

3. Optimización no lineal.

3.1. Introducción.

En este capítulo se presentan algunos métodos básicos para la optimización de problemas que son no lineales por la naturaleza de la función objetivo, de las restricciones o de las dos simultáneamente. En primer lugar se muestran algunos algoritmos básicos de optimización de problemas sin restricciones. La optimización de problemas que no están sujetos a restricciones se pueden realizar empleando derivadas primeras, segundas o sin emplear derivadas. En este capítulo se muestra un algoritmo de optimización de orden cero, es decir, que no requiere el empleo de derivadas. Este algoritmo es el método de la Sección Áurea. También se desarrolla el algoritmo de primer orden más conocido que es el método del gradiente. Por último para obtener mejores resultados en términos de convergencia se muestra un algoritmo de segundo orden, que es el método de Davidon-Fletcher-Powell.

La optimización de problemas no lineales sujetos a restricciones se realiza mediante la penalización de la función objetivo. Este proceso permite aplicar las técnicas de optimización de problemas sin restricciones a problemas con restricciones mediante el uso de una función objetivo nueva que contiene información de las restricciones. El proceso de penalización consiste en una nueva función objetivo que contiene a la original y a un término que cuando las restricciones estén próximas a no cumplirse incrementa el valor de la función objetivo, en un problema de minimización. Es decir se penaliza en la función objetivo el hecho de no cumplirse las restricciones.

En este capítulo se muestran algunos tipos de penalización de las restricciones haciendo aplicables los algoritmos de optimización a problemas con restricciones. En todos los procedimientos que se presentan se hace necesaria una búsqueda monodimensional para obtener el óptimo en un determinado intervalo de las variables de diseño y que se aplica en el proceso una vez que se conoce la dirección en la que la función objetivo mejora. Para ello se ha empleado el método de búsqueda por la Sección Áurea.

Una vez que se muestran las técnicas de optimización de funciones sin restricciones y las técnicas de penalización se explica la implementación realizada de estos métodos. La implementación se ha realizado en un programa de Elementos Finitos y en el lenguaje de programación *FORTRAN*.

Se ha implementado los algoritmos de optimización pese a que el programa de elementos finitos que se va a emplear cuenta con un módulo de optimización. Esto se debe a que el manual del módulo de optimización presenta una serie de incongruencias que desaconsejan el empleo del mismo. En primer lugar se presenta un método de orden 0 que realiza una aproximación de la función objetivo penalizada, esta superficie se define por una expresión cuadrática y no requiere la evaluación de derivadas. Las funciones de penalización que se muestran en dicho caso se dice que son todas del tipo interior extendida mientras que la expresión que se muestra no se corresponde a este tipo, sino a las de penalización interior, que presentan problemas de convergencia, además esta definición se realiza en función de unos parámetros sin especificar, por lo que no se tiene control de la ejecución del problema. El parámetro que se emplea en la penalización de la función objetivo no está definido el valor que toma en el inicio de la ejecución, además se indica que este parámetro se incrementa entre iteraciones, mientras que la penalización interior extendida requiere la reducción del parámetro para aproximar los límites de la región factible. La definición del parámetro de transición que se realiza no es acorde con la función de penalización empleada. Por último la actualización de las variables de diseño entre iteraciones queda en función de un parámetro cuyo origen no se especifica. El método de primer orden que se muestra en el manual del módulo de optimización indica que se emplea penalización exterior a las variables de diseño, pero no muestra las funciones que se emplean, el parámetro de penalización que se emplea en la función objetivo no está definido y además no se define el parámetro de transición de la penalización interior extendida, la forma de las funciones de penalización interior extendida que se emplean quedan en función de una serie de parámetros sin especificar, lo cual no permite controlar la ejecución del algoritmo de optimización. Por último al implementarse un método de segundo orden se consigue una mejora sustancial en la convergencia de problemas con una región factible muy amplia o con restricciones y función objetivo muy no lineal, aunque la aplicación realizada no cumple estos requisitos su implementación permite la optimización de un gran número de problemas en los que no se conozca la forma de las restricciones o de la función objetivo.

En todo el capítulo se ha empleado la nomenclatura que se muestra a continuación.

Variables de diseño \underline{x} , Función Objetivo $f(\underline{x})$, valor de la variable de diseño en la iteración k -ésima \underline{x}^k , dirección de búsqueda en la iteración k -ésima \underline{d}_k ,

3.2. Búsqueda monodimensional.

En los métodos de optimización monodimensional se debe tener definido un intervalo de la variable independiente en el que la función presente un mínimo, si el problema es de minimización, además la función debe tener carácter unimodal en el intervalo de la variable independiente en el que se quiere obtener el mínimo.

El algoritmo óptimo respecto al número de evaluaciones de la función, conocido el intervalo de incertidumbre y el de definición de la función es el algoritmo de Fibonacci. Sin embargo este algoritmo tiene la desventaja de que el número de evaluaciones debe ser conocido antes de comenzar el proceso iterativo. Este problema se solventa empleando el método de la Sección Áurea que se describe a continuación.

Dado un intervalo de incertidumbre $[a^n, b^n]$ en el que se quiere minimizar una función, el método de la sección áurea establece una relación constante entre dos intervalos consecutivos en los que se quiere minimizar la función. La función objetivo se evalúa entonces en dos puntos x_1^n y x_2^n que cumplen la relación:

$$\frac{x_2^n - a^n}{b^n - a^n} = \frac{b^n - x_1^n}{b^n - a^n} \quad (3.2.1)$$

Al evaluarse la función en x_1^n y x_2^n se pueden dar dos casos:

- i. $f(x_1^n) < f(x_2^n)$ y el nuevo intervalo de incertidumbre será $[a^n, x_2^n]$ y los nuevos puntos en los que evaluar la función serían $x_2^{n+1} = x_1^n$ y x_1^{n+1} se obtiene de la expresión (3.2.1), quedando:

$$\frac{x_2^{n+1} - a^{n+1}}{b^{n+1} - a^{n+1}} = \frac{x_1^n - a^n}{x_2^n - a^n} \quad (3.2.2)$$

- ii. $f(x_1^n) > f(x_2^n)$ y el nuevo intervalo de incertidumbre será $[x_1^n, b^n]$, luego la función se evaluará en los puntos $x_1^{n+1} = x_2^n$ y el nuevo x_2^{n+1} vendrá dado por la expresión:

$$\frac{b^{n+1} - x_1^{n+1}}{b^{n+1} - a^{n+1}} = \frac{b^n - x_2^n}{b^n - x_1^n} \quad (3.2.3)$$

De las expresiones anteriores se obtiene $\frac{1 + \sqrt{5}}{2} \approx 0.618$.

Este número se denomina número Áureo, de ahí el nombre de este algoritmo.

El proceso anterior se sigue hasta que el intervalo es menor que una cierta tolerancia o bien la función objetivo varía lo suficientemente poco dentro de una tolerancia fijada previamente. Al llegar a este punto se obtiene la posición del mínimo asumiendo que la función

objetivo en el último intervalo de incertidumbre se puede aproximar por una función cuadrática cuyo mínimo se puede obtener analíticamente y su posición también.

La función cuadrática que aproxima a la función objetivo en el intervalo final vendrá dada por la expresión.

$$f(x) = A \cdot x^2 + B \cdot x + C \quad (3.2.4)$$

Realizando la derivada respecto a la variable independiente x se obtiene la posición del mínimo en el punto en que esta derivada se anula, cuya expresión general es la que sigue.

$$x = -\frac{B}{2 \cdot A} \quad (3.2.5)$$

En el intervalo final se tendrá evaluada la función objetivo en tres puntos, por lo que la obtención de los coeficientes del ajuste cuadrático es inmediato con la expresión siguiente en función de a^{n+1} , b^{n+1} , x_2^{n+1} .

$$\begin{cases} A = \frac{f(a^{n+1}) - f(b^{n+1})}{a^{n+1} - b^{n+1}} \\ B = \frac{f(b^{n+1}) - f(x_2^{n+1})}{b^{n+1} - x_2^{n+1}} \\ C = \frac{f(x_2^{n+1}) - f(a^{n+1})}{x_2^{n+1} - a^{n+1}} \end{cases} \quad (3.2.6)$$

Introduciendo estos coeficientes en (3.2.5) se obtiene la posición del mínimo.

3.3. Optimización sin restricciones.

En muchos problemas ingenieriles se debe encontrar el óptimo de una función multivariable a la que no se le han impuesto restricciones. Los métodos que se emplean para resolver estos problemas se basan en dos formas básicas de abordarlos. Los métodos indirectos se basan en buscar el óptimo a través de unas ecuaciones derivadas de la condición de gradiente nulo. En la práctica estos métodos dan lugar a ecuaciones implícitas en el vector de variables de diseño \underline{x} . Estas ecuaciones son fuertemente no lineales por lo que resultan muy difíciles de resolver. Los métodos directos actúan directamente sobre la función a optimizar.

Los métodos directos aplicados a la optimización de funciones de varias variables independientes son iterativos. En el inicio de la iteración k -ésima se denotará por \underline{x}^k al valor estimado del óptimo. La iteración k -ésima consiste en el cálculo de un vector de búsqueda \underline{d}_k con el que se pueda obtener un nuevo estimado \underline{x}^{k+1} según una relación del tipo:

$$\underline{x}^{k+1} = \underline{x}^k + a_k \cdot \underline{d}_k \quad (3.3.1)$$

Una vez conocido \underline{d}_k , el valor a_k se obtiene mediante búsqueda lineal con el algoritmo de la sección áurea que se ha comentado previamente. Imponiendo

$$\text{MIN. } a_k f(\underline{x}^k + a_k \underline{d}_k) \quad \text{con } a_{k1} \leq a_k \leq a_{k2} \quad (3.3.2)$$

donde a_{k1} y a_{k2} son valores que hacen que \underline{x}^{k+1} cumpla las restricciones correspondientes.

Los distintos métodos que existen se basan en la elección de la dirección de búsqueda. Se supondrá, a partir de ahora, que la función a optimizar va a minimizarse, sin pérdida de generalidad ya que para maximizar sólo hay que minimizar la función objetivo cambiada de signo. Los distintos métodos que se emplean se clasifican en función de la información necesaria para la búsqueda del mínimo. Los métodos que emplean derivadas primeras de la función, tal como el método del gradiente, se denominan de primer orden. Los métodos que requieren las derivadas segundas de la función se conocen con el nombre de métodos de segundo orden.

3.3.1. Métodos de Primer Orden.

Estos métodos que emplean la información correspondiente al gradiente requieren que éste sea conocido, bien de forma analítica o mediante diferencias finitas. La eficiencia de estos métodos es pobre en el caso de funciones con derivadas primeras discontinuas.

El mejor conocido de los métodos de primer orden es el del gradiente o de Cauchy. Resulta un punto de partida para otros métodos más sofisticados. Es un método lento y su empleo está limitado a consideraciones académicas.

El método del gradiente procede realizando una búsqueda lineal a lo largo de la dirección dada por:

$$\underline{d} = - \frac{\nabla f(\underline{x})}{\|\nabla f(\underline{x})\|} \quad (3.3.3)$$

El algoritmo procede de la siguiente manera:

- i. Partir de un estimado inicial \underline{x}^k .
- ii. Si $\|\nabla f(\underline{x}^k)\| \leq \epsilon$, el proceso ha terminado siendo \underline{x}^k el óptimo buscado, ϵ es un valor lo suficientemente pequeño para poder suponer que el gradiente es nulo; En caso contrario ir a iii.
- iii. Definir la nueva dirección de búsqueda. $\underline{d}_k = - \frac{\nabla f(\underline{x}^k)}{\|\nabla f(\underline{x}^k)\|}$.
- iv. Realizar una minimización por búsqueda lineal en $f(\underline{x}^k + a_k \cdot \underline{d}_k)$ con $a_k = 0$.
- v. Definir un nuevo estimado $\underline{x}^{k+1} = \underline{x}^k + a_k \cdot \underline{d}_k$.
- vi. Repetir el proceso.

3.3.2. Métodos de Segundo Orden.

Estos métodos hacen uso de las derivadas primeras y segundas de la función a optimizar. Entre ellos el método de Newton es el prototipo de métodos de esta familia. El método de Newton se basa en aproximar la función a optimizar en un entorno del mínimo por una función cuadrática, que es la que se minimiza exactamente. En esta aproximación cuadrática se emplea la matriz hessiana que es una matriz simétrica formada con $\underline{H}(\underline{x}^k) = \nabla^2 f(\underline{x}^k)$. En un problema de minimización es necesario que la matriz $\underline{H}(\underline{x}^k)$ sea definida positiva, en el caso en que la matriz hessiana no sea definida positiva o sea singular no se puede obtener el óptimo con este método. Para solventar este problema se emplean los métodos de Newton modificados, que sustituyen la matriz hessiana no definida positiva por otra matriz próxima a la hessiana pero definida positiva.

En la práctica la matriz hessiana o bien no se puede calcular o es muy difícil y costosa de calcular. Por lo que un procedimiento práctico es partir de un estimado próximo a $\underline{H}^{-1}(\underline{x}^k)$ e irlo actualizando secuencialmente en cada iteración. Los algoritmos de métrica variable son aquellos en los que la dirección de búsqueda viene definida como:

$$\underline{d}_k = -\underline{S}^k \cdot \nabla f(\underline{x}^k) \quad (3.3.4)$$

donde \underline{S}^k es una matriz definida positiva que aproxima a $\underline{H}^{-1}(\underline{x}^k)$.

Con estos algoritmos se evita la evaluación de las segundas derivadas, exigidas en el método de Newton. De todas formas a la matriz \underline{S}^k se le exige que refleje, en cierta medida, propiedades del hessiano de la función para conseguir una convergencia rápida del problema. La matriz aproximada a la inversa del hessiano se emplea para definir la dirección de búsqueda de tal manera que cerca del óptimo se comporte como la inversa del hessiano. Dados dos estimados \underline{x}^k y \underline{x}^{k+1} el hessiano se puede aproximar por:

$$\underline{H}(\underline{x}^{k+1}) \approx \frac{\nabla f(\underline{x}^{k+1}) - \nabla f(\underline{x}^k)}{\underline{x}^{k+1} - \underline{x}^k} \quad (3.3.5)$$

haciendo $\underline{q}^k = \nabla f(\underline{x}^{k+1}) - \nabla f(\underline{x}^k)$ y $\underline{p}^k = \underline{x}^{k+1} - \underline{x}^k$ la expresión anterior resulta:

$$\underline{q}^k \approx \underline{H}(\underline{x}^{k+1}) \cdot \underline{p}^k \quad (3.3.6)$$

Si se supone que el hessiano puede considerarse constante en n sucesivos pasos del algoritmo se podrá hacer:

$$\underline{q}^k \approx \underline{H} \cdot \underline{p}^k \quad \forall k = 1, 2, \dots, n \quad (3.3.7)$$

y queda determinado \underline{H} para que se cumplan las ecuaciones anteriores.

Para un problema en vías de resolución, si el número de pasos resueltos es $k < n$, la matriz \underline{H}^1 no podrá ser construida totalmente y el sistema de ecuaciones

$$\underline{S}^{i?1} \cdot \underline{q}^i \text{ ? } \underline{p}^i \quad 1 = i < n \quad (3.3.8)$$

admite infinitas soluciones de la matriz \underline{S} . Dependiendo de cómo se defina \underline{S} se tendrá uno u otro método. A continuación se describe uno de ellos.

Método de Davidon-Fletcher-Powell en este método la matriz aproximada a la inversa del hessiano se actualiza sumándole dos matrices de rango unidad, por lo que este método es conocido como método de corrección de rango dos. Partiendo de la expresión (3.3.8) y suponiendo la matriz \underline{S}^{k+1} de la forma:

$$\underline{S}^{k?1} \text{ ? } \underline{S}^k \text{ ? } \underline{A}^k \quad (3.3.9)$$

Sustituyendo esta última expresión en la anterior se obtiene:

$$\underline{A}^k \cdot \underline{q}^k \text{ ? } \underline{p}^k \text{ ? } \underline{S}^k \cdot \underline{q}^k \quad (3.3.10)$$

cuya solución es de la forma:

$$\underline{A}^k \text{ ? } \frac{\underline{p}^k \cdot \underline{y}^{kT}}{\underline{y}^{kT} \cdot \underline{q}^k} \text{ ? } \frac{\underline{S}^k \cdot \underline{q}^k \cdot \underline{z}^{kT}}{\underline{z}^{kT} \cdot \underline{q}^k} \quad (3.3.11)$$

donde \underline{y}^{kT} , \underline{z}^{kT} son vectores arbitrarios no ortogonales a \underline{q}^k .

Como \underline{H}^1 es simétrica se eligen los vectores \underline{y}^{kT} y \underline{z}^{kT} de manera tal que \underline{A}^k sea también simétrica, siéndolo simultáneamente \underline{S}^{k+1} . La forma más simple de asegurarlo es haciendo $\underline{y}^k = \underline{p}^k$ y $\underline{z}^k = \underline{q}^k$. Finalmente la fórmula de Davidon-Fletcher-Powell será:

$$\underline{S}^{k?1} \text{ ? } \frac{\underline{p}^k \cdot \underline{p}^{kT}}{\underline{p}^{kT} \cdot \underline{q}^k} \text{ ? } \frac{\underline{S}^k \cdot \underline{q}^k \cdot \underline{q}^{kT} \cdot \underline{S}^k}{\underline{q}^{kT} \cdot \underline{S}^k \cdot \underline{q}^k} \text{ ? } \underline{S}^k \quad (3.3.12)$$

El algoritmo basado en este método consta de los siguientes pasos:

- i. Definición de una matriz inicial \underline{S}^0 definida positiva y de un estimado inicial \underline{x}^0 .
- ii. Obtención de la dirección de búsqueda $\underline{d}_k = \underline{S}^k \cdot \underline{g}^k$, donde $\underline{g}^k \text{ ? } f(\underline{x}^k)$.
- iii. Minimización de $f(\underline{x}^k + a \cdot \underline{d}_k)$ con respecto a $a = 0$ para obtener el nuevo estimado \underline{x}^{k+1} y así $\underline{g}^{k?1} \text{ ? } f(\underline{x}^{k?1})$.
- iv. Definición de $\underline{q}^k \text{ ? } f(\underline{x}^{k?1}) \text{ ? } f(\underline{x}^k) \text{ ? } \underline{g}^{k?1} \text{ ? } \underline{g}^k$, $\underline{p}^k \text{ ? } \underline{x}^{k?1} \text{ ? } \underline{x}^k$ y

$$\underline{S}^{k?1} \text{ ? } \frac{\underline{p}^k \cdot \underline{p}^{kT}}{\underline{p}^{kT} \cdot \underline{q}^k} \text{ ? } \frac{\underline{S}^k \cdot \underline{q}^k \cdot \underline{q}^{kT} \cdot \underline{S}^k}{\underline{q}^{kT} \cdot \underline{S}^k \cdot \underline{q}^k} \text{ ? } \underline{S}^k \quad (3.3.13)$$

3.4. Optimización con restricciones.

A continuación se van a presentar una serie de algoritmos que permiten resolver problemas de optimización no lineales sujetos a restricciones que tampoco tienen por qué ser lineales. En este caso el problema de optimización quedaría formulado de la siguiente manera:

$$\text{Minimizar: } f(\underline{x}) \quad (3.4.1)$$

$$\text{Sujeto a: } g_j(\underline{x}) = 0 \quad j = 1, m \quad (3.4.2)$$

$$h_k(\underline{x}) = 0 \quad k = 1, l \quad (3.4.3)$$

Junto a estas restricciones están también las de las variables de diseño \underline{x}_j que se encuentran restringidas a un intervalo de validez.

El procedimiento general es minimizar la función objetivo como una función sin restricciones pero penalizando las violaciones de las restricciones. Como estos métodos pueden dar lugar a problemas mal condicionados, la penalización se impone inicialmente de forma moderada y se va incrementando según se avanza en la optimización. Ello requiere la optimización de varios problemas penalizados para obtener el óptimo de la función objetivo que respeta las restricciones. Estos métodos se conocen por sus siglas en inglés *SUMT* (*sequential unconstrained minimization techniques*). La pseudo-función objetivo que se emplea en estos métodos son de la forma:

$$f(\underline{x}, ?_p) = f(\underline{x}) + ?_p \cdot P(\underline{x}) \quad (3.4.4)$$

donde $f(\underline{x})$ es la función objetivo original y $P(\underline{x})$ es la función de penalización y cuya forma depende del *SUMT* que se emplee. El escalar $?_p$ es un factor multiplicativo que determina el valor de la penalización y que permanece constante para cada optimización de la pseudo-función objetivo. El subíndice p denota el número de la minimización de la función penalizada. El factor multiplicativo $?$ se modifica de una minimización a otra, aumentando o disminuyendo según el tipo de penalización.

A continuación se muestran tres técnicas de penalización. La primera de ellas es la más sencilla de implementar y se conoce como penalización exterior puesto que sólo penaliza la función objetivo cuando no se cumplen las restricciones. El segundo método, llamado penalización interior, penaliza en mayor medida la función objetivo cuanto más se aproximan las variables de diseño a no cumplir las restricciones, de manera que si el estimado inicial se encuentra en la región factible, el proceso de minimización impide que el estimado salga del espacio disponible. Cada uno de los métodos anteriores tiene sus ventajas e inconvenientes, las ventajas de ambos métodos se incorporan en un tercer método llamado penalización interior extendida. Un conocimiento adecuado de estas técnicas y de las dificultades que presenta la

optimización de problemas no lineales sujetos a restricciones permite comprender la importancia de una formulación adecuada del problema.

En cada forma de penalización se mostrará un ejemplo de la función de penalización, este ejemplo se obtiene de la siguiente restricción:

$$3 \cdot 10^5 \leq N \leq 5 \cdot 10^5 \quad (3.4.5)$$

esta restricción se transforma para que quede como dos restricciones del tipo estándar, además se escalan para que no dependan de las dimensiones empleadas, la forma final de estas restricciones es la que sigue:

$$\frac{3 \cdot 10^5 - N}{3 \cdot 10^5} \leq 0 \quad (3.4.6)$$

$$\frac{N - 5 \cdot 10^5}{5 \cdot 10^5} \leq 0 \quad (3.4.7)$$

3.4.1. Penalización exterior.

En este caso la función de penalización se obtiene de:

$$P(\underline{x}) = \begin{cases} 0, & \text{si } g_j(\underline{x}) \leq 0 \text{ y } h_k(\underline{x}) \leq 0 \\ \rho \sum_{j=1}^m \max\{0, g_j(\underline{x})\}^2 + \sum_{k=1}^l \max\{0, h_k(\underline{x})\}^2 \end{cases} \quad (3.4.8)$$

En la expresión anterior se puede ver que no se impone penalización si todas las restricciones se cumplen ($g_j(\underline{x}) = 0$, $h_k(\underline{x}) = 0$), pero si alguna de ellas no se cumple se penaliza con el cuadrado de la restricción. Se hace así para conseguir una pendiente nula en la frontera de la región factible, asegurando una pendiente continua de la pseudo-función objetivo. A pesar de ello la segunda derivada no es continua en la frontera por lo que puede dar lugar a problemas numéricos si se emplean métodos de segundo orden.

Considerando ahora el factor multiplicativo ρ , se observa que un valor pequeño tiene como resultado una minimización sencilla de la función objetivo penalizada pero se permite una mayor violación de las restricciones. Por otro lado si se emplea un valor grande se asegurará una solución dentro de la región factible pero es más difícil de obtener debido a un peor condicionamiento del problema. El proceder habitual será entonces comenzar con un valor de ρ pequeño, optimizar la función objetivo penalizada, incrementar el valor de la penalización, actualizar la función objetivo penalizada y obtener un nuevo valor del óptimo partiendo del punto obtenido en la iteración anterior. Se continúa con el proceso anterior hasta que se alcanza un valor del óptimo satisfactorio.

Para el ejemplo comentado anteriormente la forma de la función objetivo penalizada sería como se muestra en la ilustración siguiente:

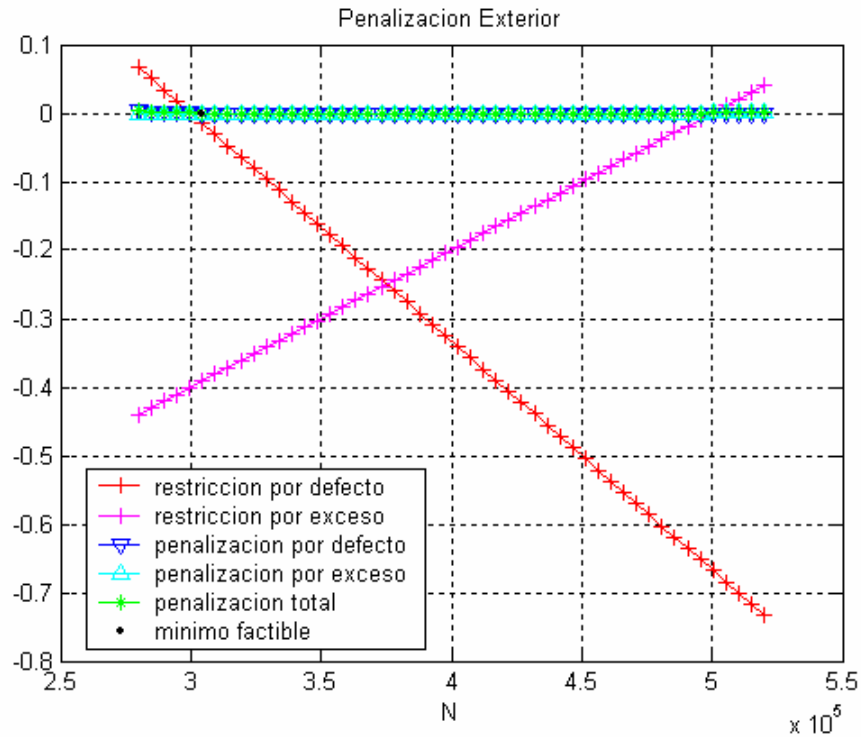


Ilustración 3-1. Penalización exterior, se muestra una vista general de las restricciones, de cada función de penalización y de la función de penalización total.

Se muestra de manera detallada a continuación la zona en la que se encuentran las funciones de penalización.

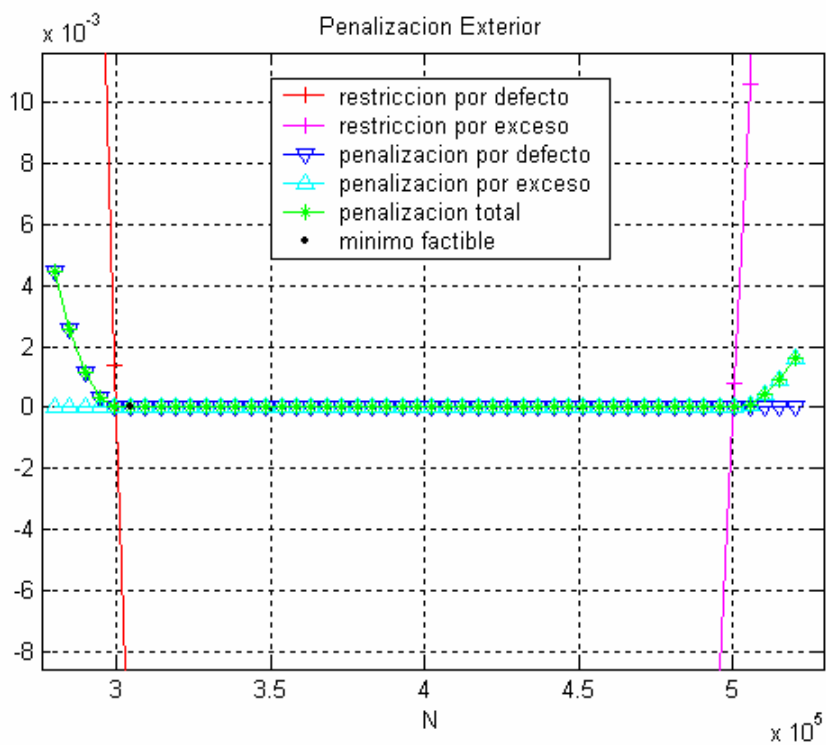


Ilustración 3-2. Funciones de penalización en la penalización exterior.

Se puede observar como sólo se penalizan las restricciones en la zona externa de la región factible.

3.4.2. Penalización interior.

Con este método se obtiene una serie de soluciones factibles pero a expensas de una función de penalización más compleja. La diferencia radica en la forma de penalizar las restricciones de desigualdad y en que el parámetro de penalización se disminuye en vez de aumentarlo durante el proceso de optimización.

La función de penalización más común en el caso de penalización interior es:

$$P(\underline{x}) = \sum_{j=1}^m \frac{1}{g_j(\underline{x})} \quad (3.4.9)$$

De esta forma la pseudo-función objetivo incluyendo las restricciones de igualdad quedaría:

$$\varphi(\underline{x}, \rho, \rho') = f(\underline{x}) + \rho \sum_{j=1}^m \frac{1}{g_j(\underline{x})} + \rho' \sum_{k=1}^l h_k(\underline{x})^2 \quad (3.4.10)$$

El último término de (3.4.10) es idéntico al de (3.4.8) y corresponde a la penalización exterior anterior ya que $h_k(\underline{x})$ debe aproximarse a cero todo lo posible. Por ello ρ tiene la misma evolución creciente de antes.

Se requiere que la función $g_j(\underline{x})$ sea negativa para que la solución sea factible, con lo que el segundo término de (3.4.10) será positivo según se aproxima $g_j(\underline{x})$ a cero la penalización tiende a infinito. El parámetro de penalización ρ' se emplea para distinguirlo del parámetro de penalización en la penalización exterior. Este parámetro ρ' empieza teniendo un valor alto y se disminuye cada vez que se obtiene un nuevo valor de las variables de diseño en el óptimo, comenzando la siguiente optimización con el valor anterior de las variables de diseño.

En el ejemplo comentado anteriormente la forma de las funciones de penalización y de las restricciones son las siguientes.

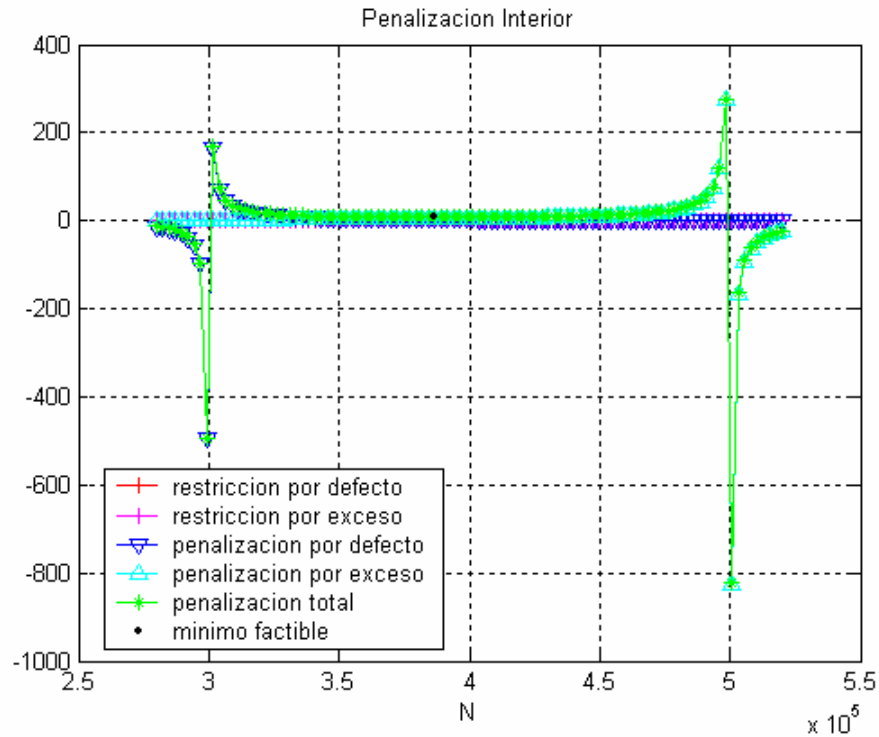


Ilustración 3-3. Penalización interior, se muestra una vista general de las restricciones, de cada función de penalización y de la función de penalización total.

Se mostrará a continuación una representación más detallada de las funciones de penalización.

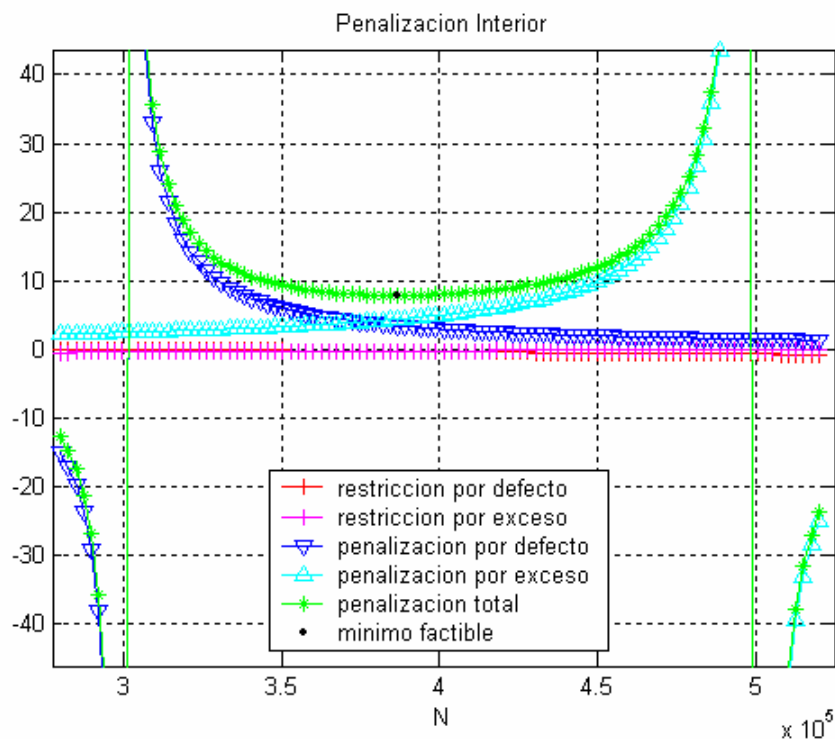


Ilustración 3-4. Detalle de la penalización interior, se pueden observar los problemas de continuidad que se presentan en el límite de la región factible.

3.4.3. Penalización interior extendida.

En este método se intenta aprovechar las ventajas de los dos métodos anteriores a la hora de penalizar las restricciones de desigualdad. Para las penalizaciones de igualdad se emplea la penalización exterior por lo que se omitirá en la formulación que sigue por brevedad.

En este método la función de penalización toma la forma:

$$P(\underline{x}) = \sum_{j=1}^m \tilde{g}_j(\underline{x}) \quad (3.4.11)$$

donde:

$$\tilde{g}_j(\underline{x}) = \frac{1}{g_j(\underline{x})} \quad \text{si} \quad g_j(\underline{x}) = e \quad (3.4.12)$$

$$\tilde{g}_j(\underline{x}) = \frac{2 \cdot g_j(\underline{x})}{e^2} \quad \text{si} \quad g_j(\underline{x}) > e \quad (3.4.13)$$

El parámetro e es un número negativo pequeño que marca la transición entre la penalización interior dada por (3.4.12) y la interior extendida dada por (3.4.13). Para obtener una secuencia de diseños factibles es necesario elegir e de manera que la pseudo-función objetivo tenga una pendiente positiva en la frontera de la región factible. Haftka y Starnes (referencia [14]) recomiendan un valor de e dado por:

$$e = -C \cdot \frac{1}{p} \quad \text{con} \quad 1/3 = a = 1/2 \quad (3.4.14)$$

donde C es una constante. Inicialmente se toma e dentro del rango $-0.3 = e = -0.1$. También se selecciona el valor de $1/p$ de manera que los términos de la penalización interior y exterior tengan inicialmente el mismo valor. Así se obtiene el valor de C que se debe emplear en la penalización. Con esta definición de la penalización se consigue que el óptimo permanezca en el interior de la región factible.

La forma de las funciones de penalización y de las restricciones en el ejemplo mostrado anteriormente se muestran a continuación para varios valores del parámetro e .

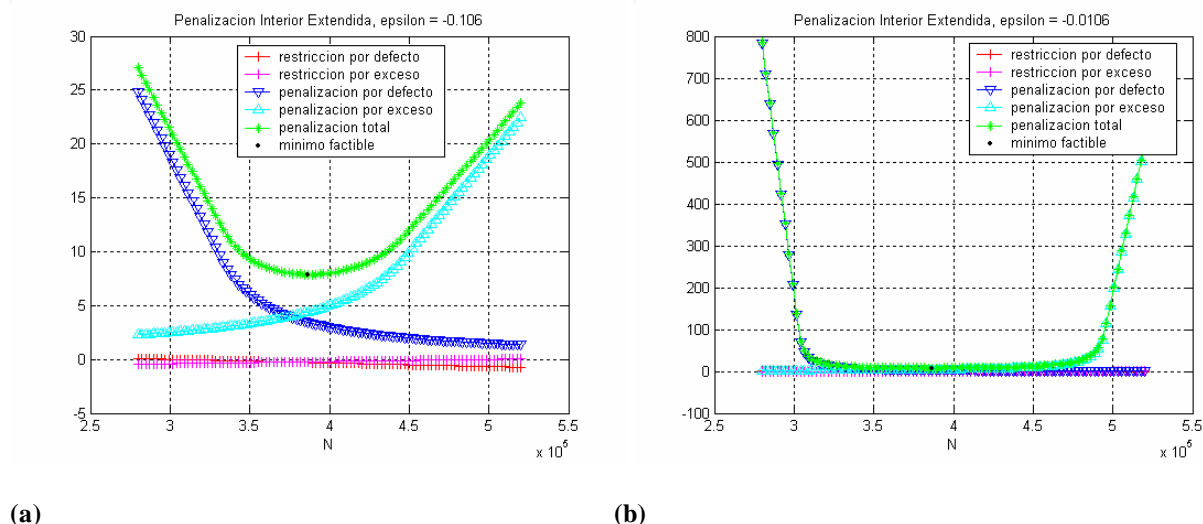


Ilustración 3-5. Penalización interior extendida, se muestran las restricciones, las funciones de penalización y de la función de penalización total. Se muestran dos valores distintos de ϵ , en (a) un valor inicial recomendado de -0.106 y en (b) un valor actualizado de -0.0106 . Se puede observar como se desplaza el mínimo de la función de penalización total.

3.5. Implementación de los algoritmos de optimización.

Se han implementado dos algoritmos de optimización para su aplicación en un programa de elementos finitos, *ANSYS*[®]. Este programa cuenta con un lenguaje de programación propio denominado *APDL*, siglas de *Ansys Parametric Design Language*. Los algoritmos de optimización implementados en *APDL* son paramétricos por lo que su validez es general dentro de la aplicabilidad de cada uno de los algoritmos en concreto. Además se han implementado los algoritmos de optimización en *FORTRAN* para su comprobación en una serie de problemas mecánicos con solución conocida, de manera que permite estudiar la validez de los resultados proporcionados por los algoritmos implementados en *APDL*.

3.5.1. Algoritmo de Primer Orden.

El algoritmo de primer orden implementado es el método del gradiente, el cual es el método básico de optimización no lineal. Este método es de aplicación general aunque cuenta con una convergencia lenta, por lo que su uso está restringido a problemas sencillos y aplicaciones académicas. La implementación en *APDL* de este algoritmo requiere la estructuración del programa en una serie de funciones que dirigen el flujo del programa y ayudan a la resolución del problema mecánico estudiado con el método de los elementos finitos.

En primer lugar es necesario definir el problema mecánico a resolver mediante el MEF y las variables de éste que se desean optimizar, llamadas variables de diseño. Dicho problema mecánico se deberá modelar de forma paramétrica, permitiendo la variación de las variables de

diseño. Se deben extraer las variables que conformarán la función objetivo a minimizar y las restricciones a que está sujeta la solución y que serán posteriormente penalizadas. Una vez obtenidas estas variables se forma la función objetivo penalizada con el parámetro de penalización considerado, se optimiza la función objetivo penalizada y se comprueba la convergencia variando el parámetro de penalización. A continuación se muestra el diagrama de flujo del algoritmo implementado.

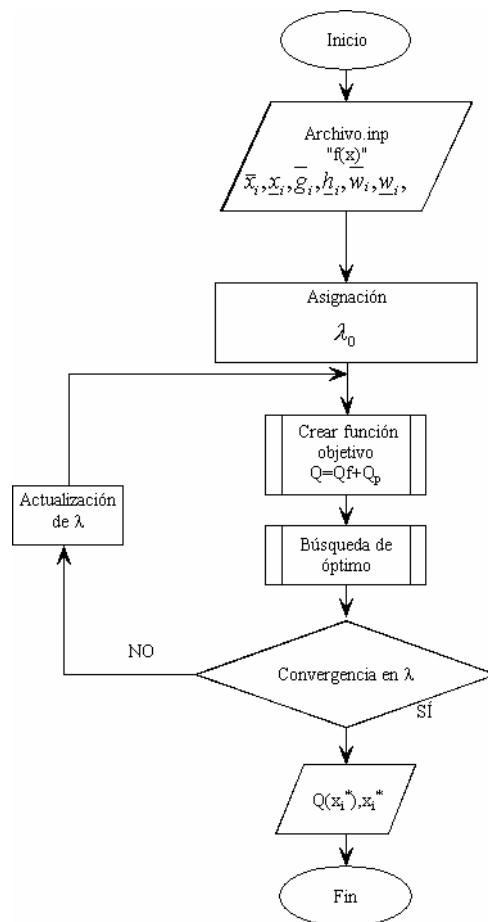


Ilustración 3-6. Diagrama de flujo del proceso general de optimización. En el archivo .inp se encuentra el problema mecánico a optimizar.

Una vez obtenidas las variables que conforman la función objetivo y las restricciones se forma la función objetivo penalizada mediante el método de penalización interior extendida que se comentó en el apartado anterior. El diagrama de flujo que representa este proceso es el siguiente:

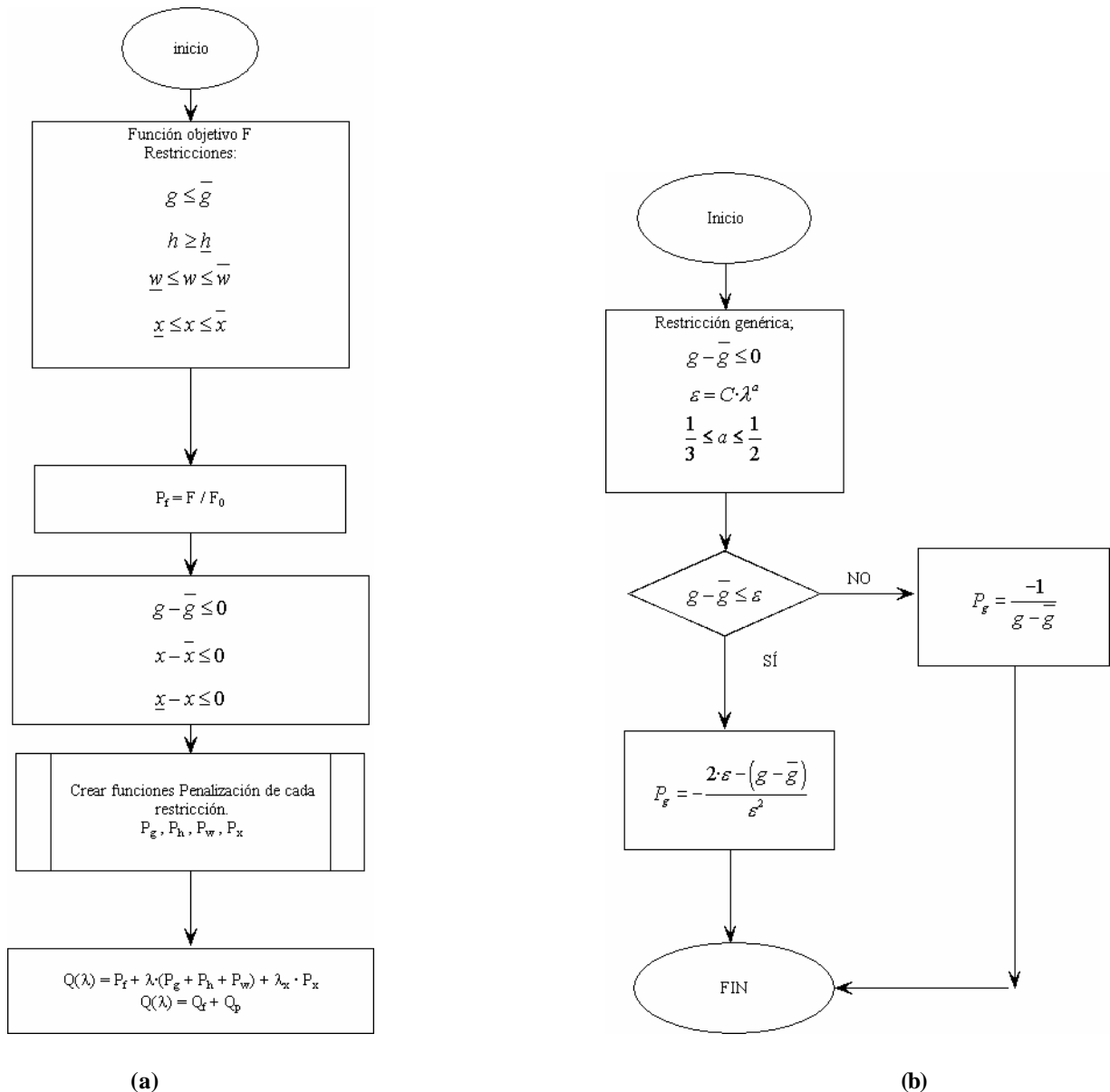


Ilustración 3-7. Proceso de creación de la función objetivo penalizada (a) y penalización de una restricción genérica (b).

Una vez obtenida la función objetivo penalizada se realiza la minimización propiamente dicha. Con el método del gradiente se obtiene la dirección de máxima disminución de la función objetivo. El cálculo del gradiente se realiza de forma aproximada mediante diferencias finitas. Para ello se parte de un valor inicial de las variables de diseño y con pequeños incrementos de cada una de las variables de diseño por separado se evalúa la función objetivo. Esto requiere la resolución de un problema mecánico distinto en cada caso. Una vez obtenido el gradiente es necesario calcular el valor máximo que pueden alcanzar las variables de diseño en esa dirección, puesto que debe estar definido un rango de validez de las variables de diseño. En la dirección dada por el gradiente y con el límite obtenido se realiza una minimización monodimensional mediante el método de la Sección Áurea. Este proceso general se realiza de la manera que sigue:

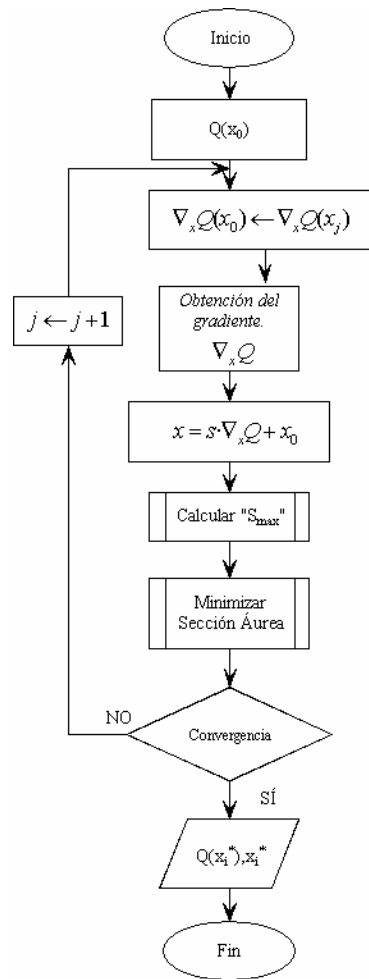


Ilustración 3-8. Diagrama de flujo del proceso de minimización de la función objetivo penalizada mediante el método del gradiente.

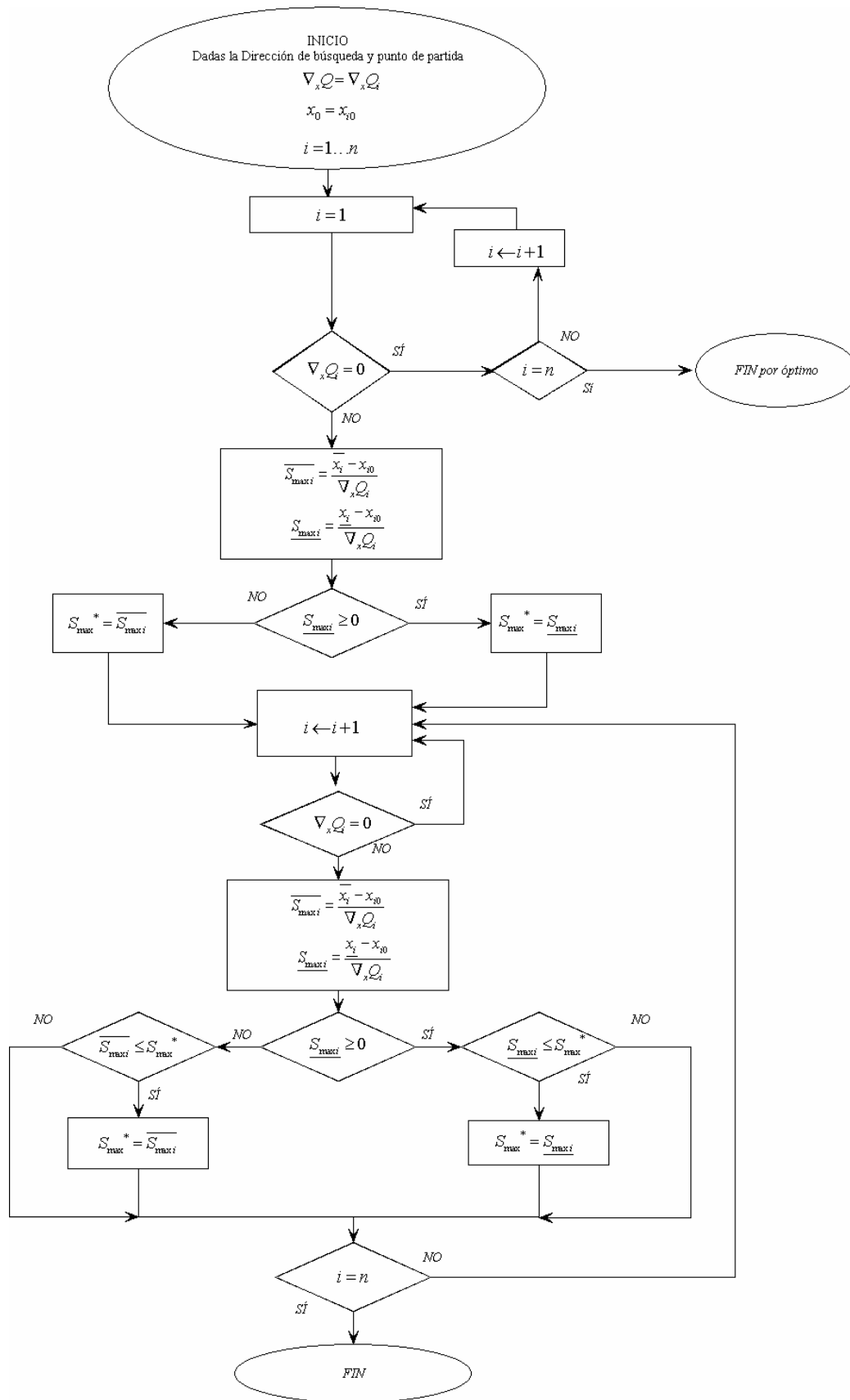


Ilustración 3-9. Obtención del valor límite de las variables de diseño en la dirección dada por el gradiente.

El proceso de minimización monodimensional mediante el método de la Sección Áurea se realiza con la evaluación de la función objetivo en una serie de puntos dados en la dirección del gradiente. Se compara el valor de la función objetivo penalizada en estos puntos y se reduce el intervalo en el que se encuentra el mínimo de la función objetivo penalizada. Una vez que se obtiene un intervalo lo suficientemente estrecho o una variación de la función objetivo lo suficientemente pequeña, entre distintas evaluaciones de la función, se finaliza el proceso de reducción del intervalo y se obtiene el mínimo mediante la aproximación con una función cuadrática con las expresiones (3.2.4) a (3.2.6). Este proceso se presenta de forma esquemática en la ilustración que sigue.

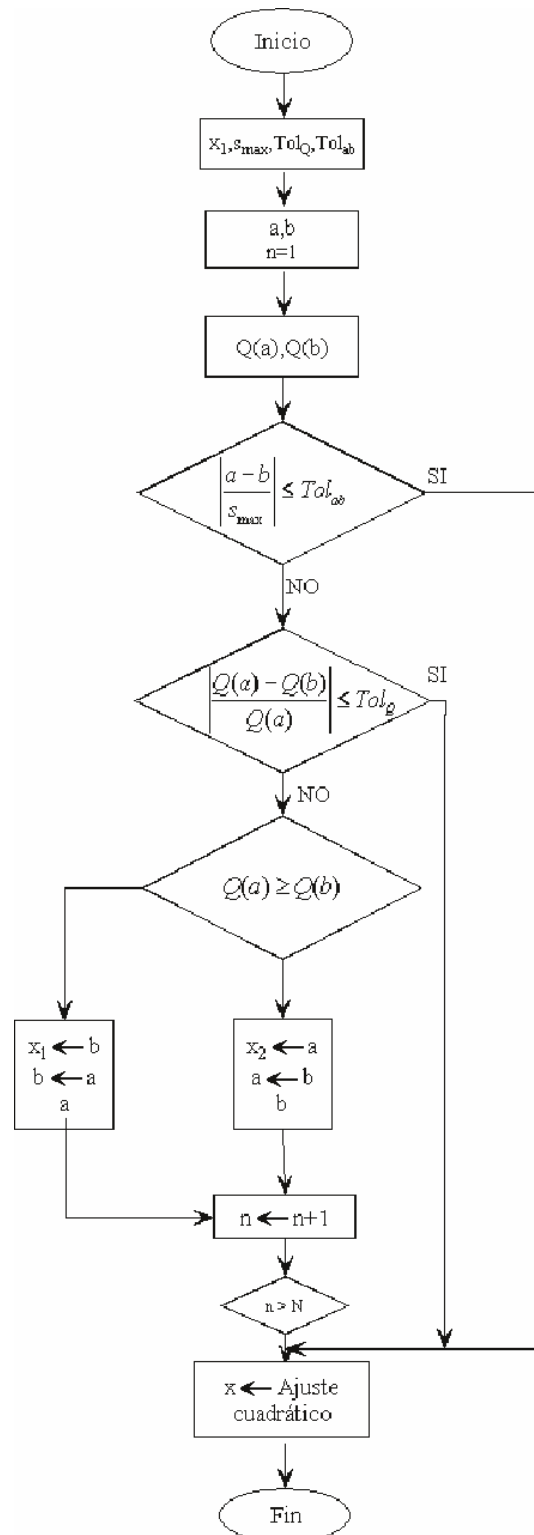


Ilustración 3-10. Proceso de búsqueda monodimensional mediante el algoritmo de la Sección Áurea

Una vez finalizada cada iteración de optimización se actualiza el punto inicial de las variables de diseño con el óptimo de la iteración anterior. El proceso global de optimización finaliza si se dan alguna de las dos situaciones siguientes, que el óptimo obtenido se encuentre próximo al punto de partida o que el gradiente sea nulo en el óptimo.

3.5.2. Algoritmo de Segundo Orden.

El método del gradiente presentado anteriormente requiere un gran número de iteraciones hasta llegar al óptimo. Como cada iteración requiere una serie de evaluaciones del problema mecánico a optimizar se hace necesaria la implementación de un algoritmo que proporcione una convergencia más rápida, reduciéndose así el tiempo de ejecución.

El algoritmo de segundo orden que se ha implementado es el algoritmo de Davidon-Fletcher-Powell. Este algoritmo proporciona una dirección de búsqueda más eficiente pero requiere un mayor uso de memoria puesto que es necesario almacenar el gradiente calculado en la iteración anterior para calcular la matriz que aproxima a la inversa de la hessiana.

El proceso general de optimización mediante este algoritmo es análogo al seguido en el método del gradiente, con la diferencia del cálculo de la dirección de búsqueda. La nueva dirección de búsqueda se obtiene con la actualización de la matriz que aproxima la inversa de la hessiana. Esta actualización se realiza mediante la definición de una serie de vectores en los que se almacena el gradiente de iteraciones anteriores.

En el proceso de optimización se parte de un punto inicial y de una estimación de la matriz inversa de la hessiana que debe ser definida positiva, por lo que una matriz con escalares positivos en la diagonal es un estimado aceptable. En el punto inicial se calcula el gradiente de la función objetivo penalizada mediante diferencias finitas y se obtiene la dirección de búsqueda con la expresión

$$\underline{d}_0 = -\underline{S}^0 \cdot \nabla f(\underline{x}^0) \quad (3.5.1)$$

donde \underline{d}_0 es la dirección de búsqueda inicial, \underline{S}^0 es la estimación inicial de la matriz que aproxima la inversa de la hessiana y \underline{x}^0 es el vector inicial de las variables de diseño.

Una vez obtenida la dirección de búsqueda se debe obtener el valor máximo que pueden alcanzar las variables de diseño en esa dirección y realizar la minimización monodimensional mediante el método de la Sección Áurea tal y como se realiza en el método del gradiente. Una vez obtenido el punto en el que se alcanza el mínimo se realiza una comprobación de convergencia. Si no se ha llegado al óptimo se calcula en el nuevo punto el gradiente mediante diferencias finitas y se actualiza la matriz que aproxima a la inversa de la hessiana. El diagrama de flujo del proceso descrito se muestra a continuación.

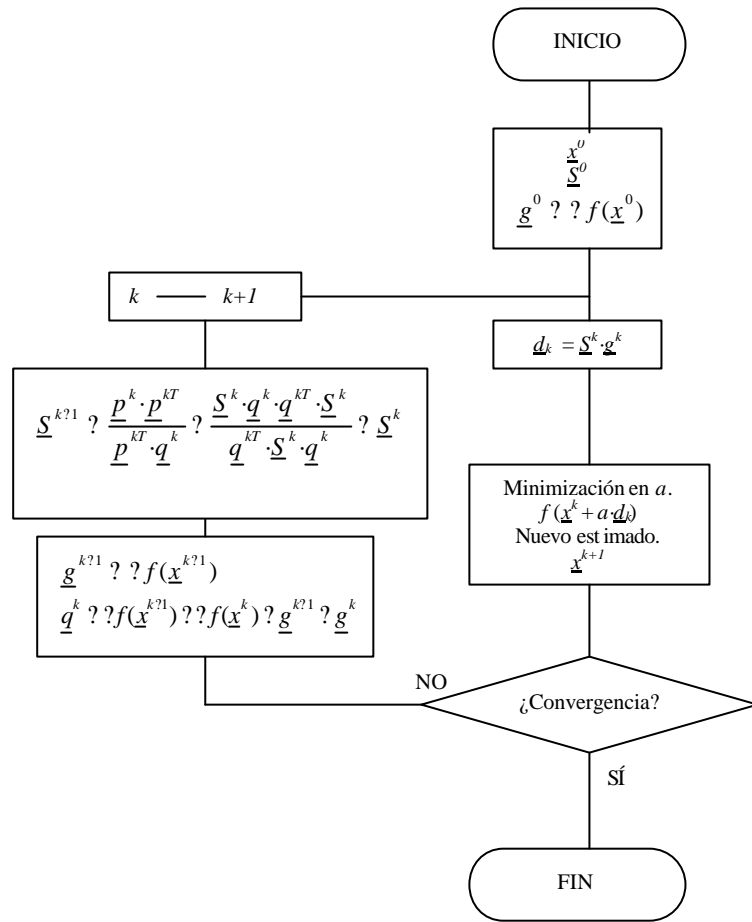


Ilustración 3-11. Diagrama de flujo del método de Davidon-Fletcher-Powell para la optimización multidimensional

Este método de segundo orden, al proporcionar una convergencia más rápida, permite un menor número de simulaciones del problema mecánico a optimizar, por lo que el tiempo de ejecución es menor que el necesario en el algoritmo de primer orden. Esto ocurre a costa de un mayor requerimiento de memoria, el cual no es significativo hasta valores muy grandes del número de variables de diseño.

El tiempo de ejecución está condicionado principalmente por la resolución mediante el MEF del problema mecánico, ya que la evaluación de las restricciones y de la función objetivo se realiza a través de estos resultados. Por ello es primordial obtener un algoritmo de ejecución de rápida convergencia, que permita un menor número de evaluaciones, es decir, de simulaciones del problema mecánico en Elementos Finitos.