

# TEMA 8.

## INTRODUCCION A LA OPTIMIZACIÓN NUMÉRICA

1. Introducción
2. Nomenclatura
3. Optimización multivariable sin restricciones
4. Búsqueda Unidimensional
5. Métodos directos
6. Métodos indirectos de primer orden
7. Métodos indirectos de segundo orden
8. RESUMEN
9. Programacion en Matlab®



## 1. Introducción

Aumentar la presión para reducir costes, mejorar la calidad del producto, minimizar riesgos medio ambientales, son algunas de las motivaciones para desarrollar herramientas de optimización para los complejos problemas de operación y diseño de una planta química. Varios factores han contribuido a este desarrollo. En primer lugar la disponibilidad de ordenadores y su creciente capacidad de cálculo ha facilitado la aplicación de complejos modelos matemáticos. En segundo lugar, el desarrollo y mejoría de los modelos económicos que permiten decidir entre diferentes procesos competitivos. Y en tercer lugar el reciente desarrollo del software para optimización que ha proporcionado la herramienta adecuada para el uso de los modelos matemáticos tanto de operaciones físicas como económicas, para la identificación de las mejores soluciones. De hecho, muchas de las decisiones en la industria química relacionadas con diseño de procesos y operaciones están basadas en técnicas de optimización que combinan los nuevos desarrollos en algoritmos de optimización, modelado de sistemas, y arquitectura y software de ordenadores.

Aunque muchas de las técnicas matemáticas fueron desarrolladas por investigadores relacionados con investigación de operaciones, análisis numérico y ciencia de los ordenadores, los ingenieros químicos han jugado un importante papel en alguno de estos desarrollos (de hecho un importante porcentaje de las personas implicadas en investigación de operaciones son ingenieros químicos). A continuación daremos una visión general de la optimización en ingeniería química haciendo un énfasis especial en los desarrollos más significativos de los últimos 10 años.

Si no se asume ninguna estructura especial para el problema que se está tratando (el problema es considerado como una caja negra) entonces las técnicas de **búsqueda directa** son los métodos más sencillos de aplicar, pero son también los que más tiempo consumen. Estos métodos realizan una búsqueda pseudo aleatoria o sistemática (simplex flexible, algoritmos genéticos, etc.) para mejorar la función objetivo. Si el problema presenta restricciones estos métodos usan métodos costosos y a menudo con dificultades intrínsecas, por ejemplo los métodos de penalización. Como contrapartida el método es fácil de

aplicar a problemas con restricciones sencillas y probablemente encontrará soluciones cercanas al óptimo global. Sin embargo debido al hecho de que a menudo requiere miles de evaluaciones de la función objetivo su rendimiento será altamente dependiente de la selección de los parámetros del algoritmo.

La aproximación más frecuente para resolver problemas de optimización consiste en clasificarlos de acuerdo al tipo de función objetivo y al tipo de restricciones y desarrollar métodos específicos que aprovechen la estructura especial de dichos problemas.

Un problema general viene expresado de la forma:

$$\begin{aligned} & \text{Min } f(\mathbf{x}, \mathbf{y}) \\ & \text{s.a } \mathbf{g}(\mathbf{x}, \mathbf{y}) \leq 0 \\ & \quad \mathbf{h}(\mathbf{x}, \mathbf{y}) = 0 \\ & \quad \mathbf{x} \in X \subseteq \mathfrak{R}^n \\ & \quad \mathbf{y} \in \{0, 1\}^m \end{aligned}$$

donde  $F(x,y)$  es la función objetivo a minimizar (maximizar) que esta sujeta a restricciones de desigualdad, igualdad, tanto de variables continuas  $x$ , como variables discretas  $y$ .

El caso más conocido, es el de la **programación lineal (LP)**. En este caso la función objetivo es lineal y las restricciones que forman el problema también son lineales. La solución óptima de un LP cae en un vértice del espacio formado por la región factible. Cualquier solución local debe ser además la solución global del problema. Estos problemas han sido resueltos durante muchos años utilizando el algoritmo simplex, basado en métodos algebraicos. El cambio principal en los métodos de solución de los problemas de programación lineal ha sido el desarrollo de problemas de punto interior, que tratan con transformaciones no lineales y cuyos requerimientos de cálculo están, teóricamente acotados por un polinomio expresado en términos del tamaño del problema. Esta propiedad no la comparte el algoritmo del simplex, que teóricamente tiene un crecimiento exponencial. Aunque este comportamiento es rara vez observable en la práctica. El algoritmo simplex permite resolver

problemas de manera eficiente, que contengan hasta 50000 ó 100000 restricciones. Los métodos de punto interior funcionan mejor con problemas de hasta 500000 ó incluso 1000000 restricciones. La estructura particular de ciertos problemas de programación lineal (problema de flujo, de transporte, asignación...) han llevado al desarrollo de métodos específicos que han permitido resolver problemas con millones de variables.

La **programación lineal mixta (MILP)** es la extensión de los problemas de programación lineal cuando aparecen variables discretas. Los modelos donde aparecen variables discretas permiten acercarnos a formular problemas del mundo real basados en decisiones lógicas (esto es variables 0 ó 1) o donde el número de posibilidades es discreto. La forma más común de resolver los MILP es la técnica de "branch and bound" (ramificación y acotamiento) que consiste en resolver un subconjunto de problemas de programación lineal dentro de un árbol de decisión de variables discretas. Otras aproximaciones comunes suponen el uso de técnicas de plano de corte que hace el MILP resoluble como un LP añadiendo ciertas restricciones especiales. Teóricamente estos problemas son de difícil solución, debido a la naturaleza combinatoria del problema que no esta polinomialmente acotado a medida que aumenta el número de variables discretas. Sin embargo el uso conjunto de técnicas de Branch and Bound y de plano de corte ha permitido desarrollar algoritmos para resolver problemas que se consideraban irresolubles no hace muchos años

Para el caso en el cual todas, o al menos algunas de las funciones son no lineales y además sólo aparecen variables continuas, aparece un problema de **programación no lineal (NLP)**. Si la función objetivo y las restricciones son diferenciables, los óptimos locales vienen definidos por las condiciones de optimalidad de Kuhn-Tucker. Estos son, probablemente los modelos más comunes en Ingeniería Química.

Si bien problemas que contenían hasta 100 variables para un NLP eran considerados problemas grandes hace 10 años. La resolución de problemas con varios miles de variables es algo bastante común hoy en día. Los métodos de gradiente reducido y la programación cuadrática sucesiva, los cuales se generan de aplicar el método de Newton a las condiciones de optimalidad de Kuhn-Tucker, son los principales algoritmos para resolver los NLP. El método de

gradiente reducido es mejor para resolver problemas con la mayoría de las restricciones lineales. Para problemas altamente no lineales es mejor la programación cuadrática sucesiva.

Una limitación de estos métodos es que su convergencia sólo está garantizada a un óptimo local. Para problemas que tienen una función objetivo convexa y una región factible convexa, este problema no existe, porque estos problemas sólo tienen un óptimo local que coincidirá con el óptimo global. En la práctica probar la convexidad de un problema no lineal es a menudo difícil, sino imposible, por lo que encontrar un óptimo local es considerado a menudo una solución satisfactoria sobre todo si ello significa una mejora en el proceso.

La extensión de la programación no lineal para tratar con variables discretas lleva a la **programación no lineal con variables enteras mixta (MINLP)**. Algoritmos como la descomposición de Benders generalizada o la “aproximación exterior” (outer approximation) son los más utilizados para resolver estos problemas. Estos métodos que suponen que la función debe ser diferenciable resuelven una secuencia de problemas NLP y MILP. El primero optimiza las variables continuas y el segundo las discretas. Como en el caso anterior la solución sólo está garantizada para problemas convexos.

Finalmente, todos los métodos mencionados anteriormente suponen que los problemas están formulados en forma de ecuaciones algebraicas. Muy a menudo, sin embargo, dentro del problema aparecen ecuaciones diferenciales como restricciones del problema. Una de las aproximaciones principales para resolver este problema consiste en aproximar la ecuación diferencial por ecuaciones algebraicas, las cuales ya pueden ser incluidas dentro del programa general de programación no lineal. Otra manera de resolver el problema es considerar la ecuación diferencial como una ecuación a parte que se resuelve en un bloque independiente y se introduce dentro del NLP como una ecuación implícita.

En este capítulo, sin embargo, sólo se van a introducir los rudimentos más básicos de la optimización numérica. En particular se estudiará la optimización numérica para problemas sin restricciones (o cuyas restricciones pueden ser resueltas de forma implícita y transferidas a la función objetivo)

## 2. Nomenclatura

$a, b, c$	
<b>B</b>	En el algoritmo simplex flexible: centroide de todos los vértices excepto el que da pero valor de función objetivo.
$f(\mathbf{x}), f(\mathbf{x}, \mathbf{y})$	Función escalar. Generalmente hace referencia a la función objetivo en un problema de optimización. Puede venir en función de variables reales ( $\mathbf{x}$ ) o enteras, generalmente binarias ( $\mathbf{y}$ ).
$f'(x)$	Primera derivada de la función escalar $f$ respecto de la variable escalar $x$
$f''(x)$	Segunda derivada de la función escalar $f$ respecto de la variable escalar $x$
$F_B$	Número phi. $F_B = 1 - \frac{3 - \sqrt{5}}{2} \approx 0.618$
$F_s$	Uno menos phi.
$i, j, k, p, q$	Índices, significado variable según contexto.
<b>H</b>	Matriz hessiana
<b>I</b>	Matriz identidad
$L^0, L^k$	En búsqueda unidireccional, longitud del intervalo en la etapa inicial y en la etapa $k$ respectivamente.
$m$	Pendiente de una recta
$n$	Número de puntos que forman la figura simplex.
<b>Q</b>	Matriz definida positiva
$Q_{\max}$	En el método simplex, peor valor de función objetivo (desarrollado para minimizar)
$Q_{\min}$	En el método simplex, mejor valor de función objetivo (desarrollado para minimizar)
$Q^*$	En el método simplex, valor de función objetivo tras reflexión.
$Q^{**}$	En el método simplex, valor de función objetivo tras expansión.
$Q^{***}$	En el método simplex, valor de función objetivo tras contracción.
$s^k$	Dirección de búsqueda en la etapa $k$ .

$\mathbf{x}$	Vector de variables reales.
$\mathbf{x}^*$	Solución óptima de un problema de optimización.
$\mathbf{x}^k$	Valor del vector de variables $\mathbf{x}$ en la iteración $k$ .
$\mathbf{y}$	Vector de variables binarias (0,1).
$y$	Variable escalar
$\beta$	Parámetro escalar
$\gamma_r$	En el método simplex, coeficiente de reflexión.
$\gamma_e$	En el método simplex, coeficiente de expansión.
$\gamma_c$	En el método simplex, coeficiente de contracción.
$\lambda$	Longitud de paso en búsqueda unidireccional
$\varepsilon$	Tolerancia para la terminación de la búsqueda del óptimo
$\delta$	Factor de aceleración en búsqueda unidireccional
$\mu$	Longitud de paso en el método de Powell.
$\  \cdot \ $	Hace referencia a la norma de un vector o matriz



### 3. Optimización multivariable sin restricciones

#### 3.1. Introducción

La optimización numérica de funciones no lineales requiere la utilización de técnicas de optimización eficientes y robustas. La eficiencia es importante porque la solución de estos problemas se lleva a cabo por un procedimiento iterativo. La robustez (habilidad para encontrar la solución) es una propiedad deseable dado que el comportamiento de las funciones no lineales puede ser impredecible en su comportamiento: puede presentar máximos, mínimos y puntos de silla. En algunas regiones el avance hacia el óptimo puede ser muy lento necesitando mucho tiempo de cálculo etc. Afortunadamente se posee mucha experiencia utilizando métodos de optimización numérica lo que permite contar con buenos algoritmos y conocer sus limitaciones y posibilidades.

En este capítulo discutiremos la solución del problema sin restricciones:

Encontrar  $\mathbf{x}^* = [x_1, x_2, \dots, x_n]^T$  que minimice  $f(x_1, x_2, \dots, x_n) \equiv f(\mathbf{x})$

La mayor parte de los procedimientos iterativos que son efectivos, alternan la optimización en dos fases

- (a) Elección de una dirección  $\mathbf{s}^k$
- (b) Movimiento en la dirección  $\mathbf{s}$ , (en alguna extensión, o hasta encontrar un mínimo) para encontrar un nuevo punto  $\mathbf{x}^{k+1} = \mathbf{x}^k + \Delta\mathbf{x}$  donde  $\Delta\mathbf{x}^k$  se suele llamar el *tamaño del paso*.

Además de (a) y (b) un algoritmo debe especificar:

- (c) Un vector de valores iniciales  $\mathbf{x}^0 = [x_1^0, x_2^0, \dots, x_n^0]^T$
- (d) Un criterio de convergencia para la terminación del algoritmo.

La mayor parte de los algoritmos de cálculo siguen una metodología similar. Se determina un punto inicial, se evalúa la función en ese punto y se elige una dirección de búsqueda. Se comienza entonces un movimiento en la dirección de búsqueda, hasta encontrar un óptimo en esa dirección, o bien hasta que se produzca una mejoría determinada. A continuación se selecciona una nueva dirección y así sucesivamente.

Los métodos NLP sin restricciones que discutiremos en este capítulo difieren en como se generan las direcciones de búsqueda. Algunos métodos utilizan información de las derivadas, mientras que otros se basan solamente en evaluaciones de la función objetivo. Comenzaremos con algunos métodos que no usan derivadas y después presentaremos los métodos que usan información de las derivadas.

### *3.2. Optimización de funciones sin restricciones*

#### *Búsqueda unidireccional*

Una buena técnica de optimización de funciones de una única variable es fundamental por al menos tres razones:

- 1.- En muchos problemas las restricciones se pueden incluir dentro de la función objetivo, por lo que la dimensionalidad del problema se reduce a una variable.
- 2.- Algunos problemas sin restricciones, inherentemente incluyen una única variable.
- 3.- Las técnicas de optimización con y sin restricciones, generalmente incluyen pasos de búsqueda unidireccional en sus algoritmos.

Antes de la aparición de los ordenadores de alta velocidad, los métodos de optimización estaban prácticamente limitados a los *métodos indirectos* en los

cuales el cálculo del extremo potencial estaba restringido al uso de derivadas y la condiciones necesaria de optimalidad. Los modernos ordenadores han hecho posible los métodos directos, esto es la búsqueda de un óptimo por comparación sucesiva de los valores de la función  $f(x)$  en una secuencia de puntos  $x_1, x_2, x_3...$  sin la necesidad de hacer intervenir derivadas analíticas.

Para llevar a cabo los métodos directos de minimización numérica solamente se usa el valor de la función objetivo. Se comienza con un valor inicial de  $x$  y se continúa seleccionando valores de  $x$  de acuerdo con una estrategia pre-seleccionada. El proceso termina cuando  $f(x^{k+1}) - f(x^k) < \varepsilon$  donde el superíndice  $k$  designa el número de iteración y  $\varepsilon$  es la tolerancia pre-especificada o criterio de tolerancia.

Los métodos indirectos tienen la ventaja inherente de que la convergencia es generalmente más rápida incluso aun cuando se utilicen métodos numéricos para calcular las derivadas. Sin embargo, en problemas de ingeniería esta ventaja es muchas veces neutralizada por la falta de interés de determinaciones precisas de la función objetivo debido a la falta de precisión de los coeficientes que muchas veces se utilizan.

Los métodos numéricos directos, tienen la ventaja de que pueden tratar fácilmente con problemas que incluyan discontinuidades, puntos de inflexión y puntos finales. También el carácter de  $f(x)$  en las vecindades de un extremo es fácilmente estudiable.

Para resolver un problema de optimización no lineal sin restricciones bastaría derivar cada una de las funciones e igualar a cero. Aparecería entonces un sistema de  $n$  ecuaciones no lineales. Sin embargo el problema así planteado podría ser incluso más difícil que el problema de optimización original, por eso muchos autores prefieren resolver el problema de optimización por métodos directos usando algunas de las técnicas que veremos en próximas secciones.

Para aplicar los métodos de búsqueda directa se debe comenzar por acotar el punto donde se encuentra el óptimo, y asegurarse de que la función es unimodal en el intervalo considerado. Es muy difícil determinar, a priori, si una

función es unimodal, pero en muchos casos prácticos las funciones presentan esta característica.

Un método de optimización para una única variable, podría consistir en dividir el intervalo de búsqueda en una rejilla (numero de intervalos), y calcular la función objetivo en cada uno de los puntos de la rejilla. El óptimo será el mejor de todos los valores obtenidos. Por otra parte es casi imposible llevar a la práctica estos métodos en un espacio múltiple de variables.

Para seleccionar el método de optimización se debe alcanzar un compromiso entre la complejidad del procedimiento y el número total de evaluaciones que se debe realizar.

### Acotación del óptimo

Casi todos los procedimientos de búsqueda unidimensional requieren que el óptimo esté acotado dentro de un intervalo conocido como primer punto de la estrategia de búsqueda.

Existen varias estrategias que se pueden usar para acotar el óptimo, la más sencilla consiste en fijar un punto y comenzar a movernos una distancia fija en una dirección. Por ejemplo, si fijamos como punto inicial el cero, podemos movernos 0.01 cada vez. Aunque el método da buenos resultados es bastante ineficaz. Una alternativa es hacer una transformación de  $x$ , por ejemplo a una escala logarítmica o bien incluir un factor de aceleración:

$$x_{k+1} = x_k + \delta 2^{k-1}$$

De todas maneras, en una buena parte de los problemas de optimización los límites de las variables vienen dados de forma natural por consideraciones físicas: Máxima y mínima temperatura permitida, límites de presión, valores de las fracciones molares etc...

Los métodos de búsqueda unidimensional se pueden clasificar en métodos directos, métodos indirectos de búsqueda secuencial, y métodos de interpolación polinómica. Veremos ahora cada uno de ellos.

## 4. Búsqueda Unidimensional:

### 4.1. Métodos Indirectos

Han sido desarrollados, básicamente tres métodos para llevar a cabo la búsqueda directa unidireccional, basados en las condiciones de optimalidad. Estos son:

- 1.- Método de Newton
- 2.- Aproximaciones finitas al método de Newton (Métodos cuasi-Newton)
- 3.- Métodos de secante.

Para comparar la eficacia de cada método, es útil examinar la velocidad de convergencia de cada método. Las velocidades de convergencia se pueden clasificar de muchas maneras, las más comunes son:

$$\text{Lineal} \quad \frac{\|x^{k+1} - x^*\|}{\|x^k - x^*\|} \leq c \quad 0 \leq c \leq 1$$

$$\text{Orden } p \quad \frac{\|x^{k+1} - x^*\|}{\|x^k - x^*\|^p} \leq c \quad c > 0; p \geq 1 \text{ (El más rápido en la práctica)}$$

$$\text{Superlineal} \quad \lim_{k \rightarrow \infty} \frac{\|x^{k+1} - x^*\|}{\|x^k - x^*\|} \rightarrow 0 \quad \text{(Habitualmente rápido)}$$

#### 4.1.1. Método de Newton

De acuerdo con la primera condición necesaria para que una función tuviera un mínimo local se debe cumplir que  $f'(x)=0$ . Por lo tanto podemos aplicar el método de Newton a la derivada, así:

$$x^{k+1} = x^k - \frac{f'(x^k)}{f''(x^k)}$$

Asegurándonos que en la etapa  $k$ ,  $f(x^{k+1}) < f(x^k)$ , para minimización. Realmente lo que hace el método de Newton es aproximar la función por una función cuadrática en  $x^k$ .

Las ventajas del método de Newton son:

- 1.- El procedimiento es cuadráticamente convergente ( $p=2$ ), siempre que  $f''(x) \neq 0$ .
- 2.- Para una función cuadrática el mínimo se obtiene en una única iteración.

Las desventajas son:

- 1.- Se debe calcular tanto  $f'(x)$  como  $f''(x)$ .
- 2.- Si  $f''(x) \rightarrow 0$  el método converge muy lentamente.
- 3.- Si existe más de un extremo, el método podría no converger al extremo deseado. Además el método podría oscilar.

#### 4.1.2. Métodos cuasi Newton

Los métodos cuasi Newton, se utilizan si la derivada de la función objetivo es difícil de calcular, o ésta viene dada de forma numérica. Se basan en sustituir las derivadas por aproximaciones en diferencias finitas.

Métodos de secante

Los métodos de secante toman dos puntos,  $x^p$  y  $x^q$  y resuelve una ecuación similar a la dada en el método de Newton:

$$f'(x^k) + m(x - x^k) = 0$$

donde  $m$  es la pendiente de la línea que conecta  $x^p$  con  $x^q$ . dada por:

$$m = \frac{f'(x^q) - f'(x^p)}{x^q - x^p}$$

El método de la secante aproxima la segunda derivada por una línea recta. Cuando  $x^q \rightarrow x^p$  el valor de  $m$  se aproximará al valor de la segunda derivada. En este sentido el método de la secante se podría considerar también un método cuasi Newton. Admitiendo que la función es unimodal, el método de la siguiente comienza con dos puntos cualquiera del intervalo de tal manera que la primera derivada tenga signos diferentes. Calculando el cero de la ecuación de partida se obtiene:

$$\tilde{x}^* = x^q - \frac{f'(x^q)(x^q - x^p)}{[f'(x^q) - f'(x^p)]}$$

Los dos puntos seleccionados para el paso siguiente son  $\tilde{x}^*$  y  $x^p$  ó  $x^q$  dependiendo de los signos de  $f'(x^p)$  y de  $f'(x^q)$  con respecto al de  $f'(\tilde{x}^*)$ .

El método de la secante parece bastante “crudo” pero funciona bastante bien en la práctica. El orden de convergencia para funciones de una sola variable es de  $(1 + \sqrt{5}) / 2 \approx 1.6$ . Su convergencia es ligeramente menor que la del método de Newton de diferencias finitas, pero en muchas ocasiones funciona mejor que este en lo que a número de evaluaciones de la función objetivo se refiere.

#### 4.2. Búsqueda Unidireccional: Métodos de eliminación de región

Los métodos de eliminación de región para una búsqueda unidimensional se basan en eliminar una región, en cada etapa, del intervalo en el que está comprendido el mínimo. Cuando la región posible es suficientemente pequeña la búsqueda termina.

El elemento básico dentro del sistema de eliminación de regiones es la comparación de valores de  $f(x)$  en dos o más puntos dentro del intervalo de  $x$ . Debemos asumir que  $f(x)$  es unimodal, y que tiene un mínimo (es convexa) dentro de  $[a, b]$ .

Si partimos de dos puntos de test sean  $x_1, x_2$  deberíamos elegirlos de tal manera que la búsqueda fuera lo más eficiente posible. Si usamos un espaciado igual, esto es  $x_1 - a = b - x_2 = x_2 - x_1$  el método de búsqueda se llama '*búsqueda de dos puntos a intervalos iguales*'. El intervalo de incertidumbre se reduce en  $1/3$  en cada iteración. Así si  $L^0$  es la longitud del intervalo original  $(b-a)$  y  $L^k$  es el intervalo después de  $k$  iteraciones, como en cada iteración se llevan a cabo dos evaluaciones de la función objetivo, entonces  $L^k$  tras  $k$  iteraciones viene dado por:

$$L^k = \left(\frac{2}{3}\right)^k L^0$$

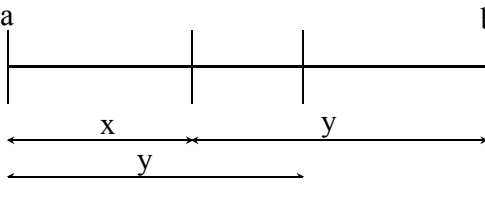
Un método más eficiente usa  $x_1 - a = b - x_2$  pero  $x_1$  y  $x_2$  están colocados muy cerca uno del otro (idealmente una distancia infinitesimal). Este método es el '*método de la bisección o búsqueda dicotómica*'. La incertidumbre del intervalo después de  $k$  iteraciones vendrá dada por:

$$L^k = \left(\frac{1}{2}\right)^k L^0$$

Sin embargo los métodos más eficientes de búsqueda por eliminación de regiones son los métodos de Fibonacci y de la sección áurea, los cuales usan una relación constante para dividir el intervalo en segmentos. Pasaremos a discutir el método de la sección áurea que viene de los números de Fibonacci.



La estrategia empleada en el método de la sección áurea es localizar dos puntos interiores del intervalo eliminado en cada iteración de tal manera que se cumpla la relación:

$$\frac{x+y}{y} = \frac{y}{x}$$


Además solamente una nueva evaluación se realiza en cada etapa de búsqueda. Para calcular la relación dentro del método de la sección áurea tenemos que resolver:

$$x + y = 1$$

$$\frac{1}{y} = \frac{y}{x}$$

Esto nos lleva a que:

$$x = \frac{3 - \sqrt{5}}{2} = F_s \approx 0.382 \quad ; \quad y = 1 - \frac{3 - \sqrt{5}}{2} = F_B \approx 0.618$$

Para un determinado intervalo  $L^k$  se aplican las fracciones  $F_s$  y  $F_B$  para calcular las distancias apropiadas. Si  $a^k$  y  $b^k$  son los extremos del intervalo en la etapa  $k$  de búsqueda los dos nuevos puntos interiores estarán localizados en

$$x_1^{k+1} = a^k + F_s L^k$$

$$x_2^{k+1} = b^k - F_s L^k$$

Nótese en cada nueva iteración uno de los nuevos puntos estará fijo de la iteración anterior. Así después de  $k$  etapas la longitud del intervalo será:

$$L^k = (0.618)^k L^0$$

Donde se han llevado a cabo un total de  $k+1$  evaluaciones de la función objetivo.

### 4.3. Búsqueda Unidireccional: Métodos de aproximación polinómica

Otra clase de métodos de minimización unidimensional localizan un punto  $x$  cercano a  $x^*$ , el valor de la variable independiente correspondiente al mínimo de  $f(x)$ , por interpolación y extrapolación utilizando aproximaciones polinómicas a  $f(x)$ . Se han propuesto tanto aproximaciones cuadráticas como cúbicas usando tanto los valores de la función solamente como los de sus derivadas primeras. Se ha comprobado que los métodos de interpolación polinómica son normalmente ligeramente mejores que el método de la sección áurea.

#### *Interpolación cuadrática*

Si comenzamos con tres puntos, por ejemplo  $x_1, x_2, x_3$  en orden creciente, y no necesariamente igualmente espaciados, pero contenidos dentro de la zona de búsqueda  $(a,b)$ , podemos aproximarlos a un polinomio de grado 2,  $f(x) = a + bx + cx^2$  de tal manera que dicho polinomio pasa exactamente por esos tres puntos y debe presentar un mínimo en:

$$\tilde{x}^* = -\frac{b}{2c}$$

Si suponemos que  $f(x)$  se evalúa en los tres puntos, podríamos calcular los valores de  $a, b, c$  resolviendo el sistema de tres ecuaciones lineales:

$$f(x_1) = a + bx_1 + cx_1^2$$

$$f(x_2) = a + bx_2 + cx_2^2$$

$$f(x_3) = a + bx_3 + cx_3^2$$

lo que nos lleva a :

$$\tilde{x}^* = \frac{1}{2} \left[ \frac{(x_2^2 - x_3^2)f_1 + (x_3^2 - x_1^2)f_2 + (x_1^2 - x_2^2)f_3}{(x_2 - x_3)f_1 + (x_3 - x_1)f_2 + (x_1 - x_2)f_3} \right]$$

con  $f_1 \equiv f(x_1)$ ;  $f_2 \equiv f(x_2)$ ;  $f_3 \equiv f(x_3)$

Para ilustrar la primera etapa del procedimiento de búsqueda examinaremos la siguiente figura. Señalar que solamente una nueva evaluación de la función objetivo se lleva a cabo en cada etapa de búsqueda.

- I.- Si  $x^*$  cae entre  $x_2$  y  $x_3$
- (a)  $f^* < f_2$   
 $f^* < f_3$  elegir  $x_2, x^*, x_3$
- (b)  $f^* > f_2$   
 $f^* < f_3$  elegir  $x_1, x_2, x^*$

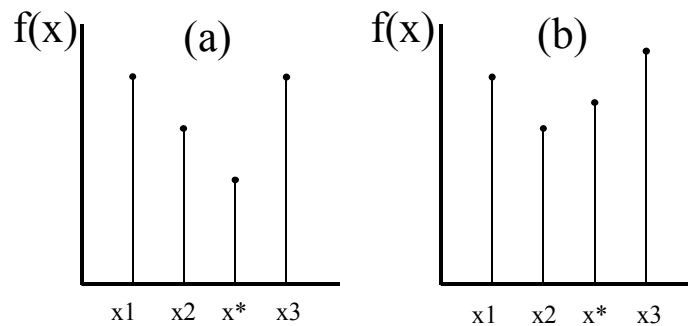


Figura 1. Selección de la región a eliminar

Si el óptimo cae entre  $x_1$  y  $x_2$  se procede de forma análoga al caso anterior. La búsqueda termina cuando se ha alcanzado la precisión deseada.

## 5. Métodos directos

Los métodos directos no hacen uso de la información proporcionada por las derivadas. Bajo estas circunstancias, estos métodos se pueden usar con bastante efectividad, pero son muy ineficientes comparados con los métodos basados en derivadas. Tienen la ventaja de que estos métodos son muy simples de entender y muy fáciles de programar.

### 5.1. Métodos de búsqueda aleatoria

Un método aleatorio simplemente selecciona un vector inicial  $\mathbf{x}^0$ , evalúa la función objetivo en ese punto y entonces aleatoriamente selecciona otro vector  $\mathbf{x}^1$ . Tanto la dirección de búsqueda como la longitud de búsqueda son elegidas simultáneamente. Después de una o más etapas, el valor de  $f(\mathbf{x}^k)$  se compara con el mejor valor previo de  $f(\mathbf{x})$  y se toma la decisión de continuar o terminar el procedimiento. Existen diversas variaciones de este algoritmo, aunque estrictamente hablando sólo se alcanza la solución cuando  $k \rightarrow \infty$ , pero desde un punto de vista práctico, si el objetivo tiene una forma muy plana se pueden encontrar soluciones subóptimas bastante aceptables. Aunque el método es bastante ineficiente por si mismo, puede dar valores aceptables de partida para otros métodos.

### 5.2. Métodos de busca en rejilla.

Los métodos básicos de diseño de experimentos discutidos en muchos textos de estadística, se pueden aplicar también a minimización de funciones. Se pueden seleccionar una serie de puntos alrededor de un punto base de referencia, de acuerdo a algunos de los diseños del tipo que se muestra en la siguiente figura. Después se pasa al punto que más mejora la función objetivo y se continua la búsqueda. Sin embargo el sistema es muy ineficaz, por ejemplo con  $n=10$  y una búsqueda factorial a tres niveles deberíamos realizar  $3^{10}-1=59048$  evaluaciones de la función objetivo, lo cual es obviamente prohibitivo.

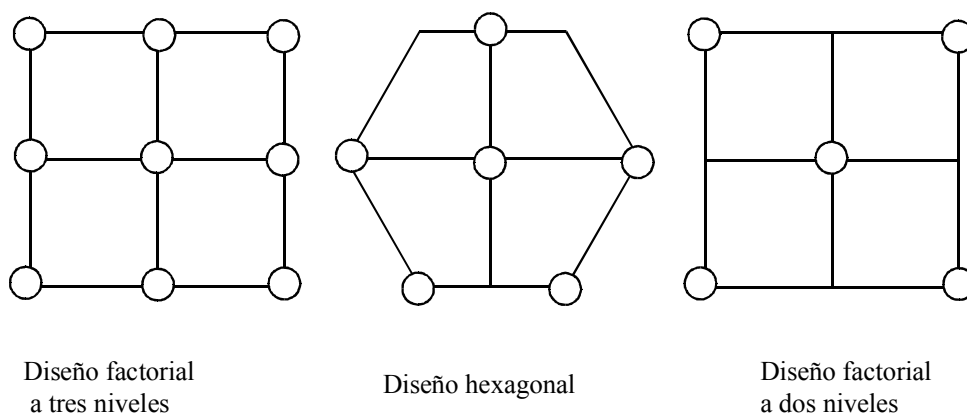


Figura 2. Alternativas para la selección de puntos en los métodos de búsqueda en rejilla.

### 5.3. Búsqueda Univariante

Otro método muy sencillo de optimización consiste en seleccionar  $n$  direcciones fijas de búsqueda, para  $n$  variables, (habitualmente los ejes coordenados) de tal manera que  $f(\mathbf{x})$  se minimiza de forma secuencial usando búsquedas unidimensionales en cada una de las direcciones previamente seleccionadas. El método suele ser bastante ineficaz incluso llegando a puntos muy alejados del óptimo de los cuales no puede salir.

### 5.4. Método simplex flexible.

Este método se basa en tomar una figura regular (conocida como simplex) como base. Así en 2 dimensiones tal figura debería ser un triángulo equilátero. Los experimentos se localizan de tal manera que la función objeto se evalúa en cada uno de los vértices que forman la figura geométrica.

Los valores de la función objetivo obtenida en cada uno de los vértices se comparan entre sí rechazando el peor valor de todos formando una nueva figura geométrica por reflexión del peor de los puntos respecto a los que no han sido rechazados. En el simplex original se mantiene la figura geométrica, mientras que la modificación de Neadler y Mead permite distorsiones de ésta para acelerar el proceso de búsqueda.

Veamos de forma sistemática cada uno de los pasos del simplex:

#### Selección de la figura "simplex" inicial:

Para visualizarlo de forma sencilla comenzaremos con el caso de dos dimensiones. Así la distancia entre un punto  $\mathbf{x}_1=(x_{11}, x_{12})$  y otro  $\mathbf{x}_2=(x_{21}, x_{22})$  viene dada por la expresión:

$$(\mathbf{x}_1 - \mathbf{x}_2)^2 = (x_{11} - x_{21})^2 + (x_{12} - x_{22})^2 = a^2$$

procediendo de tal manera las distancias entre dos puntos vendrán dadas por una expresión similar.

$$(x_{1j} - x_{1k})^2 + (x_{2j} - x_{2k})^2 = \sum_{i=1}^2 (x_{ij} - x_{ik})^2 = a^2 \quad j \neq k$$

como tenemos tres puntos podemos hacer  $C_2^3 = \frac{3!}{2!1!} = 3$  así en dos dimensiones (3 vértices) podemos definir tres distancias entre vértices (todas ellas iguales). Por la tanto, especificando un punto base  $\mathbf{x}_1$  el punto  $\mathbf{x}_2$  estará localizado en un punto cualquiera de una circunferencia de radio 'a' centrada en  $\mathbf{x}_1$ . Una vez elegido el punto  $\mathbf{x}_2$ , el punto  $\mathbf{x}_3$  deberá estar en la intersección entre las circunferencias de radio 'a' centradas en  $\mathbf{x}_1$  y en  $\mathbf{x}_2$  respectivamente. Existen dos posibilidades.

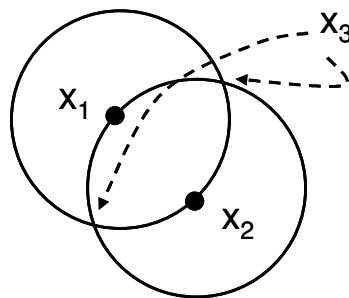


Figura 3.- Generación de la figura simplex inicial

De forma similar en n dimensiones tenemos n+1 puntos. La tabla siguiente es una propuesta para comenzar un simplex de n variables cuando  $\mathbf{x}_1$  es el origen.

Punto j	$x_{1j}$	$x_{2j}$	$x_{3j}$	$x_{4j}$	$x_{n-1j}$	$x_{nj}$
1	0	0	0	0	0	0
2	p	q	q	q	q	q
3	q	p	q	q	q	q
n	q	q	q	q	p	q



Si en lugar del origen se toma cualquier otro punto  $x_1$  basta realizar una traslación.

Se puede encontrar otra serie de puntos, pero este es uno de los más simples para hallar p y q y la condición de que formen parte de la figura simplex, se puede aplicar a cada par de puntos; así entre los puntos 1 y 2:

$$(0-p)^2 + (n-1)(0-q)^2 = a^2$$

$$p^2 + (n-1)q^2 = a^2$$

Entre los puntos 2 y 3:

$$2(p-q)^2 + (n-1)(q-q)^2 = a^2$$

$$2(p-q)^2 = a^2$$

Resolviendo el sistema:

$$p = \frac{a}{n\sqrt{2}}(n-1 + \sqrt{n+1})$$

$$q = \frac{a}{n\sqrt{2}}(-1 + \sqrt{n+1})$$

Una vez fijada la figura inicial del simplex, se sigue una búsqueda secuencial, en la que en cada paso se eliminará un vértice y se incluirá otro nuevo. (supondremos que queremos minimizar).

Paso 0:

Calcular la función objetivo en cada uno de los puntos (vértices), hallar el vértice con el valor máximo ' $Q_{\max}$ ' el vértice con valor mínimo ' $Q_{\min}$ ' y el centroide de todos los vértices excepto aquel que ha dado el peor valor:

$$x_{j,B} = \frac{1}{n} \sum_{i=1}^n x_i$$

El procedimiento a seguir a continuación consiste en calcular un nuevo vértice y eliminar aquel que dio un peor valor. ( $Q_{max}$ )

Paso 1: Reflexión:

Se calcula un nuevo vértice  $\vec{Q}^*$  cuyas coordenadas son:

$$\vec{Q}^* = (1 + \gamma_r) \vec{B} - \gamma_r Q_{max}$$

donde  $\gamma_r$  es el llamado coeficiente de reflexión, una cantidad positiva que generalmente suele ser la unidad. Realmente lo que hemos hecho ha sido reflejar el punto con el peor valor respecto del centroide. Las siguientes figuras lo ilustran para el caso de 2 y 3 variables.

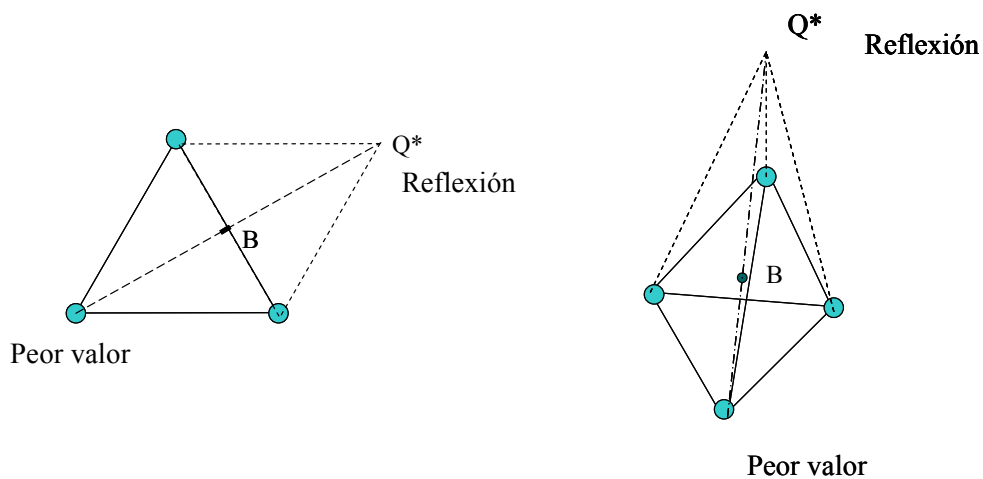


Figura 4.- Reflexión en el algoritmo simplex flexible.

Después de la reflexión pueden ocurrir tres posibilidades:

- (a) Si  $Q_{min} < Q^* < Q_{max}$  se reemplaza el vértice  $Q_{max}$  por  $Q^*$ , se recalcula  $Q_{max}$  y B (centroide) y se vuelve a la etapa 1
- (b) Si  $Q^* < Q_{min}$  se extiende  $Q^*$  hasta  $Q^{**}$  (expansión)

$$\vec{Q}^{**} = \gamma_e \vec{Q}^* + (1 - \gamma_e) \vec{B}$$

donde  $\gamma_e$  es un coeficiente de expansión, generalmente 2.



**b1.-** Si  $Q^* < Q_{\min}$  se reemplaza  $Q_{\max}$  por  $Q^{**}$  y se vuelve al punto 1 (recalcular  $Q_{\max}$ ,  $Q_{\min}=Q^{**}$ , B).

**b2.-** Si  $Q^{**} > Q_{\min}$  se reemplaza  $Q_{\max}$  por  $Q^*$  y se vuelve al punto 1 (recalcular  $Q_{\max}$ ,  $Q_{\min}=Q^{**}$ , B).

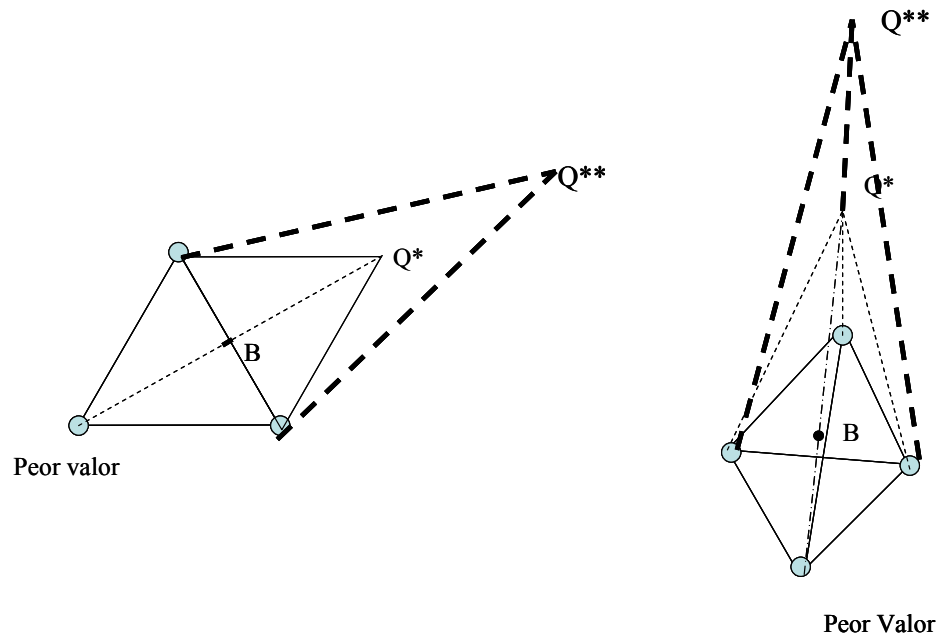


Figura 5.-Expansión en el algoritmo simples flexible

(c) Si  $Q^* > Q_i$  excepto de  $Q_{\max}$  se lleva a cabo una contracción obteniéndose un nuevo punto  $Q^{***}$ :

$$Q^{***} = \gamma_c Q^* + (1 - \gamma_c) \vec{B}$$

donde  $\gamma_c$  es un coeficiente de contracción, generalmente 0.5. Y se vuelve al punto 1, excepto en el caso que el nuevo vértice sea peor que  $Q_{\max}$  en cuyo caso se sigue con c1.

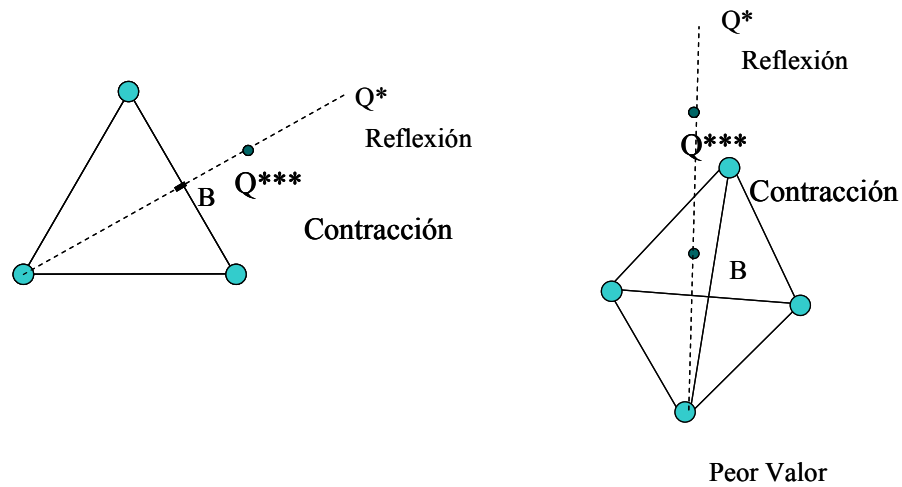


Figura 6.- Contracción en el algoritmo simples flexible

(c1). Cambio todos los vértices excepto  $Q_{\min}$ . Se genera una nueva figura dividiendo a la mitad la distancia de todos los vértices a  $Q_{\min}$ .

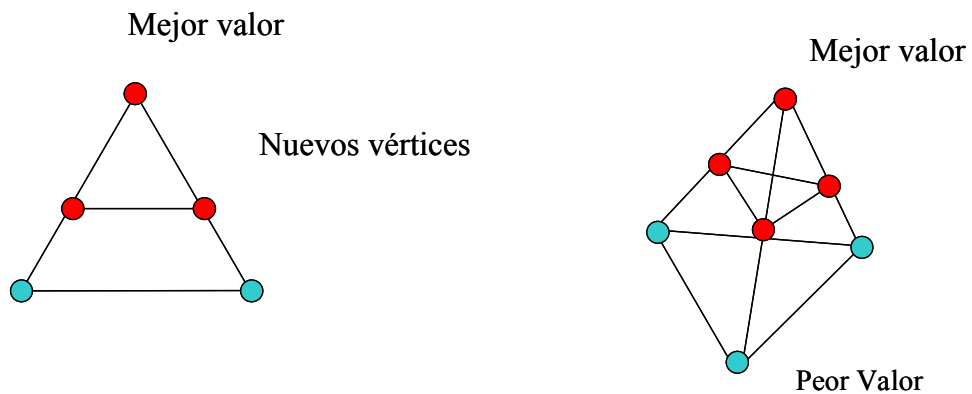


Figura 7.- Cambio de vértices y reducción de la figura simplex.

Se vuelve entonces a la etapa 1.

### 5.5. Direcciones conjugadas. Método de Powell

La experiencia ha demostrado que las direcciones llamadas conjugadas son mucho más efectivas como direcciones de búsqueda que otras como pueden ser la búsqueda univariante o las direcciones ortogonales.

Dos direcciones  $\mathbf{s}^i$  y  $\mathbf{s}^j$  se dice que son *conjugadas* una con respecto a la otra si:

$$(\mathbf{s}^i)^T \mathbf{Q} (\mathbf{s}^j) = 0$$

En general un conjunto de  $n$  direcciones de búsqueda linealmente independientes  $\mathbf{s}^0, \mathbf{s}^1, \mathbf{s}^2, \dots, \mathbf{s}^{n-1}$  se dice que son conjugadas con respecto a una matriz definida positiva  $\mathbf{Q}$  si

$$(\mathbf{s}^i)^T \mathbf{Q} (\mathbf{s}^j) = 0 \quad 0 \leq i \neq j \leq n-1$$

En optimización la matriz  $\mathbf{Q}$  es la matriz hessiana de la función objetivo,  $\mathbf{H}$ . Para una función cuadrática  $f(\mathbf{x})$  de  $n$  variables, para la cual  $\mathbf{H}$  es una matriz constante está garantizado que se obtiene el óptimo en  $n$  etapas de búsqueda unidireccional si se obtiene exactamente el mínimo de cada etapa. Una dirección conjugada en general no es una dirección única. Sin embargo, en dos dimensiones, si se elige una dirección  $\mathbf{s}^1$  y  $\mathbf{Q} \mathbf{s}^2$  queda completamente especificada.

La ortogonalidad es un caso especial de la conjugación cuando  $\mathbf{Q}=\mathbf{I}$ . Aunque es corriente encontrar en la bibliografía conjuntos de métodos conocidos como de direcciones conjugadas estas, estrictamente hablando sólo existen para funciones cuadráticas o aproximaciones cuadráticas de la función objetivo en la etapa  $k$ .

¿Cómo se puede calcular las direcciones conjugadas sin usar derivadas? Este es un concepto básico que lo referiremos a la siguiente figura. Comenzamos con el punto  $\mathbf{x}^0$ . Localizamos el punto  $\mathbf{x}^a$  que es el mínimo en una dirección cualquiera  $\mathbf{s}$ . Elegimos otro punto cualquiera (distinto del primero)  $\mathbf{x}^1$ . y localizamos el punto  $\mathbf{x}^b$  que es el mínimo partiendo del punto  $\mathbf{x}^1$  en la misma dirección  $\mathbf{s}$ . Los puntos  $\mathbf{x}^a$  o  $\mathbf{x}^b$  se obtienen minimizando  $f(\mathbf{x}^0 + \lambda \mathbf{s})$  con respecto a  $\lambda$ . Si  $f(\mathbf{x})$  es una función cuadrática:

$$f(\mathbf{x}) = f(\mathbf{x}^0) + \nabla^T f(\mathbf{x}^0) \Delta \mathbf{x}^0 + \frac{1}{2} (\Delta \mathbf{x}^0)^T \mathbf{H} (\Delta \mathbf{x}^0)$$

Se puede demostrar que el óptimo cae en la línea que une  $\mathbf{x}^a$  con  $\mathbf{x}^b$ . Sin embargo esta última afirmación no es válida para otro tipo de funciones.

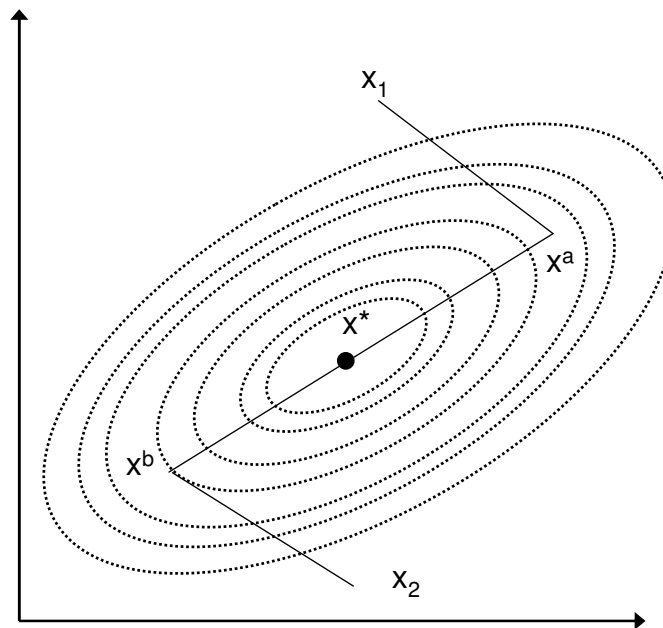


Figura 8.-Ilustración del cálculo de direcciones conjugadas. El óptimo cae en la línea que une  $x^a$  con  $x^b$

Powell desarrolló un método basado en las direcciones conjugadas aplicable a cualquier tipo de funciones. Aunque Powell introdujo importantes modificaciones a su método para conseguir la adaptación. El método sigue una serie de pasos:

*Paso 1.-* El método comienza realizando una búsqueda en  $n$  direcciones linealmente independientes  $(s_1^0, s_2^0, s_3^0, \dots, s_n^0)$  que suelen tomarse paralelos a los ejes coordenados. Partiendo del punto base  $x_0^0$  se lleva a cabo una búsqueda unidimensional en la dirección  $s_1^0$  para llegar al punto  $x_1^0$ . Este punto ( $x_1^0$ ) se toma como punto de partida para una nueva búsqueda unidireccional, en este caso en la dirección  $s_2^0$ , y así sucesivamente hasta acabar en el punto  $x_n^0$  (La figura ilustra el caso para dos dimensiones).

*Paso 2.-* Buscamos el punto particular  $x_k^0$  para el cual se ha obtenido una mejoría mayor de la función objetivo respecto al punto anterior  $x_{k-1}^0$ . Definimos dos magnitudes:

$$\Delta^k = [f(x_{k-1}^0) - f(x_k^0)]$$

$$\mu = x_0^0 - x_n^0$$

*Paso 3.-* Determinamos:

$$f_t^0 = f(2\mathbf{x}_n^0 - \mathbf{x}_0^0)$$

y llevamos a cabo dos comparaciones.

Si  $f_t^0 \geq f(\mathbf{x}_0^0)$  y/o

$$(f(\mathbf{x}_0^0) - 2f(\mathbf{x}_n^0) + f_t^0)(f(\mathbf{x}_0^0) - f(\mathbf{x}_n^0) - \Delta) \geq \frac{\Delta(f(\mathbf{x}_0^0) - f_t^0)}{2}$$

Entonces la dirección  $\mu$  no es una buena dirección de búsqueda y repetiríamos la búsqueda comenzando desde el punto  $\mathbf{x}_n^0$  como punto base. En caso contrario se procede a incorporar la dirección  $\mu$  al conjunto de direcciones de búsqueda, sustituyendo a la dirección que peor resultado hubiese obtenido.

En la nueva etapa de búsqueda conviene que la última dirección investigada (en la etapa de búsqueda unidireccional) sea  $\mu$ .

Las dos desigualdades anteriores comprueban, la primera si se obtiene una mejora en la dirección al pasar del punto  $\mathbf{x}_0^0$  al punto  $\mathbf{x}_n^0$ , y la segunda, que la función descienda de manera pronunciada y no a través de una zona plana. LA siguiente figura ilustra la aplicación del método de Powell.

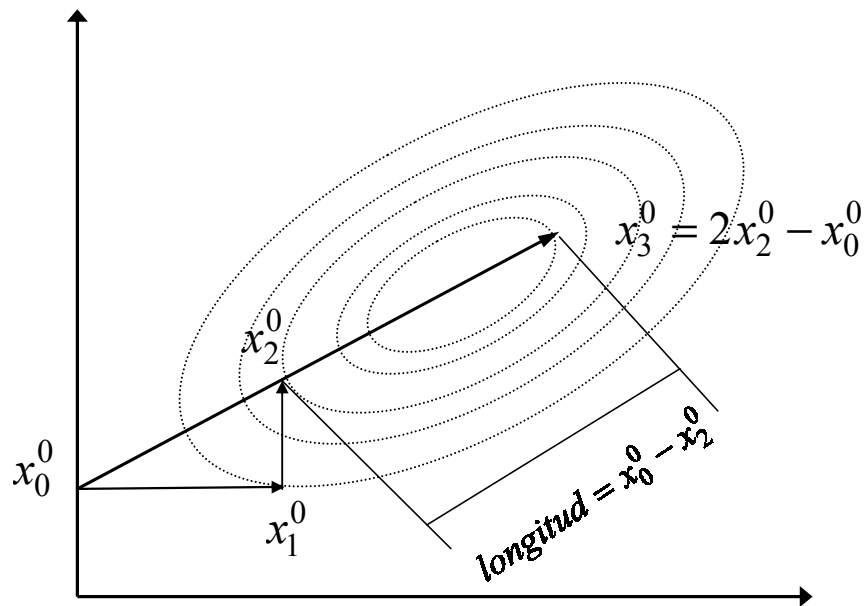


Figura 9.- Ilustración gráfica del método de Powell.

## 6. MÉTODOS INDIRECTOS: MÉTODOS DE PRIMER ORDEN.

Los métodos indirectos, en contraste con los métodos descritos en las secciones previas hacen uso de las derivadas en la determinación de la dirección de búsqueda. Sin embargo, nuestra clasificación en métodos directos e indirectos, podría no estar clara del todo debido a la aproximación de las derivadas por diferencias finitas, lo que estrictamente hablando hace a estos métodos 'libres de derivadas'. Una buena dirección de búsqueda debería reducir (para minimización) la función objetivo, de tal manera que si  $\mathbf{x}^0$  es el punto inicial y  $\mathbf{x}^1$  es un nuevo punto:

$$f(\mathbf{x}^1) < f(\mathbf{x}^0)$$

Una dirección  $\mathbf{s}$  es llamada de descenso si satisface la siguiente relación en un punto dado:

$$\nabla^T f(\mathbf{x}) \mathbf{s} < 0$$

### 6.1. Método del Gradiente (Máximo descenso)

Se recordará que el gradiente es un vector en un punto  $\mathbf{x}$  que proporciona la dirección (local) de máxima variación de la función. El vector gradiente es un vector ortogonal al contorno de la función en el punto. Por lo tanto en la búsqueda de un mínimo la dirección de movimiento será contra-gradiente:

$$\mathbf{s}^k = -\nabla f(\mathbf{x})$$

En el método de máximo descenso la transición de un punto  $\mathbf{x}^k$  a otro  $\mathbf{x}^{k+1}$  viene dada por la siguiente expresión:

$$\mathbf{x}^{k+1} = \mathbf{x}^k + \Delta \mathbf{x}^k = \mathbf{x}^k + \lambda^k \mathbf{s}^k = \mathbf{x}^k - \lambda^k \nabla f(\mathbf{x}^k)$$

donde  $\Delta \mathbf{x}^k$  = Vector desde  $\mathbf{x}^k$  hasta  $\mathbf{x}^{k+1}$

$\mathbf{s}^k$  = Dirección de búsqueda de máximo descenso

$\lambda^k$  = Escalar que determina la longitud de paso en la dirección  $\mathbf{s}^k$

El gradiente negativo da la dirección de movimiento, pero no la longitud de dicho movimiento. Por lo tanto existen varios procedimientos posibles dependiendo de la elección de  $\lambda^k$ . Entre los diversos métodos que existen para la selección de la longitud de paso, dos merecen una mención especial. El primero de ellos emplea una búsqueda unidireccional en la dirección del gradiente. El segundo especifica a priori la longitud del paso para cada iteración. Claramente la principal dificultad con la segunda aproximación es que a menudo no se sabe a priori la longitud de paso adecuada. ¿Cómo se debería cambiar esta longitud de paso y como debería reducirse a medida que nos aproximamos al mínimo para conseguir la convergencia? una ilustración nos indicará el problema: Supongamos una función cuadrática de dos variables como la de la figura siguiente:

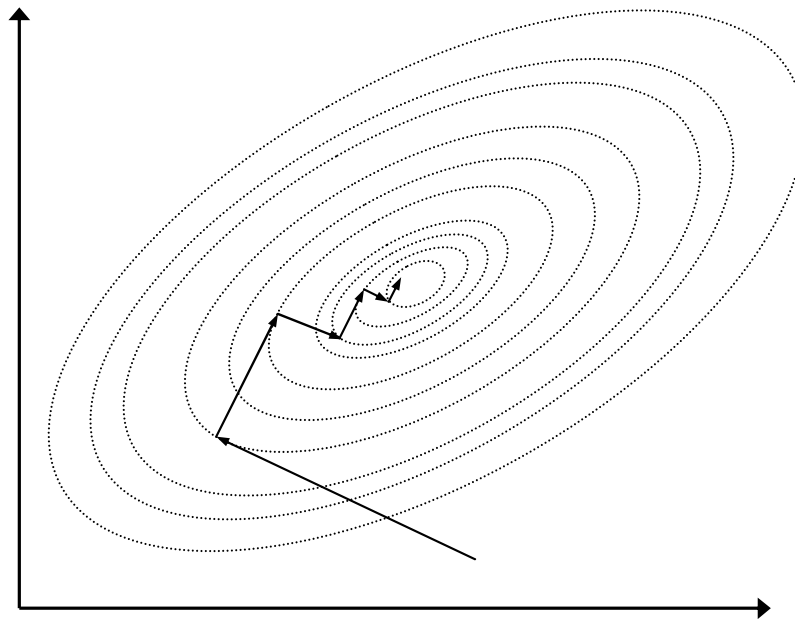


Figura 10.- Oscilación del método de máximo descenso

Si en cada paso se realiza una optimización total en la dirección contraria al gradiente los pasos sucesivos del método de máximo descenso son ortogonales uno con respecto al anterior. Este resultado, que parece peculiar, ocurre para una determinada  $f(\mathbf{x})$  debido a que la derivada de  $f(\mathbf{x})$  a lo largo de la línea  $\mathbf{s}(\lambda)$  viene dado, utilizando la regla de la cadena por:

$$\frac{df}{d\lambda} = \sum_i \frac{d}{d\lambda} x_i(\lambda) \frac{df}{dx_i} = \sum_i s_i \frac{\partial f}{\partial x_i} = \mathbf{s}^T \nabla f$$

en el paso final de la búsqueda queremos que  $\frac{df}{dx_i} = 0$  y por lo tanto  $\mathbf{s}^T \nabla f(x^{k+1}) = 0$ . En la práctica, por lo tanto no es deseable llevar a cabo suboptimizaciones con demasiada precisión. Mientras que el método del gradiente puede producir progresos muy satisfactorios en la reducción de  $f(\mathbf{x})$  en la primeras iteraciones tiende a hacerse muy lento en las últimas. Alargando excesivamente los cálculos.

El algoritmo práctico lo podemos resumir en los siguientes pasos:

- 1.- Elegir un valor inicial  $\mathbf{x}^0$ . En pasos sucesivos será  $\mathbf{x}^k$ .



2.- Calcular, analítica o numéricamente las derivadas parciales

$$\frac{\partial f(\mathbf{x})}{\partial x_j} \quad j = 1, 2, 3, \dots, n$$

3.- Calcular el vector de búsqueda

$$\mathbf{s} = -\nabla f(\mathbf{x}^k)$$

4.- Usar la relación  $\mathbf{x}^{k+1} = \mathbf{x}^k + \lambda^k \mathbf{s}^k$  para obtener el siguiente punto. El valor de  $\lambda^k$  puede ser de valor fijo o calculado en cada paso mediante una búsqueda unidireccional.

5.- Comparar  $f(\mathbf{x}^{k+1})$  con  $f(\mathbf{x}^k)$ . Si el cambio es menor que una tolerancia pre-especificada terminar, en caso contrario volver al paso dos y continuar con las iteraciones.

Un método estricto de descenso máximo puede terminar en cualquier punto estacionario, es decir, puede llegar a un mínimo local o a un punto de silla. Para asegurarnos que tipo de resultado hemos obtenido debemos asegurarnos que la matriz Hessiana es definida positiva. Por otra parte la dificultad básica del método de gradiente es que es muy sensible al escalado de  $f(\mathbf{x})$  por lo que la convergencia puede ser muy lenta y producirse un número enorme de oscilaciones. Desde este punto de vista el método del gradiente no es muy efectivo.

### 6.2. Método del gradiente conjugado

El método del gradiente conjugado debido a Fletcher y Reeves (1964) combina las características de la convergencia cuadrática del método de las direcciones conjugadas con las del método del gradiente. El método supone una importante mejora del método del gradiente con sólo un pequeño incremento en el esfuerzo de cálculo. El método del gradiente conjugado, esencialmente, combina la información obtenida del vector gradiente con la información acerca del vector gradiente de iteraciones previas. Lo que hace el método es calcular la nueva

dirección de búsqueda utilizando una combinación lineal del gradiente en la etapa considerada y el de la etapa anterior. La principal ventaja del método es que necesita almacenar muy poca cantidad de información con lo que puede ser programado fácilmente incluso en calculadoras.

Los pasos de cálculo se comentan a continuación:

- 1.- En  $\mathbf{x}^0$  (punto inicial) calcular  $f(\mathbf{x}^0)$  y calcular  $\mathbf{s}^0 = -\nabla f(\mathbf{x}^0)$
- 2.- Almacenar  $\nabla f(\mathbf{x}^0)$  y calcular  $\mathbf{x}^1 = \mathbf{x}^0 + \lambda^0 \mathbf{s}^0$  minimizando  $\lambda$  mediante una búsqueda unidireccional en la dirección  $\mathbf{s}^0$ .
- 3.- Calcular  $f(\mathbf{x}^1)$   $\nabla f(\mathbf{x}^1)$  la nueva dirección de búsqueda es una combinación lineal de  $\mathbf{s}^0$  y  $\nabla f(\mathbf{x}^1)$ :

$$\mathbf{s}^1 = -\nabla f(\mathbf{x}^1) + \mathbf{s}^0 \frac{\nabla^T f(\mathbf{x}^1) \nabla f(\mathbf{x}^1)}{\nabla^T f(\mathbf{x}^0) \nabla f(\mathbf{x}^0)}$$

para la etapa k-ésima la relación es:

$$\mathbf{s}^{k+1} = -\nabla f(\mathbf{x}^{k+1}) + \mathbf{s}^k \frac{\nabla^T f(\mathbf{x}^{k+1}) \nabla f(\mathbf{x}^{k+1})}{\nabla^T f(\mathbf{x}^k) \nabla f(\mathbf{x}^k)}$$

Para una función cuadrática se puede demostrar que dos direcciones de búsqueda son conjugadas. Después de n iteraciones conviene comenzar otra vez desde el principio tomando el último punto  $k=n$  como nuevo punto de partida.

- 4.- Realizar el test de convergencia, (la función objetivo ha disminuido), y terminar el algoritmo cuando  $\|\mathbf{s}^k\|$  sea menor que alguna tolerancia preestablecida.

Algunas de las dificultades que aparecen en el método de Powell también aparecen en el método del gradiente conjugado. Se puede producir una

dependencia lineal de las direcciones de búsqueda. Esto se puede eliminar reiniciando el proceso cada cuatro o cinco etapas completas.

### 7. MÉTODOS INDIRECTOS: MÉTODOS DE SEGUNDO ORDEN.

Desde el punto de vista de las direcciones de búsqueda el método del máximo descenso se puede interpretar como un movimiento ortogonal a una aproximación lineal (tangente) a la función objetivo en el punto  $\mathbf{x}^k$ . Examinemos ahora la siguiente figura y realicemos una aproximación cuadrática de  $f(\mathbf{x})$  en  $\mathbf{x}^k$ .

$$f(\mathbf{x}) \approx f(\mathbf{x}^k) + \nabla^T f(\mathbf{x}^k) \Delta \mathbf{x}^k + \frac{1}{2} (\Delta \mathbf{x}^k)^T \mathbf{H}(\mathbf{x}^k) \Delta \mathbf{x}^k$$

donde  $\mathbf{H}(\mathbf{x})$  es la matriz Hessiana evaluada en el punto  $\mathbf{x}^k$ . Por lo tanto es posible tener en cuenta la curvatura de la función en las proximidades del punto de forma análoga a como se hacía en el método de Newton.

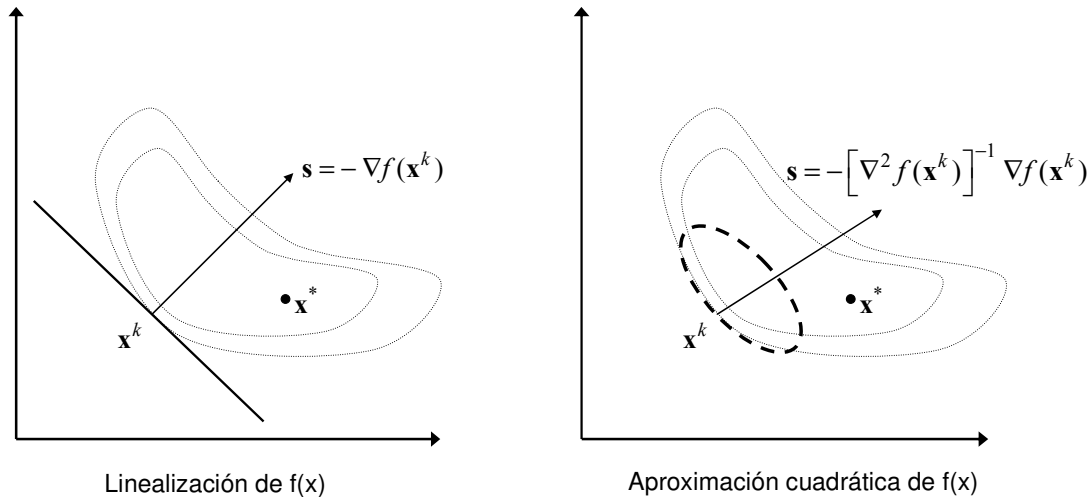


Figura 11.- Aproximación lineal y cuadrática de una función

### 7.1. El Método de Newton.

El método de Newton hace uso de la aproximación de segundo orden de la función utilizando las derivadas segundas con respecto a cada una de las variables independientes. De esta forma es posible tener en cuenta la curvatura de la función en el punto e identificar las mejores direcciones de búsqueda.

El mínimo de  $f(x)$  se obtiene diferenciando la aproximación cuadrática de  $f(x)$  con respecto a cada una de las variables e igualando a cero. Así:

$$\nabla f(\mathbf{x}) = \nabla f(\mathbf{x}^k) + \mathbf{H}(\mathbf{x}^k) \Delta \mathbf{x}^k$$

o bien

$$\mathbf{x}^{k+1} - \mathbf{x}^k = \Delta \mathbf{x}^k = -[\mathbf{H}(\mathbf{x}^k)]^{-1} \nabla f(\mathbf{x}^k)$$

Si  $f(x)$  es una función cuadrática el mínimo se alcanza en un único paso. Pero para una función general no lineal el óptimo no se alcanza en un único paso, por lo que se puede modificar la ecuación anterior para introducir el parámetro de longitud de paso, con lo que queda:

$$\mathbf{x}^{k+1} - \mathbf{x}^k = \Delta \mathbf{x}^k = -\lambda^k [\mathbf{H}(\mathbf{x}^k)]^{-1} \nabla f(\mathbf{x}^k)$$

Obsérvese que la dirección  $\mathbf{s}$  viene ahora dada por :

$$\mathbf{s} = -[\mathbf{H}(\mathbf{x}^k)]^{-1} \nabla f(\mathbf{x}^k)$$

La longitud de paso se puede calcular vía cualquier método de optimización numérica de los ya comentados o bien analíticamente con lo que se obtendría:

$$\lambda^{opt} = -\frac{\nabla^T f(\mathbf{x}^k) \mathbf{s}^k}{(\mathbf{s}^k)^T \mathbf{H}(\mathbf{x}^k) \mathbf{s}^k}$$

En el método de Newton estrictamente hablando el valor de  $\lambda$  es la unidad. El problema también se puede resolver sin necesidad de invertir la matriz hessiana resolviendo directamente el sistema de ecuaciones lineales en  $\Delta \mathbf{x}$ , aplicando la factorización adecuada y evitando los errores de redondeo, en la medida de lo

posible, que puedan aparecer como consecuencia del proceso de inversión de matrices.

$$\mathbf{H}(\mathbf{x}^k) \Delta \mathbf{x}^k = -\nabla f(\mathbf{x}^k)$$

El método de Newton es el método de minimización más rápido, cuando funciona bien. Pero presenta las siguientes desventajas:

- 1.- El método no encuentra necesariamente un óptimo global (pero esto es una característica de todos los métodos que hemos comentado hasta el momento)
- 2.- Requiere la inversión de matrices o la resolución de un sistema de  $n$  ecuaciones lineales
- 3.- Necesita de las derivadas primera y segunda, las cuales en la práctica no son fáciles de obtener.
- 4.- El método puede llevar a un punto de silla si la matriz hessiana no es definida positiva.

La dificultad 3 se puede aliviar utilizando métodos apropiados de diferencias finitas para el cálculo de las derivadas primera y segunda. La convergencia global se puede reforzar si, al realizar un paso de Newton propiamente dicho ( $\lambda=1$ ) no se obtiene mejoría de la función objetivo, se vuelve al punto anterior y se realiza una búsqueda con el valor de  $\lambda$  variable.

Para evitar la dificultad 4 se debe ser muy cuidadoso con la técnica que se utiliza para garantizar que la matriz hessiana o su inversa sea definida positiva. (en algunos casos debido al número de condición de la matriz la inversión de ésta podría llevar a matrices no definidas positivas). Por otra parte sólo está garantizado que el valor de la función objetivo mejora si la matriz hessiana es definida positiva.  $\mathbf{H}(\mathbf{x})$  es definida positiva sólo para funciones estrictamente convexas, pero las funciones no lineales en general la matriz  $\mathbf{H}(\mathbf{x})$  puede cambiar de punto a punto y el método de Newton podría llevarnos a direcciones que no reducen el valor de la función objetivo (o al menos no a las direcciones

óptimas). En otras palabras si los valores propios de la matriz hessiana son todos positivos sabemos que estamos aproximando nuestra función por una cuadrática de contornos circulares o elipsoidales que tienen un mínimo. Pero si al menos dos valores propios presentan signos opuestos podríamos movernos en dirección a un punto de silla en lugar de hacia el mínimo.

El método de Newton tiene la ventaja de la convergencia cuadrática sólo en la vecindad del óptimo, donde la función objetivo puede ser aproximada bastante bien por una función cuadrática. Pero en zonas más alejadas otros métodos pueden presentar velocidades de convergencia mayores.

#### *Forzando a la matriz Hessiana a ser Definida Positiva*

Marquardt, Levenverg y otros han sugerido que la matriz Hessiana de  $f(\mathbf{x})$  puede ser modificada para obligarla a ser definida positiva y bien condicionada en cada etapa de búsqueda. El procedimiento consiste en añadir elementos a la diagonal principal de  $\mathbf{H}(\mathbf{x})$ :

$$\tilde{\mathbf{H}}(\mathbf{x}) = [\mathbf{H}(\mathbf{x}) + \beta \mathbf{I}]$$

donde  $\beta$  es una constante positiva para asegurar que  $\tilde{\mathbf{H}}(\mathbf{x})$  sea definida positiva cuando  $\mathbf{H}(\mathbf{x})$  no lo es. También es posible usar

$$\left[ \tilde{\mathbf{H}}(\mathbf{x}) \right]^{-1} = [\mathbf{H}^{-1}(\mathbf{x}) + \gamma \mathbf{I}]$$

donde  $\gamma$  es un escalar de suficiente valor para el mismo propósito. Para estar seguro de que valor de  $\beta$  usar se debe calcular el valor propio más pequeño (más negativo) de la matriz hessiana en el punto y hacer  $\beta > -\min\{\alpha_1\}$ , donde  $\alpha_1$  es un valor propio de  $\mathbf{H}(\mathbf{x})$ . Remarcar que si el valor de  $\beta$  es demasiado grande la diagonal principal se hace tan dominante que el método se convierte en el de máximo descenso. El problema que aparece ahora es que en cada paso se deben calcular los valores propios de la matriz hessiana. Se han desarrollado algunos algoritmos que utilizan valores arbitrarios de dicho parámetro que se

modifican en cada paso de acuerdo a los valores previos obtenidos. Aunque el método más común consiste en utilizar una factorización de Cholesky de la matriz Hessiana.

### 7.2 Métodos de cuasi-Newton (métodos de secante)

Los métodos de secante comienzan con la misma ecuación usada en el método de Newton:

$$\nabla f(\mathbf{x}) + \mathbf{H}(\mathbf{x}^k) \Delta \mathbf{x}^k = 0$$

Pero a diferencia del método de Newton estamos interesados en conseguir una actualización de la matriz  $\mathbf{H}(\mathbf{x})$ , a la que llamaremos  $\hat{\mathbf{H}}(\mathbf{x})$  usando solamente las derivadas parciales primeras de la función, a semejanza de lo que hacíamos en los métodos de secante unidimensionales. Si queremos aproximar la matriz inversa, los métodos de secante o quasi-Newton calculan un nuevo vector  $\mathbf{x}$  a partir de otro de una etapa precedente a través de ecuaciones análogas a las del método de Newton:

$$\mathbf{x}^{k+1} - \mathbf{x}^k = -\lambda^k [\hat{\mathbf{H}}(\mathbf{x}^k)]^{-1} \nabla f(\mathbf{x}^k)$$

donde  $[\hat{\mathbf{H}}(\mathbf{x})]^{-1}$  se conoce en ocasiones como *Matriz de dirección* y representa una matriz que es una aproximación de la verdadera inversa de la hessiana. Desde este punto de vista si la matriz  $[\hat{\mathbf{H}}(\mathbf{x})]^{-1}$  es la identidad el método se convierte en el de descenso rápido.

Se han desarrollado varias fórmulas para actualizar las matrices hessianas. Supongamos por un momento que  $f(\mathbf{x})$  es una función cuadrática entonces podríamos elegir dos puntos,  $\mathbf{x}^k$  y  $\mathbf{x}^{k+1}$  y un punto de referencia  $\mathbf{x}^p$  :

$$\nabla f(\mathbf{x}^{k+1}) = \nabla f(\mathbf{x}^p) + \hat{\mathbf{H}}(\mathbf{x}^{k+1} - \mathbf{x}^p)$$

$$\nabla f(\mathbf{x}^k) = \nabla f(\mathbf{x}^p) + \hat{\mathbf{H}}(\mathbf{x}^k - \mathbf{x}^p)$$

$$\nabla f(\mathbf{x}^{k+1}) - \nabla f(\mathbf{x}^k) = \hat{\mathbf{H}}(\mathbf{x}^{k+1} - \mathbf{x}^k)$$

Para una función no cuadrática, la matriz  $\widehat{\mathbf{H}}$  se podría obtener resolviendo las ecuaciones de la secante en los puntos  $\mathbf{x}^k$  y  $\mathbf{x}^{k+1}$

$$\widehat{\mathbf{H}}(\mathbf{x}^k) \Delta \mathbf{x}^k = \Delta \mathbf{g}^k \quad \text{donde} \quad \Delta \mathbf{g}^k \equiv \nabla f(\mathbf{x}^{k+1}) - \nabla f(\mathbf{x}^k)$$

una relación equivalente sería .  $\Delta \mathbf{x}^k = \widehat{\mathbf{H}}^{-1}(\mathbf{x}^k) \Delta \mathbf{g}^k$

Debido a que tenemos solamente n ecuaciones y un total de  $n^2$  incógnitas (cada uno de los elementos de la matriz hessiana a actualizar) aparece un número infinito de posibles soluciones a las ecuaciones anteriores. Se debería elegir una matriz  $\widehat{\mathbf{H}}(\mathbf{x})$  lo más parecida posible a  $\mathbf{H}^k$  en algún determinado sentido. Se han hecho muchas sugerencias al respecto, a continuación se muestran las dos actualizaciones más utilizadas y que han demostrado dar los mejores resultados en la práctica. Ambas mantienen la matriz hessiana definida positiva y simétrica. Son las actualizaciones de Devidon-Fletcher-Powell (DFP) y la de Broyden-Fletcher-Goldfard y Shanno (BFGS).

DFP:

$$\Delta(\widehat{\mathbf{H}}^k)^{-1} = \frac{(\Delta \mathbf{x}^k)(\Delta \mathbf{x}^k)^T}{(\Delta \mathbf{x}^k)^T(\Delta \mathbf{g}^k)} - \frac{(\widehat{\mathbf{H}}^k)^{-1}(\Delta \mathbf{g}^k)(\Delta \mathbf{g}^k)^T [(\widehat{\mathbf{H}}^k)^{-1}]^T}{(\Delta \mathbf{g}^k)^T (\widehat{\mathbf{H}}^k)^{-1} (\Delta \mathbf{g}^k)}$$

$$\Delta \widehat{\mathbf{H}}^k = \frac{(\Delta \mathbf{g}^k - \widehat{\mathbf{H}}^k \Delta \mathbf{x}^k)(\Delta \mathbf{g}^k)^T + \Delta \mathbf{g}^k (\Delta \mathbf{g}^k - \widehat{\mathbf{H}}^k \Delta \mathbf{x}^k)^T}{(\Delta \mathbf{g}^k)^T (\Delta \mathbf{x}^k)} - \frac{(\Delta \mathbf{g}^k - \widehat{\mathbf{H}}^k \Delta \mathbf{x}^k)^T \Delta \mathbf{x}^k \Delta \mathbf{g}^k (\Delta \mathbf{g}^k)^T}{[(\Delta \mathbf{g}^k)^T \Delta \mathbf{x}^k] [(\Delta \mathbf{g}^k)^T \Delta \mathbf{x}^k]}$$

BFGS:

$$\Delta \widehat{\mathbf{H}}^k = \frac{\Delta \mathbf{g}^k (\Delta \mathbf{g}^k)^T}{(\Delta \mathbf{g}^k)^T \Delta \mathbf{x}^k} - \frac{\widehat{\mathbf{H}}^k \Delta \mathbf{x}^k (\Delta \mathbf{x}^k)^T \widehat{\mathbf{H}}^k}{(\Delta \mathbf{x}^k)^T \widehat{\mathbf{H}}^k \Delta \mathbf{x}^k}$$

$$\Delta(\widehat{\mathbf{H}}^k)^{-1} = \frac{[\Delta \mathbf{x}^k - (\widehat{\mathbf{H}}^k)^{-1} \Delta \mathbf{g}^k] (\Delta \mathbf{x}^k)^T + \Delta \mathbf{x}^k [\Delta \mathbf{x}^k - (\widehat{\mathbf{H}}^k)^{-1} \Delta \mathbf{g}^k]^T}{(\Delta \mathbf{g}^k)^T \Delta \mathbf{x}^k} - \frac{[\Delta \mathbf{x}^k - (\widehat{\mathbf{H}}^k)^{-1} \Delta \mathbf{g}^k]^T \Delta \mathbf{g}^k \Delta \mathbf{x}^k (\Delta \mathbf{x}^k)^T}{[(\Delta \mathbf{g}^k)^T \Delta \mathbf{x}^k] [(\Delta \mathbf{g}^k)^T \Delta \mathbf{x}^k]}$$



## 8. RESUMEN

En este capítulo se ha pretendido dar una visión muy general de los métodos de optimización más utilizados en cálculo numérico. Cualquiera de los métodos basados en programación matemática: programación lineal no lineal o mixta tanto lineal como no lineal, requerirían casi un nuevo libro para cada uno de ellos.

A pesar de todo no hemos querido dejar de presentar al menos los métodos más utilizados de optimización para problemas sin restricciones, tanto directos –libres de derivadas- entre los que el método simplex flexible de Nedler y Mead ha sido el que más se ha popularizado y ha sido y continúa siendo muy utilizado en muy diversos ámbitos, como indirectos – que usan información de derivadas- entre los que el método de Newton o los métodos quasi-Newton son sin duda los más utilizados.

## 9. Programacion en Matlab®

Se añade a continuación el código MATLAB de los siguientes métodos numéricos antes descritos.

1. Método de búsqueda unidireccional utilizando sección áurea.
2. Método de búsqueda unidireccional utilizando interpolación polinómica (cuadrática)
3. Implementación del método simplex flexible.

### 9.1. Sección Áurea

```
function [x, obj] = golden(fun,x1,x4)

% Búsqueda unidireccional (minimizando) por el método de la Sección
% Áurea
%
% fun = archivo donde se calcula la función objetivo (debe ser una
% función escalar)
% x1 = punto de un extremo del intervalo
% x4 = punto del otro extremo del intervalo
%
% NOTA: se asume que existe un único mínimo en el intervalo x1, x4.

tol = 1e-5; % Tolerancia para la terminación del problema, calculada
% como la norma de la longitud de intervalo.
```

```
F1 = (3-sqrt(5))/2;
F2 = 1-F1;
L = x4 - x1;
error = norm(L);

x2 = x1 + F1.*L;
x3 = x1 + F2.*L;

f1 = feval(fun,x1);
f2 = feval(fun,x2);
f3 = feval(fun,x3);
f4 = feval(fun,x4);

while error > tol

    if f3 <= f2

        L = x4 - x2;
        x1 = x2;
        f1 = f2;
        x2 = x3;
        f2 = f3;

        x3 = x4 - F1.*L;
        f3 = feval(fun, x3);

    elseif f2 <= f3

        L = x3 - x1;
        x4 = x3;
        f4 = f3;
        x3 = x2;
        f3 = f2;

        x2 = x1 + F1.*L;
        f2 = feval(fun, x2);
    end

    error = norm(L);
end
x = x1;
obj= f1;
```

### 9.2. Interpolación cuadrática

```
function [x,obj] = optim_poly(objfun,x1,x3)

% METODO DE INTERPOLACION POLINOMICA para encontrar el mínimo de la
% función objfun en el intervalo [x1,x3]. Se admite que el mínimo es
% unico

tol = 1e-5;    % error maximo en el resultado

x2 = (x1+x3)/2; % el primer punto intermedio se toma como la media
                % de los otros

f1 = feval(objfun,x1);
```

```

f2 = feval(objfun,x2);
f3 = feval(objfun,x3);

error =1;
cont = 0;
while error > tol
    cont = cont +1;
    A= [x1^2, x1, 1; x2^2, x2, 1; x3^2, x3 1];
    d= [f1;f2;f3];
    coef = inv(A)*d;
    a= coef(1);
    b= coef(2);
    c= coef(3);

    xm = -b/(2*a);
    fm = feval(objfun, xm);

    % los dos primeros if son para el caso de funciones crecientes o
    % decrecientes al menos en lo que respecta a los tres puntos
    % considerados.

    if f1 >= f2 & f2 >= f3
        x1 = x2;
        f1 = f2;
        x2 = (x1 + x3)/2;
        f2 = feval(objfun,x2);
        y = f3;
        x = x3;
    elseif f1 <= f2 & f2 <= f3
        x3 = x2;
        f3 = f2;
        x2 = (x1 + x3)/2;
        f2 = feval(objfun,x2);
        y = f1;
        x = x1;
    elseif xm >= x1 & xm <= x2
        if fm >= f2
            x1 = xm;
            f1 = fm;
        else
            x3 = x2;
            f3 = f2;
            x2 = xm;
            f2 = fm;
        end
        y = fm;
        x = xm;
    elseif xm > x2 & xm <= x3
        if fm >= f2;
            x3 = xm;
            f3 = fm;
        else
            x1 = x2;
            f1 = f2;
            x2 = xm;
            f2 = fm;
        end
        y = fm;
        x = xm;
    end

    error = norm(x1-x3,1);

```

```
end
obj = y;
```

### 9.3. Ejemplo

Como ejemplo de aplicación veremos como es posible encontrar el óptimo de la función

$$\sin(x) \tan\left(\frac{x}{\pi}\right) \quad \text{En el intervalo } [0,3].$$

Podemos ejecutar las siguientes líneas:

```
f = inline('-sin(x).*tan(x/pi)');
[x, obj] = golden(f,0,3)
```

O bien:

```
[x, obj] = optim_poly(f,0,3)
```

En ambos casos obtenemos

```
x = -0.6828
obj = 2.1472
```

### 94. Simplex Flexible

```
% Método simplex flexible de Nedler y Mead.
% simplex(obj,xo)
%
% obj = nombre del fichero donde se encuentra la función objetivo
% xo = vector inicial para comenzar la optimización

function [y,x]=simplex(obj,xo)

% PARAMETROS DEL SIMPLEX

N=length(xo)+1; % numero de puntos igual a variables independientes mas
                % una

a= 1;           % longitud del lado de la figura simplex inicial;
ITER_MAX=200;  % numero máximo de iteraciones permitidas
tol=1e-8;      % separación entre los vértice mínima

x=zeros(N,N-1); % Matriz de vértices del simplex. cada fila un
                % punto

MEJOR=zeros(1,2); % índice y valor de función objetivo del mejor
                 % vértice
```

## Tema 8: Introducción a la optimización numérica

---

```
PEOR=zeros(1,2);    % índice y valor de función objetivo del peor
                    % vértice

% generación de la figura del simplex.

p = a/(N*2^0.5)*(N-1+(N+1)^0.5);
q = a/(N*2^0.5)*(-1+(N+1)^0.5);

for i=1:N
    for j=1:N-1
        if i==1
            x(i,j)=x0(j);
        elseif i==j+1
            x(i,j)=x0(j)+p;
        else
            x(i,j)=x0(j)+q;
        end
    end
end

% Evaluación de la función objetivo en los vértices iniciales
% y clasificación

for i=1:N
    fobj(i)=feval(obj,x(i,:));
end
MEJOR(2) = min(fobj);
PEOR(2) = max(fobj);

MEJOR(1) = min(find(fobj==MEJOR(2)));
PEOR(1) = min(find(fobj==PEOR(2)));

% comienzan las iteraciones
error=1;
cont=0;    %contador de iteraciones
cont2=0;    % contador de iteraciones sin mejora
mejora=MEJOR(2);

while error >= tol
    cont=cont + 1;

    MEJOR(2) = min(fobj);
    PEOR(2) = max(fobj);

    MEJOR(1) = min(find(fobj==MEJOR(2)));
    PEOR(1) = min(find(fobj==PEOR(2)));

    if abs(mejora- MEJOR(2)) < tol
        cont2=cont2+1;
    else
        mejora = MEJOR(2);
        cont2=0;
    end
end

% Localización del centroide
suma=zeros(1,N-1);
for j=1:N-1
    for i=1:N
        if i~=PEOR(1)
```

## Cálculo numérico en Ingeniería

---

```
        suma(j)= suma(j) + x(i,j);
    end
end
end
xc = suma/(N-1);    % Este es el centroide.

% REFLEXION
%disp('reflexion');

xpeor=x(PEOR(1,:);
xr= xc + (xc-xpeor);
fobjr = feval(obj, xr);

if fobjr < MEJOR(2)    % SI MEJORA INTENTAR EXPANSION

    xe= xc + (xc-xpeor)*2;
    fobje= feval(obj,xe);
    if fobje < fobjr    % si la expansión tiene éxito acepto
                        % movimiento

        fobj(PEOR(1)) = fobje;    % sustituyo este nuevo punto por el
            %peor
        x(PEOR(1),:)=xe;
    else

        fobj(PEOR(1))= fobjr;
        x(PEOR(1),:)=xr;
    end

elseif fobjr > PEOR(2)    % SI ES EL PEOR DE TODOS CONTRACCION

    xco = xc + (xc-xpeor)*0.5;
    fobjc = feval(obj,xco);

    if fobjc < MEJOR(2)    % si la contraccion tiene éxito
                        % acepto movimiento

        %disp('contracción aceptada')
        fobj(PEOR(1)) = fobjc;    % sustituyo este nuevo punto por
            % el peor
        x(PEOR(1),:)=xco;
    else    % CAMBIO DE VERTICES
        %disp('
            Cambio de vértices')
        for i=1:N
            if i~=MEJOR(1);
                x(i,:)=x(MEJOR(1),:) - 0.5*(x(MEJOR(1),:)-x(i,:));
                fobj(i)=feval(obj,x(i,:));
            end
        end
    end
else
    %disp('nuevo vértice en reflexion normal')
    fobj(PEOR(1))= fobjr;
    x(PEOR(1),:)=xr;
end

% Criterios para terminación
if cont >= ITER_MAX
    error=0;
end
if cont2 > 10
```

```
        error=0;  
    end  
end % el del while  
y=fobj;
```