

Universidad Central de Venezuela
Facultad de Ciencias
Escuela de Computación
Postgrado en Ciencias de la Computación

**Enfoque Espectral
para Ecuaciones No Lineales de Gran Escala,
y una Aplicación en Geofísica**



Trabajo presentado por el

Msc. William René La Cruz Bastidas

Ante la Ilustre
Universidad Central de Venezuela
Para optar por el título de

Doctor en Ciencias mención Computación

Tutor: Profesor Marcos Raydán

Caracas, Mayo de 2004

*Dedicado a
Janetsi y Victor Jehosmar,
quienes me obsequian
la alegría de vivir*

Agradecimientos

Al Prof. Marcos Raydán, quien me orientó sabiamente tanto en la realización de esta Tesis, como también en la culminación de mis estudios de doctorado, enseñándome a descubrir el inmenso universo de la investigación.

A mi esposa Janetsi, quien con su amor, tolerancia, paciencia y apoyo, me ayudó a alcanzar los objetivos terminales de mis estudios de doctorado.

A mi hijo Victor Jehosmar, quien con sus cuatro años de edad me enseñó lo bello de vivir y ser querido.

A todos muchas gracias.

Resumen

El objetivo principal de la investigación es el diseño y estudio de un algoritmo para sistemas de ecuaciones no lineales de gran escala. Este algoritmo lo denominamos SANE, y es una adaptación del Método del Gradiente Espectral en la resolución de sistemas de ecuaciones no lineales.

Realizamos un análisis de convergencia y demostramos que SANE converge globalmente cuando se aplica a sistemas de ecuaciones lineales que consideran matrices simétrica y positivo (negativo) definidas. Con el objeto de utilizar SANE para resolver cualquier sistemas de ecuaciones no lineales, combinamos a SANE con dos búsquedas lineales no monótonas. Estas combinaciones dieron lugar a dos versiones globalizadas de SANE, que denominamos SANEG y SANEF. Demostramos que SANEG y SANEF convergen globalmente, y realizamos experimentos donde comparamos su eficiencia numérica. En estas pruebas encontramos que SANEG es más eficiente que SANEF.

Con el propósito de comparar la eficiencia numérica de SANEG con métodos tipo Newton y tipo Secante, diseñamos experimentos numéricos que comprendían sistemas de ecuaciones no lineales, cuya dimensión se encontraba entre cincuenta y cincuenta mil variables. Los resultados arrojados por estos experimentos mostraron que SANEG es robusto, ya que no falló en la resolución de ninguno de los sistemas de ecuaciones no lineales estudiados. También se pudo constatar que, en promedio, SANEG resuelve en un tiempo de cómputo menor que los demás algoritmos, los sistemas de ecuaciones no lineales propuestos.

Por último, escogimos el problema del trazado de rayos sísmicos en tres dimensiones para estudiar el comportamiento numérico de SANEG. A tal efecto, creamos un nuevo modelo matemático que describe el trazado de rayos sísmicos utilizando un sistema de ecuaciones no lineales. Las experiencias numéricas mostraron que SANEG resuelve eficazmente el problema del trazado de rayos sísmicos en tres dimensiones, para el modelo matemático planteado.

Contenido

Introducción	1
1 Preliminares	3
1.1 Métodos iterativos para sistemas de ecuaciones no lineales	3
1.1.1 Técnicas de globalización para sistemas de ecuaciones no lineales . .	7
1.2 Método Gradiente Espectral	9
1.2.1 Caso cuadrático	10
1.2.2 Caso no cuadrático	11
2 Método Espectral para Ecuaciones No Lineales	13
2.1 Nuevo algoritmo para ecuaciones no lineales	13
2.1.1 Esquema general de SANE	14
2.2 Algoritmo SANE globalizado	17
2.2.1 Algoritmo SANEG	18
2.2.2 Algoritmo SANEF	25
2.2.3 Comparación numérica de SANEG y SANEF	32
2.3 Comparación numérica de SANEG con algoritmos Newton Inexactos y el algoritmo de Broyden	35
3 Trazado de Rayos Sísmicos en Medios Heterogéneos en 3D	44
3.1 Modelo de lentitud ondular	44
3.1.1 Funcional de tiempo de viaje y principio de Fermat	45
3.2 Trazado de rayos entre dos puntos	46
3.2.1 Ejemplos numéricos del trazado de rayo entre dos puntos	50
3.3 Trazado de rayos en medios con interfaces	51
3.4 SANEG y el trazado de rayos sísmicos en 3D	53

3.4.1	Ejemplos numéricos del trazado de rayos en 3D	54
	Conclusión	59
	A Funciones de Prueba	62
	Bibliografía	72

Lista de Figuras

2.1	Comportamiento no monótono de SANE.	17
2.2	Comportamiento de SANEG y SANEF para $F_1(x)$	34
2.3	Comportamiento de SANEG y SANEF para $F_2(x)$	34
3.1	Trazado de rayo entre dos puntos en un medio con velocidad $v(x, y, z) = 100$	50
3.2	Trazado de rayo entre dos puntos en un medio con velocidad $v(x, y, z) =$ $10x + 3y - 4z + 10$	51
3.3	Medio con una interface.	52
3.4	Tipos de rayos.	54
3.5	Trazado de rayos correspondiente al Ejemplo 1.	56
3.6	Trazado de rayos correspondiente al Ejemplo 2.	58

Lista de Tablas

2.1	Iteraciones, evaluaciones de $F(x)$, búsquedas lineales, tiempo de CPU y $\ F(x^*)\ $, para SANEG y SANEF.	34
2.2	Índices	38
2.3	Funciones de prueba: Newton-GMRES.	39
2.4	Funciones de prueba: Newton-Bi-CGSTAB.	40
2.5	Funciones de prueba: Newton-TFQMR.	41
2.6	Funciones de prueba: Broyden.	42
2.7	Funciones de prueba: SANEG.	43
3.1	Resultados numéricos del Ejemplo 1.	56
3.2	Resultados numéricos del Ejemplo 2.	57

Introducción

Existen muchos problemas que requieren la solución de un sistema de ecuaciones no lineales como parte de su resolución. Los métodos Newton y Casi Newton se emplean, tradicionalmente, para resolver sistemas de ecuaciones no lineales ([33], [38], [31], [39]). Estos métodos son atractivos porque convergen rápidamente para un iterado inicial suficientemente bueno. El gran inconveniente de estos métodos es que deben resolver un sistema de ecuaciones lineales en cada iteración usando la matriz Jacobiana o una aproximación de la misma. Por ello, cuando la dimensión del problema es muy grande ($n \gg 10000$), se tornan computacionalmente costosos, lo cual produce un elevado tiempo de cómputo.

Un procedimiento satisfactorio para la solución de un sistema de ecuaciones no lineales, para valores grandes de n , es usar los métodos Newton Inexactos [19, 38], que resuelven inexactamente el sistema de ecuaciones lineales a través de procesos iterativos [4, 13, 14, 38]. La inexactitud ocurre ya que los métodos iterativos internos se detienen prematuramente en la resolución del sistema de ecuaciones lineales, ocasionando de esta forma un bajo costo computacional por iteración. Modernas implementaciones usan procesos iterativos basados en el subespacio de Krylov [4, 13, 14, 38] (por ejemplo, TFQMR [27], GMRES [54] y Bi-CGSTAB [60]).

Nosotros desarrollamos un nuevo enfoque para la resolución de sistemas de ecuaciones no lineales, $F(x) = 0$, denominado SANE, que considera la combinación del uso sistemático de $\pm F(x)$ como dirección de descenso, con las ideas del Método Gradiente Espectral, que es un método no monótono de bajo costo computacional para encontrar mínimos locales. El Método Gradiente Espectral fue introducido por Barzilai y Borwein [3], su convergencia para cuadráticas fue establecida por Raydán [51], y su esquema globalizado fue discutido recientemente por Raydán [52]. Éste método emplea una variante de la búsqueda lineal no monótona propuesta por Grippo, Lampariello y Lucidi [34].

En este trabajo también presentamos un nuevo modelo matemático para el trazado

de rayos sísmicos en medios heterogéneos en tres dimensiones. Tal modelo consiste en considerar el trazado de rayos sísmicos como un sistema de ecuaciones no lineales, que se construye discretizando el rayo y forzando a que se satisfaga el Principio de Fermat entre dos puntos consecutivos del mismo. El resto del trabajo lo organizamos de la siguiente manera. En el Capítulo 1 desarrollamos los conceptos e ideas que fundamentarán el análisis posterior; y estudiamos algunas propiedades importantes del Método Gradiente Espectral. En el Capítulo 2 estudiamos algunas características de SANE y demostramos su convergencia global. También presentamos dos versiones globalizadas de SANE, estudiamos su convergencia y realizamos un estudio numérico de su eficiencia computacional. En el Capítulo 3 desarrollamos el nuevo enfoque para el trazado de rayos sísmicos en medios heterogéneos en tres dimensiones y damos algunos ejemplos numéricos. Finalmente, presentamos las conclusiones del estudio investigativo realizado.

Capítulo 1

Preliminares

A continuación mostraremos una breve exposición de los esquemas generales de algunos métodos iterativos para la resolución de sistemas de ecuaciones no lineales y las técnicas de globalización que usualmente se emplean para estos sistemas. También esbozaremos los aspectos más importantes del método Gradiente Espectral. (Para una presentación detallada del método Gradiente Espectral se recomienda los artículos de Raydán [52, 51] y la reciente recopilación de Fletcher [26].)

1.1 Métodos iterativos para sistemas de ecuaciones no lineales

Para comenzar definamos sistema de ecuaciones no lineales.

Definición 1.1 (Sistema de ecuaciones no lineales) *Sea $F : \mathbb{R}^n \rightarrow \mathbb{R}^m$ una función cualquiera. Se denomina sistema de ecuaciones no lineales a la ecuación*

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

En otras palabras, consiste en encontrar al menos un punto x^ perteneciente a \mathbb{R}^n tal que $F(x^*) = 0$. Nosotros estudiaremos el caso $m = n$.*

La solución numérica de (1.1) normalmente se obtiene mediante un esquema iterativo, es decir, moviéndose en cada iteración de un estimado x_k de x^* a un mejor estimado x_{k+1} .

Los métodos iterativos para sistemas de ecuaciones no lineales lo clasificamos en tres grupos: *Método de Newton, Métodos Casi-Newton y Métodos Newton Inexacto.*

Método de Newton

El método de Newton tiene la forma:

$$\begin{aligned} &x_0 \in \mathbb{R}^n \text{ dado} \\ &\text{para } k = 0, 1, \dots \text{ hacer} \\ &\quad x_{k+1} = x_k - J(x_k)^{-1}F(x_k), \end{aligned}$$

donde $J(x)$ es la matriz Jacobiana de F evaluada en x . En el contexto de sistemas de ecuaciones el método de Newton aparece en el siglo XVII ([46], [50]). 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 de ecuaciones lineales $J(x_k)s_k = -F(x_k)$, lo cual resulta muy costoso cuando la dimensión del problema es grande ($n \gg 10^4$). Se han realizado estudios donde se combina el método de Newton con una estrategia de búsqueda direccional no monótona tratando de hacerlo más eficiente computacionalmente, pero los resultados no son tan satisfactorios ([34], [31]).

Métodos Casi-Newton

En general los métodos casi-Newton tienen la forma:

$$\begin{aligned} &x_0 \in \mathbb{R}^n \text{ dado} \\ &\text{para } k = 0, 1, \dots \text{ hacer} \\ &\quad x_{k+1} = x_k - H_k^{-1}F(x_k), \end{aligned}$$

donde H_k es una matriz $n \times n$ que 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 de ecuaciones lineales $H_k s_k = -F(x_k)$ resulta a menudo más barata que la resolución del sistema $J(x_k)s_k = -F(x_k)$. Entre los métodos casi-Newton más utilizados se pueden nombrar:

- *Newton Modificado*. El método de Newton modificado tiene la forma:

$$\begin{aligned} &x_0 \in \mathbb{R}^n \text{ dado} \\ &\text{para } k = 0, 1, \dots \text{ hacer} \\ &\quad x_{k+1} = x_k - J(x_0)^{-1}F(x_k). \end{aligned}$$

- *Newton con Diferencia Finita.* El método de Newton con diferencia finita tiene la forma:

$$x_0 \in \mathbb{R}^n \text{ dado}$$

para $k = 0, 1, \dots$ hacer

$$x_{k+1} = x_k - H_k^{-1}F(x_k),$$

donde $H_k \approx J(x_k)$ y se define como

$$(H_k)_j = \frac{1}{h_k} [F(x_k + h_k e_j) - F(x_k)]$$

con $h_k > 0$ pequeño y e_j el vector cuya única componente distinta de cero e igual a uno es la j -ésima.

- *Métodos tipo Secante.* La forma general de los métodos tipo secante es:

$$x_0 \in \mathbb{R}^n \text{ y } A_0 \text{ dados}$$

para $k = 0, 1, \dots$ hacer

$$x_{k+1} = x_k - A_k^{-1}F(x_k)$$

tal que $A_k \in \mathbb{R}^{n \times n}$ satisface la ecuación de la secante,

$$A_k s_{k-1} = y_{k-1}, \tag{1.2}$$

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:

- *Método de Broyden* ([15]).

$$x_0 \in \mathbb{R}^n \text{ y } A_0 \in \mathbb{R}^{n \times n} \text{ dados}$$

para $k = 0, 1, \dots$ hacer

$$\text{Resolver } A_k s_k = -F(x_k)$$

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

$$y_k = F(x_{k+1}) - F(x_k)$$

$$A_{k+1} = A_k + \frac{(y_k - A_k s_k) s_k^t}{s_k^t s_k}.$$

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)”.

– Método “Malo” de Broyden.

$x_0 \in \mathbb{R}^n$ y $A_0 \in \mathbb{R}^{n \times n}$ dados

para $k = 0, 1, \dots$ hacer

$$\text{Resolver } A_k s_k = -F(x_k)$$

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

$$y_k = F(x_{k+1}) - F(x_k)$$

$$A_{k+1} = A_k + \frac{(y_k - A_k s_k) y_k^t}{y_k^t A_k s_k} A_k.$$

– Método de Broyden Inverso.

$x_0 \in \mathbb{R}^n$ y $A_0^{-1} \in \mathbb{R}^{n \times n}$ dados

para $k = 0, 1, \dots$ hacer

$$s_k = -A_k^{-1} F(x_k)$$

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

$$y_k = F(x_{k+1}) - F(x_k)$$

$$A_{k+1}^{-1} = A_k^{-1} + \frac{(s_k - A_k^{-1} y_k) s_k^t A_k^{-1}}{s_k^t A_k^{-1} y_k}.$$

Para un estudio más detallado de los métodos casi-Newton, se recomienda el libro de Dennis y Schnabel [21], y también el libro de Fletcher [24].

Métodos Newton Inexactos

Un enfoque apropiado para valores grandes de n es usar los métodos tipo *Newton Inexactos*. 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 ([19, 23, 43, 55]). 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 [19] y [38] 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

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

y $\{\eta_k\}$ es una sucesión de números reales positivos dada con anterioridad.

Implementaciones modernas de métodos Newton inexactos emplean métodos iterativos basados en el espacio de Krylov para resolver el sistema lineal [4, 13, 14, 38]. Como ejemplo de estas implementaciones son:

- *Newton inexacto con GMRES (Newton-GMRES)*. Newton inexacto con la regla de Armijo y la fórmula de Eisenstat-Walker [23]. Además utiliza GMRES [54] (Generalized minimum residual method) para resolver sistemas de ecuaciones lineales. Newton-GMRES posee código en MATLAB dado por Kelley [38].
- *Newton Inexacto con Bi-CGSTAB (Newton-Bi-CGSTAB)*. Newton inexacto con la regla de Armijo y la fórmula de Eisenstat-Walker [23]. Además utiliza Bi-CGSTAB [60] (Bi-conjugate gradient stabilized method) para resolver sistemas de ecuaciones lineales. Newton-Bi-CGSTAB posee código en MATLAB dado por Kelley [38].
- *Newton Inexacto con TFQMR (Newton-TFQMR)*. Newton inexacto con la regla de Armijo y la fórmula de Eisenstat-Walker [23]. Además utiliza TFQMR [27] (Transpose-free quasi-minimal residual algorithm) para resolver sistemas de ecuaciones lineales. Newton-TFQMR posee código en MATLAB dado por Kelley [38].

Es muy importante agradecer al Prof. C. T. Kelley por suministrar gratuitamente el código en Matlab de los algoritmos Newton-GMRES, Newton-Bi-CGSTAB y Newton-TFQMR. Tales códigos se pueden obtener por el protocolo ftp de math.ncsu.edu en el directorio:

pub/kelley/matlab

ó conectándose a la página Web:

[//www.siam.org/books/kelley/kelley.html](http://www.siam.org/books/kelley/kelley.html)

1.1.1 Técnicas de globalización para sistemas de ecuaciones no lineales

Diremos que un algoritmo es globalmente convergente si genera una sucesión $\{x_k\}$ con la propiedad de que para cualquier iterado inicial, $x_0 \in \mathbb{R}^n$, converge a una solución del sistema de ecuaciones no lineales (1.1). De esta forma, entenderemos por técnica de globalización a un procedimiento numérico que unido con un algoritmo, garantice llegar a una solución de (1.1), comenzando con cualquier iterado inicial.

De las distintas técnicas de globalización que se encuentran en la literatura especializada, emplearemos la *búsqueda direccional*. Utilizamos esta técnica debido a su simplicidad, fácil implementación y, además, es la que mejor se adapta a esquemas muy simples de bajo costo.

Supongamos que deseamos resolver el problema $\min_{x \in \mathbb{R}^n} f(x)$ con f una función definida de \mathbb{R}^n en \mathbb{R} . Entonces, la búsqueda direccional consiste en escoger el tamaño del paso, λ_k , en cada iteración k de tal manera de garantizar convergencia. Un tipo de búsqueda direccional de relativa importancia es la *búsqueda lineal*, que consiste en satisfacer, para $d_k \in \mathbb{R}^n$ una dirección de descenso de f en x_k , algunas de las siguientes condiciones:

Condición α de Armijo. Dado $\alpha \in (0, 1)$, escoger $\lambda_k > 0$ tal que

$$f(x_k + \lambda_k d_k) \leq f(x_k) + \alpha \lambda_k \nabla f(x_k)^t d_k.$$

Condición β de Goldstein. Dado $\beta \in (0, 1)$, escoger $\lambda_k > 0$ tal que

$$f(x_k + \lambda_k d_k) \geq f(x_k) + \beta \lambda_k \nabla f(x_k)^t d_k.$$

Condición β de Wolfe. Dado $\beta \in (0, 1)$, escoger $\lambda_k > 0$ tal que

$$\nabla f(x_k + \lambda_k d_k)^t d_k \geq \beta \nabla f(x_k)^t d_k.$$

Normalmente los algoritmos emplean una búsqueda lineal que considera una combinación de las condiciones anteriores. La combinación más utilizada es la condición α de Armijo combinada con la condición β de Goldstein ó la condición β de Wolfe, tomando β en el intervalo $(\alpha, 1)$. Estas búsquedas lineales como tal son monótonas, es decir, exigen descenso de f en cada iteración. Existen otras búsquedas lineales que no son monótonas, de las cuales hablaremos cuando estudiemos el método Gradiente Espectral.

Para utilizar una búsqueda lineal en la resolución de sistemas no lineales, primero se define una función $f : \mathbb{R}^n \rightarrow \mathbb{R}$ como

$$f(x) = \|F(x)\|_2^2, \tag{1.3}$$

denominada *función mérito*. Luego, se aplica algún proceso iterativo para sistemas de ecuaciones no lineales unido con las condiciones de Armijo, Goldstein ó Wolfe. Se observa

que el gradiente y la Hessiana de $f(x)$ son, respectivamente:

$$\nabla f(x) = 2J_F(x)^t F(x) \quad (1.4)$$

y

$$\nabla^2 f(x) = 2J_F(x)^t J_F(x) + 2T(x), \quad (1.5)$$

donde

$$T(x) = \sum_{i=1}^n f_i(x) \nabla^2 f_i(x),$$

y las funciones $f_i : \mathbb{R}^n \rightarrow \mathbb{R}$ son tales que $F(x) = (f_1(x), \dots, f_n(x))^t$.

1.2 Método Gradiente Espectral

Sea $f : \mathbb{R}^n \rightarrow \mathbb{R}$ diferenciable. Con el propósito de resolver el problema $\min \{f(x)\}$, con $x \in \mathbb{R}^n$, consideremos el siguiente problema asociado:

$$\text{encontrar } x^* \in \mathbb{R}^n \text{ tal que } \nabla f(x^*) = 0. \quad (1.6)$$

La solución numérica de (1.6) regularmente se obtiene mediante un esquema iterativo. Con mucha frecuencia, cada iteración envuelve el cálculo de un paso Casi-Newton, $s_{CN} = -A_k^{-1} \nabla f(x_k)$, donde $A_k \in \mathfrak{R}^{n \times n}$ es una aproximación del Hessiano de f en x_k . Al finalizar cada iteración, A_k se actualiza y se obtiene A_{k+1} , una aproximación del Hessiano de f en x_{k+1} . Esta aproximación, usualmente, se escoge de forma tal que satisfaga la ecuación de la secante (1.2), $A_k s_{k-1} = y_{k-1}$, donde $s_k = x_{k+1} - x_k$ y $y_k = g_{k+1} - g_k$, con $g_k = \nabla f(x_k)$. Además de satisfacer la ecuación de la secante, la matriz A_{k+1} debe restringirse a un conjunto de matrices que posean propiedades deseables (simétricas, positivo definidas, etc.).

La propiedad que caracteriza a los métodos espectrales se fundamenta en la siguiente observación: el escalar $\alpha_{k+1} \in \mathbb{R}$ que resuelve de forma única el sistema lineal sobredeterminado $y_k = \alpha_{k+1} s_k$, en el sentido de los mínimos cuadrados, está dado por

$$\alpha_{k+1} = \frac{s_k^t y_k}{s_k^t s_k}, \quad (1.7)$$

si $s_k \neq 0$. Es decir, restringiendo la matriz A_{k+1} a la familia de múltiplos escalares de la identidad y exigiendo que la ecuación de la secante se satisfaga en el sentido de los mínimos cuadrados. En otras palabras, un método espectral para resolver el problema

$\min \{f(x)\}$ con $x \in \mathbb{R}^n$, es un método tipo secante que satisface la ecuación (1.7). Bajo estas restricciones se obtiene el método Gradiente Espectral.

Algoritmo 1.1 (*Gradiente Espectral*)

Dados $x_0 \in \mathbb{R}^n$ y $\alpha_0 \in \mathbb{R}$
para $k = 0, 1, \dots$ hasta la convergencia

1. $s_k = -\frac{1}{\alpha_k}g_k$
 2. $x_{k+1} = x_k + s_k$
 3. $y_k = g_{k+1} - g_k$
 4. $\alpha_{k+1} = \frac{s_k^t y_k}{s_k^t s_k}$
-

1.2.1 Caso cuadrático

Si se considera el problema (1.6) cuando $f(x) = \frac{1}{2}x^t Ax - b^t x + c$ es una función cuadrática, donde A es una matriz simétrica y positivo definida, $b \in \mathbb{R}^n$ y $c \in \mathbb{R}$, entonces α_{k+1} en (1.7) se escribe como

$$\alpha_{k+1} = \frac{s_k^t A s_k}{s_k^t s_k} = \frac{g_k^t A g_k}{g_k^t g_k}. \tag{1.8}$$

Es decir, para este caso α_{k+1} resulta el cociente de Rayleigh de A evaluado en g_k .

El método del gradiente espectral para funciones cuadráticas es no monótono, es decir, no garantiza descenso en la función objetivo en cada iteración ([3]).

Por otra parte, existe una relación entre cualquier método que use la dirección del gradiente negativo, para minimizar funciones cuadráticas, y el método de las potencias con desplazamiento para el cálculo aproximado de autovalores y autovectores ([25] y [32]). La relación entre el gradiente espectral y el método de las potencias con desplazamiento, permite establecer la convergencia global del gradiente espectral cuando se aplica a una funcional cuadrática cuya Hessiana es positivo definida. Este importante resultado se expresa en el siguiente teorema que corresponde al Teorema 1 en Raydán [51].

Teorema 1.1 *Sea $q(x)$ una función cuadrática estrictamente convexa. Sea $\{x_k\}$ la sucesión generada por el método del gradiente espectral y x^* el único minimizador de q . Entonces, o bien $x_j = x^*$ para algún j finito, o la sucesión $\{x_k\}$ converge a x^* .*

En cuanto a la velocidad de convergencia del gradiente espectral, recientemente se comienza a estudiar. Algunos resultados parciales se pueden encontrar en [3], [25] y [32].

En particular, Barzilai y Borwein establecen convergencia R-superlineal si A posee solo dos autovalores distintos. Sin embargo, Fletcher [25] argumenta que, en general, puede esperarse sólo convergencia R-lineal. Un resultado más reciente de Dai y Liao [18] establece convergencia R-lineal en el caso cuadrático convexo.

1.2.2 Caso no cuadrático

Para aplicar el método del gradiente espectral al problema general de minimización sin restricciones:

$$\min_{x \in \mathbb{R}^n} \{f(x)\}, \quad (1.9)$$

donde $f : \mathbb{R}^n \rightarrow \mathbb{R}$ continuamente diferenciable, es necesario asociarlo a una técnica de globalización que permita garantizar la convergencia a puntos críticos desde cualquier iterado inicial.

La técnica de globalización que ha resultado muy eficiente en la práctica es la siguiente búsqueda lineal:

$$f(x_{k+1}) \leq \max_{0 \leq j \leq M} [f(x_{k-j})] + \gamma g_k^t(x_{k+1} - x_k), \quad (1.10)$$

donde M es un entero no negativo y γ es un número positivo pequeño. Este tipo de búsquedas lineales no monótonas fueron introducidas por Grippo, Lampariello y Lucidi [34]. Cuando $M > 0$, la condición (1.10) permite que la función objetivo aumente su valor en algunas iteraciones y aún así garantizar convergencia global ([34]). Esta propiedad teórica se asocia en forma efectiva con la naturaleza no monótona del método Gradiente Espectral. A continuación se muestra el algoritmo del método Gradiente Espectral Globalizado.

Algoritmo 1.2 (*Gradiente Espectral Globalizado*)

Dados α_0 , un entero $M > 0$, $\delta > 0$, $0 < \sigma_1 < \sigma_2 < 1$, $0 < \varepsilon < 1$, $0 < \gamma < 1$.

Tomar un iterado inicial $x_0 \in \mathbb{R}^n$ y asignar $k = 0$.

Paso 1: Si $\|g_k\|_2 = 0$ parar el proceso.

Paso 2: Si $\alpha_k \leq \varepsilon$ o $\alpha_k \geq 1/\varepsilon$, entonces asignar $\alpha_k = \delta$.

Paso 3: Asignar $\lambda = 1/\alpha_k$.

Paso 4: Si $f(x_k - \lambda g_k) \leq \max_{0 \leq j \leq \min(k, M)} [f(x_{k-j})] - \gamma \lambda g_k^t g_k$,

entonces ir al Paso 6, de lo contrario, ir al Paso 5.

Paso 5: Escoger $\sigma \in [\sigma_1, \sigma_2]$, asignar $\lambda = \sigma \lambda$ e ir al Paso 4.

Paso 6: Asignar $\lambda_k = \lambda$, $x_{k+1} = x_k - \lambda_k g_k$, $y_k = g_{k+1} - g_k$, e ir al Paso 7.

Paso 7: Asignar $\alpha_{k+1} = - \left(\frac{g_k^t y_k}{\lambda_k g_k^t g_k} \right)$, hacer $k = k + 1$, e ir al Paso 1.

La convergencia del Gradiente Espectral Globalizado se establece en el siguiente teorema que corresponde al Teorema 2.1 en Raydán [52].

Teorema 1.2 Si $\Omega_0 = \{x : f(x) \leq f(x_0)\}$ es un conjunto acotado, y $f : \mathbb{R}^n \rightarrow \mathbb{R}$ es continuamente diferenciable en algún vecindario abierto N de Ω_0 , entonces la sucesión $\{x_k\}$ generada por el algoritmo del gradiente espectral globalizado satisface lo siguiente: o bien $g(x_j) = 0$ para algún j finito, o las siguientes propiedades se cumplen:

(i) $\lim_{k \rightarrow \infty} \|g_k\|_2 = 0$;

(ii) ninguno de los puntos límites de $\{x_k\}$ es un máximo local de f ;

(iii) si el número de puntos estacionarios de f en Ω_0 es finito, entonces la sucesión $\{x_k\}$ converge.

El Gradiente Espectral ha mostrado ser muy robusto y efectivo en sus aplicaciones numéricas. Se ha utilizado en el problema químico de conformación molecular ([32] y [61]), en el trazado de rayos sísmicos ([17]) y en la tomografía de reflexión sísmica ([16]), en la estimación de constantes ópticas ([8]), en el problema de escalamiento multidimensional que aparece en psicometría y en estadística ([12], [41] y [58]), y en la diferenciación automática de funciones ([9]). Versiones precondicionadas del método se han usado en la solución numérica de ecuaciones en derivadas parciales ([29], [41] y [44]). El método se ha extendido al problema de minimizar funciones sobre conjuntos convexos ([6], [30] y [10]). También se ha extendido al problema general de programación no lineal mediante técnicas Lagrangeanas ([22]).

Capítulo 2

Método Espectral para Ecuaciones No Lineales

Tradicionalmente los métodos empleados para resolver el sistema (1.1) son: Newton, Casi-Newton y Newton Inexacto ([33], [38], [31], [39], [56]). El gran inconveniente computacional de estos métodos es la resolución de un sistema de ecuaciones lineales en cada iteración. Cuando el problema es de gran dimensionalidad ($n \gg 10^4$), se tornan computacionalmente costosos, lo cual produce un elevado tiempo de cómputo. Por lo tanto, es de gran importancia el diseño de un *método para sistemas de ecuaciones no lineales de gran escala de bajo costo computacional*. Con este propósito en mente, los objetivos de este capítulo son el diseño y estudio de la convergencia de un método para sistemas de ecuaciones no lineales de gran escala que considere las ideas espectrales de Raydán [52], combinándolas con el uso sistemático de las direcciones $d = \pm F(x)$. Concretamente, adaptaremos el Método Gradiente Espectral, dado en el Algoritmo 1.1, para la resolución del sistema de ecuaciones no lineales (1.1).

2.1 Nuevo algoritmo para ecuaciones no lineales

Consideraremos una función $F : \mathbb{R}^n \rightarrow \mathbb{R}^n$ continuamente diferenciable. Denotaremos con $F_k = F(x_k)$, con $J_k = J(x_k)$ a la matriz Jacobiana de F evaluada en $x_k \in \mathbb{R}^n$ y con $\|\cdot\|$ la norma Euclidea. Definamos la función mérito $f : \mathbb{R}^n \rightarrow \mathbb{R}$ como en (1.3).

En la siguiente sección mostraremos la adaptación del Algoritmo 1.1 en la resolución del sistema de ecuaciones no lineales (1.1). A tal algoritmo lo hemos denominado *SANE* (Spectral Approach for Nonlinear Equations).

2.1.1 Esquema general de SANE

En el Algoritmo 2.1 observamos el esquema general de SANE.

Algoritmo 2.1 (*SANE*)

Dados $x_0 \in \mathbb{R}^n$ y $\alpha_0 > 0$, *asignar* $k = 0$.
Paso 1: Si $\|F_k\| = 0$, *parar el proceso*.
Paso 2: Si $F_k^t J_k F_k = 0$, *parar el proceso*.
Paso 3: *Asignar* $d_k = -\text{sgn}(F_k^t J_k F_k) F_k$.
Paso 4: $x_{k+1} = x_k + (1/\alpha_k) d_k$.
Paso 5: $y_k = F_{k+1} - F_k$.
Paso 6: $\alpha_{k+1} = \text{sgn}(F_k^t J_k F_k) \left(\frac{\alpha_k d_k^t y_k}{d_k^t d_k} \right)$.

Observaciones:

- (i) En el Paso 3 se escoge la dirección de descenso que se utilizará en cada iteración, donde la función $\text{sgn}(t)$ se denomina función signo y se define como:

$$\text{sgn}(t) = \begin{cases} 1 & t > 0 \\ -1 & t < 0. \end{cases}$$

- (ii) En el Paso 6 se calcula el tamaño de paso espectral

$$\alpha_{k+1} = \text{sgn}(F_k^t J_k F_k) \left(\frac{\alpha_k d_k^t y_k}{d_k^t d_k} \right).$$

El valor de $\text{sgn}(F_k^t J_k F_k)$ en el lado derecho de esta ecuación, es para garantizar que α_{k+1} sea positivo con la mayor frecuencia, lo cual se fundamentará en el Lema 2.1.

Aparte de que SANE y el método Gradiente Espectral resuelven problemas distintos, existen otras dos diferencias fundamentales entre ellos, siendo una de ellas el Paso 2 de SANE. El Paso 2 es un criterio de parada que se activa cuando SANE encuentra un iterado x_k que lo hace funcionar en forma errónea. Si para algún k el Algoritmo 2.1 genera x_k tal que $F_k^t J_k F_k = 0$, éste no sabría escoger la dirección de búsqueda, más aún, ninguna de las direcciones, $-F_k$ o F_k , serían de descenso para la función mérito. Bajo estas condiciones el Algoritmo 2.1 no puede garantizar la convergencia a una solución del sistema (1.1). Por ello, estos casos anómalos deben identificarse y cuando el Algoritmo 2.1 llegue a uno de ellos (si existiesen), es mejor parar el proceso. La otra diferencia entre SANE y el método

Gradiente Espectral es la dirección de búsqueda empleada. Las direcciones utilizadas por SANE son $d_k = \pm F_k$ que, generalmente, no son gradientes de alguna función; en cambio, la dirección utilizada por el método Gradiente Espectral es la dirección del gradiente negativo.

Veamos que, efectivamente, $d_k = -\text{sgn}(F_k^t J_k F_k) F_k$, con $F_k \neq 0$ y $F_k^t J_k F_k \neq 0$, es una dirección de descenso para la función mérito $f(x)$ dada en (1.3). Como

$$\text{sgn}(F_k^t J_k F_k) F_k^t J_k F_k > 0$$

para todo k , entonces

$$\nabla f(x_k)^t d_k = 2F_k^t J_k (-\text{sgn}(F_k^t J_k F_k) F_k) = - (2\text{sgn}(F_k^t J_k F_k) F_k^t J_k F_k) < 0.$$

Por lo tanto, $d_k = -\text{sgn}(F_k^t J_k F_k) F_k$ es una dirección de descenso para $f(x)$.

El siguiente lema nos indica que en la mayoría de los casos α_{k+1} es positivo.

Lema 2.1 *Sea α_{k+1} calculado como en el Paso 6 del Algoritmo 2.1. Entonces, α_{k+1} es positivo si se cumple alguno de los siguientes casos:*

- (i) $F_k^t F_{k+1} < 0$;
- (ii) $F_k^t F_{k+1} > 0$ y $\|F_{k+1}\| < \|F_k\|$.

Demostración. Como

$$\alpha_{k+1} = \text{sgn}(F_k^t J_k F_k) \left(\frac{\alpha_k d_k^t y_k}{d_k^t d_k} \right) = - \left(\frac{\alpha_k F_k^t y_k}{F_k^t F_k} \right),$$

$\alpha_k > 0$ y $F_k^t F_k > 0$, entonces

$$\text{sgn}(\alpha_{k+1}) = -\text{sgn}(F_k^t y_k),$$

es decir, $\alpha_{k+1} > 0$ si, y sólo si $F_k^t y_k < 0$. Supongamos que se cumple (i). Así,

$$F_k^t y_k = F_k^t F_{k+1} - F_k^t F_k < 0,$$

luego, $\alpha_{k+1} > 0$. Supongamos que se cumple (ii). Así,

$$0 < F_k^t F_{k+1} \leq \|F_k\| \|F_{k+1}\| < \|F_k\|^2 = F_k^t F_k,$$

luego,

$$F_k^t y_k = F_k^t F_{k+1} - F_k^t F_k < 0,$$

de esta forma, $\alpha_{k+1} > 0$. □

Nótese que α_{k+1} sólomente es negativo, y en consecuencia tomamos $\delta > 0$ en el Paso 3, cuando $F_k^t F_{k+1} > 0$ y $\|F_{k+1}\| \geq \|F_k\|$, es decir, cuando no se observa descenso en la función mérito.

Continuando con el análisis de SANE, pasemos ahora a estudiar su convergencia global. El siguiente teorema muestra la convergencia global del Algoritmo 2.1 cuando $F(x)$ es el gradiente de una cuadrática estrictamente convexa.

Teorema 2.1 *Sea $F(x) = Ax - b$, donde $b \in \mathbb{R}^n$ y A una matriz simétrica y positivo (negativo) definida. Sea $\{x_k\}$ la sucesión generada por el Algoritmo 2.1. Entonces, o bien $F(x_i) = 0$ para algún i finito, o la sucesión $\{F_k\}$ converge a 0.*

Demostración. Sea $q : \mathbb{R}^n \rightarrow \mathbb{R}$ una función definida como:

$$q(x) = \frac{1}{2}x^t Ax - b^t x + c,$$

donde $c > 0$. Es claro que $q(x)$ es una cuadrática estrictamente convexa, ya que A es simétrica y positivo definida; además, $\nabla q(x) = Ax - b = F(x)$. Por otra parte, se observa que $J_k = A$ para todo k , por ello, $F_k^t J_k F_k > 0$ para todo k . Por lo tanto, el Algoritmo 2.1 no se detendrá con el criterio de parada del Paso 2 y, además, la dirección que siempre utilizará es $d_k = -F_k = -\nabla q(x_k)$, y los Pasos 4, 5 y 6, adquieren, respectivamente, la forma:

$$\begin{aligned} x_{k+1} &= x_k - (1/\alpha_k)\nabla q(x_k), \\ y_k &= \nabla q(x_{k+1}) - \nabla q(x_k), \\ \alpha_{k+1} &= -\frac{\alpha_k \nabla q(x_k)^t y_k}{\nabla q(x_k)^t \nabla q(x_k)}. \end{aligned}$$

En consecuencia, el Algoritmo 2.1 aplicado a $F(x) = Ax - b$ no es más que el método Gradiente Espectral en el caso cuadrático aplicado a $q(x)$. Por lo tanto, en virtud del Teorema 1.1 podemos concluir que la sucesión $\{x_k\}$ converge a x^* , donde x^* es la solución única del sistema de ecuaciones lineales $Ax = b$. De esta forma, de la continuidad de F se infiere que la sucesión $\{F_k\}$ converge a 0. \square

Otra característica teórica muy importante que debemos resaltar de SANE es que es *no monótono*, es decir, en cada iteración no se garantiza el descenso de la función mérito. De hecho, como SANE es una adaptación para sistemas de ecuaciones no lineales del

método Gradiente Espectral, y siendo éste no monótono, entonces SANE también lo es. Verifiquemos mediante un ejemplo numérico que SANE efectivamente es no monótono. Sea $F(x) = Ax$, donde $A = \text{diag}(1, 2, \dots, n)$ (matriz diagonal con $a_{ii} = i$, $i = 1, \dots, n$). Es claro que A es positivo definida, además, $x^* = (0, 0, \dots, 0)^t$ es la única solución del sistema de ecuaciones $F(x) = 0$. En la Figura 2.1 se observa la gráfica de $\|F_k\|$, para $n = 500$, hasta lograr que $\|F_k\| < 10^{-14}$. El punto inicial es $x_0 = (0.5, \dots, 0.5)^t$ y $\alpha_0 = 1.5$. Se puede notar que SANE es netamente no monótono.

Figura 2.1: Comportamiento no monótono de SANE.

2.2 Algoritmo SANE globalizado

Cuando $F : \mathbb{R}^n \rightarrow \mathbb{R}^n$ es cualquier función diferenciable y $x_0 \in \mathbb{R}^n$ es un iterado inicial arbitrario, no siempre se puede garantizar que SANE converja a la solución del sistema (1.1). Para solventar este inconveniente es necesario combinar a SANE con una técnica de globalización. Presentaremos dos versiones de SANE globalizado utilizando dos búsquedas lineales distintas. Una de estas versiones la denominamos SANEG y la otra SANEF.

2.2.1 Algoritmo SANEG

Como SANE es una adaptación del método Gradiente Espectral, un candidato inmediato que debemos considerar como técnica de globalización, es la búsqueda lineal empleada por Raydán [52] en minimización irrestricta, que es una variación de la búsqueda lineal (1.10). La búsqueda lineal empleada por Raydán [52] es:

$$f(x_k + \lambda d_k) \leq \max_{0 \leq j \leq \min(k, M)} [f(x_{k-j})] + \gamma \lambda g_k^t d_k, \quad (2.1)$$

donde $d_k = -g_k$, $g_k = \nabla f(x_k)$, $\gamma \in (0, 1)$ y M es un entero no negativo. Tomando en (2.1) a $f(x)$ como la función mérito dada en (1.3) y $d_k = -\text{sgn}(F_k^t J_k F_k) F_k$, entonces la búsqueda lineal (2.1) adquiere la forma:

$$f(x_k + \lambda d_k) \leq \max_{0 \leq j \leq \min(k, M)} [f(x_{k-j})] + 2\gamma \lambda F_k^t J_k d_k. \quad (2.2)$$

La combinación de SANE con la búsqueda lineal (2.2) origina el Algoritmo 2.2 denominado *SANEG*.

Algoritmo 2.2 (*SANEG*)

Sean $\alpha_0 > 0$, un entero $M > 0$, $\gamma > 0$, $0 < \sigma_1 < \sigma_2 < 1$, $0 < \varepsilon < 1$, y $\delta \in [\varepsilon, 1/\varepsilon]$.

Tomar $x_0 \in \mathbb{R}^n$ un punto inicial y asignar $k = 0$.

Paso 1: *Si $\|F_k\| = 0$, parar el proceso.*

Paso 2: *Si $|F_k^t J_k F_k| / F_k^t F_k < \varepsilon$, parar el proceso.*

Paso 3: *Si $\alpha_k \leq \varepsilon$ o $\alpha_k \geq 1/\varepsilon$, entonces asignar $\alpha_k = \delta$.*

Paso 4: *Tomar $d_k = -\text{sgn}(F_k^t J_k F_k) F_k$.*

Paso 5: *Asignar $\lambda = 1/\alpha_k$.*

Paso 6: *Si $f(x_k + \lambda d_k) \leq \max_{0 \leq j \leq \min(k, M)} [f(x_{k-j})] + 2\gamma \lambda F_k^t J_k d_k$, ir al Paso 8.*

Paso 7: *Escoger $\sigma \in [\sigma_1, \sigma_2]$, asignar $\lambda = \sigma \lambda$ e ir al Paso 6.*

Paso 8: *Asignar $\lambda_k = \lambda$, $x_{k+1} = x_k + \lambda_k d_k$, $y_k = F_{k+1} - F_k$.*

Paso 9: *Asignar $\alpha_{k+1} = \text{sgn}(F_k^t J_k F_k) \left(\frac{d_k^t y_k}{\lambda_k d_k^t d_k} \right)$, hacer $k = k + 1$ e ir al Paso 1.*

Observaciones.

- (i) El escalar $F_k^t J_k F_k$, que frecuentemente aparece en el algoritmo, puede aproximarse empleando la siguiente fórmula

$$F_k^t J_k F_k = F_k^t \left[\frac{F(x_k + h F_k) - F_k}{h} \right], \quad (2.3)$$

donde h es un número positivo pequeño. La aproximación dada en (2.3) no requiere el conocimiento explícito de la matriz Jacobiana, la cual no se tiene para problemas muy grandes. No se requiere del producto $J_k^t F_k$, que no puede aproximarse sin el conocimiento explícito de la matriz Jacobiana. Por tanto, en nuestro caso, el gradiente $\nabla f(x)$ dado por (1.4) no está disponible.

- (ii) El Paso 2 es un criterio de parada cuando se encuentra un punto anómalo x_k , es decir, cuando $F_k^t J_k F_k = 0$. En general, el Paso 2 de SANEG debe considerarse ya que de lo contrario SANEG puede converger a un punto \bar{x} para el cual $\|F(\bar{x})\|$ es distante a cero pero el vector $F(\bar{x})$ es ortogonal a $\nabla f(\bar{x})$. Claramente, esto ocurre cuando la parte simétrica de $J(x)$ es indefinida para algún $x \in \mathbb{R}^n$.

A continuación estudiaremos las características de SANEG. El siguiente lema nos garantiza que el Algoritmo 2.2 ejecuta una iteración, es decir, que el algoritmo está bien definido.

Lema 2.2 *El Algoritmo 2.2 está bien definido.*

Demostración. Para $\gamma > 0$ es fácil ver que la desigualdad

$$f(x_k + \lambda d_k) \leq \max_{0 \leq j \leq \min(k, M)} [f(x_{k-j})] + 2\gamma \lambda F_k^t J_k d_k$$

es cierta para $\lambda > 0$ suficientemente pequeño, dado que cuando $\lambda \rightarrow 0$, por continuidad, $f(x_k + \lambda d_k) \rightarrow f(x_k)$, pero

$$\max_{0 \leq j \leq \min(k, M)} [f(x_{k-j})] + 2\gamma \lambda F_k^t J_k d_k \rightarrow \max_{0 \leq j \leq \min(k, M)} f(x_{k-j}).$$

Por lo tanto, para $\lambda > 0$ suficientemente pequeño

$$f(x_k + \lambda d_k) \leq \max_{0 \leq j \leq \min(k, M)} [f(x_{k-j})],$$

es decir, el Algoritmo 2.2 ejecuta, efectivamente, una iteración. \square

El siguiente lema muestra que la sucesión $\{x_k\}$ generada por el Algoritmo 2.2 está contenida en cierto conjunto de nivel.

Lema 2.3 *Si $\{x_k\}$ es la sucesión generada por el Algoritmo 2.2, entonces $\{x_k\}$ está contenida en el conjunto*

$$\Omega_0 = \{x : 0 \leq f(x) \leq f(x_0)\}. \quad (2.4)$$

Demostración. Es evidente que $f(x_k) \geq 0$ para todo k . Supongamos que $x_k \in \Omega_0$ para $1 \leq k \leq i$, donde i es un entero positivo. Como $\lambda_k > 0$ y $F_k^t J_k d_k < 0$, del Paso 6 se puede obtener, para $k = i + 1$:

$$\begin{aligned} f(x_{i+1}) &\leq \max_{0 \leq j \leq \min(i, M)} f(x_{i-j}) + 2\gamma \lambda_i F_i^t J_i d_i \\ &\leq f(x_0) + 2\gamma \lambda_i F_i^t J_i d_i \\ &\leq f(x_0). \end{aligned}$$

Por lo tanto, $\{x_k\} \subset \Omega_0$. □

Para el resto del análisis necesitamos la siguiente hipótesis.

Hipótesis A.

- (i) El conjunto de nivel Ω_0 dado en (2.4) es acotado.
- (ii) $F(x)$ es continuamente diferenciable en un conjunto abierto que contiene a Ω_0 .
- (iii) $J(x)$ es no singular en todo $x \in \Omega_0$.

Para probar la convergencia global del Algoritmo 2.2 necesitamos el siguiente lema.

Lema 2.4 *Bajo la hipótesis A, si $\{x_k\}$ es la sucesión generada por el Algoritmo 2.2, entonces existen constantes positivas c_1 , c_2 y c_3 tales que:*

$$\|d_k\| \leq c_1 \|\nabla f(x_k)\|, \tag{2.5}$$

$$\|\nabla f(x_k)\| \leq c_2 \|d_k\|, \tag{2.6}$$

$$F_k^t J_k d_k \leq -c_3 \|\nabla f(x_k)\|^2, \tag{2.7}$$

para todo k .

Demostración. Sean T_1 y T_2 constantes positivas tales que $\|J(x)^{-1}\| \leq T_1$ y $\|J(x)\| \leq T_2$, para todo $x \in \Omega_0$. Como $\|d_k\| = \|F_k\|$ y $F_k = (1/2)J_k^{-t}\nabla f(x_k)$, entonces

$$\begin{aligned} \|d_k\| &\leq \frac{1}{2} \|J_k^{-1}\| \|\nabla f(x_k)\| \\ &\leq \frac{T_1}{2} \|\nabla f(x_k)\|, \end{aligned}$$

es decir, $c_1 = T_1/2 > 0$.

En el mismo orden de ideas, como $\nabla f(x_k) = 2J_k^t F_k$, entonces

$$\begin{aligned}\|\nabla f(x_k)\| &\leq 2\|J_k\|\|F_k\| \\ &= 2\|J_k\|\|d_k\| \\ &\leq 2T_2\|d_k\|,\end{aligned}$$

es decir, $c_2 = 2T_2 > 0$.

Por último, por el Paso 2 obtenemos que:

$$|F_k^t J_k F_k| \geq \varepsilon \|F_k\|^2. \quad (2.8)$$

Del Paso 4 se tiene que $d_k = -\text{sgn}(F_k^t J_k F_k) F_k$, luego

$$F_k^t J_k d_k = F_k^t J_k (-\text{sgn}(F_k^t J_k F_k) F_k) = \begin{cases} -F_k^t J_k F_k, & F_k^t J_k F_k > 0 \\ F_k^t J_k F_k, & F_k^t J_k F_k < 0. \end{cases} \quad (2.9)$$

Usando (2.8), (2.9), $\|d_k\| = \|F_k\|$ y (2.6), se sigue que:

$$F_k^t J_k d_k \leq -\varepsilon \|F_k\|^2 = \varepsilon (-\|d_k\|^2) \leq \varepsilon (-c_2^{-2} \|\nabla f(x_k)\|^2) = -(\varepsilon c_2^{-2}) \|\nabla f(x_k)\|^2,$$

en consecuencia, $c_3 = \varepsilon c_2^{-2} > 0$. □

Teorema 2.2 *Bajo la hipótesis A, el Algoritmo 2.2 termina en una iteración finita i donde $F_i = 0$ o $|F_i^t J_i F_i| < \varepsilon \|F_i\|^2$, o genera un sucesión $\{x_k\}$ tal que*

$$\lim_{k \rightarrow \infty} \|F_k\| = 0.$$

Demostración. El conjunto $\Omega_0 \subset \mathbb{R}^n$ es compacto, ya que es cerrado y acotado. Sea $m(k) = \min(k, M)$. Es evidente que

$$m(0) = 0 \text{ y } 0 \leq m(k) \leq \min(m(k-1) + 1, M), \text{ para } k \geq 1.$$

Sea $l(k)$ un entero tal que:

$$\begin{aligned}k - m(k) &\leq l(k) \leq k, \\ f(x_{l(k)}) &= \max_{0 \leq j \leq m(k)} [f(x_{k-j})].\end{aligned}$$

La sucesión $\{f(x_{l(k)})\}$ es no-creciente. De hecho, tomando en cuenta que $m(k+1) \leq m(k) + 1$, podemos escribir:

$$\begin{aligned}f(x_{l(k+1)}) &= \max_{0 \leq j \leq m(k+1)} [f(x_{k+1-j})] \\ &\leq \max_{0 \leq j \leq m(k)+1} [f(x_{k+1-j})] \\ &= \max [f(x_{l(k)}), f(x_{k+1})] = f(x_{l(k)}).\end{aligned}$$

Del Paso 6 se obtiene, para $k > M$:

$$\begin{aligned}
f(x_{l(k+1)}) &= f(x_{l(k+1)-1} + \lambda_{l(k+1)-1} d_{l(k+1)-1}) \\
&\leq \max_{0 \leq j \leq m(l(k)-1)} [f(x_{l(k)-1-j})] + \gamma \lambda_{l(k)-1} \nabla f(x_{l(k)-1})^t d_{l(k)-1} \quad (2.10) \\
&= f(x_{l(k)-1}) + \gamma \lambda_{l(k)-1} \nabla f(x_{l(k)-1})^t d_{l(k)-1}.
\end{aligned}$$

Ahora, como $f(x_k) \leq f(x_0)$ para todo k , $\{x_k\} \subset \Omega_0$ y Ω_0 es un conjunto compacto, entonces la sucesión $\{f(x_{l(k)})\}$ admite un punto límite. También, como $\lambda_k > 0$ y $\nabla f(x_k)^t d_k < 0$, se sigue de (2.10) que:

$$\lim_{k \rightarrow \infty} \lambda_{l(k)-1} \nabla f(x_{l(k)-1})^t d_{l(k)-1} = 0. \quad (2.11)$$

Por el Lema 2.4 existen constantes positivas c_1 y c_3 tales que

$$\lambda_k \nabla f(x_k)^t d_k \leq -c_3 \lambda_k \|\nabla f(x_k)\|^2 \leq -(c_3/c_1^2) \lambda_k \|d_k\|^2$$

para todo k , además, como $0 < \lambda_k \leq \max(\varepsilon^{-1}, \delta^{-1})$, (2.11) implica:

$$\lim_{k \rightarrow \infty} \lambda_{l(k)-1} \|d_{l(k)-1}\| = 0. \quad (2.12)$$

Probaremos ahora que $\lim_{k \rightarrow \infty} \lambda_k \|d_k\| = 0$. Sea

$$\hat{l}(k) = l(k + M + 2).$$

Primero mostraremos, por inducción, que para algún $i \geq 1$

$$\lim_{k \rightarrow \infty} \lambda_{\hat{l}(k)-i} \|d_{\hat{l}(k)-i}\| = 0 \quad (2.13)$$

y

$$\lim_{k \rightarrow \infty} f(x_{\hat{l}(k)-i}) = \lim_{k \rightarrow \infty} f(x_{l(k)}). \quad (2.14)$$

(Aquí y en lo siguiente asumiremos, sin pérdida de generalidad, que el índice $k \geq i - 1$.) Si $i = 1$, como $\{\hat{l}(k)\} \subset \{l(k)\}$, (2.13) se sigue de (2.12). Esto implica que $\|x_{\hat{l}(k)} - x_{\hat{l}(k)-1}\| \rightarrow 0$, lo cual indica que (2.14) se cumple, dado que $f(x)$ es uniformemente continua en Ω_0 . Asumamos ahora que (2.13) y (2.14) se cumplen para un cierto entero positivo i . Por el Paso 6 podemos escribir:

$$f(x_{\hat{l}(k)-i}) \leq f(x_{l(\hat{l}(k)-i-1)}) + \gamma \lambda_{\hat{l}(k)-i-1} \nabla f(x_{\hat{l}(k)-i-1})^t d_{\hat{l}(k)-i-1}.$$

Tomando límite para $k \rightarrow \infty$, se tiene, por (2.14):

$$\lim_{k \rightarrow \infty} \lambda_{\hat{l}(k)-(i+1)} \nabla f \left(x_{\hat{l}(k)-(i+1)} \right)^t d_{\hat{l}(k)-(i+1)} = 0.$$

Usando los mismos argumentos empleados para la derivación de (2.12) de (2.11), obtenemos:

$$\lim_{k \rightarrow \infty} \lambda_{\hat{l}(k)-(i+1)} \left\| d_{\hat{l}(k)-(i+1)} \right\| = 0.$$

Esto implica $\left\| x_{\hat{l}(k)-i} - x_{\hat{l}(k)-(i+1)} \right\| \rightarrow 0$, además, (2.14) se cumple y $f(x)$ es uniformemente continua en Ω_0 , por lo tanto:

$$\lim_{k \rightarrow \infty} f \left(x_{\hat{l}(k)-(i+1)} \right) = \lim_{k \rightarrow \infty} f \left(x_{\hat{l}(k)-i} \right) = \lim_{k \rightarrow \infty} f \left(x_{l(k)} \right).$$

Con esto concluimos que (2.13) y (2.14) se cumplen para algún $i \geq 1$.

Ahora para algún k :

$$x_{k+1} = x_{\hat{l}(k)} - \sum_{i=1}^{\hat{l}(k)-k-1} \lambda_{\hat{l}(k)-i} d_{\hat{l}(k)-i}. \quad (2.15)$$

Por la forma como definimos $l(k)$ y $\hat{l}(k)$ tenemos:

$$\hat{l}(k) - k - 1 = l(k + M + 2) - k - 1 \leq M + 1,$$

ahora (2.15) implica, por (2.13):

$$\lim_{k \rightarrow \infty} \left\| x_{k+1} - x_{\hat{l}(k)} \right\| = 0. \quad (2.16)$$

Como la sucesión $\{f(x_{l(k)})\}$ admite un límite, se sigue de la continuidad uniforme de $f(x)$ en Ω_0 que:

$$\lim_{k \rightarrow \infty} f(x_k) = \lim_{k \rightarrow \infty} f \left(x_{\hat{l}(k)} \right). \quad (2.17)$$

Por el Paso 6 tenemos:

$$f(x_{k+1}) \leq f(x_{l(k)}) + \gamma \lambda_k \nabla f(x_k)^t d_k.$$

Tomando límite para $k \rightarrow \infty$, por (2.17) obtenemos:

$$\lim_{k \rightarrow \infty} \lambda_k \nabla f(x_k)^t d_k = 0$$

lo cual implica, considerando el Lema 2.4, que:

$$\lim_{k \rightarrow \infty} \lambda_k \|d_k\| = 0 \quad (2.18)$$

y

$$\lim_{k \rightarrow \infty} \lambda_k \|\nabla f(x_k)\|^2 = 0. \quad (2.19)$$

Sea \bar{x} un punto de acumulación de $\{x_k\}$ y llamemos de nuevo $\{x_k\}$ a una subsucesión convergente a \bar{x} . La ecuación (2.19) es cierta si $\lim_{k \rightarrow \infty} \|\nabla f(x_k)\| = 0$, o si $\inf \lambda_k = 0$. Si $\lim_{k \rightarrow \infty} \|\nabla f(x_k)\| = 0$, entonces $J(\bar{x})^t F(\bar{x}) = 0$, pero $J(x)$ es no singular en Ω_0 , por lo tanto, $F(\bar{x}) = 0$ lo cual era lo que se deseaba demostrar. Si $\inf \lambda_k = 0$, entonces existe una subsucesión $\{x_k\}_K$ tal que

$$\lim_{k \in K} \lambda_k = 0.$$

En ese caso, por la forma en que λ_k es escogido en el Paso 6 del Algoritmo 2.2, existe un índice \bar{k} suficientemente grande tal que para todo $k \geq \bar{k}$ y $k \in K$, existe ρ_k ($0 < \sigma_1 \leq \rho_k \leq \sigma_2$) para el cual $\lambda_k/\rho_k > 0$ no satisface la búsqueda direccional no monótona, es decir,

$$\begin{aligned} f\left(x_k + \frac{\lambda_k}{\rho_k} d_k\right) &> \max_{0 \leq j \leq \min(k, M)} [f(x_{k-j})] + \gamma \frac{\lambda_k}{\rho_k} \nabla f(x_k)^t d_k \\ &\geq f(x_k) + \gamma \frac{\lambda_k}{\rho_k} \nabla f(x_k)^t d_k. \end{aligned}$$

Por lo tanto,

$$\frac{f\left(x_k + \frac{\lambda_k}{\rho_k} d_k\right) - f(x_k)}{\lambda_k/\rho_k} > \gamma \nabla f(x_k)^t d_k.$$

Mediante el Teorema del Valor Medio, esta relación se puede escribir como

$$\nabla f(x_k + t_k d_k)^t d_k > \gamma \nabla f(x_k)^t d_k, \text{ para todo } k \in K, k \geq \bar{k}, \quad (2.20)$$

donde t_k es un escalar en el intervalo $[0, \lambda_k/\rho_k]$ que tiende a cero cuando $k \in K$ tiende a infinito. Si consideramos una subsucesión conveniente tal que $d_k/\|d_k\| \rightarrow \bar{d}$ cuando $k \rightarrow \infty$, $k \in K$ y observando que $(x_k + t_k d_k) \rightarrow \bar{x}$ cuando $k \rightarrow \infty$, $k \in K$, entonces al dividir ambos miembros de la ecuación (2.20) por $\|d_k\|$ y luego tomando límite, se deduce que

$$(1 - \gamma) \nabla f(\bar{x})^t \bar{d} \geq 0.$$

Como $(1 - \gamma) > 0$ y $\nabla f(\bar{x})^t \bar{d} < 0$, entonces $\nabla f(\bar{x})^t \bar{d} = 0$, pero de (2.7) tenemos

$$-\nabla f(x_k)^t d_k = -2F_k^t J_k d_k \geq 2c_3 \|\nabla f(x_k)\|^2 \geq 0$$

para todo k , luego $\nabla f(\bar{x}) = J(\bar{x})^t F(\bar{x}) = 0$, pero $J(x)$ es no singular en Ω_0 , por lo tanto, $F(\bar{x}) = 0$ con lo cual se completa la prueba del teorema. \square

Para finalizar con el análisis del Algoritmo 2.2, mostraremos su convergencia, sin el Paso 2, cuando la parte simétrica de $J(x)$ es positivo (negativo) definida para todo $x \in \Omega_0$.

Corolario 2.1 *Bajo la hipótesis **A**, si $J_S(x) = (J(x) + J(x)^t)/2$ es positivo (negativo) definida para todo $x \in \Omega_0$, entonces el Algoritmo 2.2, sin el Paso 2, termina en una iteración finita i donde $F_i = 0$ o genera una sucesión $\{x_k\}$ tal que $\lim_{k \rightarrow \infty} \|F_k\| = 0$.*

Demostración. Asumamos sin pérdida de generalidad que $J_S(x)$ es positivo definida para todo $x \in \Omega_0$. Por continuidad y compacidad existe $\mu_{\min} > 0$ tal que $\mu_{\min} = \lambda_{\min}(J_S(x_k))$ para todo k , donde $\lambda_{\min}(J_S(x))$ representa el autovalor más pequeño de $J_S(x)$. De esta forma, como

$$F_k^t J_k F_k = F_k^t J_S(x_k) F_k \geq \lambda_{\min}(J_S(x_k)) \|F_k\|^2 \geq \mu_{\min} \|F_k\|^2 > 0,$$

para todo $k \geq 0$, entonces usando el Teorema 2.2 obtenemos que $\lim_{k \rightarrow \infty} \|F_k\| = 0$. \square

2.2.2 Algoritmo SANEF

Haciendo una revisión documental, encontramos que Li y Fukushima [39] utilizan la siguiente búsqueda lineal:

$$\|F(x_k + \lambda d_k)\| \leq (1 + \eta_k) \|F_k\| - \gamma \lambda^2 \|d_k\|^2, \quad (2.21)$$

donde $\gamma > 0$ y la sucesión de números positivos $\{\eta_k\}$ es tal que $\sum_{k=0}^{\infty} \eta_k \leq \eta < \infty$ para cierto $\eta > 0$. Esta búsqueda lineal tiende a forzar muy rápidamente a la monotonía, lo cual no encaja bien con la naturaleza no monótona de SANE. Por ello, combinar esta búsqueda lineal con SANE no funcionaría satisfactoriamente.

Con el objetivo de emplear la búsqueda lineal (2.21) con SANE, decidimos combinar (2.21) con (2.2) obteniendo la siguiente búsqueda lineal:

$$f(x_k + \lambda d_k) \leq f(x_k) + \eta_k \left(\max_{0 \leq j \leq \min(k, M)} [f(x_{k-j})] \right) - \gamma \lambda^2 \|d_k\|^2. \quad (2.22)$$

Cuando k tiende a infinito η_k tiende a cero, por lo tanto, la influencia del factor que produce la no monotonía en (2.22) disminuye cuando el número de iteraciones aumenta. Esto nos indica que a medida que aumente el número de iteraciones, (2.22) se convertirá en una búsqueda lineal monótona, pero no con tanta rapidez como (2.21). Al combinar (2.22) con SANE obtenemos el Algoritmo 2.3 que lo denominamos *SANEF*.

Algoritmo 2.3 (SANEF)

Dados $\alpha_0 > 0$, un entero $M > 0$, $\gamma > 0$, $0 < \sigma_1 < \sigma_2 < 1$, $0 < \varepsilon < 1$, $\delta \in [\varepsilon, 1/\varepsilon]$, $\eta > 0$ y una sucesión $\{\eta_k\}$ de números positivos tales que $\sum_{k=0}^{\infty} \eta_k \leq \eta < \infty$, tomar un punto inicial $x_0 \in \mathbb{R}^n$ y asignar $k = 0$.

Paso 1: Si $\|F_k\| = 0$, parar el proceso.

Paso 2: Si $|F_k^t J_k F_k| / F_k^t F_k < \varepsilon$, parar el proceso.

Paso 3: Si $\alpha_k \leq \varepsilon$ o $\alpha_k \geq 1/\varepsilon$, entonces asignar $\alpha_k = \delta$.

Paso 4: Tomar $d_k = -\text{sgn}(F_k^t J_k F_k) F_k$.

Paso 5: Asignar $\lambda = 1/\alpha_k$.

Paso 6: Si $f(x_k + \lambda d_k) \leq f(x_k) + \eta_k \left(\max_{0 \leq j \leq \min(k, M)} [f(x_{k-j})] \right) - \gamma \lambda^2 \|d_k\|^2$, entonces ir al Paso 8.

Paso 7: Escoger $\sigma \in [\sigma_1, \sigma_2]$, asignar $\lambda = \sigma \lambda$ e ir al Paso 6.

Paso 8: Asignar $\lambda_k = \lambda$, $x_{k+1} = x_k + \lambda_k d_k$, $y_k = F_{k+1} - F_k$ e ir al Paso 9.

Paso 9: Asignar $\alpha_{k+1} = -\left(\frac{F_k^t y_k}{\lambda_k F_k^t F_k} \right)$, hacer $k = k + 1$ e ir al Paso 1.

Veamos las características de SANEF. El Algoritmo 2.3 está bien definido, lo cual es una consecuencia de la continuidad de $F(x)$.

Lema 2.5 *El Algoritmo 2.3 está bien definido.*

El siguiente lema garantiza que la sucesión generada por el Algoritmo 2.3 está contenida en cierto conjunto de nivel.

Lema 2.6 *Si $\{x_k\}$ es una sucesión generada por el Algoritmo 2.3, entonces $\{x_k\}$ está contenida en el conjunto*

$$\Phi_0 = \{x : 0 \leq f(x) \leq e^n f(x_0)\}. \quad (2.23)$$

Demostración. Es claro que $f(x_k) \geq 0$ para todo k . Probemos que $f(x_k) \leq e^n f(x_0)$ para algún k . Para ello, primero probemos por inducción que

$$f(x_k) \leq f(x_0) \prod_{i=0}^{k-1} (1 + \eta_i). \quad (2.24)$$

Veamos que (2.24) se cumple para $k = 1$. Usando el Paso 6 del Algoritmo 2.3 podemos escribir:

$$\begin{aligned} f(x_1) &\leq f(x_0) + \eta_0 \left(\max_{0 \leq j \leq \min(0, M)} [f(x_{0-j})] \right) \\ &= f(x_0) + \eta_0 f(x_0) = f(x_0)(1 + \eta_0). \end{aligned}$$

Ahora supongamos que la ecuación (2.24) se cumple para $0 \leq k < l$ con $l \geq 2$ y veamos que se cumple para $k \geq l$. Empleando el Paso 6 del Algoritmo 2.3 y (2.24) podemos escribir, para $k \geq l$:

$$\begin{aligned}
f(x_k) &\leq f(x_{k-1}) + \eta_{k-1} \left(\max_{0 \leq j \leq \min(k-1, M)} [f(x_{k-1-j})] \right) \\
&\leq f(x_0) \prod_{i=0}^{k-2} (1 + \eta_i) + \eta_{k-1} \left(\max_{0 \leq j \leq \min(k-1, M)} \left[f(x_0) \prod_{i=0}^{k-2-j} (1 + \eta_i) \right] \right) \\
&= f(x_0) \prod_{i=0}^{k-2} (1 + \eta_i) + \eta_{k-1} f(x_0) \prod_{i=0}^{k-2} (1 + \eta_i) \\
&= f(x_0) \prod_{i=0}^{k-2} (1 + \eta_i) (1 + \eta_{k-1}) \\
&= f(x_0) \prod_{i=0}^{k-1} (1 + \eta_i),
\end{aligned}$$

con lo cual se demuestra que se satisface (2.24). De esta forma, como se cumple (2.24) podemos escribir:

$$\begin{aligned}
f(x_{k+1}) &\leq f(x_0) \prod_{i=0}^k (1 + \eta_i) \\
&\leq f(x_0) \left(\frac{1}{k+1} \sum_{i=0}^k (1 + \eta_i) \right)^{k+1} \\
&= f(x_0) \left(1 + \frac{1}{k+1} \sum_{i=0}^k \eta_i \right)^{k+1} \\
&\leq f(x_0) \left(1 + \frac{\eta}{k+1} \right)^{k+1} \\
&\leq e^\eta f(x_0).
\end{aligned}$$

Esto completa la prueba. □

El siguiente lema corresponde al Lema 3.3 en [20].

Lema 2.7 Sean $\{a_k\}$ y $\{r_k\}$ sucesiones de números positivos que satisfacen:

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

Entonces $\{a_k\}$ converge.

Para continuar con el análisis de SANEF necesitamos la siguiente hipótesis.

Hipótesis B.

- (i) El conjunto de nivel Φ_0 dado en (2.23) es acotado.
- (ii) $F(x)$ es continuamente diferenciable en un conjunto abierto que contiene a Φ_0 .
- (iii) $J(x)$ es no singular en todo $x \in \Phi_0$.

Para establecer la convergencia global del Algoritmo 2.3 necesitamos los siguientes cuatro lemas.

Lema 2.8 *Bajo la hipótesis B, si $\{x_k\}$ es la sucesión generada por el Algoritmo 2.3, entonces existen constantes positivas a_1 , a_2 y a_3 tales que*

$$\|d_k\| \leq a_1 \|\nabla f(x_k)\|, \quad (2.25)$$

$$\|\nabla f(x_k)\| \leq a_2 \|d_k\|, \quad (2.26)$$

$$F_k^t J_k d_k \leq -a_3 \|\nabla f(x_k)\|^2, \quad (2.27)$$

para todo k .

Demostración. Sean L_1 y L_2 constantes positivas tales que $\|J(x)^{-1}\| \leq L_1$ y $\|J(x)\| \leq L_2$, para todo $x \in \Phi_0$. Como $\|d_k\| = \|F_k\|$ y $F_k = \frac{1}{2} J_k^{-t} \nabla f(x_k)$, entonces

$$\begin{aligned} \|d_k\| &\leq \frac{1}{2} \|J_k^{-1}\| \|\nabla f(x_k)\| \\ &\leq \frac{L_1}{2} \|\nabla f(x_k)\|, \end{aligned}$$

es decir, $a_1 = L_1/2 > 0$.

Bajo el mismo orden de ideas, como $\nabla f(x_k) = 2J_k^t F_k$, entonces

$$\begin{aligned} \|\nabla f(x_k)\| &\leq 2\|J_k\| \|F_k\| \\ &= 2\|J_k\| \|d_k\| \\ &\leq 2L_2 \|d_k\|, \end{aligned}$$

es decir, $a_2 = 2L_2 > 0$.

Por último, por el Paso 2 del Algoritmo 2.3 obtenemos que:

$$|F_k^t J_k F_k| \geq \varepsilon \|F_k\|^2. \quad (2.28)$$

Del Paso 4 del Algoritmo 2.3 se tiene que $d_k = -\text{sgn}(F_k^t J_k F_k) F_k$, luego

$$F_k^t J_k d_k = F_k^t J_k (-\text{sgn}(F_k^t J_k F_k) F_k) = \begin{cases} -F_k^t J_k F_k, & F_k^t J_k F_k > 0 \\ F_k^t J_k F_k, & F_k^t J_k F_k < 0. \end{cases} \quad (2.29)$$

Usando (2.28), (2.29), $\|d_k\| = \|F_k\|$ y (2.26), podemos escribir:

$$F_k^t J_k d_k \leq -\varepsilon \|F_k\|^2 = \varepsilon (-\|d_k\|^2) \leq \varepsilon (-a_2^{-2} \|\nabla f(x_k)\|^2) = -(\varepsilon a_2^{-2}) \|\nabla f(x_k)\|^2,$$

en consecuencia, $a_3 = \varepsilon a_2^{-2} > 0$. □

Lema 2.9 *Bajo la hipótesis B, si $\{x_k\}$ es la sucesión generada por el Algoritmo 2.3, entonces*

$$\sum_{k=0}^{\infty} \|x_{k+1} - x_k\|^2 < \infty.$$

Demostración. Por el Paso 6 se tiene, para $k \geq 0$:

$$f(x_{k+1}) \leq f(x_k) + \eta_k \left(\max_{0 \leq j \leq \min(k, M)} [f(x_{k-j})] \right) - \gamma \lambda_k^2 \|d_k\|^2.$$

De esta forma, usando esta desigualdad, $x_{k+1} = x_k + \lambda_k d_k$ y el Lema 2.6, podemos escribir:

$$\begin{aligned} \|x_{k+1} - x_k\|^2 &= \lambda_k^2 \|d_k\|^2 \\ &\leq \frac{\eta_k}{\gamma} \left(\max_{0 \leq j \leq \min(k, M)} [f(x_{k-j})] \right) + \frac{1}{\gamma} (f(x_k) - f(x_{k+1})) \\ &\leq \frac{\eta_k}{\gamma} e^\eta f(x_0) + \frac{1}{\gamma} (f(x_k) - f(x_{k+1})). \end{aligned}$$

Sumando obtenemos:

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

que era lo que deseabamos demostrar. □

Lema 2.10 *Bajo la hipótesis B, si $\{x_k\}$ es la sucesión generada por el Algoritmo 2.3, entonces la sucesión $\{\|F_k\|\}$ converge.*

Demostración. Como $f(x_k) \leq f(x_0)e^\eta$ y $(1 + e^\eta\eta_k) \geq 1$ para todo k , por el Paso 6 podemos escribir, para $k \geq 0$:

$$\begin{aligned} f(x_{k+1}) &\leq f(x_k) + \eta_k \left(\max_{0 \leq j \leq \min(k, M)} [f(x_{k-j})] \right) \\ &\leq f(x_k) + f(x_0)e^\eta\eta_k \\ &\leq f(x_k) (1 + e^\eta\eta_k) + \eta_k f(x_0)e^\eta \\ &\leq f(x_k) (1 + m_0 e^\eta\eta_k) + m_0 e^\eta\eta_k, \end{aligned}$$

donde $m_0 = \max[f(x_0), 1]$. Luego, si tomamos $a_k = \|F_k\|^2$ y $r_k = m_0 e^\eta\eta_k > 0$, entonces

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

además,

$$\sum_{k=0}^{\infty} r_k = m_0 e^\eta \sum_{k=0}^{\infty} \eta_k < m_0 e^\eta \eta < \infty,$$

por lo tanto, por el Lema 2.7 la sucesión $\{a_k\}$ converge; en otras palabras, la sucesión $\{\|F_k\|^2\}$ converge. Como $\|F_k\| \geq 0$ y $\{\|F_k\|^2\}$ converge, entonces la sucesión $\{\|F_k\|\}$ converge. \square

Lema 2.11 *Bajo la hipótesis B, si $\{x_k\}$ es la sucesión generada por el Algoritmo 2.3, entonces*

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

Demostración. Por el Lema 2.9 tenemos que

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

además, $x_{k+1} - x_k = \lambda_k d_k$, luego

$$\lim_{k \rightarrow \infty} \lambda_k \|d_k\| = 0. \tag{2.30}$$

Por el Lema 2.8 existen constantes positivas a_1 y a_2 tales que

$$\|d_k\| \leq a_1 \|\nabla f(x_k)\| \leq (a_1 a_2) \|d_k\|, \tag{2.31}$$

para todo k . Usando (2.30), (2.31) y dado que $\lambda_k > 0$ para todo k , se deduce que:

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

que era lo que deseabamos demostrar. \square

Nuestro primer teorema establece que SANEF termina prematuramente con $F_i = 0$ o $F_i^t J_i F_i = 0$, o converge para algún iterado inicial.

Teorema 2.3 *Bajo la hipótesis **B**, el Algoritmo 2.3 termina en una iteración finita i donde $F_i = 0$ o $|F_i^t J_i F_i| < \varepsilon \|F_i\|^2$, o genera una sucesión $\{x_k\}$ tal que*

$$\lim_{k \rightarrow \infty} \|F_k\| = 0.$$

Demostración. Por el Lema 2.10 la sucesión $\{\|F_k\|\}$ converge. Por lo tanto, para demostrar que $\{\|F_k\|\}$ converge a 0, es suficiente demostrar que $F(\bar{x}) = 0$ si \bar{x} es un punto de acumulación de $\{x_k\}$.

Sea \bar{x} un punto de acumulación de $\{x_k\}$ y llamemos de nuevo $\{x_k\}$ a una subsucesión convergente a \bar{x} . Por el Lema 2.11 se tiene que

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

lo cual es cierto si $\lim_{k \rightarrow \infty} \|\nabla f(x_k)\| = 0$, o si $\inf \lambda_k = 0$. Si $\lim_{k \rightarrow \infty} \|\nabla f(x_k)\| = 0$, entonces $\nabla f(\bar{x}) = 2J(\bar{x})^t F(\bar{x}) = 0$, pero $J(x)$ es no-singular en Φ_0 , luego $F(\bar{x}) = 0$, con lo cual se demuestra que la conclusión es cierta. Si $\inf \lambda_k = 0$, entonces existe una subsucesión $\{x_k\}_K$ tal que

$$\lim_{k \in K} \lambda_k = 0.$$

En ese caso, por la forma en que λ_k es escogido en el Paso 6 del Algoritmo 2.3, existe un índice \bar{k} suficientemente grande tal que para todo $k \geq \bar{k}$ y $k \in K$, existe ρ_k ($0 < \sigma_1 \leq \rho_k \leq \sigma_2$) para el que $\lambda_k/\rho_k > 0$ no satisface la búsqueda direccional no monótona, es decir,

$$\begin{aligned} f\left(x_k + \frac{\lambda_k}{\rho_k} d_k\right) &> f(x_k) + \eta_k \left(\max_{0 \leq j \leq \min(k, M)} [f(x_{k-j})] \right) - \gamma \frac{\lambda_k^2}{\rho_k^2} \|d_k\|^2 \\ &\geq f(x_k) - \gamma \frac{\lambda_k^2}{\rho_k^2} \|d_k\|^2. \end{aligned}$$

Por lo tanto,

$$\frac{f\left(x_k + \frac{\lambda_k}{\rho_k} d_k\right) - f(x_k)}{\lambda_k/\rho_k} > \left(-\gamma \frac{\lambda_k}{\rho_k} \|d_k\|^2 \right) \geq \left(-\gamma \frac{\lambda_k}{\sigma_2} \|d_k\|^2 \right).$$

Mediante el Teorema del Valor Medio, esta relación se puede escribir como

$$\nabla f(x_k + t_k d_k)^t d_k > \left(-\gamma \frac{\lambda_k}{\sigma_2} \|d_k\|^2 \right), \text{ para todo } k \in K, k \geq \bar{k}, \quad (2.32)$$

donde t_k es un escalar en el intervalo $[0, \lambda_k/\rho_k]$ que tiende a cero cuando $k \in K$ tiende a infinito. Si consideramos una subsucesión conveniente tal que $d_k \rightarrow \bar{d}$ cuando $k \rightarrow \infty$, $k \in K$ y observando que $(x_k + t_k d_k) \rightarrow \bar{x}$ cuando $k \rightarrow \infty$, $k \in K$, entonces tomando límite en (2.32) se deduce que

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

Como $\nabla f(\bar{x})^t \bar{d} \leq 0$, entonces $\nabla f(\bar{x})^t \bar{d} = 0$, pero de (2.27) obtenemos

$$-\nabla f(x_k)^t d_k = -2F_k^t J_k d_k \geq 2a_3 \|\nabla f(x_k)\|^2 \geq 0$$

para todo k , luego $\nabla f(\bar{x}) = 2J(\bar{x})^t F(\bar{x}) = 0$, pero $J(x)$ es no-singular en Φ_0 , por lo tanto $F(\bar{x}) = 0$, con lo cual se completa la prueba del teorema. \square

Para finalizar, mostraremos la convergencia fuerte del Algoritmo 2.3, sin el Paso 2, cuando la parte simétrica de $J(x)$ es positivo (negativo) definida para todo $x \in \Phi_0$.

Corolario 2.2 *Bajo la hipótesis **A**, si $J_S(x) = (J(x) + J(x)^t)/2$ es positivo (negativo) definida para todo $x \in \Phi_0$, entonces el Algoritmo 2.3, sin el Paso 2, termina en una iteración finita i donde $F_i = 0$ o genera una sucesión $\{x_k\}$ tal que $\lim_{k \rightarrow \infty} \|F_k\| = 0$*

Demostración. Asumamos sin pérdida de generalidad que $J_S(x)$ es positivo definida para todo $x \in \Phi_0$. Por continuidad y compacidad existe $\mu_{\min} > 0$ tal que $\mu_{\min} = \lambda_{\min}(J_S(x_k))$ para todo k , donde $\lambda_{\min}(J_S(x))$ representa el autovalor más pequeño de $J_S(x)$. De esta forma, como

$$F_k^t J_k F_k = F_k^t J_S(x_k) F_k \geq \lambda_{\min}(J_S(x_k)) \|F_k\|^2 \geq \mu_{\min} \|F_k\|^2 > 0,$$

para todo $k \geq 0$, entonces usando el Teorema 2.3 obtenemos $\lim_{k \rightarrow \infty} \|F_k\| = 0$. \square

2.2.3 Comparación numérica de SANEG y SANEF

Para estudiar el comportamiento computacional de los algoritmos SANEG y SANEF, consideraremos dos funciones muy particulares que a continuación se definen, además, también se dan los iterados iniciales.

- *Función exponencial 3:* $F_1(x) = (g_1(x), \dots, g_n(x))^t$, donde

$$\begin{aligned} g_i(x) &= \frac{i}{10} \left(1 - x_i^2 - e^{-x_i^2}\right), \quad \text{para } i = 2, 3, \dots, n-1, \\ g_n(x) &= \frac{n}{10} \left(1 - e^{-x_n^2}\right), \\ x_0 &= \left(\frac{1}{4n^2}, \frac{2}{4n^2}, \dots, \frac{n}{4n^2}\right)^t. \end{aligned}$$

- *Función estrictamente convexa 2* [52, p. 30]: $F_2(x) = (g_1(x), \dots, g_n(x))^t$ es el gradiente de la función estrictamente convexa $h(x) = \sum_{i=1}^n \frac{i}{10} (e^{x_i} - x_i)$,

$$\begin{aligned} g_i(x) &= \frac{i}{10} (e^{x_i} - 1), \quad \text{para } i = 1, 2, \dots, n, \\ x_0 &= (1, 1, \dots, 1)^t. \end{aligned}$$

Para estos experimentos usamos MATLAB 6.1 en una computadora personal Pentium II de 350 MHz. Los valores de los parámetros de los Algoritmos 2.2 y 2.3 que usamos son: $\gamma = 10^{-4}$, $\varepsilon = 10^{-8}$, $\sigma_1 = 0.1$, $\sigma_2 = 0.5$, $\alpha_0 = 1$, $M = 10$,

$$\delta = \begin{cases} 1 & \text{si } \|F_k\| \geq 1, \\ \|F_k\| & \text{si } 10^{-5} \leq \|F_k\| \leq 1, \\ 10^{-5} & \text{si } \|F_k\| < 10^{-5}, \end{cases} \quad (2.33)$$

y $\eta_k = 1/(k+1)^2$ para el Algoritmo 2.3. El escalar $F_k^t J_k F_k$ lo calculamos por (2.3), con $h = 10^{-7}$.

Para la escogencia de λ en el Paso 7 de los Algoritmos 2.2 y 2.3, empleamos el siguiente modelo parabólico de dos puntos (Kelley [38, pp. 142-143]). Denotando con $\lambda_c > 0$ el valor actual de λ , tomamos λ como

$$\lambda = \begin{cases} \sigma_1 \lambda_c & \text{si } \lambda_t < \sigma_1 \lambda_c, \\ \sigma_2 \lambda_c & \text{si } \lambda_t > \sigma_2 \lambda_c, \\ \lambda_t & \text{de lo contrario,} \end{cases} \quad (2.34)$$

donde

$$\lambda_t = \frac{-\lambda_c^2 F(x_k)^t J(x_k) F(x_k)}{\|F(x_k + \lambda_c d_k)\|^2 - \|F(x_k)\|^2 - 2F(x_k)^t J(x_k) F(x_k) \lambda_c}. \quad (2.35)$$

En todos los experimentos los algoritmos 2.2 y 2.3 pararon el proceso cuando

$$\frac{\|F(x_k)\|}{\sqrt{n}} \leq e_a + e_r \frac{\|F(x_0)\|}{\sqrt{n}}, \quad (2.36)$$

donde $e_a = 5 \times 10^{-7}$ y $e_r = 5 \times 10^{-7}$.

Un método falla cuando se tiene alguna de las siguientes situaciones:

- el número de iteraciones es mayor o igual a 500;
- el número de backtracking en una búsqueda lineal es mayor o igual a 100;
- si para alguna iteración k se cumple: $|F_k^t J_k F_k| / F_k^t F_k < \varepsilon$.

Función (n)	SANEG					SANEF				
	IT	F	BK	T	NF	IT	F	BK	T	NF
1(100)	93	192	5	0.121	4.8e-6	516	2977	414	1.482	5.0e-6
2(100)	60	135	13	0.065	5.3e-5	262	893	246	0.366	5.4e-5

Tabla 2.1: Iteraciones, evaluaciones de $F(x)$, búsquedas lineales, tiempo de CPU y $\|F(x^*)\|$, para SANEG y SANEF.

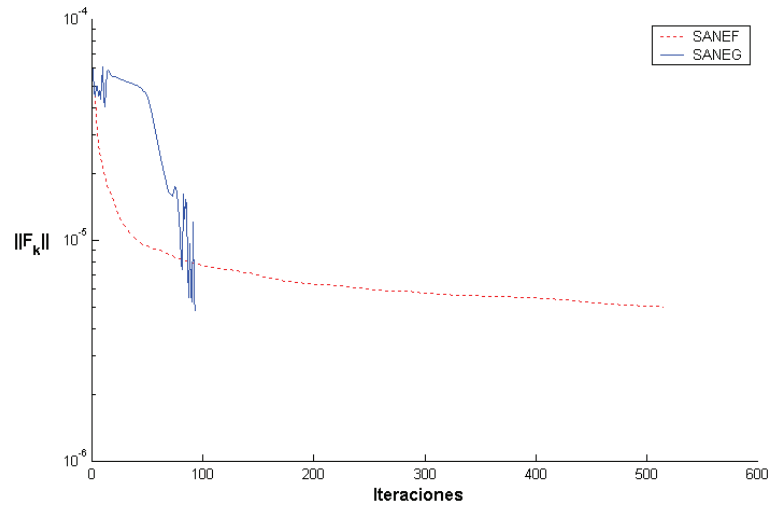


Figura 2.2: Comportamiento de SANEG y SANEF para $F_1(x)$.

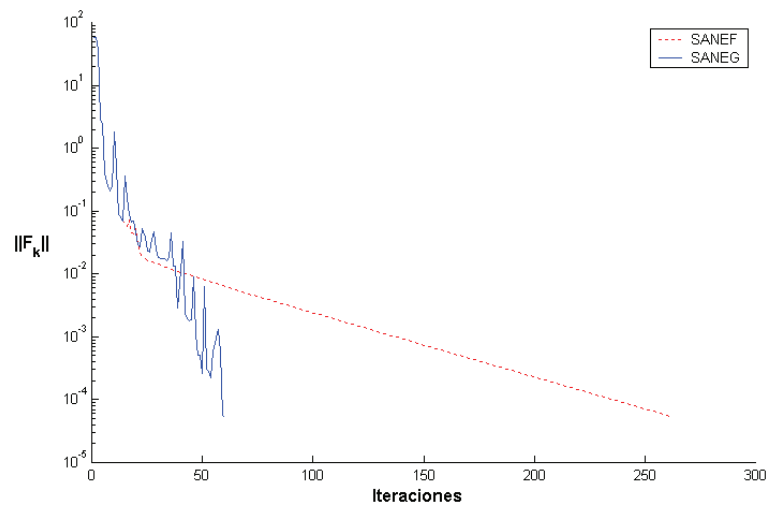


Figura 2.3: Comportamiento de SANEG y SANEF para $F_2(x)$.

Para $n = 100$, los resultados numéricos se muestran en la Tabla 2.1 y los comportamientos gráficos de los algoritmos se observan en las Figuras 2.2 y 2.3. Reportamos el número de iteraciones (IT), número de evaluaciones de la función $F(x)$ (F), número de backtrakings (BK), tiempo de cómputo expresado en segundos (T) y $\|F(x_k)\|$ (NF), donde x_k es el último iterado generado por el algoritmo.

En la Figura 2.2 observamos que en las primeras 100 iteraciones SANEF disminuye a $\|F_k\|$ más rápidamente que SANEG, pero en forma monótona. A partir de la iteración 110, SANEG disminuye a $\|F_k\|$ más rápidamente que SANEF, el cual se torna monótono y lento. En la Figura 2.3 observamos que en las primeras 20 iteraciones SANEG y SANEF se comportan igual, pero al aumentar las iteraciones SANEG es más eficiente y su comportamiento es no monótono, en cambio, SANEF es monótono y lento.

Los resultados numéricos obtenidos prueban que SANEG es más eficiente computacionalmente que SANEF. Esto se debe, como mencionamos anteriormente, a que la búsqueda lineal (2.22) fuerza a la monotonía, que no es muy conveniente cuando se utiliza en combinación con SANE. En la próxima sección realizaremos un estudio comparativo de la eficiencia numérica de SANEG con respecto a métodos tipo Newton Inexactos y el método de Broyden.

2.3 Comparación numérica de SANEG con algoritmos Newton Inexactos y el algoritmo de Broyden

Compararemos la eficiencia computacional de SANEG con los siguientes métodos iterativos:

- *Newton-GMRES*;
- *Newton-Bi-CGSTAB*;
- *Newton-TFQMR*;
- *Broyden Globalizado (Broyden)*. Broyden con la regla de Armijo. Broyden posee código en MATLAB dado por Kelley [38].

Resolvimos 100 sistemas de ecuaciones no lineales, $F(x) = 0$, construidos con 50 funciones $F(x)$ que se muestran en el **Anexo A** denominado **Funciones de Prueba**. En este anexo también se muestran los iterados iniciales, x_0 , utilizados.

Empleamos los índices de robustez, eficiencia, combinación de robustez y eficiencia ([11]) y calidad de solución, para coleccionar los datos para la comparación.

(1) *Índice de robustez*

$$R_A = \frac{a_A}{r}, \quad (2.37)$$

donde a_A es el número de problemas resueltos por el algoritmo A y r es el mayor número de problemas resueltos por alguno de los algoritmos probados. R_A es el porcentaje de casos en los que el algoritmo A encuentra una solución.

(2) *Índice de eficiencia*

$$E_A = \frac{\sum_{i=1}^m e_i}{a_A}, \quad (2.38)$$

donde

$$e_i = \begin{cases} 0, & \text{si el algoritmo } A \text{ falla en el problema } i, \\ 1, & \text{si } t_{ib} = 0 \text{ y } t_{iA} = 0, \\ t_{ib}/t_{iA}, & \text{si } t_{iA} \neq 0, \end{cases}$$

donde t_{iA} es el tiempo de CPU empleado por el algoritmo A en la resolución del problema i , $t_{ib} = \min_A[t_{iA}]$, es decir, es el mejor resultado entre todos los resultados de los algoritmos para el problema i , y m es el total de problemas a resolver.

(3) *Combinación de robustez y eficiencia*

$$ER_A = E_A \times R_A. \quad (2.39)$$

(4) *Índice de calidad de solución*

$$C_A = \frac{\sum_{i=1}^m c_i}{a_A}, \quad (2.40)$$

donde

$$c_i = \begin{cases} 0, & \text{si el algoritmo } A \text{ falla en el problema } i, \\ 1, & \text{si } N_{ib} = 0 \text{ y } N_{iA} = 0, \\ N_{ib}/N_{iA}, & \text{si } N_{iA} \neq 0, \end{cases}$$

donde $N_{iA} = \|F(x_k)\|$ con x_k el último iterado obtenido por el algoritmo A para el problema i y $N_{ib} = \min_A[N_{iA}]$, es decir, la mejor solución obtenida por alguno de los algoritmos para el problema i .

Para Newton-GMRES, Newton-Bi-CGSTAB, Newton-TFQMR y Broyden, tomamos los valores de todos los parámetros que están muy bien descritos en Kelley [38, pp. 143-144]. Para SANEG usamos $\gamma = 10^{-4}$, $\varepsilon = 10^{-10}$, $\sigma_1 = 0.1$, $\sigma_2 = 0.5$, $\alpha_0 = 1$, $M = 10$ y δ como en (2.33). Utilizamos (2.3), con $h = 10^{-7}$, para aproximar al escalar $F_k^t J_k F_k$. Para la escogencia de λ en el Paso 7, empleamos el modelo parabólico de dos puntos dado por (2.34) y (2.35).

Los tests numéricos fueron realizados en un computador personal Pentium II de 350 MHz con una precisión de máquina $\simeq 10^{-16}$. Empleamos el criterio de parada (2.36) con las siguientes consideraciones: (i) para las funciones 1 hasta la 47, utilizamos $e_a = 10^{-5}$ y $e_r = 10^{-6}$; y (ii) para las funciones 48, 49 y 50, empleamos $e_a = 5 \times 10^{-4}$ y $e_r = 10^{-10}$. Un método falla, y usamos el símbolo (*) para denotarlo, cuando alguna de las siguientes situaciones es cierta:

- (a) el número de iteraciones es mayor o igual a 500;
- (b) el número de backtracking en una búsqueda lineal es mayor o igual a 100.

El algoritmo SANEG también puede fallar si para alguna iteración k se cumple:

$$|F_k^t J_k F_k| / F_k^t F_k < \varepsilon.$$

La Tabla 2.2 muestra los índices de robustez, eficiencia, combinación de robustez y eficiencia, y calidad de solución. Los resultados numéricos se muestran en forma detallada en las Tablas 2.3, 2.4, 2.5, 2.6 y 2.7. Reportamos el número de iteraciones (IT), número de evaluaciones de la función $F(x)$ (F), número de búsquedas lineales requeridas (BL), tiempo de cómputo expresado en segundos (T) y $\|F(x_k)\|$ (NF), donde x_k es el último iterado obtenido por el algoritmo.

La Tabla 2.2 nos indica que SANEG es el algoritmo *más robusto* ($R_A = 1$) de los estudiados, ya que de 100 problemas no falló en ninguno de ellos. En cuanto a la eficiencia, SANEG resultó ser el algoritmo *más eficiente* con un índice de eficiencia de $E_A = 0.7929$.

Es claro que la combinación de la robustez y la eficiencia de un algoritmo determinan la confiabilidad del mismo. Por tanto, según los resultados mostrados en la Tabla 2.2, SANEG posee el valor más alto del índice de combinación de eficiencia y robustez ($ER_A = 0.7929$). Por ello, podemos asegurar que SANEG es el algoritmo *más confiable* para los problemas tratados. También apreciamos en la Tabla 2.2 que la calidad de la solución de

SANEG es la menor, pero no sustancialmente distante de la calidad de las soluciones de los otros algoritmos.

Índices	Newton-GMRES	Newton-Bi-CGSTAB	Newton-TFQMR	Broyden	SANEG
R_A	0.9500	0.9400	0.9100	0.4400	1.0000
E_A	0.5525	0.6357	0.6049	0.6073	0.7929
ER_A	0.5249	0.5975	0.5504	0.3689	0.7929
C_A	0.5183	0.5584	0.5577	0.5616	0.5085

Tabla 2.2: Índices

En cuanto al número de iteraciones, es muy importante resaltar que el número de iteraciones de SANEG es mayor que el de los otros algoritmos. Por último, SANEG requiere, generalmente, de un mayor número de búsquedas lineales.

Función(n)	IT	F	BL	T	NF	Función(n)	IT	F	BL	T	NF
1(1000)	3	16	0	0.100	0.000295	26(250)	11	235	9	0.120	1.52e-5
1(10000)	1	3	0	0.190	0.00095	26(1000)	7	103	4	0.240	0.000218
2(500)	7	318	0	3.716	0.000199	27(100)	4	27	0	0.030	0.000106
2(2000)	7	318	0	9.704	0.000338	27(1000)	2	9	0	0.060	1.89e-5
3(100)	10	186	0	0.140	6.25e-5	28(1000)	*	*	*	*	*
3(500)	11	226	0	0.320	0.000217	28(10000)	*	*	*	*	*
4(99)	14	462	23	0.271	5.19e-5	29(50)	13	195	0	2.203	5.4e-5
4(999)	14	462	23	1.522	0.000165	29(500)	13	195	0	79.324	0.000274
5(99)	3	19	0	0.040	1.52e-10	30(100)	1	3	0	0.020	9.78e-6
5(999)	3	19	0	0.130	4.82e-10	30(1000)	1	3	0	0.020	3.09e-5
6(100)	6	51	0	0.050	8.48e-11	31(100)	11	155	0	0.070	0.00456
6(10000)	6	51	0	1.292	8.24e-10	31(1000)	10	122	0	0.250	0.00973
7(100)	5	47	1	0.050	8.28e-7	32(99)	3	17	0	0.040	4.3e-5
7(10000)	5	47	1	1.402	8.28e-6	32(9999)	3	17	0	0.851	0.000432
8(1000)	5	36	0	0.150	7.04e-11	33(500)	5	49	2	0.131	7.93e-5
8(10000)	5	36	0	1.062	2.23e-10	33(1000)	6	66	2	0.250	1.23e-5
9(100)	4	27	0	0.190	5.03e-6	34(1000)	4	24	0	0.140	9.48e-8
9(1000)	4	27	0	11.106	1.6e-5	34(5000)	4	24	0	0.531	2.12e-7
10(100)	20	1449	75	0.290	0.000104	35(5000)	2	8	0	0.230	6.13e-5
10(5000)	20	1449	75	4.737	0.000737	35(10000)	1	3	0	0.221	0.000644
11(99)	5	65	11	0.240	0.000606	36(1000)	30	2634	39	2.714	0.0148
11(399)	5	65	11	0.761	0.00122	36(5000)	29	2441	36	10.915	0.0723
12(1000)	*	*	*	*	*	37(1000)	12	227	3	0.882	0.0216
12(10000)	5	74	1	5.939	0.000793	37(5000)	14	297	3	3.455	0.0438
13(100)	4	47	5	0.050	0.00569	38(1000)	14	328	5	1.121	0.0176
13(1000)	5	63	5	0.220	0.0019	38(5000)	13	258	2	3.946	0.0827
14(2500)	11	206	0	1.322	0.0161	39(1000)	4	27	0	0.140	0.00102
14(10000)	11	208	0	4.757	0.134	39(5000)	4	27	0	0.601	0.00227
15(5000)	4	24	0	0.441	0.000532	40(1000)	11	246	2	0.541	0.000267
15(15000)	4	24	0	1.151	0.000919	40(5000)	11	246	2	2.133	0.000597
16(500)	6	70	0	0.251	0.000101	41(1000)	5	40	0	0.190	0.000221
16(2000)	6	70	0	0.611	0.000157	41(5000)	5	40	0	0.831	0.000495
17(100)	6	65	1	0.080	5.47e-5	42(1000)	3	15	0	0.110	1.12e-5
17(1000)	6	64	1	0.360	0.000102	42(5000)	3	15	0	0.370	2.85e-6
18(100)	6	66	1	0.201	2.25e-5	43(500)	4	30	0	0.160	0.000107
18(1000)	6	71	2	0.741	0.000144	43(1000)	2	8	0	0.081	0.000298
19(100)	6	66	1	0.140	4.79e-5	44(100)	7	105	0	0.110	0.00116
19(1000)	6	72	2	0.791	6.51e-5	44(500)	7	105	0	0.300	0.00116
20(500)	6	78	2	0.241	5.29e-5	45(500)	14	228	0	0.331	0.191
20(1000)	6	76	2	0.320	9.58e-5	45(1000)	14	240	0	0.591	0.0735
21(1000)	4	24	0	0.100	1.35e-6	46(1000)	2	8	0	0.440	0.000296
21(50000)	4	24	0	3.615	9.52e-6	46(5000)	2	8	0	2.103	0.00014
22(100)	8	220	0	0.331	8.32e-5	47(999)	6	60	0	0.290	8.08e-5
22(1000)	8	336	0	4.837	0.0021	47(4999)	6	59	0	1.212	0.000361
23(399)	*	*	*	*	*	48(500)	6	130	0	0.981	0.00898
23(9999)	*	*	*	*	*	48(1000)	5	64	0	0.802	0.00836
24(1000)	1	3	0	0.030	0.00138	49(100)	5	97	0	0.470	0.00385
24(15000)	2	8	0	0.550	3.04e-7	49(500)	5	88	0	0.972	0.00547
25(500)	1	3	0	0.030	0.000572	50(33)	7	173	2	0.300	0.00248
25(5000)	1	3	0	0.111	0.029	50(133)	6	110	1	0.311	0.00887

Tabla 2.3: Funciones de prueba: Newton-GMRES.

Función(n)	IT	F	BL	T	NF	Función(n)	IT	F	BL	T	NF
1(1000)	4	34	0	0.090	0.000101	26(250)	10	215	5	0.070	1.16e-5
1(10000)	1	4	0	0.180	0.000846	26(1000)	7	128	4	0.170	4.29e-5
2(500)	9	1328	10	1.983	0.000225	27(100)	6	107	0	0.040	0.000343
2(2000)	11	1889	10	6.740	0.00045	27(1000)	4	46	0	0.121	0.00675
3(100)	10	287	0	0.090	5.41e-5	28(1000)	*	*	*	*	*
3(500)	10	355	0	0.470	0.00022	28(10000)	*	*	*	*	*
4(99)	7	132	2	0.101	4.02e-5	29(50)	13	286	0	3.234	5.4e-5
4(999)	7	132	2	0.691	0.000128	29(500)	13	286	0	119.341	0.000274
5(99)	3	29	0	0.040	4.48e-8	30(100)	1	4	0	0.010	9.78e-6
5(999)	3	29	0	0.130	8.16e-8	30(1000)	1	4	0	0.030	3.09e-5
6(100)	7	255	0	0.140	6.23e-5	31(100)	10	175	0	0.050	0.00954
6(10000)	7	255	0	5.658	0.000622	31(1000)	10	175	0	0.210	0.00954
7(100)	5	70	1	0.030	6.09e-7	32(99)	3	25	0	0.030	4.19e-7
7(10000)	5	70	1	1.372	6.12e-6	32(9999)	3	25	0	0.891	4.21e-6
8(1000)	4	34	0	0.090	1.12e-6	33(500)	7	106	1	0.161	3.04e-5
8(10000)	4	34	0	0.741	3.54e-6	33(1000)	7	106	1	0.250	6.75e-5
9(100)	4	34	0	0.210	7.56e-5	34(1000)	4	34	0	0.130	9.48e-8
9(1000)	4	34	0	13.469	0.000239	34(5000)	4	34	0	0.541	2.12e-7
10(100)	45	8775	123	0.531	8.99e-5	35(5000)	2	11	0	0.190	6.13e-5
10(5000)	45	8775	123	11.136	0.000635	35(10000)	1	4	0	0.201	0.000644
11(99)	5	90	12	0.290	0.00102	36(1000)	30	3436	39	2.283	0.0332
11(399)	*	*	*	*	*	36(5000)	33	4051	47	10.285	0.0316
12(1000)	*	*	*	*	*	37(1000)	13	360	4	0.841	0.0184
12(10000)	46	88312	584	257.891	0.000788	37(5000)	12	292	4	3.044	0.0601
13(100)	4	60	5	0.040	0.00535	38(1000)	13	366	7	1.002	0.0556
13(1000)	5	81	5	0.240	0.00183	38(5000)	13	324	2	3.395	0.0551
14(2500)	11	365	0	1.553	0.0198	39(1000)	4	40	0	0.130	0.000317
14(10000)	12	460	0	6.419	0.0871	39(5000)	4	40	0	0.611	0.000709
15(5000)	4	34	0	0.370	0.000532	40(1000)	11	289	1	0.451	0.000135
15(15000)	4	34	0	1.052	0.000919	40(5000)	11	289	1	2.033	0.000301
16(500)	5	60	0	0.120	6.17e-5	41(1000)	5	58	0	0.170	3.66e-6
16(2000)	5	58	0	0.351	0.000209	41(5000)	5	58	0	0.911	8.19e-6
17(100)	5	57	1	0.050	7.7e-5	42(1000)	2	11	0	0.060	3.11e-8
17(1000)	5	57	1	0.270	0.000126	42(5000)	1	4	0	0.130	0.000409
18(100)	5	62	2	0.110	8.85e-5	43(500)	3	25	0	0.101	0.000108
18(1000)	5	62	2	0.661	6.33e-5	43(1000)	1	4	0	0.040	0.000288
19(100)	5	62	2	0.101	5.95e-5	44(100)	7	166	1	0.090	0.000351
19(1000)	5	62	2	0.630	7.25e-5	44(500)	7	166	1	0.290	0.000351
20(500)	5	76	2	0.151	2.63e-5	45(500)	14	329	0	0.271	0.212
20(1000)	5	76	2	0.250	4.76e-5	45(1000)	14	337	0	0.500	0.0741
21(1000)	4	34	0	0.070	2.26e-5	46(1000)	3	21	0	0.832	1.02e-5
21(50000)	4	34	0	3.075	0.000159	46(5000)	2	11	0	2.874	0.000236
22(100)	7	201	0	0.090	0.000114	47(999)	6	83	1	0.221	0.000131
22(1000)	8	526	0	1.352	0.0017	47(4999)	6	87	1	1.231	7.35e-5
23(399)	*	*	*	*	*	48(500)	8	532	8	0.981	0.0108
23(9999)	*	*	*	*	*	48(1000)	5	82	0	0.521	0.00722
24(1000)	1	4	0	0.020	0.00138	49(100)	5	164	0	0.160	0.00248
24(15000)	2	11	0	0.431	1.79e-7	49(500)	4	68	0	0.240	0.0107
25(500)	1	4	0	0.020	0.000572	50(33)	9	787	14	0.481	0.0022
25(5000)	1	4	0	0.100	0.029	50(133)	8	529	9	0.501	0.00644

Tabla 2.4: Funciones de prueba: Newton-Bi-CGSTAB.

Función(n)	IT	F	BL	T	NF	Función(n)	IT	F	BL	T	NF
1(1000)	3	23	0	0.070	0.000285	26(250)	11	313	9	0.080	1.52e-5
1(10000)	1	4	0	0.131	0.00095	26(1000)	7	146	4	0.190	0.000218
2(500)	56	113327	3	29.101	0.000225	27(100)	*	*	*	*	*
2(2000)	66	164193	2	99.494	0.000451	27(1000)	4	272	0	0.831	0.00799
3(100)	10	381	0	0.130	5.05e-5	28(1000)	*	*	*	*	*
3(500)	11	395	0	0.480	8.68e-5	28(10000)	*	*	*	*	*
4(99)	9	365	22	0.211	9.74e-5	29(50)	13	286	0	2.103	5.4e-5
4(999)	9	371	22	1.492	7.27e-9	29(500)	13	286	0	79.274	0.000274
5(99)	2	15	0	0.020	1.89e-8	30(100)	1	4	0	0.010	9.78e-6
5(999)	2	15	0	0.090	6.01e-8	30(1000)	1	4	0	0.021	3.09e-5
6(100)	6	75	0	0.030	2.74e-8	31(100)	10	195	0	0.050	0.00958
6(10000)	6	75	0	1.032	2.91e-7	31(1000)	10	179	0	0.180	0.00973
7(100)	5	70	1	0.030	7.91e-8	32(99)	3	25	0	0.020	4.3e-5
7(10000)	5	70	1	1.241	1.78e-5	32(9999)	3	25	0	0.731	0.000433
8(1000)	5	50	0	0.080	8.13e-5	33(500)	5	69	2	0.120	7.93e-5
8(10000)	5	50	0	0.731	0.000257	33(1000)	6	94	2	0.210	1.23e-5
9(100)	4	38	0	0.170	6.26e-5	34(1000)	4	34	0	0.100	9.48e-8
9(1000)	4	38	0	11.717	0.000198	34(5000)	4	34	0	0.390	2.12e-7
10(100)	*	*	*	*	*	35(5000)	2	11	0	0.140	6.13e-5
10(5000)	*	*	*	*	*	35(10000)	1	4	0	0.151	0.000644
11(99)	5	220	3	0.841	0.00061	36(1000)	14	800	6	1.752	0.00517
11(399)	5	218	2	2.814	0.00123	36(5000)	23	3301	25	15.322	0.0359
12(1000)	*	*	*	*	*	37(1000)	12	456	4	1.352	0.0274
12(10000)	*	*	*	*	*	37(5000)	12	428	4	5.168	0.0529
13(100)	4	60	5	0.030	0.00622	38(1000)	14	1219	16	3.785	0.0492
13(1000)	5	81	5	0.210	0.00263	38(5000)	13	444	2	5.378	0.0289
14(2500)	11	309	0	1.402	0.0174	39(1000)	4	40	0	0.110	0.00102
14(10000)	11	315	0	5.047	0.142	39(5000)	4	40	0	0.541	0.00227
15(5000)	4	34	0	0.270	0.000532	40(1000)	10	315	1	0.461	0.000139
15(15000)	4	34	0	0.781	0.000919	40(5000)	10	315	1	2.043	0.000312
16(500)	5	68	0	0.160	3.31e-5	41(1000)	4	48	0	0.170	0.000921
16(2000)	5	76	0	0.591	7.57e-6	41(5000)	4	48	0	0.821	0.00206
17(100)	5	65	1	0.050	0.000109	42(1000)	3	21	0	0.070	1.12e-5
17(1000)	6	83	1	0.331	0.000142	42(5000)	3	21	0	0.271	2.85e-6
18(100)	5	65	1	0.120	1.27e-5	43(500)	3	103	0	0.731	4.08e-5
18(1000)	5	68	2	0.671	4.96e-5	43(1000)	2	13	0	0.100	0.000148
19(100)	5	65	1	0.110	1.65e-5	44(100)	9	369	1	0.170	0.000235
19(1000)	5	70	2	0.721	3.35e-5	44(500)	9	367	1	0.551	0.000282
20(500)	5	82	2	0.180	7.3e-5	45(500)	14	343	0	0.270	0.182
20(1000)	5	98	2	0.361	5.76e-5	45(1000)	14	377	0	0.781	0.0646
21(1000)	4	34	0	0.060	1.35e-6	46(1000)	2	11	0	0.431	0.000296
21(50000)	4	34	0	2.273	9.52e-6	46(5000)	2	11	0	2.063	0.00014
22(100)	7	329	0	0.211	2.54e-5	47(999)	5	62	0	0.210	0.000102
22(1000)	8	780	0	2.984	0.000571	47(4999)	5	64	0	1.252	0.000178
23(399)	*	*	*	*	*	48(500)	5	316	0	1.442	0.0112
23(9999)	*	*	*	*	*	48(1000)	4	138	0	1.773	0.0119
24(1000)	1	4	0	0.010	0.00138	49(100)	5	490	0	0.480	0.00495
24(15000)	2	11	0	0.331	4.21e-8	49(500)	3	95	0	0.591	0.00791
25(500)	1	4	0	0.020	0.000572	50(33)	6	451	5	0.481	0.00327
25(5000)	1	4	0	0.070	0.029	50(133)	6	531	2	1.021	0.00324

Tabla 2.5: Funciones de prueba: Newton-TFQMR.

Función(n)	IT	F	BL	T	NF	Función(n)	IT	F	BL	T	NF
1(1000)	5	20	0	0.050	0.000283	26(250)	*	*	*	*	*
1(10000)	2	5	0	0.130	0.000562	26(1000)	*	*	*	*	*
2(500)	*	*	*	*	*	27(100)	*	*	*	*	*
2(2000)	*	*	*	*	*	27(1000)	*	*	*	*	*
3(100)	*	*	*	*	*	28(1000)	1	2	0	0.020	0
3(500)	*	*	*	*	*	28(10000)	1	2	0	0.090	0
4(99)	*	*	*	*	*	29(50)	7	42	1	1.002	5.5e-5
4(999)	*	*	*	*	*	29(500)	1	3	1	11.436	7.06e-30
5(99)	*	*	*	*	*	30(100)	1	2	0	0.000	9.78e-6
5(999)	*	*	*	*	*	30(1000)	1	2	0	0.020	3.09e-5
6(100)	*	*	*	*	*	31(100)	*	*	*	*	*
6(10000)	*	*	*	*	*	31(1000)	*	*	*	*	*
7(100)	*	*	*	*	*	32(99)	*	*	*	*	*
7(10000)	*	*	*	*	*	32(9999)	*	*	*	*	*
8(1000)	1	2	0	0.010	0	33(500)	*	*	*	*	*
8(10000)	1	2	0	0.100	0	33(1000)	*	*	*	*	*
9(100)	6	27	0	0.120	9.47e-8	34(1000)	5	20	0	0.070	9.76e-6
9(1000)	6	27	0	7.351	3.01e-7	34(5000)	5	20	0	0.280	2.18e-5
10(100)	*	*	*	*	*	35(5000)	4	57	12	0.331	4.23e-7
10(5000)	*	*	*	*	*	35(10000)	3	51	14	0.591	7.86e-5
11(99)	5	93	16	0.170	0.000882	36(1000)	*	*	*	*	*
11(399)	5	93	16	0.601	0.00177	36(5000)	*	*	*	*	*
12(1000)	5	30	3	0.080	0.000147	37(1000)	*	*	*	*	*
12(10000)	4	21	3	0.471	0.000506	37(5000)	*	*	*	*	*
13(100)	*	*	*	*	*	38(1000)	*	*	*	*	*
13(1000)	*	*	*	*	*	38(5000)	122	25502	277	42.732	0.139
14(2500)	*	*	*	*	*	39(1000)	*	*	*	*	*
14(10000)	*	*	*	*	*	39(5000)	*	*	*	*	*
15(5000)	6	27	0	0.250	5.6e-7	40(1000)	*	*	*	*	*
15(15000)	6	27	0	0.741	9.64e-7	40(5000)	43	3755	137	11.266	0.000742
16(500)	41	2463	78	1.142	0.000163	41(1000)	9	282	32	0.280	0.000674
16(2000)	23	552	22	1.542	0.000385	41(5000)	10	336	34	1.713	0.000928
17(100)	32	1267	39	0.270	0.000178	42(1000)	3	9	0	0.040	1.42e-5
17(1000)	35	1629	46	2.374	0.000332	42(5000)	2	6	1	0.130	8.89e-5
18(100)	24	624	24	0.310	0.000107	43(500)	14	290	25	0.281	0.000222
18(1000)	23	653	27	2.043	0.000331	43(1000)	3	15	3	0.070	0.00027
19(100)	24	624	24	0.301	0.000106	44(100)	*	*	*	*	*
19(1000)	26	937	37	2.573	0.000335	44(500)	*	*	*	*	*
20(500)	*	*	*	*	*	45(500)	*	*	*	*	*
20(1000)	*	*	*	*	*	45(1000)	*	*	*	*	*
21(1000)	6	27	0	0.050	1.77e-5	46(1000)	*	*	*	*	*
21(50000)	6	27	0	2.254	0.000124	46(5000)	*	*	*	*	*
22(100)	3	10	1	0.010	0	47(999)	*	*	*	*	*
22(1000)	*	*	*	*	*1	47(4999)	29	1791	61	6.620	0.000729
23(399)	*	*	*	*	*	48(500)	*	*	*	*	*
23(9999)	*	*	*	*	*	48(1000)	*	*	*	*	*
24(1000)	*	*	*	*	*	49(100)	*	*	*	*	*
24(15000)	*	*	*	*	*	49(500)	*	*	*	*	*
25(500)	1	11	9	0.030	8.61e-10	50(33)	*	*	*	*	*
25(5000)	1	14	12	0.200	9.68e-7	50(133)	*	*	*	*	*

Tabla 2.6: Funciones de prueba: Broyden.

Función(n)	IT	F	BL	T	NF	Función(n)	IT	F	BL	T	NF
1(1000)	5	10	0	0.030	0.000152	26(250)	21	45	3	0.050	4.75e-5
1(10000)	2	4	0	0.111	0.000562	26(1000)	62	136	11	0.451	0.000298
2(500)	199	469	70	0.881	0.000221	27(100)	26	57	3	0.050	0.000382
2(2000)	342	804	118	5.358	0.000445	27(1000)	6	16	2	0.050	0.00437
3(100)	16	34	2	0.040	7.8e-5	28(1000)	1	2	0	0.010	0
3(500)	20	45	4	0.160	0.000128	28(10000)	1	2	0	0.050	0
4(99)	257	722	162	2.023	5.56e-5	29(50)	11	23	1	1.873	6.61e-5
4(999)	225	563	100	9.684	0.000311	29(500)	1	3	1	12.408	2.07e-15
5(99)	202	600	101	1.091	9.8e-5	30(100)	1	2	0	0.000	9.78e-6
5(999)	202	600	101	2.714	0.000311	30(1000)	1	2	0	0.010	3.09e-5
6(100)	35	73	2	0.080	0.00105	31(100)	1	4	1	0.000	8.91e-6
6(10000)	35	73	2	1.372	0.0105	31(1000)	1	4	1	0.020	8.99e-9
7(100)	25	53	2	0.070	0.000103	32(99)	5	11	1	0.020	7.81e-5
7(10000)	25	53	2	1.523	0.00103	32(9999)	5	11	1	0.501	0.000785
8(1000)	1	2	0	0.010	0	33(500)	16	32	0	0.110	0.000157
8(10000)	1	2	0	0.070	0	33(1000)	17	34	0	0.231	8.34e-6
9(100)	7	14	0	0.230	5.13e-5	34(1000)	5	10	0	0.070	9.76e-6
9(1000)	7	14	0	15.813	0.000162	34(5000)	5	10	0	0.290	2.18e-5
10(100)	19	73	7	0.081	0.000106	35(5000)	4	24	1	0.291	3.12e-6
10(5000)	19	75	6	1.311	6.58e-5	35(10000)	3	23	1	0.520	8.11e-5
11(99)	12	36	4	0.291	0.000903	36(1000)	94	218	21	1.503	0.0404
11(399)	12	36	4	0.991	0.00181	36(5000)	38	81	3	2.143	0.0587
12(1000)	6	14	2	0.081	0.000199	37(1000)	57	127	9	1.241	0.0263
12(10000)	5	12	2	0.460	0.000548	37(5000)	28	61	3	2.304	0.0884
13(100)	19	40	1	0.060	0.00709	38(1000)	30	64	2	0.841	0.0316
13(1000)	20	42	1	0.281	0.00487	38(5000)	30	64	2	3.165	0.106
14(2500)	16	35	1	0.450	0.0243	39(1000)	37	80	5	0.300	0.000444
14(10000)	16	36	1	1.543	0.189	39(5000)	37	80	5	1.622	0.000992
15(5000)	6	12	0	0.150	5.6e-7	40(1000)	95	219	25	0.852	0.000243
15(15000)	6	12	0	0.441	9.64e-7	40(5000)	80	185	23	3.465	0.000522
16(500)	16	33	1	0.090	0.0002	41(1000)	83	189	20	0.791	0.000208
16(2000)	19	38	0	0.350	0.000223	41(5000)	83	189	20	4.997	0.000504
17(100)	11	23	1	0.040	0.000162	42(1000)	3	6	0	0.040	6.99e-6
17(1000)	10	21	1	0.241	0.0004	42(5000)	3	6	0	0.150	5.71e-8
18(100)	14	30	2	0.140	2.63e-5	43(500)	7	15	1	0.060	0.000221
18(1000)	11	24	2	0.671	0.000314	43(1000)	2	5	1	0.051	0.0003
19(100)	14	30	2	0.150	2.02e-5	44(100)	424	1229	269	1.642	0.0012
19(1000)	13	28	2	0.781	0.000315	44(500)	405	1167	254	4.146	0.00129
20(500)	22	46	2	0.120	0.000124	45(500)	23	51	2	0.200	0.162
20(1000)	30	64	4	0.371	8.49e-5	45(1000)	26	59	5	0.391	0.0662
21(1000)	6	12	0	0.030	5.81e-6	46(1000)	4	8	0	0.661	2.18e-5
21(50000)	6	12	0	0.981	4.06e-5	46(5000)	3	6	0	2.584	0.000184
22(100)	64	136	6	0.130	1.75e-5	47(999)	17	36	2	0.180	0.000151
22(1000)	113	272	26	0.871	0.00323	47(4999)	20	41	1	1.132	0.000269
23(399)	141	359	65	0.791	0.000172	48(500)	23	47	1	0.300	0.0102
23(9999)	157	389	69	15.823	0.00102	48(1000)	23	47	1	0.561	0.0147
24(1000)	1	2	0	0.010	0	49(100)	68	165	23	0.221	0.00413
24(15000)	1	2	0	0.040	0	49(500)	37	84	9	0.250	0.0107
25(500)	1	10	1	0.020	8.61e-10	50(33)	43	95	8	0.260	0.00247
25(5000)	1	13	1	0.130	1.51e-6	50(133)	32	70	5	0.231	0.00889

Tabla 2.7: Funciones de prueba: SANEG.

Capítulo 3

Trazado de Rayos Sísmicos en Medios Heterogéneos en 3D

El objetivo principal del trazado de rayos de un modelo emisor-receptor es obtener todas las posibles llegadas que contribuyen con una energía significativa dentro de un intervalo de tiempo de interés, para generar sismogramas sintéticos exactos y completos.

Tradicionalmente el problema del trazado de rayos se describe mediante un sistema de ecuaciones diferenciales de primer orden o un problema de valores en la frontera en dos puntos ([37], [36], [62], [63], [48], [47]). Este tipo de estudio considera el algoritmo de Newton para resolver internamente sistemas no-lineales, lo cual hace que estos procedimientos de trazado de rayos sean computacionalmente costosos.

Otros investigadores resuelven el trazado de rayos como un problema de optimización usando el principio de Fermat, que encuentra el rayo que posea el mínimo tiempo de viaje entre el emisor y el receptor ([59], [49], [42], [17]). Bajo este enfoque, el trazado de rayos se hace menos costoso computacionalmente y los resultados son satisfactorios.

Nosotros modelaremos el trazado de rayos sísmicos como un sistema de ecuaciones no lineales que busca describir el rayo de Fermat entre el emisor y el receptor. Cabe resaltar que con este procedimiento también se realiza una discretización finita del rayo, pero la discretización es más fina que la empleada por el grupo de investigadores antes citado.

3.1 Modelo de lentitud ondular

Cuando una onda sonora o sísmica es emitida dentro de un medio, transcurre un tiempo para ver la influencia del progreso de la onda desde un punto cercano al emisor a un punto más distante. El tiempo que le toma a la onda el viajar de un punto de interés a

otro próximo se denomina *tiempo de viaje*. Para un medio que no sufre cambios físicos ni químicos durante el paso del sonido, la onda tiene una velocidad definida con la que siempre viaja entre dos puntos del medio. A esta velocidad la llamamos *velocidad de onda promedio* o *velocidad de onda*. También definimos la *velocidad de onda local* asociada con cada punto del medio considerando el promedio de velocidad de onda para dos puntos cercanos (Berryman [5]).

Formalicemos matemáticamente los conceptos expuestos anteriormente. Sea $\mathbf{x} = (x, y, z)^t$ un punto perteneciente a \mathbb{R}^3 . Denotaremos con $v(\mathbf{x})$ a la velocidad de onda local en el punto \mathbf{x} . A la función $v : \mathbb{R}^3 \rightarrow \mathbb{R}$ se le denomina velocidad del medio.

Definición 3.1 (Lentitud) *La lentitud local ó, simplemente, lentitud es el inverso de la velocidad de onda local. Denotaremos con $s(\mathbf{x})$ a la lentitud en el punto \mathbf{x} , es decir,*

$$s(\mathbf{x}) = \frac{1}{v(\mathbf{x})}. \quad (3.1)$$

La función $s : \mathbb{R}^3 \rightarrow \mathbb{R}$ se denomina lentitud del medio

Definición 3.2 (Rayo) *Sean $\mathbf{x}^e \in \mathbb{R}^3$ un emisor y $\mathbf{x}^r \in \mathbb{R}^3$ un receptor. Diremos que una curva C en \mathbb{R}^3 es un rayo si sus extremos inicial y final son respectivamente \mathbf{x}^e y \mathbf{x}^r .*

Cuando se le asigne a un medio una velocidad y, por ende, una lentitud, diremos que se tiene un modelo de lentitud.

3.1.1 Funcional de tiempo de viaje y principio de Fermat

El tiempo de viaje de una onda sísmica es la integral de la lentitud a lo largo de un rayo que conecta a un emisor con un receptor. Sea C un rayo cualquiera que conecta un emisor y un receptor en un modelo de lentitud. Entonces, el funcional lineal τ^C definido como

$$\tau^C(s) = \int_C s(x, y, z) dl \quad (3.2)$$

se denomina funcional de tiempo de viaje a lo largo de C , donde dl denota la distancia infinitesimal a lo largo de C .

El *principio de Fermat* afirma que el rayo correcto entre dos puntos es aquel de menor tiempo de viaje, es decir, es el que minimiza a $\tau^C(s)$ con respecto a C . Sea τ^* el funcional

de tiempo de viaje a lo largo del rayo de Fermat (de menor tiempo). Entonces el principio de Fermat afirma que

$$\tau^*(s) = \min_{C \in \mathcal{C}} \tau^C(s), \quad (3.3)$$

donde \mathcal{C} denota el conjunto de todos los rayos continuos que conectan al emisor con el receptor. El rayo particular que produce el mínimo en (3.3) se denota C^* . Si existe más de un rayo que produce el mínimo tiempo de viaje, entonces C^* denota algún miembro en el conjunto de rayos minimizadores. Sustituyendo (3.2) en (3.3), el principio de Fermat adquiere la forma:

$$\tau^*(s) = \int_{C^*} s(x, y, z) dl = \min_{C \in \mathcal{C}} \int_C s(x, y, z) dl. \quad (3.4)$$

El funcional lineal $\tau^*(s)$ es *estacionario con respecto a la más pequeña variación en el rayo C^** .

3.2 Trazado de rayos entre dos puntos

Sean $\mathbf{x}^e \in \mathbb{R}^3$ un emisor y $\mathbf{x}^r \in \mathbb{R}^3$ un receptor. Sea C_e^r el rayo de Fermat que conecta a \mathbf{x}^e con \mathbf{x}^r . Denotemos con $\mathcal{C}(\mathbf{x}^e, \mathbf{x}^r)$ al conjunto de rayos que conectan a \mathbf{x}^e con \mathbf{x}^r . De esta forma,

$$\tau_{er}(s) = \int_{C_e^r} s(x, y, z) dl = \min_{C \in \mathcal{C}(\mathbf{x}^e, \mathbf{x}^r)} \int_C s(x, y, z) dl, \quad (3.5)$$

donde $dl = [dx^2 + dy^2 + dz^2]^{1/2}$.

Supongamos que deseamos aproximar N puntos del rayo C_e^r , distintos al emisor y al receptor. En otras palabras, tenemos $N + 2$ puntos del rayo C_e^r , $\mathbf{x}^0, \mathbf{x}^1, \mathbf{x}^2, \dots, \mathbf{x}^N, \mathbf{x}^{N+1}$, donde $\mathbf{x}^0 = \mathbf{x}^e = (x_0, y_0, z_0)^t$, $\mathbf{x}^{N+1} = \mathbf{x}^r = (x_{N+1}, y_{N+1}, z_{N+1})^t$ y $\mathbf{x}^i = (x_i, y_i, z_i)^t$ para $i = 0, 1, \dots, N + 1$.

Para un medio heterogéneo, supongamos que la longitud del rayo entre dos puntos consecutivos del mismo, $(x_{i-1}, y_{i-1}, z_{i-1})$ y (x_i, y_i, z_i) , se aproxima por la longitud del segmento de recta que une a tales puntos,

$$l_i = \sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2 + (z_i - z_{i-1})^2}. \quad (3.6)$$

De esta forma, el tiempo de viaje de la ecuación integral (3.5) se aproxima por la suma

$$\mathbf{t} = \sum_{i=1}^{N+1} s_i l_i, \quad (3.7)$$

donde

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

Ahora, acorde al principio de Fermat, la verdadera dirección del rayo tiene tiempo estacionario de \mathbf{x}^{i-1} a \mathbf{x}^i , así, la trayectoria actual del rayo puede determinarse igualando a cero las derivadas parciales de \mathbf{t} de la ecuación (3.7) con respecto a x_i , y_i y z_i . Esto origina un sistema de $3N$ ecuaciones no lineales con $3N$ incógnitas,

$$\begin{cases} \frac{\partial t}{\partial x_i} = \frac{\partial s_i}{\partial x_i} l_i + \frac{\partial l_i}{\partial x_i} s_i + \frac{\partial s_{i+1}}{\partial x_i} l_{i+1} + \frac{\partial l_{i+1}}{\partial x_i} s_{i+1} = 0, \\ \frac{\partial t}{\partial y_i} = \frac{\partial s_i}{\partial y_i} l_i + \frac{\partial l_i}{\partial y_i} s_i + \frac{\partial s_{i+1}}{\partial y_i} l_{i+1} + \frac{\partial l_{i+1}}{\partial y_i} s_{i+1} = 0, \\ \frac{\partial t}{\partial z_i} = \frac{\partial s_i}{\partial z_i} l_i + \frac{\partial l_i}{\partial z_i} s_i + \frac{\partial s_{i+1}}{\partial z_i} l_{i+1} + \frac{\partial l_{i+1}}{\partial z_i} s_{i+1} = 0, \end{cases} \quad (3.8)$$

para $i = 1, 2, \dots, N$. Por otra parte, se tiene que

$$\begin{aligned} \frac{\partial l_i}{\partial x_i} &= \frac{x_i - x_{i-1}}{l_i}, & \frac{\partial l_{i+1}}{\partial x_i} &= \frac{-(x_{i+1} - x_i)}{l_{i+1}}, \\ \frac{\partial l_i}{\partial y_i} &= \frac{y_i - y_{i-1}}{l_i}, & \frac{\partial l_{i+1}}{\partial y_i} &= \frac{-(y_{i+1} - y_i)}{l_{i+1}}, \\ \frac{\partial l_i}{\partial z_i} &= \frac{z_i - z_{i-1}}{l_i}, & \frac{\partial l_{i+1}}{\partial z_i} &= \frac{-(z_{i+1} - z_i)}{l_{i+1}}. \end{aligned}$$

Empleando las expresiones anteriores, el sistema (3.8) se puede reescribir como:

$$\begin{cases} \frac{\partial s_i}{\partial x_i} l_i + \frac{(x_i - x_{i-1})}{l_i} s_i + \frac{\partial s_{i+1}}{\partial x_i} l_{i+1} - \frac{(x_{i+1} - x_i)}{l_{i+1}} s_{i+1} = 0, \\ \frac{\partial s_i}{\partial y_i} l_i + \frac{(y_i - y_{i-1})}{l_i} s_i + \frac{\partial s_{i+1}}{\partial y_i} l_{i+1} - \frac{(y_{i+1} - y_i)}{l_{i+1}} s_{i+1} = 0, \\ \frac{\partial s_i}{\partial z_i} l_i + \frac{(z_i - z_{i-1})}{l_i} s_i + \frac{\partial s_{i+1}}{\partial z_i} l_{i+1} - \frac{(z_{i+1} - z_i)}{l_{i+1}} s_{i+1} = 0, \end{cases} \quad (3.9)$$

para $i = 1, 2, \dots, N$. De esta forma, el sistema de ecuaciones no lineales (3.9) describe la discretización en N puntos del rayo de Fermat que va del punto \mathbf{x}^e al punto \mathbf{x}^r . Equivalentemente, el sistema (3.9) se puede expresar como

$$F(X) = (f_1(X), \dots, f_{3N}(X))^t = 0, \quad (3.10)$$

donde $X = (x_1, \dots, x_N, y_1, \dots, y_N, z_1, \dots, z_N)^t \in \mathbb{R}^{3N}$, y

$$\begin{cases} f_i(X) = \frac{\partial s_i}{\partial x_i} l_i + \frac{(x_i - x_{i-1})}{l_i} s_i + \frac{\partial s_{i+1}}{\partial x_i} l_{i+1} - \frac{(x_{i+1} - x_i)}{l_{i+1}} s_{i+1}, \\ f_{i+N}(X) = \frac{\partial s_i}{\partial y_i} l_i + \frac{(y_i - y_{i-1})}{l_i} s_i + \frac{\partial s_{i+1}}{\partial y_i} l_{i+1} - \frac{(y_{i+1} - y_i)}{l_{i+1}} s_{i+1}, \\ f_{i+2N}(X) = \frac{\partial s_i}{\partial z_i} l_i + \frac{(z_i - z_{i-1})}{l_i} s_i + \frac{\partial s_{i+1}}{\partial z_i} l_{i+1} - \frac{(z_{i+1} - z_i)}{l_{i+1}} s_{i+1}, \end{cases} \quad (3.11)$$

para $i = 1, 2, \dots, N$.

Es claro que cuando el medio posee velocidad de onda constante, el rayo de Fermat que une el emisor con el receptor es un segmento de recta. Veamos que efectivamente el segmento de recta que une a \mathbf{x}^e con \mathbf{x}^r es solución del sistema (3.10). En el siguiente teorema establecemos que los puntos pertenecientes al segmento de recta que une a \mathbf{x}^e con \mathbf{x}^r son las únicas soluciones de (3.10), cuando la velocidad del medio es constante.

Teorema 3.1 *Supongamos que la velocidad del medio es constante, $v(x, y, z) = d$. El punto $X = (x_1, \dots, x_N, y_1, \dots, y_N, z_1, \dots, z_N)^t \in \mathbb{R}^{3N}$ es una solución del sistema $F(X) = 0$ dado por (3.10) si, y sólo si los puntos $\mathbf{x}^1, \mathbf{x}^2, \dots, \mathbf{x}^N$, pertenecen al segmento de recta que une al emisor \mathbf{x}^e con el receptor \mathbf{x}^r .*

Demostración. Como $v(x, y, z) = d$, las funciones $f_i(X)$ del sistema $F(X) = 0$ dado por (3.10), las podemos escribir como:

$$\begin{cases} f_i(X) = \frac{1}{d} \left(\frac{x_i - x_{i-1}}{l_i} - \frac{x_{i+1} - x_i}{l_{i+1}} \right), \\ f_{i+N}(X) = \frac{1}{d} \left(\frac{y_i - y_{i-1}}{l_i} - \frac{y_{i+1} - y_i}{l_{i+1}} \right), \\ f_{i+2N}(X) = \frac{1}{d} \left(\frac{z_i - z_{i-1}}{l_i} - \frac{z_{i+1} - z_i}{l_{i+1}} \right), \end{cases} \quad (3.12)$$

para $i = 1, 2, \dots, N$.

(\Rightarrow) Supongamos que el punto $X = (x_1, \dots, x_N, y_1, \dots, y_N, z_1, \dots, z_N)^t \in \mathbb{R}^{3N}$ es una solución del sistema $F(X) = 0$ dado por (3.10). Luego, del sistema (3.12) se obtiene que

$$\frac{x_i - x_{i-1}}{x_{i+1} - x_i} = \frac{l_i}{l_{i+1}}, \quad (3.13)$$

$$\frac{y_i - y_{i-1}}{y_{i+1} - y_i} = \frac{l_i}{l_{i+1}}, \quad (3.14)$$

$$\frac{z_i - z_{i-1}}{z_{i+1} - z_i} = \frac{l_i}{l_{i+1}}. \quad (3.15)$$

Las ecuaciones (3.13), (3.14) y (3.15) indican que el punto \mathbf{x}^i pertenece a la recta que pasa por los puntos \mathbf{x}^{i-1} y \mathbf{x}^{i+1} , para $i = 1, 2, \dots, N$. Por lo tanto, los puntos $\mathbf{x}^1, \mathbf{x}^2, \dots, \mathbf{x}^N$, pertenecen a la recta que pasa por los puntos $\mathbf{x}^0 = \mathbf{x}^e$ y $\mathbf{x}^{N+1} = \mathbf{x}^r$. Ahora bien, como $l_i \geq 0$ para todo i , de las ecuaciones (3.13), (3.14) y (3.15) se deduce que $x_i \geq x_{i-1}$ ó $x_i \leq x_{i-1}$, $y_i \geq y_{i-1}$ ó $y_i \leq y_{i-1}$, $z_i \geq z_{i-1}$ ó $z_i \leq z_{i-1}$. En consecuencia, los puntos $\mathbf{x}^1, \mathbf{x}^2, \dots, \mathbf{x}^N$, pertenecen al segmento de recta que une a \mathbf{x}^e con \mathbf{x}^r .

(\Leftarrow) Supongamos que los puntos $\mathbf{x}^1, \mathbf{x}^2, \dots, \mathbf{x}^N$, pertenecen al segmento de recta que une a \mathbf{x}^e con \mathbf{x}^r . Entonces, cada punto \mathbf{x}^i se puede expresar, sin pérdida de generalidad, como

$$\mathbf{x}^i = \lambda_i \mathbf{x}^r + (1 - \lambda_i) \mathbf{x}^e, \quad \lambda_i \in [0, 1]$$

con $\lambda_i \geq \lambda_{i-1}$, para $i = 1, 2, \dots, N$. Luego,

$$\begin{aligned} x_i - x_{i-1} &= (\lambda_i - \lambda_{i-1})(x_{N+1} - x_0), & x_{i+1} - x_i &= (\lambda_{i+1} - \lambda_i)(x_{N+1} - x_0), \\ y_i - y_{i-1} &= (\lambda_i - \lambda_{i-1})(y_{N+1} - y_0), & y_{i+1} - y_i &= (\lambda_{i+1} - \lambda_i)(y_{N+1} - y_0), \\ z_i - z_{i-1} &= (\lambda_i - \lambda_{i-1})(z_{N+1} - z_0), & z_{i+1} - z_i &= (\lambda_{i+1} - \lambda_i)(z_{N+1} - z_0). \end{aligned}$$

Así,

$$\begin{aligned} \frac{x_i - x_{i-1}}{l_i} - \frac{x_{i+1} - x_i}{l_{i+1}} &= \frac{(\lambda_i - \lambda_{i-1})(x_{N+1} - x_0)}{l_i} - \frac{(\lambda_{i+1} - \lambda_i)(x_{N+1} - x_0)}{l_{i+1}} \\ &= \frac{(\lambda_i - \lambda_{i-1})(x_{N+1} - x_0)}{(\lambda_i - \lambda_{i-1}) \|\mathbf{x}^r - \mathbf{x}^e\|} - \frac{(\lambda_{i+1} - \lambda_i)(x_{N+1} - x_0)}{(\lambda_{i+1} - \lambda_i) \|\mathbf{x}^r - \mathbf{x}^e\|} \\ &= 0. \end{aligned}$$

De la misma forma obtenemos:

$$\begin{aligned} \frac{y_i - y_{i-1}}{l_i} - \frac{y_{i+1} - y_i}{l_{i+1}} &= 0, \\ \frac{z_i - z_{i-1}}{l_i} - \frac{z_{i+1} - z_i}{l_{i+1}} &= 0. \end{aligned}$$

Por lo tanto, $f_i(X) = f_{i+N}(X) = f_{i+2N}(X) = 0$ para $i = 1, \dots, N$. En otras palabras, el punto $X = (x_1, \dots, x_N, y_1, \dots, y_N, z_1, \dots, z_N)^t \in \mathbb{R}^{3N}$ es una solución del sistema $F(X) = 0$ dado por (3.10). \square

3.2.1 Ejemplos numéricos del trazado de rayo entre dos puntos

A continuación mostramos dos ejemplos numéricos del trazado de rayo entre dos puntos. Utilizamos al algoritmo SANEG para resolver el sistema $F(X) = 0$ dado por (3.10). Los valores de los parámetros de SANEG que usamos son: $\gamma = 10^{-4}$, $\varepsilon = 10^{-10}$, $\sigma_1 = 0.1$, $\sigma_2 = 0.5$, $\alpha_0 = 1$, $M = 10$ y δ como en (2.33). Utilizamos (2.3), con $h = 10^{-7}$, para aproximar al escalar $F_k^t J_k F_k$. Utilizamos el modelo parabólico de dos puntos dado por (2.34) y (2.35). El criterio de parada para SANEG es:

$$\|F_k\| \leq 7.5 \times 10^{-3} \|F_0\|.$$

En ambos ejemplos aproximamos $N = 10$ puntos del rayo.

Ejemplo 1

Consideramos un medio con velocidad constante, $v(x, y, z) = 100$. Tenemos un emisor $\mathbf{x}^e = (5, 5, 0)^t$ y un receptor $\mathbf{x}^r = (10, 10, -10)^t$. En la Figura 3.1 observamos el rayo inicial y el rayo final. El algoritmo SANEG tardó 0.651 segundos y necesito de 71 iteraciones, 180 evaluaciones de la función F y 31 búsquedas lineales, para satisfacer el criterio de parada. Además, $\|F_0\| = 0.0085496$ y $\|F_k\| = 5.5247 \times 10^{-5}$. El tiempo de viaje del rayo inicial es 0.30941 y el tiempo de viaje del rayo final es 0.25576.

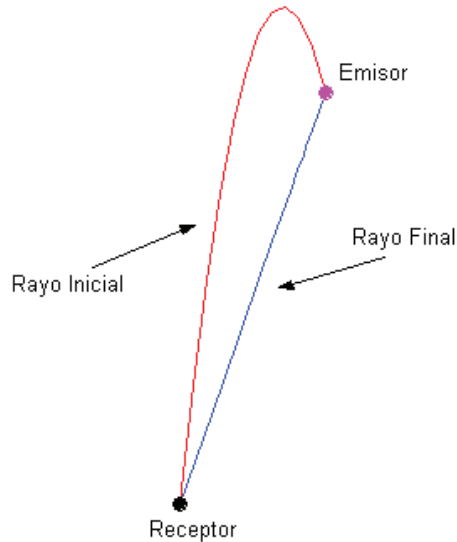


Figura 3.1: Trazado de rayo entre dos puntos en un medio con velocidad $v(x, y, z) = 100$.

Ejemplo 2

Consideramos un medio con velocidad $v(x, y, z) = 10x + 3y - 4z + 10$ (el signo “-” aparece junto a la variable z , ya que empleamos una escala negativa para la profundidad). Tenemos un emisor $\mathbf{x}^e = (5, 5, 0)^t$ y un receptor $\mathbf{x}^r = (10, 10, -10)^t$. En la Figura 3.2 observamos el rayo inicial y el rayo final. El algoritmo SANEG tardó 0.851 segundos y necesito de 80 iteraciones, 195 evaluaciones de la función F y 23 búsquedas lineales, para satisfacer el criterio de parada. Además, $\|F_0\| = 0.010745$ y $\|F_k\| = 7.6919 \times 10^{-5}$. El tiempo de viaje del rayo inicial es 0.24444 y el tiempo de viaje del rayo final es 0.19644.

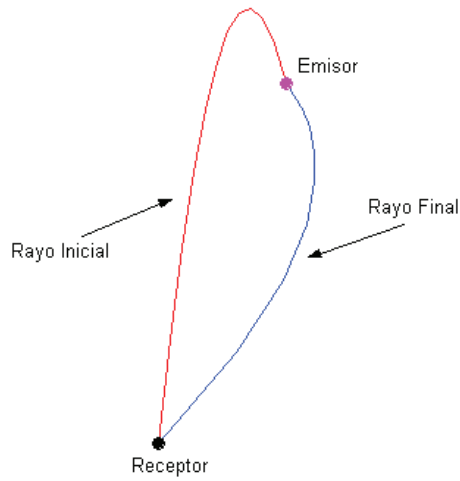


Figura 3.2: Trazado de rayo entre dos puntos en un medio con velocidad $v(x, y, z) = 10x + 3y - 4z + 10$.

3.3 Trazado de rayos en medios con interfaces

La teoría antes desarrollada se puede aplicar al trazado de rayos en medios que no tienen interfaces, es decir, donde la velocidad es una función continua en todo el medio. Por simplicidad, comencemos con un caso en el cual existe una sola discontinuidad en el campo de velocidad, o sea, cuando el rayo cruza una interface descrita por la superficie $z = \phi(x, y)$, obteniendo dos regiones, I y II, donde la velocidad de onda es distinta en cada región, v_I y v_{II} , respectivamente (ver Figura 3.3).

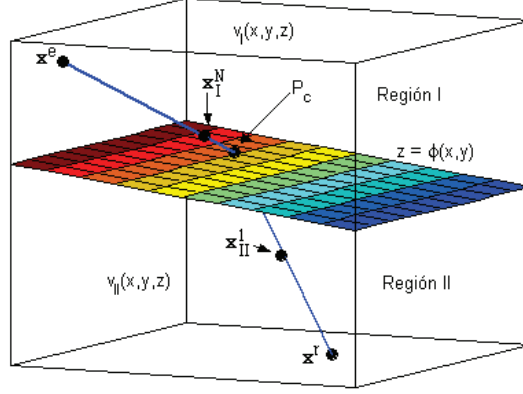


Figura 3.3: Medio con una interface.

Si se indica con subíndices las cantidades en las dos regiones y se denota con $P_c = (x_c, y_c, z_c)^t$ el punto de cruce del rayo con la interface, el siguiente sistema de ecuaciones no lineales se obtiene de (3.9) y del procedimiento descrito arriba:

$$\left\{ \begin{array}{l}
 \frac{\partial s_{I_i}}{\partial x_{I_i}} l_{I_i} + \frac{(x_{I_i} - x_{I_{i-1}})}{l_{I_i}} s_{I_i} + \frac{\partial s_{I_{i+1}}}{\partial x_{I_i}} l_{I_{i+1}} - \frac{(x_{I_{i+1}} - x_{I_i})}{l_{I_{i+1}}} s_{I_{i+1}} = 0, \\
 \frac{\partial s_{I_i}}{\partial y_{I_i}} l_{I_i} + \frac{(y_{I_i} - y_{I_{i-1}})}{l_{I_i}} s_{I_i} + \frac{\partial s_{I_{i+1}}}{\partial y_{I_i}} l_{I_{i+1}} - \frac{(y_{I_{i+1}} - y_{I_i})}{l_{I_{i+1}}} s_{I_{i+1}} = 0, \\
 \frac{\partial s_{I_i}}{\partial z_{I_i}} l_{I_i} + \frac{(z_{I_i} - z_{I_{i-1}})}{l_{I_i}} s_{I_i} + \frac{\partial s_{I_{i+1}}}{\partial z_{I_i}} l_{I_{i+1}} - \frac{(z_{I_{i+1}} - z_{I_i})}{l_{I_{i+1}}} s_{I_{i+1}} = 0, \\
 \frac{\partial s_{II_i}}{\partial x_{II_i}} l_{II_i} + \frac{(x_{II_i} - x_{II_{i-1}})}{l_{II_i}} s_{II_i} + \frac{\partial s_{II_{i+1}}}{\partial x_{II_i}} l_{II_{i+1}} - \frac{(x_{II_{i+1}} - x_{II_i})}{l_{II_{i+1}}} s_{II_{i+1}} = 0, \\
 \frac{\partial s_{II_i}}{\partial y_{II_i}} l_{II_i} + \frac{(y_{II_i} - y_{II_{i-1}})}{l_{II_i}} s_{II_i} + \frac{\partial s_{II_{i+1}}}{\partial y_{II_i}} l_{II_{i+1}} - \frac{(y_{II_{i+1}} - y_{II_i})}{l_{II_{i+1}}} s_{II_{i+1}} = 0, \\
 \frac{\partial s_{II_i}}{\partial z_{II_i}} l_{II_i} + \frac{(z_{II_i} - z_{II_{i-1}})}{l_{II_i}} s_{II_i} + \frac{\partial s_{II_{i+1}}}{\partial z_{II_i}} l_{II_{i+1}} - \frac{(z_{II_{i+1}} - z_{II_i})}{l_{II_{i+1}}} s_{II_{i+1}} = 0,
 \end{array} \right. \quad (3.16)$$

para $i = 1, 2, \dots, N$, donde $\mathbf{x}_I^0 = \mathbf{x}^e$, $\mathbf{x}_I^{N+1} = P_c = \mathbf{x}_{II}^0$ y $\mathbf{x}_{II}^{N+1} = \mathbf{x}^r$. Como el punto de cruce del rayo es otra incógnita, es necesario agregar tres ecuaciones auxiliares al sistema anterior. Observamos que \mathbf{x}_I^N y \mathbf{x}_{II}^1 son los puntos anterior y posterior, respectivamente,

al punto P_c (ver Figura 3.3). Nosotros tomaremos como ecuaciones auxiliares a:

$$\begin{cases} \frac{\partial s_{I_N}}{\partial x_c} l_{I_N} + \frac{(x_c - x_{I_N})}{l_{I_N}} s_{I_N} + \frac{\partial s_{II_1}}{\partial x_c} l_{II_1} - \frac{(x_{II_1} - x_c)}{l_{II_1}} s_{II_1} = 0, \\ \frac{\partial s_{I_N}}{\partial y_c} l_{I_N} + \frac{(y_c - y_{I_N})}{l_{I_N}} s_{I_N} + \frac{\partial s_{II_1}}{\partial y_c} l_{II_1} - \frac{(y_{II_1} - y_c)}{l_{II_1}} s_{II_1} = 0. \end{cases} \quad (3.17)$$

La última ecuación auxiliar viene dada por la ecuación

$$z_c = \phi(x_c, y_c), \quad (3.18)$$

que garantiza que el punto P_c pertenece a la interface. Esta última ecuación no necesita agregarse al sistema (3.16), ya que con las ecuaciones (3.17) se encuentran los valores de x_c y y_c , que sirven para calcular el valor de z_c a través de la ecuación (3.18).

De esta forma, el sistema (3.16) unido con las ecuaciones (3.17) y (3.18), describe el rayo que une a los puntos \mathbf{x}^e y \mathbf{x}^r y que atraviesa una interface en el punto de cruce P_c . En general, se puede usar una técnica similar cuando se tiene un número finito de interfaces. Para no entrar en detalles técnicos, simplemente se repite el procedimiento anterior para las m interfaces $\phi_i(x, y) = z$, $i = 1, 2, \dots, m$, que subdividen la región de interés en σ subregiones, en las que tenemos expresiones diferentes para la velocidad. Nosotros también asumimos que el rayo cruza las interfaces y entra en varias subregiones en un orden dado.

3.4 SANEG y el trazado de rayos sísmicos en 3D

Sean $\mathbf{x}^e = (x_0, y_0, 0)^t$ un emisor y $\mathbf{x}^r = (x_{N+1}, y_{N+1}, 0)$ un receptor. Emplearemos a SANEG para resolver el sistema no lineal que describe un rayo que atraviesa m interfaces definidas por las superficies $z = \phi_i(x, y)$, $i = 1, 2, \dots, m$. Supongamos que las interfaces no se intersectan entre sí en la región de interés. Llamemos capa a la subregión delimitada entre dos interfaces consecutivas. Como hay m interfaces, tendremos un medio de $m - 1$ capas. Consideraremos rayos que parten de \mathbf{x}^e y cruzan $1, 2, \dots$, ó m interfaces, hasta llegar a \mathbf{x}^r .

En la Figura 3.4 se muestra un medio con $m = 3$ interfaces definidas respectivamente por las ecuaciones: $z = \phi_0(x, y)$, $z = \phi_1(x, y)$ y $z = \phi_2(x, y)$. En esta figura observamos dos rayos C_1 y C_2 . El rayo C_1 va del emisor se refleja en la primera interface y llega al

receptor. El rayo C_2 va del emisor cruza la primera interface, se refleja en la segunda interface, luego cruza por la primera interface y por último llega al receptor. También observamos que C_1 atraviesa sólo una capa; en cambio C_2 atraviesa dos capas. Los rayos que atraviesan una sólo capa y se reflejan en la segunda interface, lo clasificaremos como tipo 1; a los rayos que atraviesan dos capas y se reflejan en la interface 3, lo clasificaremos como tipo 2. De esta forma, tendremos rayos de tipo $1, 2, \dots, m - 1$. Así, C_1 es un rayo tipo 1 y C_2 es un rayo tipo 2.

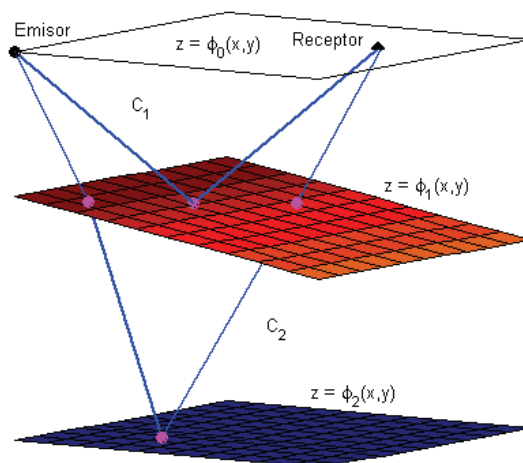


Figura 3.4: Tipos de rayos.

Aproximaremos N puntos del rayo entre dos puntos de cruces consecutivos y un punto por cada interface en donde éste cruza. En total, tendremos $2(k - 1)(N + 1) - 1$ puntos del rayo ha aproximar y el sistema de ecuaciones no lineales correspondiente tendrá una dimensión de $2(k - 1)(3N + 2) - 2$ ecuaciones con $2(k - 1)(3N + 2) - 2$ incógnitas, para cada rayo que cruza $k - 1$ interfaces y se refleja en la k -ésima interface, con $k = 1, 2, \dots, m$. Por ejemplo, aproximar $N = 5$ puntos del rayo C_2 entre cada punto de cruce, implicaría resolver un sistema no lineal de 66 ecuaciones con 66 incógnitas.

3.4.1 Ejemplos numéricos del trazado de rayos en 3D

A continuación damos dos ejemplos numéricos del trazado de rayos sísmicos en 3D. Los valores de los parámetros de SANEG que usamos son: $\gamma = 10^{-4}$, $\varepsilon = 10^{-10}$, $\sigma_1 = 0.1$, $\sigma_2 = 0.5$, $\alpha_0 = 1$, $M = 10$ y δ como en (2.33). Utilizamos (2.3), con $h = 10^{-7}$, para aproximar al escalar $F_k^t J_k F_k$. Utilizamos el modelo parabólico de dos puntos dado por

(2.34) y (2.35).

Para los ejemplos tomamos $N = 25$ puntos del rayo entre dos puntos de cruce consecutivos y un punto en cada interface donde éste cruce; por lo tanto, aproximamos $52(k-1)-1$ puntos de un rayo de tipo $k = 1, 2, \dots, m-1$. El algoritmo SANEG termina el proceso, si para cierta iteración i se satisface la desigualdad:

$$\|F_i\| \leq c \left(\frac{k}{k+1} \right) \|F_0\|,$$

para cada rayo de tipo $k = 1, 2, \dots, m-1$, donde $c = 0.9$ si $k = 1$, y $c = 0.8$ de lo contrario.

Ejemplo 1

El modelo consiste de un emisor $\mathbf{x}^e = (0, 0, 0)^t$ y de 150 receptores $\mathbf{x}^r = (x, y, 0)^t$, donde las coordenadas x e y son generadas aleatoriamente. El modelo posee 6 interfaces definidas como:

$$\begin{aligned}\phi_0(x, y) &= 0, \\ \phi_1(x, y) &= -(1/7)x + 1300, \\ \phi_2(x, y) &= (4/35)x + 2300, \\ \phi_3(x, y) &= (4/35)x + 2900, \\ \phi_4(x, y) &= 3200, \\ \phi_5(x, y) &= 3500,\end{aligned}$$

donde la ecuación $z = \phi_0(x, y)$ describe la primera interface en la cual están dispuestos el emisor y los receptores, y la ecuación $z = \phi_5(x, y)$ describe la última interface donde se reflejarán los rayos que lleguen a esta interface. Las velocidades en cada capa son:

$$\begin{aligned}v_1(x, y, z) &= 1.7x + y + 0.6z + 800, \\ v_2(x, y, z) &= 1.5x + y + 0.2z + 500, \\ v_3(x, y, z) &= 1.4x + y + 0.4z + 300, \\ v_4(x, y, z) &= 1.2x + y + 0.2z + 200, \\ v_5(x, y, z) &= 1.1x + y + 0.1z + 100.\end{aligned}$$

En la Tabla 3.1 se muestran los promedios de: iteraciones (IT), evaluaciones de la función $F(x)$ (EF), tiempo de CPU (T) expresado en segundos, para cada tipo de rayo.

Tipo de rayo	IT	EF	BL	T
1	18	52	13	0.30262
2	16	46	12	0.80468
3	13	37	10	1.0971
4	12	36	10	1.3958
5	12	34	8	1.7321

Tabla 3.1: Resultados numéricos del Ejemplo 1.

La Figura 3.5 ilustra el trazado de todos los rayos del disparo. En esta figura se aprecian 750 rayos. El tiempo promedio que requiere SANEG para obtener un rayo de cualquier tipo es 1.0665 segundos. Además, SANEG realizó un total de 10643 iteraciones, 30720 evaluaciones de $F(x)$ y 7928 búsquedas lineales para trazar 750 rayos.

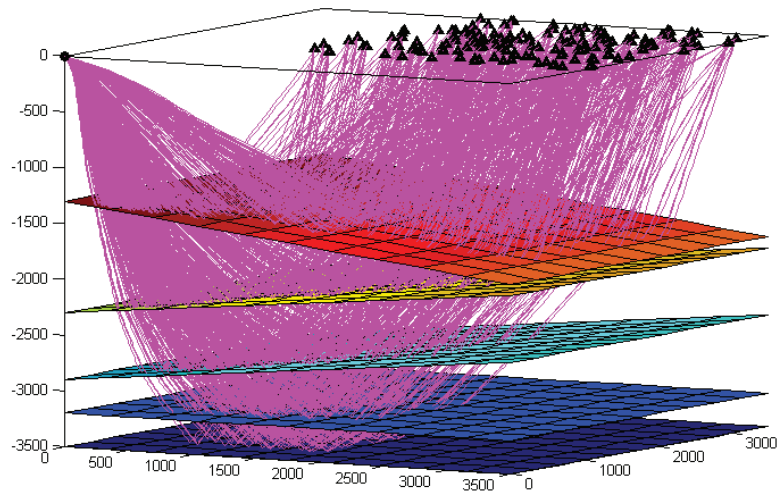


Figura 3.5: Trazado de rayos correspondiente al Ejemplo 1.

Ejemplo 2

El modelo consiste de un emisor $\mathbf{x}^e = (0, 0, 0)^t$ y de 150 receptores $\mathbf{x}^r = (x, y, 0)^t$, donde las coordenadas x e y son generadas aleatoriamente. El modelo posee 5 interfaces definidas como:

$$\begin{aligned}\phi_0(x, y) &= 0, \\ \phi_1(x, y) &= 40p \left(\frac{4x}{3800} - 2, \frac{4y}{3800} - 2 \right) - 1200, \\ \phi_2(x, y) &= 55p \left(\frac{5x}{3000} - 2.5, \frac{5y}{3000} - 2.5 \right) - 2500, \\ \phi_3(x, y) &= 70p \left(\frac{6x}{4500} - 3, \frac{6y}{4500} - 3 \right) - 3800, \\ \phi_4(x, y) &= 5000,\end{aligned}$$

donde la función $p(x, y)$ es:

$$p(x, y) = 3(1 - x)^2 e^{-(x^2 + (y+1)^2)} - 10 \left(\frac{x}{5} - x^3 - y^5 \right) e^{-(x^2 + y^2)} - (1/3) e^{-((x+1)^2 + y^2)}.$$

Las velocidades en cada capa son:

$$\begin{aligned}v_1(x, y, z) &= 2.2x + 2.5y + 0.5z + 800, \\ v_2(x, y, z) &= 1000 e^{-(x+y)/8000}, \\ v_3(x, y, z) &= \left(\frac{4}{7200} \right) (x^2 + y^2), \\ v_4(x, y, z) &= 1.2x + 1.3y + 0.1z + 200.\end{aligned}$$

En la Tabla 3.2 se muestran los promedios de: iteraciones (IT), evaluaciones de la función $F(x)$ (EF), tiempo de CPU (T) expresado en segundos, para cada tipo de rayo.

Tipo de rayo	IT	EF	BL	T
1	11	29	7	0.18435
2	8	21	4	0.41671
3	13	38	9	1.1554
4	13	36	9	1.4757

Tabla 3.2: Resultados numéricos del Ejemplo 2.

La Figura 3.6 ilustra el trazado de todos los rayos del disparo. En esta figura se aprecian 600 rayos. El tiempo promedio que requiere SANEG para obtener un rayo de cualquier tipo es 0.80805 segundos. Además, SANEG realizó un total de 6585 iteraciones, 18673 evaluaciones de $F(x)$ y 4321 búsquedas lineales para trazar los 600 rayos.

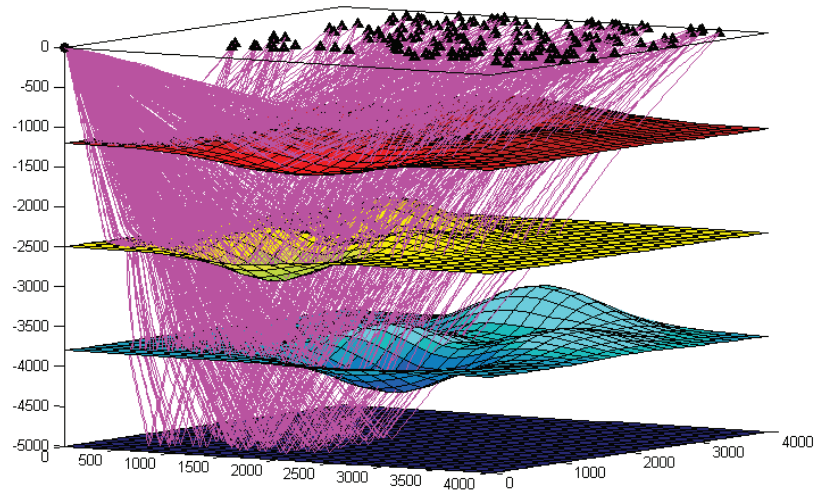


Figura 3.6: Trazado de rayos correspondiente al Ejemplo 2.

Conclusión

El trabajo de investigación realizado consta de dos grandes partes. La primera parte considera el diseño y estudio teórico-práctico de un método para la resolución de un sistema de ecuaciones no lineales. La segunda parte contempla el desarrollo de un nuevo enfoque para el trazado de rayos sísmicos en medios heterogéneos en 3D.

Construimos el método SANE como una adaptación del método Gradiente Espectral en la resolución del sistema de ecuaciones no lineales, $F(x) = 0$. Demostramos la convergencia global de SANE cuando $F(x)$ es el gradiente de una función cuadrática estrictamente convexa (cóncava), Teorema 2.1. Verificamos numéricamente que SANE es no monótono, que es una característica que heredó del método Gradiente Espectral.

Con el propósito de aplicar a SANE en la resolución del sistema $F(x) = 0$, cuando $F(x)$ es cualquier función $F : \mathbb{R}^n \rightarrow \mathbb{R}^n$, consideramos dos búsquedas lineales para combinarlas con SANE. La asociación de cada una de estas búsquedas lineales con SANE originó un algoritmo. El primero de ellos lo denominamos SANEG, que es la conjunción de SANE con una variante de la búsqueda lineal (2.1). El segundo algoritmo lo denominamos SANEF, el cual es la asociación de SANE con una variante de la búsqueda lineal (2.21).

Demostramos que SANEG y SANEF convergen globalmente, Teoremas 2.2 y 2.3 respectivamente. En el Paso 2 de ambos algoritmos se exige que en cada iteración se satisfaga $|F_k^t J_k F_k| / |F_k^t F_k| > \varepsilon$, con lo cual se garantiza que $d = -\text{sgn}(F_k^t J_k F_k) F_k$ es una dirección de descenso de la función mérito $f(x) = \|F(x)\|^2$. Demostramos que SANEG y SANEF, sin el Paso 2, convergen globalmente cuando la parte simétrica de $J(x)$ es positivo (negativo) definida, Corolarios 2.1 y 2.2 respectivamente.

Verificamos numéricamente que SANEG es más eficiente que SANEF y que éste último algoritmo tiende a la monotonía cuando aumenta el número de iteraciones, caso contrario a SANEG. Escogimos a SANEG para realizar un estudio comparativo de la eficiencia computacional con métodos tipo Newton Inexactos y el método de Broyden. Los resul-

tados de este experimento numérico verifican que SANEG es en promedio más eficiente y robusto que los otros métodos con los que se comparó. Inferimos que estos resultados se deben a que SANEG utiliza reiteradamente las direcciones elementales $F(x)$ y $-F(x)$. La sencillez de estas direcciones ocasiona que el costo computacional sea muy bajo por iteración. En cambio, los métodos tipo Newton Inexactos y el método de Broyden, requieren de la resolución numérica de un sistema de ecuaciones lineales para generar las direcciones de búsqueda, incurriendo, de esta forma, en un elevado costo de cómputo por iteración. También es importante acotar que SANEG almacena sólo un vector por iteración, lo cual lo hace muy competitivo con respecto a los métodos Newton inexactos que emplean algoritmos iterativos para sistemas de ecuaciones lineales basados en el subespacio de Krylov.

Otra observación relevante es que el número $|\alpha_k|$ puede, en algunos momentos, acotar la condición de J_k . Por ello, el algoritmo SANEG probablemente tenga problemas cuando J_k sea muy mal condicionado en alguna iteración k . Cuando esto sucede el algoritmo es inestable numéricamente, originando un mayor número de iteraciones y búsquedas lineales. También cabe resaltar que el valor de M juega un papel importante en el funcionamiento de SANEG; por ejemplo, si consideramos la función 4 con $n = 999$ y tomamos $M = 8$, SANEG converge en 109 iteraciones, realizando 264 evaluaciones de F y 40 búsquedas lineales, en un tiempo de 3.22 s, lo cual mejora los resultados obtenidos para $M = 10$.

En cuanto a la segunda parte de la investigación, desarrollamos un nuevo enfoque para el trazado de rayos sísmicos en medios heterogéneos en 3D, basado en el Principio de Fermat. Asociamos un sistema de ecuaciones no lineales al trazado de un rayo entre dos puntos, forzando a que en un número finito de puntos del rayo se satisfaga la estacionalidad del Principio de Fermat. Demostramos que cuando la velocidad del medio es constante, el rayo entre dos puntos es el segmento de recta que une tales puntos (Teorema 3.1). De la misma forma, asociamos un sistema de ecuaciones no lineales al trazado de un rayo que atraviesa varias capas con distintas velocidades de onda. Como el nuevo enfoque para el trazado de rayos considera la resolución de un sistema de ecuaciones no lineales, diseñamos experimentos numéricos para aplicar el método SANEG en el trazado de rayos sísmicos. Los resultados obtenidos en estos experimentos nos indican que SANEG resuelve satisfactoriamente el problema del trazado de rayos.

Finalmente, como problemas de investigación de gran importancia y que aún se encuen-

tran sin respuesta, podemos enumerar los siguientes:

- *Diseñar una búsqueda lineal que no considere aproximación de ningún tipo de la Matriz Jacobiana de $F(x)$ ó del escalar $F(x)^t J(x) F(x)$, y construir un algoritmo con la asociación de SANE con tal búsqueda lineal, que sea tan o más eficiente que SANEG.*
- *Diseñar la versión preconditionada de SANEG y estudiar su convergencia global .*
- *Diseñar y estudiar una versión dinámica del nuevo enfoque para el trazado de rayos sísmicos. Es decir, asociar un sistema de ecuaciones no lineales al trazado de rayos que contemple que el número de puntos ha aproximar en cada capa varie según la distancia y la trayectoria del rayo entre cada interface.*

Anexo A

Funciones de Prueba

A continuación se enumeran 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.*

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

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

2. *Función exponencial 2.*

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

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

3. *Función exponencial 3.*

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

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

4. *Función diagonal de tres variables premultiplicada por una matriz ortogonal (n múltiplo de 3) [31].*

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

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

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

5. *Función diagonal de tres variables premultiplicada por una matriz ortogonal, combinada con una función trigonométrica inversa* (n múltiplo de 3) [28].

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

$$\begin{aligned} f_{3i-2}(x) &= 64(x_{3i-2} + x_{3i-1} + x_{3i}) - 0.64 + 0.48\text{atan}(x_{3i}) \\ &\quad + 0.60(c_1 + c_2x_{3i-1} + c_3x_{3i-1}^2 + c_4x_{3i-1}^3), \\ f_{3i-1}(x) &= 0.48 - 48(x_{3i-2} + x_{3i-1} + x_{3i}) + 0.36\text{atan}(x_{3i}) \\ &\quad + 0.80(c_1 + c_2x_{3i-1} + c_3x_{3i-1}^2 + c_4x_{3i-1}^3), \\ f_{3i}(x) &= 0.60 - 60(x_{3i-2} + x_{3i-1} + x_{3i}) + 0.80\text{atan}(x_{3i}), \end{aligned}$$

donde

$$\begin{aligned} c_1 &= 13.901020408163270000 \\ c_2 &= -1.405612244897960000 \\ c_3 &= -2.218367346938776000 \\ c_4 &= -0.2770408163265306000 \end{aligned}$$

Iterado inicial: $x_0 = (-1.99, 2, 10^{-5}, -1.99, 2, 10^{-5}, \dots)^t$.

6. *Función extendida de Rosenbrock* (n par) [31].

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) [28].

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) [28].

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* [38].

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 tomará $c = 0.9$).

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

10. Función mala Powell (n par) [28].

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) [31].

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.

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

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

13. Función trigonométrica desplazada y aumentada en una esfera Euclidea [28].

$$\begin{aligned} f_i(x) &= n - 1 - \sum_{j=1}^{n-1} \cos(x_j - 1) + i(1 - \cos(x_i - 1)) - \operatorname{sen}(x_i - 1), \\ &\text{para } i = 1, \dots, n - 1, \\ f_n(x) &= \sum_{j=1}^n x_j^2 - 10000. \end{aligned}$$

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

14. Función Singular.

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

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

15. *Función Logarítmica.*

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

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

16. *Función de Broyden Tridiagonal [33].*

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

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

17. *Función Trigexp [33].*

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

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

18. *Función de Banda Variable 1 [33].*

$$\begin{aligned} f_1(x) &= -2x_1^2 + 3x_1 - 2x_2 + 0.5x_{\alpha_1} + 1, \\ f_i(x) &= -2x_i^2 + 3x_i - x_{i-1} - 2x_{i+1} + 0.5x_{\alpha_i} + 1, \text{ para } i = 2, 3, \dots, n-1, \\ f_n(x) &= -2x_n^2 + 3x_n - x_{n-1} + 0.5x_{\alpha_n} + 1. \end{aligned}$$

donde α_i números enteros aleatorios en el intervalo $[\alpha_{i_{\min}}, \alpha_{i_{\max}}]$, donde $\alpha_{i_{\min}} = \max\{1, i-2\}$ y $\alpha_{i_{\max}} = \min\{n, i+2\}$, para todo i .

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

19. *Función de Banda Variable 2 [33].*

$$\begin{aligned} f_1(x) &= -2x_1^2 + 3x_1 - 2x_2 + 0.5x_{\alpha_1} + 1, \\ f_i(x) &= -2x_i^2 + 3x_i - x_{i-1} - 2x_{i+1} + 0.5x_{\alpha_i} + 1, \text{ para } i = 2, 3, \dots, n-1, \\ f_n(x) &= -2x_n^2 + 3x_n - x_{n-1} + 0.5x_{\alpha_n} + 1. \end{aligned}$$

donde α_i son números enteros aleatorios en el intervalo $[\alpha_{i_{\min}}, \alpha_{i_{\max}}]$, donde $\alpha_{i_{\min}} = \max\{1, i-10\}$ y $\alpha_{i_{\max}} = \min\{n, i+10\}$, para todo i .

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

20. *Función 20 [33].*

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

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

21. *Función estrictamente convexa 1 [52].*

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

22. *Función estrictamente convexa 2* [52].

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

23. *Función 23* (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: $x_0 = (0, 0, \dots, 0)^t$.

24. *Función lineal de rango full.*

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

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

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

26. *Función penalidad I.*

$$\begin{aligned} f_i(x) &= \sqrt{10^{-5}}(x_i - 1), \text{ para } i = 1, 2, \dots, n-1, \\ f_n(x) &= \left(\frac{1}{4n}\right) \sum_{j=1}^n x_j^2 - \frac{1}{4}. \end{aligned}$$

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

27. *Función casi morena.*

$$\begin{aligned} f_i(x) &= x_i + \sum_{j=1}^n x_j - (n+1), \text{ para } i = 1, 2, \dots, n-1, \\ f_n(x) &= \prod_{j=1}^n x_j - 1. \end{aligned}$$

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

28. *Función de dimensión variable.*

$$\begin{aligned} f_i(x) &= x_i - 1, \text{ para } i = 1, 2, \dots, n-2, \\ f_{n-1}(x) &= \sum_{j=1}^{n-2} j(x_j - 1), \\ f_n(x) &= \left(\sum_{j=1}^{n-2} j(x_j - 1) \right)^2. \end{aligned}$$

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

29. *Función geométrica.*

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

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

30. *Función singular de Powel extendida* (n múltiplo de 4) [45].

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

$$\begin{aligned} f_{4i-3}(x) &= x_{4i-3} + 10x_{4i-2}, \\ f_{4i-2}(x) &= \sqrt{5}(x_{4i-1} - x_{4i}), \\ f_{4i-1}(x) &= (x_{4i-2} - 2x_{4i-1})^2, \\ f_{4i}(x) &= \sqrt{10}(x_{4i-3} - x_{4i})^2. \end{aligned}$$

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

31. *Función 31.*

$$\begin{aligned} f_1(x) &= \sum_{j=1}^n x_j^2, \\ f_i(x) &= -2x_1x_i, \text{ para } i = 2, 3, \dots, n. \end{aligned}$$

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

32. *Función de valle tridimensional* (n múltiplo de 3) [28].

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

$$\begin{aligned} f_{3i-2}(x) &= (c_2x_{3i-2}^3 + c_1x_{3i-2}) \exp\left(\frac{-x_{3i-2}^2}{100}\right) - 1, \\ f_{3i-1}(x) &= 10(\text{sen}(x_{3i-2}) - x_{3i-1}), \\ f_{3i}(x) &= 10(\text{cos}(x_{3i-2}) - x_{3i}), \end{aligned}$$

donde

$$\begin{aligned} c_1 &= 1.003344481605351 \\ c_2 &= -3.344481605351171 \times 10^{-3} \end{aligned}$$

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

33. *Función complementaria (n par).*

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

$$\begin{aligned}f_{2i-1}(x) &= \left(x_{2i-1}^2 + \left(x_{2i-1} e^{x_{2i-1}} - \frac{1}{n} \right)^2 \right)^{1/2} - x_{2i-1} - x_{2i-1} e^{x_{2i-1}} + \frac{1}{n}, \\f_{2i}(x) &= \left(x_{2i}^2 + (3x_i + \text{sen}(x_{2i}) + e^{x_{2i}})^2 \right)^{1/2} - x_{2i} - 3x_{2i} - \text{sen}(x_{2i}) - e^{x_{2i}}.\end{aligned}$$

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

34. *Función mínima.*

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

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

35. *Función Guía.*

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

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

36. *Sistema tridiagonal [40].*

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

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

37. *Sistema cinco-diagonal [40].*

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

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

38. *Sistema siete-diagonal* [40].

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

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

39. *Función extendida de Freudenstein y Roth* (n par) [7].

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

40. *Problema extendido de Cragg y Levy* (n múltiplo de 4) [45].

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: $x_0 = (4, 2, 2, 2, 4, 2, 2, 2, \dots)^t$.

41. *Problema extendido de Wood* (n múltiplo de 4) [35].

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

42. *Problema tridiagonal exponencial* [7].

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

Iterado inicial: $x_0 = (1.5, \dots, 1.5)^t$.

43. *Problema discreto de valores en el borde* [45].

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

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

44. *Problema de Brent* [1].

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

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

45. *Problema de Troesch* [53].

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

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

46. *Sistema trigonométrico* [57].

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

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

47. *Sistema trigonométrico-exponencial* [57].

$$\begin{aligned} f_1(x) &= 3(x_1 - x_3)^3 - 5 + 2x_3 + \sin(x_1 - x_2 - x_3)\sin(x_1 + x_2 - x_3), \\ f_i(x) &= -6(x_{i-2} - x_i)^3 + 10 - 4x_{i-1} \\ &\quad - 2\sin(x_{i-2} - x_{i-1} - x_i)\sin(x_{i-2} + x_{i-1} - x_i) \\ &\quad + 2\sin(x_i - x_{i+1} - x_{i+2})\sin(x_i + x_{i+1} - x_{i+2}) \\ &\quad + 3(x_i - x_{i+2})^3 - 5 + 2x_{i+1}, \text{ si } \text{mod}(i, 2) = 1, 1 < i < n, \\ f_i(x) &= 4x_i - (x_{i-1} - x_{i+1} \exp(x_{i-1} - x_i - x_{i+1}) - 3), \text{ si } \text{mod}(i, 2) = 0, 1 < i < n, \\ f_n(x) &= -6(x_{n-2} - x_n)^3 + 10 - 4x_{n-1} \\ &\quad - 2\sin(x_{n-2} - x_{n-1} - x_n)\sin(x_{n-2} + x_{n-1} - x_n). \end{aligned}$$

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

48. *Fluido en un canal* [2].

Es el sistema de diferencias finita asociado a la ecuación diferencial ordinaria no lineal

$$u'''' = R(u'u'' - uu'''), \quad R = 500$$

sobre el intervalo unitario con condiciones de borde $u(0) = 0$, $u'(0) = 0$, $u(1) = 1$, $u'(1) = 0$. Usamos las diferencias finitas estandar de 5-puntos sobre una rejilla uniforme de n nodos internos. La aproximación inicial es la discretización de $u_0(x) = 0$.

49. *Problema de valores en el borde en dos puntos* [39].

Es el sistema de diferencias finita

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

asociado al problema de valores en el borde en dos puntos

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

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

50. *Sistema de ecuaciones diferenciales.*

Es el sistema de diferencias finita asociado al sistema de tres ecuaciones diferenciales ordinarias no lineales

$$u'' = (u' + v' + w')u' + 1, \quad v'' = (u' + v' + w')v' + 1, \quad w'' = (u' + v' + w')w' + 1,$$

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

La aproximación inicial es: $u_0 = v_0 = w_0 = \left(\frac{1}{n}, \dots, \frac{1}{n}\right)^t$.

Bibliografía

- [1] G. Alefeld, A. Gienger, and F. Potra. Efficient validation of solutions of nonlinear systems. *SIAM Journal on Numerical Analysis*, 31:252–260, 1994.
- [2] B.M. Averick, R.G. Carter, and J.J. More. The Minpack-2 test problem collection. Technical Report ANL/MCS-TM-150, Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, 1991.
- [3] J. Barzilai and J. Borwein. Two point step size gradient methods. *IMA J. Numer. Anal.*, 8:141–148, 1988.
- [4] S. Bellavia and B. Morini. A globally convergent Newton-GMRES subspace method for systems of nonlinear equations. *SIAM J. Sci.*, 23:940–960, 2001.
- [5] J. G. Berryman. Lecture Notes on Nonlinear Inversion and Tomography: I. Borehole Seismic Tomography. Technical Report CA 94550, University of California Lawrence Livermore National Laboratory, Livermore, October 1991.
- [6] R. H. Bielschowsky, A. Friedlander, F. A. M. Gomes, J. M. Martínez, and M. Raydán. An adaptive algorithm for bound constrained quadratic minimization. *Investigación Operativa*, 7:67–102, 1997.
- [7] Y. Bing and G. Lin. An efficient implementation of Merrill’s method for sparse or partially separable systems of nonlinear equations. *SIAM Journal on Optimization*, 2:206–221, 1991.
- [8] E. G. Birgin, I. Chambouleyron, and J. M. Martínez. Estimation of the optical constants and the thickness of thin films using unconstrained optimization. *Journal of Computational Physics*, 151:862–880, 1999.

- [9] E. G. Birgin and Y. G. Evtushenko. Automatic differential and spectral projected gradient methods for optimal control problems. *Optimization Methods and Software*, 10:125–146, 1998.
- [10] E. G. Birgin, J. M. Martínez, and M. Raydán. Nonmonotone spectral projected gradient methods on convex sets. *SIAM Journal on Optimization*, 10(4):1196–1211, 2000.
- [11] I. D. L. Bogle and J. D. Perkins. A new sparsity-preserving quasi-Newton update for solving nonlinear equations. *SIAM J. Sci. Stat. Comp.*, 11:621–630, 1990.
- [12] I. Borg and P. J. F. Groenen. *Modern Multidimensional Scaling: Theory and Applications*. Springer-Verlag, Berlin, 1997.
- [13] P. N. Brown and Y. Saad. Hybrid Krylov methods for nonlinear systems of equations. *SIAM J. Sci. Comput.*, 11:450–481, 1990.
- [14] P. N. Brown and Y. Saad. Theory of nonlinear Newton-Krylov algorithms. *SIAM J. Opt.*, 4:297–330, 1994.
- [15] C. G. Broyden. A class of the methods for solving nonlinear simultaneous equations. *Math. Comput.*, 25:577–593, 1965.
- [16] Z. Castillo, D. Cores, and M. Raydán. Low cost optimization techniques for solving the nonlinear seismic reflection tomography problem. *Optimization and Engineering*, 1:155–169, 2000.
- [17] D. Cores, G. Fung, and R. Michelena. A fast and global two point low storage optimization technique for tracing rays in 2d and 3d isotropic media. *Journal of Applied Geophysics*, 45:273–287, 2000.
- [18] Y. H. Dai and L. Z. Liao. R-linear convergence of the Barzilai and Borwein gradient method. *IMA Journal of Numerical Analysis*, 22:1–10, 2002.
- [19] R. Dembo, S. Eisenstat, and T. Steihaug. Inexact Newton methods. *SIAM J. Numer. Anal.*, 19:400–408, 1982.

- [20] J. E. Dennis and J. J. Moré. A characterization of superlinear convergence and its applications to quasi-newton methods. *Math. Comput.*, 28:549–560, 1974.
- [21] J. E. Dennis and R. B. Schnabel. *Numerical Methods for Nonlinear Equations and Unconstrained Optimization*. Prentice-Hall, Englewood Cliffs, NJ, 1983.
- [22] M. A. Diniz-Ehrhardt, M. A. Gomes-Ruggiero, J. M. Martínez, and A. Santos. Augmented Lagrangian algorithms based on the spectral gradient for solving nonlinear programming problems. Technical Report, Institute of Mathematics, University of Campinas, Brazil, 2000.
- [23] S. C. Eisenstat and H. Walker. Globally convergent inexact Newton methods. *SIAM J. Opt.*, 4:393–422, 1994.
- [24] R. Fletcher. *Practical Methods of Optimization*. Wiley, New York, 1987.
- [25] R. Fletcher. Low storage methods for unconstrained optimization. *Lectures in Applied Mathematics (AMS)*, 26:165–179, 1990.
- [26] R. Fletcher. On the Barzilai-Borwein method. Technical Report NA/207, Department of Mathematics, University of Dundee, Dundee, Scotland, UK, 2001.
- [27] R. W. Freund. A transpose-free quasi-minimal residual algorithm for non-hermitian linear systems. *SIAM Journal on Scientific Computing*, 4:470–482, 1993.
- [28] A. Friedlander, M. A. Gomes-Ruggiero, D. N. Kozakevich, J. M. Martínez, and S. A. Santos. Solving nonlinear systems of equations by mean of quasi-Newton methods with a nonmonotone strategy. *Optimization Methods and Software*, 8:25–51, 1997.
- [29] A. Friedlander, J. M. Martínez, B. Molina, and M. Raydán. Gradient method with retards and generalizations. *SIAM J. Numer. Anal.*, 36(1):275–289, 1999.
- [30] A. Friedlander, J. M. Martínez, and M. Raydán. A new method for large-scale constrained convex quadratic minimization problems. *Opt. Methods and Software*, 5:57–74, 1995.
- [31] M. G. Gasparo. A nonmonotone hybrid method for nonlinear systems. *Optimization Meth & Soft*, 13:79–94, 2000.

- [32] W. Glunt, T. L. Hayden, and M. Raydán. Molecular conformation from distance matrices. *J. Comp. Chem.*, 14:114–120, 1993.
- [33] M. Gomes-Ruggiero, J. Martínez, and A. Moretti. Comparing Algorithms for Solving Sparse Nonlinear Systems of Equations. *SIAM J. Sci. Stat. Comp.*, 23:459–483, 1992.
- [34] L. Grippo, F. Lampariello, and S. Lucidi. A nonmonotone line search technique for Newton’s method. *SIAM J. Numer. Anal.*, 23(4):707–716, 1986.
- [35] S. Incerti, F. Zirilli, and V. Parisi. Algorithm 111. A fortran subroutine for solving systems of nonlinear simultaneous equations. *Computer Journal*, 24:87–91, 1981.
- [36] P. L. Jackson. *Digital simulation of seismic waves*. PhD thesis, University of Michigan, Michigan, 1970.
- [37] H. Jacob. Three dimensional seismic ray tracing in a laterally heterogeneous spherical earth. *J. Geophys. Res.*, 75:6685–6689, 1970.
- [38] C. T. Kelley. *Iterative Methods for Linear and Nonlinear Equations*. SIAM, Philadelphia, 1995.
- [39] D. Li and M. Fukushima. A derivative-free line search and global convergence of Broyden-Like Method for nonlinear equations. *Optimization Meth & Soft.*, 13:181–201, 2000.
- [40] G. Li. Successive column correction algorithms for solving sparse nonlinear systems of equations. *Mathematical Programming*, 43:187–207, 1989.
- [41] F. Luengo, M. Raydán, W. Glunt, and T. Hayden. Preconditioned spectral gradient method. *Numerical Algorithms*, 30:241–258, 2002.
- [42] W. Mao and G. W. Stuart. Rapid multi-wave-type ray tracing in complex 2D and 3D isotropic media. *Geophysics*, 62(1):298–308, 1997.
- [43] J. M. Martínez. Local convergence theory of inexact Newton methods based on structured least change secante updates. *Math. Comp.*, 55:143–167, 1990.
- [44] B. Molina and M. Raydán. Preconditioned Barzilai-Borwein method for the numerical solution of partial differential equations. *Numerical Algorithms*, 13:45–60, 1996.

- [45] J.J. More, B.S. Garbow, and K.E. Hillstom. Testing unconstrained optimization software. *ACM Transactions on Mathematical Software*, 7:17–41, 1981.
- [46] I. Newton. *The Mathematical Papers of Isaac Newton (7 volumes)*. D. T. Whiteside, ed., Cambridge University Press, Cambridge, 1967-1976.
- [47] V. Pereyra. Two-point seismic-ray tracing in general 3D media. *Geophysical Prospecting*, 40:267–287, 1992.
- [48] V. Pereyra, W. H. Lee, and H. B. Keller. Solving two-point seismic-ray tracing problems in a heterogeneous medium. *Bulletin of the Seismological of America*, 70(1):79–99, 1980.
- [49] W. Prothero, W. J. Taylor, and J. A. Eickemeyer. A fast two-point three-dimensional ray tracing algorithm using a simple search method. *Bull. Seismol. Soc. Am.*, 78:1190–1198, 1988.
- [50] J. Raphson. *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.
- [51] M. Raydán. On the Barzilai-Borwein choice of steplength for the gradient method. *IMA J. Numer. Anal.*, 13:321–326, 1993.
- [52] M. Raydán. The Barzilai and Borwein Gradient Method for the large Scale Unconstrained Minimization Problems. *SIAM*, 7(1):26–33, 1997.
- [53] S.M. Roberts and J.S. Shipman. On the closed form solution of Troesch’s Problem. *Journal of Computational Physical*, 21:291–304, 1976.
- [54] Y Sadd and M. H. Shultz. GMRES: generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM Sci. Stat. Comput.*, 7:856–869, 1986.
- [55] A. H. Sherman. On Newton-iterative methods for the solution of systems of nonlinear equations. *SIAM J. Numer, Anal.*, 15:775–771, 1978.
- [56] M.V. Solodov and B.F. Svaiter. A globally convergent Inexact Newton Method for Systems of Monotone Equations. *Semismooth and Smooting Methods*, pages 355–369, 1998.

- [57] P.L. Toint. Numerical solution of large sets of algebraic equations. *Mathematics of Computation*, 46:175–189, 1986.
- [58] M. W. Trosset. Numerical algorithms for MDS. In R. Klar and P. Opitz, editors, *Classification and Knowledge Organization*, pages 80–92. Springer, Berlin, 1994.
- [59] J. Um and C. Thurber. A fast algorithm for two-point seismic ray tracing. *Bull. Seismol. Soc. Am.*, 77:972–986, 1987.
- [60] H. A. Vander Vorst. Bi-CGSTAB: a fast and smoothly convergent variant Bi-CG for the solution of non-symmetric linear systems. *SIAM Sci. Stat. Comput.*, 13:631–644, 1992.
- [61] C. Wells, W. Glunt, and T. L. Hayden. Searching conformational space with the spectral distance geometry algorithm. *Journal of Molecular Structure (Theochem)*, 308:263–271, 1994.
- [62] R. L. Wesson. *Seismic ray computation in laterally inhomogeneous crustal models*. PhD thesis, Stanford University, California, 1970.
- [63] R. L. Wesson. Traveltime inversion for laterally inhomogeneous crustal velocity models. *Bull. Seismol. Soc. Am.*, 61:729–746, 1971.