



03063
2
2ej.

Universidad Nacional Autónoma de México

Unidad Académica de los Ciclos Profesional y de Posgrado del
Colegio de Ciencias y Humanidades

FALLA DE ORIGEN

**Método de tunelización exponencial
para resolver el problema de optimización global sin
restricciones**

Tesis
Que para obtener el título de:
Maestro en Ciencias de la Computación
Presenta
Carlos Barrón Romero

México 1991



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas Tesis Digitales Restricciones de uso

DERECHOS RESERVADOS © PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis está protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

Resumen de la tesis:

El método de tunelización exponencial para resolver el problema de optimización global sin restricciones



Carlos Barrón Romero

El trabajo describe un nuevo método para resolver el problema de optimización global sin restricciones:

$$\min_{x \in H} f(x)$$

donde $f: R^n \rightarrow R$ es una función continua que tiene derivadas parciales continuas hasta de orden 2 y $H = [a_1, b_1] \times [a_n, b_n]$ es un hipercubo de R^n .

Este nuevo método se nombró método de tunelización Exponencial y es una continuación natural de los métodos de tunelización Original (Alejandro V. Levy, Antonio Montalvo, et.al.) y Clásico (Alejandro V. Levy, Susana Gómez, Alfredo Cortés, Apolinar Calderón, et. al.).

Los métodos de Tunelización son una de las principales aportaciones que México ha aportado al área de la Optimización Global, fueron desarrollados en el Instituto de Investigaciones en Matemáticas Aplicadas y en Sistemas (IIMAS) de la Universidad Nacional Autónoma de México.

En la tesis se describe un panorama de los métodos de optimización global, destacando el método de tunelización Clásico, se expone el nuevo método y la transformación exponencial de tunelización y se introduce la metodología de tunelización (Susana Gómez y Carlos Barrón). Se dan los resultados de experimentos computacionales de un análisis de sensibilidad de los parámetros del nuevo método y de una comparación con los métodos de tunelización Clásico y Original para una gran variedad de problemas con distinto grado de dificultad.

Los resultados experimentales que se reportan en el trabajo muestran que el nuevo método es superior a las versiones anteriores.

Vo.Bo.

Susana Gómez,
Investigadora del IIMAS

Contenido

Prefacio	1
1 Panorama de la optimización local y global sin restricciones	4
1.1 El problema de optimización global sin restricciones	4
1.1.1 Condiciones de existencia del punto mínimo global	5
1.1.2 Condiciones de optimalidad	6
1.2 Aspectos de la solución de los problemas de optimización global	11
1.2.1 Funciones apéndice	14
1.2.2 Función aplana	14
1.3 Métodos locales para resolver problemas de optimización sin restricciones	14
1.3.1 Métodos de Newton y de métrica variable	15
1.3.2 Métodos de direcciones conjugadas	16
1.4 Métodos para el problema de optimización global sin restricciones	17
1.4.1 Método de Solís y Wets	18
1.4.2 Método de la cobertura	20
1.5 Aplicaciones de la optimización global	23
2 Métodos de tunelización: original y clásico	26
2.1 Origen del método de tunelización	26
2.2 Función de tunelización clásica	30
2.3 Propiedades de los métodos de tunelización	36
2.4 Método de tunelización clásico	37
2.4.1 Cambios en la fase de tunelización del método clásico	39
2.4.2 Algoritmo de la fase de tunelización del método clásico	42
2.4.3 Detalles de la programación del método clásico	43
3 Método de tunelización exponencial	46
3.1 Función de tunelización exponencial	46
3.1.1 Efectos geométricos de las funciones de tunelización y de relleno	49
3.1.2 Funciones de relleno	51
3.2 Método de tunelización exponencial	52
3.2.1 Algoritmo de la fase de tunelización del método exponencial	53
3.2.2 Detalles de la programación del método exponencial	54
3.2.3 Factor para eliminar los mínimos del mismo nivel	55
3.3 Diferencias entre las funciones de tunelización	55
3.4 Diferencias entre los métodos de tunelización	56

3.5	Metodología de tunelización	57
4	Diseño y programación de los métodos de tunelización	61
4.1	Lenguajes de programación para algoritmos numéricos	61
4.2	Descripción de un sistema aritmético	63
4.3	Cálculo de los parámetros de un sistema aritmético	69
4.4	Manejo de números grandes	71
4.5	Detalles del diseño y programación de los métodos de tunelización	72
4.6	Rutinas de los programas: CLASICO y EXPONEN	75
5	Experimentos numéricos	81
5.1	Una forma de evaluar los métodos de optimización global	81
5.2	Descripción de los experimentos numéricos	82
5.3	Análisis de sensibilidad del método de tunelización exponencial	90
5.4	Comparación de los métodos de tunelización: original, clásico y exponencial	94
5.5	Comparación de los métodos de las funciones de relleno y de tunelización: clásico y exponencial	95
5.6	El potencial de Lennard-Jones 12-6 y los métodos de tunelización: clásico y exponencial	96
6	Conclusiones	100
	Bibliografía	102
A	Ambientes computacionales para la investigación	107

Tablas

1.1	ϵ y σ para el potencial de Lennard-Jones 12-6	25
2.1	Cambios entre los métodos de tunelización original y clásico	40
2.2	Cambios entre los métodos de tunelización original y clásico (cont.)	41
3.1	Factores de los métodos de tunelización	56
5.1	Análisis de sensibilidad del parámetro λ^* del método de tunelización exponencial	88
5.2	Análisis de sensibilidad del parámetro λ^* del método de tunelización exponencial (cont.)	89
5.3	Método de tunelización exponencial, % de éxito por rangos de los 13 experimentos para distintos valores del parámetro λ^*	91
5.4	Método de tunelización exponencial, total de óptimos locales de los 13 experimentos para distintos valores del parámetro λ^*	91
5.5	Comparación de los métodos de tunelización	92
5.6	Comparación de los métodos de tunelización (cont.)	93
5.7	Comparación de los métodos de tunelización, % de éxito por rangos de los 13 experimentos	94
5.8	Comparación de los métodos de las funciones de relleno y de tunelización	95
5.9	Potencial mínimo de Lennard-Jones 12-6 de 7 partículas esféricas idénticas alcanzado a partir de una configuración casi óptima	96
5.10	Potencial mínimo de Lennard-Jones 12-6 de 7 partículas esféricas idénticas alcanzado a partir de las configuraciones 1 y 2	97
5.11	Potencial mínimo de Lennard-Jones 12-6 de 7 partículas esféricas diferentes alcanzado a partir de las configuraciones 1 y 2	98
5.12	Potencial mínimo de Lennard-Jones 12-6 de 72 partículas esféricas diferentes alcanzado a partir de una configuración inicial	99

Figuras

1.1	Método de la Cobertura aplicado a una función continua	23
1.2	Potencial Lennard-Jones 12-6 de dos partículas esféricas idénticas	25
1.3	Configuración óptima de 19 partículas esféricas idénticas	25
2.1	Método de Goldstein y Price, paso 1 y 2	28
2.2	Método de Goldstein y Price, paso 3 y 4	29
2.3	Función de tunelización clásica de la función de Shubert	32
2.4	Función de tunelización clásica con rampeo y sin rampeo	33
2.5	Efecto de la potencia del polo λ^* en la función de tunelización clásica	35
2.6	Función de tunelización clásica de la función de Shubert con un polo móvil	39
2.7	“Polo” si x^* no es exactamente un punto mínimo local	44
3.1	Función de tunelización clásica con rampeo y un polo en 0	49
3.2	Funciones de tunelización exponencial con un polo en 0	50
3.3	Función de relleno de f_2	51
3.4	Función de tunelización exponencial de f_2	51
4.1	Diagrama de bloques de los programas de tunelización	73
5.1	Función original del experimento 13	86
5.2	Función del experimento 13 modificada con dos apéndices	86
5.3	Mínimos de la función del experimento 13 modificada con dos apéndices	87
5.4	Configuración uno de 7 de partículas	96
5.5	Configuración dos de 7 de partículas	96
A.1	Ambiente ideal para la investigación y la elaboración de reportes	109

Prefacio

El trabajo describe un nuevo método para resolver el problema de optimización global sin restricciones:

$$\min_{z \in H} f(x)$$

donde $f: \mathbb{R}^n \mapsto \mathbb{R}$ es una función continua que tiene derivadas parciales hasta de orden 2 y continuas y $H = [a_1, b_1] \times \cdots \times [a_n, b_n]$ es un hipercubo de \mathbb{R}^n .

Dentro del área de la optimización cae el problema y el método que se describe. El área de la optimización engloba toda la teoría de los métodos que buscan uno o más puntos óptimos de problemas similares al anterior. En nuestro caso, un punto es un punto óptimo global en el sentido de que le corresponde el menor (problema de minimización) o el mayor (problema de maximización) valor posible de f dentro los límites fijados por el conjunto H .

Al nuevo método se le nombró "método de tunelización exponencial" y es una continuación natural de los métodos de tunelización original¹ y clásico².

Los Métodos de tunelización (V. Levy, Montalvo, Gómez, et. al.) han sido una de las principales aportaciones que México ha dado al área de la optimización y fueron desarrollados en el Instituto de Investigaciones en Matemáticas Aplicadas y en Sistemas (IIMAS) de la Universidad Nacional Autónoma de México (UNAM).

Las aportaciones de este trabajo son la introducción del método de tunelización exponencial (la función de tunelización exponencial y su algoritmo) y la actualización del algoritmo original de donde resultó el método de tunelización clásico (Gómez, et al.). De la introducción de la nueva transformación y de los distintos factores que se usan en las funciones de tunelización surge de manera natural una generalización que se denominó metodología de tunelización (Gómez y Barrón, ver el artículo [Gom91b]).

Para resolver los problemas de optimización global hay pocos programas y paquetes de software en todo el mundo. La gran mayoría de los paquetes de software son únicamente para resolver problemas de optimización en forma local, es decir, no necesariamente encuentran la solución global. Por ejemplo, el paquete GRG2 (ver el manual [Las80]) menciona que para resolver en forma global a un problema de optimización, se realicen, con un número grande de distintos puntos iniciales, varias minimizaciones y de las puntos de las soluciones se elija, como una posible aproximación a un punto mínimo global del problema, el que corresponda

¹En este trabajo se partió del algoritmo original, que se nombró "método de tunelización original" (ver los artículos [Vel85], [Mon79], [Vel80c]) y del primer código del método para el problema de optimización global sin restricciones, programa de Antonio Montalvo.

²El método original se actualizó de acuerdo a los estudios del manejo de parámetros que se reportan en los artículos: [Gom87a], [Vel83] y [Cor85] y dio origen al método que nombramos "método de tunelización clásico". Los métodos, original y clásico, usan la misma transformación funcional y a lo largo del presente trabajo se denomina función de tunelización clásica.

con el valor más pequeño de la función del problema.

Con tal panorama, se fijaron dos objetivos prácticos importantes: la elaboración de los programas de computadora de los métodos y la realización de experimentos numéricos con ellos.

El contenido del trabajo lo forman 6 capítulos cuyo contenido se detalla a continuación. En el capítulo 1 se describe un panorama resumido del área de la optimización para temas afines al método y al problema que nos ocupa. En el capítulo 2 se explican el método original y método de tunelización clásico y se dan las principales diferencias entre ellos. En el capítulo 3 se expone el nuevo método y su transformación exponencial y se introduce la metodología de tunelización. En el capítulo 4 se dan los detalles del diseño y de la programación de los métodos de tunelización.³ En el capítulo 5 se dan los resultados de experimentos computacionales de los métodos de tunelización: original, clásico y exponencial. Para esta parte experimental se diseñaron y seleccionaron experimentos numéricos de distinto tamaño y con diverso grado de complejidad. Esta parte consistió del estudio de los efectos que tienen los parámetros del nuevo método de tunelización y de un estudio comparativo de los todos métodos de tunelización, incluyendo una comparación contra los resultados de un método similar, el método de la funciones de relleno del investigador Ge Renpu⁴. Finalmente, en el capítulo 6 se dan las conclusiones del trabajo⁵.

Este trabajo se realizó en las áreas de computación y de métodos numéricos gracias a las facilidades y a los recursos del departamento de Métodos Matemáticos y Numéricos del IIMAS-UNAM.

En las investigaciones anteriores participaron casi todos los miembros del departamento: Alejandro Velasco, Antonio Montalvo, Susana Gómez, Apolinar Calderón, Alfredo Cortés, et al. (perdón por cualquier omisión). Algunas de estas personas ya no trabajan en el Instituto, pero, gracias a todos ellos por sus investigaciones.

Debo mencionar en forma especial a la Dra. Susana Gómez, quien es la especialista y quien ha llevado las investigaciones de los el métodos de tunelización, desde las primeras investigaciones hasta las actuales; las principales publicaciones son [Gom84], [Gom87a], [Vel80b], [Vel80a], [Vel84]. El último artículo brinda un panorama general de todos los trabajos de tunelización e incluye aplicaciones de optimización global⁶ para problemas con restricciones y para problemas de mínimos cuadrados. La Dra. Gómez fue la que me invitó a colaborar en la investigación de los métodos de tunelización y quien dirigió esta tesis.

Los programas de los métodos se escribieron y depuraron en las microcomputadoras del departamento de Métodos Matemáticos y Numéricos del IIMAS pero también los departamentos de Matemática y Mecánica, de Computación, de Estadística, la Secretaría Técnica y la Dirección del IIMAS-UNAM nos ayudaron al permitirnos usar sus equipos y su software⁷. Muchas gracias a Arturo Olvera, Fernando Magariños, Santiago Morales, Mónica Ardisson,

³ Junto con este documento hay otro complementario, el manual de referencia de los programas de tunelización, [Bar91]. El manual explica como utilizar los programas de tunelización, describe los requisitos computacionales para correrlos y contiene los resultados de un ejemplo completo de un problema: su codificación en FORTRAN, los archivos de datos, de parámetros y de resultados.

⁴ Ver el artículo [Ren90].

⁵ Los resultados de la investigación realizada en este trabajo se difunden, además, a nivel nacional en la publicación: [Gom91b] y en un futuro a nivel internacional en el artículo, [Gom91a].

⁶ Ver el capítulo 1.

⁷ Ver el apéndice A para una descripción del ambiente computacional que se utilizó para realizar este trabajo.

Carlos Velarde, Felipe Bracho, Hernando Mutis, Adriana Ducoing, Ricardo Weder e Ignacio Méndez.

La parte experimental que se reporta se realizó en los equipos más grandes de la UNAM que están a cargo de la Dirección General de Servicios de Cómputo Académico de la UNAM. Una parte de los experimentos numéricos se realizaron en los equipos: Burroughs A-12 y IBM 4381 y otra gran parte en la supercomputadora del tipo IBM 3090 del Centro Científico de IBM en Palo Alto, California, Estados Unidos, ya que, para este equipo, se obtuvo una beca del programa de becas del Proyecto de Supercómputo de la UNAM. Gracias a Víctor Guerra, Christine Allen y demás miembros de la Comisión de Supercómputo de la UNAM del Consejo Asesor de Cómputo.

Capítulo 1

Panorama de la optimización local y global sin restricciones

En este capítulo se describen la formulación matemática del problema de optimización local y global; las condiciones de existencia y de optimalidad de un punto óptimo; aspectos numéricos de la solución; algunos métodos numéricos y dos aplicaciones de la optimización global.

La terminología matemática se restringe al problema y a sus métodos. Dentro de la exposición se incluyen referencias a la literatura especializada y algunas demostraciones.

Algunos de los libros de optimización que describen un panorama completo de la optimización son [McC83], [Gil81] y [Fle87].

Dentro del área de optimización global, no hay muchas publicaciones, por ser un tema muy especializado. Recientemente se creó la revista *Journal of Global Optimization* del editor R. Horst, publicada por *Kluwer Academic Press*.

Un excelente artículo, que da una visión completa del estado del arte del área de la optimización global hasta 1986 es el artículo [Rin86]. Contiene una gran cantidad de referencias de los diversos métodos de optimización global; los clasifica en determinísticos y estocásticos, y bajo esta división, los describe destacando sus principales cualidades.

1.1 El problema de optimización global sin restricciones

El problema que se desea resolver en este trabajo es

$$\min_{z \in H} f(x) \quad (1.1)$$

donde $f: \mathbb{R}^n \mapsto \mathbb{R}$ es una función de clase¹ C^2 y H es un hipercubo² de \mathbb{R}^n .

El problema 1.1 es un caso particular del problema de optimización sin restricciones sobre un conjunto

$$\min_{z \in A} f(x) \quad (1.2)$$

donde A es un conjunto de \mathbb{R}^n y $f: \mathbb{R}^n \mapsto \mathbb{R}$ es una función.

¹Ver la definición 1.6.

²Ver la definición 1.4.

A un problema como el anterior se le nombra *problema de optimización global sin restricciones sobre un conjunto*. Estrictamente éste es un problema restringido pero cuando no haya lugar a confusiones con el problema de optimización global sin restricciones,

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

nos referiremos a él como el *problema de optimización global sin restricciones* y partir del capítulo siguiente nos referiremos al problema 1.1 como el problema de optimización global sin restricciones.

Definición 1.1 Dado un problema de optimización global sin restricciones sobre un conjunto nos referimos a la función f como la *función objetivo*, al conjunto A en donde se busca una solución como la *región factible* y a cualquier punto de A como un *punto factible*. ■

Note que para el problema 1.3 cualquier punto es un punto factible y en cambio para los problemas 1.1 y 1.2 esto no se cumple.

Definición 1.2 Una *solución global* de un problema 1.2 es el par $(x^*, f(x^*))$, donde x^* es un punto factible que cumple que

$$\forall x \in A \quad f(x^*) \leq f(x).$$

Al punto x^* se le llama *punto mínimo global de f en A* o *punto mínimo global (PMG)* y al valor $f(x^*)$ se le denomina *mínimo global de f en A* o *mínimo global del problema (MG)*. ■

Es fácil ver que una solución de alguno de los problemas 1.1 o 1.2 puede no serlo del problema 1.3.

En las siguientes secciones se dan para los tres problemas las condiciones para la existencia de soluciones y las propiedades que las caracterizan.

1.1.1 Condiciones de existencia del punto mínimo global

La existencia de un PMG se basa en que la función del problema sea semicontinua inferiormente³ y en que la región factible sea un conjunto compacto de \mathbb{R}^n . Un conjunto de \mathbb{R}^n que es cerrado y acotado es un ejemplo de un conjunto compacto. La semicontinuidad es lo mínimo que se pide para tener un PMG. Para los fines de este trabajo la teoría desarrollada es para funciones que tienen más que la propiedad de semicontinuidad, ya que, se supondrá que son funciones continuas y en algunos casos, además, de clase C^2 (ver la definición 1.6).

La definición de función continua se da a continuación y después se enuncia el teorema de las condiciones de existencia de una solución global.

Definición 1.3 (Función continua) Una función $f: \mathbb{R}^n \rightarrow \mathbb{R}$ es una *función continua en \mathbb{R}^n* o *función continua* si $\forall x, y \in \mathbb{R}^n$ y $\forall \epsilon > 0$, $\exists \delta > 0$ tal que si $\|x - y\| < \delta$ entonces $|f(x) - f(y)| < \epsilon$, donde $|\cdot|$ es la función valor absoluto y $\|\cdot\|$ es la función norma euclidiana⁴. ■

³Ver la página 49 del libro [Roy68].

⁴La función norma euclidiana es

$$\|x\| = \sqrt{x^T x} = \sqrt{x_1^2 + \dots + x_n^2}$$

donde $x \in \mathbb{R}^n$, $x = (x_1, x_2, \dots, x_n)^T$, x se considera una matriz de orden $n \times 1$ y $(\cdot)^T$ es la operación de transposición.

Teorema 1.1 (Condición de existencia) ⁵ Sea $f: \mathbb{R}^n \mapsto \mathbb{R}$ una función continua y $A \subset \mathbb{R}^n$ un conjunto compacto. Entonces f alcanza su mínimo, o sea, $\exists x^*$ tal que

$$f(x^*) = \min_{x \in A} f(x).$$

Antes de aplicar el resultado anterior al problema 1.1 se define claramente al conjunto H .

Definición 1.4 (Hipercubo) Un conjunto $H \subset \mathbb{R}^n$ es un *hipercubo de \mathbb{R}^n* , si es el producto cartesiano de n intervalos cerrados con más de un punto, o sea,

$$H = [a_1, b_1] \times \cdots \times [a_n, b_n] \quad a_i < b_i, \quad i = 1, \dots, n.$$

Se tiene el siguiente resultado para el problema 1.1.

Corolario 1.1 *Un problema 1.1 siempre tiene solución.*

Demostración. La afirmación se tiene por el teorema anterior. f es continua por ser de clase C^2 y H es un conjunto compacto por construcción pues es un conjunto cerrado y acotado.

Note que para el problema 1.3 el teorema de la condición de existencia no es aplicable (\mathbb{R}^n no es compacto).

1.1.2 Condiciones de optimalidad

Antes de enunciar las condiciones que se cumplen en un punto mínimo se introducen las definiciones básicas para enunciarlas.

Definición 1.5 (Derivadas Parciales) Dada una función $f: \mathbb{R}^n \mapsto \mathbb{R}$, la *derivada parcial de la función f respecto a la i -ésima componente en $x \in \mathbb{R}^n$* es

$$\lim_{h \rightarrow 0} \frac{f(x_1, \dots, x_{i-1}, x_i + h, x_{i+1}, \dots, x_n) - f(x_1, \dots, x_{i-1}, x_i, x_{i+1}, \dots, x_n)}{h}.$$

Al límite anterior, cuando existe, se le denota por

$$\frac{\partial f(x)}{\partial x_i}.$$

Cuando una función tiene derivadas parciales en \mathbb{R}^n con respecto a todas sus componentes se define la función *gradiente de f* como la función vectorial, $\nabla f: \mathbb{R}^n \mapsto \mathbb{R}^n$,

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

⁵Ver la página 101 del libro [Apo78].

Si además las funciones $\frac{\partial f}{\partial x_i}: \mathbb{R}^n \rightarrow \mathbb{R}$, $i = 1, \dots, n$ tienen todas sus derivadas parciales en \mathbb{R}^n entonces se define el *Hessiano de f* como la función matricial, $\nabla^2 f: \mathbb{R}^n \mapsto \mathbb{R}^n \times \mathbb{R}^n$,

$$\nabla^2 f(x) = \begin{pmatrix} \frac{\partial^2 f(x)}{\partial x_1 x_1} & \frac{\partial^2 f(x)}{\partial x_1 x_2} & \dots & \frac{\partial^2 f(x)}{\partial x_1 x_n} \\ \frac{\partial^2 f(x)}{\partial x_1 x_2} & \frac{\partial^2 f(x)}{\partial x_2 x_2} & \dots & \frac{\partial^2 f(x)}{\partial x_2 x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f(x)}{\partial x_1 x_n} & \frac{\partial^2 f(x)}{\partial x_2 x_n} & \dots & \frac{\partial^2 f(x)}{\partial x_n x_n} \end{pmatrix}.$$

El Hessiano de f en x es una matriz que se nombra *matriz Hessiana*.

Definición 1.6 (Clase) Las funciones de *clase C^0* son todas las funciones continuas⁶. Las funciones de *clase C^1* son aquellas funciones de clase C^0 y cuyas primeras derivadas parciales son también funciones de C^0 . En general, las funciones de *clase C^n* son todas las funciones de clase C^0 , cuyas primeras derivadas parciales son funciones de clase C^{n-1} .

Definición 1.7 (Dirección factible en x respecto A) Sean A un conjunto de \mathbb{R}^n y $x \in A$ un punto. Un vector $d \in \mathbb{R}^n$ es una *dirección factible en x con respecto a A* si $\exists \beta_0 > 0$, tal que

$$(x + \beta d) \in A \quad \forall \beta, 0 \leq \beta \leq \beta_0.$$

Definición 1.8 (Bola abierta) Sean $x \in \mathbb{R}^n$ un punto y $\epsilon > 0$ un número real, se define la *bola abierta con centro en x y radio ϵ* de la siguiente forma

$$B_\epsilon(x) = \{y \in \mathbb{R}^n \mid \|y - x\| < \epsilon\}.$$

Definición 1.9 (Punto interior) Sean A un conjunto de \mathbb{R}^n y x un punto de A . x es un *punto interior de A* si $\exists \epsilon > 0$ tal que

$$B_\epsilon(x) \subset A.$$

Definición 1.10 (Conjunto convexo) Sea A un conjunto de \mathbb{R}^n . A es un *conjunto convexo* si $\forall x, y \in A$ y $\forall \lambda, 0 \leq \lambda \leq 1$, se tiene que

$$(\lambda x + (1 - \lambda)y) \in A.$$

Definición 1.11 (Función convexa) Una función $f: \mathbb{R}^n \mapsto \mathbb{R}$ es una *función convexa en $A \subset \mathbb{R}^n$* , conjunto convexo, si $\forall x, y \in A$ y $\forall \theta \in [0, 1]$ se tiene que

$$f(\theta x + (1 - \theta)y) \leq \theta f(x) + (1 - \theta)f(y).$$

Una función es una *función estrictamente convexa en A* si la desigualdad anterior se cumple estrictamente para todo $x \neq y \in A$ y $\theta \in (0, 1)$.

⁶Ver la definición 1.3.

Definición 1.12 (Punto mínimo) Sean $f: \mathbb{R}^n \mapsto \mathbb{R}$ una función y $x^* \in \mathbb{R}^n$ un punto. x^* es un *punto mínimo local* o *punto mínimo de f* si $\exists \epsilon > 0$ tal que

$$f(x^*) \leq f(x), \quad \forall x \in B_\epsilon(x^*).$$

Al valor $f(x^*)$ se le llama *mínimo local* o *mínimo*.

Si la desigualdad se cumple estrictamente, $f(x^*) < f(x)$, entonces x^* es un *punto mínimo estricto*.

Dados un conjunto $A \subset \mathbb{R}^n$ y x^* un punto de A . x^* es un *punto mínimo local de f en A* si $\exists \epsilon > 0$ tal que

$$f(x^*) \leq f(x) \quad \forall x \in B_\epsilon(x^*) \cap A.$$

Los teoremas de las condiciones de optimalidad son muy conocidos, para este trabajo se tomaron de los libros [Her85] y [Lue73].

Teorema 1.2 (Condiciones necesarias, caso sin restricciones) Sea $f: \mathbb{R}^n \mapsto \mathbb{R}$ una función de clase C^2 , si $x^* \in \mathbb{R}^n$ es un punto mínimo local de f entonces se tienen las siguientes condiciones:

- $\nabla f(x^*) = 0$ (condición de primer orden).
- $p^T \nabla^2 f(x^*) p \geq 0, \quad \forall p \in \mathbb{R}^n$ (condición de segundo orden).

Demostración. Dado $p \in \mathbb{R}^n, \|p\| = 1$, arbitrario se define la función

$$h(\alpha) = f(x^* + \alpha p) \quad \alpha \in [-1, 1].$$

Claramente h tiene un mínimo en 0. Se tiene entonces que $\frac{dh(0)}{d\alpha} = 0$, de aquí, por la regla de derivación de la cadena, se tiene que $0 = \nabla f(x^*)^T p$ de donde finalmente, $\nabla f(x^*) = 0$ o bien es perpendicular al vector p . Se afirma que $\nabla f(x^*) = 0$ ya que el resultado también vale para un vector p' perpendicular a p .

El segundo punto se obtiene de un desarrollo de Taylor exacto de segundo orden,

$$f(x^* + \alpha p) = f(x^*) + \alpha^2 p^T \nabla^2 f(z_\alpha) p$$

donde z_α es un punto de la recta $[x^*, x^* + \alpha p]$. De donde usando que x^* es un mínimo local se obtiene,

$$0 \leq f(x^* + \alpha p) - f(x^*) = \alpha^2 p^T \nabla^2 f(z_\alpha) p.$$

Dividiendo entre α^2 y pasando al límite, $\alpha \rightarrow 0$, se tiene que $z_\alpha \rightarrow x^*$ y $\nabla^2 f$ continua (f es de clase C^2), se obtiene

$$0 \leq p^T \nabla^2 f(x^*) p.$$

Las condiciones necesarias las cumplen todos los puntos mínimos, ya sean, locales o globales. De la condición de primer orden surge en forma muy natural el concepto de punto estacionario.

Definición 1.13 (Punto estacionario) Sean $f: \mathbb{R}^n \mapsto \mathbb{R}$ una función de clase C^1 y $x \in \mathbb{R}^n$ un punto. x es un punto estacionario de f si

$$\nabla f(x^*) = 0.$$

El concepto de punto estacionario es usado en los métodos numéricos; la gran mayoría de los métodos se aproximan a una solución por medio de puntos estacionarios.

Teorema 1.3 (Condiciones suficientes) Sea $f: \mathbb{R}^n \mapsto \mathbb{R}$ una función de clase C^2 . Si x^* es un punto de \mathbb{R}^n tal que,

- $\nabla f(x^*) = 0$;
- $p^T \nabla^2 f(x^*) p > 0 \forall p \in \mathbb{R}^n$,

entonces x^* es un punto mínimo (local) estricto de f .

Demostración. Suponga que x^* no es un punto mínimo estricto, esto significa que $\forall \epsilon > 0$ se tiene que

$$\exists x \in B_\epsilon(x^*) \text{ tal que } f(x) \leq f(x^*).$$

Tomando un x que que cumpla con la suposición anterior y de un desarrollo de Taylor en $x^* + \alpha p$, donde $p = x - x^*$, se obtiene una contradicción a la segunda hipótesis, ya que se tiene

$$0 \geq f(x^* + \alpha p) - f(x^*) = \alpha^2 p^T \nabla^2 f(z_\alpha) p$$

donde z_α es un punto de la recta $[x^*, x^* + \alpha p]$. Dividiendo entre α^2 y pasando al límite, $\alpha \rightarrow 0$, se tiene que $z_\alpha \rightarrow x^*$ y $\nabla^2 f$ continua (f es de clase C^2), se obtiene

$$0 \geq p^T \nabla^2 f(x^*) p.$$

Este resultado es importante, ya que, en la práctica dado un punto estacionario de f se prueba si éste satisface las condiciones suficientes del teorema anterior para determinar si x^* es un punto mínimo de f .

Las condiciones de optimalidad para el problema sin restricciones no son las que se aplican para los problemas 1.2 y 1.1. En los resultados siguientes se utiliza el problema 1.2 y después se dan resultados para el caso particular en que $A = H$.

En un problema 1.2 la región factible es una parte de \mathbb{R}^n y un mínimo local de un problema sin restricciones puede no ser factible para éste, por otro lado, un punto mínimo local de un problema 1.2 puede no serlo del problema sin restricciones.

En nuestro caso basta con usar las definiciones de dirección factible en x respecto A y de conjunto convexo como conceptos básicos para establecer las condiciones necesarias de optimalidad del problema 1.2.

Teorema 1.4 (Condiciones necesarias, caso restringido, $A \subset \mathbb{R}^n$) Dado un problema 1.2 con f de clase C^2 y $A \subset \mathbb{R}^n$ un conjunto convexo. Si $x^* \in A$ es un punto mínimo de f en A se tienen entonces las condiciones:

- $d^T \nabla f(x^*) \geq 0, \forall d$ dirección factible en x^* respecto a A (condición de primer orden).
- Si además se supone que $d^T \nabla f(x^*) = 0 \forall d$ dirección factible en x^* respecto a A entonces $d^T \nabla^2 f(x^*) d \geq 0$ (condición de segundo orden).

Demostración. De la definición de punto mínimo local de f respecto a A se tiene, $\exists \epsilon > 0$ tal que $f(x^*) \leq f(x) \forall x \in B_\epsilon(x^*) \cap A$. Es fácil ver que $B_\epsilon(x^*)$ es un conjunto convexo y que la intersección de ambos (que nombraremos C_ϵ) es un conjunto convexo. Si la intersección es sólo el punto x^* el resultado se tiene ya que $d = 0$ es la única dirección factible. Si la intersección es no vacía entonces tomando una dirección factible d en x^* respecto a A se tiene que $\exists x \in C_\epsilon$ tal que $x = x^* + \beta d$ y $f(x) \geq f(x^*)$ para algún $\beta \in (0, \epsilon)$. Note que si no existiera entonces d no sería factible en x^* respecto a A y si no se se cumpliera que $f(x) \geq f(x^*)$ entonces x^* no sería un punto mínimo de f respecto a A .

Ahora por el teorema del valor medio se tiene

$$f(x) - f(x^*) = \beta d^T \nabla f(z_\beta)$$

donde z_β es un punto de la recta que une x^* y x . Usando que $f(x) - f(x^*) \geq 0$ y dividiendo por $\beta > 0$ se llega a la desigualdad

$$d^T \nabla f(z_\beta) \geq 0.$$

Tomando límite cuando $\beta \rightarrow 0$ se tiene la condición de primer orden.

De manera análoga se obtiene la condición de segundo orden, pero en este caso se utiliza un desarrollo de Taylor hasta el segundo término. Sea d una dirección factible en x^* respecto a A , de la hipótesis se tiene $d^T \nabla f(x^*) = 0$. El desarrollo de Taylor que resulta en este caso es

$$f(x) = f(x^*) + \beta^2 d^T \nabla^2 f(z_\beta) d$$

donde z_β es algún punto de la recta $[x^*, x]$. Usando que $f(x) - f(x^*) \geq 0$ y dividiendo entre β^2 se obtiene la desigualdad

$$d^T \nabla^2 f(z_\beta) d \geq 0.$$

La condición de segundo orden se obtiene pasando al límite $\beta \rightarrow 0$. ■

Corolario 1.2 Dadas las hipótesis del teorema anterior. Si x^* es un punto interior de A entonces se tienen las condiciones del teorema 1.2.

Demostración. El resultado es inmediato, ya que si x^* es un punto interior de A entonces cualquier $d \in \mathbb{R}^n$ es una dirección factible en x^* respecto a A . ■

Para los problemas 1.1 y 1.2 el concepto de punto estacionario de la definición 1.13 no es suficiente, ya que no cubre los mismos casos que la condición de primer orden del caso restringido. Para salvar esta diferencia, a lo largo de este trabajo se usará la siguiente definición de punto estacionario dentro de un conjunto.

Definición 1.14 (Punto estacionario de f en $A \subset \mathbb{R}^n$) Sean $f: \mathbb{R}^n \rightarrow \mathbb{R}$ una función de clase C^1 , $A \subset \mathbb{R}^n$ un conjunto no vacío y x un punto factible. x es un punto estacionario de f en A si se cumple la condición de primer orden del teorema anterior, es decir, que

$$d^T \nabla f(x^*) \geq 0 \quad \forall d \text{ dirección factible en } x^*.$$

■

Corolario 1.3 Sean $f: \mathbb{R}^n \rightarrow \mathbb{R}$ una función de clase C^1 , $A \subset \mathbb{R}^n$ un conjunto y x^* un punto interior de A . Si x^* es un punto estacionario de la función f en A entonces x^* es un punto estacionario de f .

Demostración. El resultado se tiene ya que como x^* es un punto interior de A , cualquier dirección $d \in \mathbb{R}^n$ es factible. Sea $d \in \mathbb{R}^n$, $d \neq 0$, y tomando $-d$ se deduce que $\nabla f(x) = 0$, ya que, $d^T \nabla f(x) \geq 0$ y $-d^T \nabla f(x) \geq 0$. ■

De lo anterior es claro que no todo punto mínimo de f en A es un punto mínimo de f . Para que sea un punto mínimo de f , por el corolario anterior se necesita que x^* sea un punto interior de A y un punto mínimo de f en A .

Note que los puntos en donde se anula el gradiente son puntos estacionarios pero pueden no corresponder a una solución mínima; pueden ser puntos máximos (donde el valor de la función alcanza un máximo) o bien puntos silla (donde los valores de la función crecen en una dirección pero decrecen en otra).

De lo expuesto en esta sección, para el problema 1.1 se tienen las siguientes propiedades:

- H es un conjunto compacto y convexo (ver la página 5 y la definición 1.10.);
- H conjunto compacto y dada una función f de clase C^0 por el teorema 1.1 se tiene que el problema 1.1 tiene solución.
- H conjunto compacto y dada una función f convexa en H entonces el problema 1.1 tiene un mínimo global⁷;
- con las hipótesis del párrafo anterior, pero suponiendo que f es una función estrictamente convexa entonces el problema 1.1 tiene una solución única, $(x^*, f(x^*))$ ⁸;
- los puntos mínimos del problema 1.1 son puntos estacionarios de f en H .

1.2 Aspectos de la solución de los problemas de optimización global

Una solución del problema 1.1 es un par $(x^*, f(x^*))$ (ver la página 5) que en general no es fácil de calcular ya sea numéricamente o, bien, analíticamente. La mayoría de los métodos numéricos calculan sólo una aproximación a puntos estacionarios que muy probablemente son puntos mínimos.

En esta sección se estudia el efecto de la forma de la función sobre una aproximación numérica de una solución⁹. Este estudio es con la finalidad de resaltar que no necesariamente se puede garantizar que el método termina en una buena solución para cualquier función. Se incluyen en esta sección dos subsecciones en donde se describen funciones, muy sencillas de formular, con las que se pueden plantear problemas de minimización muy difíciles de resolver numéricamente.

⁷Ver la página 119 del libro [Lue73].

⁸Idem.

⁹Ver el artículo [Dix76].

Casi todos los métodos numéricos solucionan el problema de optimización sin restricciones en forma iterativa con la siguiente propiedad:

$$\lim_{i \rightarrow \infty} x^i = x^*$$

donde $x^0 \in \mathbb{R}^n$ es un punto inicial dado, la sucesión de puntos x^1, x^2, \dots se genera por el método y x^* es un punto mínimo del problema.

Y cada método utiliza una condición de paro, la cual sirve para detener el método, es decir, para que deje de generar puntos de la sucesión, ya que, en principio la sucesión que genera el método es infinita. La condición de paro de un método iterativo no garantiza que el punto en donde se detiene el método es una buena aproximación a un punto mínimo¹⁰. La aproximación calculada es en el mejor de los casos una aproximación a un punto estacionario, razón por la cual es necesario verificar que se trata de un punto mínimo.

El problema de detectar cuando se termina en una aproximación a un punto mínimo es un tema que se desarrolla junto con el método iterativo. El método se construye usando propiedades que, en principio, aseguran la terminación en un punto cercano a un punto mínimo, pero esto no es siempre posible garantizarlo para cualquier función, por ejemplo, si se supone la función del problema 1.1 es convexa es muy probable que cualquier método termine en una buena aproximación a una solución.

Generalmente se le deja al usuario elegir la tolerancia con que aceptará una aproximación a una solución. Las tolerancias para aceptar una solución son

- $0 < \epsilon_X \ll 1$ es la tolerancia en la aproximación a un punto mínimo,
- $0 < \epsilon_G \ll 1$ es la tolerancia en la aproximación a un punto sea estacionario y
- $0 < \epsilon_F \ll 1$ es la tolerancia en la aproximación a un mínimo.

Suponiendo que se tiene una solución del problema 1.1, se definen los siguientes conjuntos:

$$\begin{aligned} X &= \{x \in \mathbb{R}^n \mid \|x - x^*\| \leq \epsilon_X\} \\ G &= \{x \in \mathbb{R}^n \mid \|P_H(x, \nabla f(x))\| \leq \epsilon_G\} \\ F &= \{x \in \mathbb{R}^n \mid |f(x) - f(x^*)| \leq \epsilon_F\} \end{aligned}$$

Cada uno de los conjuntos anteriores definen conjuntos de puntos de distinto tamaño y forma en \mathbb{R}^n .

Cualquiera que sea el método las condiciones de paro funcionan de la siguiente forma, dado x^0 un punto inicial, el método genera una sucesión $\{x^j\}$ que converge (al menos en teoría) a x^* . Una buena aproximación a la solución es el par $(x^k, f(x^k))$ donde k es el primer entero tal que x^k esté en alguno de los conjuntos X o G , o bien, $f(x^k)$ esté en el conjunto F .

Note que el cálculo de una aproximación a una solución depende: del sistema aritmético del equipo de cómputo¹¹, del método numérico¹² y de las características de la función.

¹⁰No se dan los detalles de la convergencia de los métodos, que puede ser finita o infinita, con velocidad de convergencia lineal, superlineal, etcétera, para mayor información ver el libro [Dah74].

¹¹Ver la sección 4.2.

¹²Ver la sección 4.5.

El punto por analizar es como la forma de la función afecta el cálculo a una aproximación a una solución y para el análisis que damos se usan los conjuntos X y F ; el análisis que se propone es independiente del método para resolver el problema de optimización¹³ y del sistema aritmético.

No es difícil visualizar que los casos extremos son las funciones con cambios de valor muy pequeños respecto a cambios grandes en el argumento, o bien al contrario (la construcción de funciones con estas características se da en las secciones 1.2.1 y 1.2.2).

La forma de la función se caracteriza matemáticamente a continuación.

Definición 1.15 Sean $f: \mathbb{R}^n \mapsto \mathbb{R}$ una función continua y x^* un punto mínimo de f . f tiene largo $m > 0$ y ancho ϵ en x^* si $\forall x \in \mathbb{R}^n$, $\|x - x^*\| = \epsilon$ se tiene que

$$|f(x) - f(x^*)| \geq m.$$

Definición 1.16 (Apéndice) Sean $f: \mathbb{R}^n \mapsto \mathbb{R}$ una función continua y x^* un punto mínimo de f . f tiene un apéndice en x^* de ancho δ , $\delta \ll \frac{1}{2}$, si en tal punto f es de largo mayor a 1 para un ancho menor o igual a δ .

Definición 1.17 (Plana) Sean $f: \mathbb{R}^n \mapsto \mathbb{R}$ una función continua y x^* un punto mínimo de f . f es casi plana en x^* de largo ϵ , $\epsilon \ll \frac{1}{2}$, si en tal punto f es de largo menor ϵ para un ancho mayor o igual a 1.

La caracterización anterior de la forma de las funciones conduce inmediatamente a los siguientes resultados respecto a la aproximación que un método numérico puede dar como resultado.

Teorema 1.5 Sean $f: \mathbb{R}^n \mapsto \mathbb{R}^n$ una función de clase C^0 y x^* un punto mínimo de f . Si f tiene un apéndice en x^* de largo m y ancho δ entonces existe una aproximación, x^a ,

$$\|x^a - x^*\| \ll \frac{1}{2} \text{ y } |f(x^a) - f(x^*)| \geq 1.$$

Demostración. El resultado se sigue de las definiciones anteriores.

Teorema 1.6 Sean $f: \mathbb{R}^n \mapsto \mathbb{R}^n$ una función de clase C^0 y x^* un punto mínimo de f . Si f es casi plana en x^* de orden ϵ entonces existe una aproximación, x^a , tal que

$$|f(x^a) - f(x^*)| \ll \frac{1}{2} \text{ y } \|x^a - x^*\| \geq 1.$$

Demostración. El resultado se sigue de las definiciones anteriores.

Los dos teoremas anteriores muestran que independientemente del método numérico y del sistema aritmético cuando la función tiene partes “casi planas” o partes “puntiagudas” es muy posible que un método termine en malas aproximaciones a una solución.

Una aplicación de lo anterior es la construcción de problemas difíciles de resolver numéricamente. En las subsecciones siguientes se describen funciones que pueden servir para formar funciones con un apéndice o de forma casi plana. Un ejemplo de una función que las usa es el experimento 5.13 de este trabajo.

¹³Para este trabajo los métodos locales que se usaron son del tipo métrica variable, gradiente conjugado y direcciones conjugadas, ver las secciones 1.3.1 y 1.3.2. La rutina CONMIN se basa en los dos primeros métodos y la rutina MINI se basa en el algoritmo de Fletcher-Reeves, ver la sección 4.5.

1.2.1 Funciones apéndice

En esta sección se describen dos formas de construir funciones con partes “puntiagudas” en el sentido de la definición 1.16.

Definición 1.18 Una función apéndice aditiva, $A_+ : \mathbb{R}^n \mapsto \mathbb{R}$, está dada por

$$A_+(x) = -\mu \cdot \exp(-\alpha \|x - x^*\|^2) \quad (1.4)$$

donde x^* es la posición del apéndice, $\exp(\cdot)$ es la función exponencial, $\mu > 0$ es el largo y $\alpha \leq 1$ es el ancho del apéndice. ■

Definición 1.19 Una función apéndice multiplicativa, $A_* : \mathbb{R}^n \mapsto \mathbb{R}$, está dada por

$$A_*(x) = \mu \cdot \exp(-\alpha \|x - x^*\|^2) + 1 \quad (1.5)$$

donde x^* es la posición del apéndice, $\exp(\cdot)$ es la función exponencial, $\mu > 0$ es el largo y $\alpha \leq 1$ es el ancho del apéndice. ■

Al sumar una función 1.4 a otra se introduce un nuevo mínimo para μ suficientemente grande, por otro lado, el efecto de multiplicar una función por la función 1.5 es el de amplificar un mínimo ya existente (si no se aplica en un punto mínimo no se forma un apéndice).

1.2.2 Función aplana

En forma similar, en esta sección, se describe una forma de construir funciones con partes “casi planas” en el sentido de la definición 1.17.

Definición 1.20 Una función que aplana, $A_p : \mathbb{R}^n \mapsto \mathbb{R}$, está dada por

$$A_p(x) = \mu \cdot \exp(-\alpha \|x - x^*\|^2) + 1 \quad (1.6)$$

donde x^* es la posición del centro del aplanado, $\exp(\cdot)$ es la función exponencial, $0 < \mu \leq \frac{1}{2}$ es el largo y $\alpha > 1$ es el ancho. ■

El efecto de multiplicar una función por una función 1.6 es la reducción de los valores de la función alrededor de x^* .

1.3 Métodos locales para resolver problemas de optimización sin restricciones

Para resolver los problemas de optimización sin restricciones existen gran variedad de métodos. Generalmente, las soluciones de los métodos resultan de aproximaciones a puntos estacionarios (definición 1.14), es decir, puntos que están en G (ver la sección 1.2).

Para tales puntos es necesario verificar las condiciones suficientes de optimalidad dadas en el teorema 1.3, sin embargo, puede no ser suficiente, por ejemplo, la función x^4 tiene en $x^* = 0$ un PMG pero no cumple lo expuesto en el teorema 1.3.

Una forma de solucionar el problema 1.1 es calculando todos los puntos estacionarios de la función f , una vez que se tienen, se comprueba cuales son puntos mínimos, se ordenan de acuerdo a sus valores y se determina un PMG. Huelga decir que, esta forma de resolver el problema no es práctica en general.

La forma práctica de resolver el problema de optimización sin restricciones es usando métodos numéricos, por ejemplo¹⁴: descenso acelerado, Newton, direcciones conjugadas, métrica variable, etcétera.

Para simplificar la exposición de los métodos numéricos que se describen en la siguientes secciones se introduce la proyección sobre un hipercubo.

Definición 1.21 (Proyección sobre H) Sea $H \subset \mathbb{R}^n$ un hipercubo, $H = [a_1, b_1] \times \cdots \times [a_n, b_n]$, $a_i < b_i$, $i = 1, \dots, n$, se define la *función de proyección sobre H*, $P_H : H \times \mathbb{R}^n \mapsto \mathbb{R}^n$, $P_H(x, d) = \alpha(y_1, \dots, y_n)^T$, donde

$$y_i = \begin{cases} d_i, & \text{si } a_i < x_i < b_i; \\ 0, & \text{si } a_i = x_i \text{ o } b_i = x_i; \end{cases} \quad i = 1, \dots, n$$

y α es una constante,

$$\alpha = \max\{1, \beta_1, \dots, \beta_n\}$$

donde cada constante β_j , $j = 1, \dots, n$, se obtiene de

$$\beta_j = \begin{cases} \left| \frac{2_j - x_j}{y_j} \right|, & \text{si } y_j \neq 0 \text{ y } x_j + y_j < a_j; \\ \left| \frac{2_j - b_j}{y_j} \right|, & \text{si } y_j \neq 0 \text{ y } b_j < x_j + y_j; \\ 0, & \text{en otro caso.} \end{cases}$$

En los siguientes apartados se describen los métodos de Newton o de métrica variable (de los métodos de direcciones conjugadas) el algoritmo de Fletcher-Reeves.

1.3.1 Métodos de Newton y de métrica variable

Estos métodos buscan puntos de G (ver la sección 1.2).

Los pasos que realiza son

1. Dado $x^0 \in H$ un punto arbitrario. Se calcula $\nabla^2 f(x^0)$ (Hessiano¹⁵) o una aproximación de él por medio diferencias de finitas o de, entre muchas otras fórmulas, la fórmula BFGS¹⁶ (esta matriz la nombraremos B).
2. Se construye una dirección de búsqueda

$$d = P_H(x^i, B^{-1} \nabla f(x^i))$$

donde $P_H(\cdot, \cdot)$ es la función proyección¹⁷.

Si $d = 0$ se termina; $x^{i+1} \in G$.

¹⁴Ver los libros: [McC83], [Gil81], [Her85] y [Fle87].

¹⁵Ver la definición 1.5 de derivada parcial.

¹⁶Ver la página 193 del libro [Fle87].

¹⁷Ver la definición 1.21.

3. En otro caso, se determina $x^{i+1} = x^i + \alpha d$ de resolver el problema

$$\min_{0 < \alpha < \alpha_M} f(x^i + \alpha d)$$

donde α_M es un valor real tal que los puntos $x^{i+1} = x^i + \alpha d$ son factibles $\forall \alpha, 0 < \alpha < \alpha_M$, o α debe satisfacer un criterio de suficiente descenso¹⁸, como por ejemplo,

$$f(x^i + \alpha d) < f(x^i) + \epsilon_2 \alpha d^T \nabla f(x^i)$$

donde $\epsilon_2 > 0$ es un parámetro dado.

4. Si $\|P_H(x, \nabla f(x^{i+1}))\| < \epsilon$ se termina; $x^{i+1} \in G$.

5. Se calcula $\nabla^2 f(x^{i+1})$ o se actualiza la aproximación y se repite a partir del paso 2.

La dirección de búsqueda d que se menciona en el paso 2 se obtiene de aproximar

$$\nabla f(x^i + d) = 0$$

por la definición de derivada, de donde se obtiene

$$\nabla f(x^i + d) \approx \nabla f(x^i) + \nabla^2 f(x^i) d$$

Suponiendo que $\nabla f(x^i + d) \approx 0$, la dirección de búsqueda d se calcula resolviendo el sistema de ecuaciones

$$-\nabla f(x^i) = B d$$

donde B es $\nabla^2 f(x^i)$ o una aproximación a él.

Para evitar resolver el sistema anterior se usan las formulas BFGS o DFP¹⁹ que aproximan directamente a la inversa del Hessiano (definición 1.5). El uso de las fórmulas mencionadas reduce a un producto matricial el cálculo de la dirección.

1.3.2 Métodos de direcciones conjugadas

Los algoritmos clásicos de direcciones conjugadas son gradiente conjugado y el algoritmo de Fletcher-Reeves²⁰.

Los métodos de direcciones conjugadas se basan en la Q -ortogonalidad, que se describe a continuación.

Definición 1.22 Sea Q una matriz simétrica

- los vectores $d_1, d_2 \in \mathbb{R}^n$ son Q -ortogonales o conjugados respecto a Q si $d_1^T Q d_2 = 0$.
- los vectores $d_1, d_2, \dots, d_k \in \mathbb{R}^n$ son Q -ortogonales si $d_i^T Q d_j = 0$ para todo $i \neq j$.

donde d_k son vectores de \mathbb{R}^n y $d^T Q d$ es un producto matricial.

Si Q es la matriz identidad entonces los vectores de \mathbb{R}^n que cumplen las propiedades anteriores se denominan ortogonales.

¹⁸ Ver el artículo [Sha78].

¹⁹ Ver el libro [Fle87].

²⁰ Ver los libros [Her85] y [McC83].

El algoritmo de Fletcher-Reeves²¹ utiliza la matriz identidad y los pasos que tiene son

1. Dados la función $f: \mathbb{R}^n \mapsto \mathbb{R}$ de clase C^1 , el punto arbitrario $x^0 \in \mathbb{R}^n$ y la tolerancia ϵ^2 , $1 \gg \epsilon^2 > 0$, en la solución.

Se calcula

$$g = P_H(x^0, \nabla f(x^0))$$

donde $P_H(\cdot, \cdot)$ es la función proyección²² y $g_g = g_0^T g_0$.

Si $g_g \leq \epsilon$ se para; $x^0 \in G$.

2. Se hace $k = 0$, $d_0 = g$ y $g_0 = g_g$.
3. Se resuelve el problema de optimización unidimensional,

$$\min_{\lambda \in \mathbb{R}, 0 < \lambda < \lambda_M} f(x^k + \lambda d_k)$$

donde λ_M es tal que los puntos $(x^k + \lambda d_k)$ son factibles $\forall \lambda, 0 < \lambda \leq \lambda_M$.

Se hace $x^{k+1} = x^k + \lambda^* d_k$, donde λ^* es el punto mínimo del problema anterior.

Se pone $k = k + 1$ y se calcula $g = P_H(x^k, \nabla f(x^k))$ y $g_g = g^T g$.

Si $g_g \leq \epsilon^2$ se termina; $x^k \in G$.

En otro caso, si $k = n$, se hace $x^0 = x^k$ y se continúa en el paso 2.

Se calcula $d_k = -g + t d_{k-1}$ donde $t = g_g / g_0$.

Se hace $g_0 = g_g$ y se repite este paso.

Este algoritmo termina en un punto de G y cuando la función objetivo es cuadrática el algoritmo termina a lo más en n iteraciones.

1.4 Métodos para el problema de optimización global sin restricciones

La clasificación de los métodos de optimización global que dan Rinnoy Kan y Timmer (ver el artículo [Rin86]) los divide en determinísticos y estocásticos.

Bajo esta división, los últimos se caracterizan por usar alguna función de probabilidad, que usan para ir aproximándose al punto mínimo global, y en algunos casos les auxilia para probar algún criterio de probabilidad que garantice la optimalidad global de la solución.

Los métodos determinísticos, no utilizan conceptos de probabilidad, se aproximan a la solución del problema aprovechando las características de la función objetivo y de la región factible.

Sin profundizar mucho en los distintos métodos se describen dos: uno estocástico y otro determinístico.

²¹Se programó usando el algoritmo de Fletcher-Reeves la rutina de minimización MINI para los programas de tunelización de este trabajo. Ver el capítulo 4.

²²Ver la definición 1.21.

1.4.1 Método de Solís y Wets

El método estocástico que se describe fue desarrollado por Francisco Solís y Roger J.B. Wets²³; lo incluimos para difundir el trabajo del investigador mexicano Francisco Solís²⁴.

Su método es para el problema

dada una función $f: \mathbb{R}^n \rightarrow \mathbb{R}$ y S conjunto de \mathbb{R}^n , se busca un punto $x^* \in S$ que minimiza a f en S o bien es una aproximación aceptable al ínfimo de f en S .

Este problema es más general que el problema 1.1, ya que la función f y el conjunto S son más arbitrarios (el único requisito del conjunto S es que sea medible y de medida positiva).

El método consta de dos partes, una de optimización global y otra de optimización local. Para la optimización global, los pasos del llamado algoritmo conceptual son

1. Se hace $k = 0$ y se busca x^0 en S .
2. Se genera ξ^k del espacio muestral $(\mathbb{R}^n, \mathcal{B}, \mu_k)$.
3. Se hace $x^{k+1} = D(x^k, \xi^k)$, se selecciona μ_{k+1} , se calcula $k = k + 1$ y se repite desde el paso 2.

donde $D: S \times \mathbb{R}^n \rightarrow S$ satisface la condición

$$f(D(x, \xi)) \leq f(x) \text{ y si } \xi \in S, f(D(x, \xi)) \leq f(\xi),$$

\mathcal{B} es una familia de conjuntos de Borel de \mathbb{R}^n y μ_k son medidas de probabilidad definidas sobre \mathcal{B} .

Para la optimización local (paso 3 del algoritmo anterior) se introducen dos algoritmos de optimización local estocásticos basados en el llamado algoritmo básico.

1. Sea $k = 0$, se selecciona x^0 , se hace $\#n = 0$, $\#f = 0$, $b^0 \in \mathbb{R}^n$, $b^0 =$. Se fijan ρ_{-1} , ρ_{1b} , ex , ct , S_{ex} y f_{ct}
2. Sea

$$\rho_k = \begin{cases} \rho_{k-1} \cdot ex, & \text{si } \#s \geq S_{ex}; \\ \rho_{k-1} \cdot ct, & \text{si } \#f \geq f_{ct}; \\ \rho_{k-1}, & \text{en otro caso.} \end{cases}$$

Si $\rho_k \leq \rho_{1b}$ se termina; x^k es un punto óptimo local.

En otro caso, se calcula un vector aleatorio $\xi \in \mathbb{R}^n$ usando la distribución multivariada μ_k .

²³Ver el artículo [Sol81].

²⁴El método fue mencionado en la plática *Stochastic Minimization of Lipschitz Functions* de R. Mladineo de la universidad *Rider College* en la reunión *Recent Advances in Global Optimization* celebrada en la Universidad de Princeton (New Jersey, U.S.A.) del 10 al 11 de mayo de 1991.

3. Sea

$$x^{k+1} = \begin{cases} \xi, & \text{si } f(\xi) < f(x^k), \text{ se hace } \#s = \#s + 1, \#f = 0 \text{ y} \\ & b^{k+1} = 0.4(\xi - x^k) + 0.2b^k; \\ 2x^k - \xi, & \text{si } f(2x^k - \xi) < f(x^k) \leq f(\xi), \text{ se hace } \#s = \#s + 1, \#f = 0 \text{ y} \\ & b^{k+1} = b^k - 0.4(\xi - x^k); \\ x^k, & \text{en otro caso, se hace } \#s = 0, \#f = \#f + 1 \text{ y} \\ & b^{k+1} = 0.5b^k. \end{cases}$$

y se continúa en el paso 2.

Los parámetros que aparecen tienen la siguiente interpretación:

- k es el contador de las iteraciones;
- ρ_k es el diámetro de la región de donde se obtiene el vector aleatorio ξ ;
- ρ_{lb} es la cota inferior de ρ_k . Si ρ_k alcanza este valor, la región de donde se extrae ξ es tan pequeña que no tiene caso continuar la búsqueda para disminuir el valor de f . La única forma de que ρ_k sea menor o igual a este valor es cuando se ha tenido una serie grande de fracasos en la disminución del valor de f ;
- b^k es el factor de sesgo para inclinar el muestreo en favor de la dirección de donde se ha tenido éxito en la disminución del valor de f (los autores del algoritmo remiten al artículo [Mat65] para mayor información acerca de este parámetro);
- $\#s$ y $\#f$ son para llevar el número de éxitos y fracasos en la disminución del valor de f . Si $\#s$ alcanza el valor s_{ez} se expande el diámetro ρ_k por el factor de expansión ex . Si $\#f$ alcanza el valor f_{ct} se reduce el diámetro ρ_k por el factor de reducción ct .

Los valores para los parámetros que los autores recomiendan son: $s_{ez} = 5$, $f_{ct} = 3$, $ex = 2$, $ct = 0.5$ y $\rho_{-1} = 1$.

En teoría ρ_{lb} puede ser cero, en este caso, para las funciones adecuadas el algoritmo puede generar una secuencia de puntos que convergen a un punto mínimo local. Sin embargo, es deseable que el algoritmo se detenga cuando se esté suficientemente cerca de la solución local. En la práctica se elige un valor pequeño y positivo de ρ_{lb} de acuerdo con la exactitud deseada y la dimensión del problema.

Las dos versiones del algoritmo básico se construyen usando las distribuciones multivariadas: normal con matriz de covarianza $\rho_k I$ (I es la matriz identidad) y uniforme en el hipercubo de lado ρ_k , centradas en $(x^k + b^k)$.

Finalmente, las versiones para optimización global basadas en el algoritmo conceptual para una medida uniforme μ_k sobre $S \in \mathbb{R}^n$ (conjunto acotado de Borel) utilizan $D : S \times \mathbb{R}^n \mapsto S$, dada por

$$D(x, \xi) = \begin{cases} A(\xi), & \text{si } f(A(\xi)) < f(x); \\ x, & \text{en otro caso,} \end{cases}$$

donde $A(x)$ es el punto mínimo de f obtenido de un algoritmo A de optimización local usando ξ como punto inicial.

Los algoritmos locales que usaron para los resultados que reportan son: de direcciones conjugadas (ver la sección 1.3.2) y el básico con distribución uniforme en hipercubos.

En general, cuando en el algoritmo conceptual se usan las versiones del algoritmo básico no se sabe si el punto en que termina es un PMG. La solución local no necesariamente pertenece a G (ver la sección 1.2).

Por otro lado, cuando se usa el método de direcciones conjugadas en lugar del algoritmo básico es más probable que se termine en un PMG, ya que, en cada etapa local se termina en un punto de G y muy probablemente el algoritmo conceptual barre toda la región S .

1.4.2 Método de la cobertura

El método que describimos se reporta en el artículo [Rin86] y fue publicado originalmente en 1971. La versión que se describe se tomó del artículo [Evt85] y se conoce como método de la Cobertura (*The Method of Covering*).

Este método resuelve el problema 1.1 y los conceptos que usa se dan a continuación.

Definición 1.23 Sean $f: \mathbb{R}^n \mapsto \mathbb{R}$ una función, $H \subset \mathbb{R}^n$ un conjunto compacto y x^1, x^2, \dots, x^k una secuencia de puntos de H .

- Se llama *registro* a la cantidad:

$$R_k = \min_{1 \leq j \leq k} \{f(x^j)\}.$$

Además, los puntos de la secuencia que satisfacen $R_k = f(x^j)$ se les nombra *puntos del registro*.

- Dado $\epsilon, 1 \gg \epsilon > 0$, se define un *conjunto de cobertura* como

$$Z_k = \{x \in \mathbb{R}^n \mid R_k - \epsilon \leq f(x)\}.$$

Teorema 1.7 Dadas las condiciones de la definición anterior, se tiene que

- $\{R_1, R_2, \dots, R_k\}$ es una secuencia monótona decreciente, es decir, $R_1 \geq R_2 \geq \dots \geq R_k$.
- $\{Z_1, Z_2, \dots, Z_k\}$ es una secuencia de conjuntos creciente, es decir, $Z_1 \subset Z_2 \subset \dots \subset Z_k$.
- Para el conjunto de cobertura Z_k se tiene que

$$R_k - \epsilon \leq \min_{z \in Z_k} f(z).$$

Demostración. Las relaciones anteriores se tienen por inducción y de la definición.

La última relación significa que los puntos de Z_k no son de interés para continuar la búsqueda del mínimo de f ; no mejoran el valor del registro R_k en más de ϵ .

Luego, el área de búsqueda se puede reducir a $H \setminus Z_k$. En particular cuando

$$H \subset Z_k$$

el problema está resuelto, el valor del registro R_k y los puntos del registro son una aproximación a la solución.

Usando lo anterior, la forma de construir los conjuntos de cobertura dan origen a una gran variedad de métodos de optimización global.

El método más simple de este tipo es para funciones de Lipschitz.

Definición 1.24 (Función de Lipschitz) Sean $f: \mathbb{R}^n \rightarrow \mathbb{R}$ una función y $S \subset \mathbb{R}^n$ un conjunto no vacío. f es una función de Lipschitz en S , si $\exists \lambda > 0$ y $\forall x, y \in S$ se tiene que

$$|f(x) - f(y)| \leq \lambda \|x - y\|.$$

El método para estas funciones se basa en

- la desigualdad

$$R_k - \epsilon \leq f(x)$$

que se cumplirá para toda $x \in Z_k$, si

$$R_k - \epsilon \leq f(y) - \lambda \|x - y\|,$$

- los conjuntos

$$B_{jk} = \{x \in \mathbb{R}^n \mid \|x - x^j\| \leq r_{jk}\},$$

$$r_{jk} = \frac{f(x^j) - R_k + \epsilon}{\lambda},$$

para la secuencia x^1, x^2, \dots, x^k de puntos de S .

El problema está resuelto cuando para algún k se cubra a S , es decir,

$$S \subset \bigcup_{j=1}^k B_{jk}.$$

Un algoritmo para realizar lo anterior dado S un conjunto compacto no vacío y f una función de Lipschitz en S tiene los siguientes pasos.

1. Elegir k puntos en S y determinar R_k .
2. Calcular r_{jk} .

Si $S \subset \bigcup_{j=1}^k B_{jk}$; parar.

3. Buscar un punto x^{k+1} tal que $f(x^{k+1}) < R_k$, hacer $R_{k+1} = f(x^{k+1})$, $k = k + 1$ y continuar en 2.

El algoritmo está bien definido ya que S es un conjunto compacto (ver la página 5), el error en la determinación del mínimo es ϵ y del punto mínimo es $\frac{\epsilon}{\lambda}$ (se trata alguno de los puntos del registro, $R_k = f(x^j)$).

Para terminar esta sección se da un teorema que muestra la aplicabilidad del método de la Cobertura a funciones continuas.

Teorema 1.8 (Cobertura de una función continua) Sea $f: \mathbb{R}^n \rightarrow \mathbb{R}$ una función continua y $S \subset \mathbb{R}^n$ un conjunto compacto no vacío.

Entonces para todo $\epsilon > 0$, existe una familia finita de conjuntos que cubre S y se tiene que

$$|f(a_j) - f(x^*)| \leq \epsilon \quad \|a_j - x^*\| \leq r_j,$$

donde x^* es un PMG de f en S , $a_j \in S$ son puntos y r_j son los radios de conjuntos de la forma

$$V(a, r) = \{x \in \mathbb{R}^n \mid \|x - a\| < r, |f(x) - f(a)| \leq \frac{\epsilon}{2}\}.$$

Demostración. Primero, de las hipótesis se tiene que la función es uniformemente continua²⁵.

En efecto, dados $\frac{\epsilon}{2}$, $a \in S$ y $r > 0$, se forman los conjuntos

$$V(a, r) = \{x \in \mathbb{R}^n \mid \|x - a\| < r \text{ y } |f(x) - f(a)| \leq \frac{\epsilon}{2}\}.$$

Es inmediato que los conjuntos anteriores forman una colección de conjuntos que cubren S , o sea,

$$S \subset \bigcup_{a \in S, r > 0} V(a, r).$$

Siendo S compacto, es posible tomar un número finito de conjuntos de la colección que cubran a S , o sea,

$$S \subset \bigcup_{a_i \in S, \frac{r_i}{2} > 0, i=1, \dots, m} V(a_i, \frac{r_i}{2}).$$

Sea $\delta = \min_{i=1, \dots, m} \{\frac{r_i}{2}\}.$

Se afirma que δ satisface la definición de continuidad uniforme para ϵ y f en S .

En efecto, sean x y $p \in S$ arbitrarios tales que: $\|x - p\| < \delta$.

Como la colección finita cubre S , existe $a_i \in S$ tal que

$$\text{si } x \in V(a_i, \frac{r_i}{2}) \text{ entonces } |f(x) - f(a_i)| \leq \frac{\epsilon}{2}$$

y se tiene también que

$$\|p - a_i\| \leq \|p - x\| + \|x - a_i\| \leq \delta + \frac{r_i}{2} \leq r_i$$

de donde $p \in V(a_i, r_i)$ y $|f(p) - f(a_i)| \leq \frac{\epsilon}{2}$.

Por lo tanto,

$$|f(x) - f(p)| \leq |f(x) - f(a_i)| + |f(a_i) - f(p)| \leq \epsilon.$$

O sea, f es uniformemente continua en S y se termina la demostración ya que la subcolección que cubre a S determina la diferencia entre los valores de f y la distancia entre los puntos.

En efecto, por el teorema 1.1, existe el punto mínimo global $x^* \in S$ de la función y la subcolección cubre S entonces $\exists a_j \in S$ tal que

$$x^* \in V(a_j, \frac{r_j}{2}),$$

es decir, finalmente,

$$\|a_j - x^*\| \leq r_j \text{ y } |f(a_j) - f(x^*)| \leq \epsilon.$$

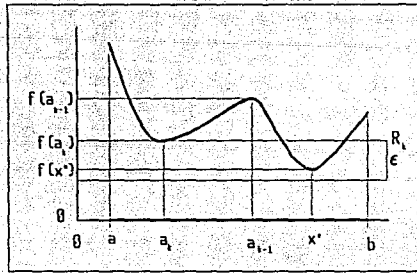


Figura 1.1 Método de la Cobertura aplicado a una función continua

Para concluir note lo siguiente.

- El teorema anterior nos dice que un PMG es localizable, pero no lo determina.
- El método de la cobertura aproxima a un MG, sin embargo, para un ϵ puede dar un resultado erróneo, ya que, puede terminar en un punto de registro que esté muy alejado del PMG correspondiente, tal como se muestra en la figura 1.1.

La familia de métodos de las referencias [Mee88], [Mee89], [Hor88] y [Hor89] que seccionan al conjunto S utilizan lo que se expuso, más otras técnicas: Ramificación y Cota (*Branch and Bound*), Análisis de Intervalos (*Interval Analysis*), etcétera. Todos en general son imprácticos, ya que, requieren demasiadas secciones o iteraciones para garantizar una tolerancia pequeña en la solución.

Por ejemplo, el número de hipercubos de lado $\delta > 0$ con los que se puede cubrir a un hipercubo H de lados de longitud l_1, \dots, l_n es aproximadamente

$$\max_{i=1, \dots, n} \{ \lceil l_i / \delta \rceil \}^n,$$

es decir, el número de secciones que se necesitan es exponencial respecto a la dimensión del problema.

- Para las funciones de Lipschitz (definición 1.24) el error en la determinación del mínimo es proporcional a la longitud del lado (δ), es decir, $\epsilon = \frac{\sqrt{2}\delta}{\lambda}$.

1.5 Aplicaciones de la optimización global

Hablar de aplicaciones propias de la optimización global es difícil. Cuando se aplica el método científico a problemas de tipo industrial (producción, distribución, almacenamiento), financiero (equilibrio económico, inversiones), político, etcétera, es necesario formular modelos matemáticos que los representen de una forma adecuada²⁶.

²⁵ Ver la página 109 del libro [Apo76].

²⁶ Aplicar el método científico es un tema demasiado extenso, que va más allá de los fines de esta tesis, se requieren conocimientos profundos de la aplicación y auxiliarse de otras técnicas para modelar correctamente.

Para esta sección, se buscaron aplicaciones en donde se utilizan los métodos de la optimización, y en particular, las que reportan tener relación directa con la optimización global.

Las aplicaciones que se mencionan en esta sección son: Costo y diseño de un vehículo espacial para aterrizar y Potencial de Lennard-Jones.

La primera se tomo de *Selected Applications of NonLinear Programming*, páginas 58-82 del libro [Bra68]. El objetivo de esta aplicación es seleccionar el menor costo asociado a un diseño de un vehículo espacial. El modelo matemático que se propone consta de una función objetivo, asociada al costo de la nave, y de una serie de restricciones, relacionadas con la eficiencia, la masa, etcétera.

La función objetivo tiene como base las relaciones del costo estimado (*Cost-Estimating Relationships, CER's*), que sirven para asociar los parámetros y las variables de diseño con algún tipo de nave o un subsistema afín. Las partes que la componen se dividen en tres costos: de investigación y de desarrollo, de estructura (*hardware*), y de operación.

El modelo matemático conduce a un problema de optimización global, ya que el mínimo global constituye la mejor selección para la construcción de la nave espacial. Sobra decir que, el gasto involucrado es tan grande que no se puede permitir despilfarro; el costo de la configuración inicial, de uno de los ejemplos que reportan, es \$ 3.36 billones²⁷ y el óptimo es \$ 2.53 billones.

La siguiente aplicación es de Física-Química, el artículo original es [Len32]; la descripción que damos viene en el artículo [Wat76].

Los altos costos de experimentación para la elaboración de nuevos compuestos químicos hacen necesario la utilización de modelos matemáticos que permitan predecir la estructura molecular final. Una forma de simulación se logra por medio de modelos matemáticos de la fuerza intermolecular y de los fenómenos de dispersión.

El modelo matemático de Lennard-Jones 12-6 para cuantificar el potencial V entre N partículas es

$$V = \sum_{i>j \geq 1}^N v_{ij},$$

donde

$$v_{ij} = 4\epsilon \left\{ \left(\frac{\sigma_{ij}}{r_{ij}} \right)^{12} - \left(\frac{\sigma_{ij}}{r_{ij}} \right)^6 \right\}$$

es el potencial entre las partículas i y j ;

$$\sigma_{ij} = \frac{1}{2}(\sigma_i + \sigma_j),$$

$\epsilon > 0$, y σ_k , $k = 1, 2, \dots, N$ son constantes que dependen del tipo de partícula y r_{ij} es la distancia entre las partículas i y j ,

$$r_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2},$$

(x_k, y_k) , $k = 1, 2, \dots, N$ son las coordenadas de las partículas en el plano $X \times Y$. El problema se puede plantear en el espacio y en este caso la distancia es

$$r_{ij} = \sqrt[3]{(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2}$$

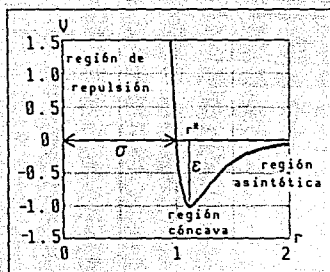


Figura 1.2 Potencial Lennard-Jones 12-6 de dos partículas esféricas idénticas

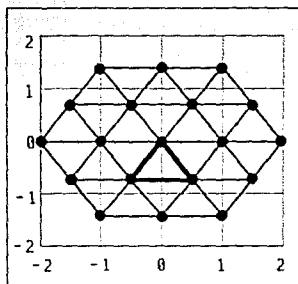


Figura 1.3 Configuración óptima de 19 partículas esféricas idénticas

donde (x_k, y_k, z_k) , $k = 1, 2, \dots, N$ son las coordenadas de las partículas en el espacio $X \times Y \times Z$.

Los valores típicos de ϵ y σ para las interacciones de las moléculas de gases inertes se dan en la tabla 1.1.

Tabla 1.1 ϵ y σ para el potencial de Lennard-Jones 12-6

átomo	He-He	Ne-Ne	Ar-Ar	Kr-Kr	Xe-Xe
σ (Å)	2.556	2.749	3.405	3.600	4.100
$\frac{\epsilon}{k}$ (K)	10.220	35.600	119.800	171.000	221.000

La figura 1.2 muestra la forma de la función potencial de dos partículas esféricas idénticas respecto a la distancia entre ellas. El potencial mínimo se tiene cuando $r = \sqrt[6]{2}\sigma$; la región alrededor del punto óptimo es una concavidad pequeña y constituye la zona de mayor interés; las partículas no pueden estar a una distancia menor a σ , ya que, esta zona es de repulsión. Del otro lado de esta zona, la atracción entre las partículas es muy pequeña cuando la distancia entre ellas es mucho mayor al valor óptimo (región asintótica).

Para el potencial óptimo, las posiciones de 3 partículas esféricas idénticas (de igual valor de σ) forman un triángulo equilátero de lado $\sqrt{2}\sigma$. Para un mayor número de partículas esféricas idénticas se tiene la siguiente conjetura.

Conjetura 1.1 Las posiciones para 4 o más partículas esféricas idénticas en el plano es con base a triángulos equiláteros que se van agrupando de manera que con 7, 19, 37, 61, ... partículas se forman hexágonos. ■

En la figura 1.3 se aprecia como para 7 y 19 partículas los triángulos se agrupan formando hexágonos.

²⁷El billón americano o simplemente billón equivale a mil millones.

Capítulo 2

Métodos de tunelización: original y clásico

En este apartado se describen los orígenes de los algoritmos de los métodos de tunelización (método de tunelización original y clásico), se describen las características de la función de tunelización, las propiedades de los métodos de tunelización y se describe el nuevo algoritmo llamado método de tunelización clásico.

2.1 Origen del método de tunelización

El método de tunelización tiene su origen en los artículos de V. Levy, Gómez, Montalvo, Calderón, et al., que comienzan en 1979 con la tesis doctoral Antonio Montalvo, [Mon79] y que concluyen en 1987 con los artículos [Gom87b] y [Gom87a].

La versión del método de V. Levy y Montalvo, que nombramos método original, tiene dos publicaciones fundamentales: la tesis doctoral [Mon79] y el artículo [Vel85]. Paralelamente, Gómez y V. Levy desarrollaron el método de tunelización para el problema de optimización global con restricciones, sus primeros resultados los publicaron en la comunicación técnica [Vel80b] y posteriormente sacaron el artículo [Gom84]. En 1984 publicaron el artículo [Vel84] que contiene un resumen de todas las aplicaciones del método de tunelización al área de la optimización global (problemas de mínimos cuadrados y problema de optimización global con restricciones y sin restricciones).

Para la realización de este trabajo nos basamos en las publicaciones: [Gol71], [Mon79], [Vel84], [Vel85], [Cor85] y [Gom87b]. De éstos surgió una versión del método de tunelización que llamamos clásico, que esperábamos fuera más eficiente y que diera mejores resultados que la versión original¹.

Un método parecido y que surgió antes del método de tunelización original, es el método de Goldstein y Price que apareció en el artículo [Gol71]. Su método resuelve el problema,

$$\min_{x \in \mathbb{R}^n} f(x) \quad (2.1)$$

donde $f: \mathbb{R}^n \rightarrow \mathbb{R}$ es una función de clase C^2 , mediante una búsqueda que va de un mínimo a otro menor realizando dos fases: deflación y minimización.

¹En la sección 5.4 se comparan los métodos de tunelización original y clásico.

El método de Goldstein y Price requiere de un punto inicial arbitrario, $x_0 \in \mathbb{R}^n$, dado por el usuario y luego realiza los siguientes pasos.

1. Minimización. Dado un punto x_0 se encuentra un punto mínimo local x_1^* por cualquier método de minimización.
2. Deflación. Con el punto mínimo, x_1^* , se construye la función f_2 que se despeja de

$$f_1(x) = f_1(x_1^*) + \frac{[x - x_1^*]^T \nabla^2 f_1(x_1^*) [x - x_1^*]}{2!} f_2(x_1^*, x), \quad (2.2)$$

donde f_1 es la función f del problema 2.1.

La función f_2 se minimiza por cualquier método de optimización local. Si no se encuentra un punto mínimo se termina el algoritmo; x_1^* es un PMG.

En otro caso, sea x_2^* un punto mínimo de f_2 .

Si el valor de f_2 en x_2^* es negativo, este punto se toma como punto de arranque y se repite el paso 1.

Si el valor de f_2 es positivo o cero, el proceso de deflación continua en el paso 3.

3. Con los puntos mínimos: $x_1^*, \dots, x_{k-2}^*, x_{k-1}^*$ de las funciones f_1, \dots, f_{k-1} se construye la función f_k que se despeja de

$$f_{k-1}(x_1^*, \dots, x_{k-2}^*, x) = f_{k-1}(x_1^*, \dots, x_{k-1}^*) + \frac{[x - x_{k-1}^*]^T B_{k-1} [x - x_{k-1}^*]}{2!} f_k(x_1^*, \dots, x_{k-1}^*, x), \quad (2.3)$$

donde $B_{k-1} = \nabla_x^2 f_{k-1}(x_1^*, \dots, x_{k-1}^*)$ (ver la definición de derivada parcial, definición 1.5).

Con cualquier método de minimización se busca un punto mínimo de f_k . Si f_k no tiene más mínimos, se termina el algoritmo; x_1^* es un PMG.

Si el mínimo de f_k es negativo, este punto se toma como punto de arranque y se repite el paso 1, en otro caso, se repite este paso.

Dos puntos importantes del algoritmo son

- La etapa de deflación termina cuando se encuentra un punto de arranque que permita disminuir el valor del último mínimo de la función f , o bien, cuando f_k no tiene más mínimos. En general no hay una forma práctica de conocer cuando se tienen todos los mínimos de la función final f_k (ver la conjetura de Goldstein y Price más adelante).
- Los puntos mínimos de la función f y los puntos mínimos de las funciones f_j , $j = 2, 3, \dots$, no necesariamente coinciden. La diferencia en la posición se debe a los factores que intervienen en la construcción de las funciones f_j , $j = 2, 3, \dots$.

Los autores del algoritmo establecen lo siguiente.

Conjetura 2.1 (Goldstein y Price) Cuando la función f tiene un número finito de puntos estacionarios, las funciones f_j , $j = 2, 3, \dots$ permiten encontrar un nuevo punto de arranque en un número finito de pasos, excepto cuando se trata del mínimo global. En este caso, el proceso de búsqueda de la fase de deflación debe terminar al construir una función f_k que es positiva en todo punto de \mathbb{R}^n . ■

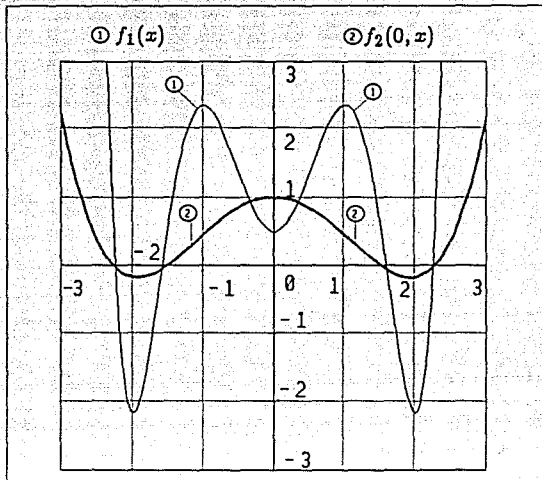


Figura 2.1 Método de Goldstein y Price, paso 1 y 2

Para las funciones polinomiales el proceso es finito, ya que, o se encuentra un punto de arranque o bien se termina en un polinomio de grado cero que es una constante positiva.

Para ilustrar geoméricamente el método, se resuelve detalladamente el problema 2.1 con la función

$$f(x) = \frac{x^6}{3} - \frac{5x^4}{2} + 4x^2 + \frac{1}{2},$$

sus puntos mínimos son $-2, 0$ y 2 , y se muestran en la figura 2.1.

Suponga que se toma un punto arbitrario cerca del punto cero, en este caso, el algoritmo se conduce de la siguiente forma.

1. La fase de minimización termina en el punto mínimo 0 .
2. La fase de deflación realiza lo siguiente.
 - (a) Construye la función $f_2(0, x)$ (en la figura 2.1 se observa que esta función tiene dos puntos mínimos: $-\sqrt{3.75}$ y $\sqrt{3.75}$, que están cerca de los puntos mínimos de f).
 - (b) Minimiza la función f_2 y sin pérdida de generalidad se puede suponer que encuentra el punto mínimo $-\sqrt{3.75}$ (el proceso es similar si encuentra el punto $\sqrt{3.75}$).
 - (c) Como $f_2(-\sqrt{3.75})$ es negativo, se termina esta fase.
3. La siguiente fase de minimización, minimiza la función f comenzando en el punto $-\sqrt{3.75}$ y encuentra el punto mínimo -2 .
4. La siguiente fase de deflación realiza los siguientes pasos.
 - (a) Construye $f_2(-2, x)$ y la minimiza.

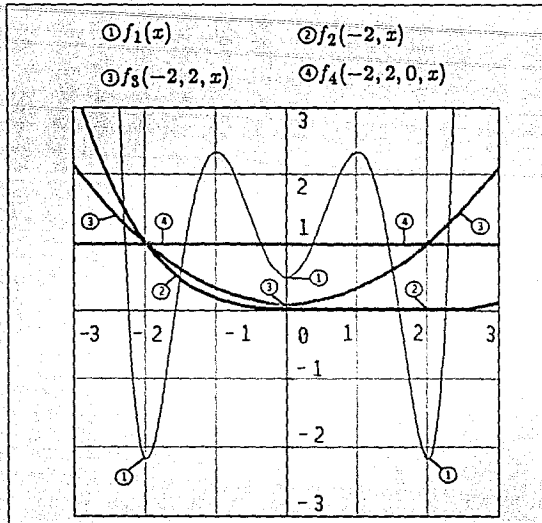


Figura 2.2 Método de Goldstein y Price, paso 3 y 4

- (b) Encuentra el punto mínimo 2, cuyo mínimo es 0 (ver la figura 2.2).
- (c) Como $f_2(-2,0) \geq 0$ construye ahora $f_3(-2,2,x)$ (ver la figura 2.2) y la minimiza.
- (d) Encuentra un mínimo positivo en el punto 0.
- (e) Construye la función $f_4(-2,2,0,x)$.
- (f) Como la función $f_4(-2,2,0,x)$ (ver la figura 2.2) es constante y positiva, se termina la fase de deflación.

5. Se termina el proceso, ya que, la función f_4 es positiva en todas partes y por tanto el punto -2 es un PMG.

En las figuras 2.1 y 2.2, se observa que el efecto geométrico del algoritmo de Goldstein y Price durante la fase de deflación, sobre cada uno de los puntos mínimos de las funciones f_k , es el de rellenar suavemente cada valle de los puntos mínimos.

Una forma parecida de rellenar los valles de los puntos mínimos la propone el investigador chino Renpu en el artículo [Ren90]. Su método se basa en funciones que rellenan los valles de los puntos mínimos, de aquí el nombre de las funciones que utiliza (la definición de las funciones de relleno se da en la página 51). En el capítulo 5 se describe el algoritmo de Renpu y se comparan los resultados del artículo [Ren90] con los que se obtuvieron para este trabajo usando los métodos de tunelización.

2.2 Función de tunelización clásica

El método de tunelización de V. Levy y Montalvo (ver los artículos [Mon79] y [Vel85]) cambia la fase de deflación y la función de deflación del método de Goldstein y Price. En lugar de la fase de deflación se realiza una fase de tunelización que consiste en la búsqueda de un cero de la función de tunelización.

En el método de tunelización se elimina al último punto mínimo creando una función (función de tunelización) que tiende a crecer indefinidamente conforme nos acerquemos a éste. Como se muestra en la figura 2.3, el efecto geométrico es el de rellenar el valle del punto mínimo pero no en la forma tan suave que proponen Goldstein y Price o Renpu.

La función de tunelización es

$$T_c(x, x^*, \lambda^*) = \frac{f(x) - f(x^*)}{\|x - x^*\|^{2\lambda^*}} \quad (2.4)$$

(su forma completa se describe más adelante).

A esta función se le nombra la función de tunelización clásica y la utilizan los métodos: original y clásico. Los dos factores básicos que forman a la función de tunelización son

- $f(x) - f(x^*)$ translación;
- $\frac{1}{\|x - x^*\|^{2\lambda^*}}$ cociente.

Note que hay una correspondencia entre los factores de la función de tunelización con los de la función $f_2(x^*, x)$ (ecuación 2.2). En efecto, el factor de traslación es el mismo, pero, el factor cociente de la función de tunelización $\frac{1}{\|x - x^*\|^{2\lambda^*}}$ se cambia por $\frac{2}{(x - x^*)^2 \nabla^2 f(x^*) (x - x^*)}$ en la función deflacionada.

Formalmente, la selección del factor cociente se basa en lo siguiente.

Definición 2.1 (Polo) Sea $g: \mathbb{R}^n \mapsto \mathbb{R}$ una función y $x^* \in \mathbb{R}^n$ un punto. x^* es un *polo positivo* (o *polo*, simplemente) de g , si para todo $M > 0$, $\exists \delta > 0$ tal que

$$\text{si } \|x - x^*\| < \delta, \text{ entonces } g(x) \geq M.$$

O en forma equivalente, si

$$\lim_{x \rightarrow x^*} g(x) = \infty.$$

En forma análoga se puede definir un polo negativo.

Definición 2.2 (Fuerza del polo) Sean $f: \mathbb{R}^n \mapsto \mathbb{R}$ una función y $x^* \in \mathbb{R}^n$ un punto mínimo de f . El *orden* o *fuerza del polo* en x^* es el primer entero λ^* tal que

$$\lim_{x \rightarrow x^*} \frac{f(x) - f(x^*)}{\|x - x^*\|^{2\lambda^*}} = \infty. \quad (2.5)$$

El sentido de la última definición está dado por la definición anterior, es decir, la ecuación 2.5 significa que $\forall M > 0, \exists \delta > 0$ tal que

$$\text{si } \|x - x^*\|^{2\lambda^*} < \delta \text{ entonces } T_c(x, x^*, \lambda^*) \geq M.$$

El siguiente teorema demuestra la factibilidad de crear polos en funciones de clase C^2 .

Teorema 2.1 Sean $f: \mathbb{R}^n \mapsto \mathbb{R}$ una función de clase C^2 y $x^* \in \mathbb{R}^n$ un punto mínimo de f que cumple las condiciones suficientes de optimalidad del teorema 1.9.

Entonces $\exists \lambda^* \gg 0$ entero tal que

$$\lim_{x \rightarrow x^*} \frac{f(x) - f(x^*)}{\|x - x^*\|^{2\lambda^*}} = \infty.$$

Demostración. Usando el teorema 9.2 del libro [Hes75], página 14, se tiene que $\exists \delta_0 > 0$, $m \in \mathbb{R}$ positivo tal que,

$$\text{si } \|x - x^*\| < \delta_0 \text{ entonces } f(x) \geq f(x^*) + m\|x - x^*\|^2.$$

Despejando m y multiplicando por

$$\frac{1}{\|x - x^*\|^{2\lambda^*}}$$

donde $\lambda^* \geq 2$ es un valor entero arbitrario, se obtiene la desigualdad

$$\frac{f(x) - f(x^*)}{\|x - x^*\|^{2\lambda^*}} \geq \frac{m}{\|x - x^*\|^{2(\lambda^*-1)}}.$$

El resultado se tiene, ya que,

$$\lim_{x \rightarrow x^*} \frac{m}{\|x - x^*\|^{2(\lambda^*-1)}} = \infty.$$

De los teoremas anteriores se tiene la viabilidad de construir funciones de tunelización. Para mostrar el efecto que produce la función de tunelización clásica se usó la función de Shubert,

$$f(x) = \sum_{i=1}^5 \text{icos}\{(i+1)x + 1\}. \tag{2.6}$$

La gráfica de la función de tunelización clásica usando la función de Shubert con un polo en 4.1416 se muestra en la figura 2.3.

Una propiedad de la función de tunelización que se explota en forma práctica se da en el siguiente corolario.

Corolario 2.1 Sea $f: \mathbb{R}^n \mapsto \mathbb{R}$ una función que cumple las condiciones del teorema anterior y sea λ^* la potencia del polo en x^* . Entonces $\forall \lambda > \lambda^*$ se tiene un polo en x^* , es decir,

$$\lim_{x \rightarrow x^*} \frac{f(x) - f(x^*)}{\|x - x^*\|^{2\lambda}} = \infty.$$

Demostración. El resultado se obtiene de que si $\|x - x^*\| < 1$ entonces $\forall \lambda > \lambda^*$ se tiene la siguiente desigualdad,

$$\|x - x^*\|^{2\lambda^*} > \|x - x^*\|^{2\lambda}.$$

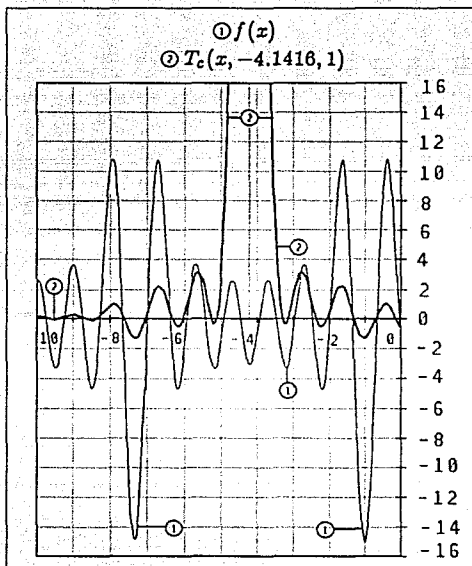


Figura 2.3 Función de tunelización clásica de la función de Shubert

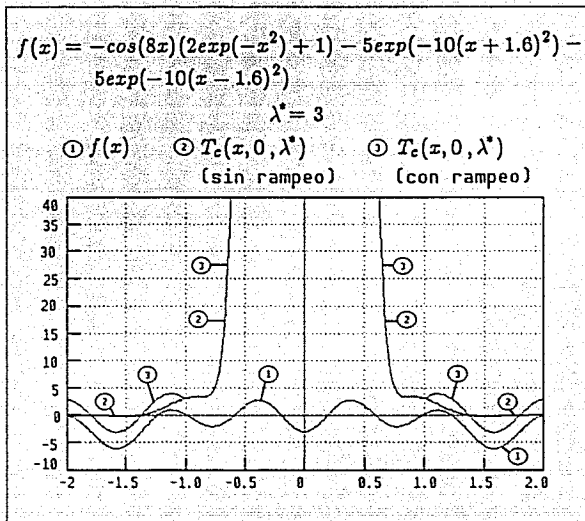


Figura 2.4 Función de tunelización clásica con rampeo y sin rampeo

Este último resultado implica que no es necesario calcular exactamente la fuerza del polo.

Otra propiedad de la función de tunelización es que se aplana para puntos alejados del polo, es decir, $\|x - x^*\| \rightarrow \infty, T_c(x, x^*, \lambda^*) \rightarrow 0$. En la figura 2.3 se aprecia el efecto de aplanamiento.

Tal propiedad no es deseable desde el punto de vista de los métodos numéricos, ya que, si la función se vuelve muy plana alenta la convergencia del método. Este efecto de aplanamiento se quita de los métodos, original y clásico, de manera diferente.

En el método original de V. Levy y Montalvo, ellos introdujeron la siguiente función de rampeo (ver los artículos: [Mon79] y [Vel85]).

$$r'(x, \epsilon_r, \lambda^*, x^*) = \begin{cases} \frac{1}{\|x - x^*\|^{2\lambda^*}}, & \text{si } \|x - x^*\| \leq 1 - \epsilon_r; \\ \frac{1}{\|x - x^*\|^{2\lambda}}, & \text{si } 1 + \epsilon_r \geq \|x - x^*\| > 1 - \epsilon_r; \\ 1, & \text{en otro caso;} \end{cases}$$

donde λ se calcula de la siguiente forma:

$$\lambda = \frac{\lambda^*}{2}(1 + (1 - \|x - x^*\|)/\epsilon_r).$$

Para esta función de rampeo la potencia del factor cociente dentro de la bola $B_{1-\epsilon_r}(x^*)$ es λ^* y en la intersección de esta bola con la bola $B_{1+\epsilon_r}(x^*)$ gradualmente se disminuye la potencia desde λ^* hasta cero y fuera de la bola $B_{1+\epsilon_r}(x^*)$ la potencia vale cero. La manera en que esta función de rampeo disminuye su valor introduce una variación extraña al salir del conjunto diferencia $B_{1+\epsilon_r}(x^*) \setminus B_1(x^*)$, ya que dentro de este conjunto su valor es menor que 1 y fuera de él su valor es 1.

El método clásico evita que el efecto de aplanamiento se propague fuera de la bola $B_1(x^*)$ usando la siguiente función de rampeo.

Definición 2.3 (Función de rampeo) Una función de rampeo $r: \mathbb{R}^n \times \mathbb{R} \times \mathbb{R} \times \mathbb{R}^n \rightarrow \mathbb{R}$ está dada por

$$r(x, \epsilon_r, \lambda^*, x^*) = \begin{cases} \frac{1}{\|x-x^*\|^{2\lambda^*}}, & \text{si } \|x-x^*\| \leq 1-\epsilon_r; \\ \frac{1}{\|x-x^*\|^{2\lambda}}, & \text{si } 1 \geq \|x-x^*\| > 1-\epsilon_r; \\ 1, & \text{en otro caso;} \end{cases}$$

donde λ se calcula de la siguiente forma:

$$\lambda = \frac{\lambda^*}{\epsilon_r} (1 - \|x - x^*\|).$$

Note que la función de rampeo anterior disminuye el efecto del factor cociente de la siguiente forma: dentro de la bola $B_{1-\epsilon_r}(x^*)$ toma el valor: $\frac{1}{\|x-x^*\|^{2\lambda^*}}$ y en la intersección de ésta con la bola $B_1(x^*)$ gradualmente disminuye su valor hasta el valor de 1 y fuera de la bola $B_1(x^*)$ toma el valor de 1.

La ecuación 2.4 de la función de tunelización cuando se usa la función de rampeo cambia a la siguiente forma

$$T_\epsilon(x, x^*, \lambda^*) = (f(x) - f(x^*))r(x, \epsilon_r, \lambda^*, x^*).$$

En la figura 2.5 se muestran los efectos geométricos que producen distintos valores de la potencia del polo en la función de tunelización clásica con rampeo.

En la figura 2.5 se nota un suavizamiento de la forma de la función de tunelización clásica para valores grandes de λ . El suavizamiento de la función significa que disminuye el número de puntos estacionarios y en el teorema 3.8 se demuestra que las funciones de tunelización para un valor de λ^* adecuado no tienen puntos estacionarios en el conjunto

$$\{x \in H \mid f(x) - f(x^*) \geq \epsilon\},$$

donde $\epsilon > 0$ es un valor real dado.

Los resultados que siguen caracterizan los puntos estacionarios de la función de tunelización clásica.

Teorema 2.2 (λ^* para crear un punto estacionario de T_ϵ) Sean $f: \mathbb{R}^n \rightarrow \mathbb{R}$ una función de clase C^2 y $x^* \in \mathbb{R}^n$ un punto mínimo de f . Sea $x_0 \in \mathbb{R}^n$ un punto tal que, $x_0 \neq x^*$,

1. $f(x_0) - f(x^*) > 0$, y el gradiente $\nabla f(x_0)$ y el vector $(x_0 - x^*)$ tengan la misma dirección;
2. o bien, $f(x_0) - f(x^*) < 0$, y el gradiente $\nabla f(x_0)$ y el vector $(x_0 - x^*)$ tengan direcciones opuestas;

entonces en cada caso $\exists \lambda^* > 0$ tal que x_0 es un punto estacionario de $T_\epsilon(x, x^*, \lambda^*)$.

Demostración. En ambos casos las hipótesis garantizan que la suma de los vectores $\nabla f(x_0)$ y $(x_0 - x^*)$ se puede hacer cero.

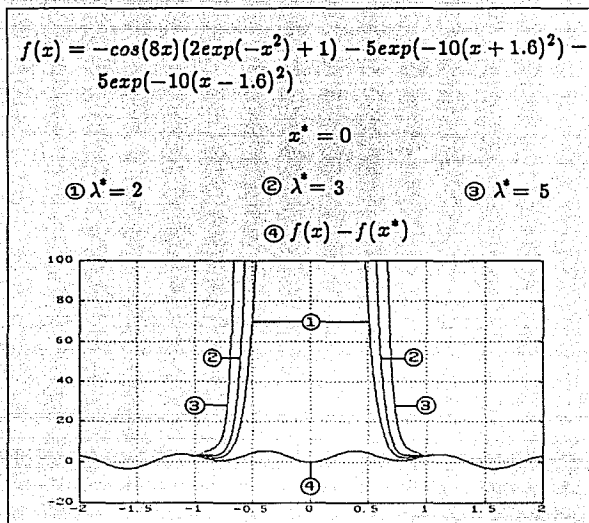


Figura 2.5 Efecto de la potencia del polo λ^* en la función de tunelización clásica

El resultado se obtiene por medio de la siguiente construcción: sea λ^* tal que

$$\begin{aligned}
 0 &= \nabla_x T_c(x_0, x^*, \lambda^*) \\
 &= \frac{\nabla f(x_0)}{\|x_0 - x^*\|^{2\lambda^*}} - 2\lambda^*(f(x_0) - f(x^*)) \frac{x_0 - x^*}{\|x_0 - x^*\|^{2(\lambda^*+1)}} \\
 &= \nabla f(x_0) - \frac{2\lambda^*(f(x_0) - f(x^*))(x_0 - x^*)}{\|x_0 - x^*\|^2}
 \end{aligned}$$

Multiplicando por $(x_0 - x^*)^T$, se despeja λ^* ,

$$\lambda^* = \frac{(x_0 - x^*)^T \nabla f(x_0)}{2(f(x_0) - f(x^*))} > 0.$$

Por lo que, con este valor de λ^* la función $T_c(x, x^*, \lambda^*)$ tiene un punto estacionario en x_0 .

Note que este teorema sólo afirma que en x_0 se tiene un punto estacionario de la función $T_c(x, x^*, \lambda^*)$ y no necesariamente que se tiene un polo en x^* .

Corolario 2.2 (Eliminación de un punto estacionario) Con las hipótesis del teorema anterior, $\exists \lambda^* > 0$ tal que $T_c(x, x^*, \lambda^*)$ tiene un punto estacionario en x_0 . Se afirma que el punto x_0 no es punto estacionario de las funciones $T_c(x, x^*, \lambda_i^*)$ con $\lambda_i^* > \lambda^*$.

Demostración. La afirmación se obtiene de que si $\lambda_1^* > \lambda^*$, tomando un $\epsilon > 0$ tal que $\lambda_1^* = \lambda^* + \epsilon$ y sustituyendo λ_1^* en el gradiente de $T_\epsilon(x, x^*, \lambda_1^*)$ se obtiene que

$$\nabla_x T_\epsilon(x_0, x^*, \lambda_1^*) = -\frac{2\epsilon(f(x_0) - f(x^*))(x_0 - x^*)}{\|x_0 - x^*\|^{2\lambda_1^*}} \neq 0.$$

Corolario 2.3 Con las hipótesis del teorema anterior, $\exists \lambda^* > 0$ tal que $T_\epsilon(x, x^*, \lambda^*)$ tiene un punto estacionario en x_0 . Se afirma que las funciones $T_\epsilon(x, x^*, \lambda_1^*)$ con $\lambda_1^* > \lambda^*$ apropiado tienen un polo en x^* y el punto x_0 no es un punto estacionario de éstas.

Demostración. El teorema 2.1 nos permite encontrar λ_2^* tal que la función $T_\epsilon(x, x^*, \lambda_2^*)$ tenga un polo en x^* . La afirmación se sigue tomando $\lambda_1^* > \max\{\lambda_2^*, \lambda^*\}$ y del corolario 2.1 que afirma que el polo en x^* se mantiene para todos los valores mayores a la potencia λ_2^* . ■

De los teoremas anteriores se tienen resultados similares para la función de tunelización exponencial en la sección 3.1.

En la sección 3.5 se dan resultados más generales para las funciones de tunelización clásica y exponencial.

2.3 Propiedades de los métodos de tunelización

En esta sección se describen las propiedades de los métodos de tunelización. Casi todas las propiedades se heredaron del método original de V. Levy y Montalvo, ya que los cambios introducidos en los nuevos métodos: clásico (sección 2.4) y exponencial (capítulo 3), no las cambiaron.

La principal propiedad que tienen los métodos de tunelización es la propiedad de descenso que consiste en que los valores de f de los puntos obtenidos en las fases de minimización y de tunelización forman una sucesión decreciente, es decir,

$$f(x_0^*) \geq f(x_0^o) \geq f(x_1^*) \geq f(x_1^o) \dots f(x_n^*) \geq f(x_n^o)$$

donde los marcados con $*$ son los puntos mínimos calculados durante las fases de minimización y los marcados con o son los puntos obtenidos de las fases de tunelización.

En los métodos clásico y exponencial se tiene la propiedad anterior, más la propiedad de descenso estricto. Esta propiedad se obtiene al dar un ϵ negativo y pequeño para resolver el problema de la fase de tunelización; el problema de la fase de tunelización consiste en buscar $x^o \in H$, tal que,

$$T(x^o, x^*, \lambda^*) \leq \epsilon$$

donde T es la función de tunelización clásica o la exponencial.

Esta propiedad se puede usar cuando no interesan los mínimos del mismo nivel.

Debido a las dos propiedades anteriores se dice que los métodos de tunelización son métodos de descenso.

Dado que los valores de la función f en los puntos mínimos siempre forman una sucesión de mínimos decreciente, los métodos se pueden detener en algún punto mínimo que se considere interesante y se pueden volver a ejecutar comenzando en ese punto.

Cuando no se encuentra un cero en la fase de tunelización, el último mínimo es el candidato para ser el MG.

Todos los puntos mínimos que se calculan en las fases de minimización pertenecen al conjunto G (salvo que el método de minimización local falle), es decir, son puntos estacionarios (ver la sección 1.2).

2.4 Método de tunelización clásico

El método clásico es muy parecido al método original y como se mencionó en la sección 2.3, ambos, son métodos con la propiedad de descenso.

Ambos métodos (incluso el nuevo método exponencial que se dará en el capítulo 3) realizan dos fases para resolver el problema 1.1.

1. Fase de minimización. Consiste en resolver el problema 1.1 en forma local con cualquier método de optimización, es decir, dado $x^0 \in H$ se busca un punto mínimo de f en H , o sea, un punto x^* tal que,

$$\exists \epsilon > 0, \quad f(x^*) \leq f(x^0) \text{ y } f(x^*) \leq f(x), \quad \forall x \in B_\epsilon(x^*) \cap H.$$

2. Fase de tunelización. Consiste en dado $x^0 \in H$, se construye la función de tunelización

$$T_\epsilon(x, x^*, \lambda^*) = \frac{f(x) - f(x^*)}{\|x - x^*\|^{2\lambda^*}}$$

y se resuelve el problema de tunelización que consiste en buscar $x^0 \in H$ tal que,

$$T_\epsilon(x^0, x^*, \lambda^*) \leq \epsilon \tag{2.7}$$

donde $|\epsilon| \ll 1$ es un valor dado², x^* es el último punto mínimo y λ^* es la potencia del polo³.

Si la fase de tunelización no encuentra x^0 es posible que el último mínimo local sea un PMG.

La descripción completa del método original se encuentra en la comunicación técnica [Mon79] de Antonio Montalvo y Alejandro V. Levy. Los cambios que se realizaron en el algoritmo original para llegar al método clásico se resumen en las tablas 2.1 y 2.2 (en la sección 2.4.2 se describe el algoritmo de la fase de tunelización del método clásico).

En la fase de tunelización se requieren calcular:

- direcciones de búsqueda y
- la función de tunelización, en cualquiera de sus ecuaciones: 2.4, 2.10 y 2.12.

La derivación de la dirección de búsqueda se basa en el método de Newton-Rapshon (ver, por ejemplo, el libro [Dah74]).

Dada la función de tunelización T_ϵ la derivación de la dirección de búsqueda es como sigue,

² ϵ es la tolerancia dada por el usuario para aceptar la solución del problema de tunelización.

³Ver la definición 2.2 y el corolario 2.1.

Sea $x^1 \in H$ un punto arbitrario, $x^1 \neq x^*$.

Se busca $d \in \mathbb{R}^n$, tal que

$$T_c(x^1 + d, x^*, \lambda^*) \approx 0.$$

Aproximandolo por la definición de derivada se tiene,

$$T_c(x^1 + d, x^*, \lambda^*) = T_c(x^1, x^*, \lambda^*) + \nabla_z T_c(x^1, x^*, \lambda^*)^T d + R(d)$$

donde $\lim_{d \rightarrow 0} \frac{R(d)}{\|d\|^2} = 0$.

Multiplicando por $\nabla_z T_c(x^1, x^*, \lambda^*)$, igualando a cero y despejando d se obtiene finalmente

$$d = - \frac{T_c(x^1, x^*, \lambda^*)}{\|\nabla_z T_c(x^1, x^*, \lambda^*)\|^2} \nabla_z T_c(x^1, x^*, \lambda^*). \quad (2.8)$$

Note que el cálculo de la dirección de búsqueda está indeterminado en los puntos estacionarios de la función de tunelización; para evitarlos se introduce el factor del polo móvil,

$$\frac{1}{\|x - x_m\|^{2\lambda_m}} \quad (2.9)$$

donde x_m es un punto estacionario (o una aproximación a él) de T_c y λ_m es su orden o potencia.

Y en las vecindades de los puntos estacionarios de T_c en vez de T_c se usa la función,

$$T_{c1}(x, x^*, \lambda^*, x_m, \lambda_m) = \frac{T_c(x, x^*, \lambda^*)}{\|x - x_m\|^{2\lambda_m}}. \quad (2.10)$$

La función anterior se usa durante la fase de tunelización solamente cuando se requiera, es decir, el polo móvil se prende, se apaga y se recorre según se necesite en forma automática.

La figura 2.6 muestra la gráfica de la función de tunelización clásica sin rampeo usando la función de Shubert (ecuación 2.6) con un polo en 0 y un polo móvil en π .

Otro caso en el que se cambia la función de tunelización T_c es cuando se tienen mínimos del mismo nivel ($f(x_1^*) = \dots = f(x_k^*)$). En este caso se aplica el factor de los polos al mismo nivel,

$$\frac{1}{\prod_{j=1}^k \|x - x_j^*\|^{2\lambda_j}} \quad (2.11)$$

donde x_j^* , $j = 1, \dots, k$ son los puntos correspondientes de los mínimos del mismo nivel y λ_j^* , $j = 1, \dots, k$ son sus potencias.

En este caso la función T_c se cambia en la fase de tunelización por

$$T_{c2}(x, x^*, \lambda^*, x_1^*, \lambda_1^*, \dots, x_k^*, \lambda_k^*) = \frac{T_c(x, x^*, \lambda^*)}{\prod_{j=1}^k \|x - x_j^*\|^{2\lambda_j^*}}. \quad (2.12)$$

La función T_{c2} sólo se usa en todas las fases de tunelización que comiencen con el mismo valor mínimo y basta que se encuentre un mínimo de menor valor al de la fase anterior para dejar de usarla.

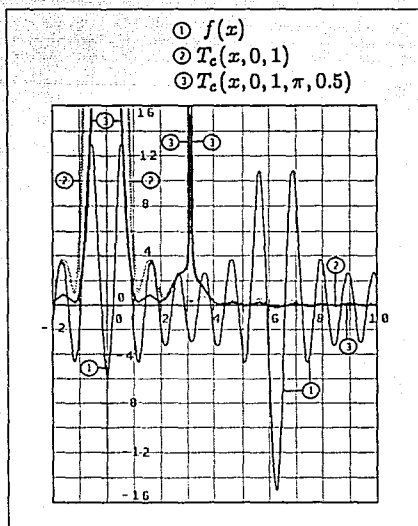


Figura 2.6 Función de tunelización clásica de la función de Shubert con un polo móvil

2.4.1 Cambios en la fase de tunelización del método clásico

El algoritmo que se usa en la fase de tunelización del método clásico no es el mismo que el de Montalvo y Levy. En el algoritmo de la fase de tunelización del método clásico se cambiaron el manejo del polo móvil [Gom87b], las pruebas para prender y apagar el polo móvil, etcétera (las tablas 2.1 y 2.2 mencionan todos los cambios).

Las nuevas condiciones para prender el polo móvil, la fórmula para actualizar la posición del polo móvil, la distancia entre el punto de arranque y el polo móvil, y la introducción del valor de referencia para apagarlo se deben a las investigaciones que reportan Gómez, V. Levy, Cortés y Calderón en el artículo [Gom87b], en la tesis [Cor85] y en la comunicación técnica [Gom87a] para la versión del método de tunelización aplicado al problema de mínimos cuadrados.

El producto $r^T d$, la condición para actualizar la posición del polo móvil, el que la tolerancia del problema de tunelización sea negativo, el que siempre se realice la fase de tunelización y la función de rampeo (definición 2.3) se introducen en este trabajo.

El usar el producto $r^T d$ dentro de la condición para apagar el polo móvil es una idea que surge de la propiedad geométrica que se tienen en los valles de los polos: dentro del valle de un polo (x^*) se cumple que una dirección de descenso d y una dirección radial $r = x - x^*$ forman un ángulo menor a $\frac{\pi}{2}$. Entonces es posible apagar el polo móvil, si la dirección calculada sin el polo móvil apunta en la dirección radial, ya que, se trata de una dirección que se aleja del polo x^* .

La condición para actualizar la posición del polo móvil una vez que se supera la distancia de 0.25 es para que su efecto sobre el cálculo de la dirección se mantenga.

Tabla 2.1 Cambios entre los métodos de tunelización original y clásico

original	clásico
<p>Condiciones para prender el polo móvil o incrementar su fuerza:</p> <ul style="list-style-type: none"> • $f(x^k) \geq f(x^{k^n})$ donde x^k es el último punto generado en la fase de tunelización, $x^{k^n} = x^k + \frac{1}{2^{20}}d$, d es la la dirección de descenso en x^k. • $\nabla_z T_c(x, x^*, \lambda^*) = 0$. 	<p>Condiciones para prender el polo móvil o incrementar su fuerza:</p> <ul style="list-style-type: none"> • $\nabla_z T_c(x, x^*, \lambda^*) = 0$. • Cuando la búsqueda unidimensional no consigue descenso.
<p>Condición para apagar el polo móvil:</p> <ul style="list-style-type: none"> • $d_1^T d_2 > 0$ donde d_1 es la dirección generada con la función $T_c(x^k, x^*, \lambda^*)$ y d_2 es la dirección generada con la función $T_c(x^k, x^*, \lambda^*, x_m, \lambda_m)$; x_m y λ_m son el punto móvil y la potencia del móvil. 	<p>Condiciones para apagar el polo móvil:</p> <ul style="list-style-type: none"> • $r^T d > 0$ donde d es la dirección generada con la función $T_c(x^k, x^*, \lambda^*)$ y r es la dirección radial $x^k - x^*$, x^* es el último punto mínimo y x^k es el último punto generado en la fase de tunelización. • $T_c(x^k, x^*, \lambda^*, x_m, \lambda_m) < V_{ref}$ donde $V_{ref} = 0.9(f(x^m) - f(x^*))$, f es la función objetivo y x^m es el punto donde se prendio el móvil.
<p>Condición para actualizar la posición del polo móvil: $\ x^k - x^{k-1}\ > 1.0$ donde x^k, x^{k-1} son los dos últimos puntos generados en la fase de tunelización.</p>	<p>Condición para actualizar la posición del polo móvil: $\ x^k - x^*\ > 0.25$ donde x^* es el último punto donde se sitió el polo móvil.</p>
<p>Punto de arranque cuando se prende el polo móvil:</p> $x^{k^n} = x^k + \frac{1}{2^{20}}d$ <p>donde d es la dirección de descenso en x^k, x^k es un punto que cumple las condiciones para prender el polo móvil.</p>	<p>Punto de arranque cuando se prende el polo móvil:</p> $x^k = x_m + \epsilon_{a2}r$ <p>donde r es un vector de entradas aleatorias, $\ r\ = 1$, x_m es un punto que cumple las condiciones para prender el polo móvil y $\epsilon_{a2} =$</p> $\begin{cases} \frac{1}{10}, & \text{si } f(x) - f(x^*) \geq 0.1; \\ \frac{f(x) - f(x^*)}{10}, & \text{en otro caso.} \end{cases}$

El que siempre se realice la fase de tunelización en el método clásico aún cuando se alcance la potencia máxima del polo es una decisión arbitraria que se tomó considerando que este tipo de error no afecta al método exponencial; no incluirla en el método clásico lo dejaría en desventaja al compararlo con el método exponencial. Las experiencias obtenidas de los experimentos numéricos demostraron que el método clásico funciona adecuadamente aún cuando no se determine exactamente la potencia del polo.

El cálculo de la función de rampeo del método original se elimina en el método clásico, ya que, introduce una variación extraña al pasar por los puntos que están en el conjunto diferencia de las bolas $B_{1+\epsilon_r}(x^*) \setminus B_1(x^*)$.

Tabla 2.2 Cambios entre los métodos de tunelización original y clásico (cont.)

original	clásico
<p>Actualización de la posición del polo móvil: $x_m = x^k + \beta \frac{x^{k-1} - x^k}{\ x^{k-1} - x^k\ }$ donde x^k, x^{k-1} son los dos últimos puntos generados en la fase de tunelización y β es el primer número de la sucesión $\{\frac{1}{2^i}\}$ tal que $\ x_m - x^k\ < 1$.</p>	<p>Actualización de la posición del polo móvil: $x_m = x^k + 0.1 \frac{x^k - x_m}{\ x^k - x_m\ }$ donde x^k es el último punto generado en la fase de tunelización, x_m es el polo móvil.</p>
<p>Se resuelve el problema: se busca $x^o \in H$ tal que</p> $T_\epsilon(x^o, x^*, \lambda^*) \leq \epsilon$ <p>donde $0 \leq \epsilon \ll 1$ es un valor positivo pequeño, que marca el nivel de aceptación de la fase de tunelización.</p>	<p>Se resuelve el problema: se busca $x^o \in H$ tal que</p> $T_\epsilon(x^o, x^*, \lambda^*) \leq \epsilon$ <p>donde $0 \leq \epsilon \ll 1$ es un valor pequeño (positivo o negativo o cero). Cuando es negativo el método de tunelización tiene la propiedad de descenso estricto (ver la sección 2.3).</p>
<p>No se realiza la fase de tunelización cuando al computar la potencia del polo en el punto mínimo se alcanza el valor máximo.</p>	<p>La fase de tunelización siempre se realiza, aún cuando se alcanza la potencia máxima del polo en el punto mínimo.</p>
<p>Potencia de la función de rampeo $\lambda =$</p> $\begin{cases} \lambda^*, & \text{si } \gamma \leq 1 - \epsilon_r; \\ \frac{\lambda^*}{2} \left(1 + \frac{1-\gamma}{\epsilon_r}\right), & \text{si } 1 + \epsilon_r \geq \gamma > 1 - \epsilon_r; \\ 0, & \text{en otro caso;} \end{cases}$ <p>donde $\gamma = \ x - x^*\$.</p>	<p>Potencia de la función de rampeo $\lambda =$</p> $\begin{cases} \lambda^*, & \text{si } \gamma \leq 1 - \epsilon_r; \\ \frac{\lambda^*}{\epsilon_r} (1 - \gamma), & \text{si } 1 \geq \gamma > 1 - \epsilon_r; \\ 0, & \text{en otro caso;} \end{cases}$ <p>donde $\gamma = \ x - x^*\$.</p>

2.4.2 Algoritmo de la fase de tunelización del método clásico

La versión actual del algoritmo de la fase de tunelización del método clásico de tunelización es

1. Dados $x^* \in H$ punto mínimo y ϵ tolerancia de la solución del problema 2.7, $0 \leq |\epsilon| \ll 1$.
2. Se determina si hay mínimos del mismo nivel y si los hay se usa la función T_{ϵ_2} (ecuación 2.12) en esta fase de tunelización (ver la sección 2.4.3 para mayor información de este paso).
3. Se selecciona un punto aleatorio de arranque x^o tal que

$$\|x^o - x^*\| = 0.1.$$

Y se determina la potencia del polo λ^* , $1 \leq \lambda^* \leq \lambda_{M1}$ (en la sección 2.4.3 se describe como se realiza este paso).

4. Si el número de veces que se ha construido la dirección de búsqueda excede a n_d se continua en 8.

En otro caso se construye la dirección de búsqueda, d , factible (ver la sección 2.4.3 para los detalles de como se realiza este paso).

Si el polo móvil está prendido se calcula $r_d = r^T d'$ donde d' es la dirección generada con la función de tunelización sin el polo móvil y r es la dirección radial $x^o - x^*$.

Si $r_d > 0$ se hace $\lambda_m = 0$ (se apaga el polo móvil) y $d = d'$.

Si $d \neq 0$ se continúa en el paso 6.

5. Si la potencia del polo móvil es cero se toma valor de referencia $V_{ref} = (f(x) - f(x^*)) * 0.9$ y se calcula un punto de arranque,

$$x^o = x_m + \epsilon_{a2} r$$

donde r es un vector de entradas aleatorias, $\|r\| = 1$, x_m es el último punto generado en esta fase (cumple las condiciones para prender el polo móvil) y

$$\epsilon_{a2} = \begin{cases} 0.1, & \text{si } f(x) - f(x^*) \geq 0.1; \\ 0.1(f(x) - f(x^*)), & \text{en otro caso.} \end{cases}$$

Se incrementa la potencia del polo móvil $\lambda_m = \lambda_m + \lambda_s$.

Si $\lambda_m > \lambda_{M2}$ se continúa en el paso 8.

(Se usa la función de tunelización con el factor del polo móvil, ecuación 2.10, hasta que se apague el polo móvil).

Se continua en el paso 4.

6. Se realiza una búsqueda unidimensional de descenso por bisección, es decir, se selecciona el primer β_k , de $\{1, \frac{1}{2}, \frac{1}{2^2}, \frac{1}{2^3}, \dots, \beta_{n_1}\}$ tal que

$$T_c(x^o, x^*, \lambda^*) \geq T_c(x^o + \beta_k d, x^*, \lambda^*).$$

Si no hay descenso se continúa en el paso 5.

Se hace $x^1 = x^o + \beta_k d$.

7. Si $T_c(x^1, x^*, \lambda^*) \leq \epsilon$ se termina; x^1 es una solución del problema 2.7.

En otro caso, se hace $x^o = x^1$. Si el exponente del polo móvil es cero se continúa en el paso 4.

Si $(f(x) - f(x^*)) \geq V_{\epsilon, f}$ se apaga el polo móvil. Si el polo móvil está prendido y $\|x - x^*\| > 0.25$ se actualiza la posición del polo móvil usando

$$x_m = x^o - 0.1 \frac{x^o - x_m}{\|x^o - x_m\|}.$$

Se continúa en el paso 4.

8. Si ya se han calculado más de $4n+1$ puntos de arranque se termina; posiblemente x^* es un PMG.

Si el número de puntos de arranque es menor o igual a $2n$ se genera un punto de arranque en la dirección de los ejes de coordenadas y se continúa en el paso 4. En otro caso, se genera un punto aleatorio de arranque factible x^o (definición 1.7).

Si $T_c(x^o, x^*, \lambda^*) \leq \epsilon$ se termina; este punto una solución del problema 2.7. En otro caso, se continúa en el paso 4.

2.4.3 Detalles de la programación del método clásico

En el paso 4 del algoritmo clásico para que una dirección de búsqueda se convierta en una dirección de búsqueda factible se usa la función de proyección P_H (definición 1.21), o sea, $d' = P_H(x^o, d)$, donde d' es la dirección de búsqueda factible, x^o es el último punto generado en la fase de tunelización y d es la dirección calculada por la ecuación 2.8.

En el paso 3 la determinación de la potencia del polo se realiza de la siguiente forma,

1. dado $x^o \neq x^*$, se toma $\lambda^* = 1$.
2. Se construye la dirección de búsqueda d con la ecuación 2.8.
3. Se calcula el punto $x^1 = x^o + \frac{d}{2-\lambda^*}$.
4. Si $T_c(x^o, x^*, \lambda^*) > T_c(x^1, x^*, \lambda^*)$ se termina; se tiene la potencia del polo.

En otro caso se calcula $\lambda^* = \lambda^* + 1$.

Si $\lambda^* > \lambda_{M1}$ se termina y se escribe el mensaje "no se determino la potencia del polo", en otro caso se continúa en el paso 2.

En el paso 2 del algoritmo clásico se requiere determinar si hay polos del mismo nivel, para realizar esto se usan dos pilas: la pila M_x para los puntos y la pila M_f para los valores de la función f . El manejo de las pilas se realiza mediante las funciones: $push$, pop y top tales que, dada una pila $M = [e_n, e_{n-1}, \dots, e_1]$ tienen las siguientes propiedades:

- $push(M, e) = [e, e_n, \dots, e_1]$;
- $pop(M) = [e_{n-1}, \dots, e_1]$;
- $top(M) = e_n$ y $top([\]) = \infty$.

El método de tunelización clásico inicia con las pilas vacías, $M_x = []$ y $M_f = []$, y en el paso 2 se realiza lo siguiente.

1. Si $top(M_f) = f(x^*)$ se hace $push(M_f, f(x^*))$, $push(M_x, x^*)$ y se termina; hay mínimos del mismo nivel.
2. En otro caso, mientras que $top(M_f) \neq \infty$ se hace $pop(M_f)$ y $pop(M_x)$.

Note que cuando hay polos al mismo nivel se usa la ecuación 2.12 en vez de la ecuación 2.4 durante toda la fase de tunelización que comienza.

El algoritmo clásico a diferencia del original, no se detiene cuando no se puede calcular la potencia del polo, la cual no se puede determinar cuando, por ejemplo, la función del problema es un polinomio y alguna de sus raíces es un punto mínimo de una multiplicidad mayor a λ_{M1} .

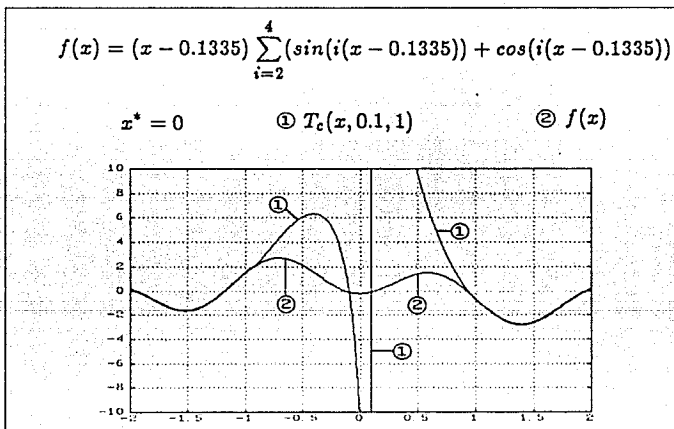


Figura 2.7 “Polo” si x^* no es exactamente un punto mínimo local

Otro caso en el que no se puede determinar la potencia del polo es cuando el punto mínimo no fue localizado exactamente; en la figura 2.7 se muestra la gráfica de una función de tunelización en la que no se forma un polo debido a que el punto mínimo de la función f no fue exactamente determinado.

Otro detalle importante es el cambio que sufren las ecuaciones 2.4, 2.10 y 2.12 de las distintas formas de la función de tunelización cuando se usa la función de rampeo (definición 2.3). En vez de las ecuaciones anteriores se usan

$$T_c(x, x^*, \lambda^*) = (f(x) - f(x^*))r(x, \epsilon_r, \lambda^*, x^*),$$

$$T_{c1}(x, x^*, \lambda^*, x_m, \lambda_m) = T_c(x, x^*, \lambda^*)r(x, \epsilon_r, \lambda_m, x_m) \text{ y}$$

$$T_{c2}(x, x^*, \lambda^*, x_1^*, \lambda_1^*, \dots, x_k^*, \lambda_k^*) = T_c(x, x^*, \lambda^*) \prod_{j=1}^k r(x, \epsilon_r, \lambda_j^*, x_j^*)$$

respectivamente.

Para terminar esta sección se dan los parámetros de la fase de tunelización clásica y los valores que se utilizaron en las corridas de los experimentos numéricos.

- La potencia máxima del polo $\lambda_{M1} \geq 2$ es un número entero (se usó 5).
- La potencia máxima del polo móvil $\lambda_{M2} > 0$ es un número real (se usó 1.0).
- El paso de incremento de la potencia del polo móvil $\lambda_s > 0$ es un número real (se usó 0.1).
- El radio de la función de rampeo $\epsilon_r, 1 \gg \epsilon_r > 0$ es un número real (se usó 0.05).
- El número de bisecciones n_b del paso 6 es un valor entero (se usaron 7 o 15).
- El número de direcciones n_d del paso 4 es un número entero (se usaron 25 o 50 o 100).

Capítulo 3

Método de tunelización exponencial

En este capítulo se describen la función de tunelización exponencial y sus características, se muestran los efectos geométricos de ésta y se comparan con los de la función de tunelización clásica y con los de las funciones de relleno, se describe el algoritmo de la fase de tunelización del método exponencial y en la última sección se dan resultados generales para las funciones de tunelización y se esboza la metodología de tunelización.

3.1 Función de tunelización exponencial

En la sección 2.2 se dio la definición de polo (definición 2.1) y se mostró que el factor cociente,

$$\frac{1}{\|x - x^*\|^{2\lambda^*}}$$

sirve para construirlo.

La nueva función de tunelización exponencial es otra forma de construir los polos; en su forma elemental tiene dos factores: translación y exponencial. Su nombre se debe al uso del factor exponencial en vez del factor cociente de la función de tunelización clásica¹. El factor exponencial para construir el polo en un punto mínimo es

$$\exp\left(\frac{\lambda}{\|x - x^*\|^2}\right)$$

donde x^* es el último punto mínimo calculado en la fase de minimización y $\lambda > 0$ es un parámetro (es el equivalente a la potencia del polo clásico) que determina el ancho del polo.

El siguiente teorema demuestra la posibilidad de crear polos usando el factor exponencial.

Teorema 3.1 (Polo exponencial) Sean $f: \mathbb{R}^n \mapsto \mathbb{R}$ una función de clase C^2 y $x^* \in \mathbb{R}^n$ un punto mínimo de f que cumple las condiciones suficientes de optimalidad de teorema 1.3. Entonces la función

$$T_\varepsilon(x, x^*, \lambda^*) = (f(x) - f(x^*)) \exp\left(\frac{\lambda^*}{\|x - x^*\|^2}\right) \quad (3.1)$$

tiene un polo en x^* para todo $\lambda^* > 0$.

¹Ver los factores básicos de la función de tunelización clásica, página 30.

Demostración. Usando el teorema 3.2 del libro [Hes75], página 14, se tiene que $\exists \delta_0 > 0$, $m \in \mathbb{R}$ positivo tal que,

$$\text{si } \|x - x^*\| < \delta_0 \text{ entonces } f(x) \geq f(x^*) + m\|x - x^*\|^2.$$

Restando $f(x^*)$ a la desigualdad anterior, multiplicando por el factor

$$\exp\left(\frac{\lambda^*}{\|x - x^*\|^2}\right)$$

y desarrollándolo en serie del lado derecho, se obtiene

$$T_\epsilon(x, x^*, \lambda^*) \geq m(\|x - x^*\|^2 + \lambda^* + \sum_{i=0}^{\infty} \frac{\lambda^{*2+i}}{(2+i)! \|x - x^*\|^{2+i}}).$$

Dado que λ^* es fijo, se tiene que la sumatoria,

$$\sum_{i=0}^{\infty} \frac{\lambda^{*2+i}}{(2+i)! \|x - x^*\|^{2+i}} \rightarrow \infty$$

cuando $x \rightarrow x^*$. De donde se sigue que $T_\epsilon(x, x^*, \lambda^*)$ tiene un polo en x^* . ■

Algunas propiedades notables del factor exponencial se dan en los siguientes teoremas.

Teorema 3.2 (Alcance del factor exponencial) Dadas las hipótesis del teorema 3.1 y $\lambda^* > 0$ un valor arbitrario entonces $\forall \epsilon > 0$, $\exists M > 0$ tal que

$$\text{si } \|x - x^*\| \geq M \text{ entonces } |T_\epsilon(x, x^*, \lambda^*) - (f(x) - f(x^*))| \leq \epsilon.$$

Demostración. Usando la definición de T_ϵ se tiene

$$|T_\epsilon(x, x^*, \lambda^*) - (f(x) - f(x^*))| = |f(x) - f(x^*)| \left| \exp\left(\frac{\lambda^*}{\|x - x^*\|^2}\right) - 1 \right|.$$

El resultado se tiene, ya que,

$$\lim_{\|x - x^*\| \rightarrow \infty} \left(\exp\left(\frac{\lambda^*}{\|x - x^*\|^2}\right) - 1 \right) = 0.$$

Teorema 3.3 (Geometría del polo exponencial) Sean $\lambda^* > 0$ y $M > 0$ dos valores reales arbitrarios. El factor exponencial cumple que

$$\exp\left(\frac{\lambda^*}{\|x - x^*\|^2}\right) \leq \exp\left(\frac{\lambda^*}{l^2}\right) = M$$

$$\forall x, \|x - x^*\|^2 \geq l^2 \text{ donde } l^2 = \frac{\lambda^*}{\ln(M)}.$$

Demostración. l^2 se obtiene despejandola de

$$\exp\left(\frac{\lambda^*}{l^2}\right) = M.$$

Y como la función \exp es creciente, es decir, si $a \geq b > 0$ se tiene que $\exp(a) \geq \exp(b)$.

Dado $x \in \mathbb{R}^n$ un punto tal que $\|x - x^*\| = l^2 \geq l^2$, el resultado se tiene en forma inmediata, ya que,

$$\frac{\lambda^*}{l^2} \geq \frac{\lambda^*}{l^2}.$$

Los dos teoremas anteriores muestran la influencia que tiene el factor exponencial sobre la translación $f(x) - f(x^*)$; en el primer teorema se demuestra que para los puntos alejados del polo los valores de $T_\epsilon(x, x^*, \lambda^*)$ y $f(x) - f(x^*)$ son muy similares y en el segundo teorema² se muestra la relación que hay entre un valor fijo M y la distancia al polo $l = \|x - x^*\|$.

En la sección 2.2 se dan resultados que caracterizan los puntos estacionarios de la función de tunelización clásica (teorema 2.2, etcétera); en forma similar, el siguiente teorema caracteriza los puntos estacionarios de la función de tunelización exponencial.

Teorema 3.4 (λ^* para crear o eliminar puntos estacionarios de T_ϵ) *Dada $f: \mathbb{R}^n \rightarrow \mathbb{R}$ una función de clase C^1 y dado $x^* \in \mathbb{R}^n$ un punto mínimo de f . Sea $x_0 \in \mathbb{R}^n$ un punto tal que, $x_0 \neq x^*$,*

1. $f(x_0) - f(x^*) > 0$, y el gradiente $\nabla f(x_0)$ y el vector $(x_0 - x^*)$ tengan la misma dirección;
2. o bien, $f(x_0) - f(x^*) < 0$, y el gradiente $\nabla f(x_0)$ y el vector $(x_0 - x^*)$ tengan direcciones opuestas;

entonces para cada caso $\exists \lambda^* > 0$ tal que el punto x_0 es un punto estacionario de $T_\epsilon(x, x^*, \lambda^*)$. Además, para $\lambda_1^* > \lambda^*$ la función $T_\epsilon(x, x^*, \lambda_1^*)$ no tiene a x_0 como un punto estacionario.

Demostración. En ambos casos las hipótesis garantizan que la suma de los vectores $\nabla f(x_0)$ y $(x_0 - x^*)$ se puede hacer cero.

El primer resultado se obtiene por medio de la siguiente construcción: sea λ^* tal que

$$\begin{aligned} 0 &= \nabla_x T_\epsilon(x_0, x^*, \lambda^*) \\ &= \exp\left(\frac{\lambda^*}{\|x_0 - x^*\|^2}\right) \left(\nabla f(x_0) - \frac{2\lambda^*(f(x_0) - f(x^*))(x_0 - x^*)}{\|x_0 - x^*\|^4}\right) \\ &= \nabla f(x_0) - \frac{2\lambda^*(f(x_0) - f(x^*))(x_0 - x^*)}{\|x_0 - x^*\|^4} \end{aligned}$$

Multiplicando por $(x_0 - x^*)^T$, se despeja λ^* ,

$$\lambda^* = \frac{(x_0 - x^*)^T \nabla f(x_0) \|x_0 - x^*\|^2}{2(f(x_0) - f(x^*))} > 0.$$

Es decir, para λ^* dado por la expresión anterior x_0 es un punto estacionario de $T_\epsilon(x, x^*, \lambda^*)$.

Para la segunda parte, sea $\epsilon > 0$ un valor arbitrario, se toma $\lambda_1^* = \lambda^* + \epsilon$ y se sustituye en el gradiente de $T_\epsilon(x, x^*, \lambda_1^*)$, entonces se obtiene

$$\nabla_x T_\epsilon(x_0, x^*, \lambda_1^*) = \exp\left(\frac{\lambda_1^*}{\|x_0 - x^*\|^2}\right) \left(\nabla f(x) - \frac{2\epsilon(f(x) - f(x^*))(x - x^*)}{\|x - x^*\|^4}\right) \neq 0.$$

O sea, x_0 no es un punto estacionario de $T_\epsilon(x, x^*, \lambda_1^*)$. ■

Los resultados anteriores de la función de tunelización exponencial son similares a los que se dieron en la sección 2.2 para la función de tunelización clásica. Además de estos y como una introducción a la metodología de tunelización se dan otros en la sección 3.5 que son validos

²En el paso 5 del algoritmo de la fase de tunelización del método exponencial se calcula la distancia del punto de arranque basándose en el teorema de la geometría del factor exponencial.

para las dos funciones de tunelización. Por ejemplo, el teorema 3.8 demuestra que para un valor adecuado de λ^* las funciones clásica y exponencial no tienen puntos estacionarios en el conjunto

$$\{x \in H \mid f(x) - f(x^*) \geq \epsilon\}$$

donde $\epsilon > 0$ es valor dado.

Las dos subsecciones siguientes muestran los efectos geométricos de las funciones de tunelización y se describen las funciones de relleno de Renpu Ge. Las funciones de relleno se describen para resaltar que usan a la función exponencial $\exp(\cdot)$ en forma diferente a la función de tunelización exponencial.

3.1.1 Efectos geométricos de las funciones de tunelización y de relleno

Las figuras 3.1 y 3.2 muestran lo similares que son las funciones de tunelización clásica con rampeo y la exponencial. Para estas figuras se usó la función

$$f(x) = -\cos(8x) - 2\exp(-x^2).$$

La figura 3.1 corresponde a la función de tunelización clásica con el polo en 0 y $\lambda^* = 1$.

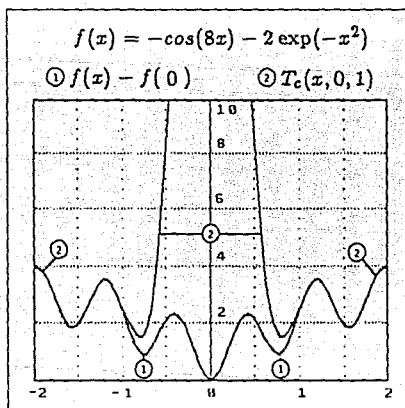


Figura 3.1 Función de tunelización clásica con rampeo y un polo en 0

En la figura 3.2 se muestran las formas que adoptan para $\lambda^* = 0.1, 1, 2$ y 3 (numerador del factor exponencial) las funciones de tunelización exponencial con el polo en 0. En esta figura se aprecia la similitud que tienen las funciones de tunelización clásica ($\lambda^* = 1$) y exponencial ($\lambda^* = 0.1$). También se nota que el ancho del polo aumenta en forma directa al aumentar λ^* para la función exponencial.

De aquí la conjetura acerca de la eficiencia del método exponencial respecto a la eficiencia de los métodos clásico y el original.

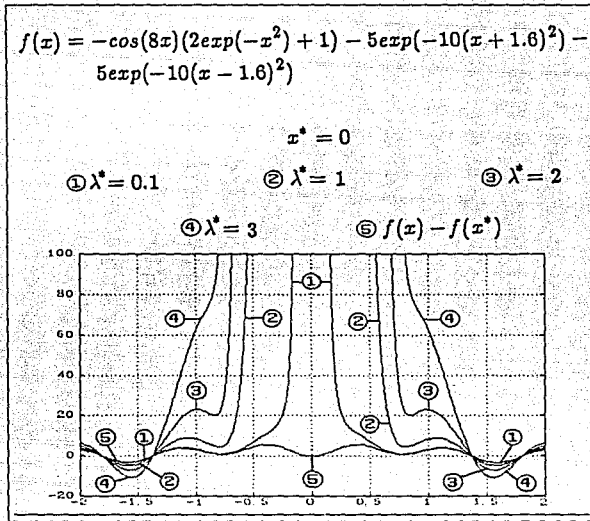


Figura 3.2 Funciones de tunelización exponencial con un polo en 0

Conjetura 3.1 (Método exponencial vs clásico) *Para cualquier problema 1.1 $\exists \lambda^* > 0$ del factor exponencial tal que el método de tunelización exponencial resuelve el problema 1.1 en menos evaluaciones de la función objetivo que el método clásico o que el método original.*

■

La conjetura anterior se apoya en los efectos geométricos, cabe mencionar que, para algunos valores de λ^* del factor exponencial se crean puntos estacionarios adicionales, como se aprecia en la figura 3.2 para $\lambda^* = 0.01$, que no existen en la función de tunelización clásica y que pueden ocasionar que el método exponencial requiera de un mayor número de evaluaciones de la función objetivo que el método clásico. A pesar de esto, el método exponencial, a diferencia del método clásico, con el parámetro λ^* abre la posibilidad de realizar una búsqueda más eficiente (disminuyendo el número de evaluaciones de la función objetivo) porque fácilmente puede hacer más anchos los polos y de esta manera quitar zonas de puntos cuyos valores no son interesantes.

3.1.2 Funciones de relleno

El uso de factores de tipo exponencial no es exclusivo del método de tunelización exponencial, el investigador Renu (que se mencionó en la página 29) los utiliza en sus funciones de relleno del artículo [Ren90].

En el método de funciones de relleno se utiliza la función exponencial $\exp(\cdot)$ y el efecto geométrico es más parecido al del algoritmo de Goldstein y Price³.

Las funciones de relleno se definen de la siguiente forma

$$R(x, x^*, r, \rho) = \frac{\exp\left(\frac{\|x-x^*\|^2}{\rho^2}\right)}{r + f(x)}$$

donde f es la función del problema 1.1; x^* es un punto mínimo de f ; $r > 0$ y $\rho \neq 0$ son dos parámetros del método de las funciones de relleno.

Para mostrar los efectos geométricos de las funciones de relleno se tomó la función,

$$f_2(x) = -\sum_{i=2}^4 \{ \text{sen}(i(x - 0.1335)) + \text{cos}(i(x - 0.1335)) \}$$

en el intervalo $-2 \leq x \leq 2$.

Las gráficas de las funciones f_2 y de relleno en el punto mínimo 0, con $r = 10$ y $\rho = 0.5$ se muestran en la figura 3.3.

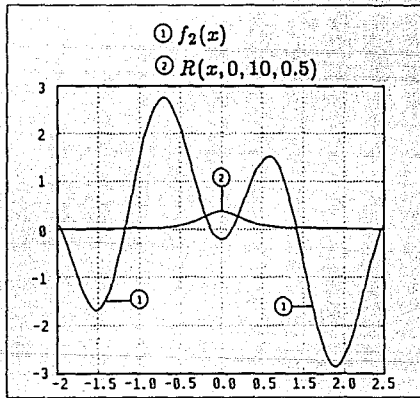


Figura 3.3 Función de relleno de f_2

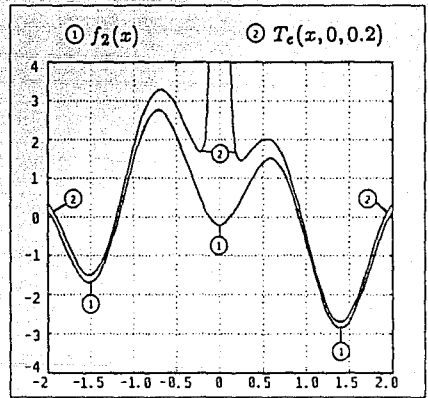


Figura 3.4 Función de tunelización exponencial de f_2

Por otra parte, en la figura 3.4 se muestran las gráficas de las funciones f_2 y de tunelización exponencial con el polo en 0.

Comparando las figuras 3.3 y 3.4 es evidente que el efecto de las funciones de relleno es menos energético que el efecto geométrico de los polos de las funciones de tunelización (ver la definición de polo, definición 2.1).

³El algoritmo de Goldstein y Price se da en la página 29 de este trabajo.

3.2 Método de tunelización exponencial

El método exponencial realiza, al igual que el método clásico de tunelización, dos fases para resolver el problema 1.1:

1. Fase de minimización. Dado $x^0 \in H$ se busca un mínimo local de f en H , es decir, se busca un punto $x^* \in H$ tal que,

$$\exists \epsilon > 0, \quad f(x^*) \leq f(x^0) \text{ y } f(x^*) \leq f(x), \quad \forall x \in B_\epsilon(x^*) \cap H.$$

2. Fase de tunelización. Dados $x^* \in H$ y $\epsilon, 0 \leq |\epsilon| \ll 1$, se busca un punto $x^0 \in H$ tal que,

$$T_\epsilon(x^0, x^*, \lambda^*) \leq \epsilon. \tag{3.2}$$

Note que lo único que se cambió del método clásico (a este nivel de descripción) es que en el método exponencial se usa la función T_ϵ en lugar de la función T_c en la fase de tunelización, básicamente la fase de tunelización clásica y la exponencial requieren de cálculos muy parecidos:

- para la dirección de búsqueda y
- para la evaluación de la función de tunelización.

Para la función de tunelización exponencial se omite la derivación de la dirección de búsqueda por ser similar a la que se dio para obtener la ecuación 2.8. En este caso, la dirección de búsqueda del problema de tunelización exponencial (ecuación 3.2) es

$$d = - \frac{T_\epsilon(x, x^*, \lambda^*)}{\|\nabla_x T_\epsilon(x, x^*, \lambda^*)\|^2} \nabla_x T_\epsilon(x, x^*, \lambda^*). \tag{3.3}$$

Cuando los puntos estacionarios de T_ϵ no permiten calcular la dirección de búsqueda se introduce, para evitarlos, el factor exponencial del polo móvil,

$$\exp\left(\frac{\lambda_m}{\|x - x_m\|^2}\right) \tag{3.4}$$

donde x_m es un punto estacionario (o una aproximación a él) de $T_\epsilon(x, x^*, \lambda^*)$.

En la vecindades de los puntos estacionarios se cambia la función T_ϵ por

$$T_{\epsilon 1}(x, x^*, \lambda^*, x_m, \lambda_m) = T_\epsilon(x, x^*, \lambda^*) \exp\left(\frac{\lambda_m}{\|x - x_m\|^2}\right). \tag{3.5}$$

Finalmente, para los mínimos del mismo nivel se introduce un factor similar al que se usa en $T_{\epsilon 2}$ para evitarlos, en este caso se cambia T_ϵ por

$$T_{\epsilon 2}(x, x^*, \lambda^*, x_1^*, \lambda_1^*, \dots, x_k^*, \lambda_k^*) = T_\epsilon(x, x^*, \lambda^*) \prod_{j=1}^k \exp\left(\frac{\lambda_j^*}{\|x - x_j^*\|^2}\right). \tag{3.6}$$

La ecuación anterior se usa en la fases de tunelización en lugar de T_ϵ cuando hay mínimos del mismo nivel, pero basta encontrar un mínimo de menor valor para volver a usar la función T_ϵ en la siguiente fase de tunelización.

Los pasos del algoritmo de la fase de tunelización exponencial son muy similares a los pasos de la misma fase del método clásico. En la sección 3.4 se dan las diferencias entre ambos algoritmos.

3.2.1 Algoritmo de la fase de tunelización del método exponencial

La versión del algoritmo de la fase de tunelización del método exponencial es

1. Dados $x^* \in H$ un punto mínimo y ϵ una tolerancia de la solución del problema 3.2, $0 \leq |\epsilon| \ll 1$.
2. Se determina si hay mínimos del mismo nivel y si los hay se usa la función T_{22} (ecuación 3.6) en esta fase de tunelización (ver la sección 3.2.2 para mayor información de este paso).
3. Se selecciona un punto aleatorio de arranque x^o tal que

$$\|x^o - x^*\| = 0.1.$$

4. Si el número de veces que se ha construido la dirección de búsqueda excede a n_d se continúa en 8.

En otro caso se construye la dirección de búsqueda d factible (ver la sección 3.2.2 para los detalles de como se realiza este paso).

Si el polo móvil está prendido se calcula $r_d = r^T d'$ donde d' es la dirección generada con la función de tunelización sin el polo móvil y r es la dirección radial $x^o - x^*$.

Si $r_d > 0$ se hace $\lambda_m = 0$ (se apaga el polo móvil) y $d = d'$.

Si $d \neq 0$ se continúa en el paso 6.

5. Si la potencia del polo móvil es cero se toma valor de referencia $V_{ref} = (f(x) - f(x^*)) * 0.9$.

Se incrementa la potencia del polo móvil $\lambda_m = \lambda_m + \lambda_d$.

Si $\lambda_m > \lambda_{M2}$ se continúa en el paso 8.

Se calcula el punto de arranque,

$$x^o = x_m + \frac{\lambda_m}{\ln(M)} r$$

donde r es un vector de entradas aleatorias, $\|r\| = 1$, x_m es el último punto generado (cumple las condiciones para prender el polo móvil) y $M > 0$ es un parámetro.

(Se usa la función de tunelización con el factor del polo móvil, ecuación 3.5, hasta que se apague el polo móvil).

Se continúa en el paso 4.

6. Se realiza una búsqueda unidimensional de descenso por bisección, es decir, se selecciona el primer β_k , de $\{1, \frac{1}{2}, \frac{1}{3}, \frac{1}{4}, \dots, \beta_{n_1}\}$ tal que

$$T_2(x^o, x^*, \lambda^*) \geq T_2(x^o + \beta_k d, x^*, \lambda^*).$$

Si no hay descenso se continúa en el paso 5.

Se hace $x^1 = x^o + \beta_k d$.

7. Si $T_\epsilon(x^1, x^*, \lambda^*) \leq \epsilon$ se termina; x^1 es una solución del problema 3.2.

En otro caso, se hace $x^0 = x^1$. Si el exponente del polo móvil es cero se continúa en el paso 4.

Si $(f(x) - f(x^*)) \geq V_{ref}$ se apaga el polo móvil. Si el polo móvil está prendido y $\|x - x^*\| > \frac{\lambda_m + 1}{\ln(M)}$ se actualiza la posición del polo móvil usando

$$x_m = x^0 - \frac{\lambda_m}{\ln(M)} \frac{x^0 - x_m}{\|x^0 - x_m\|}.$$

Se continúa en el paso 4.

8. Si ya se han calculado más de $4n+1$ puntos de arranque se termina; posiblemente x^* es un PMG.

Si el número de puntos de arranque es menor o igual a $2n$ se genera un punto de arranque en la dirección de los ejes de coordenadas y se continúa en el paso 4. En otro caso, se genera un punto aleatorio de arranque factible x^0 (definición 1.7). Si $T_\epsilon(x^0, x^*, \lambda^*) \leq \epsilon$ se termina; este punto es una solución del problema 3.2. En otro caso, se continúa en el paso 4.

3.2.2 Detalles de la programación del método exponencial

En el paso 4 del algoritmo exponencial para que la dirección d sea una dirección factible se aplica la función de proyección P_H (definición 1.21), o sea, $d' = P_H(x^0, d)$, donde d' es la dirección de búsqueda factible, x^0 es el último punto generado en la fase de tunelización y d es la dirección calculada por la ecuación 3.3.

En el paso 2 del algoritmo exponencial se requiere determinar si hay polos del mismo nivel y usan dos pilas: M_x pila para los puntos y M_f pila para los valores de la función f . El proceso necesario es idéntico al del método de tunelización clásico para el paso 2 que se describió en la sección 2.4.3.

Note que cuando hay polos al mismo nivel se usa la ecuación 3.6 en vez de la ecuación 3.1 durante toda la fase de tunelización que comienza.

Para terminar esta sección se dan los parámetros de la fase de tunelización exponencial y los valores que se utilizaron en las corridas de los experimentos numéricos.

- La potencia máxima del polo móvil $\lambda_{M2} > 0$ es un número real (se usó 2.0).
- El paso de incremento de la potencia del polo móvil $\lambda_i > 0$ es un número real (se usó 0.1).
- El parámetro $M > 0$ es un valor real, se usa para calcular la posición del polo móvil y para decidir cuando se actualiza su posición. Este parámetro se introduce por la relación que se establece entre el valor del factor exponencial y la anchura del polo en el teorema 3.3 (se usó 100).
- El número de bisecciones n_b del paso 6 es un valor entero (se usaron 7 o 15).
- El número de direcciones n_d del paso 4 es un número entero (se usaron 25 o 50 o 100).

3.2.3 Factor para eliminar los mínimos del mismo nivel

Los problemas con mínimos del mismo nivel son muy comunes, en el capítulo 5 se han elegido algunos. Hay dos alternativas para que los métodos clásico y exponencial tengan la propiedad de descenso estricto:

1. Usar un valor ϵ pequeño y negativo en los problemas de la fase de tunelización, ecuaciones 2.7 y 3.2.
2. O bien, usar el siguiente factor

$$\exp\left(\frac{\lambda_f}{(f(x) - f(x^*))^2}\right) \quad (3.7)$$

donde $\lambda_f > 0$.

El factor exponencial anterior elimina los valores del mismo nivel que $f(x^*)$ y es introducido para evitar las expresiones de la función de tunelización dadas por las ecuaciones 3.6 y 2.12.

Cuando se usa el factor anterior las funciones de la fase de tunelización clásica y exponencial son respectivamente:

$$T_{e'}(x, x^*, \lambda^*) = T_e(x, x^*, \lambda^*) \exp\left(\frac{\lambda_f}{(f(x) - f(x^*))^2}\right)$$

$$T_{e''}(x, x^*, \lambda^*) = T_e(x, x^*, \lambda^*) \exp\left(\frac{\lambda_f}{(f(x) - f(x^*))^2}\right).$$

El usar las funciones anteriores no cambia los pasos de los algoritmos de la fase de tunelización, sólo afectan la ejecución y únicamente en el paso en donde se determinan los polos del mismo nivel, ya que, nunca se activaran las funciones de las ecuaciones 3.6 y 2.12. Además, la fase de tunelización termina con un valor menor al que entró y no se requiere la memoria de las pilas de los puntos y de los valores de la función que se mencionaron en las secciones 2.4.3 y 3.2.2.

3.3 Diferencias entre las funciones de tunelización

La tabla 3.1 resume los factores que se usan en la construcción de las funciones de tunelización y los ubica de acuerdo a la equivalencia en su uso.

El numerador $\lambda^* > 0$ del factor exponencial siempre crea un polo en el punto mínimo. La potencia del factor cociente de la función clásica para crear un polo en el punto mínimo requiere de un proceso para determinarlo y puede no ser determinado porque el punto mínimo no está exactamente localizado o, bien, porque se alcanza un valor máximo prefijado λ_{M1} .

El teorema 3.3 muestra la relación directa que hay entre el numerador λ^* y el ancho del polo que se crea. Para evitar el aplanamiento se introduce en la función clásica unas funciones de rampeo que limitan los efectos de los factores clásicos a un radio de uno.

La dirección de búsqueda de ambos métodos fuera de la zona de influencia (radio uno para el clásico y radio variable para el exponencial) es aproximadamente

$$-\frac{f(x)}{\|\nabla f(x)\|^2} \nabla f(x).$$

Tabla 3.1 Factores de los métodos de tunelización

clásico	exponencial
Polo en el punto mínimo $\frac{1}{\ x - x^*\ ^{2\lambda^*}}$	Polo en el punto mínimo $\exp\left(\frac{\lambda^*}{\ x - x^*\ ^2}\right)$
Polo móvil $\frac{1}{\ x - x_m\ ^{2\lambda_m}}$	Polo móvil $\exp\left(\frac{\lambda_m}{\ x - x_m\ ^2}\right)$
Polos en puntos mínimos del mismo nivel $\frac{1}{\prod_{j=1}^k \ x - x_j^*\ ^{2\lambda_j^*}}$	Polos en puntos mínimos del mismo nivel $\prod_{j=1}^k \exp\left(\frac{\lambda_j^*}{\ x - x_j^*\ ^2}\right)$

En el caso de la función T_c este efecto es una consecuencia del teorema 3.2, ya que, lejos del punto mínimo el factor exponencial va convergiendo asintóticamente a uno y su efecto se diluye.

Por otro lado, si al factor cociente de la función clásica no se le aplica la función de rampeo, un resultado equivalente no se tiene, ya que, lejos del punto mínimo la función T_c converge a cero⁴.

El resultado de la función de tunelización clásica con rampeo (ver la definición 2.3, ver la sección 2.4.3), para los puntos alejados del polo en un radio mayor o igual a uno es exactamente $f(x) - f(x^*)$. ¡Este efecto se tiene aproximadamente y de manera natural con los factores exponenciales!

3.4 Diferencias entre los métodos de tunelización

Las diferencias en los métodos ocurren en los algoritmos de las fases de tunelización. La fase de minimización y el ciclo entre una fase de minimización y una fase de tunelización es el mismo en ambos métodos.

El algoritmo de la fase de tunelización exponencial cambia respecto al de la fase del clásica en los siguientes aspectos:

- Se elimina la determinación de la potencia del polo.
- El cálculo de la función exponencial no usa rampeo (definición 2.3) para evitar que el efecto del polo se propague.

⁴Para evitar que los factores clásicos tomen valores muy pequeños, que hagan que la función T_c se aplane y se vuelva casi cero, V. Levy y Montalvo en los artículos [Mon79] y [Vel85], introducen una función de rampeo, ver la página 33 de este trabajo para mayores detalles.

- El cálculo de la posición del polo móvil usa el factor $\frac{\lambda_m}{\ln(M)}$.

Los parámetros comunes en ambos métodos son:

- $\epsilon_m > 0$ tolerancia para el problema de minimización local;
- ϵ tolerancia del problema de la fase de tunelización, problema 2.7 y problema 3.2;
- 0.1 se toma como radio para elegir el punto de arranque en la fase de tunelización;
- λ_{M2} valor máximo para la potencia del polo móvil;
- λ , valor para incrementar la potencia del polo móvil;
- N_d número de veces que se construye una dirección de búsqueda;
- N_b número máximo de bisecciones en el descenso y
- λ_f numerador del factor exponencial para eliminar los mínimos del mismo nivel.

Los parámetros que sólo utiliza el método clásico son

- λ_{M1} valor de la potencia máxima del polo;
- $\epsilon_r > 0$ radio para la función de rampeo en el cálculo de la función T_c .

El parámetro que sólo utiliza el método exponencial es M valor real positivo que se usa para calcular la posición del polo móvil y para decidir cuando se actualiza su posición (ver el teorema 3.3 y la sección 3.2.2).

3.5 Metodología de tunelización

En las secciones 2.2 y 3.1 se tienen resultados similares para las funciones de tunelización clásica y exponencial. A continuación se dan resultados generales que valen para las dos funciones.

Teorema 3.5 (Puntos estacionarios del mismo nivel) Sean $f: \mathbb{R}^n \rightarrow \mathbb{R}$ una función de clase C^2 , $x^* \in \mathbb{R}^n$ un punto mínimo de f y $\lambda^* > 0$ tal que T tenga un polo en x^* . Sea $x_0 \in \mathbb{R}^n$ un punto mínimo de f tal que $x_0 \neq x^*$ y $f(x_0) = f(x^*)$ entonces x_0 es un punto estacionario de T donde T es cualesquiera de las funciones T_c o T_e .

Demostración. Las funciones gradiente de las funciones T_c y de T_e son

$$\begin{aligned} \nabla_x T_c(x, x^*, \lambda^*) &= \frac{\nabla f(x)}{\|x - x^*\|^{2\lambda^*}} - 2\lambda^*(f(x) - f(x^*)) \frac{x - x^*}{\|x - x^*\|^{2(\lambda^*+1)}} \\ \nabla_x T_e(x, x^*, \lambda^*) &= \exp\left(\frac{\lambda^*}{\|x - x^*\|^2}\right) \left(\nabla f(x) - \frac{2\lambda^*(f(x) - f(x^*))(x - x^*)}{\|x - x^*\|^4} \right). \end{aligned}$$

En ambos casos se anulan ya que $\nabla f(x_0) = 0$ y $(f(x_0) - f(x^*)) = 0$.

Teorema 3.6 Con las hipótesis del teorema anterior y sean $x, y \in \mathbb{R}^n$ dos puntos tales que $x \neq y$, $T(x, x^*, \lambda^*) = T(y, x^*, \lambda^*)$, $x^* \neq x + \gamma(y - x) \forall \gamma \in [0, 1]$ entonces $\exists x \in \mathbb{R}^n$ tal que $\nabla_x T(x^*, x^*, \lambda^*) = 0$ o es perpendicular al vector $(y - x)$.

Demostración. Sin importar si se trata de la función clásica o de la exponencial, ambas por construcción son continuas en $\mathbb{R}^n \setminus \{x^*\}$. Sea la función $h: [0, 1] \rightarrow \mathbb{R}$ dada por

$$h(\gamma) = T(x + \gamma(y - x), x^*, \lambda^*).$$

La función h es continua y se tiene que $h(0) = h(1)$, así que por el teorema de Rolle⁵ $\exists \gamma' \in [0, 1]$ tal que $\frac{dh(\gamma')}{d\gamma} = 0$. Aplicando la regla de derivación de la cadena se tiene que

$$\frac{dh(\gamma')}{d\gamma} = \nabla_x T(x + \gamma'y, x^*, \lambda^*)(y - x).$$

Se sustituye γ' en la última ecuación, se tiene $x' = x + \gamma'y$. Finalmente, se tiene que

$$\nabla_x T(x', x^*, \lambda^*) = 0 \text{ o es perpendicular al vector } (y - x)$$

Teorema 3.7 (Puntos mínimos abajo del nivel 0) Sean $f: \mathbb{R}^n \rightarrow \mathbb{R}$ una función de clase C^2 , $H \subset \mathbb{R}^n$ un hipercubo, $x^* \in \mathbb{R}^n$ un punto mínimo de f en H , $\lambda^* > 0$ tal que T tenga un polo en x^* y $x_0 \in \mathbb{R}^n$ un punto mínimo de f en H tal que, $x_0 \neq x^*$, $f(x_0) < f(x^*)$. Entonces la función T tiene un mínimo menor que cero en una vecindad de x_0 donde T es cualesquiera de las funciones de tunelización.

Demostración. De las hipótesis $\exists \epsilon > 0$ tal que

$$\forall y \in B_\epsilon(x_0) \cap H, \text{ se tiene } f(y) \geq f(x_0).$$

Por construcción las funciones de tunelización son continuas en $H \setminus \{x^*\}$.

Si $x^* \notin B_\epsilon(x_0)$ se tiene que T tiene un mínimo en $B_\epsilon(x_0) \cap H$. Si $x^* \in B_\epsilon(x_0)$, tomando $\epsilon' = \frac{\|x_0 - x^*\|}{2}$ se tiene que T tiene un mínimo en $B_{\epsilon'}(x_0) \cap H$.

En ambos casos como $f(x_0) - f(x^*) < 0$ se tiene que el mínimo de T es negativo. ■

Teorema 3.8 (Suavizamiento de las funciones de tunelización) Sean $f: \mathbb{R}^n \rightarrow \mathbb{R}$ una función de clase C^2 , $H \subset \mathbb{R}^n$ un hipercubo, $x^* \in \mathbb{R}^n$ un punto mínimo de f en H , $\lambda^* > 0$ tal que T tenga un polo en x^* y $x_0 \in \mathbb{R}^n$ un punto mínimo de f en H tal que, $x_0 \neq x^*$, $f(x_0) < f(x^*)$. Entonces $\exists \lambda_c \gg 1$ tal que la función T tiene un polo en x^* y no tiene puntos estacionarios en los conjuntos:

$$\{x \in H \mid f(x) - f(x^*) \geq \epsilon\},$$

$$\{x \in H \mid f(x) - f(x^*) \leq -\epsilon\},$$

donde T es cualesquiera de las funciones T_c o T_ϵ y ϵ es un valor real positivo y pequeño.

⁵Ver la página 132 del libro [Apo76].

Demostración. Las funciones gradiente de las funciones T_e y de T_c son

$$\nabla_x T_e(x, x^*, \lambda^*) = \frac{\nabla f(x)}{\|x - x^*\|^{2\lambda^*}} - 2\lambda^*(f(x) - f(x^*)) \frac{x - x^*}{\|x - x^*\|^{2(\lambda^*+1)}} \quad (3.8)$$

$$\nabla_x T_c(x, x^*, \lambda^*) = \exp\left(\frac{\lambda^*}{\|x - x^*\|^2}\right) \left(\nabla f(x) - \frac{2\lambda^*(f(x) - f(x^*))(x - x^*)}{\|x - x^*\|^4} \right), \quad (3.9)$$

Quitando los factores comunes se tiene para (ecuación 3.8)

$$\nabla f(x) - 2\lambda^*(f(x) - f(x^*)) \frac{x - x^*}{\|x - x^*\|^2} \neq 0,$$

y para (ecuación 3.9)

$$\nabla f(x) - \frac{2\lambda^*(f(x) - f(x^*))(x - x^*)}{\|x - x^*\|^4} \neq 0.$$

Como la norma de $\nabla f(x)$ es acotada en H se puede seleccionar $\lambda_c \gg 1$ tal la suma de los vectores no sea 0. Note que los vectores del segundo miembro de las desigualdades anteriores no son nulos, ya que, para el primer conjunto se tiene que, $f(x) - f(x^*) \geq \epsilon$ y para el segundo conjunto se tiene que $f(x) - f(x^*) \leq -\epsilon$. En ambos casos para un valor $\lambda_c > 0$ suficientemente grande no se anula la suma de los vectores de las desigualdades anteriores.

Para la función de tunelización clásica por el corolario 2.1 se tiene que el polo en x^* se mantiene con $\lambda_c \gg 1$ suficientemente grande.

En ambos casos la función T tiene un polo en x^* y no tiene puntos estacionarios en los conjuntos:

$$\begin{aligned} & \{x \in H \mid f(x) - f(x^*) \geq \epsilon\}, \\ & \{x \in H \mid f(x) - f(x^*) \leq -\epsilon\}. \end{aligned}$$

De los resultados anteriores se tienen propiedades generales⁶ de las funciones de tunelización:

- Los mínimos de la función f del mismo nivel que el último mínimo no “desaparecen” de la función de tunelización.
- La función de tunelización se suaviza tanto en $\{x \in H \mid f(x) - f(x^*) > 0\}$ como en $\{x \in H \mid f(x) - f(x^*) < 0\}$.
- Se tiene que $\forall x \in H$,

$$f(x) - f(x^*) \geq 0 \text{ si y solo si } T(x, x^*, \lambda^*) \geq 0,$$

$$f(x) - f(x^*) < 0 \text{ si y solo si } T(x, x^*, \lambda^*) < 0.$$

- El polo x^* no desaparece cuando se aumenta λ^* .

⁶Las propiedades de los métodos de tunelización se dan en la sección 2.3.

De lo expuesto anteriormente, se tiene que, las funciones de tunelización tienen propiedades generales que no son exclusivos de los factores que las componen. En la tabla 3.1 se dan todos los factores que se manejan en las funciones de tunelización clásica y exponencial.

En este sentido, de no particularizar en los factores que intervienen para crear a las funciones de tunelización, es que se amplió el método de tunelización clásico a la metodología de tunelización.

El problema generalizado por resolver durante la fase de tunelización es dados $x^* \in H$ y $\epsilon, 0 \leq |\epsilon| \ll 1$, se busca un punto $x^\circ \in H$ tal que

$$T(x^\circ, x^*, \lambda^*) \leq \epsilon \quad (3.10)$$

donde T es una función de tunelización.

Resulta así que, la metodología consiste en el diseño y selección de factores adecuados que permitan crear nuevas funciones de tunelización. En el diseño de la función de tunelización se deben de buscar los factores que generen un polo (definición 2.1) en x^* y que además resulten apropiados para el tipo de las funciones objetivo del problema 1.1.

Desde este punto de vista los factores de la función de tunelización exponencial son sólo nuevas formas de crear polos. Hay otras funciones de tunelización que son inmediatas de imaginar, por ejemplo, combinando los factores exponenciales y los factores de la función clásica: el factor clásico del polo móvil (ecuación 2.9) en lugar del factor exponencial del polo móvil (ecuación 3.4).

Esta forma de plantear la metodología de tunelización es nueva, anteriormente se mencionaba en los trabajos de V. Levy, Gómez, et al. la metodología en el sentido del método de tunelización clásico y las aplicaciones de éste al área de la optimización global (problema de mínimos cuadrados y problema de optimización global con restricciones y sin restricciones).

Para terminar esta sección damos el factor

$$\frac{1}{((x - x^*)^T B(x - x_{min}))^\lambda}$$

donde x^* es un punto mínimo de f , B es una matriz $n \times n$ definida positiva y λ es un número positivo y se demuestra que para un valor adecuado de λ se crea una nueva función de tunelización.

Teorema 3.9 Sean $f: \mathbb{R}^n \mapsto \mathbb{R}$ una función de clase C^2 y $x^* \in \mathbb{R}^n$ un punto mínimo de f que cumple las condiciones suficientes de optimalidad del teorema 1.3. Entonces la función

$$T_B(x, x^*, \lambda) = \frac{f(x) - f(x^*)}{\{(x - x^*)^T B(x - x_{min})\}^{2\lambda}}$$

tiene un polo en x^* para $\lambda > 0$ adecuado, es decir, cumple con la definición de polo (definición 2.1) en x^* .

Demostración. El resultado es inmediato, ya que, hay una equivalencia de normas en un espacio de dimensión finita. ■

Capítulo 4

Diseño y programación de los métodos de tunelización

La primera sección describe el estado del arte de los lenguajes de programación para algoritmos numéricos, la siguiente explica los puntos principales de un sistema aritmético en forma funcional, la que sigue contiene un algoritmo para obtener los parámetros básicos de un sistema aritmético de una computadora y las secciones restantes contienen una descripción detallada de los aspectos que se siguieron en la elaboración de los programas de los métodos de tunelización.

4.1 Lenguajes de programación para algoritmos numéricos

Es frecuente encontrar descripciones de como realizar o resolver numéricamente un problema en libros y revistas de análisis numérico o temas afines, y a estas descripciones informales se les llama algoritmos. Estudiando la situación actual de los llamados algoritmos numéricos encontramos varias deficiencias: el concepto de algoritmo es informal, la transcripción del algoritmo, o sea, programarlo en un código de computadora no garantiza que el funcionamiento sea tal como lo especifica el algoritmo y las herramientas de especificación formal de algoritmos numéricos no existen.

La descripción informal de los algoritmos numéricos es una práctica generalizada en revistas internacionales y libros especializados, y no tiene correspondencia con el modelo de un algoritmo formal (máquina de Turing, autómatas finitos o conjuntos regulares, etcétera).

Actualmente, no existen lenguajes formales que permitan especificar algoritmos numéricos y que faciliten su verificación o prueba, su modularidad, su reusabilidad y su traducción a un programa de computadora.

Los lenguajes más utilizados y más difundidos para especificar los métodos numéricos son FORTRAN, ALGOL y APL (C y Pascal son más recientes en esta área).

Ninguno de estos tiene cualidades sobresalientes como para que se puede usar para especificar formalmente un algoritmo numérico. La buena traducción de un algoritmo en cualquiera de estos lenguajes depende de la habilidad, de los conocimientos del área específica del método numérico y de la experiencia del programador.

Por otro lado, han surgido recientemente lenguajes formales para construir los archivos de

datos que requieren algunos de los programas de computadora de optimización, por ejemplo, *AMPL: A Mathematical Programming Language*¹. AMPL no es un lenguaje para especificar algoritmos, su función es traducir los datos de acuerdo a la formulación matemática de un problema de optimización al formato que requiere el programa de computadora correspondiente, y reciprocamente, traduce los resultados del programa de manera que su interpretación resulte más fácil.

Para equipos especiales, los llamados sistemas distribuidos (redes de computadoras) y las llamadas supercomputadoras, en los últimos años se han desarrollado para el área de métodos numéricos extensiones de los lenguajes C y FORTRAN, por ejemplo, DINO (*Distributed Numerically Oriented language*)² es una extensión a C⁺⁺.

En una línea similar (pero comercial) la compañía IBM realizó una nueva versión de FORTRAN³ que maneja los conceptos de operación escalar, de operación vectorial y de procesos paralelos. Junto con las adiciones al lenguaje se ofrece una biblioteca científica con los métodos numéricos más comunes (producto e inversión de matrices, solución de ecuaciones lineales, etcétera), así como la opción de que por medio del compilador, en forma casi automática, los programas escritos en versiones anteriores de FORTRAN al recompilarse usen las nuevas facilidades.

Otras nuevas opciones, que existen para apoyar el área de los métodos numéricos, son los programas de computadora, fáciles y prácticos de manejar, que proporcionan un laboratorio numérico para auxiliar en el desarrollo de prototipos y en la realización de experimentos numéricos, por ejemplo, MathCad y MatLab (ver el manual [Mat87]). Estos programas proporcionan un lenguaje, una biblioteca y la forma en que trabajan es interactiva, o sea, interpretan comandos y programas (como los interpretes del conocido lenguaje BASIC). En general, sus bibliotecas no usan en los subprogramas el concepto formal de modulo, más bien, los subprogramas son equivalentes a las subrutinas y a las funciones que se definen en el lenguaje Pascal.

Resumiendo, no es fácil abordar el tema de la especificación formal de los algoritmos numéricos, ya que, en general los algoritmos numéricos tienen muchos aspectos difíciles de formalizar, algunas áreas de los métodos numéricos son sumamente especializadas y otras involucran conceptos matemáticos difíciles de modelar mediante una computadora. Y aunque han habido nuevas aportaciones al área de los métodos numéricos, no existe, todavía, el LENGUAJE DE LOS MÉTODOS NUMÉRICO que facilite la especificación formal y modular de algoritmos numéricos. Las modas de los últimos años han puesto en boga los lenguajes Pascal, Modula y C, y muchas publicaciones recientes de métodos numéricos los utilizan, pero ninguno de estos o de los anteriores, tiene características sobresalientes como para ser considerado EL LENGUAJE DE LOS MÉTODOS NUMÉRICOS.

La decisión de usar algún lenguaje particular para programar métodos numéricos no es simple, en nuestro caso, se consideraron principalmente los siguientes puntos:

- El poder usar otros subprogramas.
- Las bibliotecas de rutinas disponibles.

¹ Ver el artículo [Fou87].

² Ver los artículos [Ros88b] y [Ros88a].

³ Ver el reporte técnico [Dub85].

- La transportabilidad entre cualquier tipo de equipos, microcomputadoras o computadoras.
- La posibilidad de correr ejemplos que usen gran cantidad de memoria principal.
- Las facilidades para realizar cálculos.
- La legibilidad del código.
- La estructuras de control y de datos.
- La facilidades de modularización.

FORTRAN es el lenguaje que se seleccionó, a pesar de que, no cumple satisfactoriamente los últimos tres puntos a los otros puntos restantes los satisface (sin mucho entusiasmo).

También se usó el programa MatLAB para la elaboración de las gráficas, para realizar pequeños experimentos numéricos y para diseñar la función del experimento 5.13 del capítulo 5⁴.

4.2 Descripción de un sistema aritmético

Las computadoras disponen de un sistema aritmético, ya sea en circuitos lógicos (procesador numérico) o en programas de computadora (software numérico) o en ambos, para realizar las operaciones aritméticas básicas: suma, resta, multiplicación y división, y algunas operaciones más complejas, generalmente se incluyen las funciones: trigonométricas, logarítmica, exponencial, raíz cuadrada, etcétera.

Las características de un sistema aritmético depende directamente del software y de las características de la computadora, por ejemplo, el compilador de lenguaje Pascal de Borland tiene versiones especiales para realizar las operaciones básicas con el coprocesador numérico o con programas de aritmética decimal, las computadoras VAX pueden manejar una tarjeta de punto flotante para realizar operaciones numéricas, la versión 2 de FORTRAN de IBM se diseñó para generar código que maneja el equipo Vectorial 3090 (IBM *Vector Facility*), etcétera.

Tomar demasiado en cuenta el equipo ocasiona que los programas se vuelvan dependientes e impide la transportabilidad de los programas.

Para escribir códigos transportables de métodos numéricos es indispensable conocer las bases teóricas de los sistemas aritméticos de computadora. Al respecto existen gran variedad de publicaciones, por ejemplo, los artículos [Kul86], [Mal72], [Mal74], etcétera y la gran mayoría de los libros de métodos numéricos le dedican los primeros capítulos (ver el libro [Dah74]).

Para un manejo adecuado de los sistemas aritméticos, independientemente de los equipo especiales y de los programas de cómputo que realizan las operaciones aritméticas, se describe en esta sección un modelo funcional de un sistema aritmético.

En nuestro enfoque funcional los sistemas aritméticos los descomponemos en dos partes: en un sistema de representación de números (sistema numérico) y en un sistema de cálculo de las operaciones aritméticas básicas (sistema aritmético).

⁴El apéndice A describe todos los recursos computacionales que se utilizaron para realizar este trabajo.

La forma en que se introduce un sistema aritmético es totalmente constructiva: primero, con el sistema numérico se definen sintácticamente los números y se les asocia un único significado (que obviamente es su valor numérico) y segundo, el sistema aritmético se describe en función del sistema numérico, de las operaciones básicas de los números reales y de un mapeo en el sistema numérico.

Las ventajas de este enfoque es que permiten definir algoritmos de cálculo bien definidos, ya que que representan funciones completas⁵.

Hay varios sistemas de representación de números, de hecho se pueden inventar una infinidad⁶ pero restringiéndose a los símbolos conocidos (dígitos arábigos y letras) los dos más conocidos son de punto fijo y de punto flotante. Los dos son tan populares que cualquier computadora o calculadora electrónica los usa internamente para representar números enteros (punto fijo) y números reales (punto flotante).

Definición 4.1 (sistema de punto fijo) Un sistema numérico de punto fijo o sistema de punto fijo se denota por

$$PF(b, m, n)$$

donde b es un número entero positivo denominado base del sistema, m es el número de dígitos y n , $0 \leq n \leq m$ es la posición del punto decimal.

Los números de $PF(b, m, n)$ se representan en el siguiente esquema

$$s d_{m-1} d_{m-2} \dots \bullet \dots d_0$$

donde s es el signo $\{+, -\}$; d_j , $j = 0, \dots, m-1$, son los dígitos, $d_j \in \{0, 1, \dots, b-1\}$; y \bullet es el punto decimal.

El significado de una cadena en el esquema anterior es el valor numérico calculado de la fórmula de punto fijo,

$$(d_{m-1-n} b^{m-1-n} + d_{m-2-n} b^{m-2-n} + \dots + d_0 b^0 - n)(s1)$$

donde el producto, la suma y la potencia son las operaciones de los números racionales. ■

Definición 4.2 (Sistema de punto flotante) Un sistema numérico de punto flotante o sistema de punto flotante se denota por

$$FL(b, m, n)$$

donde b es un número entero positivo denominado base del sistema, m es el número de dígitos de la mantisa y n es el número de dígitos del exponente.

Las cadenas de un sistema numérico de punto flotante tienen el siguiente esquema

$$s \bullet d_1 d_2 \dots d_m b^{s d_1' d_2' \dots d_n'}$$

donde s es el signo $\{+, -\}$, \bullet es el punto decimal, b es la base, $d_i, d_i' \in \{0, 1, \dots, b-1\}$, $i = 1, \dots, m$, $j = 1, \dots, n$, son los dígitos.

⁵Ver el libro [Man74].

⁶Ilans Freundenthal, ha diseñado un lenguaje de señales de radio para la comunicación Cómica, LINCOS, ver el libro [Fre80], suponiendo, entre otras cosas, que las propiedades de los números son universales.

Si se pide que $d_1 \neq 0$ y se denomina *sistema de punto flotante de números normalizados*. Salvo se indique lo contrario, siempre se considerará de números normalizados aunque no se mencione.

En el esquema anterior se llama mantisa a $s, d_1 d_2 \dots d_m$ y exponente a $s d_1^d d_2^d \dots d_n^d$.

El valor de la mantisa está dado por

$$m = (d_1 b^{-1} + d_2 b^{-2} + \dots + d_m b^{-m})(s1)$$

donde el producto, la suma y la potencia son las operaciones de los números racionales.

El exponente está en $PF(b, n, 0)$, por lo que su valor e está dado por la fórmula de punto fijo correspondiente.

Un número representado en este sistema tiene como significado el valor calculado de la fórmula de punto flotante,

$$m \cdot b^e$$

donde m es el valor de la mantisa, b es la base, e es el valor del exponente y las operaciones producto y potencia son tomadas de los números racionales. ■

Además de los sistemas numéricos para poder realizar cálculos numéricos se requiere de una función que mapee un subconjunto acotado de números reales \mathbb{R} sobre un sistema de representación particular, por ejemplo, $FL(b, m, n)$ con $b = 2$, $m = 5$ y $n = 3$. Más adelante se aclarará a que subconjunto de \mathbb{R} nos referimos propiamente.

Estas funciones transforman un número real x en una aproximación \tilde{x} de manera que el error de la representación,

$$|x - \tilde{x}|$$

esté controlado. Por el tipo de error que tienen estas funciones se dividen en dos clases: redondeo y truncación.

Antes de dar las funciones de truncación y de redondeo se introduce el siguiente esquema de representación de números reales,

$$s0.d_1 d_2 \dots d_i \dots b^{s d_1^d d_2^d \dots d_j^d \dots} \tag{4.1}$$

donde s es el signo $\{+, -\}$, $.$ es el punto decimal; b es la base; d_i y d_j^d , $i = 1, 2, \dots, j = 1, 2, \dots$, son dígitos, $d_i, d_j^d \in \{0, 1, \dots, b - 1\}$; y con la condición de que $d_1 \neq 0$ (número normalizado).

El esquema anterior sirve para representar a cualquier número real (para hacer única la representación no se permiten las terminaciones infinitas de $b - 1$, es decir, que a partir de algún k se tenga que $d_j = b - 1, j = k, k + 1, \dots$).

Definición 4.3 Sean $FL(b, m, n)$ un sistema de punto flotante y $PF(b, m, n)$ un sistema de punto fijo. Una función que mapea los reales

- en $FL(b, m, n)$ es de *truncación* si el error de las mantisas de x y su imagen \tilde{x} está acotado por b^{-m} para todas las representaciones posibles en él (ver la definición 4.4) y si además $\tilde{x} \leq x$ (note que corta la mantisa de un número en el esquema 4.1 de manera que se ocupen los m dígitos de la mantisa del esquema de $FL(b, m, n)$).
- en $PF(b, m, n)$ es de *truncación* si el error de x y su imagen \tilde{x} está acotado por b^{-n} para todas las representaciones posibles en él (ver la definición 4.4) y si además $\tilde{x} \leq x$ (note que corta los dígitos de un número de manera que se ocupen los m dígitos del esquema de $PF(b, m, n)$).

- en $FL(b, m, n)$ es de *redondeo* si el error de las mantisas de x y su imagen \tilde{x} está acotado por $\frac{1}{2}b^{-m}$ para todas las representaciones posibles en él (note que corta la mantisa de un número en el esquema 4.1 pero puede cambiar algunos dígitos de manera que cumpla con el error y se ocupen los m dígitos de la mantisa del esquema de $FL(b, m, n)$).
- en $PF(b, m, n)$ es de *redondeo* si el error de x y su imagen \tilde{x} está acotado por $\frac{1}{2}b^{-n}$ para todas las representaciones posibles en él (note que corta los dígitos de un número pudiendo cambiar el último dígito de manera que cumpla con el error y que se ocupen los m dígitos del esquema de $PF(b, m, n)$).

Como consecuencia de la definición anterior, para un sistema numérico dado, de punto flotante o de punto fijo, sólo existe una función de truncación, pero hay varias funciones de redondeo, por ejemplo, dado $x = 0.1234566666$ y el sistema $FL(10, 4, 2)$ se tiene que,

- $\tilde{x}^1 = 0.1234$ para la función de truncación y
- $\tilde{x}^2 = 0.1235$ o $\tilde{x}^3 = 0.1234$ son los dos posibles resultados de unas funciones de redondeo.

Un ejemplo de un algoritmo que define una función de redondeo para $b = 10$ se describe en el siguiente párrafo. Dado un número real

$$x = s0.d_1d_2 \dots d_j b^{s d_1 d_2 \dots d_n} \in D \subset \mathbb{R}$$

y un sistema de representación $FL(10, m, n+1)$. La función de redondeo $\tilde{\cdot} : D \mapsto FL(10, m, n+1)$ está dada por

Si $j \leq m$

$$\tilde{\cdot}(x) = \tilde{x} = s_0 d_1 d_2 \dots d_j b^{s d_1 d_2 \dots d_n}$$

en otro caso,

$$\tilde{\cdot}(x) = \tilde{x} = s_0 d_1 d_2 \dots d_{m-1} d_m b^{s d_1 d_2 \dots d_n}$$

donde

$$d_m = \begin{cases} d_m, & \text{si } 0 \leq d_{m+1} \leq 5; \\ 1 + d_m, & \text{en otro caso.} \end{cases}$$

1. Si $1 + d_m < b$ terminar; se tiene la representación.

2. Se hace $j = k$.

3. Se calcula $d_j = 0$ y $1 + d_{j-1}$.

Si $1 + d_{j-1} < b$ terminar; se tiene la representación.

En otro caso se calcula $j = j - 1$, Si $j > 1$ se repite este paso, en otro caso se continúa en el siguiente paso.

4. Se suma 1 al exponente, se hace $d_1 = 1$ y se termina.

El algoritmo anterior es formal, hay garantía de que siempre termina y sus pasos están bien definidos (no hay forma de manejar una mantisa infinita, por ejemplo, la de $\sqrt{2}$).

El algoritmo anterior resuelve de una manera explícita el problema de que se pueda realizar la representación en $FL(10, m, n + 1)$, ya que sólo permite números en $D \subset \mathbb{R}$,

$$D = (-10^{9n+1}, +10^{9n+1}).$$

La razón de tal restricción se debe a que un sistema numérico (de punto fijo o de punto flotante) sólo puede representar un número finito de racionales dentro de un intervalo acotado de \mathbb{R} .

Definición 4.4 Sea $x \in \mathbb{R}$ un número real, $FL(b, m, n)$ un sistema de punto flotante y $PF(b, m, n)$ un sistema de punto fijo. x tiene una *representación en un sistema numérico*

- $FL(b, m, n)$ si su imagen bajo la función (de redondeo o truncación) está en el intervalo

$$D = [-0.(b-1)^m b^{(b-1)^n}, 0.(b-1)^m b^{(b-1)^n}].$$

- $PF(b, m, n)$ si su imagen bajo la función (de redondeo o truncación) está en el intervalo

$$D = \left\{ -\overbrace{(b-1) \cdots (b-1)}^m \cdot \underbrace{(b-1) \cdots (b-1)}_n \right\}.$$

La intersección de las representaciones de $FL(b, m, n)$ o $PF(b, m, n)$ con el intervalo correspondiente forman el *conjunto de las representaciones* de ellos. ■

Teorema 4.1 La cardinalidad del conjunto de las representaciones de un sistema numérico $FL(b, m, n)$ es $4 \times (b-1) \times b^{n+m-1}$ y para un sistema numérico $PF(b, m, n)$ es $2 \times b^m$.

Demostración. El número de mantisas es $2 \times (b-1) \times b^{m-1}$ y el número de exponentes es $2 \times b^n$. ■

De lo anterior es claro que las funciones de redondeo o de truncación sólo están definidas para una parte de los números reales, por lo que, cuando se trata de representar un número y , $|y| \gg 0$ ($y \notin D$), al igual que los sistemas aritméticos de computadora, utilizaremos un símbolo especial llamado *overflow* ∞ . Desgraciadamente, para la gran mayoría de las computadoras el símbolo anterior significa parar la ejecución del programa, ya que para él no están definidas las operaciones aritméticas.

En nuestro caso este símbolo no detiene la ejecución, se añade para definir sin ninguna excepción a las operaciones aritméticas.

Para construir funcionalmente un sistema aritmético sobre un sistema numérico, las definiciones de las operaciones básicas (+, -, ×, /) usan la composición de funciones y las operaciones aritméticas de los números reales.

Usando todo lo que hemos expuesto, la siguiente definición es un modelo funcional con el cual el comportamiento de un sistema aritmético es completo.

Definición 4.5 Sea SN un sistema numérico de punto flotante $FL(b, m, n)$ o de punto fijo $PF(b, m, n)$, $D \subset \mathbb{R}$ es el subconjunto apropiado para SN y $\bar{\cdot} : D \rightarrow SN \cup \{\infty\}$ una función de redondeo o de truncación donde $\bar{\cdot}$ se extiende a todos los números reales de la siguiente forma: dado $x \in \mathbb{R}$, si x no se puede representar en SN entonces $\bar{\cdot}(x) = \infty$, en otro caso, $\bar{\cdot}(x) = \bar{x}$.

Dados $x, y \in SN$ las operaciones aritméticas se definen como sigue:

- La suma $\oplus : SN \times SN \mapsto SN \cup \{\infty\}$ es $\oplus(x, y) = \overline{\tau}(x + y)$.
- La resta $\ominus : SN \times SN \mapsto SN \cup \{\infty\}$ es $\ominus(x, y) = \overline{\tau}(x - y)$.
- El producto $\otimes : SN \times SN \mapsto SN \cup \{\infty\}$ es $\otimes(x, y) = \overline{\tau}(x \times y)$.
- La división $\oslash : SN \times SN \mapsto SN \cup \{\infty\}$ si $y \neq 0$, $\oslash(x, y) = \overline{\tau}(\frac{x}{y})$, en otro caso ∞ .

La notación de operación binaria se define,

$$x \odot y = \odot(x, y)$$

donde \odot puede ser $\oplus, \ominus, \otimes, \oslash$.

A un sistema aritmético definido sobre el sistema de Representación SN con la función de redondeo $\overline{\tau}$ y las operaciones: \oplus, \ominus, \otimes y \oslash , se le denota por

$$(SN, \overline{\tau}, +, -, \times, /).$$

Un elemento importante de un SN es el llamado ϵ del sistema aritmético.

Teorema 4.2 Sea $(FL(b, m, n), \overline{\tau}, +, -, \times, /)$ un sistema aritmético con $m < b^{(b-1)^n}$.

Entonces $\exists \epsilon \in FL(b, m, n)$, $\epsilon \neq 0$ tal que

$$1 \oplus \epsilon = \epsilon \oplus 1 = 1.$$

Demostración. La condición sobre el tamaño de la mantisa garantiza la existencia y la conmutación se tiene de la conmutabilidad de la operación suma de los números reales. ■

Note que para un sistema aritmético sobre un sistema numérico de punto fijo no es posible encontrar números distintos de cero que cumplan la afirmación del teorema anterior, en este caso, el ϵ del sistema no es calculable.

Definición 4.6 Dado un sistema aritmético $(FL(b, m, n), \overline{\tau}, +, -, \times, /)$ se define ϵ , el ϵ del sistema aritmético, como

$$\epsilon = \max\{ \epsilon \in FL(b, m, n) \mid \epsilon \neq 0, 1 \oplus \epsilon = 1 \}.$$

Por el teorema anterior, es claro que, el ϵ es calculable sólo cuando se trata de un sistema aritmético sobre un sistema de punto flotante.

La utilidad del ϵ se da en el siguiente teorema.

Teorema 4.3 Dado un sistema aritmético $(FL(b, m, n), \overline{\tau}, +, -, \times, /)$, $x, y \in (FL(b, m, n)$, $x >> y$, $x \neq 0$.

- Si $|\frac{x}{y}| \leq \epsilon$ entonces $x = x \oplus y$
- Si $|\frac{x}{y}| > \epsilon$ entonces $x \neq x \oplus y$

Demostración. Es inmediata de la definición de ϵ . ■

4.3 Cálculo de los parámetros de un sistema aritmético

Es útil conocer los parámetros del sistema numérico de la computadora con que se trabaja, esto es la base, el tamaño de la mantisa y si trunca o redondea. En los artículos de Michael Malcon, [Mal72] y [Mal74], se dan algoritmos para calcularlos.

Los usos del epsilon del sistema aritmético, ver el teorema 4.3, son

- el error relativo pedido en el cálculo de un resultado distinto de cero debe ser mayor o igual a ϵ ;
- los cálculos iterativos de la forma

$$x + y_n$$

donde $x \neq 0$ y $y_n \rightarrow 0$, cuando $n \rightarrow \infty$ se deben de detener en el momento que se tenga $|y_n| \leq \epsilon|x|$.

Note que el primer punto no impide que el resultado tenga un error menor a ϵ , pero si evita que se pidan tolerancias menores a las que soporta el sistema aritmético. El segundo evita que se realicen iteraciones y cálculos que no mejoran el resultado.

En esta sección se describe un algoritmo que calcula aproximaciones al epsilon, al menor y al mayor de los números de un sistema aritmético ($FL(b, m, n), \bar{r}, +, -, \times, /$). En adelante siempre que se hable de las operaciones $\oplus, \ominus, \otimes, \oslash$ de un sistema aritmético se usaran por comodidad los símbolos aritméticos usuales ($+, -, \times, /$).

Definición 4.7 Una variable numérica en SN se representa por un símbolo, por ejemplo, a y se representa $a \in SN$ y esto significa que los valores de a son de SN o ∞ .

Note que el sentido de nuestra definición es que a toma valores numéricos que son una representación numérica de SN o bien se trata del símbolo ∞ (overflow).

Un algoritmo para obtener aproximadamente ϵ es el siguiente.

Sea $(FL(b, m, n), \bar{r}, +, -, \times, /)$ un sistema aritmético y $\epsilon, \epsilon, a \in FL(b, m, n)$, $m < b^{(b-1)^n}$ unas de variables.

1. Se hace $\epsilon = \frac{1}{2}$.
2. Se calcula $a = 1 + \epsilon$. Si $a = 1$ se termina; se tiene una aproximación al epsilon $\epsilon \approx \epsilon$. En otro caso, se calcula $\epsilon = \frac{\epsilon}{2}$ y se repite este paso.

El algoritmo anterior está bien definido y la garantía de paro esta dada por la condición sobre la mantisa.

La operación $a = 1 + \epsilon$ no está demás, se incluye para evitar que la operación se realice en dispositivos especiales (acumuladores o registros aritméticos) y que por sus propiedades cambien el significado del algoritmo al transcribirlo a un lenguaje de programación.

De hecho se comprobó que el significado del algoritmo anterior no cambió. Se escribieron varios códigos en los lenguajes de programación (FORTRAN, PASCAL y C), se ejecutaron en diversos equipos de cómputo (con coprocesador y sin él) y siempre se obtuvo una buena aproximación al epsilon del sistema aritmético del equipo de cómputo.

Para obtener las aproximaciones de los números positivos mayor y menor de un sistema aritmético, utilizamos que los exponentes se representan en un sistema de punto fijo.

Teorema 4.4 Sea $PF(b, m, n)$ un sistema de punto fijo. Si r es el número positivo mayor entonces $[-r, r]$ es el intervalo de los números representables.

Demostración. El resultado se obtiene de la definición de un sistema de punto fijo. ■

Aprovechando lo anterior, se puede definir perfectamente un algoritmo que calcule el número positivo representable más pequeño m . El mayor se obtiene de

$$M = \frac{1}{m}.$$

El algoritmo tiene los siguientes pasos. Dados $(FL(b, m, n), \bar{\cdot}, +, -, \times, /)$ y $m, M, v_0, v_1 \in FL(b, m, n)$.

1. Se hace $v_0 = 1$.

2. Se hace $v_1 = v_0$ y $v_0 = \frac{v_0}{2}$.

Si $v_0 = 0$ se continúa en el paso 3, en otro caso, se repite este paso.

3. Se calcula $m = v_1$ y $M = \frac{1}{m}$.

El funcionamiento del algoritmo anterior se basa en que en la variable v_1 se guardan los resultados de la división que son cada vez más pequeños y termina porque llega un momento en que la división debe de ser cero después de redondearla o de truncaarla.

Se elaboraron cuatro códigos basados en el algoritmo anterior para los lenguajes FORTRAN, Pascal, C y MatLab. El código en FORTRAN funcionó con coprocesador y sin él para el Compilador WatFort-77 V1.4. Los otros códigos fallaron por (*overflow*) en todos los casos.

La razón de la falla es que los exponentes de la representación en punto flotante de los compiladores y programas: PC-MatLab V3.05, RMFort V2.11 con coprocesador y sin él, Turbo Pascal V3.02A para coprocesador, Turbo Pascal V3.02A sin coprocesador, Turbo Pascal V4.0 con coprocesador y sin él, y Turbo C V2.0 con coprocesador y sin él, no cumplen los supuestos del teorema 4.4 o sea que los exponentes no están centrados en un intervalo de un sistema de punto fijo.

Modificando la forma de seleccionar y generar el valor más pequeño se construyó otro algoritmo que aproxima al número positivo más pequeño y al número positivo más grande.

Dadas las variables $m, M, v_0, v_1, v_2 \in FL(b, m, n)$.

1. Se hace $v_0 = 0.5$ y $v_1 = 1.0$

2. Se hace $v_2 = v_1$, $v_1 = v_0$ y $v_0 = v_0 \cdot v_0 \cdot v_0$. Si $v_0 = 0$ se continúa en el paso 3, en otro caso, se repite este paso.

3. Se calcula $m = v_2$ y $M = \frac{1}{m}$.

Los códigos basados en este algoritmo funcionaron perfectamente para los compiladores y programas mencionados en el párrafo anterior (usando coprocesador o sin usarlo).

4.4 Manejo de números grandes

Los métodos de tunelización generan números muy grandes en las cercanías de los polos, tales cálculos, cuando no son controlados pueden detener la ejecución por generar un *overflow*.

Con el modelo de los sistemas aritméticos de la sección anterior se obtuvieron las siguientes reglas prácticas para evitar que en el cálculo de las funciones de tunelización ocurra un *overflow*.

- Escalar de manera que los cálculos se realicen dentro del rango del sistema numérico.
- Utilizar el inverso multiplicativo cuando se realicen cálculos que puedan generar valores positivos o negativos muy grandes.

Para ejemplificar el uso de las reglas, se muestra en el siguiente teorema un algoritmo para evaluar el factor exponencial

$$\exp\left(\frac{\lambda}{d^2}\right).$$

Teorema 4.5 *Dado un sistema aritmético sobre un sistema de punto flotante, sea M una aproximación al número positivo más grande, sea $\exp(\cdot)$ la función exponencial y sea $\ln(\cdot)$ la función logaritmo natural.*

Entonces el siguiente algoritmo está bien definido y no termina en overflow.

1. Sea $d^2 \in SN$

2. Se calcula $a = \frac{d^2}{\lambda}$.

Si $a \leq \epsilon_1$ entonces se hace $r = M$, en otro caso, se calcula $r = \exp\left(\frac{1}{a}\right)$ donde $\epsilon_1 = \frac{1}{\ln(M)}$.

Demostración. La afirmación se tiene de lo siguientes argumentos

- El número ϵ_1 es calculable y no es cero, ya que, es mayor al número positivo más pequeño y menor al número positivo más grande.
- El cálculo de la función exponencial está condicionado a que

$$\frac{\lambda}{d^2} > \epsilon_1$$

de donde

$$M > \exp\left(\frac{\lambda}{\|x - x^*\|^2}\right),$$

o sea, el factor exponencial no sobrepasa el rango del sistema numérico. ■

4.5 Detalles del diseño y programación de los métodos de tunelización

Los códigos de los métodos de tunelización, original, clásico y exponencial, se elaboraron en el lenguaje FORTRAN.

El programa del método original elaborado por Antonio Montalvo (ver su tesis doctoral [Mon79]) casi no se modificó en su estructura. Se le corrigieron errores en la programación que ocasionaban que no terminara en ciertos casos y se le incluyeron las rutinas: CONMIN y RANDOM.

La rutina CONMIN, que se usó para resolver en forma local el problema 1.1, es del tipo de métrica variable (BFGS) y del tipo gradiente conjugado⁷, y fue escrita por D.F. Shano y K.H. Phua⁸. La rutina CONMIN original es para resolver el problema de optimización local sin restricciones, la copia que se tiene en la biblioteca del departamento de métodos Matemáticos y Numéricos es una versión adaptada, probada y documentada por el Matemático Apolinar Calderón Segura. Para este trabajo fue modificada para resolver el problema de optimización local dentro de un hipercubo.

Las versiones de los programas de los métodos clásico y exponencial se diseñaron modularmente. En el diseño de los módulos se consideraron los siguientes aspectos:

- el objetivo del módulo bien definido, es decir, o termina con los resultados de los cálculos o comunica por medio de una variable que los cálculos que no se pudieron realizar;
- se usan las reglas dadas en la sección 4.4 para evitar el *overflow*.
- la definición de los ciclos es legible y fueron revisados cuidadosamente para evitar iterar indefinidamente, y
- las variables se dividieron en dos clases: de comunicación (para las entradas y las salidas del módulo) e internas (para uso exclusivo del módulo).

Otros factores que influyeron el diseño son la transportabilidad, las facilidades para la introducción de la función objetivo y de su gradiente, y la facilidad de poder usar otras rutinas, por ejemplo, de optimización local.

Para hacer transportables los códigos se usó el lenguaje FORTRAN en la escritura de los módulos. Los programas resultaron totalmente compatibles con los compiladores de FORTRAN de los equipos: IBM43xx, Cyber, Burroughs, Vax y PC Compatibles.

Por otra parte, se ha comprobado experimentalmente que no es necesario adaptar los códigos para los parámetros de los sistemas aritméticos de los distintos equipos de cómputo, ya que, gracias a la rutina EPSMAQ, basada en los algoritmos de las secciones 4.3 y 4.4, los códigos se adaptan automáticamente al sistema aritmético del equipo de cómputo.

La introducción del problema se realiza a dos niveles: con una rutina FORTRAN con el subprograma de la función y su gradiente (la rutina del gradiente es opcional), y con dos archivos uno con los datos y otro con los parámetros de la corrida.

⁷Mayor información de los métodos de optimización local de la rutina CONMIN se da en las secciones 1.3.1 y 1.3.2.

⁸Ver el artículo [Sha78].

La introducción de la rutina no modifica ni requiere alterar al resto de los módulos, salvo en el caso de que la dimensión del problema requiera de un mayor número de variables.

Los archivos de datos y de parámetros de la corrida se puede escribir directamente usando un programa de edición de textos en código ASCII o se pueden generar mediante el programa TUNDAT.

El programa TUNDAT es un programa que funciona amigablemente, realiza una serie de preguntas acerca del problema y de acuerdo a las respuestas escribe los archivos de datos y de parámetros, todo la información acerca del programa TUNDAT se encuentra en el manual de referencia de los programas de tunelización, manual [Bar91].

El poder usar otras rutinas fue una gran ventaja, ya que, de otra forma la programación de todo el método no se hubiera realizado tan rápido. En particular, se utilizó una versión de la rutina CONMIN (ver el artículo [Sha78]), las modificaciones que le hicieron son: manejo de regiones de factibilidad con la forma de un hiper cubo y que termine cuando los cambios del valor de la función en forma consecutiva sean muy pequeños⁹.

Los programas del método clásico y exponencial son: CLASICO y EXPONEN, respectivamente. Las diferentes rutinas que utilizan se muestran en el diagrama de la figura 4.1.

De las rutinas mostradas en la figura 4.1 las rutinas que cambian según el método son: TUNNEW, MOVIL, TUNELO y TUNELX.

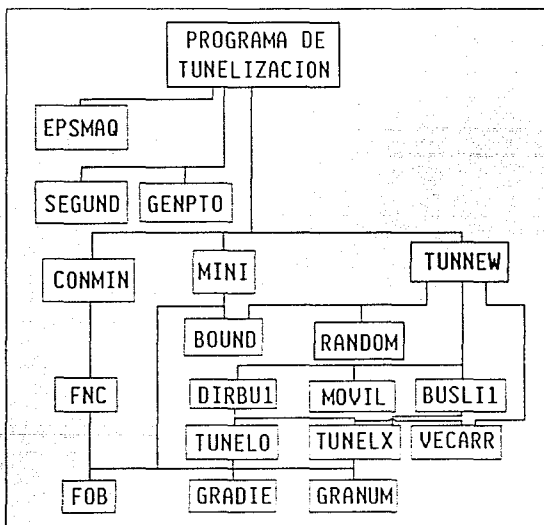


Figura 4.1 Diagrama de bloques de los programas de tunelización

Las rutinas GENPTO y la rutina GRANUM no tienen nada que ver con los métodos de

⁹El que los valores de la función cambien muy poco no significa necesariamente que se encontró un mínimo local, sin embargo, para funciones muy planas o con apéndice muy estrecho este criterio evita que se realicen iteraciones de más.

tunelización, la primera se usa para generar automáticamente los puntos de arranque y la segunda para calcular numéricamente el gradiente.

El funcionamiento de los programas CLASICO y EXPONEN es muy parecido, los pasos en los que son diferentes se marcan con el nombre del programa entre paréntesis; los pasos que realizan los programas son los siguientes.

1. Lectura y validación de los parámetros. En caso de error se escribe el mensaje adecuado y se detiene la ejecución.
2. Lectura y validación de los datos del problema. En caso de error se escribe el mensaje adecuado y se detiene la ejecución.
3. Cálculo de la función en el punto de arranque dado por el usuario.

4. Se llama a la rutina de minimización CONMIN.

Si la rutina CONMIN termina con éxito y si el valor de la función disminuyó se continúa en el siguiente paso. En otro caso, se llama a la rutina MINI.

(EXPONEN) Se continúa en el siguiente paso.

(CLASICO) Si falló la rutina MINI se escribe un mensaje y se termina el proceso, en otro caso se continúa en el siguiente paso.

5. Se guarda la solución de la fase de minimización y se llama a la rutina TUNNEW para realizar la fase de tunelización.

Si la rutina no encontró una solución del problema de tunelización clásico o exponencial (problema 2.7 o problema 3.2 respectivamente), se continúa en el paso 6.

En otro caso se repite el proceso a partir del paso 4.

6. Se escriben los resultados obtenidos.

Observaciones del algoritmo anterior.

- Para el programa CLASICO. Se debe tener en cuenta que no siempre que falle este programa en la fase de tunelización es por que se tiene la solución del problema 1.1. En dos casos hay que tener cuidado, cuando aparece el mensaje de que no pudo calcular la potencia del polo y cuando aparece el mensaje de que fallo la fase de minimización. En ambos casos es probable que el último punto encontrado en la fase de minimización no sea un punto mínimo local. Si no aparecen estos mensajes es muy probable que se termine en un PMG.
- Las rutinas de minimización: CONMIN y MINI, indican por medio de una variable la condición con que terminan. Se marcan con un 0 los puntos que tienen gradiente casi nulo y con otro valor los que no. En la descripción de CONMIN y MINI se describen los números de condición con que terminan estas rutinas.

4.6 Rutinas de los programas: CLASICO y EXPONEN

En esta sección se describen las rutinas de los programas de tunelización: CLASICO y EXPONEN.

Para mayor información de como ejecutar los programas de tunelización, de como modificarlos para resolver problemas muy grandes, de como introducir la función del problema y de como construir el archivo de parámetros consultar el manual [Bar91].

Las rutinas de los métodos de tunelización son:

Rutina CONMIN.

Objetivo: Minimizar localmente una función real sobre un hipercubo o bien sobre todo \mathbb{R}^n . Los métodos que utiliza son métrica variable (BFGS) y gradiente conjugado.

La rutina avisa por medio de una variable la condición con que termina. Las números de condición y su significado son:

$$0 \quad \|\nabla f(x^k)\| \leq \epsilon.$$

1 Se sobrepaso el número de evaluaciones.

2 La búsqueda de línea fallo.

3 El vector de búsqueda no fue de descenso.

4 Los valores de la función en 4 iteraciones han sido muy pequeños.

Los números de condición pueden tener las siguientes causas.

- La condición 2 puede ser ocasionada porque la función esté mal codificada.
- La condición 3 puede ser ocasionada por problemas de redondeo o bien porque la tolerancia de la solución de minimización ϵ_m es demasiado pequeña.
- Las condición 1 es ocasionada cuando el número de evaluaciones es pequeño respecto a la dimensión del problema o puede ser ocasionada por usar BFGS en problemas con función objetivo de muy complicada evaluación.
- Las condiciones 0 y 4 se consideran éxito, aunque no necesariamente signifique que se encontró un punto mínimo local.

El número de variables de memoria que se requieren según el método de minimización local que se seleccione se calcula como sigue

$$\begin{array}{ll} \text{gradiente conjugado} & 5 * n + 2, \\ \text{métrica variable} & n * (n + 7) / 2, \end{array}$$

donde n es la dimensión del dominio de la función objetivo.

La rutina FCN es la que introduce la función objetivo y su gradiente.

Con los programas de tunelización no requiere modificarse el llamado de la rutina CONMIN y la rutina FCN ya está incluida.

El código es una versión ligeramente modificada de la que aparece en el artículo [Sha78].

Rutina TUNNEW.

Objetivo: Esta rutina resuelve el problema de tunelización (ver las ecuaciones 2.7 o 3.2)

que consisten en la búsqueda de un punto para el cual la función de tunelización es menor ϵ (número pequeño positivo o negativo).

La programación se basa en los algoritmos expuestos en las secciones 2.4.2 y 3.2.1 para el método clásico y para el método exponencial. Debido a que los algoritmos son ligeramente diferentes se considero adecuado realizar dos rutinas una para el método clásico y otra para el método exponencial. Las diferencias se describen en la sección 3.4.

La rutina del método exponencial termina:

- porque encontró la solución del problema de tunelización o
- porque no la encontró.

El segundo caso lo indica asignando el valor 999999999 en la variable FLMO del bloque SALID. En otro caso la variable FLMO no cambia y el valor de la solución del problema se da en la variable FMIN y en el arreglo de variables XINIT. Por medio de la variable IBPRIN se le indica a la rutina que imprima los resultados parciales que se vayan generando.

La rutina del método clásico termina como la anterior, pero además puede escribir el mensaje de error de que no pudo calcular la potencia del polo.

Las rutinas que utilizan, la versión clásica y la exponencial, son: GRANUM, BOUND, RANDOM, DIRBU1, VECARR, MOVIL, BUSLI1, TUNELO, TUNELX, FOB y GRADIE.

Observaciones:

- La rutina DIRBU1 construye una dirección de búsqueda de acuerdo a las fórmulas 2.8 y 3.3.
- La rutina BUSLI1 realiza una búsqueda unidimensional usando el método de la bisección en la dirección construida por la rutina anterior.
Estas rutinas se pueden reemplazar por otras similares como por ejemplo una búsqueda unidimensional por interpolación cuadrática. Las variantes que se pueden desarrollar de estas rutinas para mejorar el funcionamiento de la rutina TUNNEW son muy amplias. No se incluyen los resultados obtenidos usando otro tipo de dirección de búsqueda o búsqueda unidimensional en este trabajo.
- Las rutinas TUNELO y TUNELX contienen los códigos de las funciones de tunelización clásica y exponencial, y de sus gradientes.
- La rutina MOVIL varia de acuerdo con el método de tunelización (ver la sección 3.4).
- La rutina RANDOM es una rutina que genera números aleatorios y se usa para generar aleatoriamente los puntos de arranque del polo móvil y los puntos de arranque aleatorios de la fase de tunelización (estos puntos se generan después de haber generado 2n puntos de arranque alrededor del último punto mínimo en la dirección de los ejes de coordenadas).
- La rutina BOUND se usa para transformar la dirección de búsqueda en una dirección de búsqueda factible (ver la definición 1.7).
- Las rutinas FOB y GRADIE contienen la codificación de la función del problema y de su gradiente.

- La rutina GRANUM se usa para aproximar numéricamente el gradiente.

Salvo por la posibilidad de cambiar DIRBU1 y BUSLI1, el usuario no puede modificar el funcionamiento de TUNNEW.

Rutina MINI.

Objetivo: Minimizar una función por el algoritmo de Fletcher-Reeves¹⁰. Se usa cuando la rutina CONMIN falla o bien cuando el valor del mínimo encontrado por CONMIN es mayor que el valor de la función de la última fase de tunelización.

La rutina avisa por medio de una variable la condición con que termina. Los números de condición y su significado son

$$0 \quad \|\nabla f(x^k)\| \leq \epsilon.$$

- 1 Se sobrepasa el número de evaluaciones de la búsqueda unidimensional.
- 4 Los valores de la función en 4 iteraciones han sido muy pequeños.

Las causas probables de los números de condición son:

- La condición 1 ocurre cuando el número de evaluaciones es pequeño respecto a la dimensión del problema, cuando el problema tiene respecto a la dirección del último gradiente una especie de apéndice y cuando el ángulo del gradiente respecto a la dirección del punto mínimo es oblicua ocasionando que no se pueda determinar con mayor precisión el valor mínimo. Para evitar esta condición se debe de dar un número grande para las evaluaciones de la bisección. Generalmente al hacer esto se obtiene el número de condición 4 y en el mejor de los casos el número de condición 0.
- Las condiciones 0 y 4 se consideran éxito, aunque no necesariamente signifique que se encontró un mínimo local.

Las rutinas que utiliza son: BOUND, FOB, GRADIE o GRANUM.

La rutina BOUND sirve para que la dirección del gradiente se convierta en una dirección factible. Las rutinas FOB y GRADIE contienen el código de la función objetivo y su gradiente respectivamente. La rutina GRANUM calcula numéricamente una aproximación del gradiente.

Rutina EPSMAQ.

Objetivo: Calcular el épsilon y una aproximación al mayor número positivo del sistema aritmético del equipo donde corran los programas. Su programación se basa en los algoritmos de las páginas 69 y 70.

El uso de la rutina es opcional, por medio de los parámetros de entrada los programas de tunelización pueden llamarla o bien usar valores de default.

Esta rutina evita que el usuario adapte los programas para los distintos sistemas aritméticos de sus equipos, sin embargo, debe tener en cuenta dos cosas:

- primero, el uso de la rutina puede variar el número de iteraciones al resolver el mismo problema en dos computadoras distintas y
- segundo, el uso de los valores de default puede generar un error de overflow si el sistema aritmético del computador resulta tener menor rango de números reales.

¹⁰Para información del algoritmo de Fletcher-Reeves ver la sección 1.3.1.

Los programas de tunelización reportan los valores de ϵ y del mayor número positivo como parte de los resultados que generan.

Rutina GRANUM.

Objetivo: Calcular numéricamente el vector de la función gradiente de la función objetivo. Se incluye para que el usuario no tenga forzosamente que codificar la función gradiente. El usuario se encarga de indicar que se use esta rutina en lugar de la rutina GRADIE.

La rutina GRADIE, que corresponde a la función gradiente de la función objetivo, es opcional y debe codificarla el usuario.

Rutina DIRBU1.

Objetivo: Construir la dirección de búsqueda (ver la página 15) usando las fórmulas 2.8 y 3.3 según se trate de la función de tunelización clásica o de la exponencial.

Las rutinas que utiliza son: VECARR, TUNELO y TUNELX.

Esta rutina siempre construye una dirección de búsqueda; en el caso de que el punto resulte ser un punto estacionario, la rutina activa un polo móvil o bien incrementa la potencia del polo móvil.

La rutina se llama cuando en los parámetros de la corrida se da el valor de 1, ya que con el valor 2 el programa llama a la rutina DIRBU2.

La rutina DIRBU2 genera una dirección de búsqueda usando una combinación lineal de la última dirección y la de las fórmulas 2.8 y 3.3. Esta rutina es una nueva opción que investigamos al final de este trabajo y no se incluyen los resultados obtenidos (en los pocos experimentos que se realizaron no se notó que disminuyeran las iteraciones de la fase de tunelización).

Rutina MOVIL.

Objetivo: Calcular la posición del polo móvil. El polo móvil se sitúa en la recta que une al último punto de la fase de tunelización y el polo del último punto mínimo.

La distancia del polo móvil cambia según se trate de la función de tunelización clásica o de la exponencial (ver la sección 3.4).

Rutina VECARR.

Objetivo: Generar un nuevo punto de arranque cuando se encuentra un punto estacionario de la función de tunelización.

De igual manera que la rutina MOVIL esta rutina tiene dos versiones una para el método clásico y otra para el método exponencial (ver la sección 3.4).

La heurística que utiliza en sus dos versiones es similar en que la posición del nuevo punto de arranque es aleatoria.

Un estudio de la distancia a la que se debe de poner un nuevo punto de arranque para el método clásico se da en la tesis de Alfredo Cortés, [Cor85].

La heurística del punto de arranque se reporta en el trabajo de Gómez, et.al, [Gom87b] y [Gom87a].

Rutina GENPTO.

Objetivo: Generar puntos iniciales factibles distribuidos uniformemente dentro de la región H .

Esta rutina se usó para generar los puntos de algunos de los experimentos que se realizaron, su uso es opcional.

El usuario puede resolver su problema sin usar esta rutina incluyendo las coordenadas del punto de inicio en el archivo de datos del problema.

Rutina BUSLI1.

Objetivo: Realizar la búsqueda unidimensional usando el método de la Bisección.

Las rutinas que utiliza son: TUNELO y TUNELX.

La rutina requiere como parámetro el número de veces que puede biseccionar en la dirección de búsqueda hasta lograr que la función de tunelización (clásica o exponencial) disminuya.

Cuando no disminuye el valor de la función de tunelización la rutina lo informa por medio de una variable que toma el valor de 1 y el valor 0 en caso contrario.

Existen otras dos rutinas de búsqueda unidimensional: BUSLI2 y BUSLI3. Las rutinas BUSLI2 y BUSLI3 realizan una búsqueda unidimensional de tipo cuadrática y de tipo cúbica respectivamente. Estas rutinas se incorporaron para mejorar el funcionamiento de la fase de tunelización (disminuir el número de iteraciones). No incluimos en este trabajo los resultados del uso de estas rutinas, ya que, junto con la rutina DIRBU2 son parte de una investigación futura. Algunos experimentos que se realizaron mostraron que la rutina BUSLI1 funcionaba mejor que las otras dos.

Los parámetros del número de bisecciones y de la selección de la rutina de búsqueda unidimensional se dan en el archivo de parámetros.

Rutina RANDOM.

Objetivo: Calcular números aleatorios. Su código se basa en el artículo [Chi89].

Esta rutina es un generador uniforme de números aleatorios en el intervalo (0, 1). Sus características son adecuadas para computadoras que manejan los números enteros en 16 bits.

Esta rutina se puede sustituir por otra similar, por ejemplo, la que trae la biblioteca estandar de FORTRAN, sin embargo, se prefirió usar ésta ya que las pruebas estadísticas que le aplicamos demostraron que las series de números que genera tienen una distribución uniforme, no se estacionan, no se repiten (tienen un ciclo grande) y la serie se puede volver a obtener a partir de la misma semilla.

Su transportabilidad a equipos grandes está garantizada por el tamaño de número entero que maneja. El tamaño del ciclo de la serie es de alrededor de 2^{30} .

Rutina BOUND.

Objetivo: Calcular el factor y la dirección para transformar la dirección de la rutina DIRBU1 en una dirección factible.

Esta rutina se diseñó para evitar salirse de la región factible durante la fase de tunelización y durante la fase de minimización de la rutina MINI (ver la definición 1.21 de la función P_H).

Rutina FOB.

Objetivo: Calcular el valor de la función objetivo.

El usuario se encarga de codificar esta función.

Rutina GRADIE.

Objetivo: Calcular el gradiente de la función objetivo.

El usuario se encarga de codificar esta rutina.

En la codificación de estas dos últimas rutinas se sugiere tomar en cuenta lo expuesto en las secciones 4.2, 4.3 y 4.4. Un ejemplo de una corrida y de como codificar e incluir estas dos últimas rutinas en los programas de tunelización se muestra en el manual [Bar91].

Capítulo 5

Experimentos numéricos

En este capítulo se dan los lineamientos que se usaron para el diseño de los experimentos y para la evaluación y comparación de los métodos. La segunda sección describe las características de las funciones de los experimentos, la siguiente da los resultados de la sensibilidad del parámetro λ^* del método de tunelización exponencial. Las secciones finales son un resumen comparativo de los resultados de los experimentos con los tres métodos y con el método de las funciones de relleno.

5.1 Una forma de evaluar los métodos de optimización global

No se tiene una forma establecida y generalizada para reportar y evaluar los resultados de los programas de métodos numéricos. La gran mayoría de los autores de métodos numéricos elaboran programas rápidos y en sus resultados destacan sólo los aspectos que consideran importantes.

El artículo de Jackson et al., [Jac91], de reciente publicación se expresa una serie de reglas acerca de la forma de reportar los resultados de los experimentos numéricos. Estas reglas resumen el sentir de un grupo de autores con distintos puntos de vista e intereses.

El artículo es interesante porque destaca que medir la eficiencia de un método numérico a través de un programa del método y de los resultados obtenidos con éste para unos cuantos experimentos no es en general suficiente.

Los principios básicos que se establecen en el artículo son

- los resultados que se publiquen deben de ser suficientes para justificar las expectativas expresadas;
- se deben de dar todos los detalles necesarios para permitir la reproducción de los resultados.

Para medir la eficiencia computacional se deben de tener en cuenta

- la selección de los problemas de prueba, su complejidad, su tamaño y la variedad de estos,

- es conveniente usar una selección de problemas de prueba de colecciones conocidas del área en cuestión,
- la selección de las medidas de eficiencia deben de incluir: el esfuerzo computacional, por ejemplo, el número de evaluaciones de la función, el número de iteraciones, el tiempo del procesador, etcétera; la capacidad de poder corregir entradas de datos invalidas; el rango de problemas que puede resolver y la facilidad de usar el código.

Para cumplir con estos principios, en este trabajo, se brindan todos los detalles tanto de los métodos, como de la elaboración de los códigos.

Y para reportar los resultados, en este trabajo, se usaron criterios similares y una selección de los ejemplos del artículo de V. Levy y Montalvo, [Vel85].

Se utilizaron en todos los experimentos los mismos puntos iniciales que ellos reportan. Se reporta un experimento que ellos no reportaron pero que se realizó con una versión corregida del programa del método original de Antonio Montalvo.

Como no se puede estimar para los métodos de tunelización un número promedio de evaluaciones de la función partiendo sólo de consideraciones teóricas (esto si se puede estimar, por ejemplo, para los métodos Estocásticos), se tomó el promedio del número de evaluaciones de los puntos iniciales como una medida heurística para comparar y medir la eficiencia de los métodos de tunelización. Concretamente se seleccionó el promedio del número de evaluaciones de la función, el promedio del número de evaluaciones de su gradiente y el éxito en la resolución del problema.

No se usó el tiempo de computadora ya que varía muchísimo y depende de la carga de trabajo externa y del tipo de máquina.

5.2 Descripción de los experimentos numéricos

Los doce primeros experimentos se seleccionaron del trabajo realizado por Alejandro V. Levy y Antonio Montalvo¹.

El experimento 5.13 se diseño usando el programa MatLab. Los últimos experimentos son problemas de Bioquímica que fueron planteados en la reunión *Large Scale Applications of Global Optimization to Biochemical problems* celebrada en el *Centre National de la Recherche Scientifique* (CNRS) de París, Francia del 14 al 30 de junio de 1990. Esta reunión fue organizada por Francesco Zirilli de la universidad de Roma, Italia y por Robert A. Donnelly de la universidad de Auburn, Estados Unidos.

De todos los experimentos se describen la función objetivo, el hipercubo, el número de óptimos locales, el número de óptimos globales y los puntos iniciales usados en las corridas.

Experimento 5.1 (Función de Goldstein)²

$$f(x) = x^6 - 15x^4 + 27x^2 + 250, \quad -4 \leq x \leq 4.$$

Este problema tiene 3 mínimos locales, 2 de los cuales son mínimos globales. Los puntos iniciales que se usaron son -3, 0 y 3.

¹Mayor información de los experimentos se encuentra en la tesis [Mon79] y en el artículo [Vel85].

²Ver el artículo [Gol71].

Experimento 5.2 (Función de Shubert en \mathbb{R})³

$$f(x) = \sum_{i=1}^5 i \cos\{(i+1)x + i\}, \quad -10 \leq x \leq 10.$$

El problema tiene 19 mínimos locales, de estos 3 son mínimos globales. Los puntos iniciales que se tomaron son -8, 0 y 8. ■

La función de Shubert⁴ en \mathbb{R}^2 es

$$f(x, y) = \left(\sum_{i=1}^5 i \cos\{(i+1)x + i\} \right) \left(\sum_{i=1}^5 i \cos\{(i+1)y + i\} \right) + \beta \{(x + 1.42513)^2 + (y + 0.80032)^2\}. \quad (5.1)$$

Experimento 5.3 (Función de Shubert en $\mathbb{R}^2, \beta = 0.0$) La ecuación 5.1 tomando $\beta = 0$ y la región factible

$$-10 \leq x, y \leq 10$$

tiene 760 mínimo locales de los cuales 18 son mínimos globales. Los puntos iniciales que se tomaron son (7.0, 7.0), (7.0, -7.0), (-7.0, 7.0) y (0.0, 0.0). ■

Experimento 5.4 (Función de Shubert en $\mathbb{R}^2, \beta = 0.5$) La ecuación 5.1 tomando $\beta = 0.5$ y la región factible

$$-10 \leq x, y \leq 10$$

tiene 760 mínimo locales. Con el factor $\beta = 0.5$ exhibe un mínimo global. Los puntos iniciales que se tomaron son (7.0, 7.0), (7.0, -7.0), (-7.0, 7.0) y (0.0, 0.0). ■

Experimento 5.5 (Función de Shubert en $\mathbb{R}^2, \beta = 1.0$) La ecuación 5.1 tomando $\beta = 1.0$ y la región factible

$$-10 \leq x, y \leq 10.$$

Este problema tiene las mismas características que el anterior y se usaron los mismos puntos iniciales. ■

Experimento 5.6 (Función del camello de 6 jorobas)⁵

$$f(x, y) = (4 - 2.1x^2 + x^4/3)x^2 + xy + (-4 + 4y^2)y^2,$$

$$-3 \leq x \leq 3, \quad -2 \leq y \leq 2.$$

Este problema tiene 6 mínimos locales, dos de los cuales son 2 mínimos globales. Los puntos que se usaron son (-2.9, -1.9), (-2.9, 1.9), (2.9, -1.9) y (2.9, 1.9). ■

³Ver el artículo [Shu72].

⁴Ver el artículo [Shu72].

⁵Ver el artículo [Dix76].

Los dos experimentos siguientes usan la función objetivo,

$$f(y_1, y_2, \dots, y_n) = \frac{\pi}{n} \{ k \sin(\pi y_1)^2 + \sum_{i=1}^{n-1} (y_i - A)^2 (1 + k \sin(\pi y_{i+1})^2) + (y_n - A)^2 \}, \quad (5.2)$$

$$y_i = 1 + (x_i - 1)/4, \quad i = 1, 2, \dots, n.$$

Experimento 5.7 Se usa la fórmula 5.2 fijando $k = 10$, $A = 1.0$ y la dimensión del problema en $n = 2$. Para este experimento la región factible es

$$-10 \leq x_i \leq 10, \quad i = 1, 2, \dots, 2.$$

En este caso se tienen aproximadamente 25 mínimos locales (5^2) y un mínimo global. Los puntos iniciales que se usaron son $(-8.0, 8.0)$, $(8.0, 8.0)$, $(-5.0, 5.0)$ y $(8.0, -8.0)$.

Experimento 5.8 Para este experimento se usó la fórmula 5.2, fijando $k = 10$, $A = 1.0$ y la dimensión del problema en $n = 4$. La región factible es

$$-10 \leq x_i \leq 10, \quad i = 1, 2, \dots, 4.$$

En este caso se tienen aproximadamente 625 mínimos locales (5^4) y un mínimo global. Los puntos iniciales que se usaron son $(-5.0, -5.0, -5.0, -5.0)$, $(5.0, 5.0, 5.0, 5.0)$, $(-5.0, -5.0, 5.0, 5.0)$ y $(5.0, 5.0, -5.0, -5.0)$.

Los experimentos siguientes usan la función,

$$f(x_1, x_2, \dots, x_n) = \frac{\pi}{n} \{ k \sin(\pi x_1)^2 + \sum_{i=1}^{n-1} [(x_i - A)^2 (1 + k \sin(\pi x_{i+1})^2) + (x_n - A)^2], \quad i = 1, 2, \dots, n. \quad (5.3)$$

Experimento 5.9 Se usa la fórmula 5.3 y los parámetros con los valores: $k = 10$ y $A = 1.0$. La dimensión del problema se fija en $n = 8$. La región de factibilidad es

$$-10 \leq x_i \leq 10, \quad i = 1, 2, \dots, 8.$$

La función exhibe aproximadamente 10^8 mínimos locales y tiene un único mínimo global. Los puntos iniciales que se tomaron son $(8.0, 8.0, 8.0, 8.0, 8.0, 8.0, 8.0, 8.0)$, $(-8.0, -8.0, -8.0, -8.0, -8.0, -8.0, -8.0, -8.0)$, $(-8.0, 8.0, -8.0, 8.0, -8.0, 8.0, -8.0, 8.0)$ y $(0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0)$.

Experimento 5.10 Se usa la fórmula 5.3 y los parámetros con los valores: $k = 10$ y $A = 1.0$. La dimensión del problema se fija en $n = 10$. La región de factibilidad es

$$-10 \leq x_i \leq 10, \quad i = 1, 2, \dots, 10.$$

La función exhibe aproximadamente 10^{10} mínimos locales y tiene un único mínimo global. Los puntos iniciales que se tomaron son $(0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0)$, $(2.0, 2.0, 2.0, 2.0, 2.0, 2.0, 2.0, 2.0, 2.0, 2.0)$, $(6.0, 6.0, 6.0, 6.0, 6.0, 6.0, 6.0, 6.0, 6.0, 6.0)$ y $(-1.0, -1.0, -1.0, -1.0, -1.0, -1.0, -1.0, -1.0, -1.0, -1.0)$.

Los dos experimentos que siguen usan la función.

$$f(x_1, x_2, \dots, x_n) = k_1 \sin(\pi l_0 x_1)^2 + k_1 \sum_{i=1}^{n-1} \{(x_i - A)^2 (1 + k_0 \sin(\pi l_0 x_{i+1})^2) + k_1 (x_n - A)^2 (1 + k_0 \sin(\pi l_1 x_n)^2)\}, \quad (5.4)$$

Experimento 5.11 Para este experimento se tomó la formula 5.4 y los parámetros con los valores: $k_0 = 1$, $k_1 = 0.1$, $A = 1$, $l_0 = 3$ y $l_1 = 2$. La dimensión de problema es 4, la región factible es

$$-10 \leq x_i \leq 10, \quad i = 1, 2, \dots, 4.$$

La función tiene aproximadamente 30^4 mínimos locales y un sólo mínimo global. Los puntos iniciales son (5.0, 5.0, 5.0, 5.0), (5.0, 5.0, -5.0, -5.0), (-5.0, -5.0, 5.0, 5.0) y (5.0, 0.0, 0.0, 5.0).

Experimento 5.12 Se usó para este experimento la formula 5.4 y los los parámetros con los valores: $k_0 = 1$, $k_1 = 0.1$, $A = 1$, $l_0 = 3$ y $l_1 = 2$. La dimensión de problema es 7, la región factible es

$$-5 \leq x_i \leq 5, \quad i = 1, 2, \dots, 7.$$

La función tiene aproximadamente 30^7 mínimos locales y un sólo mínimo global. Los puntos iniciales usados son (-3.0, -3.0, -3.0, -3.0, -3.0, -3.0, -3.0), (-3.0, -3.0, -3.0, 0.0, 3.0, 3.0, 3.0), (3.0, 3.0, 3.0, 0.0, -3.0, -3.0, -3.0) y (3.0, 3.0, 3.0, 3.0, 3.0, 3.0, 3.0).

Experimento 5.13 (Función con apéndices) La función de este experimento se creó usando el programa MatLab. La función original es

$$f(x) = -\cos(8x) - 2\exp(-x^2).$$

La región factible que se tomo es

$$-2 \leq x \leq 2,$$

La gráfica de esta función se muestra en la figura 5.1. En esta gráfica se aprecia que el mínimo global es -3 y esta localizado en el punto 0.

Para hacer más complicado el problema a la función anterior se le modificó por medio de las funciones apéndice (ver la sección 1.2.1) en los puntos -1.196039 y 0.758525.

En el primero de los puntos se le aplico una función A_+ y en el segundo una función A_- . La gráfica de la función resultante se muestra en la figura 5.2.

La función resultante tiene el mínimo global en otra localización. En la figura 5.3 se muestra una ampliificación de la gráfica de la función modificada en donde se distinguen los dos mínimos agregados.

Los mínimos que se agregaron son:

- -3.3 en el punto 0.758525 (mínimo global) y
- -3.18 en el punto -1.196039.

El problema de optimización global con esta función es difícil de resolver numéricamente, ya que, la función objetivo tiene mínimos agregados que no son fáciles de localizar y la diferencia entre los mínimos agregados y el mínimo original es mínima (0.3), además, los mínimos agregados (uno de ellos es la solución) tienen valles muy pequeños y con una curvatura no muy suave respecto a la curvatura de la función original.

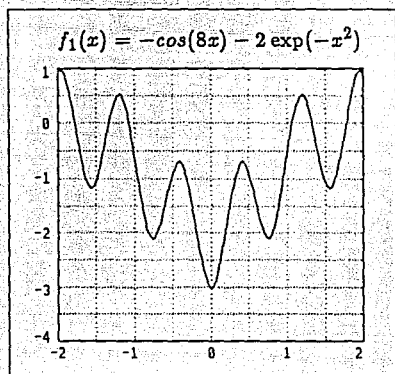


Figura 5.1 Función original del experimento 13

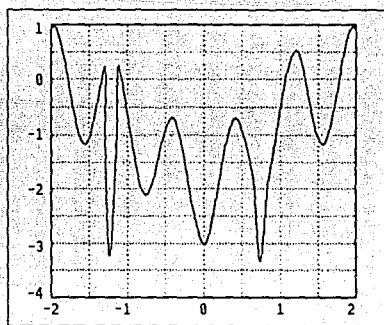


Figura 5.2 Función del experimento 13 modificada con dos apéndices

El modelo matemático de Lennard-Jones 12-6 descrito en la sección 1.5 se utilizó también en los experimentos numéricos. La función de Lennard-Jones 12-6 (ver el artículo [Len32] y el libro [Wat76]) para cuantificar el potencial V entre N partículas es

$$V = \sum_{i>j \geq 1}^N v_{ij} \tag{5.5}$$

donde

$$v_{ij} = 4\epsilon \left\{ \left(\frac{\sigma_{ij}}{r_{ij}} \right)^{12} - \left(\frac{\sigma_{ij}}{r_{ij}} \right)^6 \right\}$$

es el potencial entre las partículas i y j ;

$$\sigma_{ij} = \frac{1}{2}(\sigma_i + \sigma_j),$$

$\epsilon > 0$, σ_k , $k = 1, 2, \dots, N$ son constantes que dependen del tipo de partícula y r_{ij} es la distancia entre las partículas i y j ,

$$r_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2},$$

(x_k, y_k) , $k = 1, 2, \dots, N$ son las coordenadas de las partículas en el plano $X \times Y$.

La función de Lennard-Jones es complicada de resolver, ya que, tiene una infinidad no numerable de mínimos locales y la función no es continua (cuando al menos dos partículas se enciman el potencial crece infinitamente).

El caso más simple es con dos partículas idénticas en el plano en donde el mínimo global ocurre cuando la distancia, entre éstas, es $r^* = \sqrt[6]{2}\sigma$. Las soluciones en el plano $[x_0, y_0] \times [x_1, y_1]$ donde $x_0 + r^* < x_1$, $y_0 + r^* < y_1$ son infinitas, por ejemplo, son puntos mínimos globales todos aquellos que tengan coordenadas de la forma (a, b) y $(a, b + r^*)$ donde $a, b \in [x_0, y_0] \times [x_1, y_1]$ y en general todos los puntos $(a, b), (c, d) \in [x_0, y_0] \times [x_1, y_1]$ cuya distancia sea r^* .

A continuación se describen los experimentos planteados usando la función de Lennard-Jones 12-6.

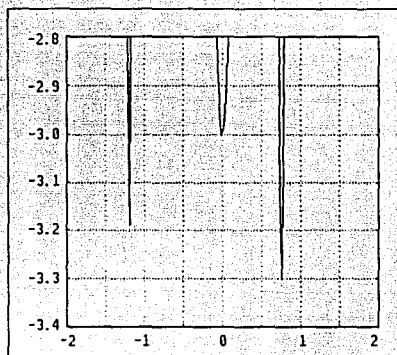


Figura 5.3 Mínimos de la función del experimento 13 modificada con dos apéndices

Experimento 5.14 (Lennard-Jones simétrico) Se usa la función de la ecuación 5.5 tomando $\epsilon = 1$ y suponiendo que se trata de 7 partículas esféricas idénticas, se fija $\sigma_{ij} = 1$. La región factible es

$$-3 \leq x_i, y_i \leq 3, \quad i = 1, 2, \dots, 7.$$

La dimensión del problema es 14. Las coordenadas de las 7 partículas que se tomaron como puntos iniciales son

1. (0,0), (2,0), (1,1), (3,1), (0,2), (2,2), (1,3),
2. (-3,0), (-2,1), (-1,0), (0,1), (1,0), (2,1), (3,0).

Experimento 5.15 (Lennard-Jones no-simétrico 7) Se usa la función de la ecuación 5.5 tomando $\epsilon = 1$ y suponiendo que se trata de 7 partículas esféricas distintas, se toma $\sigma_i = 1 + \frac{1}{5^i}$, $i = 1, \dots, 7$ y se calcula $\sigma_{ij} = \frac{\sigma_i + \sigma_j}{2}$. La región factible es

$$-3 \leq x_i, y_i \leq 3, \quad i = 1, 2, \dots, 7.$$

La dimensión del problema es 14 y las coordenadas de las 7 partículas que se tomaron como puntos iniciales son

1. (0,0), (2,0), (1,1), (3,1), (0,2), (2,2), (1,3),
2. (-3,0), (-2,1), (-1,0), (0,1), (1,0), (2,1), (3,0).

Experimento 5.16 (Lennard-Jones no-simétrico 72) Se usa la función de la ecuación 5.5 tomando $\epsilon = 1$ y suponiendo que se trata de 72 partículas esféricas distintas, se toma $\sigma_i = 1 + \frac{1}{5^i}$, $i = 1, \dots, 72$ y se calcula $\sigma_{ij} = \frac{\sigma_i + \sigma_j}{2}$. La región factible es

$$-12 \leq x_i, y_i \leq 12, \quad i = 1, 2, \dots, 72.$$

La dimensión del problema es 144 y las coordenadas de las 72 partículas que se tomaron como punto inicial son de la forma (x_i, y_j) donde

$$\begin{aligned}
 x_i &= i \quad i = 0, \dots, 11 \\
 y_j &= 2j \quad j = 0, \dots, 5 \quad \text{si } x_i \text{ es par} \\
 y_j &= 2j + 1 \quad j = 0, \dots, 5 \quad \text{si } x_i \text{ es impar}
 \end{aligned}$$

Tabla 5.1 Análisis de sensibilidad del parámetro λ^* del método de tunelización exponencial

experimento 1, $n = 1$, PMG=1				
λ^*	#eval. $f(x)$	#eval. $\nabla f(x)$	#min.loc.	% de éxito
0.12	9	5	2.33	100.00
0.24	7	6	2.33	100.00
0.48	6	5	2.33	100.00
experimento 2, $n = 1$, PMG=3				
λ^*	#eval. $f(x)$	#eval. $\nabla f(x)$	#min.loc.	% de éxito
0.12	559	147	3.67	100.00
0.24	336	141	3.33	77.78
0.48	501	178	3.00	77.78
experimento 3, $n = 2$, PMG=18				
λ^*	#eval. $f(x)$	#eval. $\nabla f(x)$	#min.loc.	% de éxito
0.12	10,009	3,865	17.25	75.00
0.24	8,933	3,441	14.75	65.28
0.48	10,950	4,376	17.25	76.39
experimento 4, $n = 2$, PMG=1				
λ^*	#eval. $f(x)$	#eval. $\nabla f(x)$	#min.loc.	% de éxito
0.12	1,317	618	6.00	100.00
0.24	787	348	3.75	75.00
0.48	492	246	3.50	50.00
experimento 5, $n = 2$, PMG=1				
λ^*	#eval. $f(x)$	#eval. $\nabla f(x)$	#min.loc.	% de éxito
0.12	958	357	5.75	75.00
0.24	510	263	4.75	100.00
0.48	1,167	556	6.5	50.00
experimento 6, $n = 2$, PMG=2				
λ^*	#eval. $f(x)$	#eval. $\nabla f(x)$	#min.loc.	% de éxito
0.12	145	79	3.50	100.00
0.24	64	46	3.50	100.00
0.48	188	102	3.00	100.00

Tabla 5.2 Análisis de sensibilidad del parámetro λ^* del método de tunelización exponencial (cont.)

experimento 7, $n = 2$, PMG=1				
λ^*	#eval. $f(x)$	#eval. $\nabla f(x)$	#min.loc.	% de éxito
0.12	327	147	2.25	100.00
0.24	94	48	2.25	100.00
0.48	248	118	3.50	100.00
experimento 8, $n = 4$, PMG=1				
λ^*	#eval. $f(x)$	#eval. $\nabla f(x)$	#min.loc.	% de éxito
0.12	1,261	444	1.75	100.00
0.24	525	214	2.00	100.00
0.48	622	271	1.75	100.00
experimento 9, $n = 8$, PMG=1				
λ^*	#eval. $f(x)$	#eval. $\nabla f(x)$	#min.loc.	% de éxito
0.12	5,992	1,916	6.25	100.00
0.24	4,978	1,624	5.25	100.00
0.48	9,553	2,052	4.75	100.00
experimento 10, $n = 10$, PMG=1				
λ^*	#eval. $f(x)$	#eval. $\nabla f(x)$	#min.loc.	% de éxito
0.12	303	144	3.50	100.00
0.24	1,608	514	4.25	100.00
0.48	4,889	1,233	4.25	100.00
experimento 11, $n = 4$, PMG=1				
λ^*	#eval. $f(x)$	#eval. $\nabla f(x)$	#min.loc.	% de éxito
0.12	2,766	828	7.00	100.00
0.24	3,678	1,165	10.50	100.00
0.48	1,490	643	9.00	100.00
experimento 12, $n = 7$, PMG=1				
λ^*	#eval. $f(x)$	#eval. $\nabla f(x)$	#min.loc.	% de éxito
0.12	9,311	2,978	11.00	100.00
0.24	1,425	701	8.75	100.00
0.48	2,675	793	7.75	100.00
experimento 13, $n = 1$, PMG=1				
λ^*	#eval. $f(x)$	#eval. $\nabla f(x)$	#min.loc.	% de éxito
0.12	1,190	449	6.00	100.00
0.24	230	114	4.75	25.00
0.48	749	337	4.50	25.00

5.3 Análisis de sensibilidad del método de tunelización exponencial

La sección está dedicada al método exponencial, los primeros 13 experimentos de la sección anterior se tomaron para comparar que tanto afecta el valor del numerador del factor exponencial al resolver el problema 1.1.

Para cada problema se usaron los puntos iniciales mencionados en la definición del experimento.

Las columnas de las tablas del análisis de sensibilidad son los promedios respecto al número de puntos iniciales y los títulos de las columnas son: $\#eval.f(x)$, $\#eval.\nabla f(x)$, $\#min.loc.$ y % de éxito, donde $\#eval.f(x)$ es el promedio del número de evaluaciones de la función, $\#eval.\nabla f(x)$ es el promedio del número de evaluaciones del gradiente de la función, $\#min.loc.$ es el promedio del número de mínimos locales encontrados y % de éxito es el promedio del porcentaje de éxito.

La interpretación de las columnas $\#eval.f(x)$ y $\#eval.\nabla f(x)$ es clara, el método se considera más eficiente cuando realiza menos evaluaciones (siempre que la columna del % de éxito tenga un valor parecido).

La columna de % de éxito se introduce para medir cuando se resuelve el problema 1.1. Su interpretación no es tan directa, ya que, se refiere al promedio del % de éxito obtenido para cada uno de los puntos iniciales de la corrida. El éxito para un punto de inicio se define como el cociente entre el número de óptimos globales encontrados y el número de óptimos globales del problema. Es decir, por ejemplo, para el problema de la función del camello de seis jorobas los porcentajes de éxito para un punto de inicio son 0%, 50% y 100%, cuando no encuentra, cuando encuentra uno y cuando encuentra los dos de los mínimos globales respectivamente. Por otro lado, si el valor que aparece en la columna % de éxito es 50% para dos puntos iniciales, significa que al menos en los 2 obtuvo 50% o bien obtuvo 100% en uno y 0% en el otro. Es evidente que un valor grande del % de éxito implica que el problema se resolvió satisfactoriamente un mayor número de veces. Es importante destacar que, el sentido que le damos a resolver un problema no es el de encontrar un mínimo global, sino que, se trata de resolverlo de manera que se encuentren todos los puntos mínimos globales.

Las tablas 5.1 y 5.2 muestran los resultados obtenidos de los 13 primeros experimentos con distintos valores del parámetro λ^* (numerador del factor exponencial).

Se puede apreciar que

- en particular para los experimentos 2, 3, 4, 5 y 13, el valor de λ^* influye notablemente el % de éxito,
- en general se observa, que el valor de λ^* influye en el número de evaluaciones y hay casos en los cuales el problema se resuelve con % de éxito de 100 % y con un número de evaluaciones muy pequeño, por ejemplo, en los experimentos 6, 7, 8, 9, 10, 11 y 12.

Los experimentos más difíciles de resolver fueron el 3 (con 18 PMG) y el 13 (función modificada con dos apéndices). En estos experimentos es en donde se aprecia mayor variación en la columna % de éxito. Curiosamente para $\lambda^* = 0.48$ se alcanza el mayor % de éxito (76.39 %) en el experimento 3 y para $\lambda^* = 0.12$ se alcanza el 100 % de éxito en el experimento 13.

Otro aspecto que se observa en las tablas es que no necesariamente el número de evaluaciones es proporcional al número de mínimos locales encontrados. Los experimentos en donde es notorio que no hay proporción son el 6, el 8, el 9 y el 10.

Hay algunos experimentos en los cuales el % de éxito disminuye conforme aumenta el valor de λ^* (experimentos 2, 4 y 13), pero también hay otros en los cuales no pasa esto, ni aumenta ni disminuye el % de éxito (experimentos 3 y 5) y en los restantes parece que los distintos valores de λ^* no afectaron el % de éxito.

Por otro lado, clasificamos en cuatro rangos el % de éxito (100-66, 66-33, 33-0 y 0) en la tabla 5.3. Para esta tabla, la columna 100-66 contiene el número de veces que el % de éxito fue mayor a 66 y menor o igual a 100, las demás columnas son similares. La columna del 0 es importante pues contabiliza los fracasos en la resolución de los problemas.

Se aprecia en la tabla 5.3 que para $\lambda^* = 0.12$ el método exponencial siempre resolvió los problemas con un % de éxito mayor al 66% y en el otro extremo, para $\lambda^* = 0.48$ sólo se obtuvieron 10 soluciones en este rango de % de éxito.

Los valores 0.12 y 0.48 de λ^* no fueron tan satisfactorios, pero en ningún caso se fracasó en la resolución de los problemas.

Tabla 5.3 Método de tunelización exponencial, % de éxito por rangos de los 13 experimentos para distintos valores del parámetro λ^*

λ^*	% de éxito			
	100-66	66-33	33-0	0
0.12	13	0	0	0
0.24	11	1	1	0
0.48	10	2	1	0

La tabla 5.4 se refiere al número total de óptimos locales que se encontraron en los 13 experimentos para los distintos valores de λ^* . Hay un comportamiento curioso, no hay gran diferencia entre el número de óptimos locales encontrados para $\lambda^* = 0.48$, que fue el que tuvo menor % de éxito, y $\lambda^* = 0.24$.

Tabla 5.4 Método de tunelización exponencial, total de óptimos locales de los 13 experimentos para distintos valores del parámetro λ^*

λ^*	opt.loc.
0.12	76.25
0.24	70.16
0.48	71.08

De los resultados de las tablas no se distingue un comportamiento bajo el cual seleccionar el valor de λ^* adecuado para resolver el problema 1.1 con el 100 % de éxito y con el menor número de evaluaciones.

Los resultados de la tabla 5.3 sólo muestran que puede ser adecuado usar valores pequeños de λ^* para obtener un mayor % de éxito.

Otro aspecto que se observa en las tablas es que no necesariamente el número de evaluaciones es proporcional al número de mínimos locales encontrados. Los experimentos en donde es notorio que no hay proporción son el 6, el 8, el 9 y el 10.

Hay algunos experimentos en los cuales el % de éxito disminuye conforme aumenta el valor de λ^* (experimentos 2, 4 y 13), pero también hay otros en los cuales no pasa esto, ni aumenta ni disminuye el % de éxito (experimentos 3 y 5) y en los restantes parece que los distintos valores de λ^* no afectaron el % de éxito.

Por otro lado, clasificamos en cuatro rangos el % de éxito (100-66, 66-33, 33-0 y 0) en la tabla 5.3. Para esta tabla, la columna 100-66 contiene el número de veces que el % de éxito fue mayor a 66 y menor o igual a 100, las demás columnas son similares. La columna del 0 es importante pues contabiliza los fracasos en la resolución de los problemas.

Se aprecia en la tabla 5.3 que para $\lambda^* = 0.12$ el método exponencial siempre resolvió los problemas con un % de éxito mayor al 66% y en el otro extremo, para $\lambda^* = 0.48$ sólo se obtuvieron 10 soluciones en este rango de % de éxito.

Los valores 0.12 y 0.48 de λ^* no fueron tan satisfactorios, pero en ningún caso se fracasó en la resolución de los problemas.

Tabla 5.3 Método de tunelización exponencial, % de éxito por rangos de los 13 experimentos para distintos valores del parámetro λ^*

λ^*	% de éxito			
	100-66	66-33	33-0	0
0.12	13	0	0	0
0.24	11	1	1	0
0.48	10	2	1	0

La tabla 5.4 se refiere al número total de óptimos locales que se encontraron en los 13 experimentos para los distintos valores de λ^* . Hay un comportamiento curioso, no hay gran diferencia entre el número de óptimos locales encontrados para $\lambda^* = 0.48$, que fue el que tuvo menor % de éxito, y $\lambda^* = 0.24$.

Tabla 5.4 Método de tunelización exponencial, total de óptimos locales de los 13 experimentos para distintos valores del parámetro λ^*

λ^*	opt.loc.
0.12	76.25
0.24	70.16
0.48	71.08

De los resultados de las tablas no se distingue un comportamiento bajo el cual seleccionar el valor de λ^* adecuado para resolver el problema 1.1 con el 100 % de éxito y con el menor número de evaluaciones.

Los resultados de la tabla 5.3 sólo muestran que puede ser adecuado usar valores pequeños de λ^* para obtener un mayor % de éxito.

Tabla 5.5 Comparación de los métodos de tunelización

experimento 1, $n = 1$, PMG=1				
método	#eval. $f(x)$	#eval. $\nabla f(x)$	#min.loc.	% de éxito
original ^a	4	3	2.33	100.00
clásico	5	4	2.33	100.00
Exp. $\lambda^* = 0.48$	6	5	2.33	100.00
experimento 2, $n = 1$, PMG=3				
método	#eval. $f(x)$	#eval. $\nabla f(x)$	#min.loc.	% de éxito
original ^a	234	69	4.33	100.00
clásico	866	142	4.33	100.00
Exp. $\lambda^* = 0.12$	559	147	3.67	100.00
experimento 3, $n = 2$, PMG=18				
método	#eval. $f(x)$	#eval. $\nabla f(x)$	#min.loc.	% de éxito
original	12,160	1,731	17.00	94.44
clásico	7,650	2,220	14.50	58.33
Exp. $\lambda^* = 48$	10,950	4,376	17.25	76.39
experimento 4, $n = 2$, PMG=1				
método	#eval. $f(x)$	#eval. $\nabla f(x)$	#min.loc.	% de éxito
original ^b	2,912	390	3.00	100.00
clásico	1,178	289	4.00	100.00
Exp. $\lambda^* = 0.12$	1,317	618	6.00	100.00
experimento 5, $n = 2$, PMG=1				
método	#eval. $f(x)$	#eval. $\nabla f(x)$	#min.loc.	% de éxito
original ^b	2,180	274	3.00	100.00
clásico	549	180	5.00	100.00
Exp. $\lambda^* = 0.24$	510	263	4.75	100.00
experimento 6, $n = 2$, PMG=2				
método	#eval. $f(x)$	#eval. $\nabla f(x)$	#min.loc.	% de éxito
original ^b	1,496	148	2.00	100.00
clásico	549	180	5.00	100.00
Exp. $\lambda^* = 0.24$	64	46	3.50	100.00
experimento 7, $n = 2$, PMG=1				
método	#eval. $f(x)$	#eval. $\nabla f(x)$	#min.loc.	% de éxito
original ^b	2,443	416	2.75	100.00
clásico	210	84	3.25	100.00
Exp. $\lambda^* = 0.24$	94	48	2.25	100.00
^a Estos resultados fueron obtenidos de un programa del método original.				
^b Estos resultados se tomaron del artículo [Vel85].				

Tabla 5.6 Comparación de los métodos de tunelización (cont.)

experimento 8, $n = 4$, PMG=1				
método	#eval. $f(x)$	#eval. $\nabla f(x)$	#min.loc.	% de éxito
original ^b	4,881	1,371	4.25	100.00
clásico	875	184	2.75	100.00
Exp. $\lambda^* = 0.24$	525	214	2.00	100.00
experimento 9, $n = 8$, PMG=1				
método	#eval. $f(x)$	#eval. $\nabla f(x)$	#min.loc.	% de éxito
original ^b	19,366	2,370	2.50	100.00
clásico	12,196	2,137	9.75	100.00
Exp. $\lambda^* = 0.24$	4,978	1,624	5.25	100.00
experimento 10, $n = 10$, PMG=1				
método	#eval. $f(x)$	#eval. $\nabla f(x)$	#min.loc.	% de éxito
original ^b	23,982	3,272	2.50	100.00
clásico	371	182	4.00	100.00
Exp. $\lambda^* = 0.12$	303	144	3.50	100.00
experimento 11, $n = 4$, PMG=1				
método	#eval. $f(x)$	#eval. $\nabla f(x)$	#min.loc.	% de éxito
original ^b	3,861	588	3.75	75.00
clásico	7,243	1,255	11.00	100.00
Exp. $\lambda^* = 0.48$	1,490	643	9.00	100.00
experimento 12, $n = 7$, PMG=1				
método	#eval. $f(x)$	#eval. $\nabla f(x)$	#min.loc.	% de éxito
original ^b	16,063	2,792	7.50	75.00
clásico	10,433	1,854	12.50	100.00
Exp. $\lambda^* = 0.24$	1,425	701	8.75	100.00
experimento 13, $n = 1$, PMG=1				
método	#eval. $f(x)$	#eval. $\nabla f(x)$	#min.loc.	% de éxito
original ^a	296	78	5.25	50.00
clásico	653	164	5.00	0.00
Exp. $\lambda^* = 0.12$	1,190	449	6.00	100.00
^a Estos resultados fueron obtenidos de un programa del método original.				
^b Estos resultados se tomaron del artículo [Vel85].				

5.4 Comparación de los métodos de tunelización: original, clásico y exponencial

Las tablas de los resultados de las corridas que se hicieron para comparar las tres versiones de los métodos de tunelización tienen las mismas columnas que las que se explicaron para la sección anterior.

Las tablas 5.5 y 5.6 contienen las columnas #eval. $f(x)$, #eval. $\nabla f(x)$, #min.loc. y % de éxito y los renglones son los promedios de los tres métodos para los puntos iniciales (para cada experimento se usaron los mismos puntos en los tres métodos).

Lo primero que resalta de las tabla 5.5 y 5.6 son las variaciones que se tienen en casi todos los resultados de los experimentos (el experimento 1 es la excepción pues las tres columnas tienen valores muy semejantes). Hay valores muy diferentes en la columna de % de éxito de estas tablas respecto a las tablas 5.1 y 5.2.

La columna de óptimos locales presenta grandes diferencias, pero lo más sobresaliente es que el método original tiene casi siempre el menor valor (esto se debe a que su % de éxito es casi siempre el menor).

Se nota que el método exponencial alcanza siempre el 100 % de éxito.

Es muy singular que los métodos original y clásico tengan una distribución muy parecida a este nivel de detalle (la única diferencia está en el experimento 13).

Este resultado no es tan sorprendente, ambos métodos usan la misma función de tunelización, la función de tunelización clásica y son muy parecidos los pasos que realizan al resolver el problema 1.1 (las diferencias entre estos métodos se dan en la tabla 2.1).

Para notar la diferencia se requiere comparar en las tablas 5.5 y 5.6 el número de evaluaciones y el % de éxito entre los métodos clásicos y original. Aquí se aprecia que los cambios entre estas versiones hacen que la versión clásica sea la que alcance un mayor % de éxito y sea la más eficiente en el sentido de que realice un menor número de evaluaciones de la función y del gradiente.

Tabla 5.7 Comparación de los métodos de tunelización, % de éxito por rangos de los 13 experimentos

método	% de éxito			
	100-66	66-33	33-0	0
original	12	1	0	0
clásico	11	1	0	1
exponencial	13	0	0	0

La tabla 5.7 separa en cuatro clases al % de éxito de las tablas 5.5 y 5.6, (las clases tienen el mismo significado que en la tabla 5.3). En esta tabla se puede apreciar que el método más efectivo resulto ser el exponencial.

Finalmente y de acuerdo con los resultados de las tablas 5.5 y 5.6, sin lugar a dudas, el método exponencial es el más eficiente, ya que, tiene los valores más grandes en la columna % de éxito y los valores más pequeños en las columnas de las evaluaciones de la función y del gradiente.

5.5 Comparación de los métodos de las funciones de relleno y de tunelización: clásico y exponencial

Una parte adicional de este trabajo fue la comparación de los métodos clásico y exponencial con el método de las funciones de relleno del investigador Ge Renpu (ver la sección 3.1.2).

Para esta comparación se tomaron los resultados y los puntos iniciales que reporta en el artículo [Ren90].

Tabla 5.8 Comparación de los métodos de las funciones de relleno y de tunelización

experimento	n	clásico ^a	exponencial ^a	relleno ^a
4	2	234	748, $\lambda^* = 0.249$	990
6	2	53	51, $\lambda^* = 0.245$	375
9	5	1,171	769, $\lambda^* = 0.245$	1,075
	10	1,692	985, $\lambda^* = 0.245$	1,253

^a Los puntos iniciales se tomaron del artículo [Ren90].
Se reporta el promedio de la suma del número de evaluaciones de $f(x)$ con el número de evaluaciones de $\nabla f(x)$ para alcanzar un PMG.

En la tabla 5.8 los experimentos cambiaron un poco, los puntos iniciales que se usaron son los reportados por Renpu y del experimento 9 se obtuvieron dos resultados cambiando la dimensión por 5 y 10 respectivamente. El éxito en la resolución de los problemas consiste en alcanzar un MG (no todos los PMG como en las secciones anteriores).

Las columnas de la tabla 5.8 reportan para cada experimento: la dimensión (n), la suma de los promedios del número de evaluaciones de la función y de su gradiente ($\#eval.f(x) + \#eval.\nabla f(x)$) para alcanzar un MG.

De los valores de la tabla se nota que el método exponencial fue el que realizó el menor número de evaluaciones de la función y del gradiente.

Lo interesante de esta comparación es que el método de las funciones de relleno no resulta mejor (al menos para estos experimentos) que los métodos de tunelización clásico y exponencial.

Esto contrasta con los resultados reportados en el artículo de Renpu en donde la comparación era muy favorable al método de las funciones de relleno respecto del método tunelización original.

5.6 El potencial de Lennard-Jones 12-6 y los métodos de tunelización: clásico y exponencial

Los 13 primeros experimentos son relativamente sencillos, comparados con los experimentos que usan la función del potencial de Lennard-Jones (del que no se conoce la solución).

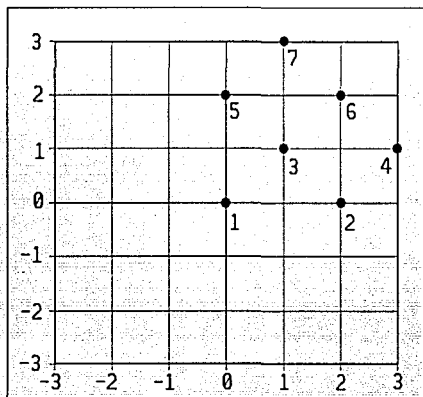


Figura 5.4 Configuración uno de 7 de partículas

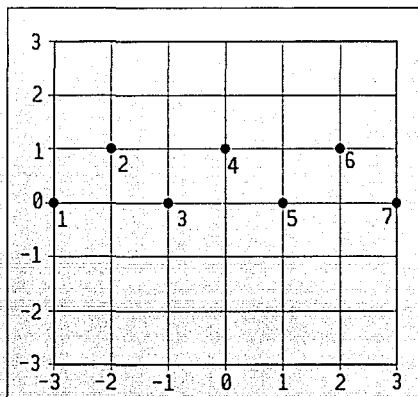


Figura 5.5 Configuración dos de 7 de partículas

Los experimentos con la función del potencial de Lennard-Jones 12-6 (ecuación 5.5) son de dos tipos: para partículas esféricas idénticas y para partículas esféricas diferentes. Los experimentos de 7 partículas se corrieron para las dos configuraciones que se muestran en las figuras 5.4 y 5.5.

Tabla 5.9 Potencial mínimo de Lennard-Jones 12-6 de 7 partículas esféricas idénticas alcanzado a partir de un configuración casi óptima

$V = -12.529230$		$V^* = -12.534867$	
$\ \nabla V\ ^2 = 1.2884740$		$\ \nabla V\ ^2 = 2.7486210 \cdot 10^{-9}$	
(x, y)		(x^*, y^*)	
0.000000	0.000000	0.00000057	0.00000000
0.561231	0.9720800	0.55923060	0.96861484
-0.561230	0.9720800	-0.55922946	0.96861484
-1.122460	0.000000	-1.11845950	0.00000000
-0.561230	-0.9720800	-0.55922946	-0.96861484
0.561231	-0.9720800	0.55923060	-0.96861484
1.122462	0.000000	1.11846060	0.00000000

Para esta sección se trató de resolver con los tres métodos el problema de Lennard-Jones pero los resultados de experimentos preliminares mostraron que el método original no lo

resolvía tan satisfactoriamente como los métodos clásico y exponencial.

Por esta razón, en esta sección sólo se dan los resultados obtenidos con los métodos clásico y exponencial.

Tabla 5.10 Potencial mínimo de Lennard-Jones 12-6 de 7 partículas esféricas idénticas alcanzado a partir de las configuraciones 1 y 2

experimento 5.14					
		clásico		exponencial	
punto inicial		#eval. $f(x)$ =66,781		#eval. $f(x)$ =26,091	
$V=-6.6779$		$V^*=-12.534867$		$V^*=-12.534867$	
(x, y)		(x^*, y^*)		(x^*, y^*)	
0.0	0.0	-0.7827	2.1058	0.3380	1.6327
2.0	0.0	-0.7054	0.1701	2.0081	0.6511
1.0	1.0	-1.9003	2.0612	0.8897	0.6597
3.0	1.0	-1.3028	1.1156	2.5748	1.6153
0.0	2.0	-2.4204	1.0710	0.9047	2.5969
2.0	2.0	-1.8230	0.1255	1.4564	1.6240
1.0	3.0	-0.1853	1.1602	2.0231	2.5882
punto inicial		#eval. $f(x)$ =124,464		#eval. $f(x)$ =33,939	
$V=-4.9841$		$V^*=-11.501291$		$V^*=-12.534867$	
(x, y)		(x^*, y^*)		(x^*, y^*)	
-3.0	0.0	-0.4681	0.2862	-0.9113	0.2577
-2.0	1.0	0.6459	2.2187	-0.2572	1.1650
-1.0	0.0	0.0969	1.2491	-0.4526	-0.7623
0.0	1.0	0.0906	-0.6830	0.2015	0.1449
1.0	0.0	1.2155	0.1251	0.6602	-0.8752
2.0	1.0	-1.0175	1.2637	0.8556	0.1052
3.0	0.0	0.6548	-0.2823	1.3142	0.0321

Para el experimento 5.14 se muestra en la tabla 5.9 la pequeña diferencia que se tiene entre el potencial de la solución encontrada por la conjetura 1.1 y la solución encontrada a partir de este punto usando la rutina de minimización CONMIN.

Aquí se puede apreciar como la precisión en las coordenadas es muy significativa respecto al valor del potencial de la solución de este problema.

En la tabla 5.10 se dan los resultados del experimento 5.14. El objetivo de este experimento es alcanzar el potencial óptimo partiendo de las configuraciones iniciales 1 y 2.

En los resultados de la tabla 5.10 se aprecia que el método exponencial alcanzó el potencial óptimo en ambos casos y con un número menor de evaluaciones de la función.

Es notable el aumento, que se nota en la tabla, en el número de evaluaciones de la función por comenzar en un punto alejado del potencial óptimo, ya que, aparentemente entre más alejado se empiece del potencial óptimo más evaluaciones de la función se realizan.

Para el experimento 5.15 de 7 partículas distintas se dan los resultados en la tabla 5.11.

Tabla 5.11 Potencial mínimo de Lennard-Jones 12-6 de 7 partículas esféricas diferentes alcanzado a partir de las configuraciones 1 y 2

experimento 5.15					
		clásico		exponencial	
punto inicial $V=-6.6779$ (x, y)		#eval. $f(x)=242,647$ min.loc.=12 $V^*=-12.5388$ (x', y')		#eval. $f(x)=101,640$ min.loc.=9 $V^*=-12.4856$ (x', y')	
0.0	0.0	0.8090	-1.5463	2.1167	1.8445
2.0	0.0	1.8307	-0.9999	1.2103	1.2808
1.0	1.0	0.8101	-0.3557	0.9637	2.1468
3.0	1.0	-0.2613	0.2672	0.3957	0.0876
0.0	2.0	1.9268	0.2183	1.2994	0.9561
2.0	2.0	-0.2852	-1.0022	2.5542	0.6836
1.0	3.0	0.8386	0.9320	1.6437	-0.2972
punto inicial $V=-4.9841$ (x, y)		#eval. $f(x)=20,327$ min.loc.=7 $V^*=-12.3871$ (x', y')		#eval. $f(x)=158,488$ min.loc.=11 $V^*=-12.5407$ (x', y')	
-3.0	0.0	0.2661	1.8953	-1.0304	-0.1718
-2.0	1.0	2.2098	1.0332	-2.0166	-2.3568
-1.0	0.0	1.4542	1.9896	-1.5276	-1.2535
0.0	1.0	-0.2607	0.7063	-0.8138	-2.2665
1.0	0.0	1.8088	-0.1527	-2.7747	-1.3983
2.0	1.0	0.9949	0.8316	-0.2627	-1.1225
3.0	0.0	0.4926	-0.3755	-2.2668	-0.1995

En los resultados de la tabla 5.11 se puede observar que los dos métodos alcanzaron los valores del orden de -12.54 . Siendo el método exponencial el que obtuvo el valor más pequeño del potencial, pero, por tan sólo una diferencia de 0.002 .

En la misma tabla se aprecia que los valores del número de evaluaciones de $f(x)$ de los métodos clásico y exponencial son muy diferentes, el método clásico realizó casi el doble de evaluaciones que el método exponencial.

Para el experimento 5.16 de 72 partículas distintas no se conoce la solución. La corrida que se realizó para este experimento fue para una sola configuración.

La tabla 5.12 muestra en sus columnas el número de evaluaciones de la función, el número de óptimos locales y el potencial que se alcanzó hasta que la fase de tunelización no obtuvo solución.

En la tabla 5.12 se aprecia que el método exponencial obtuvo la mejor solución y con sólo la cuarta parte del número de evaluaciones de la función que realizó el método clásico.

Tabla 5.12 Potencial mínimo de Lennard-Jones 12-6 de 72 partículas esféricas diferentes alcanzado a partir de una configuración inicial

experimento 5.16		
punto inicial, $V=-100.7439$	clásico	exponencial
$(x_i, y_j), x_i = i, i = 0, \dots, 11$	#eval. $f(x)=322,611$	#eval. $f(x)=58,996$
$y_j = 2j, j = 0, \dots, 5$, si x_i es par	min.loc.=41	min.loc.=20
$y_j = 2j + 1, j = 0, \dots, 5$, si x_i es impar	$V^*=-194.269628$	$V^*=-200.713784$

Capítulo 6

Conclusiones

Una vez terminado el trabajo de escritura y de experimentación de esta tesis han surgido nuevas inquietudes, como por ejemplo, la posibilidad de continuar investigando el problema de las partículas, para luego pasar a problemas más realistas de moléculas, etcétera.

Este documento es la culminación de la investigación que nos propusimos, la Dra. Gómez y un servidor, para desarrollar como tesis.

Del trabajo desarrollado concluimos en los tres puntos siguientes:

1. Funcionamiento de los programas de tunelización.
2. ¿ Qué es un método o algoritmo numérico ?
3. Métodos de tunelización o metodología de tunelización.

1. Se menciona que el método exponencial, representado por el programa EXPONEN, resultó ser el más eficiente. En teoría los tres métodos son semejantes y una comparación teórica para deducir cual funcionará mejor en la resolución del problema 1.1 está fuera de los límites de este trabajo. Por tanto, nuestras conclusiones están respaldadas por los resultados de los experimentos numéricos y los resultados de los experimentos los aceptamos con base a que los tres programas elaborados tienen la misma calidad (el programa del método original es una versión revisada y corregida del programa de Antonio Montalvo y Alejandro V. Levy, ver la tesis doctoral [Mon79]).

En el diseño de la evaluación de los métodos (ver la sección 5.1) se busco resaltar que se resolviera el problema con un alto % de éxito y con un pequeño número de evaluaciones de la función y del gradiente (ver las tablas del capítulo 5), cuya finalidad era ampliar la justificación acerca de la eficiencia (en el sentido de evaluaciones de la función conjuntamente con el % de éxito).

De los resultados experimentales se ve que el funcionamiento de los programas de los tres métodos no fue el mismo. En todos los experimentos numéricos el método exponencial obtuvo las soluciones de los problemas y, sin lugar a dudas, su funcionamiento fue el más satisfactorio.

La versión clásica resulto mejor que la versión original. La diferencia se noto en el % éxito y en que hubo una disminución efectiva del número de evaluaciones de la función y del gradiente (ver la sección 5.4).

A nivel experimental la conjetura 3.1, de que $\exists \lambda^*$ tal que el método exponencial es más eficiente que los métodos original y clásico, se comprobó.

De paso y también en forma pragmática se comprobó que el método clásico es más eficiente que el método original.

El método exponencial resultó ser el más eficiente de los tres en el sentido de la conjetura y de las medidas de eficiencia y % de éxito.

Además, la comparación que se hizo con el método de las funciones de relleno del investigador Renpu (ver la sección 5.5) resultó ser favorable para los métodos clásico y exponencial, pero aquí, sólo se consideraron unos cuantos experimentos, pocos puntos iniciales y se desconoce la calidad del programa que represento al método de la función de relleno.

Resulta satisfactorio haber llegado a resultados tan buenos para el método exponencial, sin embargo, debemos de notar que, de antemano, no se tiene el valor de λ^* que garantiza que se realizarán el menor número de evaluaciones de la función objetivo con el mayor % de éxito para un problema dado. El valor adecuado de este parámetro depende de la función del problema y una mala selección afecta la eficiencia del método exponencial, tal como se observa en algunos de los casos de las tablas 5.1 y 5.2.

2. Ya se menciona en la sección 4.1 que el uso de la palabra algoritmo para describir la forma de resolver un problema numéricamente no coincide con el concepto formal de algoritmo (máquina de Turing, autómatas o conjuntos regulares). Realizar un código de un algoritmo numérico no es una tarea sencilla, requiere de una buena formación matemática y tener una buena experiencia en la programación de métodos similares. No está demás que también conozca las bibliotecas de rutinas de métodos numéricos: MINPACK, LINPACK, MODULEF, etcétera.

Actualmente, como no hay herramientas computacionales para la especificación formal de los algoritmos numéricos, la eficiencia de un código (funcionamiento adecuado para algunos problemas) depende enteramente de la habilidad del programador.

Falta mucho por hacer para introducir mayor formalismo computacional en los llamados métodos o algoritmos numéricos. Dentro del área de métodos numéricos el concepto de algoritmo o método numérico que se maneja actualmente es del estilo de la siguiente definición.

Un algoritmo numérico es un conjunto de pasos bien definidos para resolver un problema numérico en un tiempo finito.

La definición anterior es tan formal como una receta de cocina.

En el capítulo 4 se dio un panorama de los lenguajes de programación y de los ambientes para la experimentación numérica y se explicó porque se seleccionó el lenguaje FORTRAN. También se expuso una formalización funcional de los sistemas aritméticos bajo un punto de vista funcional y con base a esta conceptualización se expusieron algoritmos numéricos formales que inclusive al codificarlos funcionaron tal como se especificaron. El resultado práctico fue la rutina EPSMAQ que permite transportar los programas de tunelización sin necesidad de readaptarlos manualmente a la estructura del sistema aritmético del equipo de cómputo.

3. El cambio que se introduce al sugerir nuevos factores para generar nuevas funciones de tunelización trajo consigo una ampliación del concepto de tunelización. En particular, este trabajo se distingue de los anteriores por el sentido con que maneja el término: metodología de tunelización. El sentido (ver sección 3.5) se refiere a la introducción de otras funciones para construir polos y no es el mismo que se manejaba en el método clásico, que era respecto a su aplicación a problemas particulares de optimización, como por ejemplo, el llamado método de tunelización para el problema de mínimos cuadrados.

El nuevo enfoque se refiere a otras funciones para construir polos. Esta idea posibilita nuevas líneas de investigación para continuar este trabajo, una de las más inmediatas es probar el factor exponencial en las mismas aplicaciones del método de tunelización clásico: problemas de mínimos cuadrados y problemas de optimización global con restricciones.

La línea de investigación que se abre con la metodología de tunelización es extensa, una parte es la creación de nuevas funciones de tunelización, otra es la selección de la función de tunelización adecuada para un problema particular, otra es el análisis de convergencia, otra los efectos de los polos móviles, etcétera.

Tal labor va más allá de los fines de este trabajo y aquí sólo se plantearon los primeros cimientos de la metodología.

El teorema 3.9 muestra que el factor

$$\frac{1}{\{(x - x^*)^T B (x - x_{\min})\}^\lambda}$$

donde x^* es un punto mínimo de f , B es una matriz $n \times n$ definida positiva y λ es un número positivo, sirve para construir la función de tunelización

$$T_B(x, x^*, \lambda) = \frac{f(x) - f(x^*)}{\{(x - x^*)^T B (x - x_{\min})\}^\lambda}$$

Tal función puede ser muy adecuada para resolver problemas en donde se utilice en forma natural el Hessiano o una aproximación de esta matriz, por ejemplo, en la resolución de problemas de minimización global sin restricciones por el método BFGS.

Por otro lado, no es difícil ver que esta nueva función de tunelización genera polos cuyas curvas de nivel no son de forma circular y que por tanto puede rellenar¹ mejor el valle del mínimo. Las curvas de nivel de los factores $\frac{1}{\|x - x^*\|^{2\lambda}}$ y $\exp\left(-\frac{\lambda}{\|x - x^*\|^2}\right)$ son circulares.

Con una conjetura acerca de cuando usar la función de tunelización T_B terminamos este trabajo.

Conjetura 6.1 (T_B) Si la matriz B es una aproximación al Hessiano en el punto mínimo x^* resulta más apropiado usarla en lugar de la función T_c en el método de tunelización clásico para problemas con funciones cuyos valles alrededor de los puntos mínimos no tengan curvas de nivel circulares. ■

El sentido de "más apropiado" es en el de realizar menos evaluaciones de la función objetivo durante la fase de tunelización con el mismo % de éxito que cuando se usa la función de tunelización clásica.

¹El termino rellenar es en el sentido de que un polo destruya el valle de un punto mínimo de manera uniforme.

Bibliografía

- [Apo76] Apostol, T.M. *Análisis Matemático*. Reverte, España, 1976.
- [Bar91] Barrón, Carlos. *Manual de referencia de los programas de tunelización: CLASICO y EXPONENCIAL*. IIMAS-UNAM, 1991.
- [Bra68] Bracken, Jerome and McCormick, Garth P. *Selected Applications of Nonlinear Programming*. John Wiley, U.S.A., 1968.
- [Chi89] Chiang, Kao. A random-number generator for microcomputers. *J. Operational Research Society*, 40(7), 1989.
- [Cor85] Cortés, Alfredo. Solución de sistemas de ecuaciones no lineales usando microcomputadora. 1985. Tesis de Licenciatura en Física y Matemáticas, Universidad Michoacana de San Nicolás de Hidalgo.
- [Dah74] Dahlquist, Germund and Björck, Åke. *Numerical Methods*. Prentice-Hall, U.S.A., 1974. Series in Automatic Computation.
- [Dix76] Dixon, L.C.W. and Gomulka, J. and Hersom, S.E. Reflections on the global optimization problem. In L.C.W. Dixon, editor, *Optimization in Action*, pages 398-433, Academic Press, England, 1976.
- [Dub85] Dubrulle, Augustin A. and Scarborough, Randolph G. and Kolsky, Harwood G. *How Write Good Vectorizable Fortran*. Technical Report G320-3478, IBM, Palo Alto Scientific Center, (1539 Page Mill Road, Palo Alto CA 94303-0821), U.S.A., 1985.
- [Evt85] Evtushenko, Yuriy G. *Numerical Optimization Techniques*. Springer-Verlag, U.S.A., 1985.
- [Fle87] Fletcher, R. *Practical Methods of Optimization*. John Wiley, U.S.A., 1987.
- [Fou87] Fourer, Robert and Gay, David M. and Kernighan, Brian W. *AMPL: A Mathematical Programming Language*. Computing Science Technical Report 133, AT&T Bell Laboratories, (Murray Hill, New Jersey 07974), U.S.A., 1987.
- [Fre60] Freudenthal, Hans. *Lincos: Design of Language for Cosmic Intercourse*. North-Holland, Netherlands, 1960. Part I.
- [Gil81] Gill, Philip E. and Murray, Walter and Wright, Margaret H. *Practical Optimization*. Academic Press, U.S.A., 1981.

- [Gol71] Goldstein, A.A. and Price, J.F. On descent from local minima. *Mathematics of Computation*, 25(115), July 1971.
- [Gom84] Gómez, Susana and Velasco Levy, Alejandro. The tunneling method for solving the constrained global optimization problem with several non-connected feasible regions. In J.P. Hennart, editor, *Workshop in Numerical Analysis: Lecture Notes in Mathematics, No. 909*, Springer Verlag, 1984.
- [Gom87a] Gómez, Susana, Calderón, Apolinar and Velasco Levy, Alejandro. *The Tunneling Algorithm for Solving the Global Zero Residual Least Squares Problem*. Comunicaciones Técnicas: Serie Naranja 370, IIMAS-UNAM, México, 1987.
- [Gom87b] Gómez, Susana, Velasco Levy, Alejandro and Calderón, Apolinar. A global zero residual least squares method. In J.P. Hennart, editor, *Workshop in Numerical Analysis: Lecture Notes in Mathematics, No. 1290*, Springer Verlag, 1987.
- [Gom91a] Gómez, Susana and Barrón, Carlos. The exponential tunneling method. *Journal of Global Optimization*, 1(2), 1991. To appear.
- [Gom91b] Gómez, Susana and Barrón, Carlos. The exponential tunneling method. *Reportes de Investigación*, 1(3), July 1991. IIMAS-UNAM, México.
- [Her85] Hernandez, Onésimo y Martinez, Efraín J. *Programación Matemática*. CONACYT, I.P.N., Centro de Investigación y Estudios Avanzados, México, Agosto 1985. Notas de un curso del IV Coloquio del Depto. de Matemáticas, Taxco, Guerrero.
- [Hes75] Hestenes, Magnus R. *Optimization Theory: The finite dimensional case*. John Wiley, U.S.A., 1975.
- [Hor88] Horst, R. Deterministic global optimization with partition sets whose feasibility is not known: application to concave minimization, reverse convex constraints, dc-programming, and lipschitzian optimization. *Journal of Optimization Theory and Applications*, 58(1), July 1988.
- [Hor89] Horst, R. On consistency of bounding operations in deterministic global optimization. *Journal of Optimization Theory and Applications*, 61(1), April 1989.
- [Jac91] Jackson, Richard H.F and Boggs, Paul T. and Nash, Stephen G. and Powell, Susan. Guidelines for reporting results of computational experiments. report of ad hoc committee. *Mathematical Programming*, 49, 1991.
- [Kul86] Kulish, U. V. and Miranker, W.L. The arithmetic of digital computer: a new approach. *SIAM Review*, 28(1), March 1986.
- [Las80] Lasdon, Leon S. and Warren, Allan and Ratner, Margery W. *GRG2 User's Guide, Guía de Operación No.69*. Banco de México, Septiembre 1980.
- [Len32] Lennard-Jones, J. E. Processes of adsorption and diffusion on solid surfaces. *Trans. Faraday Soc.*, 28, 1932.

- [Lue73] Luenberger, David G. *Introduction to Linear and Nonlinear Programming*. Addison-Wesley, U.S.A., 1973.
- [Mal72] Malcon, Michael A. Algorithms to reveal properties of floating-point arithmetic. *Communications of the ACM*, 15(11), November 1972.
- [Mal74] Malcon, Michael A. More algorithms that reveal properties of floating-point arithmetic units. *Communications of the ACM*, 17(5), May 1974.
- [Man74] Manna, Z. *Mathematical theory of computation*. Mc-Graw Hill, U.S.A., 1974.
- [Mat65] Matyas, J. Random optimization. *Automat. Remote Control*, 26, 1965.
- [Mat87] MathWorks. *MatLab User's Guide*. April 1987. V 3.05.
- [McC83] McCormick, Garth P. *Nonlinear Programming: Theory, Algorithms and Applications*. John Wiley, 1983.
- [Mee88] Meewella, C.C. and Mayne, D.Q. An algorithm for global optimization of lipschitz continuous functions. *Journal of Optimization Theory and Applications*, 57(2), May 1988.
- [Mee89] Meewella, C.C. and Mayne, D.Q. Efficient domain partitioning algorithms for global optimization of rational and lipschitz continuous functions. *Journal of Optimization Theory and Applications*, 61(2), May 1989.
- [Mon79] Montalvo, Antonio y Velasco Levy, Alejandro. *Algoritmo de Tunelización para la Optimización Global de Funciones*. Comunicaciones Técnicas: Serie Naranja 204, IMAS-UNAM, México, 1979.
- [Ren90] Renpu, Ge. A filled function method for finding a global minimizer of a function of several variables. *Mathematical Programming*, 46:191-204, 1990.
- [Rin86] Rinnoy Kan, A.H.G. and Timmer, G.T. *Global Optimization*. Technical Report 8612/A, Erasmus University Rotterdam, Econometric Institute, The Netherlands, 1986.
- [Ros88a] Rosing, Matthew and Schnabel, Robert B. *An Overview of Dino, A New Language for numerical computation on distributed memory multiprocessors*. Technical Report CU-CS-385-88, March, University of Colorado, Department of Computer Science, (Campus Box 430, University of Colorado), Boulder, CO 80309-0430, U.S.A., 1988.
- [Ros88b] Rosing, Matthew and Schnabel, Robert B. and Weaver, Robert. *Dino: Summary and Examples*. Technical Report CU-CS-386-88, March, University of Colorado, Department of Computer Science, (Campus Box 430, Boulder, CO 80309-0430), U.S.A., 1988.
- [Roy68] Royden, H.L. *Real Analysis*. The Macmillan Company, 1968.

- [Sha78] Shano, D.F. and Phua, K.H. *A Variable Method Subroutine for Unconstrained Nonlinear Minimization*. MIS technical report 28, University of Arizona, U.S.A., December 1978.
- [Shu72] Shubert, Bruno O. A sequential method seeking the global maximum of a function. *SIAM J. Num. Anal.*, 9(3), September 1972.
- [Sol81] Solís, Francisco and Wets, Roger J.B. Minimization by random search techniques. *Mathematics of Operations Research*, 6(1), February 1981.
- [Vel80a] Velasco Levy, Alejandro and Gómez, Susana. *The Tunneling Algorithm for Global Optimization of Constrained Functions*. Comunicaciones Técnicas: Serie Naranja 370, IIMAS-UNAM, México, 1980.
- [Vel80b] Velasco Levy, Alejandro and Gómez, Susana. *The Tunneling Algorithm for the global optimization of constrained functions*. Comunicaciones Técnicas: Serie Naranja 231, IIMAS-UNAM, México, 1980.
- [Vel80c] Velasco Levy, Alejandro and Montalvo, Antonio. *A Modification to the Tunneling Algorithm for Finding the Global Minima of an Arbitrary One Dimensional Scalar Function*. Comunicaciones Técnicas: Serie Naranja 240, IIMAS-UNAM, México, 1980.
- [Vel83] Velasco Levy, Alejandro and Calderón, Apolinar. *Stabilization of Newton's Method To Solve A System of Nonlinear Equations*. Comunicaciones Técnicas: Serie Naranja 329, IIMAS-UNAM, México, 1983.
- [Vel84] Velasco Levy, Alejandro and Gómez, Susana. The tunneling method applied to global optimization. In Paul T. Boggs, Richard H. Byrd and Robert B. Schnabel, editor, *Numerical Optimization*, SIAM Numerical, Colorado, U.S.A., 1984.
- [Vel85] Velasco Levy, Alejandro and Montalvo, Antonio. The tunneling algorithm for the global minimization of functions. *SIAM J. Sci. Stat. Comput.*, 6(1), January 1985.
- [Wat76] Watts, R. O. and McGee, I.J. *Liquid State Chemical Physics*, chapter 2. John Wiley, New York, U.S.A., 1976.

Apéndice A

Ambientes computacionales para la investigación

En este anexo se describen las funciones que deben ofrecer los ambientes computacionales para la investigación. La forma de trabajar de los investigadores tiende a evolucionar debido a los avances tecnológicos de los equipos de cómputos y del software.

La labor del investigador empieza con un planteamiento y termina en los reportes o artículos que escriba. El planteamiento se escribe, se diseñan los experimentos, luego se realizan los experimentos y se juntan todos los resultados en una primera versión de la publicación. Cada etapa del trabajo requiere distintos recursos, que desgraciadamente, no siempre están a la mano o no son los más adecuados. Los recursos inadecuados o su ausencia ocasionan una enorme dependencia de otras personas y la publicación se demora. Las demoras se deben, entre otras, a las transcripciones, a las revisiones del trabajo, al tiempo que se tarda en analizar los resultados de los cómputos de los experimentos, a la tarea de integrar manualmente resultados y documentos, etcétera.

Para auxiliar al investigador, cada vez aumentan más el número de equipos de cómputo y de programas que ofrecen las herramientas necesarias para que realice todas las etapas, incluyendo la labor de edición, en una forma más interactiva e integral. Por ejemplo, las estaciones de trabajo SunSparc 1 tienen: programas de procesamiento de texto de alta calidad: $\text{T}_{\text{E}}\text{X}$ y $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$; compiladores para los lenguajes FORTRAN, PASCAL, C, etcétera; manejadores de bases de datos: SunINGRES, SunUNIFY; programas de cálculo simbólico: Mathematica; programas de sistema operativo y manejadores de ventanas: UNIX, X11, SunView; programas de graficación: SunGKS, SunPHIGS; programas de comunicación: SunLink; etcétera.

En el IIMAS no existe un único ambiente para elaborar reportes, artículos, libros, etcétera. En general, los distintos departamentos del instituto utilizan recursos muy diversos para desarrollar y generar las publicaciones: VAX, Macintosh, PC compatibles, impresoras laser, Word Start, ChiWriter, Ventura, Perfect Writer, Epsilon, fácil $\text{T}_{\text{E}}\text{X}$, $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$, etcétera.

Hasta ahora, no se han planteado claramente los recursos que se deben tener en un ambiente para facilitar el desarrollo de las investigaciones y de la escritura de Textos Científicos y esto acarrea distintas formas de trabajo, dificultades en la elaboración de los reportes, falta de normas para la elaboración de los documentos, una gran dispersión de recursos de cómputo.

La solución está lejos de ser fácil, sin embargo, se proponen dos ambientes, uno ideal basado en las estaciones de trabajo y otro práctico basado en los recursos disponibles en el Instituto y en la UNAM.

La problemática de la elaboración de documentos ya ha sido tratada en el IIMAS y se han escrito, acerca del tema, varios trabajos que tratan de la tipografía orientada a la escritura de textos científicos y de la elaboración de la documentación de los programas:

- Manual de lenguaje y tipografía científica en castellano, K. Bernardo Wolf, *et al.*, editorial trillas, México, 1984.
- Fácil T_EX, Max Díaz, Comunicaciones técnicas del I.I.M.A.S. serie amarilla: Desarrollo, No. 32, México, 1982.
- Diseño de un formateador general de programas, Juan Osorio y Alfredo Cortés, Comunicaciones técnicas del I.I.M.A.S. serie amarilla: Desarrollo, No. 96, México, 1988.

Para el área de computación se tienen los lenguajes para tipografía automatizada, arreglo y diseño, de documentos (por ejemplo, T_EX) y existen sistemas más elaborados como el sistema WEB, el cual está descrito en el artículo de D.E. Knuth *Literate Programming*. El sistema WEB permite obtener de un solo documento el artículo (en T_EX) y el código del programa (en lenguaje Pascal) correspondientes.

Las bases de los ambientes de las actuales estaciones de trabajo surgieron hace varios años como una serie de programas de software que se denominaban Sistemas Integrales. Los paquetes de este tipo ofrecían básicamente

- un módulo para la escritura,
- un módulo para realizar cálculos, generalmente del tipo de Hoja Electrónica y
- un módulo para comunicaciones.

Este tipo de paquetes de software evoluciono de tal forma que se unieron equipos de cómputo y software bajo el título de estación de trabajo. Actualmente, las estaciones de trabajo proporcionan un ambiente que facilita la investigación y la elaboración de los reportes correspondientes.

La figura A.1 muestra el modelo del ambiente ideal para escribir textos científicos. Las funciones que se pueden realizar son

- administración centralizada de archivos,
- generación de gráficas,
- operación sencilla del sistema,
- intercomunicación entre aplicaciones,
- edición para generar documentos de alta calidad,
- generación de códigos 100 % compatibles en distintos lenguajes de programación,
- generación de experimentos y de prototipos de pequeña y de gran escala,
- generación de estadísticas.

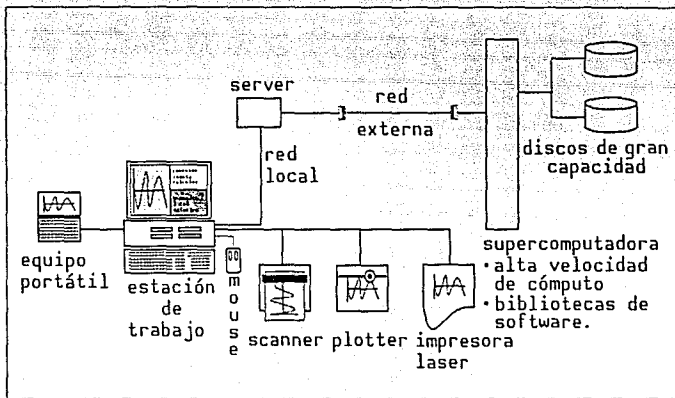


Figura A.1 Ambiente ideal para la investigación y la elaboración de reportes

Las estaciones de trabajo son el futuro, por ejemplo, las de tipo Sun ofrecen un ambiente basado en el sistema operativo UNIX, el manejo de las aplicaciones se realiza usando X11 o SUNWindow (ambos tienen una fuerte orientación a ventanas de trabajo). Es muy probable que la supercomputadora que va a adquirir la UNAM tenga un ambiente UNIX por lo que en el futuro se tendrá un ambiente de trabajo muy cercano al modelo ideal en toda la UNAM¹.

Con todas las funciones anteriores las investigaciones que se realizan, por ejemplo, en el desarrollo de métodos numéricos se harían de la siguiente manera

1. se escribe un borrador de la investigación,
2. se generan los programas correspondientes,
3. se diseñan y se escriben los experimentos numéricos,
4. se empieza a experimentar para afinar los programas en el equipo de cómputo local, y cuando se tienen listos los códigos,
5. se experimenta en el equipo de cómputo grande,
6. se elaboran las tablas o las gráficas usando los resultados directamente, se junta todo y
7. se termina de escribir el artículo.

Con los recursos adecuados todos los puntos anteriores se desarrollan en el ambiente ideal de la siguiente forma

¹Los aspectos técnicos de redes de comunicación se omiten por estar fuera del alcance de este apartado. Cabe mencionar que, la red de comunicaciones de la UNAM es un proyecto vital para la universidad, afortunadamente, se cuentan con algunos avances al respecto.

- la escritura se realiza directamente en los equipos de cómputo local: microcomputadora portátil (LapTop) o estación de trabajo (se ahorra la transcripción y se realiza en la oportunidad que se quiera, ya sea en la oficina o en la casa),
- el paso del equipo de cómputo local al equipo de cómputo grande es transparente. (no hay dos sistemas operativos distintos, sino uno solo),
- los resultados de los experimentos son incorporados al artículo en la forma de tabla o de gráfica con las herramientas adecuadas (independientemente de los distintos tipos de archivo, se tienen las facilidades para intercambiar datos y calcular estadísticas) y finalmente
- el artículo es terminado en los equipos de cómputo locales por el autor (la mayor parte de la escritura se puede realizar sin apoyo secretarial).

La realización de esta tesis no fue como en el ambiente ideal o en los ambientes de las estaciones de trabajo. Se trabajó, en lo que nombramos el ambiente práctico, con una gran variedad de productos que no estaban integrados y que no eran disponibles en un sólo equipo. Esto ocasionó que se tuvieron algunos conflictos en la intercomunicación, en el manejo de archivos, en los distintos sistemas operativos, etcétera. A pesar de estos conflictos, las funciones que se dieron para el modelo del ambiente ideal se realizaron con el equipo y software siguiente:

- Para la administración y manipulación de archivos:
 - el emulador de la terminal IBM 3270 instalado para manejar los comandos SEND y RECEIVE en lote, el programa FastWire, el editor de Norton, el sistema operativo DOS, el sistema operativo VM de la IBM 4381 y los programas Xedit, Filel, los programas de compactación y descompactación de archivos Pkarc y Pkxarc.
- Para la generación y manipulación de gráficas:
 - una microcomputadora del tipo PC con video de alta resolución VGA, un mouse, un scanner, el programa MatLab, el programa LOTUS 123, el programa Paint Brush, el programa Picinf.
- Para la operación de equipo IBM 4381: el sistema operativo VM y para la operación de las microcomputadoras PC Compatibles: el sistema operativo DOS.
- Para la intercomunicación entre aplicaciones:
 - el emulador de la terminal IBM 3270 instalado para manejar los comandos SEND y RECEIVE en lote, el programa FastWire y programas en FORTRAN.
- Para generar documentos de alta calidad tipográfica: $\text{T}_{\text{E}}\text{X}$ bajo $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$.
- Para la generación de los códigos de los métodos:
 - el compilador de FORTRAN WatFort-77, el compilador de FORTRAN RMFort y el compilador de FORTRAN de VS2 para IBM 4381.

- Para elaborar estadísticas: el programa LOTUS 123 y programas en FORTRAN.
- Para el diseño de experimentos y de prototipos de los métodos: MatLab.

Los recursos anteriores son diversos, algunos son poco conocidos, pero en conjunto dieron un buen soporte para el desarrollo y la escritura de esta tesis. Muchos de los recursos mencionados fueron prestados por el departamento de Computación, por la Secretaría Técnica del IIMAS y la DGESCA.

La forma en que se integraron los resultados de las corridas de los experimentos numéricos con el documento final fue de la siguiente forma:

1. Se enviaron de la microcomputadora PC al computador 4381 los fuentes de los programas de los métodos y los datos usando el comando SEND.
2. Se escribió un archivo de comandos de VM en la 4381. Este archivo contenía los comandos para realizar los experimentos numéricos.
3. Se escribió un archivo de comandos de DOS en la PC. Este archivo contenía los comandos RECEIVE necesarios para traer los resultados a la microcomputadora PC.
4. Por medio de un programa escrito en FORTRAN o usando LOTUS se calcularon las estadísticas y se generaron las tablas y las gráficas. En la generación de las tablas se anexaban las instrucciones necesarias de $\text{IAT}_{\text{P}}\text{X}$ para poder agregarlas al documento final.
5. Las gráficas se generaban en LOTUS o en MatLab y se copiaban directamente del video usando el programa FRIEZE de Paint Brush o se dibujaban en Paint Brush o, bien, se digitalizaban por medio del scanner usando, también, Paint Brush.

Es poco recomendable seguir tratando de juntar piezas de equipo de cómputo y de software para construir el ambiente práctico de trabajo. La posibilidad de construir un ambiente que funcione lo demuestra este trabajo, pero hay demasiados aspectos que se opondrían (costos, capacitación, políticas, pertenencia de los equipos, etcétera) como para pensar en establecerlo como una forma de trabajo institucional.

Por todo lo anterior se debe tener claro que las formas de trabajo evolucionan y que es necesario hacer un esfuerzo para mantenerse actualizado en los nuevos recursos de la tecnología computacional.

Mientras esto ocurre, en este apéndice se dan los recursos necesarios para trabajar en una imitación del ambiente ideal²...

²Al terminar de escribir esta tesis, llegaron estaciones de trabajo Sparc 1 de Sun al IIMAS: una a la maestría en Ciencias de la Computación, otra al departamento de Ciencias de la Computación, una más al departamento de Matemáticas y Mecánica y una al proyecto de Sistemas Distribuidos (Victor G. Sánchez y Hanna Oktaba).