6	7	0	0.0000
33(1000)	3	20	2	0.0000	22	46	6	0.0156	22	23	0	0.0000
33(5000)	3	22	1	0.0313	4	16	2	0.0156	17	25	2	0.0313
34(1000)	22	52	4	0.0000	63	113	20	0.0156	13	24	3	0.0000
34(5000)	12	27	1	0.0156	12	18	1	0.0000	5	13	2	0.0000
35(1000)	21	45	2	0.0000	21	27	2	0.0156	21	27	2	0.0000
35(5000)	29	63	3	0.0313	38	48	3	0.0156	31	41	4	0.0156
36(1000)	21	45	2	0.0000	28	34	2	0.0000	58	68	4	0.0156
36(5000)	44	96	7	0.0469	26	36	4	0.0313	75	116	16	0.0625
37(1000)	25	57	6	0.0000	21	27	2	0.0000	21	27	2	0.0000
37(5000)	25	57	6	0.0156	21	27	2	0.0156	31	43	5	0.0156
38(1000)	19	40	2	0.0156	25	30	2	0.0000	52	60	4	0.0156
38(5000)	19	40	2	0.0469	25	30	2	0.0469	52	60	4	0.0781
39(1000)	55	126	13	0.0000	14	20	1	0.0000	14	18	1	0.0000
39(5000)	55	126	13	0.0469	14	20	1	0.0156	14	20	1	0.0156
40(1000)	1	2	0	0.0000	1	1	0	0.0000	1	1	0	0.0000
40(5000)	1	2	0	0.0156	1	1	0	0.0000	1	1	0	0.0000
41(500)	7	15	1	0.0000	7	9	1	0.0000	11	15	2	0.0000
41(1000)	2	5	1	0.0000	3	3	0	0.0000	3	5	1	0.0000
42(1000)	36	82	9	0.0000	102	198	37	0.0156	44	52	3	0.0156
42(5000)	36	82	9	0.0313	102	198	37	0.0781	44	52	3	0.0156
43(100)	80	175	11	0.0000	86	108	9	0.0000	118	122	2	0.0000
43(500)	462	1569	311	0.1406	482	926	147	0.0781	234	252	9	0.0313
44(1000)	2	4	0	0.0000	4	4	0	0.0156	2	3	0	0.0000
44(5000)	2	4	0	0.0156	3	3	0	0.0156	2	3	0	0.0156

Tabla 3.3: Número de problemas en los que un algoritmo es ganador

Algoritmo	IT	EF	BL	T
SANE	25	12	5	3
DF-SANE	6	13	6	7
NDF-SANE	7	16	24	14
Sin decisión	50	47	53	64

Tabla 3.4: SANE vs. DF-SANE

Algoritmo	IT	EF	BL	T
SANE	34	19	13	14
DF-SANE	15	64	21	25
Empates	39	5	54	49

Tabla 3.5: SANE vs. NDF-SANE

Algoritmo	IT	EF	BL	T
SANE	30	19	14	11
NDF-SANE	16	61	24	27
Empates	42	8	50	50

Tabla 3.6: DF-SANE vs. NDF-SANE

Algoritmo	IT	EF	BL	T
DF-SANE	15	16	15	14
NDF-SANE	17	19	16	21
Empates	56	53	57	53

3.2.2 Trazado de Rayos Sísmicos

El trazado de rayos sísmicos de un emisor a un receptor es un problema sismológico clásico. Un importante uso del trazado de rayos sísmicos es la tomografía del tiempo de viaje que es un problema inverso no lineal, que usualmente se resuelve con el método Gauss-Newton ([19, 20, 29, 13]). Cada iteración de Gauss-Newton requiere de las trayectorias de los rayos y del tiempo de viaje asociado a cada trayectoria. En otras palabras, en cada iteración de Gauss-Newton se realiza un trazado de rayos que consume un elevado tiempo de cómputo dependiendo del trazador de rayos que se utilice. Por ello, es de gran importancia en exploración sísmica desarrollar algoritmos eficientes y de bajo costo para trazar rayos en medios de 3D.

La Cruz [45] desarrolló un método de bajo costo para el Trazado de Rayos Sísmicos donde obtiene la trayectoria del rayo de un emisor fijo a un receptor fijo a través de un sistema de ecuaciones no lineales basado en el principio de Fermat. Específicamente, La Cruz [45] asume que la trayectoria del rayo es una línea quebrada donde determina los puntos de cruce con las interfaces y los puntos entre capas. En otras palabras, realiza una discretización finita del rayo forzando que se satisfaga el principio de Fermat en los trozos del rayo comprendidos entre puntos consecutivos.

A continuación describimos el modelo usado. Consideramos un medio heterogéneo e isotrópico $M \subset \mathbb{R}^3$ que posee m capas delimitadas por $m + 1$ interfaces descritas por las superficies $z = \phi_0(x, y)$, $z = \phi_1(x, y)$, \dots , $z = \phi_m(x, y)$, respectivamente. En cada capa la velocidad de onda se describe como una función continua que no necesariamente es la misma para capas distintas. Supongamos que $\mathbf{x}^e \in \mathbb{R}^3$ es el emisor y $\mathbf{x}^r \in \mathbb{R}^3$ es el receptor. Sea C_e^r el rayo de Fermat (o uno de ellos) que conecta a \mathbf{x}^e con \mathbf{x}^r . El tiempo de viaje a lo largo de C_e^r viene dado por la integral

$$\tau(s) = \int_{C_e^r} s(x, y, z) dl, \quad (3.42)$$

donde dl denota la distancia infinitesimal a lo largo de C_e^r y $s(\mathbf{x})$ es la función lentitud (inversa de la velocidad) evaluada en el punto \mathbf{x} .

Supongamos que la longitud del trozo del rayo C_e^r comprendido entre los puntos consecutivos $(x_{i-1}, y_{i-1}, z_{i-1})^t$ y $(x_i, y_i, z_i)^t$, se aproxima por la longitud del segmento de recta que

une a estos puntos, a saber:

$$l_i = \sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2 + (z_i - z_{i-1})^2}. \quad (3.43)$$

De esta forma, el tiempo de viaje de la ecuación integral (3.42) se puede aproximar por la suma:

$$\mathbf{t}(X) = \sum_{i=1}^{n+1} s_i l_i, \quad (3.44)$$

donde $X = (x_1, \dots, x_{n_x}, y_1, \dots, y_{n_y}, z_1, \dots, z_{n_z})^t \in \mathbb{R}^{n_x+n_y+n_z}$,

$$s_i = s \left(\frac{x_i + x_{i-1}}{2}, \frac{y_i + y_{i-1}}{2}, \frac{z_i + z_{i-1}}{2} \right),$$

$n = n_x + n_y + n_z$, $n_x = n_y = 2(m(N+1)-1)$, $n_z = Nm$, y N es el número de puntos del rayo C_e^r en cada capa sin considerar los puntos de cruces del rayo con las interfaces que confinan la capa. Por el principio de Fermat la dirección del rayo C_e^r tiene tiempo estacionario. Luego, la trayectoria actual del rayo puede determinarse igualando a cero las derivadas parciales de \mathbf{t} de la ecuación (3.44) con respecto a x_i , y_i y z_i , obteniendo de esta forma un sistema de ecuaciones no lineales. A este sistema de ecuaciones no lineales se le deben agregar ecuaciones auxiliares para determinar los puntos de cruce del rayo con cada interface. De esta forma, se obtiene una sistema de ecuaciones no lineales

$$F(X) = 0,$$

donde $F : \mathbb{R}^n \rightarrow \mathbb{R}^n$. Para una descripción más detallada del modelo matemático del trazado de rayos sísmicos se recomienda ver a [45] y [46].

En la Figura 3.1 observamos el medio sintético utilizado para la experimentación. El medio tiene cuatro capas y cinco interfaces definidas como $z = \phi_j(x, y)$, $j = 0, \dots, 4$, donde:

$$\begin{aligned} \phi_0(x, y) &= 0, \\ \phi_1(x, y) &= 80p(x, y) - 750, \\ \phi_2(x, y) &= 150p(x, y) - 1750, \\ \phi_3(x, y) &= 250p(x, y) - 2750, \\ \phi_4(x, y) &= 350p(x, y) - 3500, \end{aligned}$$

y

$$p(x, y) = \exp\left(-\frac{(x - 1750)^2 + (y - 1750)^2}{5 \times 10^5}\right).$$

El medio posee la siguiente distribución de velocidades:

$$v_1(x, y, z) = 0.17x + 0.1y + 0.06z + 0.5,$$

$$v_2(x, y, z) = 0.15x + 0.1y + 0.02z + 0.25,$$

$$v_3(x, y, z) = 0.14x + 0.1y + 0.04z + 0.15,$$

$$v_4(x, y, z) = 0.12x + 0.1y + 0.02z + 0.1.$$

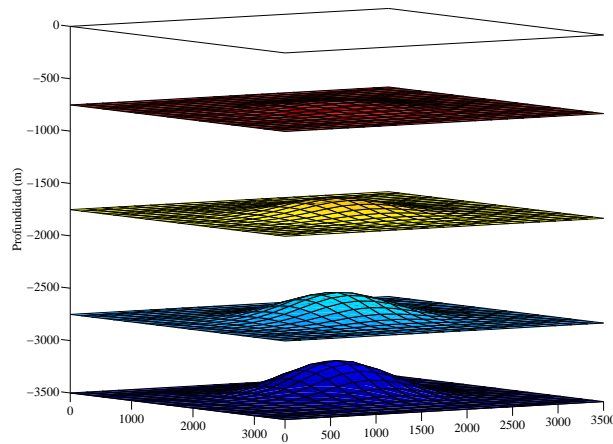


Figura 3.1: Medio Sintético

Todos los experimentos de esta parte se corrieron en un Computador Pentium IV de 3.0 GHz en FORTRAN. Implementamos a SANE, DF-SANE y NDF-SANE con los parámetros que se indican en la sección 3.2.1, excepto el valor de η_k para NDF-SANE que lo tomamos como $\eta_k = \theta(1 - 10^{-10})^k$, donde

$$\theta = \begin{cases} \|F(x_0)\|^2, & \|F(x_0)\|^2 \leq 10^5, \\ 10^4, & \|F(x_0)\|^2 > 10^5. \end{cases}$$

Consideramos dos casos: (a) Emisor y receptor ubicados en distintas capas; y (b) Emisor y receptor ubicados en la superficie (interface 0). Es de resaltar que el Algoritmo DF-SANE falló en todas las corridas realizadas, lo cual coincide con lo reportado por Moreno [58]. En SANE y NDF-SANE detuvimos el proceso utilizando el criterio de parada (3.41).

(a) Emisor y receptor ubicados en distintas capas

Tenemos un emisor $\mathbf{x}^e = (500, 500, -250)^t$ y 25 receptores $\mathbf{x}^r = (x, y, \phi_4(x, y))^t$ en la interface 4, donde el punto $(x, y)^t$ pertenece a la circunferencia de centro $(1750, 1750)^t$ y radio 500. Tomamos $N = 20$ puntos del rayo entre capas.

En la Tabla 3.7 se muestran los promedios en: iteraciones (IT), evaluaciones de la función $F(x)$ (EF), número de ocasiones que se emplea la búsqueda lineal (BL), tiempo de CPU expresado en segundos (T), tiempo de viaje del rayo inicial (TVI) y el tiempo de viaje del rayo final (TVF), para el criterio de parada (3.41) con $e_a = 5 \times 10^{-5}$ y $e_r = 10^{-10}$.

Tabla 3.7: Resultados numéricos para SANE y NDF-SANE en el caso (a)

Algoritmo	IT	EF	BL	T	TVI	TVF
SANE	13	39	8	0.0025	12.00	11.95
NDF-SANE	24	50	10	0.0019	12.00	11.94

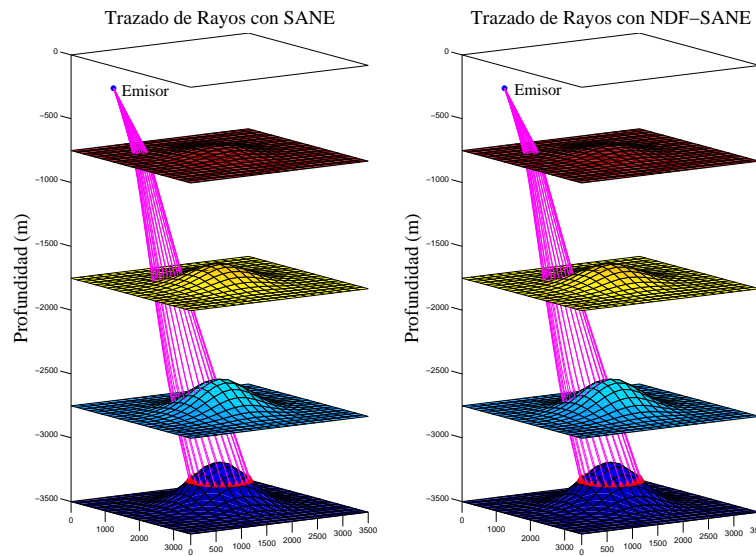


Figura 3.2: Trazado de Rayos para SANE y NDF-SANE en el caso (a)

La Figura 3.2 ilustra el trazado de todos los rayos del disparo. En esta figura se aprecian el trazado de rayo obtenido por cada algoritmo. Cada algoritmo trazó 25 rayos. El algoritmo SANE tardó 0.0625 segundos y realizó un total de 329 iteraciones y 986 evaluaciones de $F(x)$. El algoritmo NDF-SANE tardó 0.0469 segundos y realizó un total de 619 iteraciones y 1251 evaluaciones de $F(x)$.

(b) Emisor y receptor ubicados en la superficie

Tenemos un emisor $\mathbf{x}^e = (0, 0, 0)^t$ y 25 receptores $\mathbf{x}^r = (x, y, 0)^t$ en la superficie, donde los puntos $(x, y)^t$ son generados aleatoriamente en el cuadrado $[250, 3500] \times [250, 3500]$. Tomamos $N = 15$ puntos del rayo entre capas.

Para esta parte es necesario introducir el concepto de rayo tipo i , donde $i \leq m$ es un entero positivo (Ver La Cruz [46]).

Definición 3.3 (Rayo tipo i). Sea C_e^r un rayo que conecta \mathbf{x}^e con \mathbf{x}^r . Decimos que C_e^r es un rayo tipo i , si C_e^r no se intersecta consigo mismo y cruza $i - 1$ interfaces en el siguiente orden: primero cruza las interfaces $1, \dots, i - 1$, luego se refleja en la i -ésima interface y, por último, cruza las interfaces $i - 1, i - 2, \dots, 2, 1$.

En la Tabla 3.8 se muestran para cada tipo de rayo (TR) los promedios en: iteraciones, evaluaciones de la función $F(x)$, número de ocasiones que se emplea la búsqueda lineal, tiempo de CPU en segundos, tiempo de viaje del rayo inicial y el tiempo de viaje del rayo final, para el criterio de parada (3.41) con $e_a = 1.25 \times 10^{-3}$ y $e_r = 10^{-10}$.

La Figura 3.3 ilustra el trazado de todos los rayos del disparo. En esta figura se aprecian el trazado de rayo obtenido por cada algoritmo. Cada algoritmo trazó 100 rayos. El algoritmo SANE tardó 143.1563 segundos y realizó un total de 451393 iteraciones, 7546541 evaluaciones de F y 449914 búsquedas lineales. El algoritmo NDF-SANE tardó 0.0938 segundos y realizó un total de 3111 iteraciones, 3111 evaluaciones de F y cero búsquedas lineales.

Tabla 3.8: Resultados numéricos para SANE y NDF-SANE en el caso (b)

TR	SANE					NDF-SANE					TVI
	IT	EF	BL	T	TVF	IT	EF	BL	T	TVF	
1	18013	301776	17996	5.7225	22.74	81	81	0	0.0019	22.27	30.64
2	16	33	0	0.0019	32.82	17	17	0	0.0006	32.70	43.97
3	13	26	0	0.0012	41.50	13	13	0	0.0006	41.50	55.21
4	12	24	0	0.0006	50.81	12	12	0	0.0006	50.81	67.21

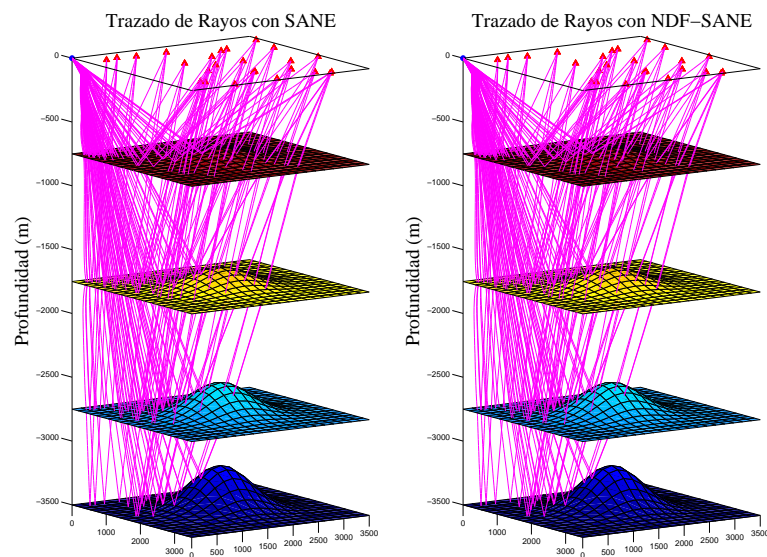


Figura 3.3: Trazado de Rayos para SANE y NDF-SANE en el caso (b)

Conclusiones y Trabajos Futuros

*“Aún cuando el error de redondeo desapareciera
el Análisis Numérico permanecería”*

Lloyd Trefethen

Introducimos una nueva técnica de globalización para minimización irrestricta. Particularmente, diseñamos una búsqueda lineal no monótona que puede interpretarse como una perturbación de la condición de Armijo. En el Teorema 2.1 establecemos las condiciones que deben satisfacer las direcciones de descenso para que la búsqueda lineal garantice la convergencia global. Las direcciones tipo Newton y tipo Gradiente satisfacen las condiciones del Teorema 2.1, en otras palabras, estas direcciones están uniformemente relacionadas con el gradiente. Gracias a esta propiedad, construimos dos versiones globales de los métodos Newton y Barzilai-Borwein [4].

En base a nuestros resultados podemos concluir que la nueva técnica de búsqueda lineal al incorporarla a los métodos Newton y Barzilai-Borwein [4], puede producir un sustancial ahorro en el número de evaluaciones de la función mérito y el número de iteraciones. Cuando la nueva técnica de búsqueda lineal se incorpora al método de Newton tiene mejor desempeño que la combinación Newton-Armijo. Quizás esto se deba a que la técnica propuesta se emplea más al inicio y en el intermedio de la ejecución del algoritmo, lo cual posiblemente mejora la eficiencia del método.

Con respecto a la condición GLL (1.8), la nueva búsqueda lineal puede comportarse generalmente mejor cuando los iterados están muy cerca de la solución del problema (1.3). Explícitamente, en algunos casos cuando los iterados x_k se encuentran dentro de una vecindad muy pequeña de la solución del problema (1.3), el factor $\max_{N \leq j \leq M} [f(x_{k-j})]$ tiende a ser constante y la condición (1.8) es muy parecida a la regla de Armijo, lo cual no encaja

muy bien con métodos tipo Gradiente, específicamente, con el método de Barzilai-Borwein [4] (caso cuadrático), que es no monótono para funciones estrictamente convexas.

En cuanto a la efectividad de NGBB en la solución del problema de Conformación Molecular, constatamos que, para el potencial de energía empleado, NGBB presenta una mejora sustancial en el número de evaluaciones de la función y el tiempo de CPU, con respecto a MatLab (`fminunc`) y GBB.

También se pudo constatar numéricamente que la escogencia de la sucesión $\{\eta_k\}_{k \in \mathbb{N}}$ es de vital importancia para el buen desempeño de la búsqueda lineal. Este hecho y los resultados obtenidos inducen a profundizar más en el estudio de la búsqueda lineal propuesta; en particular, en diseñar un procedimiento para escoger o definir la sucesión $\{\eta_k\}_{k \in \mathbb{N}}$.

En cuanto a sistemas de ecuaciones no lineales, diseñamos un nuevo algoritmo libre de derivadas, que denominamos NDF-SANE, que puede considerarse como un método Casi-Newton para la resolución de sistemas de ecuaciones no lineales. Para NDF-SANE establecimos los siguientes resultados de convergencia:

1. En todo punto límite de la sucesión $\{x_k\}_{k \in \mathbb{N}}$ generada por NDF-SANE, el gradiente de $\|F(x)\|^2$ es ortogonal al residual $F(x)$.
2. Si algún punto límite de la sucesión $\{x_k\}_{k \in \mathbb{N}}$ generada por NDF-SANE es una solución de (1.1), entonces todo punto límite es una solución.
3. Si el punto inicial x_0 se escoge muy cerca de alguna solución fuertemente aislada x_* , entonces la sucesión generada por NDF-SANE converge a x_* .

Igual que los algoritmos SANE y DF-SANE, NDF-SANE emplea sistemáticamente la dirección $\pm F(x)$. La diferencia sustancial entre estos algoritmos es la manera de escoger el tamaño de paso λ_k . Nuestros resultados numéricos indican que la forma de escoger a λ_k en NDF-SANE, es generalmente más eficiente que las utilizadas por SANE y DF-SANE. Esto se evidencia en las pruebas numéricas realizadas con el extenso conjunto de funciones test. En estas pruebas NDF-SANE obtuvo con mayor regularidad el menor número de evaluaciones de la función y el menor tiempo de CPU; además, empleó menos ocasiones la búsqueda lineal.

Quizás, el no emplear con cierta frecuencia la búsqueda lineal origine que el comportamiento NDF-SANE sea mejor. Esto es un tema que amerita ser estudiado con profundidad.

En cuanto al problema del Trazado de Rayos Sísmicos comprobamos, para el modelo sintético utilizado, que NDF-SANE compite satisfactoriamente con SANE y DF-SANE. Concretamente, NDF-SANE obtuvo en promedio el menor número de evaluaciones de la función y el menor tiempo de CPU. Un caso curioso, DF-SANE falló en todas las corridas, lo cual coincide con lo reportado por Moreno [58]. Este comportamiento de DF-SANE quizás se deba a las características del sistema de ecuaciones no lineales asociado al trazado de rayos. La Cruz [46] demostró que este sistema de ecuaciones no lineales posee infinitas soluciones y ninguna de ellas es aislada. Por lo tanto, la búsqueda lineal que emplea DF-SANE parece no encajar bien con el problema del trazado de rayos sísmicos. Hacemos esta afirmación, ya que el comportamiento de NDF-SANE para el trazado de rayos sísmicos es satisfactorio y la gran diferencia entre DF-SANE y NDF-SANE es que utilizan búsquedas lineales distintas.

Finalmente, como trabajos futuros de gran importancia podemos enumerar los siguientes:

- *Diseñar un procedimiento para la escogencia de la sucesión $\{\eta\}_{k \in \mathbb{N}}$ para la búsqueda lineal (2.1).*
- *Diseñar versiones preconditionadas de los algoritmos SANE, DF-SANE y NDF-SANE.*
- *Diseñar algoritmos híbridos para minimización irrestricta y sistemas de ecuaciones no lineales, que combinen algoritmos evolutivos y los algoritmos aquí diseñados.*

Apéndice A

Funciones Test

A continuación enumeramos las funciones

$$F(x) = (f_1(x), f_2(x), \dots, f_n(x))^t,$$

donde $f_i : \mathbb{R}^n \rightarrow \mathbb{R}$; y los correspondientes iterados iniciales x_0 .

1. Función exponencial 1 [45]

$$\begin{aligned} f_1(x) &= e^{x_1-1} - 1, \\ f_i(x) &= i(e^{x_i-1} - x_i), \text{ para } i = 2, 3, \dots, n. \end{aligned}$$

$$\text{Iterado inicial: } x_0 = \left(\frac{n}{n-1}, \frac{n}{n-1}, \dots, \frac{n}{n-1} \right)^t.$$

2. Función exponencial 2 [45]

$$\begin{aligned} f_1(x) &= e^{x_1} - 1, \\ f_i(x) &= \frac{i}{10}(e^{x_i} + x_{i-1} - 1), \text{ para } i = 2, 3, \dots, n. \end{aligned}$$

$$\text{Iterado inicial: } x_0 = \left(\frac{1}{n}, \frac{1}{n}, \dots, \frac{1}{n} \right)^t.$$

3. Función exponencial 3 [45]

$$\begin{aligned} f_i(x) &= \frac{i}{10}(1 - x_i^2 - e^{-x_i^2}), \text{ para } i = 2, 3, \dots, n-1 \\ f_n(x) &= \frac{n}{10}(1 - e^{-x_n^2}). \end{aligned}$$

$$\text{Iterado inicial: } x_0 = \left(\frac{1}{2n}, \frac{2}{2n}, \dots, \frac{n}{2n} \right)^t.$$

4. *Función diagonal de tres variables premultiplicada por una matriz ortogonal* (n múltiplo de 3) [35]

Para $i = 1, 2, \dots, n/3$,

$$\begin{aligned} f_{3i-2}(x) &= 0.6x_{3i-2} + 1.6x_{3i-2}^3 - 7.2x_{3i-1}^2 + 9.6x_{3i-1} - 4.8, \\ f_{3i-1}(x) &= 0.48x_{3i-2} - 0.72x_{3i-1}^3 + 3.24x_{3i-1}^2 - 4.32x_{3i-1} - x_{3i} + 0.2x_{3i}^3 + 2.16, \\ f_{3i}(x) &= 1.25x_{3i} - 0.25x_{3i}^3. \end{aligned}$$

Iterado inicial: $x_0 = (-1, 0.5, -1, -1, 0.5, -1, \dots)^t$.

5. *Problema de valores en el borde en dos puntos* [49]

Corresponde al sistema de diferencias finita

$$\begin{aligned} f_1(x) &= 2x_1 - x_2 + \frac{1}{(n+1)^2} (\operatorname{atan} x_1 - 1), \\ f_i(x) &= -x_{i-1} + 2x_i - x_{i+1} + \frac{1}{(n+1)^2} (\operatorname{atan} x_i - 1), \text{ para } i = 2, \dots, n-1 \\ f_n(x) &= -x_{n-1} + 2x_n + \frac{1}{(n+1)^2} (\operatorname{atan} x_n - 1), \end{aligned}$$

asociado al problema de valores en el borde en dos puntos

$$u'' = -\operatorname{atan} u + 1,$$

sobre el intervalo unitario con condiciones de borde $u(0) = 0$ y $u(1) = 0$. Usamos diferencias centrales sobre una rejilla uniforme de n nodos internos.

Iterado inicial: $x_0 = \left(\frac{n}{n}, \frac{n-1}{n}, \dots, \frac{1}{n}\right)^t$.

6. *Función extendida de Rosenbrock* (n par) [35]

Para $i = 1, 2, \dots, n/2$,

$$\begin{aligned} f_{2i-1}(x) &= 10(x_{2i} - x_{2i-1}^2), \\ f_{2i}(x) &= 1 - x_{2i-1}. \end{aligned}$$

Iterado inicial: $x_0 = (5, 1, 5, 1, \dots)^t$.

7. *Función modificada de Rosenbrock* (n par) [32]

Para $i = 1, 2, \dots, n/2$,

$$\begin{aligned} f_{2i-1}(x) &= \frac{1}{1 + \exp(-x_{2i-1})} - 0.73, \\ f_{2i}(x) &= 10(x_{2i} - x_{2i-1}^2). \end{aligned}$$

Iterado inicial: $x_0 = (0.95, 0.95, \dots)^t$.

8. Función aumentada de Rosenbrock (n múltiplo de 4) [32]

Para $i = 1, 2, \dots, n/4$,

$$\begin{aligned} f_{4i-3}(x) &= 10(x_{4i-2} - x_{4i-3}^2), \\ f_{4i-2}(x) &= 1 - x_{4i-3}, \\ f_{4i-1}(x) &= 1.25x_{4i-1} - 0.25x_{4i-1}^3, \\ f_{4i}(x) &= x_{4i}. \end{aligned}$$

Iterado inicial: $x_0 = (-1.2, 1, -1, 20, -1.2, 1, -1, 20, \dots)^t$.

9. Ecuación-H de Chandrasekhar [44]

La ecuación-H de Chandrasekhar es la discretización de

$$F(H)(\mu) = H(\mu) - \left(1 - \frac{c}{2} \int_0^1 \frac{\mu H(\nu)}{\mu + \nu} d\nu\right)^{-1} = 0,$$

empleando la regla del punto medio para aproximar la integral entre 0 y 1.

$$f_i(x) = x_i - \left(1 - \frac{c}{2n} \sum_{j=1}^n \frac{\mu_i x_j}{\mu_i + \mu_j}\right)^{-1},$$

donde $c \in [0, 1)$ y $\mu_i = (i - 1/2)/n$, para $1 \leq i \leq n$. (Para el experimento numérico se tomó $c = 0.9$).

Iterado inicial: $x_0 = (1, 1, \dots, 1)^t$.

10. Función mala de Powell (n par) [32]

Para $i = 1, 2, \dots, n/2$,

$$\begin{aligned} f_{2i-1}(x) &= 10^4 x_{2i-1} x_{2i} - 1, \\ f_{2i}(x) &= \exp(-x_{2i-1}) + \exp(-x_{2i}) - 1.0001. \end{aligned}$$

Iterado inicial: $x_0 = (0, 10, 0, 10, \dots)^t$.

11. Función mala de Powell aumentada (n múltiplo de 3) [35]

Para $i = 1, 2, \dots, n/3$,

$$\begin{aligned} f_{3i-2}(x) &= 10^4 x_{3i-2} x_{3i-1} - 1, \\ f_{3i-1}(x) &= \exp(-x_{3i-2}) + \exp(-x_{3i-1}) - 1.0001, \\ f_{3i}(x) &= \phi(x_{3i}), \end{aligned}$$

donde

$$\phi(t) = \begin{cases} 0.5t - 2 & t \leq -1 \\ (-592t^3 + 888t^2 + 4551t - 1924) / 1998 & -1 < t < 2 \\ 0.5t + 2 & t \geq 2. \end{cases}$$

Iterado inicial: $x_0 = (10^{-3}, 18, 1, 10^{-3}, 18, 1, \dots)^t$.

12. Función Trigonométrica

$$f_i(x) = 2 \left(n + i(1 - \cos x_i) - \operatorname{sen} x_i - \sum_{j=1}^n \cos x_j \right) (2 \operatorname{sen} x_i - \cos x_i).$$

$$\text{Iterado inicial: } x_0 = \left(\frac{101}{100n}, \dots, \frac{101}{100n} \right)^t.$$

13. Función trigonométrica desplazada y aumentada en una esfera Euclidea [32]

$$f_i(x) = n - 1 - \sum_{j=1}^{n-1} \cos(x_j - 1) + i(1 - \cos(x_i - 1)) - \operatorname{sen}(x_i - 1),$$

para $i = 1, \dots, n - 1,$

$$f_n(x) = \sum_{j=1}^n x_j^2 - 10000.$$

$$\text{Iterado inicial: } x_0 = \left(\frac{n}{n+1}, \dots, \frac{n}{n+1} \right)^t.$$

14. Función Singular [45]

$$f_1(x) = \frac{1}{3}x_1^3 + \frac{1}{2}x_2^2$$

$$f_i(x) = -\frac{1}{2}x_i^2 + \frac{i}{3}x_i^3 + \frac{1}{2}x_{i+1}^2, \text{ para } i = 2, 3, \dots, n - 1,$$

$$f_n(x) = -\frac{1}{2}x_n^2 + \frac{n}{3}x_n^3.$$

$$\text{Iterado inicial: } x_0 = (1, 1, \dots, 1)^t.$$

15. Función Logarítmica [45]

$$f_i(x) = \ln(x_i + 1) - \frac{x_i}{n}.$$

$$\text{Iterado inicial: } x_0 = (1, 1, \dots, 1)^t.$$

16. Función de Broyden Tridiagonal [39]

$$f_1(x) = (3 - 0.5x_1)x_1 - 2x_2 + 1,$$

$$f_i(x) = (3 - 0.5x_i)x_i - x_{i-1} - 2x_{i+1} + 1, \text{ para } i = 2, 3, \dots, n - 1,$$

$$f_n(x) = (3 - 0.5x_n)x_n - x_{n-1} + 1.$$

$$\text{Iterado inicial: } x_0 = (-1, -1, \dots, -1)^t.$$

17. Función Trigexp [39]

$$f_1(x) = 3x_1^3 + 2x_2 - 5 + \operatorname{sen}(x_1 - x_2) \operatorname{sen}(x_1 + x_2),$$

$$f_i(x) = -x_{i-1}e^{(x_{i-1}-x_i)} + x_i(4 + 3x_i^2) + 2x_{i+1} + \operatorname{sen}(x_i - x_{i+1}) \operatorname{sen}(x_i + x_{i+1}) - 8, \text{ para } i = 2, 3, \dots, n - 1,$$

$$f_n(x) = -x_{n-1}e^{(x_{n-1}-x_n)} + 4x_n - 3.$$

$$\text{Iterado inicial: } x_0 = (0, 0, \dots, 0)^t.$$

18. Función 15 [39]

$$\begin{aligned}
 f_1(x) &= -2x_1^2 + 3x_1 + 3x_{n-4} - x_{n-3} - x_{n-2} + 0.5x_{n-1} - x_n + 1, \\
 f_i(x) &= -2x_i^2 + 3x_i - x_{i-1} - 2x_{i+1} + 3x_{n-4} - x_{n-3} - x_{n-2} + 0.5x_{n-1} \\
 &\quad - x_n + 1, \text{ para } i = 2, 3, \dots, n-1, \\
 f_n(x) &= -2x_n^2 + 3x_n - x_{n-1} + 3x_{n-4} - x_{n-3} - x_{n-2} + 0.5x_{n-1} - x_n + 1.
 \end{aligned}$$

Iterado inicial: $x_0 = (0, 0, \dots, 0)^t$.

19. Función estrictamente convexa 1 [62]

F es el gradiente de la función estrictamente convexa $h(x) = \sum_{i=1}^n (e^{x_i} - x_i)$.

$$f_i(x) = e^{x_i} - 1.$$

Iterado inicial: $x_0 = \left(\frac{1}{n}, \frac{2}{n}, \dots, 1\right)^t$.

20. Función estrictamente convexa 2 [62]

F es el gradiente de la función estrictamente convexa $h(x) = \sum_{i=1}^n \frac{i}{10} (e^{x_i} - x_i)$.

$$f_i(x) = \frac{i}{10} (e^{x_i} - 1).$$

Iterado inicial: $x_0 = (1, 1, \dots, 1)^t$.

21. Función 18 (n múltiplo de 3)

Para $i = 1, 2, \dots, n/3$:

$$\begin{aligned}
 f_{3i-2}(x) &= x_{3i-2}x_{3i-1} - x_{3i}^2 - 1, \\
 f_{3i-1}(x) &= x_{3i-2}x_{3i-1}x_{3i} - x_{3i-2}^2 + x_{3i-1}^2 - 2, \\
 f_{3i}(x) &= e^{-x_{3i-2}} - e^{-x_{3i-1}}.
 \end{aligned}$$

Iterado inicial: $x_0 = (0, 0, \dots, 0)^t$.

22. Función lineal de rango full

$$f_i(x) = x_i - (2/n) \sum_{j=1}^n x_j + 1.$$

Iterado inicial: $x_0 = (100, 100, \dots, 100)^t$.

23. Función lineal de rango 2

$$\begin{aligned}
 f_1(x) &= x_1 - 1, \\
 f_i(x) &= i \sum_{j=1}^n jx_j - i, \text{ para } i = 2, 3, \dots, n.
 \end{aligned}$$

Iterado inicial: $x_0 = \left(1, \frac{1}{n}, \dots, \frac{1}{n}\right)^t$.

24. Función penalidad I

$$f_i(x) = \sqrt{10^{-5}}(x_i - 1), \text{ para } i = 1, 2, \dots, n - 1,$$

$$f_n(x) = \left(\frac{1}{4n}\right) \sum_{j=1}^n x_j^2 - \frac{1}{4}.$$

$$\text{Iterado inicial: } x_0 = \left(\frac{1}{3}, \frac{1}{3}, \dots, \frac{1}{3}\right)^t.$$

25. Función casi morena

$$f_i(x) = x_i + \sum_{j=1}^n x_j - (n + 1), \text{ para } i = 1, 2, \dots, n - 1,$$

$$f_n(x) = \prod_{j=1}^n x_j - 1.$$

$$\text{Iterado inicial: } x_0 = \left(1 - \frac{1}{n}, 1 - \frac{2}{n}, \dots, 0\right)^t.$$

26. Función de dimensión variable

$$f_i(x) = x_i - 1, \text{ para } i = 1, 2, \dots, n - 2,$$

$$f_{n-1}(x) = \sum_{j=1}^{n-2} j(x_j - 1),$$

$$f_n(x) = \left(\sum_{j=1}^{n-2} j(x_j - 1)\right)^2.$$

$$\text{Iterado inicial: } x_0 = \left(1 - \frac{1}{n}, 1 - \frac{2}{n}, \dots, 0\right)^t.$$

27. Función geométrica [45]

$$f_i(x) = \sum_{t=1}^5 \left((t/5)x_i^{(t/5-1)} \prod_{k=1, k \neq i}^n x_k^{t/5} \right).$$

$$\text{Iterado inicial: } x_0 = (1, 1, \dots, 1)^t.$$

28. Función singular de Powell extendida (n múltiplo de 4) [57]

Para $i = 1, 2, \dots, n/4$,

$$f_{4i-3}(x) = x_{4i-3} + 10x_{4i-2},$$

$$f_{4i-2}(x) = \sqrt{5}(x_{4i-1} - x_{4i}),$$

$$f_{4i-1}(x) = (x_{4i-2} - 2x_{4i-1})^2,$$

$$f_{4i}(x) = \sqrt{10}(x_{4i-3} - x_{4i})^2.$$

$$\text{Iterado inicial: } x_0 = (7.15 \times 10^{-5}, 7.15 \times 10^{-5}, \dots, 7.15 \times 10^{-5})^t.$$

29. Función 27

$$f_1(x) = \sum_{j=1}^n x_j^2,$$

$$f_i(x) = -2x_1x_i, \text{ para } i = 2, 3, \dots, n.$$

Iterado inicial: $x_0 = \left(100, \frac{1}{n^2}, \dots, \frac{1}{n^2}\right)^t$.

30. Función de valle tridimensional (n múltiplo de 3) [32]

Para $i = 1, 2, \dots, n/3$:

$$f_{3i-2}(x) = (c_2x_{3i-2}^3 + c_1x_{3i-2}) \exp\left(\frac{-x_{3i-2}^2}{100}\right) - 1,$$

$$f_{3i-1}(x) = 10(\text{sen}(x_{3i-2}) - x_{3i-1}),$$

$$f_{3i}(x) = 10(\text{cos}(x_{3i-2}) - x_{3i}),$$

donde

$$c_1 = 1.003344481605351$$

$$c_2 = -3.344481605351171 \times 10^{-3}$$

Iterado inicial: $x_0 = (-4, 1, 2, 1, 2, \dots)^t$.

31. Función complementaria (n par) [45]

Para $i = 1, 2, \dots, n/2$,

$$f_{2i-1}(x) = \left(x_{2i-1}^2 + \left(x_{2i-1}e^{x_{2i-1}} - \frac{1}{n}\right)^2\right)^{1/2} - x_{2i-1} - x_{2i-1}e^{x_{2i-1}} + \frac{1}{n},$$

$$f_{2i}(x) = \left(x_{2i}^2 + (3x_i + \text{sen}(x_{2i}) + e^{x_{2i}})^2\right)^{1/2} - x_{2i} - 3x_{2i} - \text{sen}(x_{2i}) - e^{x_{2i}}.$$

Iterado inicial: $x_0 = (1, \dots, 1)^t$.

32. Función mínima [45]

$$f_i(x) = \frac{(\ln x_i + \exp(x_i)) - \sqrt{(\ln x_i - \exp(x_i))^2 + 10^{-10}}}{2}.$$

Iterado inicial: $x_0 = (0.5, \dots, 0.5)^t$.

33. Función Guía [45]

$$f_i(x) = 0.05(x_i - 1) + 2 \text{sen} \left(\sum_{j=1}^n (x_j - 1) + \sum_{j=1}^n (x_j - 1)^2 \right) (1 + 2(x_i - 1))$$

$$+ 2 \text{sen} \left(\sum_{j=1}^n (x_j - 1) \right).$$

Iterado inicial: $x_0 = (5, \dots, 5)^t$.

34. Sistema tridiagonal [51]

$$\begin{aligned}
f_1(x) &= 4(x_1 - x_2^2), \\
f_i(x) &= 8x_i(x_i^2 - x_{i-1}) - 2(1 - x_i) + 4(x_i - x_{i+1}^2), \text{ para } i = 2, \dots, n-1 \\
f_n(x) &= 8x_n(x_n^2 - x_{n-1}) - 2(1 - x_n).
\end{aligned}$$

Iterado inicial: $x_0 = (6, \dots, 6)^t$.

35. Sistema cinco-diagonal [51]

$$\begin{aligned}
f_1(x) &= 4(x_1 - x_2^2) + x_2 - x_3^2, \\
f_2(x) &= 8x_2(x_2^2 - x_1) - 2(1 - x_2) + 4(x_2 - x_3^2) + x_3 - x_4^2, \\
f_i(x) &= 8x_i(x_i^2 - x_{i-1}) - 2(1 - x_i) + 4(x_i - x_{i+1}^2) + x_{i-1}^2 - x_{i-2} \\
&\quad + x_{i+1} - x_{i+2}^2, \text{ para } i = 3, \dots, n-2, \\
f_{n-1}(x) &= 8x_{n-1}(x_{n-1}^2 - x_{n-2}) - 2(1 - x_{n-1}) + 4(x_{n-1} - x_n^2) \\
&\quad + x_{n-2}^2 - x_{n-3}, \\
f_n(x) &= 8x_n(x_n^2 - x_{n-1}) - 2(1 - x_n) + x_{n-1}^2 - x_{n-2}.
\end{aligned}$$

Iterado inicial: $x_0 = (-5, \dots, -5)^t$.

36. Sistema siete-diagonal [51]

$$\begin{aligned}
f_1(x) &= 4(x_1 - x_2^2) + x_2 - x_3^2 + x_3 - x_4^2, \\
f_2(x) &= 8x_2(x_2^2 - x_1) - 2(1 - x_2) + 4(x_2 - x_3^2) + x_1^2 + x_3 - x_4^2 + x_4 - x_5^2, \\
f_3(x) &= 8x_3(x_3^2 - x_2) - 2(1 - x_3) + 4(x_3 - x_4^2) + x_2^2 - x_1 + x_4 - x_5^2 + x_1^2 \\
&\quad + x_5 - x_6^2, \\
f_i(x) &= 8x_i(x_i^2 - x_{i-1}) - 2(1 - x_i) + 4(x_i - x_{i+1}^2) + x_{i-1}^2 - x_{i-2} + x_{i+1} \\
&\quad - x_{i+2}^2 + x_{i-2}^2 + x_{i+2} - x_{i-3} - x_{i+3}^2, \text{ para } i = 4, \dots, n-3, \\
f_{n-2}(x) &= 8x_{n-2}(x_{n-2}^2 - x_{n-3}) - 2(1 - x_{n-2}) + 4(x_{n-2} - x_{n-1}^2) + x_{n-3}^2 \\
&\quad - x_{n-4} + x_{n-1} - x_n^2 + x_{n-4}^2 + x_n - x_{n-5}, \\
f_{n-1}(x) &= 8x_{n-1}(x_{n-1}^2 - x_{n-2}) - 2(1 - x_{n-1}) + 4(x_{n-1} - x_n^2) + x_{n-2}^2 - x_{n-3} \\
&\quad + x_n + x_{n-3}^2 - x_{n-4}, \\
f_n(x) &= 8x_n(x_n^2 - x_{n-1}) - 2(1 - x_n) + x_{n-1}^2 - x_{n-2} + x_{n-2}^2 - x_{n-3}.
\end{aligned}$$

Iterado inicial: $x_0 = (-6, \dots, -6)^t$.

37. Función extendida de Freudenstein y Roth (n par) [8]

Para $i = 1, 2, \dots, n/2$,

$$\begin{aligned}
f_{2i-1}(x) &= x_{2i-1} + ((5 - x_{2i})x_{2i} - 2)x_{2i} - 13, \\
f_{2i}(x) &= x_{2i-1} + ((x_{2i} + 1))x_{2i} - 14)x_{2i} - 29.
\end{aligned}$$

Iterado inicial: $x_0 = (9, 6, 9, 6, \dots)^t$.

38. Problema extendido de Cragg y Levy (n múltiplo de 4) [57]Para $i = 1, 2, \dots, n/4$,

$$\begin{aligned} f_{4i-3}(x) &= (\exp(x_{4i-3}) - x_{4i-2})^2, \\ f_{4i-2}(x) &= 10(x_{4i-2} - x_{4i-1})^3, \\ f_{4i-1}(x) &= \tan^2(x_{4i-1} - x_{4i}), \\ f_{4i}(x) &= x_{4i} - 1. \end{aligned}$$

Iterado inicial: $x_0 = (4, 2, 2, 2, 4, 2, 2, 2, \dots)^t$.**39. Problema extendido de Wood** (n múltiplo de 4) [42]Para $i = 1, 2, \dots, n/4$,

$$\begin{aligned} f_{4i-3}(x) &= -200x_{4i-3} (x_{4i-2} - x_{4i-3}^2) - (1 - x_{4i-3}), \\ f_{4i-2}(x) &= 200 (x_{4i-2} - x_{4i-3}^2) + 20(x_{4i-2} - 1) + 19.8(x_{4i} - 1), \\ f_{4i-1}(x) &= -180x_{4i-1} (x_{4i} - x_{4i-1}^2) - (1 - x_{4i-1}), \\ f_{4i}(x) &= 180 (x_{4i} - x_{4i-1}^2) + 20.2(x_{4i} - 1) + 19.8(x_{4i-2} - 1). \end{aligned}$$

Iterado inicial: $x_0 = (0, \dots, 0)^t$.**40. Problema tridiagonal exponencial** [8]

$$\begin{aligned} f_1(x) &= x_1 - \exp(\cos(h(x_1 + x_2))), \\ f_i(x) &= x_i - \exp(\cos(h(x_{i-1} + x_i + x_{i+1}))), \text{ para } i = 2, \dots, n-1 \\ f_n(x) &= x_n - \exp(\cos(h(x_{n-1} + x_n))), \\ h &= 1/(n+1). \end{aligned}$$

Iterado inicial: $x_0 = (1.5, \dots, 1.5)^t$.**41. Problema discreto de valores en el borde** [57]

$$\begin{aligned} f_1(x) &= 2x_1 + 0.5h^2(x_1 + h)^3 - x_2, \\ f_i(x) &= 2x_i + 0.5h^2(x_i + hi)^3 - x_{i-1} + x_{i+1}, \text{ para } i = 2, \dots, n-1 \\ f_n(x) &= 2x_n + 0.5h^2(x_n + hn)^3 - x_{n-1}, \\ h &= 1/(n+1). \end{aligned}$$

Iterado inicial: $x_0 = (h(h-1), h(2h-1), \dots, h(nh-1))^t$.**42. Problema de Brent** [2]

$$\begin{aligned} f_1(x) &= 3x_1(x_2 - 2x_1) + x_2^2/4, \\ f_i(x) &= 3x_i(x_{i+1} - 2x_i + x_{i-1}) + (x_{i+1} - x_{i-1})^2/4, \text{ para } i = 2, \dots, n-1 \\ f_n(x) &= 3x_n(20 - 2x_n + x_{n-1}) + (20 - x_{n-1})^2/4, \\ h &= 1/(n+1). \end{aligned}$$

Iterado inicial: $x_0 = (0, \dots, 0, 20, 20)^t$.

43. Problema de Troesch [63]

$$\begin{aligned}
 f_1(x) &= 2x_1 + \rho h^2 \sinh(\rho x_1) - x_2, \\
 f_i(x) &= 2x_i + \rho h^2 \sinh(\rho x_i) - x_{i-1} - x_{i+1}, \text{ para } i = 2, \dots, n-1 \\
 f_n(x) &= 2x_n + \rho h^2 \sinh(\rho x_n) - x_{n-1}, \\
 \rho &= 10, \quad h = 1/(n+1).
 \end{aligned}$$

Iterado inicial: $x_0 = (2, \dots, 2)^t$.

44. Sistema trigonométrico [68]

$$\begin{aligned}
 f_i(x) &= 5 - (l+1)(1 - \cos x_i) - \operatorname{sen} x_i - \sum_{j=5l+1}^{5l+5} \cos x_j, \\
 l &= \operatorname{div}(i-1, 5).
 \end{aligned}$$

Iterado inicial: $x_0 = \left(\frac{1}{n}, \dots, \frac{1}{n}\right)^t$.

Bibliografía

- [1] ALEFELD, G., GIENGER, A., AND POTRA, F. Efficient validation of solutions of nonlinear systems. *SIAM Journal on Numerical Analysis* 31 (1994), 252–260.
- [2] ALEFELD, G., GIENGER, A., AND POTRA, F. Efficient validation of solutions of nonlinear systems. *SIAM Journal on Numerical Analysis* 31 (1994), 252–260.
- [3] ARMIJO, L. Minimization of functions having Lipschitz-continuous first partial derivatives. *Pacific J. Math.* 16 (1966), 1–3.
- [4] BARZILAI, J., AND BORWEIN, J. Two point step size gradient methods. *IMA J. Numer. Anal.* 8 (1988), 141–148.
- [5] BELLAVIA, S., AND MORINI, B. A globally convergent Newton-GMRES subspace method for systems of nonlinear equations. *SIAM J. Sci.* 23 (2001), 940–960.
- [6] BELLAVIA, S., AND MORINI, B. A globally convergent Newton-GMRES subspace method for systems of nonlinear equations. *SIAM J. Sci. Comput.* 23 (2001), 940–960.
- [7] BIELSCHOWSKY, R. H., FRIEDLANDER, A., GOMES, F. A. M., MARTÍNEZ, J. M., AND RAYDÁN, M. An adaptive algorithm for bound constrained quadratic minimization. *Investigación Operativa* 7 (1997), 67–102.
- [8] BING, Y., AND LIN, G. An efficient implementation of Merrill’s method for sparse or partially separable systems of nonlinear equations. *SIAM Journal on Optimization* 2 (1991), 206–221.
- [9] BING, Y., AND LIN, G. An efficient implementation of Merrill’s method for sparse or partially separable systems of nonlinear equations. *SIAM J. Optim.* 2 (1997), 206–221.

-
- [10] BIRGIN, E. G., CHAMBOULEYRON, I., AND MARTÍNEZ, J. M. Estimation of the optical constants and the thickness of thin films using unconstrained optimization. *Journal of Computational Physics* 151 (1999), 862–880.
- [11] BIRGIN, E. G., AND EVTUSHENKO, Y. G. Automatic differential and spectral projected gradient methods for optimal control problems. *Optimization Methods and Software* 10 (1998), 125–146.
- [12] BIRGIN, E. G., MARTÍNEZ, J. M., AND RAYDÁN, M. Nonmonotone spectral projected gradient methods on convex sets. *SIAM Journal on Optimization* 10, 4 (2000), 1196–1211.
- [13] BISHOP, T. N., BUBE, K. P., CUTLER, R. T., LAGAN, R. T., LOVE, P. L., RESNICK, J. R., SHUEY, R. T., SPINDLER, D. A., AND WYLD, H. W. Tomographic determination of velocity and depth in laterally varying media. *Geophysics* 50 (1985), 903–923.
- [14] BORG, I., AND GROENEN, P. J. F. *Modern Multidimensional Scaling: Theory and Applications*. Springer-Verlag, Berlin, 1997.
- [15] BROWN, P., AND SAAD, Y. Hybrid Krylov methods for nonlinear systems of equations. *SIAM Journal on Scientific Computing* 11 (1990), 450–481.
- [16] BROWN, P., AND SAAD, Y. Convergence theory of nonlinear Newton-Krylov algorithms. *SIAM Journal on Optimization* 4 (1994), 297–330.
- [17] BROYDEN, C. G. A class of the methods for solving nonlinear simultaneous equations. *Math. Comput.* 25 (1965), 577–593.
- [18] CASTILLO, Z., CORES, D., AND RAYDÁN, M. Low cost optimization techniques for solving the nonlinear seismic reflection tomography problem. *Optimization and Engineering* 1 (2000), 155–169.

- [19] CHIU, S. K. L., KANASEWICH, E. R., AND PHADKE, S. Three dimensional determination of structure and velocity by seismic tomography. *Geophysics* 51 (1986), 1559–1571.
- [20] CHIU, S. K. L., AND STEWART, R. R. Tomographic determination of three-dimensional seismic velocity structure using well logs, vertical seismic profiles, and surface seismic data. *Geophysics* 52 (1987), 1085–1098.
- [21] CONN, A. R., GOULD, N. I. M., AND TOINT, P. Testing a class of methods for solving minimization problems with simple bounds on the variables. *Mathematics of Computation* 50 (1988), 399–430.
- [22] CORES, D., FUNG, G., AND MICHELENA, R. A fast and global two point low storage optimization technique for tracing rays in 2d and 3d isotropic media. *Journal of Applied Geophysics* 45 (2000), 273–287.
- [23] DEMBO, R., EISENSTAT, S., AND STEIHAUG, T. Inexact Newton methods. *SIAM J. Numer. Anal.* 19 (1982), 400–408.
- [24] DENNIS, J. E., AND MORÉ, J. J. A characterization of superlinear convergence and its applications to quasi-newton methods. *Math. Comput.* 28 (1974), 549–560.
- [25] DENNIS, J. E., AND SCHNABEL, R. B. *Numerical Methods for Nonlinear Equations and Unconstrained Optimization*. Prentice-Hall, Englewood Cliffs, NJ, 1983.
- [26] DENNIS, J. E., AND SCHNABEL, R. B. *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*. SIAM, Philadelphia, 1996.
- [27] DINIZ-EHRHARDT, M. A., GOMES-RUGGIERO, M. A., MARTÍNEZ, J. M., AND SANTOS, A. Augmented Lagrangian algorithms based on the spectral gradient for solving nonlinear programming problems. Technical Report, Institute of Mathematics, University of Campinas, Brazil, 2000.
- [28] EISENSTAT, S. C., AND WALKER, H. Globally convergent inexact Newton methods. *SIAM J. Opt.* 4 (1994), 393–422.

- [29] FARRA, V., AND MADARIAGA, R. Non-linear reflection tomography. *Geophys. J.* 95 (1988), 135–147.
- [30] FLETCHER, R. *Practical Methods of Optimization*. Wiley, New York, 1987.
- [31] FREUND, R. W. A transpose-free quasi-minimal residual algorithm for non-hermitian linear systems. *SIAM J. Sci. Stat. Comput.* 14, 2 (1993), 470–482.
- [32] FRIEDLANDER, A., GOMES-RUGGIERO, M. A., KOZAKEVICH, D. N., MARTÍNEZ, J. M., AND SANTOS, S. A. Solving nonlinear systems of equations by mean of quasi-Newton methods with a nonmonotone strategy. *Optimizations Methods and Software* 8 (1997), 25–51.
- [33] FRIEDLANDER, A., MARTÍNEZ, J. M., MOLINA, B., AND RAYDÁN, M. Gradient method with retards and generalizations. *SIAM J. Numer. Anal.* 36, 1 (1999), 275–289.
- [34] FRIEDLANDER, A., MARTÍNEZ, J. M., AND RAYDÁN, M. A new method for large-scale constrained convex quadratic minimization problems. *Opt. Methods and Software* 5 (1995), 57–74.
- [35] GASPARO, M. G. A nonmonotone hybrid method for nonlinear systems. *Optimization Meth & Soft* 13 (2000), 79–94.
- [36] GILL, P. E., AND MURRAY, W. Conjugate-gradient methods for large-scale nonlinear optimization. Technical Report SOL 79-15, System Optimization Laboratory, Department of Operations Research, Stanford University, Stanford, California, 1979.
- [37] GLUNT, W., HAYDEN, T. L., AND RAYDÁN, M. Molecular conformation from distance matrices. *J. Comp. Chem.* 14 (1993), 114–120.
- [38] GOLDSTEIN, A. A. *Constructive Real Analysis*. Harper & Row, New York, 1967.
- [39] GOMES-RUGGIERO, M., MARTÍNEZ, J., AND MORETTI, A. Comparing Algorithms for Solving Sparse Nonlinear Systems of Equations. *SIAM J. Sci. Stat. Comp.* 23 (1992), 459–483.

-
- [40] GRIPPO, L., LAMPARIELLO, F., AND LUCIDI, S. A nonmonotone line search technique for Newton's method. *SIAM J. Numer. Anal.* 23, 4 (1986), 707–716.
- [41] GRIPPO, L., AND SCIANDRONE, M. Nonmonotone globalization techniques for the Barlitzai-Borwein gradient method. *Computational Optimization and Applications* 23 (2002), 143–169.
- [42] INCERTI, S., ZIRILLI, F., AND PARISI, V. Algorithm 111. A fortran subroutine for solving systems of nonlinear simultaneous equations. *Computer Journal* 24 (1981), 87–91.
- [43] INCERTI, S., ZIRILLI, F., AND PARISI, V. Algorithm 111. a fortran subroutine for solving systems of nonlinear simultaneous equations. *Computer Journal* 24 (1981), 87–91.
- [44] KELLEY, C. T. *Iterative Methods for Linear and Nonlinear Equations*. SIAM, Philadelphia, 1995.
- [45] LA CRUZ, W. *Enfoque Espectral para Ecuaciones No Lineales de Gran Escala, y una Aplicación en Geofísica*. PhD thesis, Universidad Central de Venezuela, 2004.
- [46] LA CRUZ, W. A new technique with nonlinear equations for seismic ray tracing in 3d heterogeneous isotropic media. *Enviado a la revista Geophysics* (2005).
- [47] LA CRUZ, W. R., MARTÍNEZ, J. M., AND RAYDAN, M. Spectral residual method without gradient information for large-scale nonlinear systems of equations. (En proceso de publicación). *Math. Comput.* (2005).
- [48] LA CRUZ, W. R., AND RAYDAN, M. Nonmonotone spectral methods for large-scale nonlinear systems. *Optimization Meth. & Soft.* 18 (2003), 583–599.
- [49] LI, D. H., AND FUKUSHIMA, M. A derivative-free line search and global convergence of broyden-like method for nonlinear equations. *Optimization Meth. & Soft.* 11 (2000), 187–201.

- [50] LI, G. Successive column correction algorithms for solving sparse nonlinear systems of equations. *Mathematical Programming* 43 (1989), 187–207.
- [51] LI, G. Successive column correction algorithms for solving sparse nonlinear systems of equations. *Mathematical Programming* 43 (1989), 187–207.
- [52] LUENGO, F., RAYDÁN, M., GLUNT, W., AND HAYDEN, T. Preconditioned spectral gradient method. *Numerical Algorithms* 30 (2002), 241–258.
- [53] LUKSAN, L., AND VLCEK, J. Test problems for unconstrained optimization. Technical Report No. 897, Institute of Computer Science, Academy of Sciences of the Czech Republic, Prague, Czech Republic, 2000.
- [54] MARTÍNEZ, J. M. Local convergence theory of inexact Newton methods based on structured least change secante updates. *Math. Comp.* 55 (1990), 143–167.
- [55] MOLINA, B., AND RAYDÁN, M. Preconditioned Barzilai-Borwein method for the numerical solution of partial differential equations. *Numerical Algorithms* 13 (1996), 45–60.
- [56] MOLINA, B., AND RAYDÁN, M. *Métodos Iterativos Tipo Krylov para Sistemas Lineales*. Escuela Venezolana de Matemáticas, Asociación Matemática Venezolana, Centro de Estudios Avanzados - Instituto Venezolano de Investigaciones Científicas, Caracas, 2004.
- [57] MORE, J., GARBOW, B., AND HILLSTROM, K. Testing unconstrained optimization software. *ACM Transactions on Mathematical Software* 7 (1981), 17–41.
- [58] MORENO, J. Paralelización de los algoritmos SANE y DF-SANE para el trazado de rayos sísmicos. Master's thesis, Universidad Central de Venezuela, 2005.
- [59] NEWTON, I. *The Mathematical Papers of Isaac Newton (7 volumes)*. D. T. Whiteside, ed., Cambridge University Press, Cambridge, 1967-1976.

- [60] ORTEGA, J. M., AND RHEINBOLDT, W. C. *Iterative Solution of Nonlinear Equations in Several Variables*. Academic Press, New York, 1970.
- [61] RAPHSON, J. *Analysis aequationum universalis seu ad aequationes algebraicas resolvendas methodus generalis, et expedita, ex nova infinitarum seieum doctrina, deducta ac demonstrata*. Original in British Library, London, 1690.
- [62] RAYDÁN, M. The Barzilai and Borwein Gradient Method for the large Scale Unconstrained Minimization Problems. *SIAM* 7, 1 (1997), 26–33.
- [63] ROBERTS, S., AND SHIPMAN, J. On the closed form solution of Troesch's Problem. *Journal of Computational Physical* 21 (1976), 291–304.
- [64] ROBERTS, S. M., AND SHIPMAN, J. S. On the closed form solution of troesch's problem. *Journal of Computational Physics* 21 (1976), 291–304.
- [65] SADD, Y., AND SHULTZ, M. H. GMRES: generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM Sci. Stat. Comput.* 7 (1986), 856–869.
- [66] SHERMAN, A. H. On Newton-iterative methods for the solution of systems of nonlinear equations. *SIAM J. Numer. Anal.* 15 (1978), 775–771.
- [67] SPEDICATO, E. Computational experience with quasi-newton algorithms for minimization problems of moderately large size. *Towards Global Optimization* 2 (1978), 209–219.
- [68] TOINT, P. Numerical solution of large sets of algebraic equations. *Mathematics of Computation* 46 (1986), 175–189.
- [69] TREFETHEN, L. N., AND BAU, D. *Numerical linear algebra*. SIAM, Philadelphia, 1997.
- [70] TROSSET, M. W. Numerical algorithms for MDS. In *Classification and Knowledge Organization*, R. Klar and P. Opitz, Eds. Springer, Berling, 1994, pp. 80–92.

-
- [71] VAN DER VORST, H. A. Bi-CGSTAB: a fast and smoothly converget variant Bi-CG for the solution of non-symmetric linear systems. *SIAM Sci. Stat. Comput.* 13 (1992), 631–644.
- [72] WELLS, C., GLUNT, W., AND HAYDEN, T. L. Searching conformational space with the spectral distance geometry algorithm. *Journal of Molecular Structure (Theochem)* 308 (1994), 263–271.
- [73] WOLFE, M. A. *Numerical Methods for Unconstrained Optimization*. Van Nostrand Reinhold, New York, 1978.