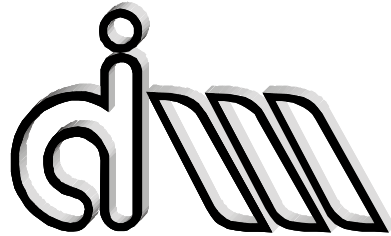


UNIVERSITAT POLITÈCNICA DE VALÈNCIA
Departamento de Ingeniería Mecánica y Materiales
Máster en Ingeniería Mecánica y Materiales



TRABAJO FIN DE MÁSTER

**DESARROLLO DE UNA METODOLOGÍA EFICIENTE
DE OPTIMIZACIÓN DE FORMA BASADO EN
ALGORITMOS DE TIPO GRADIENTE Y UN CÓDIGO
MEF DE MALLADOS CARTESIANOS
INDEPENDIENTES DE LA GEOMETRÍA**

Presentada por:
Dirigida por:

D. Luis Esparcia García
Dr. D. Juan José Ródenas García
Dr. D. José Albelda Vitoria

Valencia, Julio de 2014

Agradecimientos

*A mi familia, mis padres y mi hermano. Gracias a ellos he
podido llegar a donde estoy ahora.*

*Al DIMM-Departamento de Ingeniería Mecánica y de
Materiales por permitirme usar sus instalaciones en la realización del
proyecto.*

*A mis directores de proyecto Juanjo y Pepe por sus consejos y su
guía en la elaboración del proyecto. A Onofre por ayudarme a resolver
las dudas.*

*A mis amigos por su apoyo, a Santi por aguantarme todos estos
años de estudios, a Xavi por su apoyo moral.*

RESUMEN

La ingeniería requiere de la minimización de los costes de producción y logísticos. La reducción del tiempo de producción es un factor crítico para la reducción de los costes. La optimización de forma es un proceso necesario para tal fin, pero con un coste en tiempo a veces elevado.

En este trabajo se presentan métodos alternativos de optimización que reduzcan el tiempo de cálculo. Para ello se va a utilizar un algoritmo de optimización basado en gradiente, ya que se consideran que llegan a la solución más rápidamente que los algoritmos evolutivos.

Además, gran parte del tiempo de cálculo se utiliza en el cálculo de la malla de elementos finitos. Para acelerar este proceso, se va a utilizar el método de proyección del error, que considera los datos obtenidos en la iteración anterior del proceso de optimización.

Todo ello, sin perder la calidad del resultado que se obtendría con métodos convencionales. Estos métodos aseguran la calidad del resultado en detrimento del tiempo de cálculo. Sin embargo, el presente trabajo pretende optimizar el tiempo de cálculo.

El fin último del trabajo es buscar un método más eficaz de hacer procesos de optimización de forma.

RESÚM

La enginyeria requereix de la minimització dels costos de producció y logística. La reducció del temps de producció és un factor crític per a la reducció del costos. La optimització de forma és un procés necessari per eixa finalitat, però amb un cost en temps a vegades elevat.

En este treball es presenten mètodes alternatius de optimització que redueixen el temps de càlcul. Per això es va a utilitzar un algoritme d'optimització basat en gradients, ja que es consideren que arriben a la solució més ràpidament que els algoritmes evolutius.

A més, gran part del temps de càlcul s'utilitza en el càlcul de la malla d'elements finits. Per a accelerar este procés, es va a utilitzar el mètode de projecció de error, que considera les dades obtingudes en la iteració anterior del procés d'optimització.

Tot sense perdre la qualitat del resultat que s'obtindria amb mètodes convencionals. Aquests mètodes asseguren la qualitat del resultat en detriment del temps de càlcul. No obstant, el present treball pretén optimitzar el temps de càlcul.

La finalitat del treball es buscar un mètode més eficaç de fer processos d'optimització de forma.

ABSTRACT

Engineering requires the minimization of the production and logistics costs. The time reduction is a critical factor for the costs reduction. Shape optimization is a process needed in order to achieve that goal, but with a time cost elevated, sometimes.

In this MTh, some alternative optimization methods are presented which are intended to reduce calculation time. For this, a gradient based optimization algorithm will be used, since it is considered that they achieve the solution faster than the evolutionary algorithm.

Great part of the calculation time is used in the calculation of the finite elements mesh. In order to quicken this process, a projection error method will be used. This method takes data from the previous iteration of the optimization process.

All that may be achieved without losing result quality that would be achieved with conventional methods. Those methods assure the quality of the result whatever the calculation time. However, this MTh aims to optimize the calculation time.

The objective of the MTh is to search a more effective method of doing shape optimization processes.

Índice

1. INTRODUCCIÓN	1
1.1. IMPORTANCIA DE LA OPTIMIZACIÓN EN LA INDUSTRIA _____	1
1.2. TIPOS DE ALGORITMOS DE OPTIMIZACIÓN: MÉTODOS DETERMINISTA Y ESTOCÁSTICO _____	3
1.2.1. <i>Algoritmos deterministas en problemas sin restricciones</i> _____	3
1.2.2. <i>Algoritmos estocásticos</i> _____	4
1.3. USO DE EF EN OPTIMIZACIÓN DE FORMA ESTRUCTURAL _____	7
1.4. USO DE CGFEM PARA REDUCIR COSTE COMPUTACIONAL _____	9
1.5. DEFINICIÓN DE VARIABLES DE DISEÑO _____	13
1.6. SENSIBILIDADES EN CGFEM _____	15
1.6.1. <i>Análisis de sensibilidades en MEF</i> _____	16
2. SELECCIÓN DEL ALGORITMO DE OPTIMIZACIÓN	21
2.1. TIPOS DE ALGORITMOS BASADOS EN GRADIENTE EN PROBLEMAS DE OPTIMIZACIÓN SIN RESTRICCIONES _____	21
2.1.1. <i>Método del descenso adaptativo</i> _____	21
2.1.2. <i>Método de Newton</i> _____	23
2.1.3. <i>Programación secuencial cuadrática (SQP) con restricciones</i> _____	24
2.2. ELECCIÓN DE ALGORITMO _____	26
2.3. IMPLEMENTACIÓN DEL ALGORITMO EN CGFEM _____	27
3. OBTENCIÓN DE LAS DERIVADAS DE LAS MAGNITUDES LLEVADAS A ESTUDIO	31
3.1. MÓDULO DE DESPLAZAMIENTOS _____	31
3.2. TENSIÓN DE VON MISES _____	32
3.3. VOLUMEN _____	34

3.4.	ESTIMACIÓN DEL ERROR_____	34
4.	PROYECCIÓN DE MAGNITUDES ENTRE DISTINTAS GEOMETRÍAS	37
4.1.	INTRODUCCIÓN_____	37
4.2.	METODOLOGÍA_____	38
4.2.1.	<i>Parametrización del problema</i> _____	40
4.2.2.	<i>Proceso estándar de análisis</i> _____	40
4.2.3.	<i>Estimación del error</i> _____	40
4.3.	PROYECCIÓN DE CADA DISEÑO Y DEFINICIÓN DE LA MALLA ADAPTADA _____	41
4.4.	IMPLEMENTACIÓN _____	42
5.	EJEMPLOS NUMÉRICOS	53
5.1.	TUBERÍA CON SUPERFICIE INTERNA CIRCULAR_____	54
5.1.1.	<i>1 variable de diseño</i> _____	56
5.1.2.	<i>2 variables</i> _____	60
5.1.3.	<i>4 variables</i> _____	63
5.1.4.	<i>Comparativa de soluciones</i> _____	69
5.2.	TUBERÍA CON PAREDES RECTAS Y ARCO EN SUPERFICIE INTERNA	70
5.3.	TUBERÍA CON SUPERFICIE INTERNA ELÍPTICA _____	75
5.4.	PRESA_____	81
6.	CONCLUSIONES	91
7.	TRABAJOS FUTUROS	93
	BIBLIOGRAFÍA	95
	ANEXO	1
1.	VARIABLES	3

1.1.	REFERENCE	3
2.	FUNCIONES	5
2.1.	PROCESO DE OPTIMIZACIÓN	5
2.1.1.	<i>OptimizacionGlobal</i>	5
2.1.2.	<i>ObjectiveFunctionCylinder_4var</i>	6
2.2.	PROYECCIÓN DEL ERROR	10
2.2.1.	<i>ProyeccionError</i>	10
2.2.2.	<i>PlotDeformedShape</i>	14
2.3.	SA_SETDESVAR_NURBS_1VAR	17
2.4.	SA_STRESSVM	19

1. Introducción

1.1. Importancia de la optimización en la industria

La optimización de forma es un recurso de cálculo que permite mejorar las características mecánicas mediante cambios en la geometría de componentes mecánicos u objetos en general. En la industria se consideran principalmente las propiedades físicas y mecánicas de los componentes, pero también es importante el coste económico.

La geometría requerida se obtiene en base a los requisitos exigidos del componente. Sin embargo se puede obtener una forma válida pero no óptima. El caso más común en la industria es la minimización de la masa del componente, que en el caso de un único material es equivalente a la minimización del volumen. Puesto que los materiales se compran por kilogramos, si en la elaboración de una pieza se ahorra un kilo de ese material, al elaborar miles de piezas el ahorro puede llegar a ser considerable.

Además, hay otros factores como la manufacturación. La manufacturación de una pieza con una geometría compleja puede ser costosa. Pero con la optimización de forma, cuyo requisito sea una configuración determinada, puede simplificar la geometría y abaratar la manufacturación.

Puesto que el dinero es el motor de la industria, será necesario ofrecer precios más competitivos y obtener beneficio de ello. La optimización de forma es una herramienta importante en la consecución de ese objetivo.

Se pueden distinguir dos tipos de problemas de optimización, el problema de maximización y el problema de minimización. El problema de minimización se puede definir de la siguiente forma. Dado un espacio de decisión (espacio de búsqueda) X , un espacio objetivo Y (valores objetivo) y un conjunto de funciones $f: X \rightarrow Y$ (funciones objetivo) sujeto a unas restricciones g_j , hay que encontrar un vector paramétrico $\mathbf{x}^* \in X$ que minimice la función objetivo.



CAPÍTULO I INTRODUCCIÓN

La forma general del problema de optimización es:

$$\begin{aligned} \text{Minimiza: } & f(\mathbf{x}); & \mathbf{x} &= \{x_i\}; \quad i = 1, \dots, n \\ \text{Con:} & & \mathbf{g}(\mathbf{x}) &= \{g_j(\mathbf{x})\}; \quad j = 1, \dots, m \\ \text{Cumpliendo:} & & g_j(\mathbf{x}) &\leq 0; \quad j = 1, \dots, m \\ & & a_i &\leq x_i \leq b_i; \quad i = 1, \dots, n \end{aligned}$$

Donde x_i son las variables en base a las que se define el problema, a_i y b_i son los límites inferiores y superiores de dichas variables.

Este es el problema básico de optimización. Dependiendo del tipo de algoritmo de optimización se utilizarán estas funciones u otras con pequeñas variaciones. Hay algoritmos que utilizan información obtenida solamente de la función objetivo. Otros, sin embargo, utilizan información de las derivadas de la función objetivo e incluso de las funciones de restricción. Estas diferencias se irán comentando en los siguientes apartados.

TRABAJO FIN MÁSTER

1.2. Tipos de algoritmos de optimización: métodos determinista y estocástico

La diferencia principal entre los diferentes algoritmos de optimización es el procedimiento para calcular las direcciones necesarias para la minimización y la determinación del tamaño óptimo de cada paso a seguir.

1.2.1. Algoritmos deterministas en problemas sin restricciones

Los algoritmos de gradientes representan el grupo más importante de los algoritmos deterministas y se basan en las derivadas de la función objetivo. A partir de ellos se obtiene la dirección del incremento a imponer a las variables de optimización, en término de las variables de diseño, y una estimación del valor en módulo del incremento de dichas variables para alcanzar el mínimo o máximo de la función objetivo.

El gradiente de una función diferenciable (E) en el punto Θ es el vector de primeras derivadas parciales de E que apunta en dirección al máximo incremento de E y cuya magnitud es la máxima tasa de cambio.

$$g(\Theta) = \nabla E(\Theta) = \left[\frac{\partial E(\Theta)}{\partial E(\Theta_1)}, \dots, \frac{\partial E(\Theta)}{\partial E(\Theta_n)} \right]$$

Ec 1

Para encontrar el mínimo se quiere encontrar un Θ_{next} que cumpla con que el gradiente sea cero (p.ej. un punto estacionario como un mínimo o máximo):

$$g(\Theta_{next}) = \left(\frac{\partial E(\Theta)}{\partial \Theta} \right) \Big|_{\Theta=\Theta_{next}} = 0$$

Ec 2

El principal inconveniente de estos algoritmos es que proporcionan mínimos y máximos locales y no necesariamente el global.



1.2.2. Algoritmos estocásticos

Los algoritmos estocásticos se basan en distintos criterios para obtener las nuevas geometrías que no usan el gradiente de la función objetivo. Algunos de ellos se denominan evolutivos, ya que la solución que proporcionan va modificándose hasta que se obtiene la solución óptima siguiendo un proceso que simula la evolución de los seres vivos en la naturaleza. Entre los más utilizados son los genéticos, colonia de hormigas y recocido simulado (*simulated annealing* en inglés).

En el caso de que existan restricciones sólo se consideran en el proceso las soluciones que las satisfacen y se eliminan las que no lo hacen. En general, no se suele forzar el cumplimiento de las restricciones aunque en algunas ocasiones resulta muy complicado obtener soluciones factibles.

1.2.2.1. Algoritmo genético

Se inspiran en la evolución biológica y su base genético-molecular. Estos algoritmos hacen evolucionar una población de individuos someténdola a acciones aleatorias semejantes a las que actúan en la evolución biológica (mutaciones y recombinaciones genéticas), así como también a una selección de acuerdo con algún criterio, en función del cual se decide cuáles son los individuos más adaptados, que sobreviven, y cuáles los menos aptos, que son descartados.

Los algoritmos genéticos son de probada eficacia en caso de querer calcular funciones no derivables (o de derivación muy compleja) aunque su uso es posible con cualquier función. Deben tenerse en cuenta también las siguientes consideraciones:

- Si la función a optimizar tiene muchos máximos/mínimos locales se requerirán más iteraciones del algoritmo para "asegurar" el máximo/mínimo global.
- Requiere evaluar muchas soluciones intermedias

TRABAJO FIN MÁSTER

1.2.2.2. Colonia de hormigas

El algoritmo de la colonia de hormigas (*Ant Colony Optimization*, ACO) es una técnica probabilística para solucionar problemas computacionales que pueden reducirse a buscar los mejores caminos o rutas en grafos.

La idea original proviene de la observación de la explotación de los recursos alimentarios entre hormigas, en el que las habilidades cognitivas de las hormigas son individualmente limitadas y en conjunto son capaces de buscar el menor camino existente entre la fuente de comida y su nido o colonia.

- La primera hormiga encuentra la fuente de alimentos (F), a través de cualquier camino (a), entonces retorna a la colonia (N), dejando tras sí un rastro de feromonas;
- Las hormigas indiscriminadamente siguen cuatro caminos posibles, pero el fortalecimiento de la pista hace más atractiva la ruta más corta;
- Las hormigas toman la ruta más corta y largas porciones de otras rutas empiezan a perder su rastro de feromonas.

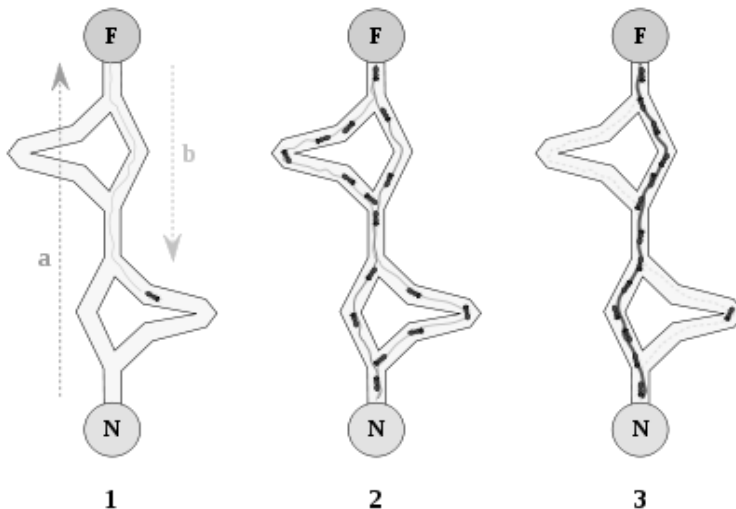


Fig. 1-1 Colonia de hormigas



CAPÍTULO I INTRODUCCIÓN

1.2.2.3. Algoritmo de recocido simulado

Simulated annealing (recocido simulado) es un algoritmo de búsqueda meta-heurística para problemas de optimización global; el objetivo general de este tipo de algoritmos es encontrar una buena aproximación al valor óptimo de una función en un espacio de búsqueda grande. A este valor óptimo se le denomina óptimo global.

El nombre e inspiración viene del proceso de recocido del acero y cerámicas, una técnica que consiste en calentar y luego enfriar lentamente el material para variar sus propiedades físicas. El calor causa que los átomos aumenten su energía y que puedan así desplazarse de sus posiciones iniciales (un mínimo local de energía); el enfriamiento lento les da mayores probabilidades de recrystalizar en configuraciones con menor energía que la inicial (mínimo global).

En cada iteración, el método de recocido simulado evalúa algunos vecinos del estado actual y probabilísticamente decide entre efectuar una transición a un nuevo estado o quedarse en el estado actual.

El método de evaluación de estados vecinos es fundamental para encontrar una solución óptima global al problema dado. Los algoritmos heurísticos, basados en buscar siempre un estado vecino mejor (con energía más baja) que el actual se detienen en el momento que encuentran un mínimo local de energía. El problema con este método es que no puede asegurar que la solución encontrada sea un óptimo global, pues el espacio de búsqueda explorado no abarca todas las posibles variaciones del sistema.

TRABAJO FIN MÁSTER

1.3. Uso de EF en optimización de forma estructural

Desde el punto de vista matemático, un problema de optimización se puede considerar como la minimización de una función $f(\mathbf{v})$ que depende de un conjunto de variables \mathbf{v} y que está sujeta a una serie de restricciones.

El problema de diseño estructural óptimo consiste en resolver el siguiente problema matemático:

$$\begin{aligned} \text{Minimiza: } f(\mathbf{v}); \quad & \mathbf{v} = \{v_i\}; \quad i = 1, \dots, n \\ & g_j(\mathbf{v}) \leq 0; \quad j = 1, \dots, m \\ & h_k(\mathbf{v}) = 0; \quad k = 1, \dots, m \\ & a_i \leq v_i \leq b_i \end{aligned}$$

Ec 3

Donde f es la función objetivo (FO), v_i son las variables de diseño y g_j son las restricciones en desigualdad, las cuales, en los problemas estructurales, normalmente se expresan en función de las tensiones y/o de los desplazamientos. Los valores a_i y b_i definen restricciones laterales. Cada individuo se caracteriza por un conjunto de valores de \mathbf{v} que se corresponden con un diseño estructural específico. La definición de cada diseño en función de los valores de \mathbf{v} se denomina la parametrización del problema de optimización. La resolución del problema de diseño óptimo consiste en hallar los valores de \mathbf{v} que definen el mejor diseño.

En la mayoría de casos de un problema estructural la función objetivo es la masa o el volumen total del componente.

Los algoritmos utilizados para la resolución de los problemas de optimización son, en general, iterativos. Con independencia del algoritmo utilizado, será necesario calcular los valores de f y g para cada uno de los distintos diseños propuestos durante el proceso iterativo de optimización. En este trabajo hemos considerado problemas de optimización de formas estructurales. Para este tipo de problemas los valores de f y g se obtienen normalmente mediante la utilización del método de los elementos finitos.

El Método de Elementos Finitos (FEM) se ha convertido en una de las técnicas más potentes en la resolución de problemas físicos. Es ampliamente



CAPÍTULO I INTRODUCCIÓN

usado para hallar soluciones aproximadas de ecuaciones diferenciales en relación a problemas de la ingeniería. Es necesario construir una malla específica para cada uno de los distintos diseños a analizar y, a continuación, utilizar el MEF para obtener la respuesta estructural de cada diseño.

Puesto que los procesos de optimización requieren de un gran número de iteraciones para su resolución, el MEF es un método adecuado para llevarlo a cabo. Sus soluciones son aproximadas pero válidas desde un punto de vista ingenieril, y tanto más aproximada cuanto más refinada es la malla.

Este trabajo fin de máster se centra en problemas de optimización 2D con representación exacta de la geometría, regidas por las ecuaciones de elasticidad lineal, usando el MEF también para determinar la sensibilidad de las propiedades de interés.



TRABAJO FIN MÁSTER

1.4. Uso de *cgFEM* para reducir coste computacional

Un programa de EF desarrollado en el Departamento de Ingeniería Mecánica y Materiales (DIMM) usa mallas independientes de la geometría. Para ello el programa implementa la teoría del Método de Elementos Finitos Generalizado (GFEM), que se explica en el siguiente apartado. Con el GFEM se tendrán dos mallas. Una malla de aproximación que cubre el dominio original y se utiliza para la construcción de las bases de aproximación y otra malla de integración dispuesta para la evaluación numérica de todas las integrales. Además los elementos de la malla de aproximación están situados según una malla cartesiana para reducir el coste computacional generado normalmente en los procesos de optimización.

Uno de los métodos de optimización más usados es el método del gradiente, basados en el cálculo de derivadas (o sensibilidades). Para conseguir esos gradientes es necesario un análisis de sensibilidad de las variables de diseño. En nuestro caso las variables de diseño vienen determinadas por el analista para describir la geometría del componente a optimizar. En los apartados siguientes se desarrollará la teoría del cálculo de la sensibilidad de forma, lo cual se utilizará como base en el módulo de cálculo.

En las últimas décadas el FEM se ha desarrollado mucho gracias a su versatilidad, robustez y flexibilidad en la rama de las matemáticas aplicadas para modelos numéricos de sistemas físicos de distintas disciplinas ingenieriles. Se ha realizado un gran esfuerzo para mejorar su fiabilidad y eficiencia.

El GFEM fue presentado por Stroboulis, Babuška y Copps [9] como combinación del FEM estándar y el Método de la Partición de la Unidad (PUM).

Las mallas del FEM estándar han de adaptarse al contorno del dominio a analizar, pero esta adaptación resulta compleja, de alto coste computacional y, en no pocas ocasiones, imprecisa. GFEM puede aumentar la precisión de la solución para dominios complejos. Se basa principalmente en una combinación de integración numérica robusta y un refinamiento especial de los elementos. Funciona bien en casi todo tipo de malla o dominio.



CAPÍTULO I INTRODUCCIÓN

Según los autores, GFEM se puede clasificar en tres categorías de acuerdo a la relación entre la malla de aproximación y el dominio de la geometría.

- GFEM I: la malla de aproximación es la clásica malla de FE para el dominio entero pero que proporciona funciones especiales en puntos donde el comportamiento es conocido. Fig. 1-2
- GFEM II: malla de aproximación es la clásica malla de FE para un dominio modificado que incluye los dominios internos originales. Se obtiene a partir del dominio original quitando varias partes del contorno y añadiendo nuevos contornos. Fig. 1-3
- GFEM III: La malla de aproximación es una malla que cubre el dominio original y no tiene contornos en común con la geometría. Es el tipo de mallas usadas en el programa desarrollado por el DIMM. Fig. 1-4

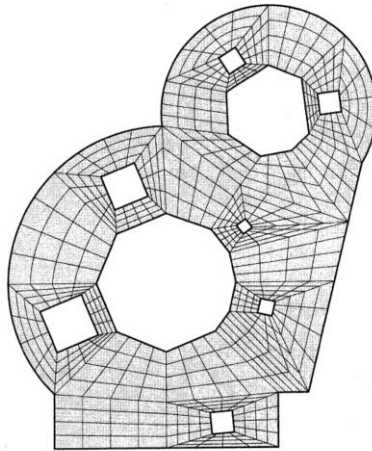


Fig. 1-2 Malla GFEM I

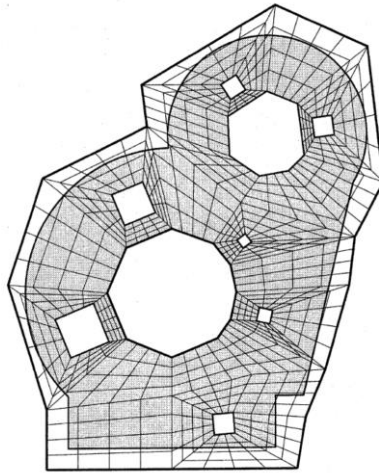


Fig. 1-3 Malla GFEM II

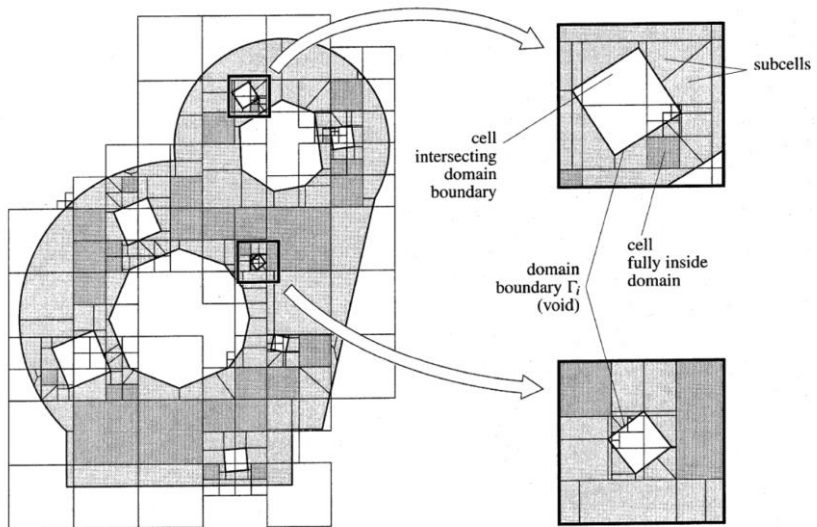


Fig. 1-4 Malla GFEM III



CAPÍTULO I INTRODUCCIÓN

FEM siempre utiliza las llamadas mallas de elementos finitos para la construcción de la aproximación y la integración numérica de la matriz de rigidez, vector de cargas, etc. La malla de elementos finitos se construye habitualmente subdividiendo el dominio en un conjunto de subdominios triangulares o cuadriláteros curvilíneos no superpuestos, los cuales se llaman elementos y tienen que satisfacer varias restricciones en cuanto a distorsión y restricciones de conexión con sus vecinos.

GFEM usa dos mallas. Una llamada malla de aproximación que se usa para la construcción de la base de aproximación. La otra concierne a la malla de integración construida usando cada elemento de la malla de aproximación por separado y se usa para la evaluación numérica de todas las integrales. En **Fig. 1-5** se muestran unos ejemplos de ambas mallas.

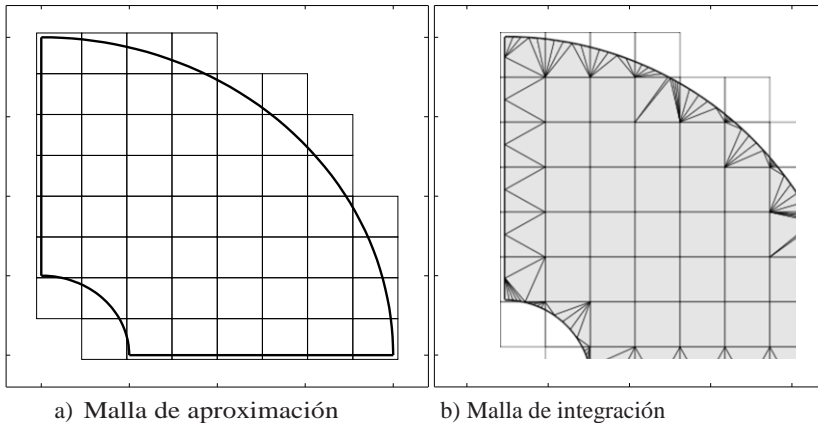


Fig. 1-5 Ejemplo de malla de cgFEM

La malla de aproximación necesita satisfacer únicamente el requisito de cubrir completamente la extensión del problema. La malla de integración se obtiene por el refinado especial de la malla de aproximación con cada elemento por separado, teniendo en cuenta la geometría local del dominio. Para el refinamiento se usa la triangulación de Delaunay, que crea subdominios e intersecciones según la curvatura del borde que cruza el elemento.

El programa desarrollado por el DIMM usa mallas independientes de la geometría pero además los elementos están dispuestos según un patrón cartesiano. Se ahorra en costes computacionales en análisis de optimización donde se requieren análisis iterativos con gran cantidad de flujo de datos. El programa se llama Cartesian Grid_FEM (cgFEM).

1.5. Definición de variables de diseño

El propósito de calcular sensibilidades de forma se basa en obtener derivadas respecto a variables escogidas por el analista que llamamos variables de diseño.

El objetivo en sensibilidades de forma es calcular variaciones en desplazamientos debidas a cambios en el contorno de la geometría. Se puede asumir que las variables de diseño serán variaciones en parámetros geométricos, coordenadas x o y , de puntos usados en la definición de curvas pertenecientes al contorno de la geometría llevada a estudio. En este trabajo se ha establecido que las variables de diseño se especifican en una función en la que el usuario puede indicar los puntos de la geometría a modificar. En este caso, los cambios vendrán realizados por el optimizador, indicando el incremento necesario de las variables de diseño.

Consideremos a modo de ejemplo una tubería sometida a presión interna en la que su superficie externa puede variar. Las coordenadas que definen el contorno deberán ser modificadas para variar la geometría. Para ello se aplica la variación como se muestra en la figura:

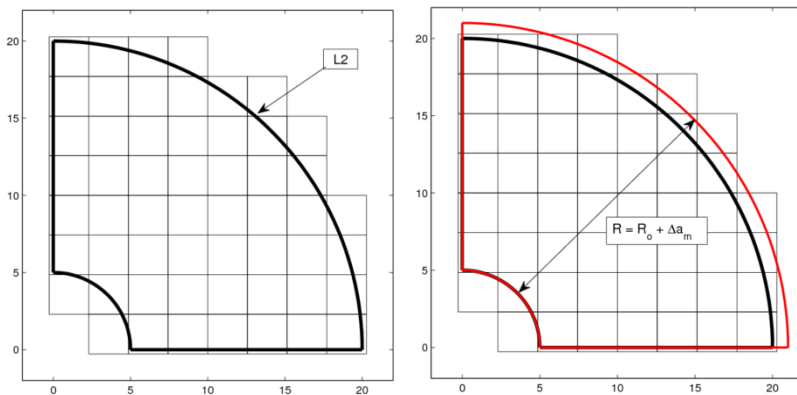


Fig. 1-6 Geometría original y geometría deformada después de aplicar la variación en el radio externo

En el presente trabajo se ha definido el radio externo mediante una spline, no mediante un arco de circunferencia. La spline está definida por 3 puntos, que son los dos extremos de la curva L2 del dibujo y un punto intermedio en la curva.



CAPÍTULO I INTRODUCCIÓN

Hay por tanto 4 variables de diseño: la coordenada x del extremo inferior, la coordenada y del extremo superior y las coordenadas x e y del punto intermedio.

Para el cálculo de sensibilidades de forma se calcularán las cuatro variables de diseño por separado, tal como se ilustra en la siguiente figura:

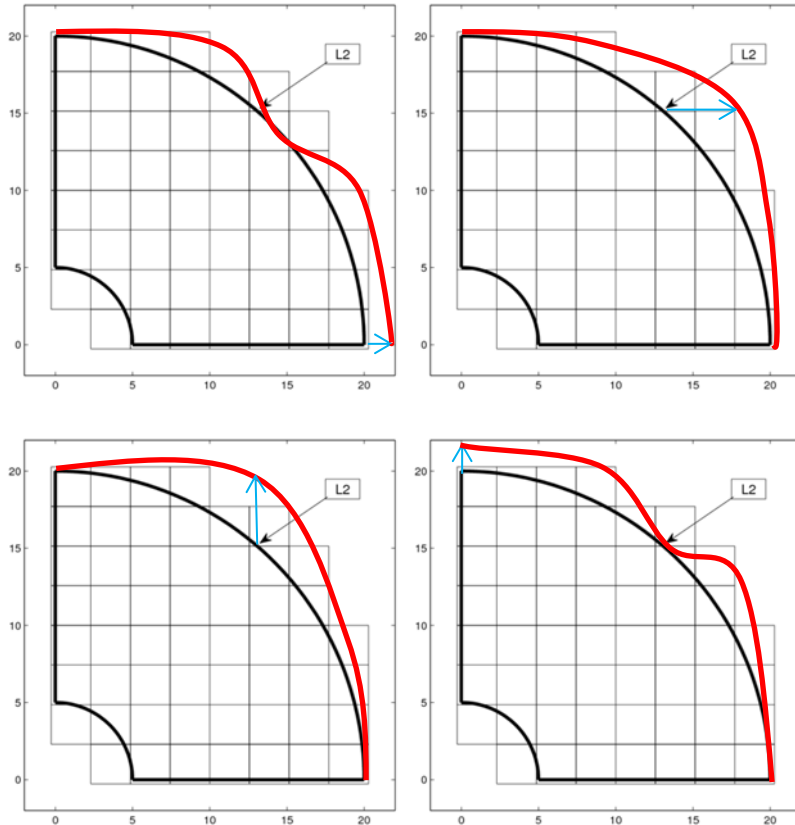


Fig. 1-7 Variación de las variables de diseño 1 a 4 (de izq-der, de arr-ab)

Se puede apreciar que, en este caso, la variación de cualquiera de las variables afectará a todo el contorno externo, debido a la configuración de la spline. Para definir la spline se utilizan ecuaciones de restricción para obligar a la curva a pasar por los puntos seleccionados. Si solo se varía uno, la curva ha de seguir pasando por el resto. Estas ecuaciones de restricción afectarán en el cálculo de las restricciones del problema y sus derivadas. El cálculo será más complejo y añade más mínimos locales, si fuera un arco de circunferencia solo habría un mínimo (que es local y global). De esta forma se puede comprobar la

TRABAJO FIN MÁSTER

influencia de cada variación por separado y si el proceso de optimización es correcto, debería obtener una solución simétrica igualmente.

En este trabajo se ha implementado una forma de combinar más de una variación para una sola variable de diseño. Es decir, según lo anteriormente comentado cada variación en la geometría supone una variable de diseño. Esto supone un análisis de sensibilidades por cada variable. Para minimizar cálculos se puede combinar más de una variación para una sola variable.

Por ejemplo, en el caso anterior se tienen 4 variables de diseño, pero estas se pueden combinar entre sí imponiendo entre ellas restricciones de manera que representen un aumento del radio externo.. También se pueden combinar las variaciones dos a dos para obtener finalmente 2 variables de diseño. Esto se puede utilizar también como restricciones geométricas de cara a la optimización.

Se puede encontrar la función que define y combina las variables de diseño en el anexo, apartado 2.3. Se proporciona como ejemplo la tubería simple que combina las variaciones de la variable 1 junto a la 4 y de la 2 junto a la 3.

1.6. Sensibilidades en CGFEM

En optimización de diseño estructural el análisis de sensibilidades es el cálculo de las derivadas de respuesta estructural (desplazamientos, tensiones, frecuencias naturales...) con respecto a variables de diseño.

El desarrollo inicial del análisis de sensibilidades se centraba en variables de diseño de tamaño, como espesor o área transversal de los componentes. Sin embargo, en muchos problemas estructurales fue necesario considerar la forma como variable de diseño. Esto es particularmente importante en la optimización del diseño de componentes. En el presente trabajo la investigación se centrará en el análisis de sensibilidades de forma y en las variables de diseño de las variables que describen la geometría del componente a optimizar.

La utilidad del análisis de sensibilidades en optimización es evidente como se ha explicado anteriormente. En general, se necesita la información de los gradientes de las funciones objetivo y restricciones se necesitan para la resolución del problema mediante técnicas de programación matemática basadas en gradiente enlazando la etapa de análisis con la de optimización. Además, la



CAPÍTULO I INTRODUCCIÓN

eficiencia de los procedimientos de optimización se basa en técnicas matemáticas que dependen ampliamente en la precisión y coste computacional del cálculo de sensibilidades.

El cálculo de sensibilidades también tiene otras aplicaciones. Si un componente es modificado ligeramente, en lugar de resolver el problema nuevamente, se usan las sensibilidades para extrapolar la respuesta estructural. Así, el coste computacional se reduce sin reducir demasiado la precisión. Esta aproximación se puede usar en software CAE.

En técnicas de optimización multinivel, los problemas tratados son normalmente complejos, con muchas variables de diseño y restricciones. En ese caso se puede dividir el problema en varios niveles para que cada subproblema contenga un número mínimo de variables y restricciones. Estos subproblemas pueden resolverse independientemente con algoritmos estándar. Ya que las variables de diseño de un subproblema se presentan como un conjunto de parámetros del subproblema, la coordinación de la solución se puede hacer mediante las sensibilidades con respecto a esos parámetros.

1.6.1. Análisis de sensibilidades en MEF

Hay muchas referencias en el campo de análisis de sensibilidades de forma. Básicamente hay dos aproximaciones, la continua y la discreta, cada una con sus ventajas e inconvenientes.

En el presente trabajo se va a utilizar la aproximación discreta analítica. La aproximación discreta analítica de cálculo de sensibilidades es la más adecuada en el cálculo hecho por MEF para no introducir errores de discretización adicionales.

Esta aproximación usa la formulación basada en el comportamiento discretizado, del MEF, $\mathbf{K}\cdot\mathbf{U}=\mathbf{f}$, para luego obtener las expresiones del cálculo de sensibilidades derivando analíticamente dicha expresión. Así, derivando $\mathbf{K}\cdot\mathbf{U}=\mathbf{f}$ con respecto a unavariable de diseño a_m resulta:

$$\frac{\partial \mathbf{K}}{\partial a_m} \mathbf{u} + \mathbf{K} \frac{\partial \mathbf{u}}{\partial a_m} = \frac{\partial \mathbf{f}}{\partial a_m}$$

Ec 4

TRABAJO FIN MÁSTER

Reorganizando términos se obtiene:

$$\mathbf{K} \frac{\partial \mathbf{u}}{\partial a_m} = \frac{\partial \mathbf{f}}{\partial a_m} - \frac{\partial \mathbf{K}}{\partial a_m} \mathbf{u} = \mathbf{f}_{ps_m}$$

Ec 5

De donde, resolviendo el sistema de ecuaciones, se obtiene la sensibilidad del desplazamiento. De estas sensibilidades se calculan otras magnitudes.

Para plantear el sistema de ecuaciones de la expresión anterior es necesario evaluar, a nivel de elemento para luego proceder al ensamblado en dicha expresión, las derivadas de todos los términos involucrados, tal y como se muestra a continuación.

1.6.1.1. Derivada de la matriz de rigidez

La matriz de rigidez viene dada por:

$$\mathbf{k}^e = \int_{\Omega^e} \mathbf{B}^T \mathbf{D} \mathbf{B} |J| d\Omega$$

Ec 6

Considerando que la matriz \mathbf{D} es constante, la derivada de \mathbf{D} respecto de las variables de diseño es cero:

$$\frac{\partial \mathbf{k}^e}{\partial a_m} = \int_{\Omega^e} \left[\frac{\partial \mathbf{B}^T}{\partial a_m} \mathbf{D} \mathbf{B} + \mathbf{B}^T \mathbf{D} \frac{\partial \mathbf{B}}{\partial a_m} \right] |J| d\Omega + \int_{\Omega^e} \left[\mathbf{B}^T \mathbf{D} \mathbf{B} \frac{\partial |J|}{\partial a_m} \right] d\Omega$$

Ec 7

Considerando la relación:

$$\frac{\partial \mathbf{B}^T}{\partial a_m} \mathbf{D} \mathbf{B} = \left[\mathbf{B}^T \mathbf{D} \frac{\partial \mathbf{B}}{\partial a_m} \right]^T$$

Ec 8



CAPÍTULO I INTRODUCCIÓN

Simplificando y aprovechando propiedades de matrices simétricas se obtiene:

$$\frac{\partial \mathbf{k}^e}{\partial a_m} = 2 \left[\int_{\Omega^e} \mathbf{B}^T \mathbf{D} \bar{\mathbf{B}} |J| d\Omega \right]_s$$

Ec 9

Con: $\bar{\mathbf{B}} = \frac{\partial \mathbf{B}}{\partial a_m} + \mathbf{B} \frac{|J|'_m}{2|J|}$ donde $|J|'_m = \frac{\partial |J|}{\partial a_m}$, términos estos que serán función de $\frac{\partial \mathbf{x}}{\partial a_m}$ que representa las derivadas de las coordenadas de los puntos materiales con respecto a las variables de diseño, denominadas habitualmente como *campo de velocidades*.

1.6.1.2. Derivada de vector de fuerzas equivalentes

El vector de fuerzas equivalentes en nodos, \mathbf{F} , se puede formar acoplando la fuerza nodal equivalente \mathbf{f}^e de los elementos. Esas fuerzas equivalentes en elementos pueden venir de fuerzas puntuales, volumétricas o fuerzas distribuidas sobre una superficie del elemento y fuerzas equivalentes inducidas por deformación inicial o tensiones.

En general, la sensibilidad con respecto a variables de diseño de las fuerzas nodales equivalentes en cada elemento tendrá dos términos. Uno dependiente de la variación de la fuerza con respecto a las variables de diseño, el otro dependiente de las derivadas tales como $\frac{\partial x}{\partial a_m}$. En el primer caso la expresión que define la dependencia de la fuerza actuante con respecto a las variables de diseño se debe mantener. En el segundo caso las derivadas necesarias son las que también son necesarias para calcular la derivada de la matriz de rigidez.



TRABAJO FIN MÁSTER

1.6.1.3. Formulación de las derivadas de la tensión

Con la expresión general:

$$\boldsymbol{\sigma} = \mathbf{D}\mathbf{B}\mathbf{u}^e$$

Ec 10

Siendo \mathbf{u}^e el vector de desplazamientos nodales del elemento e . derivando se obtiene la expresión que permitirá calcular la derivada de la tensión.

$$\frac{\partial \boldsymbol{\sigma}}{\partial a_m} = \mathbf{D}\mathbf{B} \frac{\partial \mathbf{u}^e}{\partial a_m} + \mathbf{D} \frac{\partial \mathbf{B}}{\partial a_m} \mathbf{u}^e$$

Ec 11

2. Selección del algoritmo de optimización

En trabajos anteriores desarrollados en el DIMM [1] se ha implementado en cgFEM un módulo de cálculo de sensibilidades de forma. Se usará este en el presente trabajo para desarrollar un algoritmo de optimización basado en gradiente.

A continuación se procede a comentar distintos tipos de algoritmos basados en gradientes para después elegir uno.

2.1. Tipos de algoritmos basados en gradiente en problemas de optimización sin restricciones

Se presentan a continuación algunos ejemplos de los algoritmos tomados en consideración para el trabajo. El programa cgFEM implementa únicamente el cálculo de primeras derivadas con respecto a las variable de diseño. Por lo tanto centraremos nuestro estudio en métodos que, a lo sumo, estén basados en primeras derivadas con respecto a las variables en las que se formule el problema.

2.1.1. Método del descenso adaptativo

Método del descenso adaptativo (*gradient descent*) es un algoritmo de optimización de primer orden, es decir usa las primeras derivadas. Para encontrar un mínimo local de una función toma escalones proporcionales al gradiente negativo en el punto actual. Si se tomara en positivo sería el método del ascenso adaptativo (*gradient ascent*), que encuentra un máximo local.



CAPÍTULO II ALGORITMO

El método del descenso adaptativo se basa en que la función $f(\mathbf{a})$ está definida y es diferenciable en las proximidades de un punto \mathbf{p} . $f(\mathbf{a})$ disminuye en la dirección del gradiente negativo $-\nabla f(\mathbf{p})$ y se tiene

$$\mathbf{r} = \mathbf{p} - \gamma \nabla f(\mathbf{a})$$

Ec 12

Para valores pequeños de γ entonces $f(\mathbf{p}) \geq f(\mathbf{r})$. Así pues se comienza con un punto inicial \mathbf{a}_0 y se sigue la secuencia anterior hasta obtener la convergencia para un \mathbf{a}_n .

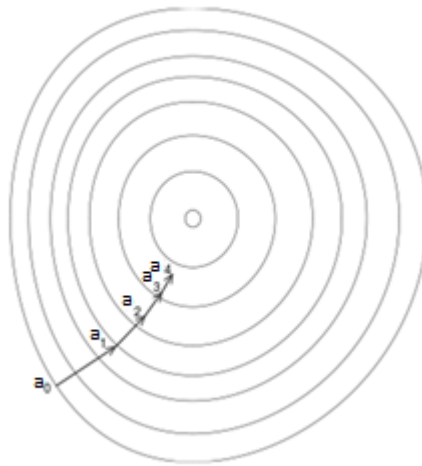


Fig. 2-1 Muestra del gradient descent

Una desventaja de este algoritmo es que para problemas convexos mal condicionados puede no haber convergencia y depende en gran medida del punto inicial tomado. Además no puede usarse para funciones no diferenciables.

TRABAJO FIN MÁSTER

2.1.2. Método de Newton

En Cálculo, el método de Newton es un método iterativo para hallar ceros de una función, $f(\mathbf{a})=0$. En optimización, se aplica a su derivada, $f'(\mathbf{a})=0$.

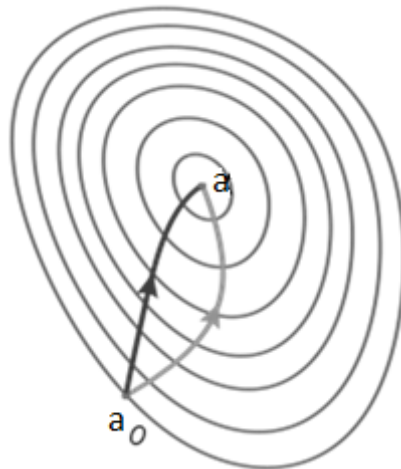


Fig. 2-2 Comparación del gradient descent (claro) y Newton (oscuro). Hace menos iteraciones y usa información de la curvatura para tomar una ruta directa

El método de Newton utiliza la expansión de Taylor de segundo orden, es decir, utiliza segundas derivadas. Finalmente se obtiene:

$$f'(\mathbf{a}_n) + f''(\mathbf{a}_n)\Delta\mathbf{a} = 0$$

$$\mathbf{a}_{n+1} = \mathbf{a}_n - \frac{f'(\mathbf{a}_n)}{f''(\mathbf{a}_n)}$$

Sea \mathbf{a}_* la solución del problema de optimización. Se cumplirá que, teniendo en cuenta que Δx ha de ser cero, se obtiene una función que convergerá a $f'(\mathbf{a}_*)=0$.

La interpretación geométrica del método de Newton es que cada iteración aproxima la función a una función cuadrática alrededor de \mathbf{a}_n .



2.1.3. Programación secuencial cuadrática (SQP) con restricciones

El algoritmo SQP (*Sequential Quadratic Programming* – Programación Secuencial Cuadrática) es un método iterativo de optimización no-lineal. Se usa en problemas en los que la función objetivo es dos veces derivable continua. Solo es necesario asegurar la continuidad de la segunda derivada, no es necesaria calcularla puesto que en el cálculo solo se utiliza la primera derivada.

El método SQP resuelve un subproblema local ajustando una función cuadrática a la función objetivo real en el punto considerado asumiendo adicionalmente una linealización de las restricciones. En el nuevo punto obtenido se vuelve a resolver un problema local cuadrático, lo que va definiendo la secuencia de problemas que conduce a la solución óptima. Si el problema no está restringido entonces el método se reduce al método de Newton para buscar un punto donde el gradiente del objetivo se anula. Si el problema solo tiene restricciones de igualdad es equivalente a usar Newton para las condiciones óptimas de primer orden.

Considerando un problema de programación no-lineal de la forma:

$$\begin{aligned} \min_x f(\mathbf{a}) \\ \text{tal que.} \quad b(\mathbf{a}) \geq 0 \\ \quad \quad \quad c(\mathbf{a}) = 0 \end{aligned}$$

El Lagrangiano para este problema es:

$$\mathcal{L}(\mathbf{a}, \lambda, \sigma) = f(\mathbf{a}) - \lambda^T b(\mathbf{a}) - \sigma^T c(\mathbf{a})$$

Ec 13

Donde λ y σ son multiplicadores de Lagrange. En una iteración \mathbf{a}_k un SQP básico define una dirección apropiada de búsqueda \mathbf{d}_k como solución al subproblema de programación cuadrática.

TRABAJO FIN MÁSTER

$$\begin{aligned} \min_x \quad & f(\mathbf{a}_k) + \nabla f(\mathbf{a}_k)^T \mathbf{d} + \frac{1}{2} \mathbf{d}^T \nabla_{xx}^2 \mathcal{L}(\mathbf{a}_k, \lambda_k, \sigma_k) \mathbf{d} \\ \text{tal que} \quad & b(\mathbf{a}_k) + \nabla b(\mathbf{a}_k)^T \mathbf{d} \geq 0 \\ & c(\mathbf{a}_k) + \nabla c(\mathbf{a}_k)^T \mathbf{d} = 0 \end{aligned}$$

Ec 14

El algoritmo SQP funciona correctamente en problemas en el que el contorno es cambiante, los saltos que adopta son en base al contorno de ese momento, no un valor global. Si un salto proporciona valores inadecuados como infinito, cambia el valor del salto para evitar este problema. Si en el problema no se llega a ninguna solución ya sea por la función objetivo o a las restricciones combina las dos funciones para intentar obtener una solución, que pese a no ser óptima cumple parcialmente las restricciones. Si no cumple las restricciones es capaz de llevar a cabo un análisis de segundo orden en el que si se halla una solución, aunque para ello debe calcular la función de restricciones varias veces prolongando el tiempo de cálculo.



2.2. Elección de algoritmo

Finalmente se escoge el método SQP puesto que es adecuado para la optimización no-lineal. El problema que se va a tratar en el presente trabajo está restringido y tiene restricciones de desigualdad. El método de Newton no es pues el más adecuado. Además se dispone de las primeras derivadas, o sensibilidades, de los valores a optimizar, forma y tensión.

Previsiblemente el método SQP es el que mejor se adapta a nuestro problema. Para el cálculo de las derivadas no es necesario cálculos extras de lo ya obtenido en el cálculo de sensibilidades. Su obtención se explicará en el siguiente apartado.

Como se ha demostrado en trabajos anteriores, se asegura la continuidad en la segunda derivada aunque no se haya comprobado explícitamente para el presente trabajo. Se cumple la condición necesaria del método SQP.

El problema tiene unas restricciones impuestas, de igualdad y desigualdad. Lo cual limita la selección de método. El método de Newton no es capaz de tener en cuenta las restricciones de desigualdad. El método SQP si es adecuado.

Bajo estas premisas se procederá al desarrollo del trabajo.

TRABAJO FIN MÁSTER

2.3. Implementación del algoritmo en cgFEM

Para la implementación del método de optimización se va a utilizar el algoritmo `GlobalSearch` con el calculador `fmincon` de MatLab, que permite obtener un mínimo global aplicando el método SQP a un conjunto de puntos iniciales del espacio de variables de diseño.

El algoritmo `GlobalSearch` se inicia definiendo n geometrías iniciales de valores $\mathbf{x}_0 = (x_{0,1}, \dots, x_{0,n})$. En este trabajo el vector \mathbf{x}_0 es el que contiene las variables de diseño de cada uno de los puntos en los que se aplicará el método SQP mediante el calculador de MatLab `fmincon` y, como se ha visto anteriormente, podrán ser las variables de diseño o una combinación de estas. La aplicación del método SQP a cada geometría inicial conduce a la obtención de un mínimo local. Si converge se guarda el valor de la función objetivo y la violación de las restricciones en una función llamada `score function`.

Después genera una serie de puntos de prueba en base a un criterio que intenta encontrar la mejor solución. A diferencia de otros algoritmos que propone los puntos uniformemente repartidos, el algoritmo de `GlobalSearch` evalúa la `score function` en todos esos puntos y se empieza el proceso de optimización con el punto que haya obtenido el mejor resultado.

De este modo, utiliza el calculador para cada punto restante en caso de poder mejorar la solución cumpliendo las restricciones. En caso de ir encontrando mejores soluciones, las va guardando. Así, si en cálculos posteriores no encuentra una solución mejor, ya la tiene calculada y no es necesario un recálculo de la solución.

En cuanto a la forma de trabajar del calculador `fmincon`, encuentra el mínimo de funciones no lineales de varias variables con restricciones.

$$\min_a f(\mathbf{a}) \text{ tal que } \begin{cases} c(\mathbf{a}) \leq 0 \\ c_{eq}(\mathbf{a}) = 0 \\ \mathbf{A} \cdot \mathbf{a} \leq \mathbf{b} \\ \mathbf{A}_{eq} \cdot \mathbf{a} = \mathbf{b}_{eq} \\ \mathbf{lb} \leq \mathbf{a} \leq \mathbf{ub} \end{cases}$$

Ec 15



CAPÍTULO II ALGORITMO

Donde $c(\mathbf{a})$ y $c_{eq}(\mathbf{a})$ son funciones que devuelven vectores con los valores de las funciones de restricción a evaluar. \mathbf{b} y \mathbf{b}_{eq} son vectores con los valores de las restricciones impuestas al problema mediante las matrices \mathbf{A} y \mathbf{A}_{eq} . Sirven para definir restricciones inherentes al problema, por ejemplo en los problemas que trataremos se usan para definir que el contorno de la geometría no se cruce a sí mismo o se deforme demasiado para evitar soluciones incoherentes o imposibles físicamente. \mathbf{lb} y \mathbf{ub} pueden ser vectores o matrices y sirven para limitar los valores máximos y mínimos de la \mathbf{x} en todo el proceso; por ejemplo que un radio externo en un tubo sea mayor que el radio interno.

Al calcular le es necesario una función objetivo, $f(\mathbf{a})$, y la función de restricción, $c(\mathbf{a})$, pudiendo haber otra función de restricción para restricciones de igualdad, $c_{eq}(\mathbf{a})$. Primeramente se evalúa la función objetivo, obteniendo los valores del término a optimizar y su derivada, o sensibilidades. Después evalúa la función de restricción obteniendo los valores de la restricción y su derivada.

Con los valores de las derivadas se pasa a usar el algoritmo de optimización SQP que determinará cual será el salto, en módulo y dirección que determinarán la siguiente geometría a analizar. La forma de trabajar de este algoritmo se ha comentado anteriormente.

Este proceso se va repitiendo en distintas iteraciones hasta que el algoritmo `GlobalSearch` considera que ha obtenido la solución óptima.

TRABAJO FIN MÁSTER

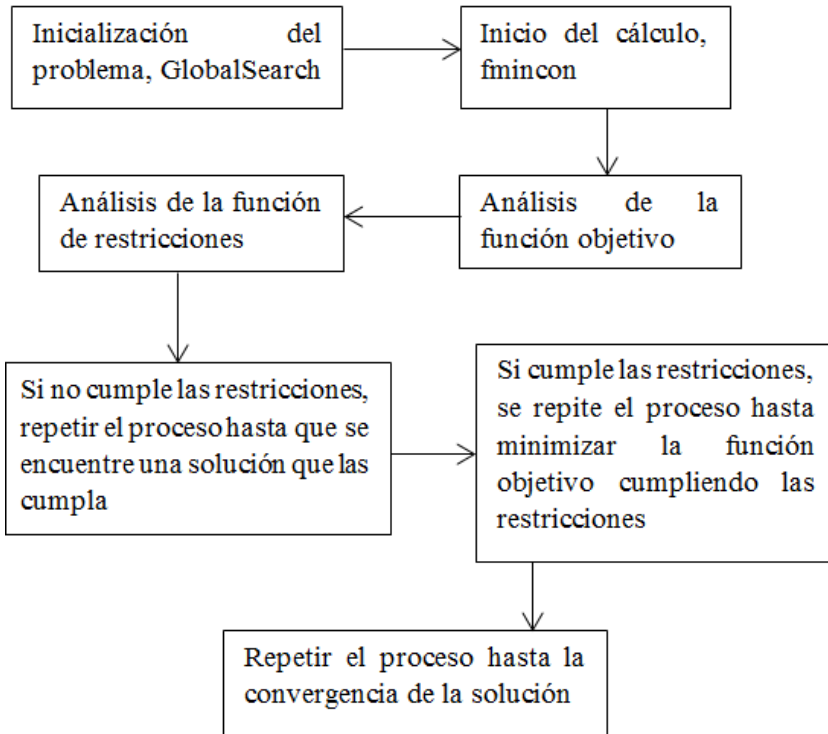


Fig. 2-3 Esquema de procedimiento de proceso de optimización

Se muestran en el apartado 2.1 del anexo las funciones utilizadas en este proceso. Son la función objetivo $f(\mathbf{a})$ y la de restricciones $c(\mathbf{a})$. Además de una función auxiliar que es la que llama al calculador de MatLab `fmincon`.

3. Obtención de las derivadas de las magnitudes llevadas a estudio

La función objetivo está normalmente ligada al volumen de la estructura, mientras que las restricciones lo están a deformaciones, tensiones y desplazamientos. Hará falta calcular sus sensibilidades.

Las sensibilidades de esos valores se obtienen derivando las expresiones utilizadas en el cálculo del MEF (Ec 7). La matriz **B** depende de las coordenadas nodales, con lo que su sensibilidad se puede obtener de la malla de sensibilidades. Esta técnica permite obtener sensibilidades de primer orden de la matriz **K**, del vector de fuerzas nodales equivalentes **f** y cualquier otra integral relacionada.

Para el desarrollo del presente trabajo es necesario el cálculo de las derivadas con respecto a las variables de diseño (sensibilidades), de las magnitudes llevadas a estudio.

En este caso se trata del módulo del desplazamiento, las tensiones, más concretamente de las tensiones de Von Mises, el volumen y el estimador de error.

3.1. Módulo de desplazamientos

Los desplazamientos suelen ser un parámetro a tener en cuenta, aunque en el presente trabajo no se van a utilizar en los ejemplos numéricos.

Los datos obtenidos en el cálculo FEM son la deformación U_x y U_y pero no su módulo. Se tiene la siguiente propiedad:

$$\mathbf{U} = (U_x, U_y)^t$$
$$|\mathbf{U}| = \sqrt{U_x^2 + U_y^2} = \sqrt{\mathbf{U}^T \mathbf{U}}$$



Derivando la expresión:

$$\begin{aligned}\frac{\partial |\mathbf{U}|}{\partial a_m} &= \frac{1}{2} [\mathbf{U}^T \mathbf{U}]^{-\frac{1}{2}} \frac{\partial}{\partial a_m} [\mathbf{U}^T \mathbf{U}] = \frac{1}{2} [\mathbf{U}^T \mathbf{U}]^{-\frac{1}{2}} 2 \frac{\partial |\mathbf{U}^T|}{\partial a_m} |\mathbf{U}| \\ \frac{\partial |\mathbf{U}|}{\partial a_m} &= [\mathbf{U}^T \mathbf{U}]^{-\frac{1}{2}} \frac{\partial |\mathbf{U}^T|}{\partial a_m} |\mathbf{U}|\end{aligned}$$

Ec 16

3.2. Tensión de Von Mises

La tensión de Von Mises se obtiene mediante la expresión:

$$\sigma_v = \frac{1}{\sqrt{2}} \sqrt{(\sigma_x - \sigma_y)^2 + (\sigma_y - \sigma_z)^2 + (\sigma_z - \sigma_x)^2 + 6(\tau_{xy}^2 + \tau_{yz}^2 + \tau_{xz}^2)}$$

Ec 17

Suponiendo un vector de tensiones $\boldsymbol{\sigma} = (\sigma_x, \sigma_y, \tau_{xy})^t$ de tensiones axial, radial y tangencial respectivamente y los casos de deformación plana y tensión plana cuyas condiciones son:

Deformación Plana:	$\sigma_z = \vartheta(\sigma_x - \sigma_y)$	$\tau_{yz} = 0$	$\tau_{xz} = 0$
Tensión Plana:	$\sigma_z = 0$	$\tau_{yz} = 0$	$\tau_{xz} = 0$

Siendo ϑ el coeficiente de Poisson. Se simplifica la Ec 17 a:

$$\sigma_v = \frac{1}{\sqrt{2}} \sqrt{\boldsymbol{\sigma} \mathbf{C} \boldsymbol{\sigma}^t}$$

Ec 18

TRABAJO FIN MÁSTER

$$\mathbf{C} = \begin{bmatrix} 2(1 - \vartheta + \vartheta^2) & -1 - 2\vartheta + 2\vartheta^2 & 0 \\ -1 - 2\vartheta + 2\vartheta^2 & 2(1 - \vartheta + \vartheta^2) & 0 \\ 0 & 0 & 6 \end{bmatrix} \quad \text{Deformación Plana}$$

$$\mathbf{C} = \begin{bmatrix} 2 & -1 & 0 \\ -1 & 2 & 0 \\ 0 & 0 & 6 \end{bmatrix} \quad \text{Tensión Plana}$$

Derivando la Ec 18 con respecto a las variables de diseño a_m se obtiene la expresión:

$$\frac{\partial \sigma_v}{\partial a_m} = \frac{1}{\sqrt{2}} \frac{1}{2} \frac{1}{\sqrt{\boldsymbol{\sigma} \mathbf{C} \boldsymbol{\sigma}^t}} \frac{\partial}{\partial a_m} [\boldsymbol{\sigma} \mathbf{C} \boldsymbol{\sigma}^t]$$

Se toman las propiedades:

$$\boldsymbol{\sigma} \mathbf{C} \boldsymbol{\sigma}^t = 2\sigma_v^2$$

$$\frac{\partial}{\partial a_m} [\boldsymbol{\sigma} \mathbf{C} \boldsymbol{\sigma}^t] = 2 \frac{\partial \boldsymbol{\sigma}}{\partial a_m} \mathbf{C} \boldsymbol{\sigma}^t$$

Se obtiene finalmente:

$$\frac{\partial \sigma_v}{\partial a_m} = \frac{1}{2} \frac{1}{\sigma_v} \frac{\partial \boldsymbol{\sigma}}{\partial a_m} \mathbf{C} \boldsymbol{\sigma}^t$$

Ec 19

Siendo $\frac{\partial \boldsymbol{\sigma}}{\partial a_m}$ las derivadas obtenidas en el cálculo de sensibilidades.

En el anexo del manual del programador, en el apartado 2.4, se muestra la función que calcula la tensión de Von Mises y su derivada.



3.3. Volumen

La derivada del volumen para cada elemento se puede calcular numéricamente mediante la siguiente expresión:

$$\frac{\partial V^e}{\partial a_m} = \sum_{i=1}^{n_{GP}} \mathbf{dJ}_i |J| H_i$$

Ec 20

La ecuación es por elemento, haciendo un sumatorio para todos los elementos se obtiene el volumen total. El sumatorio es por los puntos de Gauss del elemento.

El término $|J|H_i$ es el área asociada a cada punto de Gauss, y es un valor conocido y previamente calculado por ejemplo para evaluar la Ec 9 . El término

$$\mathbf{dJ} = \frac{|J|'}{|J|} \text{ siendo } |J|' = \frac{\partial |J|}{\partial a_m}$$

Obteniendo ese último término mencionado, cuyos cálculos se realizan con factores ya calculados previamente, se tiene el valor de la derivada del volumen. Estos valores forman parte de la función objetivo, con lo que su precisión será determinante en el resultado final.

3.4. Estimación del error

La norma energética se utiliza para cuantificar la calidad de la solución:

$$\|\mathbf{u}\|_E^2 = \int_{\Omega} \boldsymbol{\sigma}^T \mathbf{D}^{-1} \boldsymbol{\sigma} d\Omega \approx \int_{\Omega} \boldsymbol{\sigma}_{fe}^T \mathbf{D}^{-1} \boldsymbol{\sigma}_{fe} d\Omega$$

Ec 21

TRABAJO FIN MÁSTER

El error asociado con cada solución de elementos finitos es evaluado usando el estimador de error de Zienkiewicz y Zhu:

$$\|\mathbf{e}\|_{E_e}^2 \approx \eta_e^2 \int_{\Omega_e} (\boldsymbol{\sigma}^* - \boldsymbol{\sigma}_{ef})^T \mathbf{D}^{-1} (\boldsymbol{\sigma}^* - \boldsymbol{\sigma}_{ef}) d\Omega$$

Ec 22

En este trabajo las tensiones $\boldsymbol{\sigma}^*$ son recuperadas a partir de las tensiones de EF $\boldsymbol{\sigma}^h$ usando la técnica de tipo Superconvergent Patch Recovery SPR-C desarrollada en el DIMM.

El estimador de error global η^2 se obtiene como suma de todos los estimadores por elemento.

Este estimador del error es bastante robusto, fiable y supone poco coste. Los tamaños de elemento para una nueva malla se obtienen usando una estrategia de remallado que se explicará en próximos apartados.

Derivando la ecuación del estimador del error:

$$\frac{\partial \eta_e^2}{\partial a_i} = \int_{\Omega_\xi} \left[\begin{array}{l} \left(\frac{\partial \boldsymbol{\sigma}^*}{\partial a_i} - \frac{\partial \boldsymbol{\sigma}_{ef}}{\partial a_i} \right)^T \mathbf{D}^{-1} (\boldsymbol{\sigma}^* - \boldsymbol{\sigma}_{ef}) \|\mathbf{J}\| + \\ (\boldsymbol{\sigma}^* - \boldsymbol{\sigma}_{ef})^T \mathbf{D}^{-1} \left(\frac{\partial \boldsymbol{\sigma}^*}{\partial a_i} - \frac{\partial \boldsymbol{\sigma}_{ef}}{\partial a_i} \right) \|\mathbf{J}\| + \\ (\boldsymbol{\sigma}^* - \boldsymbol{\sigma}_{ef})^T \mathbf{D}^{-1} (\boldsymbol{\sigma}^* - \boldsymbol{\sigma}_{ef}) \|\mathbf{J}\| \text{tr} \left(\mathbf{J}^{-1} \frac{\partial \mathbf{J}}{\partial a_i} \right) \end{array} \right] d\xi_1 d\xi_2$$

Ec 23

4. Proyección de magnitudes entre distintas geometrías

El propósito de hacer una proyección de magnitudes es disponer de una malla de una geometría nueva utilizando los datos de sensibilidades de una geometría anterior. Para cada geometría se ha de realizar un nuevo cálculo MEF con una malla completamente nueva. Sin embargo, no es necesaria una malla completamente nueva usando este procedimiento.

Esto acelera el proceso de mallado, que supone un coste computacional elevado (en los problemas complejos supone la mayor parte del tiempo de cálculo). Para los procesos de optimización este es un hecho muy importante porque se puede disminuir el tiempo de cálculo por cada iteración realizada.

Se puede utilizar para procesos evolutivos o basados en gradiente anteriormente comentados. Aunque se hará referencia a los procesos evolutivos que son más complejos, dado que puede haber cambios de forma.

4.1. Introducción

La calidad de la solución depende de la calidad de la malla empleada. Para crear una malla adecuada se puede seguir dos estrategias:

- Adaptar una única malla a todas las geometrías. Esto puede generar elementos deformados o degenerados, teniendo que tomar medidas para minimizar los errores generados.
- Realizar un remallado adaptativo para cada diseño. Asegura buena calidad pero aumenta el coste computacional.

Con el procedimiento de proyección se pueden evitar las desventajas de ambas estrategias. En este caso se puede hacer un cálculo de referencia del cual obtener las sensibilidades o ir actualizando la referencia.



La información proyectada proporciona una malla adaptada para cada diseño de una vez, disminuyendo el coste computacional comparado con un malla adaptativo normal.

4.2. Metodología

Se presenta un esquema de la metodología en la Fig. 4-1. El proceso es como se explica a continuación. Para el caso del método basado en gradiente los individuos no son arbitrarios como lo podría parecer en el método evolutivo. A partir de una solución se obtienen distintos individuos como posible geometría en el proceso de optimización, dependiendo del salto que se considere u otros factores como se ha explicado en apartados anteriores. La metodología será pues similar en el caso del método evolutivo y del método basado en gradiente.

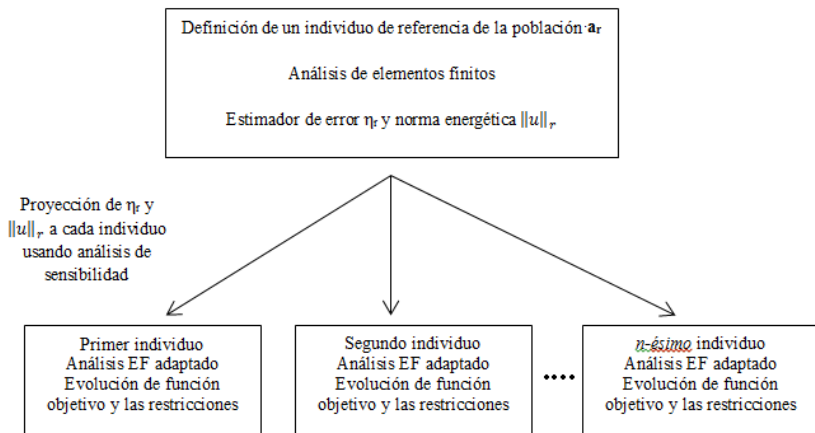


Fig. 4-1 Esquema de la metodología

TRABAJO FIN MÁSTER

1. Un individuo específico se selecciona como referencia de entre toda la población. Se tomar la referencia como fija, aunque puede dar mejor resultado usar una referencia nueva en cada generación (usando la media de las variables de diseño de los individuos de cada generación). Teniendo una población P se define:

$$a^r = \frac{1}{P} \sum_{p=1}^P a^p$$

Ec 24

Donde a^p indica los valores de las variables de diseño \mathbf{a} correspondientes al p -ésimo individuo usando la parametrización seleccionada.

2. El diseño de referencia es analizado usando una aproximación de elementos finitos normal. Incluyendo un mallado adaptativo para asegurar buena calidad de resultados.
3. Una vez obtenida una malla que proporcione resultados de la precisión deseada, se calcula un análisis de sensibilidades completo de todas las magnitudes relacionadas en el proceso de mallado adaptativo.
4. Para cada uno de los individuos a analizar, los valores de las magnitudes relacionadas en el proceso de mallado adaptativo se proyectan desde la referencia usando el correspondiente análisis de sensibilidad. La proyección proporciona sin cálculos adicionales una aproximación de los valores que se obtendrían si se hiciera un cálculo de MEF para cada individuo.
5. Se genera una malla adaptada para cada individuo usando los valores proyectados. Esta malla adaptada se usa para evaluar la función objetivo y las restricciones del individuo.



4.2.1. Parametrización del problema

Cada geometría se representa usando puntos de definición que especifican algunas curvas de interpolación. Las curvas usadas aquí son rectas, splines y NURBS, que ya se han desarrollado en anteriores trabajos. [1]

4.2.2. Proceso estándar de análisis

El procedimiento de análisis adaptativo para una geometría que se va a utilizar es:

1. Se obtiene una malla geoméricamente adaptada y se realiza el análisis FEM.
2. Se obtiene una malla adaptada según la estimación de error y se realiza el análisis FEM para el resto de mallas del proceso adaptativo.

Para generar las mallas de la primera geometría a analizar se usará el procedimiento adaptativo estándar del código cgFEM. Para las subsiguientes geometrías se usará el método de la proyección que se comentará más adelante. Después se realiza el análisis FEM.

4.2.3. Estimación del error

Es necesaria la obtención del estimador del error y de su derivada o sensibilidad. Esto ya se ha explicado en el apartado 3.4.

4.3. Proyección de cada diseño y definición de la malla adaptada

Una vez el nuevo diseño está definido, los nuevos valores del estimador del error, la “energía” y las coordenadas de la malla se puede proyectar de la referencia para cada diseño distinto como:

$$(x, y)^p = (x, y)^r + \sum_i^n (x_i^p - x_i^r) \left(\frac{\partial x}{\partial a_m}, \frac{\partial y}{\partial a_m} \right)$$

$$\eta^{p^2} = \eta^{r^2} + \sum_i^n (x_i^p - x_i^r) \frac{\partial \eta^2}{\partial a_m}$$

$$\|u\|_E^{p^2} = \|u\|_E^{r^2} + \sum_i^n (x_i^p - x_i^r) \frac{\partial \|u\|_E^2}{\partial a_m}$$

Ec 25

Los valores proyectados proporcionan la información necesaria para llevar a cabo un remallado sobre el siguiente diseño antes de realizar ningún cálculo. El estimador de error calculado a posteriori se transforma en uno a priori.

Esta proyección es muy importante porque permite el control de calidad de la malla para cada diseño previo a un nuevo cálculo. Los valores proyectados se usan para crear la información de la malla a generar en cada diseño.

La generación de una nueva malla en el procedimiento de remallado requiere la definición de un criterio de optimalidad de malla. En este trabajo se considera óptima la malla cuando el error está distribuido equitativamente entre todos los elementos de la malla a crear. Este criterio es equivalente al de minimización del número de elementos en la nueva malla para obtener el error deseado.

Una vez el análisis de un diseño dado se termina, se comprueba la calidad de los resultados con el estimador del error. Según lo esperado, en la mayoría de casos el error estimado estará por debajo de la tolerancia prescrita. Si no, se ejecutará un nuevo procedimiento de remallado adaptativo para obtener un nuevo conjunto de resultados de buena calidad.



4.4. Implementación

En el presente trabajo se ha llevado a cabo una implementación del método de la proyección. Se ha dispuesto una rutina de cálculo en el cgFEM para proyectar el error.

Como por falta de tiempo no se ha podido implementar junto al proceso de optimización se han tomado 3 geometrías arbitrarias para comprobar la implementación. Estas, en el proceso completo vendrían dadas por el optimizador. Sin embargo, como al proceso de proyección no le influye de donde provenga la geometría nueva se han hecho las pruebas necesarias mediante este método.

En primer lugar, se realiza un cálculo FEM de la primera geometría. Una vez hecho, se guardan los resultados que se usarán de referencia para la siguiente geometría. Principalmente son datos de la geometría, parámetros, elementos y nodos. También se guardan los valores del error por elemento, que serán los valores a proyectar.

Cuando se tienen valores de referencia en cada iteración se utiliza una subrutina para proyectar el error. Utilizando los valores de referencia, se pueden obtener los valores estimados en la nueva geometría. Para una propiedad cualquiera x se tiene:

$$x = x_0 + \sum \frac{\partial x}{\partial a_m} \Delta a_m$$

Ec 26

Siendo x_0 el valor de la propiedad de referencia, $\frac{\partial x}{\partial a_m}$ la derivada de la propiedad con respecto a las variables de diseño y Δa_m la variación de la variable de diseño de la nueva geometría y la de referencia. En este caso esa variación geométrica será conocida.

En primer lugar se proyecta la malla de la geometría de referencia sobre la nueva geometría. Para ello se utiliza la Ec 26 que nos permitirá proyectar las coordenadas de los nodos, contorno y posiciones de los puntos de Gauss sobre la nueva geometría. En este caso la derivada son los campos de velocidad. Se obtiene la nueva malla no cartesiana, puesto que la cartesiana original ha sido deformada en el proceso.

TRABAJO FIN MÁSTER

Del mismo modo, se procede al cálculo del nuevo valor del error por elemento. Obteniendo el área del elemento se puede obtener la densidad del error por elemento dividiendo el valor del error por el área del elemento. Este valor es el que se proyectará sobre la nueva malla.

Para proyectar se utilizan los puntos de Gauss de la nueva malla cartesiana en base a la nueva geometría. Se realiza un procedimiento que determina la posición de cada punto de Gauss, que estarán sobre un único elemento deformado de la malla anterior. A cada punto se le asigna el valor de densidad del error del elemento deformado sobre el que se encuentre.

Así pues, se ha proyectado el valor del error de la geometría de referencia sobre la nueva. Se podría después obtener el valor de error por elemento de la nueva malla a partir de los valores en los puntos de Gauss., ya que hay un valor asignado de área a cada punto de Gauss.

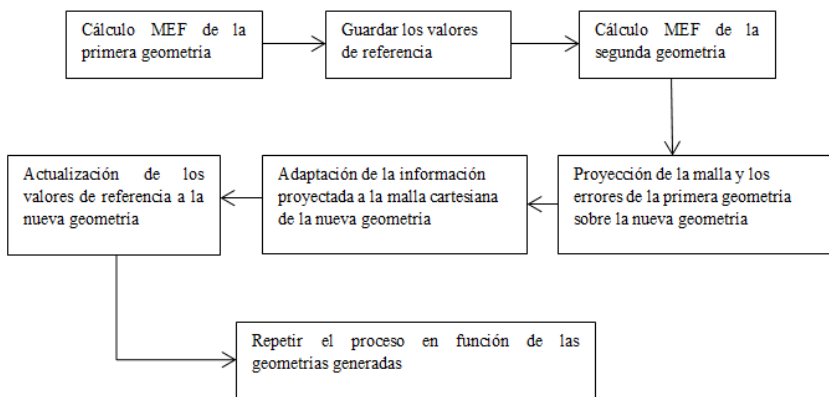


Fig. 4-2 Esquema de procedimiento de proceso de proyección

A continuación se muestra un ejemplo de cálculo del proceso. Se definen tres geometrías distintas, aunque de variaciones pequeñas.

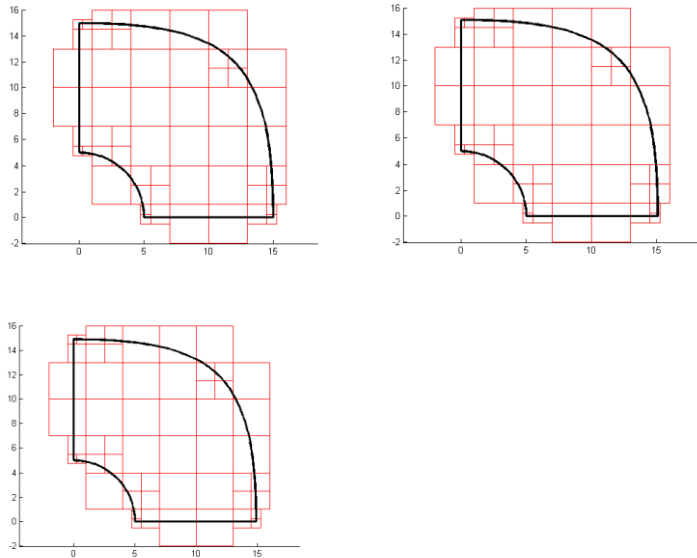


Fig. 4-3 Geometrías de ejemplo con sus mallas (Geo1 arr-izq, Geo2 arr-der, Geo3 ab-izq)

Se lleva a cabo la proyección del error obtenido en la primera geometría sobre la segunda geometría como se muestra en la siguiente figura:

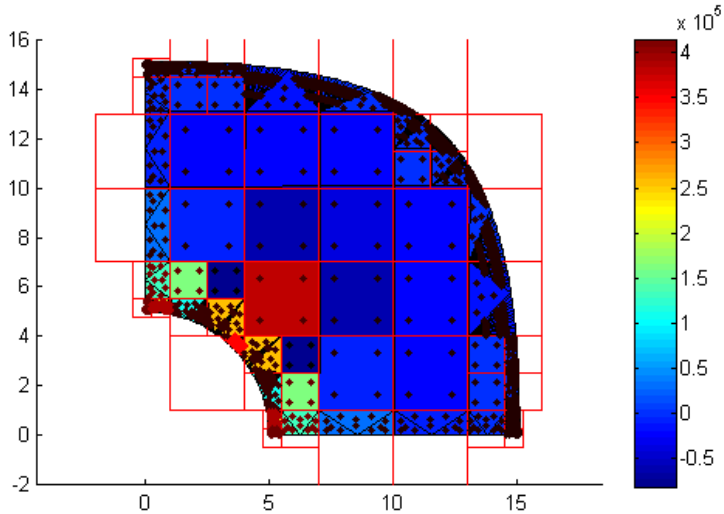


Fig. 4-4 Incremento del error al proyectar sobre la segunda geometría

TRABAJO FIN MÁSTER

La segunda geometría ha sufrido un pequeño aumento de radio con respecto a la original.

Se puede apreciar que quizá la malla pueda ser un poco basta ya que un elemento con bastante error está rodeado con otros elementos de error mucho más reducido. En el gráfico también se muestran los puntos de Gauss en los que se realiza la proyección, siendo de color rojo oscuro los que tienen el mínimo error y de rojo claro los de máximo error. Se muestra también que los elementos de contorno tienen una configuración especial según el programa cgFEM. El elemento es el mismo pero internamente está dividido en triángulos en los cuales se definen los puntos de Gauss dependiendo de la forma concreta de cada triángulo.

En referencia a los puntos de Gauss, se puede ver que la zona de mayor error se concentra en el radio interior. Esta zona es la más crítica, es donde se encuentra la mayor tensión y la que más puede influir en el proceso de optimización.

Se lleva a cabo la proyección del error obtenido en la segunda geometría sobre la tercera geometría como se muestra en la siguiente figura:

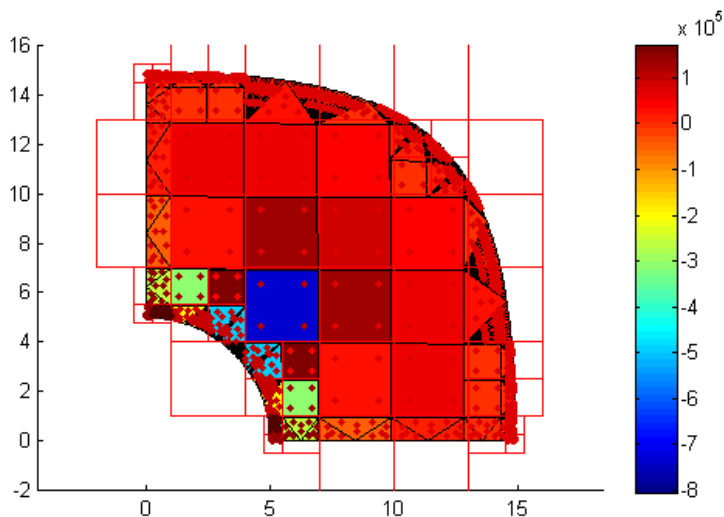


Fig. 4-5 Incremento del error al proyectar sobre la tercera geometría

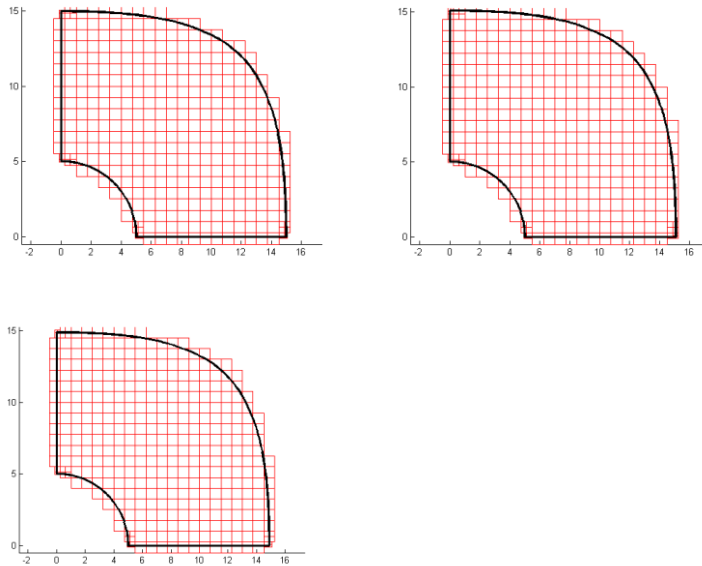
En este caso la tercera geometría ha sufrido una pequeña reducción con respecto a la primera. Es por eso que ahora hay errores negativos y la leyenda de



CAPÍTULO IV PROYECCIÓN

colores es la opuesta al caso anterior. Pese a este hecho, se puede apreciar el mismo comportamiento que antes. La zona con mayor error en módulo es la del radio interior.

Para comprobar la dependencia del error se vuelve a realizare la prueba con las mismas geometrías pero con una malla más fina.



**Fig. 4-6 Geometrías de ejemplo con sus mallas finas
(Geo1 arr-izq, Geo2 arr-der, Geo3 ab-izq)**

Para una mejor visualización se han eliminado del dibujo los puntos de Gauss. Se pasa al proceso de proyección:

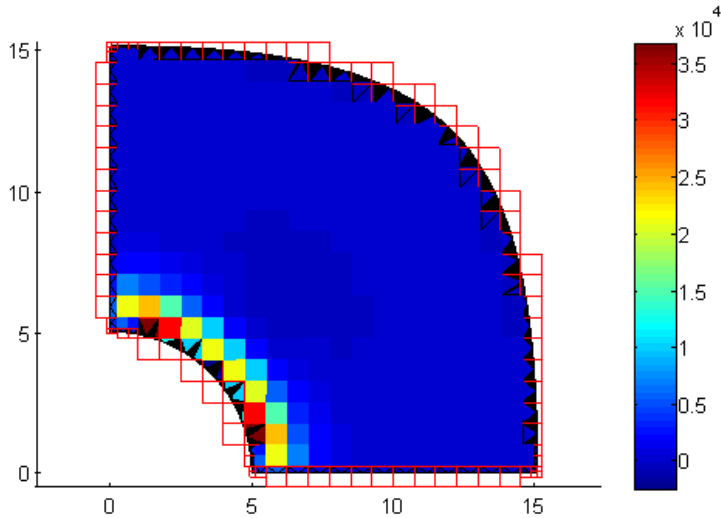


Fig. 4-7 Incremento error al proyectar sobre la segunda geometría con malla fina

Se comprueba que el error se ha visto reducido en un orden de magnitud. En el caso anterior el error máximo era aproximadamente $4 \cdot 10^5$ y ahora es $3.5 \cdot 10^4$. Se ha bajado el error haciendo la malla más fina. Además, el error está ahora concentrado en el radio interno.

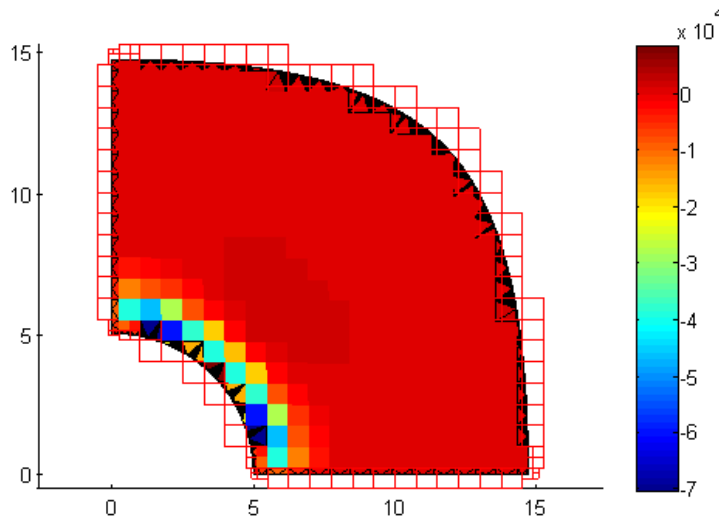


Fig. 4-8 Incremento error al proyecta sobre la tercera geometría con malla fina



CAPÍTULO IV PROYECCIÓN

Se comprueba para la tercera geometría el mismo resultado que en el caso de malla basta, pero con la reducción del error en módulo y la concentración del mayor error en el radio interno.

Como una última comprobación se va a probar el proceso para geometrías significativamente diferentes entre sí. El proceso de optimización podría proporcionar geometrías muy dispares y para ello se prueba el proceso.

Se van a utilizar las siguientes geometrías:

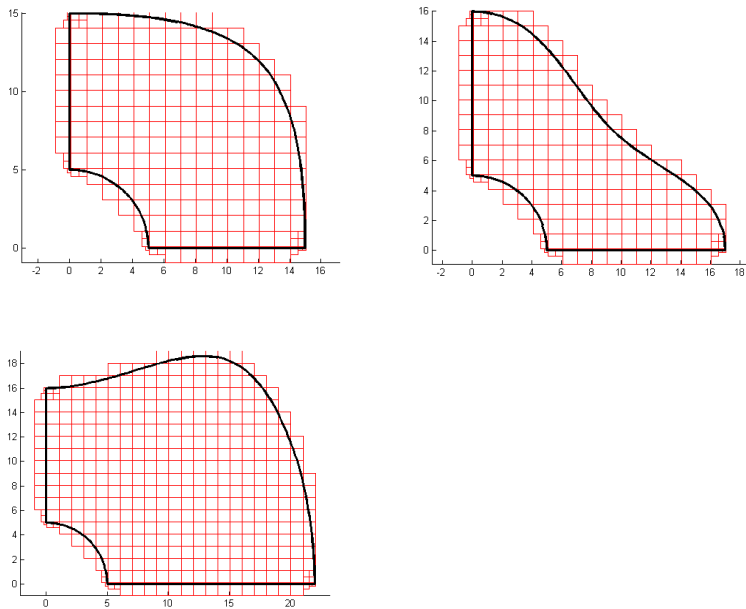


Fig. 4-9 Geometrías de ejemplos diferenciados y su malla (Geo1 arr-izq, Geo2 arr-der, Geo3 ab-izq)

Se puede apreciar una clara diferencia entre las geometrías. Se pasa pues al proceso de proyección.

TRABAJO FIN MÁSTER

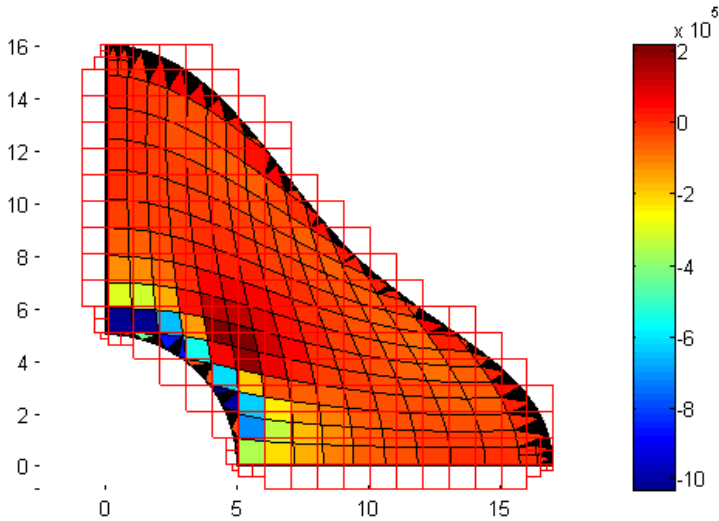


Fig. 4-10 Proyección del error en la segunda geometría distinta

Esta geometría ha sido expuesta a un aumento por los extremos y una reducción por la zona central del radio externo. Se tienen los dos casos, zonas con error positivo y zonas con error negativo. Aunque como era de esperar, el mayor error en módulo se encuentra en la zona del radio interno.

También se puede apreciar los elementos deformados de la malla anterior, efecto que en el caso anterior no se apreciaba porque las deformaciones eran imperceptibles. A partir de esos elementos deformados se proyecta el error a los puntos de Gauss coincidentes de la nueva malla cartesiana, tal y como se ha comentado anteriormente.

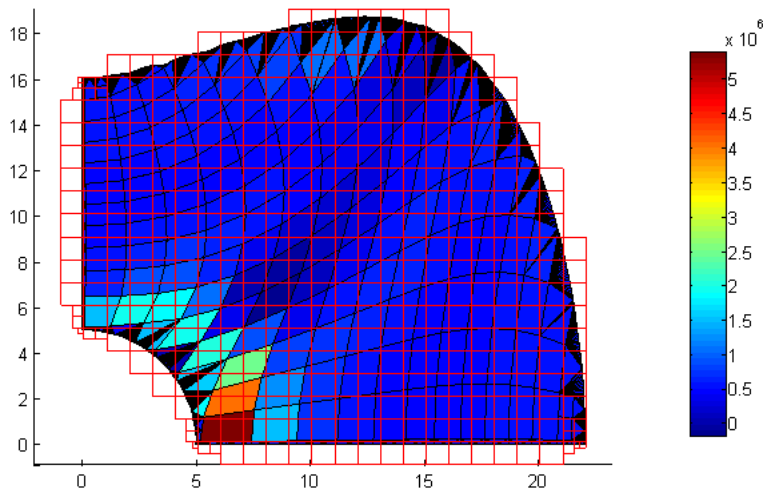


Fig. 4-11 Proyección del error en la tercera geometría distinta

En este caso las dimensiones de la geometría se ha aumentado con respecto a la anterior. El efecto de los elementos deformados es mucho más apreciable. Se aprecia que los elementos cercanos al radio interno se deforman más en comparación a otros cercanos que han sufrido una deformación similar. Este efecto viene producido por el campo de velocidades, que determina la deformación que tendrá cada elemento.

Como apunte final, la variación del error se cuantifica según la variación de la geometría. Por lo tanto, una variación de geometría positiva supondrá un aumento del error y una variación negativa supondrá una reducción del error (o mejor dicho un aumento del error en módulo).

En la siguiente figura se puede ver un detalle de los elementos deformados y los puntos de Gauss.

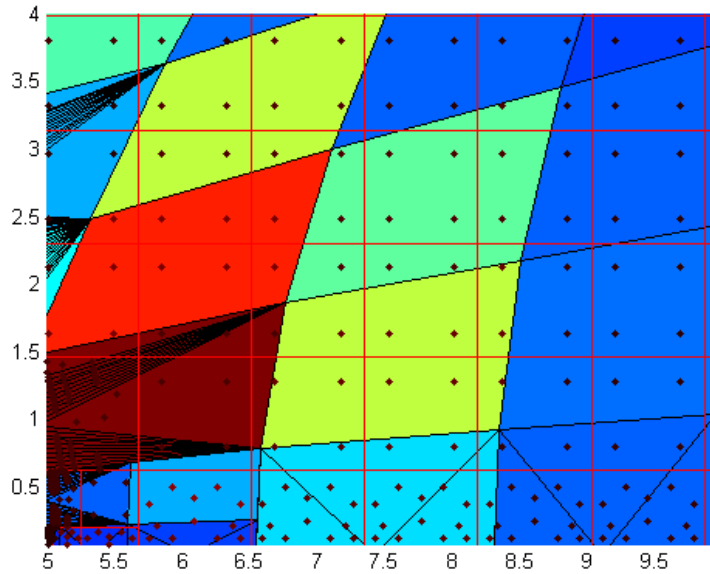


Fig. 4-12 Detalle de elementos deformados y puntos de Gauss

Los puntos de Gauss coincidentes a un determinado elemento deformado recibirán el valor de la densidad de error de dicho elemento. En un mismo elemento de la nueva malla puede haber por tanto puntos de Gauss con distinto valor de error proyectado.

5. Ejemplos numéricos

En este apartado se va a llevar a cabo varios análisis para comprobar que los resultados obtenidos son los esperados. Se van a calcular distintas geometrías con distintas opciones de optimización con el fin de evaluar la robustez del proceso utilizado.

Las tres primeras geometrías analizadas corresponden a un modelo donde se ha de optimizar la superficie externa de la sección transversal de una tubería de la que se ha modelado un cuarto. Cada una de estas tres geometrías tiene una superficie interna distinta. En el primer caso la superficie interna es circular, en el segundo caso es una superficie cuadrada con esquinas redondeadas, y en el último caso la superficie interna es una superficie elíptica. Finalmente se analizará otro ejemplo correspondiente a la optimización de la geometría de una presa de gravedad.

Por lo general, para el cálculo de estos ejemplos se han utilizado los siguientes parámetros:

Tipo de elemento: Cuadrático

Mallado: SquareMesh (malla de cuadriláteros cuadrados)

Error: 0.0001%

Nivel inicial de malla: 5

Nivel final de malla: 11

Nº iteraciones de malla: 3

Para los valores que cambien se comentan en cada caso.

En este apartado se hace referencia a un valor c que indicará si se ha superado la tensión límite, definido por:

$$c = \sigma_{Max} - \sigma_{Adm}$$

Ec 27

Siendo σ_{Adm} la tensión máxima admisible, especificada en cada caso. Para cumplir la restricción:

$$c \leq 0$$



5.1. Tubería con superficie interna circular

En este caso se tiene una tubería sometida a presión interna con un radio interno de 5 cm y radio externo a optimizar. Esta geometría tiene 4 parámetros que se pueden cambiar para la optimización. La coordenada x del extremo inferior del arco del radio externo (parámetro 1); las coordenadas x e y de un punto intermedio en el arco (parámetros 2 y 3 respectivamente); y la coordenada y del extremo superior (parámetro 4).

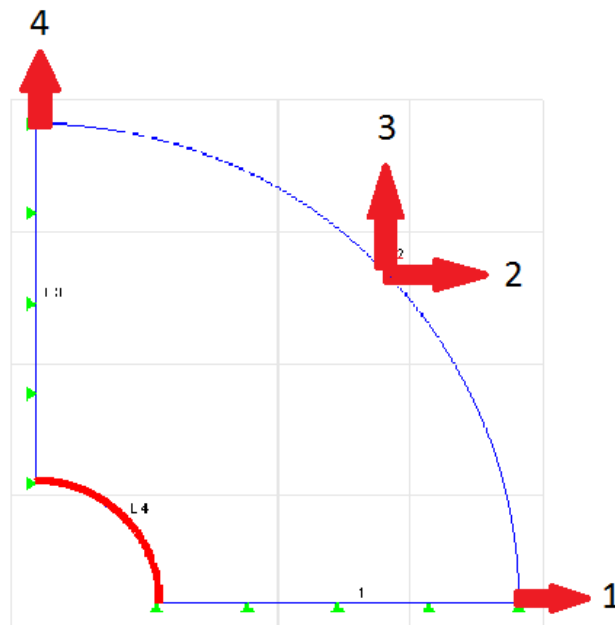


Fig. 5-1 Geometría original y parámetros geométricos

Dicho arco está representado mediante una spline, no mediante un arco de circunferencia. Esto da la posibilidad de que la solución pueda no ser simétrica y el proceso de optimización así lo calcule. Para ello se considera los 4 parámetros en tres casos distintos: a) relacionados entre sí en una sola variable de diseño que representaría un aumento de radio uniforme a lo largo de todo el contorno; b) en dos variables de manera que los parámetros 1 y 4 tomen siempre los mismos valores y que los parámetros 2 y 3 tomen también idénticos valores entre sí; y c) tomando cada uno de los cuatro parámetros valores independientes del resto.

TRABAJO FIN MÁSTER

La función objetivo del problema es el volumen y la restricción es la tensión máxima admisible, que en este caso es de $2 \cdot 10^6$ Pa. Las restricciones de este problema son simplemente que las variables deben tener un mínimo valor, en este caso las variables han de ser mayores al radio interno. Como límite superior se define un valor de 30 para asegurar que en ese tramo encuentra la solución óptima pero sin que un tamaño excesivo ralentice los cálculos.

De este caso se tiene una solución exacta para valores de:

- Presión: 900000 Pa
- Módulo Elástico: $1.01 \cdot 10^7$ Pa
- Coeficiente de Poisson: 0.3

La solución exacta en este caso corresponde a una superficie externa circular. Dado que se conoce de manera analítica las tensiones en una tubería sometida a presión interna, podremos calcular el radio externo óptimo de la misma. Las tensiones exactas en este caso vienen dada por las siguientes expresiones, en las cuales se han tenido en cuenta las coordenadas de un punto (x, y) :

$$r = \sqrt{x^2 + y^2}$$

$$k = \frac{b}{a}$$

Siendo b el radio externo y a en radio interno. Se tiene la tensión en coordenadas cilíndricas:

$$\sigma_r = \frac{p}{k^2 - 1} \left(1 - \frac{b^2}{r^2} \right)$$

$$\sigma_t = \frac{p}{k^2 - 1} \left(1 + \frac{b^2}{r^2} \right)$$

Ec 28

Siendo p el valor de la presión, σ_r y σ_t la tensión radial y tangencial respectivamente. La solución óptima en este caso será:

- Solución: 10.666 cm de radio externo, volumen 69.985 cm^3



La solución óptima teórica es simétrica. Se va a comprobar pues este hecho en los siguientes casos.

Se tiene la distribución en tensiones de la solución exacta:

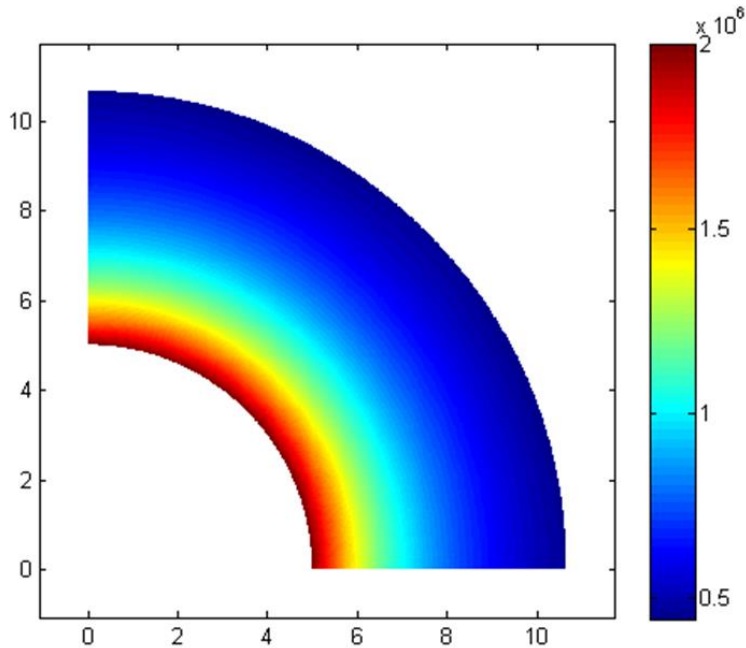


Fig. 5-2 Distribución en tensiones de la solución exacta

Se comprueba que la tensión máxima es $2 \cdot 10^6$ exactamente y la tensión se distribuye de manera uniforme por toda la geometría.

5.1.1. 1 variable de diseño

En este caso se considera una sola variable de diseño que relacionará los cuatro parámetros geométricos. En la práctica esto quiere decir que se limita su cambio a un arco de circunferencia. Para el análisis de sensibilidades se tienen en cuenta los cuatro parámetros para el cálculo. Con el programa cgFEM se puede combinar más de un parámetro a una variable de diseño

TRABAJO FIN MÁSTER

Los parámetros 1 y 4 tienen el mismo valor de la variable y los parámetros 2 y 3 la variable multiplicada por $1/\sqrt{2}$.

La geometría inicial tiene las siguientes características:

- Variable= 13
- Volumen=113.40009
- $c=-169214.16$

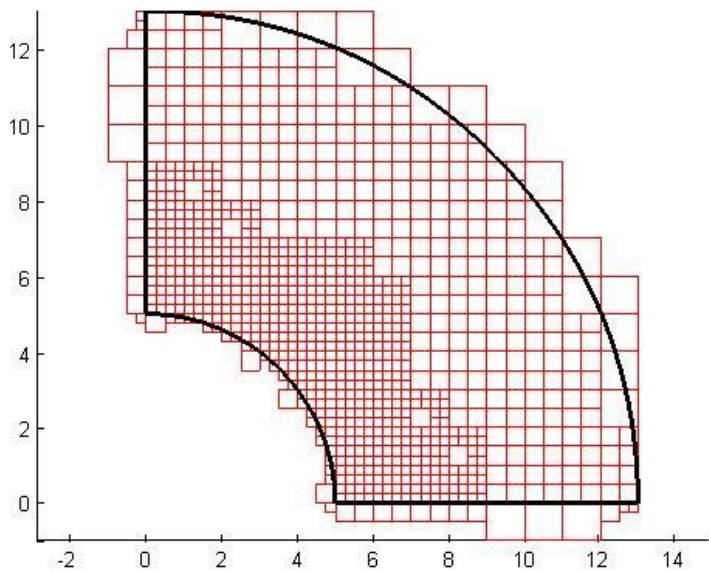


Fig. 5-3 Malla de la geometría inicial caso 1 variable

El proceso de optimización sigue un acercamiento al valor, probando distintas geometrías hasta que se acerca a la óptima y se va acercando paulatinamente en función de las sensibilidades calculadas. La solución obtenida en la optimización se da en la siguiente figura.

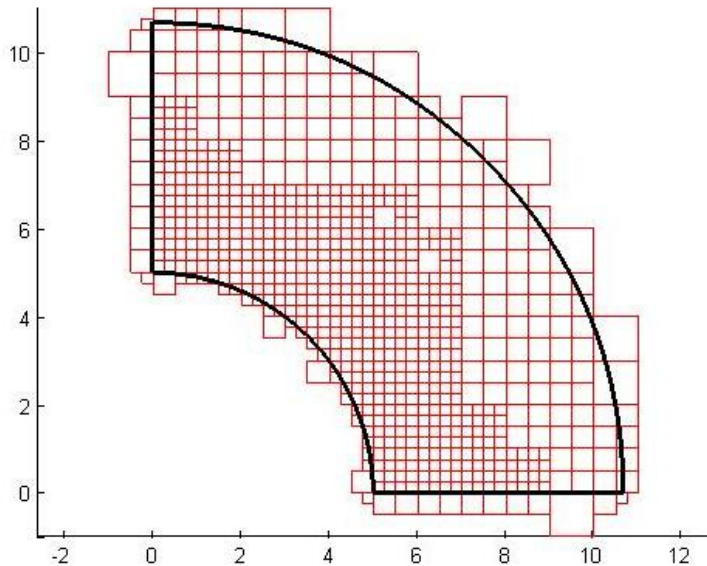


Fig. 5-4 Malla de la solución final caso 1 variable

Los resultados son:

- Variable= 10.665
- Volumen= 69.909043
- $c = -2.4142682$

La solución se ha obtenido en la iteración 77 en un tiempo de 2 horas y 21 minutos. Aunque el proceso sigue calculando probando distintos puntos como se ha explicado en la implementación del algoritmo de optimización.

Se puede comprobar la evolución de la variable por cada iteración en la siguiente figura.

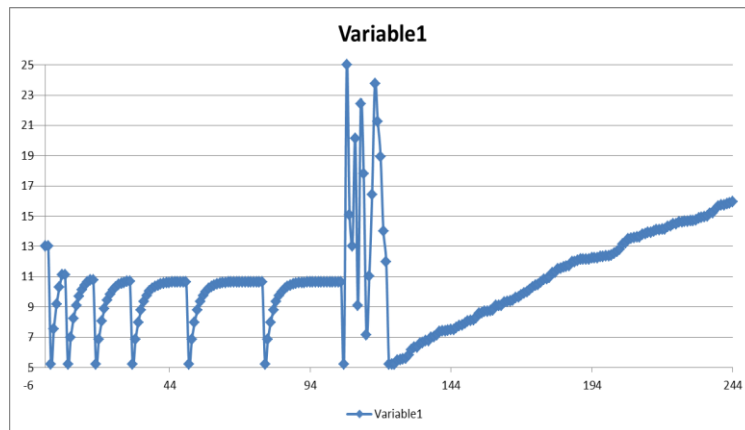


Fig. 5-5 Evolución de la variable

Se comprueba la evolución de la variable. Sigue el algoritmo que prueba desde el mínimo permitido hasta el valor óptimo. Repite el proceso pero cada vez que vuelve al mínimo converge con mayor precisión. Finalmente ha encontrado el óptimo en la iteración 77 pero sigue probando distintos puntos de inicio de la optimización. El proceso finaliza cuando el optimizador se percata de que la solución está divergiendo y considera que el óptimo es el obtenido anteriormente. La distribución de tensiones para este caso es:

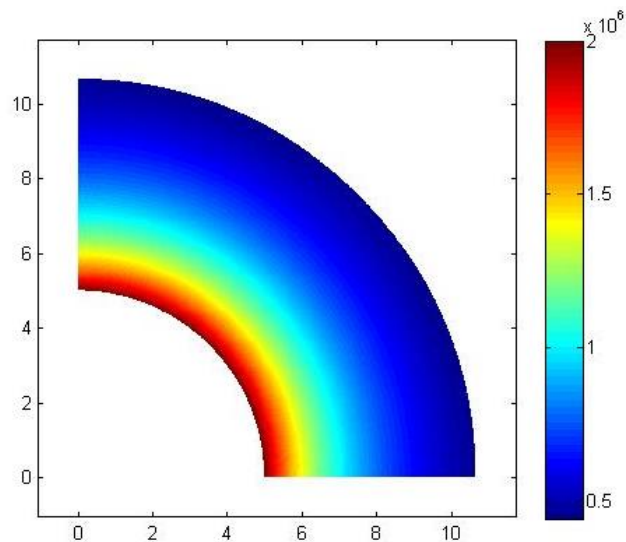


Fig. 5-6 Distribución de tensiones en nodos recuperadas caso 1 variable



El valor máximo de tensión es $2.0000968 \cdot 10^6$. Se comprueba que la tensión máxima es la deseada y está uniformemente repartida en el radio interno, como en la solución exacta.

5.1.2. 2 variables

En este caso se tienen en cuenta 2 variables de diseño. La primera variable de diseño relaciona los parámetros 1 y 4. La segunda variable relaciona los parámetros 2 y 3. Así se asegura que la solución sea simétrica pero sin restricciones tan estrictas como en el caso anterior.

La geometría inicial tiene las siguientes características:

- Variable1= 13 Variable2=12
- Volumen= 150.02307
- $c= -155237.99$

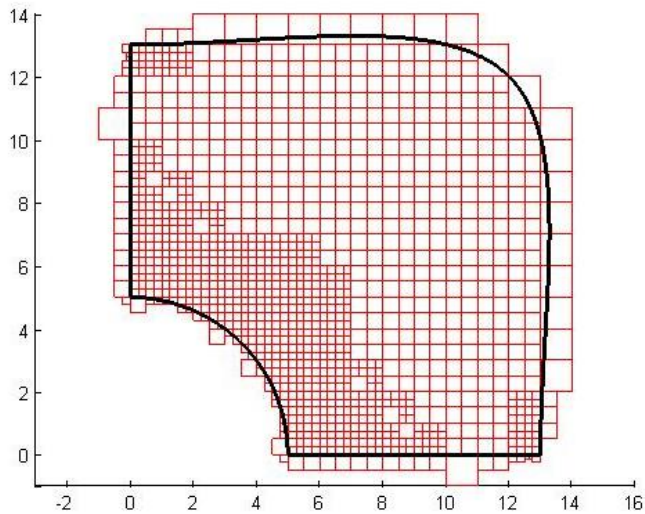


Fig. 5-7 Malla inicial de la geometría caso 2 variables

Durante el proceso iterativo va generando las geometrías hasta que encuentra la óptima.

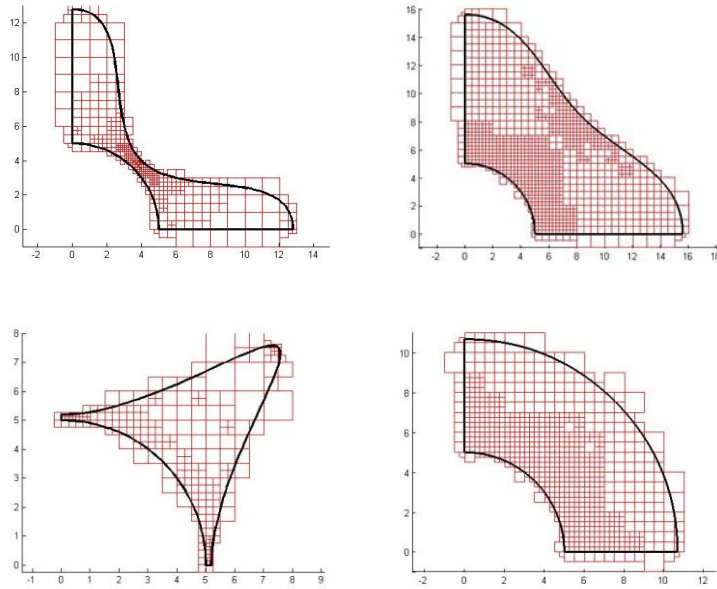


Fig. 5-8 Mallas de las iteraciones 2, 5, 12 y 49 (de arr-ab , izq-der)

Hasta que se obtiene la solución final:

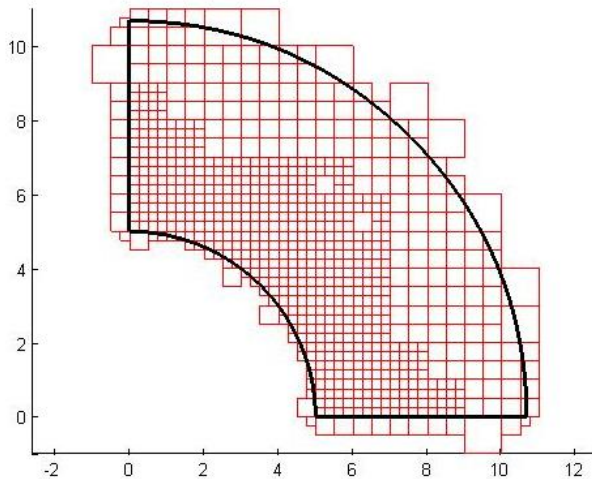


Fig. 5-9 Malla de la solución óptima caso de 2 variables



Los resultados son:

- Variable1= 10.667
- Variable2= 7.5418, que corresponde a un radio de 10.6657
- Volumen= 69.9241
- $c = -0.074487188$

Se comprueba que los valores son muy similares al caso de 1 variables. La solución se ha obtenido en la iteración 68 en un tiempo de 3 horas y 26 minutos. El tiempo es mayor que para una sola variable puesto que al haber más variables se requieren más cálculos, sobretodo en la parte de cálculo de sensibilidades.

Se observa la evolución de las variables en la siguiente figura:

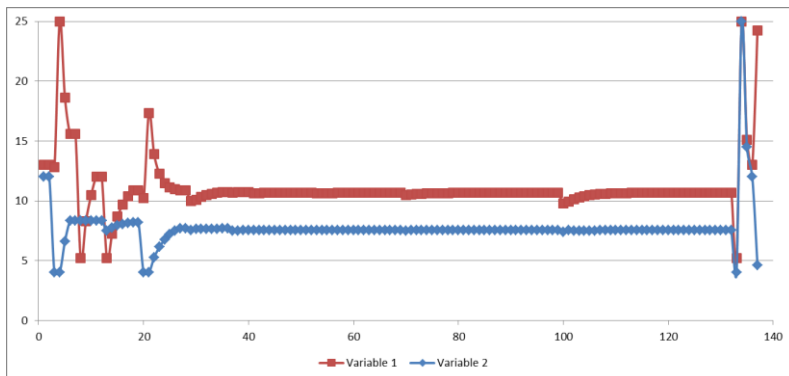


Fig. 5-10 Evolución de las variables caso 2 variables

Se comprueba que en la iteración 68 ya ha convergido aunque continúa el proceso de optimización en un intento de hallar una solución más precisa. Se puede apreciar que la convergencia al principio es más oscilante que en el caso de una variable. Esto es debido a que ahora debe satisfacer las restricciones de ambas variables teniendo en cuenta do.

TRABAJO FIN MÁSTER

Se obtiene la distribución en tensiones siguiente:

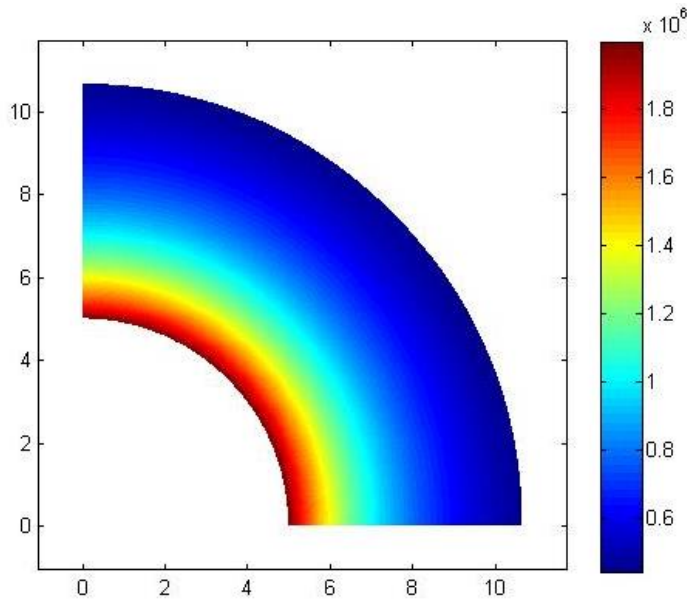


Fig. 5-11 Distribución de tensiones en nodos recuperadas caso 2 variables

El valor máximo de tensión es $1.999996 \cdot 10^6$. Cumple la restricción de tensión máxima y está repartida homogéneamente por el radio interno.

5.1.3. 4 variables

En este caso se tienen en cuenta 4 variables de diseño. Son una variable de diseño por parámetro geométrico. Se da así total libertad al proceso de optimización de hallar la solución óptima. Aunque teóricamente se sabe la solución, tiene simetría de revolución, dado que no se impone ninguna restricción, el optimizador podría proporcionar una solución asimétrica.



La geometría inicial tiene las siguientes características:

- Variables 1 y 4= 20 Variables 2 y 3= 15
- Volumen= 313.15872
- $c= -338811.15$

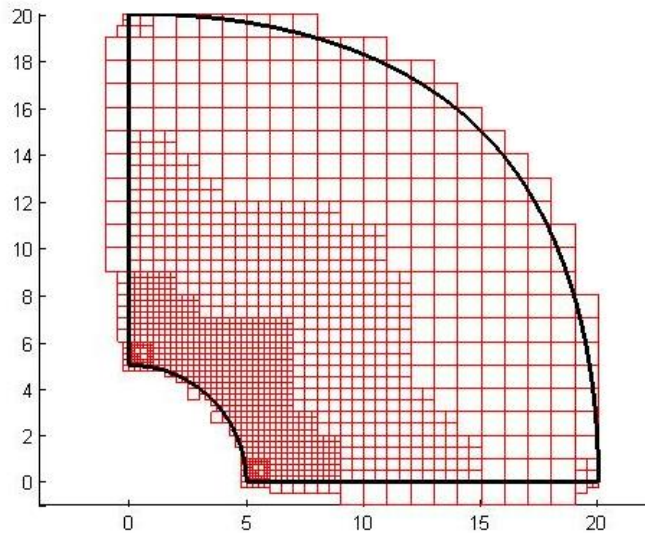


Fig. 5-12 Malla inicial caso 4 variables

El proceso iterativo parece más errático que los anteriores, dado que el algoritmo debe considerar 4 variables por separado. Se aumenta considerablemente el número de variaciones posibles que pueden mejorar la geometría actual. Aunque una vez ha encontrado el óptimo adecuado se procede a la aproximación a la solución, con una mayor cantidad de iteraciones.

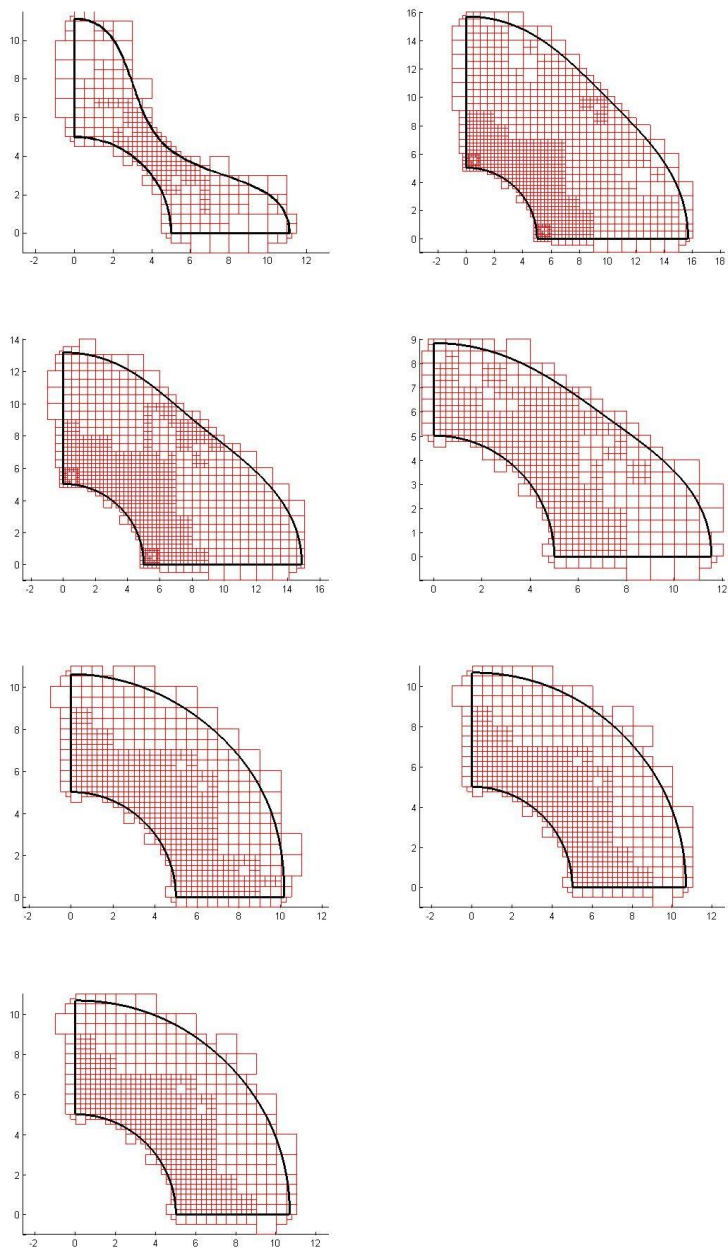


Fig. 5-13 Mallas de las iteraciones 2, 5, 10, 23, 29, 119 y 217 (de arr-ab, a izq-der)



Hasta que se obtiene la solución:

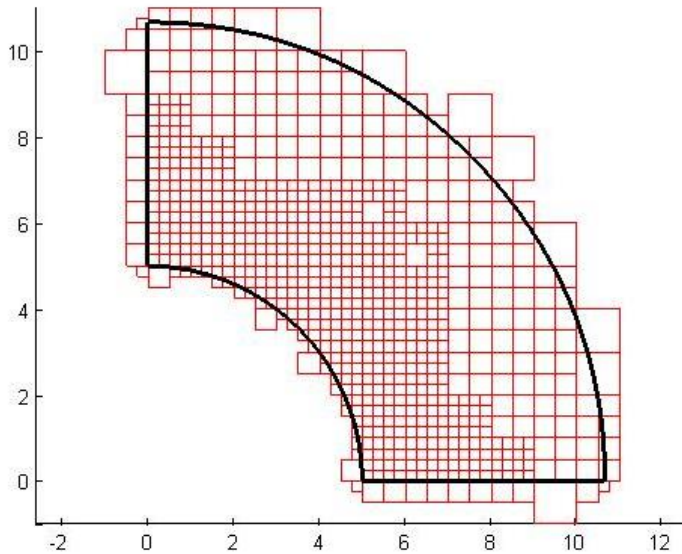


Fig. 5-14 Geometria y malla de la solución caso 4 variables

Los resultados son:

- Variable1= 10.659
- Variable2= 7.5408 y Variable3= 7.5408, que darían lugar en el punto definido por estas dos variables a un radio de 10.6643
- Variable4= 10.659
- Volumen= 69.843905
- $c = -0.0068130633$

Se ha obtenido la solución en la iteración 236 en un tiempo de 20 horas. Es 5 veces más tiempo que con dos variables, y casi 10 veces más que con una. El número de iteraciones no es proporcional al número de variables.

TRABAJO FIN MÁSTER

Se muestra la evolución de las variables en las siguientes figuras:

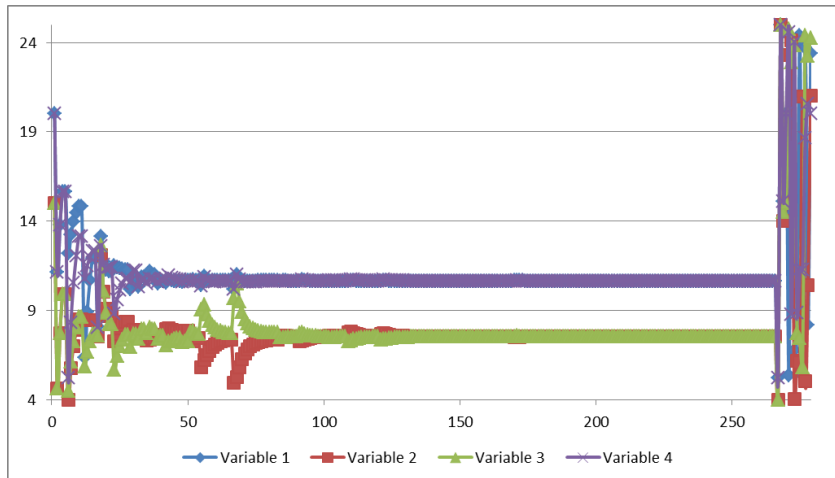


Fig. 5-15 Evolución de las variables para el caso de 4 variables

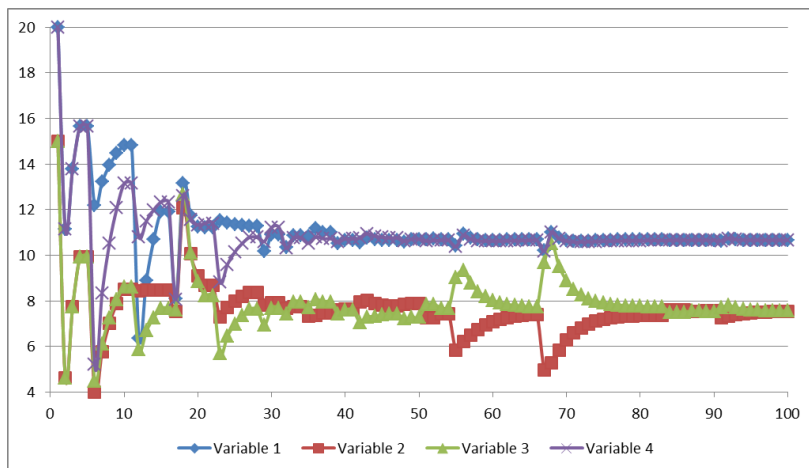


Fig. 5-16 Detalle de la evolución de las variables para el caso 4 variables

Se comprueba que tarda en converger unas 100 iteraciones, como se muestra en el detalle de la evolución. Pero una vez llega a las proximidades del punto óptimo requiere de más iteraciones hasta alcanzar el punto exacto con la mayor precisión posible. Se puede apreciar que el algoritmo tarda en alcanzar el punto óptimo más que en los casos anteriores, lo cual es lógico pues debe ajustar las 4 variables por separado. Este caso es especialmente complejo porque en la



solución analítica, las tensiones máximas se dan a lo largo de todo el radio interno, sin embargo, la solución numérica, al ser aproximada, no podrá representar fielmente este resultado. En la solución numérica, ínfimas modificaciones de las variables de diseño harán que el punto de máxima tensión cambie radicalmente que posición, lo que dificulta enormemente el proceso de convergencia.

La distribución en tensiones es la siguiente:

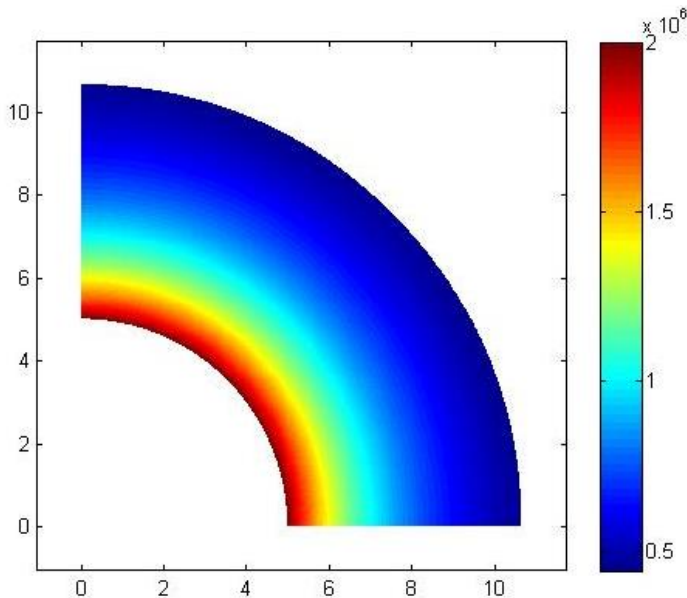


Fig. 5-17 Distribución de tensiones en nodos recuperadas caso 4 variables

El valor máximo de tensión es $2.000006993 \cdot 10^6$. Cumple la restricción de tensión y está repartida homogéneamente por el radio interno.

TRABAJO FIN MÁSTER

5.1.4. Comparativa de soluciones

Una vez obtenidos los resultados para las tres configuraciones del problema se procede a comparar los resultados en la siguiente tabla:

	Solución exacta	1 variable	2 variables	4 variables
Volumen	69.985	69.909	69.924	69.843
Var 1	10.666	10.665	10.667	10.659
Var 2	7.542	7.5413	7.5418	7.5408
Var 3	7.542	7.5413	7.5418	7.5408
Var 4	10.666	10.665	10.667	10.659
c	0	-2.41426	-0.07448	-0.00681
Tiempo	-	2h21'	3h26'	20h

Tabla 1 Comparación resultados tubería simple

La solución que más se acerca a la teórica es la de dos variables. Sin embargo, con 4 variables la c es la que más se ajusta a 0, que es lo ideal. En todo caso los valores obtenidos son bastante precisos, especialmente si se tiene en cuenta que no se puede definir una superficie circular con un spline y que el uso del MEF introduce errores de discretización en los resultados.

Desde el punto de vista de la ingeniería, las soluciones serían correctas ya que tienen todas menos del 1% de error. Para este caso concreto se utilizaría el caso de una variable por ser más rápido. Pero para problemas que se desconozca la solución exacta es recomendable usar el método que no limite la solución.



5.2. Tubería con paredes rectas y arco en superficie interna

Este caso es similar a la tubería del apartado anterior salvo por las paredes rectas de 1 cm de largo con un radio de acuerdo entre las dos rectas de 4 cm. Se van a considerar 2 variables como mínimo ya que en este caso la solución óptima es lógico que no sea un arco de circunferencia. La presión es 900000 Pa igual que en el caso anterior pero se distribuye también en las líneas rectas.

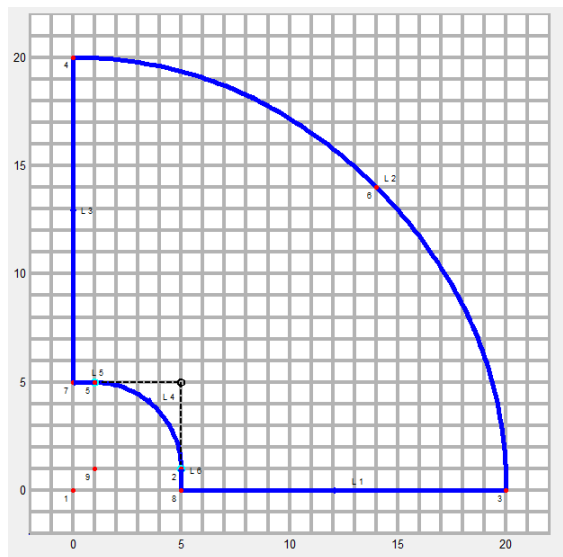


Fig. 5-18 Geometría del cilindro con paredes rectas

La geometría inicial tiene las siguientes características:

- Variable1= 20 Variable2= 14
- Volumen= 290.3176
- $c= -59865.093$

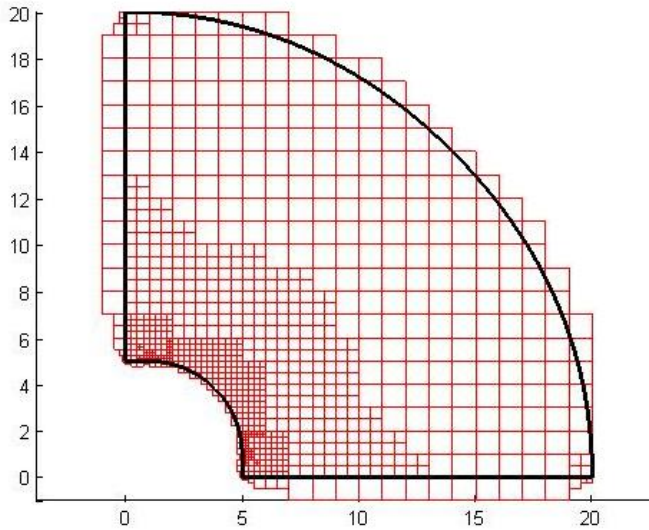


Fig. 5-19 Malla inicial caso 2 variables con paredes rectas

El proceso iterativo es parecido al de 2 variables simple comentado anteriormente. Una muestra de lo que ocurre en las iteraciones que siguen a la solución óptima se ve en la siguiente figura. Va probando distintas geometrías hasta que se da cuenta de que ya ha obtenido la mejor solución.

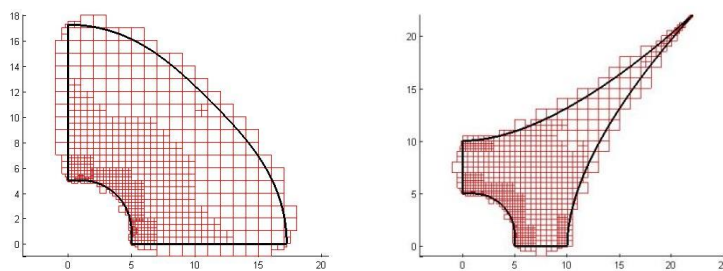


Fig. 5-20 Mallas de las iteraciones 9 y 58



Se obtiene la solución:

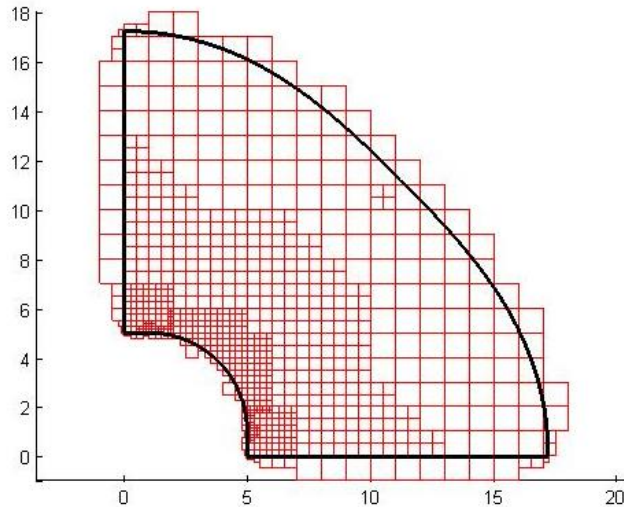


Fig. 5-21 Malla final del caso 2 variables con paredes rectas

Los resultados de la solución:

- Variable1= 17.193
- Variable2= 11.193, que representaría un punto a una distancia del origen de coordenadas de 15.839
- Volumen= 193.56489
- $c = -0.04607826$

Cumple las restricciones de tensión máxima, pero como no se dispone de la solución exacta no hay certeza de la precisión de la solución. La solución se ha obtenido en la iteración 25 en un tiempo de 1 hora y 35 minutos. El tiempo es incluso menor que para el caso simple con una variable, pero puede ser por la geometría inicial. Dependiendo de la geometría inicial escogida se puede estar más o menos cerca de la solución óptima. Requiriendo menos tiempo para el cálculo.

TRABAJO FIN MÁSTER

Se muestra la evolución en la siguiente figura:

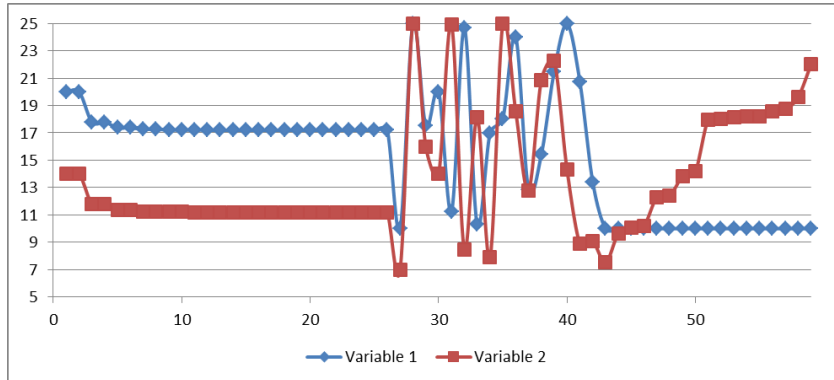


Fig. 5-22 Evolución de las variables caso tubería con paredes rectas

Se comprueba que para la iteración 25 ya ha convergido. En este caso converge más rápido que para la tubería simple con dos variables pese a ser dos problemas similares. Esto puede ser debido que con esta geometría y la elección de la geometría inicial haya facilitado la búsqueda del mínimo.



La distribución en tensiones es la siguiente:

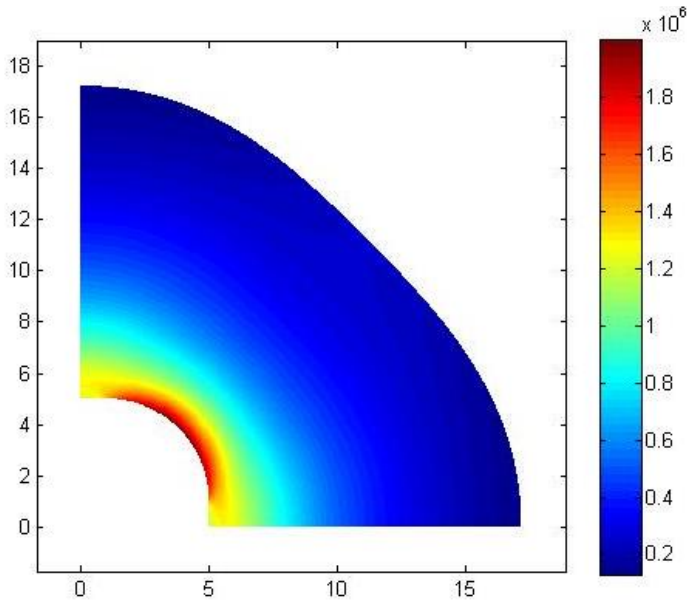


Fig. 5-23 Distribución de tensiones en nodos recuperadas caso 2 variables con paredes rectas

El valor máximo de tensión es $1.9999887 \cdot 10^6$. Cumple la restricción de tensión máxima. Sin embargo, la tensión no está homogéneamente repartida por el radio interior. La lógica nos hace pensar que se puede disminuir un poco el valor de la variable 1 ya que no tiene la máxima tensión en su respectiva zona del radio interno. Pero al realizar los cálculos para una pequeña variación del valor de la variable 1 supone superar la tensión máxima en la zona intermedia. Esto es debido a que una pequeña variación en la variable 1 afecta a la variable 2 por efecto de la spline. Cualquier cambio en la variable 2 es crítico ya que es la zona donde se ha maximizado la tensión. Además hay un efecto de concentrador de tensiones en el punto de intersección de la línea recta y la línea curva de la sección interior. Un cambio ínfimo en esa zona puede suponer un aumento considerable de la tensión en la zona.

TRABAJO FIN MÁSTER

5.3. Tubería con superficie interna elíptica

En este caso el radio interno no es un arco de circunferencia sino una elipse de eje menor 5 cm y eje mayor 7 cm. Se van a considerar pues 4 variables de diseño, puesto que no se puede considerar una posible simetría a priori.

En este caso se ha tomado como tensión máxima admisible $3 \cdot 10^6$ Pa. La presión aplicada es de 900000 Pa en la superficie elíptica.

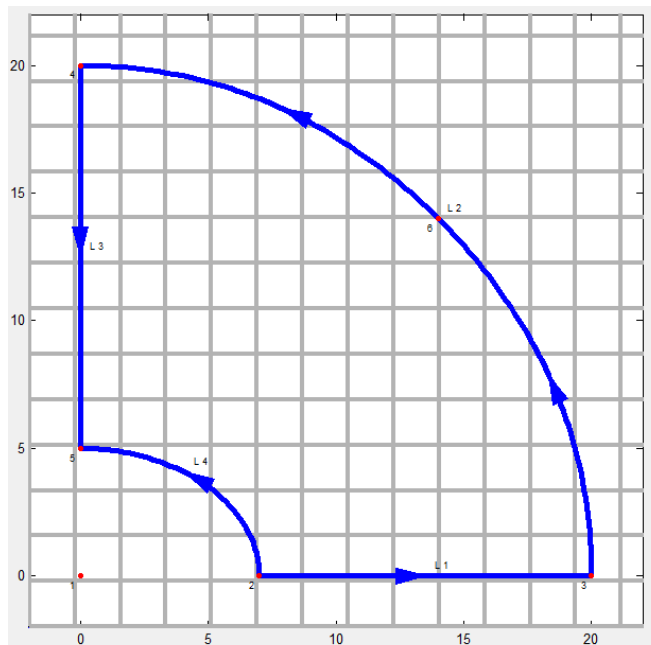


Fig. 5-24 Geometría con superficie interna elíptica

La geometría inicial tiene las siguientes características:

- Variables 1 y 4= 20 Variables 2 y 3= 15
- Volumen= 305.30474
- $c= - 358516.71$

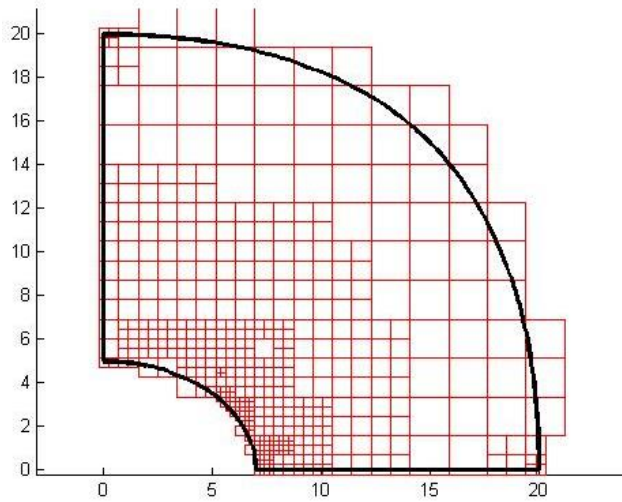


Fig. 5-25 Malla inicial caso elíptica

El proceso iterativo es similar al caso simple con 4 variables. Toma distintas geometrías hasta que obtiene la óptima.

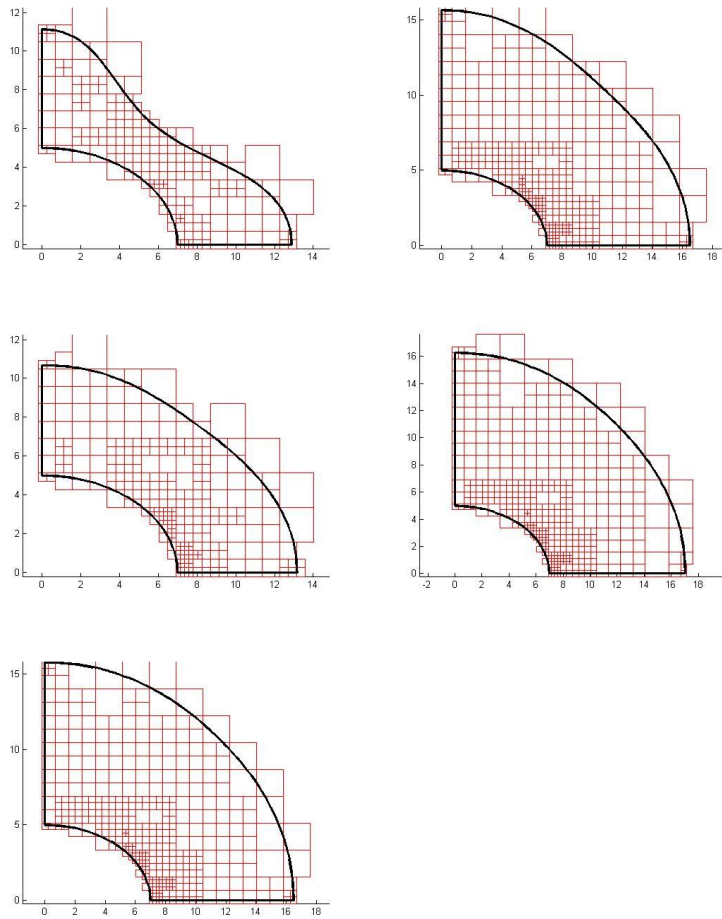


Fig. 5-26 Mallas de las iteraciones 2, 4, 8, 29 y 80(de arr-ab, de izq-der)



La solución se obtiene:

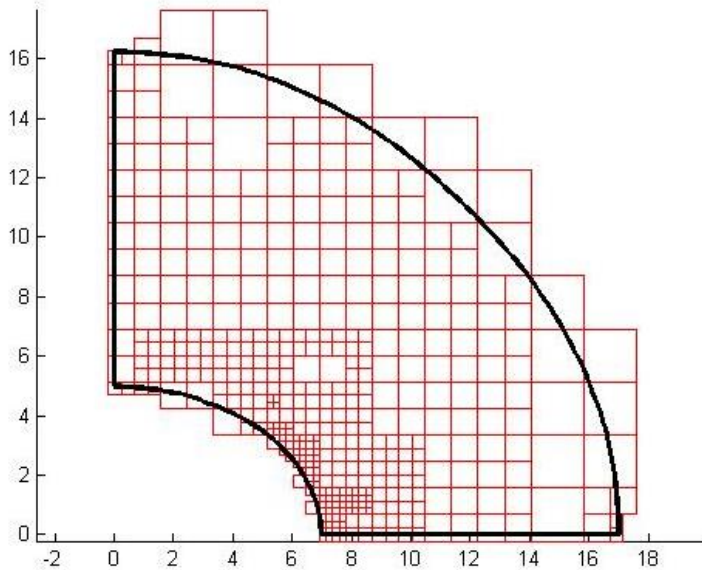


Fig. 5-27 Malla final del caso elíptico

Los resultados son:

- Variable1= 17.02
- Variable2= 11.445
- Variable3= 11.445
- Variable4= 16.25
- Volumen= 184.8168
- $c = -1.7010251$

Cumple la restricción de tensión máxima. Se comprueba que como la lógica hacía suponer, es mayor la variable 1 que la 4, quizá menor de lo esperado.

TRABAJO FIN MÁSTER

La evolución de las variables se muestra en la siguiente figura:

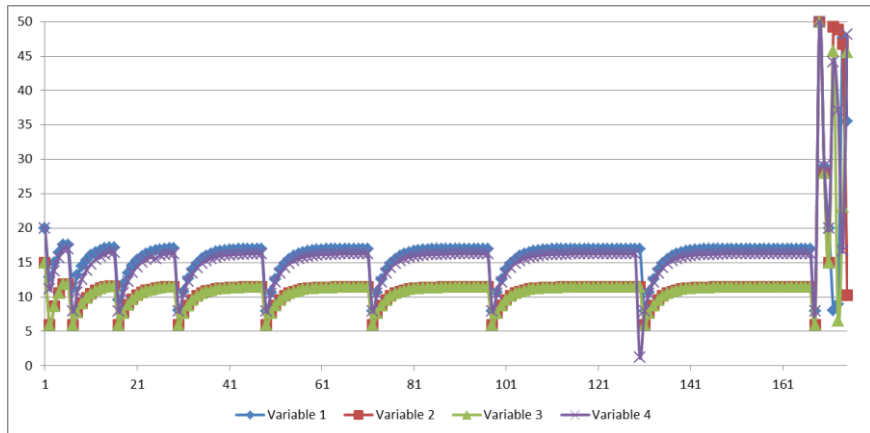


Fig. 5-28 Evolución de las variables caso elipse

El valor de las variables 2 y 3 es exactamente el mismo durante el proceso. Es un hecho curioso ya que en principio la solución es asimétrica. Podría deberse a que se trata de una spline y se ha producido tal efecto. Las variables 1 y 4 siguen una convergencia similar aunque el valor de la variable 1 es siempre mayor al de la variable 4. Otro aspecto a destacar es que la convergencia está bien encaminada desde el principio, no como en los casos anteriores. En estos la convergencia es más errática hasta que encuentra el que parece el camino correcto a la solución óptima.



La distribución de tensiones es:

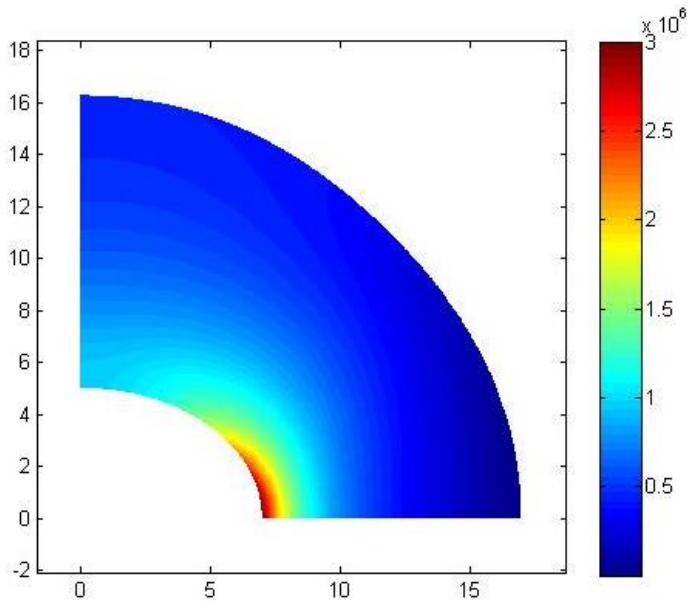


Fig. 5-29 Distribución de tensiones en nodos recuperadas caso elíptico

El valor máximo de tensión es $3.0001481 \cdot 10^6$. Cumple la restricción de tensión máxima. Como en el caso anterior se podría disminuir el valor de la variable 4, pero afecta en el resto de la curva. Además está el concentrador de tensiones que se puede apreciar claramente en la figura.

5.4. Presa

Como ejemplo de aplicación se ha elegido un problema con una geometría algo más compleja que la de los ejemplos anteriores. El problema en cuestión trata de simular la geometría de una presa de gravedad, siendo las condiciones de contorno de la parte interior del dique las presiones hidrostáticas del agua. Se ha tomado un valor del módulo de elasticidad de $13.1 \cdot 10^{10}$ Pa y el módulo de Poisson de 0.25 y una densidad de material de 2500 Kg/m^3 para la carga gravitatoria. La tensión máxima admitida es $27.5 \cdot 10^6$ Pa. El problema se muestra en la siguiente figura:

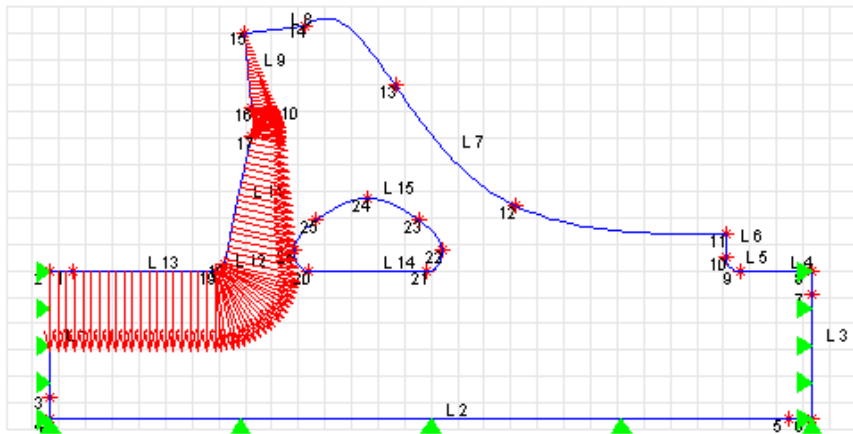


Fig. 5-30 Geometría de la presa

Las condiciones de Neumann impuestas están representadas como flechas de color rojo y simulan la presión hidrostática del agua, aplicada sobre las líneas de contorno L9, L10, L11, L12 y L13. Por otro lado las condiciones de contorno de Dirichlet impuestas son de una restricción de los desplazamientos horizontales en las líneas L1 y L3 y una restricción de desplazamientos verticales en la línea L2.

Hay que indicar que el contorno interno, definido por los puntos 20, 21, 22, 23, 24, y 25 simula el hueco interno de la presa cuya geometría se desea optimizar manteniendo el contorno externo de la misma. Las variables de diseño se muestran en la siguiente figura:

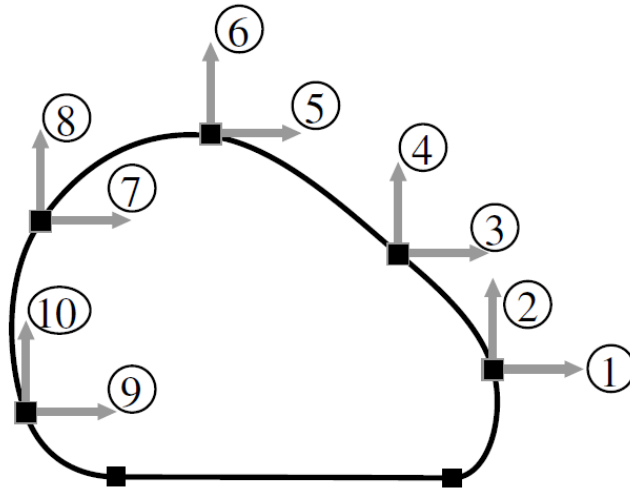


Fig. 5-31 Variables de diseño del caso de la presa

Se han impuesto unas restricciones geométricas a las variables de diseño para evitar que la solución se salga del interior de la presa.

Variable	X_0	Límite inferior	Límite superior
1	50	50	60
2	3	3	6
3	47	47	50
4	10	10	17
5	38	35	40
6	15	15	28
7	33	27	33
8	10	10	26
9	27	23.5	27
10	3	3	6

Tabla 2 Valores iniciales y límites de las variables

TRABAJO FIN MÁSTER

También hay unas restricciones entre variables:

$$V_3 - V_1 < -2$$

$$V_2 - V_4 < -3$$

$$V_5 - V_3 < -3$$

$$V_6 - V_4 < -2$$

$$V_7 - V_5 < -3$$

$$V_8 - V_6 < -2$$

$$V_9 - V_7 < -2$$

$$V_{10} - V_8 < -3$$

Estas restricciones son para hacer que no haya curvas que se crucen o muy anguladas.

La geometría inicial es la siguiente:

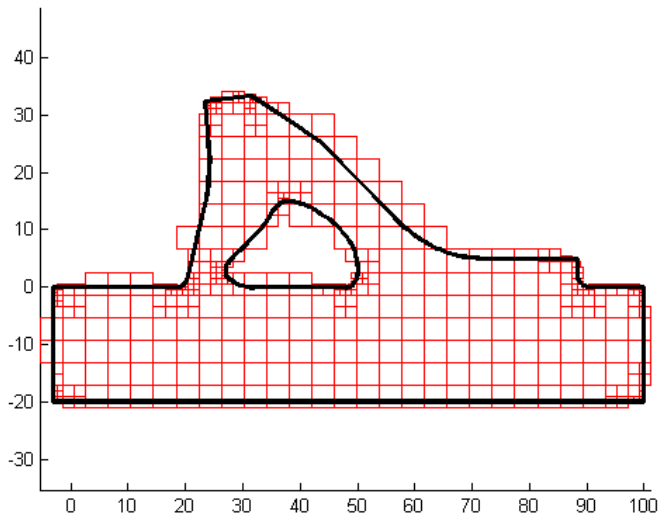


Fig. 5-32 Geometría y malla inicial caso presa



Con los valores:

- Volumen= 2895.7314
- $c = -24179736$

El proceso iterativo es el siguiente:

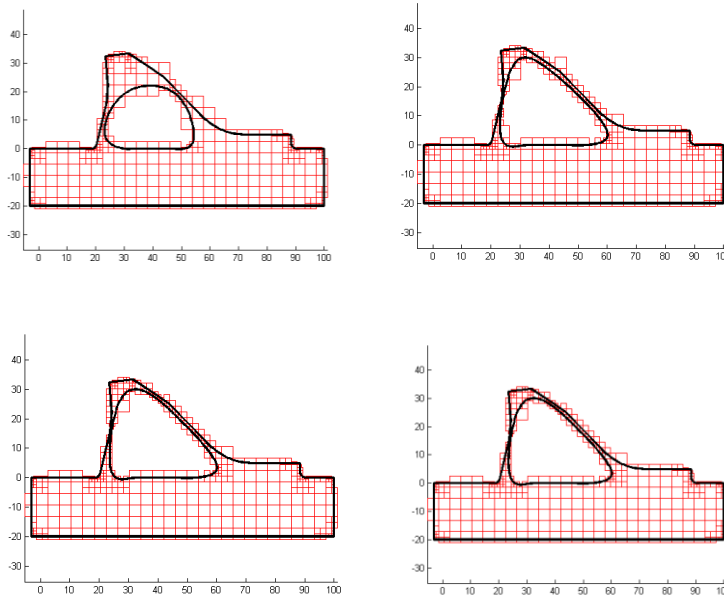


Fig. 5-33 Malla de las geometrías 2, 4, 38 y 65 (de arr-ab, de izq-der)

Como el óptimo está cerca de los valores máximos de las variables encuentra rápidamente la convergencia.

TRABAJO FIN MÁSTER

La solución obtenida es la siguiente:

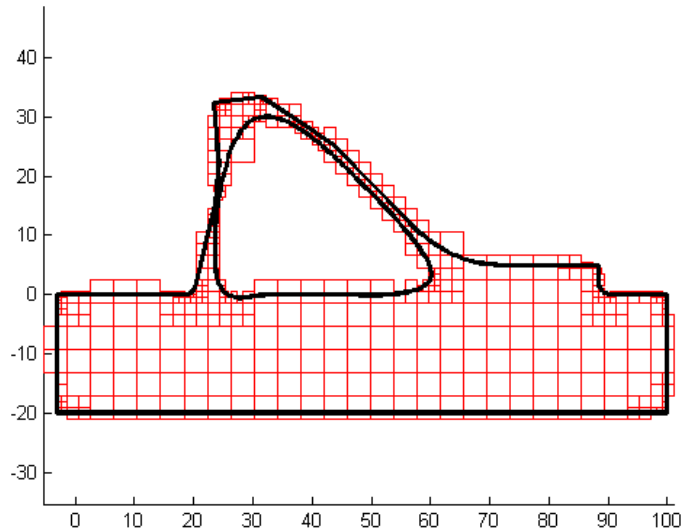


Fig. 5-34 Malla de la solución caso presa

Los resultados son:

- Volumen= 2370.7216
- $c = -4593957.5$

La solución se obtiene en la iteración 80 con un tiempo de 6 horas y 25 minutos. En este caso no es comparable a los casos anteriores puesto que solo se ha hecho refinamiento geométrico, en lugar de refinamiento adaptativo como en los casos anteriores para facilitar el tiempo de cálculo. Aun así, el tiempo de cálculo es mayor que en los otros casos ya que tiene 10 variables de diseño.



Después de hallar la solución sigue probando los puntos por si encontrara una mejor solución. Las geometrías consideradas son las siguientes:

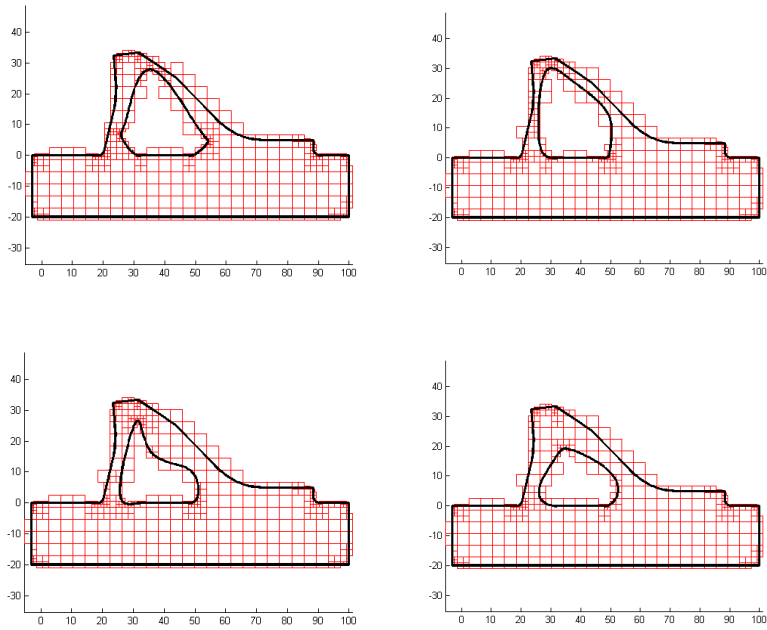


Fig. 5-35 Geometrías de las iteraciones 112, 145, 242 y 294 (de arr-ab, de izq-der)

TRABAJO FIN MÁSTER

La evolución de las variables es la siguiente:

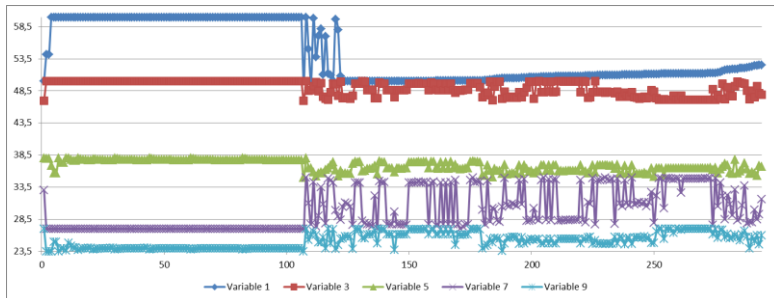


Fig. 5-36 Evolución de las variables 1, 3, 5, 7 y 9

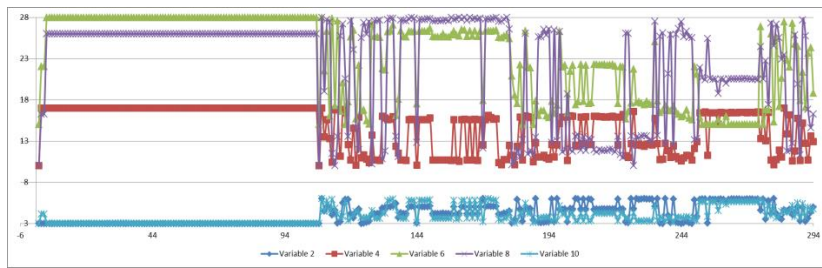


Fig. 5-37 Evolución de las variables 2, 4, 6, 8 y 10

Se puede comprobar que la convergencia al valor óptimo es rápida aunque le cuesta más iteraciones para alcanzar la precisión adecuada. También se ve que después va probando distintas geometrías sin éxito, con lo que finalmente la solución óptima es la que se ha especificado.



La distribución en tensiones de la solución es la siguiente:

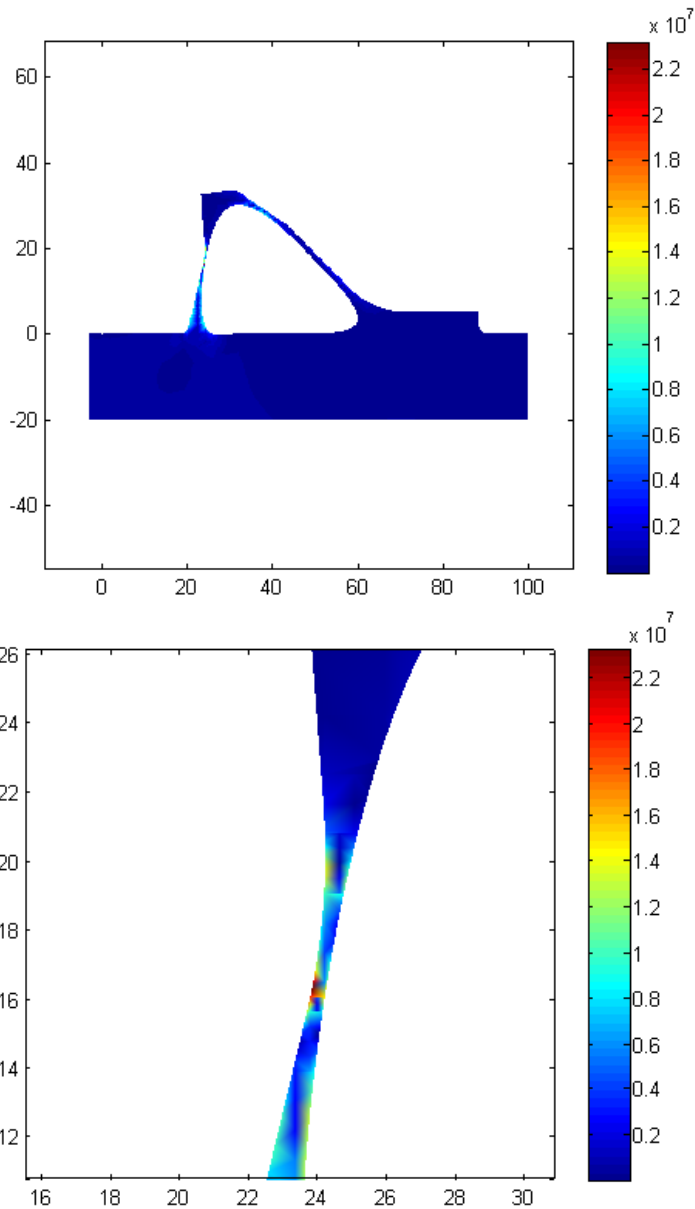


Fig. 5-38 Distribución en tensiones caso presa y detalle zona con mayor tensión



TRABAJO FIN MÁSTER

La zona de mayor tensión se encuentra en la parte izquierda. Según la geometría que se genere puede causar concentración de tensiones. La tensión máxima es $2.31949 \cdot 10^7$. Cercana a la máxima pero no en el límite. Sin embargo, al disminuir un poco la geometría se sobrepasa la tensión máxima admitida en hasta 5 MPa debido al concentrador de tensiones.

Aunque esta solución es inviable puesto que el problema fallaría a pandeo. Pero se está estudiando el proceso de optimización y no se ha incluido el caso de pandeo para facilitar los cálculos.

Se observa también que tiene tendencia a acercarse lo máximo posible a las paredes externas, en la medida de lo posible, ya que además de las restricciones geométricas tiene la limitación de la spline.

6. Conclusiones

El objetivo de minimizar el tiempo de cálculo se ha cumplido parcialmente. Se ha demostrado que el uso de un algoritmo basado en gradiente no supone una mejora considerable. Es posible obtener una solución más rápido que con un algoritmo evolutivo pero para asegurar que el mínimo calculado sea el mínimo global de la función objetivo hace falta unos cálculos adicionales que ralentizan el proceso de optimización.

Cuanto más variables se usen para definir el problema mayor es el tiempo requerido, pero más fiable podrá ser la solución porque el algoritmo tendrá un campo de soluciones más amplio donde buscar el óptimo como se ha visto en el ejemplo de la tubería simple. En este ejemplo se obtiene la misma solución con 1, 2 o 4 variables. Un problema podría haber sido que cuando se tienen 4 variables no llegara a la solución correcta porque no se ha limitado su tipo de solución. Este método podría ser aplicable a problemas de los que se desconozca la solución.

La solución obtenida depende en gran medida de la calidad de la malla y de las derivadas, o sensibilidades. Es por ello que en los ejemplos numéricos se han tomado unos valores mínimos de calidad de la malla para que estas no fueran influyentes en la solución.

La proyección del error no se ha podido implementar en el proceso de optimización, pero este podría suponer una reducción considerable del tiempo de cálculo. Es un proceso simple que no supone demasiado tiempo añadido de cálculo pero que reduciría el cálculo de la nueva malla.

7. Trabajos futuros

Por falta de tiempo no se ha podido implementar la proyección del error en el proceso de optimización. Un trabajo futuro claro es llevar a cabo esa implementación para comprobar si realmente supone una mejora en el cálculo.

Se tendría que comprobar cómo se comporta el proceso usando otras restricciones además de las tensiones, como por ejemplo los desplazamientos.

Se debería probar el proceso en otras geometrías para comprobar su robustez. También se podrían utilizar otros algoritmos y comparar cuál de ellos es el más adecuado.

Dado que en el departamento se está desarrollando la versión 3D del código cgFEM, una línea de trabajo futuro clara es la de implementar los desarrollos de este trabajo en dicho código 3D

Bibliografía

- [1] Onofre Marco. *A shape sensitivity analysis module with geometric representation by nurbs for a 2d finite element program based on cartesian meshes independent of the geometry*. MTh, Universitat Politècnica de València, Julio 2012.
- [2] Gabriel Bugeda, Juan José Ródenas, Eugenio Oñate. *An integration of a low cost adaptative remeshing strategy in the solution of structural shape optimization problems using evolutionary methods*. CAS 4108. Computers and Structures 2007.
- [3] Juan José Ródenas, Error de discretización en el cálculo de sensibilidades mediante el método de los elementos finitos. Tesis Doctoral, Universitat Politècnica de València, 2001.
- [4] Enrique Nadal, *Desarrollo de un programa de refinamiento h-adaptable para el análisis mediante GFEM de problemas elásticos con mallados independientes de la geometría*, Tesis de Máster, Universitat Politècnica de València, Diciembre 2009.
- [5] Santos López, *Optimización de componentes mecánicos mediante algoritmo genético y software comercial de elementos finitos*, Tesis de Máster, Universitat Politècnica de València, Diciembre 2008.
- [6] Fletcher, R., *An Overview of Unconstrained Optimization*, Numerical Analysis Report NA/149, Junio 1993.
- [7] Nadal, E.¹; Bordas, S.²; Ródenas, J.J.¹; Tarancón, J.E.¹ y Tur, M.¹, *Accurate Stress Recovery for the Two-Dimensional Fixed Grid Finite Element Method*, 1.Universitat Politècnica de València y 2.Cardiff University, 2010.
- [8] Elke Pahl, *Optimization: An attempt at describing the State of the Art*, International Center for Numerical Methods in Engineering (CIMNE), Barcelona, Marzo 2004.
- [9] Strouboulis, T.; Copps, K. y Babuška, I., *The generalized Finite Element Method: an Example of its Implementation and Illustration of its Performance*. International Journal for Numerical Methods in Engineering. Vol. 47, pp. 1401-1417; 2000.

Anexo

Manual del programador

1. Variables

La principal variable creada para todas las funciones creadas durante la realización de este trabajo de fin de máster es la variable global **Reference**. El resto de variables usadas son las variables ya creadas en el programa de cgFEM.

1.1. Reference

Es una variable global utilizada principalmente para guardar los valores de referencia. Su estructura es la siguiente:

- **On**: sirve para decir si la referencia está activa. 1 si esta activa, 0 inactiva.
- **SensAnaGlob**: estructura que contiene los resultados del problema de sensibilidades global.
- **dVelm**: estructura que contiene los valores de la derivada de volumen por elemento.
- **dV**: estructura que contiene los valores de la derivada de volumen de la geometría.
- **StressVM**: estructura que contiene los valores de la tensión de Von Mises.
- **dStressVM**: estructura que contiene los valores de la derivada de la tensión de Von Mises.
- **dStressOfMaxStressVM**: vector que contiene la derivada de la tensión de Von Mises en el punto de mayor tensión.
- **MaxStressVM**: valor máximo de la tensión máxima.
- **U**: estructura que contiene los valores del desplazamiento.
- **dU**: estructura que contiene los valores de la derivada del desplazamiento.

DOCUMENTO III

- **Elm**: estructura que contiene datos sobre los elementos.
- **CalcMesh**: estructura que contiene datos sobre la malla.
- **MatK**: estructura que contiene datos sobre la matriz **K**.
- **Nod**: estructura que contiene datos sobre los nodos.
- **SensAnaIter**: estructura que contiene los resultados del problema de sensibilidades por iteracion.
- **Iter**: estructura que contiene los resultados del problema de Elemento Finitos.
- **Contador**: variable que lleva la cuenta de las iteraciones realizadas en el proceso de optimización.

2. Funciones

En este apartado se presentan las funciones generadas para la realización del trabajo.

2.1. *Proceso de optimización*

En este apartado se presentan las funciones implicadas en el proceso de optimización. Las funciones proporcionadas son para el caso de la tubería simple de 4 variables. Se tienen otras homologas para el resto de casos.

2.1.1. **OptimizacionGlobal**

En primer lugar se tiene la función **OptimizacionGlobal**. Esta función llama al optimizador y al calculador del algoritmo de optimización. Aquí se definen las condiciones de la geometría inicial y las opciones del algoritmo de optimización. Al final se guardan los resultados en un fichero.

```
function OptimizacionGlobal
global Reference
x0    = [20.0 15.0 15.0 20.0];
Aineq = [-1.0 0.0 1.0  0.0;
         0.0 1.0 0.0 -1.0];
Bineq = [-0.5;
        -0.5];
lb     = [5.20 4.0 4.0 5.2];
ub     = [25.0 25.0 25.0 25.0];
Reference.Contador=0;
Reference.ContadorRest=1;
gs     =...
GlobalSearch('NumStageOnePoints',200,'NumTrialPoints',...
1000,'Display','iter','StartPointsToRun','bounds-ineqs');
opts   =...
optimset('Algorithm','sqp',...
'Display','off','GradConstr','on','GradObj','on');
```

DOCUMENTO III

```
problem =...
    createOptimProblem('fmincon','options',opts, ...
        'objective',@ObjectiveFunctionCylinder_4var,...
        'Aineq',Aineq,'bineq',Bineq,'x0',x0,'lb',lb,'ub',ub,...
        'nonlcon',@NonLinearRestrictionFunctionCylinder_4var);

[xming,fming,flagg,outptg,manyminsg] = run(gs,problem);

FileName_output = 'Solution.txt';
fid = fopen(FileName_output,'a');
fprintf(fid,'fming= %.4f\r\n',fming);
dlmwrite(FileName_output,xming,'-append');
fclose(fid);
```

2.1.2. ObjectiveFunctionCylinder_4var

El calculador del algoritmo de optimización `fmincon` llama a dos funciones que calculan la función objetivo y las restricciones. El calculador requiere de estos datos por separado. A continuación se presenta la función de `ObjectiveFunctionCylinder_4var`. La de restricción es en este trabajo es la misma en su mayoría, se explicarán las diferencias más adelante.

En esta función se definen los parámetros necesarios en el calculador del `cgFEM`. Una vez hecho el cálculo del FEM se calculan los valores demandados por el optimizador, en este caso el volumen y su derivada. Al final de la función se guardan los valores de la función por cada iteración.

```
function [Volume,GradientVolume]=ObjectiveFunctionCylinder_4var(x0)

global Geo Param FEMGui Nod Elm Errores BatchInfo...
    Iter iMesh CalcMesh GlobalMatKData GlobalIntersData...
    SensAnaGlob SensAnaIter MatK Reference

% Design variables initialization
SensAnaIter=[];
SensAnaGlob=[];
SensAnaGlob.Param.NDesVar=0;
SensAnaGlob.Param.DesVarFile=0;
SensAnaGlob.Param.AnaliticVF=0;
SensAnaGlob.Param.VFEmptyArea=100;
SensAnaGlob.Param.LoadCaseVF=2;
SensAnaGlob.Param.LoadCaseSens=1;
SensAnaGlob.Param.InterpType=zeros(1,3);
SensAnaGlob.Param.DomainType=zeros(1,3);
```

```

%=====
% Next part of the program must be modified for each example
%=====
%-----
% Mode Parameter
%-----
Param.ModeBatch=1; % Param.ModeBatch=1 Allows to run the program
without GUI
%-----
% Geometry
%-----

%Information from Optimization Program. This instruction mustn't be
modified
%-----

Ini
addpath .\samples\10bm_Cylinder\
% Load problem
load Geometria_NurbsOpt_4var
Geo.Cv(1).FixedCv=0;
Geo.Cv(2).FixedCv=0;
Geo.Cv(3).FixedCv=0;
SensAnaGlob.Param.InterpType=[0 0 1];
SensAnaGlob.Param.DesVarFile=...
    '\samples\10bm_Cylinder\sa_SetDesVar_NurbsOpt_4var.m';
Constants
FirstTime=1;
Reference.On=0;
% Iterations
Param.NumIt=0;
Param.It=12;
% Modified Geometry Points
Reference.Parameters=x0;
x0
x0
Geo.Pt.XYZ(2,4)=Reference.Parameters(4);
Geo.Pt.XYZ(1,6)=Reference.Parameters(2);
Geo.Pt.XYZ(2,6)=Reference.Parameters(3);
Geo.Pt.XYZ(1,3)=Reference.Parameters(1);
run(SensAnaGlob.Param.DesVarFile)
ReCalcAll
Geo.Material.MaxStress=2e6;
% Mesh
BatchInfo.HadaptiveMethod=1;
BatchInfo.AnalysisType='h_adaptive_mesh'; %h_adaptive_mesh; TopOpt
BatchInfo.EType=2;
BatchInfo.IniMesh=5; %malla fina=7 malla vasta=3
BatchInfo.IntsMesh=5;
BatchInfo.MaxLevelMesh=11; %malla fina=13 malla vasta=8
BatchInfo.SquareMesh=1;
BatchInfo.UniformRefinement=0;
BatchInfo.RecovType=300;
BatchInfo.TargErr=[-1 -1 -1 -1 0.001 -1]; %0.0005; %Objective
Relative error in Percentage
BatchInfo.ErrReduct=50;%30 %Max allowed error reduction (ex: 25=>
reduce to 75% of current error)
BatchInfo.MxRefInc=3; %Maximum refinement increment
BatchInfo.MxRefLevel=11; %Maximum refinement level

```

DOCUMENTO III

```
BatchInfo.MxNIter=3;%5 %Maximum number of iterations
BatchInfo.GeometricalRefinementArea=0;
BatchInfo.GeometricalRefinementCurvature=1;
BatchInfo.ExactPressSolution=0;
BatchInfo.AnalysisMode2D=PlaneStrain; %PlaneStrain=0;
PlaneStress=1;
BatchInfo.LoadCase2Solve=[1 2 3 4 5 6]; %Load cases that I need to
solve.
BatchInfo.Error_Control=0; % Mode 0=> FEM error solution
% Mode 1=> sqrt(|-2·int(s·e)|/2)
% Mode 2=> sqrt(abs(2·int(s·e))/2)
% Mode 3=> sqrt(2·int(|es|)/2)
BatchInfo.ParTrans=Quadratic; % Boundary interpolation
BatchInfo.TerrorTerm=0;

% Contour Parameter
Param.Contour=1; % Allows to calculate only at elements belonging
to contour
% Path and File
Geo.Path='.\samples\10bm_Cylinder';
Geo.File='.\samples\10bm_Cylinder\Geometria_NurbsOpt_4var.mat';
% GFEM PROGRAM
Constants; % Here are included the codes corresponding to each
magnitude
ReCalcAll % This function creates the new geometry
BatchInfo.MeshCoords=[-1.99 30;-1.99 30];% Same size.
if BatchInfo.SquareMesh
    dummymin=min(BatchInfo.MeshCoords(:,1));
    dummymax=max(BatchInfo.MeshCoords(:,2));
    BatchInfo.MeshCoords=[dummymin dummymax;dummymin dummymax];
end
Source=2; % The program will be executed without Gui AND FROM
OPTIMIZER

Geo.Material.ElastModul=10100000;

MainGE(Source)

sa_deltaV(iMesh,0);
Volume = Iter(iMesh).Volume.Vtotal
GradientVolume = Reference.dV;

sa_StressVM(iMesh,0);

Reference.MaxStressVM;
Reference.dStressOfMaxStressVM;

close all

DMesh(-1,1,0,0);
DGeo(0,0,0);
pause(3)
saveas(gcf,
strcat('Iteracion',num2str(Reference.Contador),'.fig'));

FileName_output = 'Iteraciones.txt';
fid = fopen(FileName_output,'a');
fprintf(fid, 'Iteracion= %d\r\n Volumen= %15.8G\r\n C=
%15.8G\r\n',Reference.Contador,Volume, c);
```

```

dlmwrite(FileName_output,x0,'-append');
fprintf(fid, '\r\n');
fclose(fid);

Reference.Contador=Reference.Contador+1;
Reference.Contador

% Check if the shape is valid in stresses
FEMGui.PostOK=1;

```

La principal diferencia para la función de restricción es la llamada y el cálculo posterior al cálculo de FEM.

```

function [c,ceq,gradc,gradceq]=
NonLinearRestrictionFunctionCylinder_4var(x0)
.
.
.
sa_StressVM(iMesh,0);

Reference.MaxStressVM;
Reference.dStressOfMaxStressVM;

c      = Reference.MaxStressVM-2000000
ceq    = [];
gradc  = Reference.dStressOfMaxStressVM';
gradceq = [];
.
.
.

```

DOCUMENTO III

2.2. Proyección del error

La función de proyección del error es semejante a la función objetivo mostrada anteriormente. La diferencia radica en que se generan las geometrías necesarias en esa misma función, ya que este proceso no ha sido implementado en el proceso de optimización. Para ello se especifica en cada iteración cuales son los parámetros de la geometría. Después de llamar al calculador de cgFEM se llama a la función `ProyeccionError` que es la que realiza el proceso de proyección. Se explica a continuación dicha función.

2.2.1. ProyeccionError

Esta función en primer lugar obtiene la deformada de la malla de referencia. Calcula la densidad del error del elemento. Después determina cada punto de Gauss que error le pertenece y se lo asigna.

```
function [GPV]=ProyeccionError(iMesh,iDesVar)

%función para proyectar el error de la matriz deformada sobre la
malla cartesiana original

global Elm CalcMesh Nod SensAnaGlob SensAnaIter Iter Geo MatK
Reference

ActEls=Reference.CalcMesh(iMesh).ActiveElements;
LCMatrix=Geo.Loads.Cases.Matrix;
FEMLoadCase=LCMatrix(:,9)==1;

if iDesVar==0
    nDesVar=1:Reference.SensAnaGlob.Param.NDesVar;
else
    nDesVar=iDesVar;
end
NNodes=Reference.Elm.NNodes;
% inicializar así o con bucle sacando número de triángulos
nTriTot=0;
for iElm=ActEls
    if Reference.Elm.KNum(iElm)==1
        nTriTot=nTriTot+2;
    else
        numKElm=Reference.Elm.KNum(iElm);
        nContTriElm=size(Reference.MatK(numKElm).PsiEtaVert,3);
        nTriTot=nTriTot+nContTriElm;
    end
end
```



```

end

Triangulos=zeros(3,2,nTriTot);

% inicializar AllGaussPts
XGaussPtsGlo=[ ];
YGaussPtsGlo=[ ];
% VelFieldTotal=zeros(2,4);
ErrElm=zeros(length(Reference.Elm.Top),1);
AreaElm=zeros(length(Reference.Elm.Top),1);
AreaElmdef=zeros(length(Reference.Elm.Top),1);
TriElm=[];
IndiceElementos=1;
IndiceTriangulos=0;
% figure
for iElm=ActEls
    ErrorValue=...
    Reference.Iter(iMesh).ErrorAnalysis(FEMLoadCase).ErrNorm(300,1)...
        (Reference.CalcMesh(iMesh).ElementIndexation(iElm));
    if Reference.Elm.KNum(iElm)==1

        VelFieldTotal=zeros(2,4);
        Nodes=Reference.Elm.Top(iElm,[1 2 3 4]);
        % that's new for deformed shape
        for iDV=nDesVar
            VelFieldTotal=VelFieldTotal+...
                Reference.SensAnaIter(iMesh,iDV).VelField.Nodes...
                (:,Reference.CalcMesh(iMesh).NodesIndexation(Nodes))*...
                Reference.SensAnaGlob.Param.DesVarIncGeos(iDV);
            ErrorValue=ErrorValue+...
                Reference.SensAnaIter(iMesh,iDV).ErrorAnalysis.Error...
                (Reference.CalcMesh(iMesh).ElementIndexation(iElm))*...
                Reference.SensAnaGlob.Param.DesVarIncGeos(iDV);
        end
        % obtencion de error por elemento
        ErrElm(iElm)=ErrorValue;
        % NodElm=Nod.XYZ(Nodes,:);
        NodElm=Reference.Nod.XYZ(Nodes,:);
        NodElmdef=(NodElm+VelFieldTotal);
        Tri1=[NodElmdef(1,1) NodElmdef(1,2);...
            NodElmdef(2,1) NodElmdef(2,2);...
            NodElmdef(3,1) NodElmdef(3,2)];
        Tri2=[NodElmdef(1,1) NodElmdef(1,2);...
            NodElmdef(3,1) NodElmdef(3,2);...
            NodElmdef(4,1) NodElmdef(4,2)];
        AreaElm(iElm)= ...
            (NodElm(1,2)-NodElm(1,1))*(NodElm(2,3)-NodElm(2,2));
        MaxTri1=max(Tri1);
        MinTri1=min(Tri1);
        AreaTri1=...
            (MaxTri1(:,1)-MinTri1(:,1))*(MaxTri1(:,2)-MinTri1(:,2))/2;
        MaxTri2=max(Tri2);
        MinTri2=min(Tri2);
        AreaTri2=...
            (MaxTri2(:,1)-MinTri2(:,1))*(MaxTri2(:,2)-MinTri2(:,2))/2;
        AreaElmdef(iElm)=AreaTri1+AreaTri2;
        Triangulos(:, :,IndiceTriangulos+1)=Tri1;
        Triangulos(:, :,IndiceTriangulos+2)=Tri2;
        TriElm=[TriElm;iElm;iElm];
    end
end

```

DOCUMENTO III

```
IndiceTriangulos=IndiceTriangulos+2;
else
numK=Reference.Elm.KNum(iElm);
nContTri=size(Reference.MatK(numK).PsiEtaVert,3);
for iDV=nDesVar
    ErrorValue=ErrorValue+...
    Reference.SensAnaIter(iMesh,iDV).ErrorAnalysis.Error...
    (Reference.CalcMesh(iMesh).ElementIndexation(iElm))*...
    Reference.SensAnaGlob.Param.DesVarIncGeos(iDV);
end
ErrElm(iElm)=ErrorValue;
AreaElmdef(iElm)=0;
for iContTri=1:nContTri
    VertLoc=Reference.MatK(numK).PsiEtaVert(:, :, iContTri);
    NVert=shape_f_2d(VertLoc',1,4,0);
    Nodes=Reference.Elm.Top(iElm,[1 2 3 4]');
    NodElm=Reference.Nod.XYZ(Nodes,:);
    XNod=NodElm(1,:);
    YNod=NodElm(2,:);
    % cambiar a N*CoordNodos de Nod.XYZ
    XVertGlo=NVert*XNod;
    YVertGlo=NVert*YNod;
    VertCoor=zeros(3,2);
    VertCoor(:,1)=XVertGlo;
    VertCoor(:,2)=YVertGlo;
    VertCoor=VertCoor';
    % igual para campo de velocidad
    VelFieldTotal=zeros(2,3);
    for iDV=nDesVar
VelFieldNod=(Reference.SensAnaIter(iMesh,iDV).VelField.Nodes...
    (:,Reference.CalcMesh(iMesh).NodesIndexation(Nodes)))';
        VelFieldInt=(NVert*VelFieldNod)';
        VelFieldTotal=...
            VelFieldTotal+ VelFieldInt*...
            Reference.SensAnaGlob.Param.DesVarIncGeos(iDV);
    end
    Tridef=(VertCoor+VelFieldTotal)';
    MaxTridef=max(Tridef);
    MinTridef=min(Tridef);
    AreaTridef=...
        (MaxTridef(:,1)-MinTridef(:,1))*...
        (MaxTridef(:,2)-MinTridef(:,2))/2;
    AreaElmdef(iElm)=AreaElmdef(iElm)+AreaTridef;
    Triangulos(:, :, IndiceTriangulos+1)=Tridef;
    TriElm=[TriElm;iElm];
    IndiceTriangulos=IndiceTriangulos+1;
end

end
IndiceElementos=IndiceElementos+1;
end
ActElsCur=CalcMesh(iMesh).ActiveElements;
for iElmCur=ActElsCur
    NodesCur=Elm.Top(iElmCur,[1 2 3 4]');
    NodElmCur=Nod.XYZ(NodesCur,:);
    % numero de matriz k
    numKCur=Elm.KNum(iElmCur);
    GaussPts=MatK(numKCur).PsiEtaG; % 2xGaussPts
```

```

    NCur=shape_f_2d(GaussPts',1,4,0); % GaussPtsx4
    XCur=NodElmCur(1,:);
    YCur=NodElmCur(2,:);
    XGaussPts=NCur*XCur;
    YGaussPts=NCur*YCur;
    XGaussPtsGlo=[XGaussPtsGlo; XGaussPts];
    YGaussPtsGlo=[YGaussPtsGlo; YGaussPts];
end
% Triangulos=reshape(Triangulos,3,2,[]);
GaussPtsGlo(:,1)=XGaussPtsGlo;
GaussPtsGlo(:,2)=YGaussPtsGlo;
nTri=max(size(Triangulos));

ErrElmdef=zeros(length(Elm.Top),1);
DensErrElm(ActEls)=ErrElm(ActEls)./AreaElmdef(ActEls);
% GaussPtsValue(x,y,nelm,errelm)
GPV=zeros(length(GaussPtsGlo),4);
% GPV es una matriz que guarda las coordenadas de los puntos de
% gauss en las 2 primeras columnas, el número de elemento al que
% pertenece en la tercera
% y el valor del error que le corresponde en la cuarta

% para dibujar las deformadas
PlotDeformedShape(iMesh,iDesVar)
hold on
% se verifica que punto de gauss pertenece a cada triangulo
for iTri=1:nTri
    % calculo Bounding Box
    A=Triangulos(:, :, iTri);
    MaxTri=max(A);
    MinTri=min(A);
    BB=...
        GaussPtsGlo(:,1)<=MaxTri(:,1)...
        &GaussPtsGlo(:,2)<=MaxTri(:,2)...
        &GaussPtsGlo(:,1)>=MinTri(:,1)...
        &GaussPtsGlo(:,2)>=MinTri(:,2);
    PtsInBB=GaussPtsGlo(BB,:);
    n=sum(BB);
    triflag=zeros(n,1);
    for i=1:n
        [~,~,~,flag]=sa_XY2XiEtaNotRegular(PtsInBB(i,:),A,0);
        triflag(i)=flag;
    end
    PtsInBBnGP=find(BB);
    InTri=find(triflag);
    PtsInTri=PtsInBB(InTri,:);
    PtsInTrinGP=PtsInBBnGP(InTri,:);
    % asignacion del error a los puntos de gauss
    nElm=TriElm(iTri);

    GPV(:,1:2)=GaussPtsGlo;
    GPV(PtsInTrinGP,3)=nElm;
    GPV(PtsInTrinGP,4)=DensErrElm(nElm);

end
GPVred=zeros(length(find(GPV(:,3)>0)),4);
GPVred=GPV(find(GPV(:,3)>0),:);

errmin=min(GPVred(:,4));

```

DOCUMENTO III

```
errmax=max(GPVred(:,4));  
  
for iplot=1:size(GPV,1)  
    col=(GPV(iplot,4)-errmin)/(errmax-errmin);  
    if col<0  
        plot(GPV(iplot,1),GPV(iplot,2),...  
             'LineStyle','none',...  
             'Marker','.', 'MarkerSize',10, 'Color',[0 1 0]);  
    else  
        plot(GPV(iplot,1),GPV(iplot,2),...  
             'LineStyle','none',...  
             'Marker','.', 'MarkerSize',10, 'Color',[col 0 0]);  
    end  
    hold on  
end
```

2.2.2. PlotDeformedShape

Esta es una sencilla función que sirve para dibujar los elementos deformados para comprobar que el proceso se está llevando a cabo correctamente. Una vez se tenga la certeza de que todo funciona bien, se puede obviar el uso de esta función.

```
function sa_deformada3(iMesh,iDesVar)  
  
global Elm CalcMesh Nod SensAnaGlob SensAnaIter Iter Geo Reference  
MatK  
  
ActEls=Reference.CalcMesh(iMesh).ActiveElements;  
LCMatrix=Geo.Loads.Cases.Matrix;  
FEMLoadCase=LCMatrix(:,9)==1;  
  
if iDesVar==0  
    nDesVar=1:Reference.SensAnaGlob.Param.NDesVar;  
else  
    nDesVar=iDesVar;  
end  
NNodes=Reference.Elm.NNodes;  
figure  
for iElm=ActEls  
    ErrorValue=...  
        Reference.Iter(iMesh).ErrorAnalysis(FEMLoadCase).ErrNorm...  
        {300,1}(Reference.CalcMesh(iMesh).ElementIndexation(iElm));  
    if Reference.Elm.KNum(iElm)==1  
  
        Nodes=Reference.Elm.Top(iElm,[1 2 3 4]);  
        VelFieldTotal=zeros(2,4);  
        % that's new for deformed shape
```

```

for iDV=nDesVar
VelFieldTotal=VelFieldTotal+Reference.SensAnaIter...
    (iMesh,iDV).VelField.Nodes(:,Reference.CalcMesh...
        (iMesh).NodesIndexation(Nodes))*...
    Reference.SensAnaGlob.Param.DesVarIncGeos(iDV);
ErrorValue=ErrorValue+...
Reference.SensAnaIter(iMesh,iDV).ErrorAnalysis.Error...
(Reference.CalcMesh(iMesh).ElementIndexation(iElm))*...
Reference.SensAnaGlob.Param.DesVarIncGeos(iDV);

end
XYZdef=Reference.Nod.XYZ(Nodes,:'+VelFieldTotal;
patch(XYZdef(1,:),XYZdef(2,:),ErrorValue,'EdgeColor',
'black');
else
numK=Reference.Elm.KNum(iElm);
nContTri=size(Reference.MatK(numK).PsiEtaVert,3);
for iDV=nDesVar
    ErrorValue=ErrorValue+...

Reference.SensAnaIter(iMesh,iDV).ErrorAnalysis.Error...

(Reference.CalcMesh(iMesh).ElementIndexation(iElm))*...
    Reference.SensAnaGlob.Param.DesVarIncGeos(iDV);
end
AreaElmdef=0;
for iContTri=1:nContTri
    VertLoc=Reference.MatK(numK).PsiEtaVert(:, :, iContTri);
    NVert=shape_f_2d(VertLoc',1,4,0);
    Nodes=Reference.Elm.Top(iElm,[1 2 3 4]');
    NodElm=Reference.Nod.XYZ(Nodes, :)';
    XNod=NodElm(1, :)';
    YNod=NodElm(2, :)';
    %    cambiar a N*CoordNodos de Nod.XYZ
    XVertGlo=NVert*XNod;
    YVertGlo=NVert*YNod;
    VertCoor=zeros(3,2);
    VertCoor(:,1)=XVertGlo;
    VertCoor(:,2)=YVertGlo;
    VertCoor=VertCoor';
    %    igual para campo de velocidad
    VelFieldTotal=zeros(2,3);
    for iDV=nDesVar

VelFieldNod=(Reference.SensAnaIter(iMesh,iDV).VelField.Nodes...

(:,Reference.CalcMesh(iMesh).NodesIndexation(Nodes)))';
    VelFieldInt=(NVert*VelFieldNod)';
    VelFieldTotal=...
        VelFieldTotal+VelFieldInt*...
        Reference.SensAnaGlob.Param.DesVarIncGeos(iDV);
    end
    Tridef=(VertCoor+VelFieldTotal)';
    MaxTridef=max(Tridef);
    MinTridef=min(Tridef);
    AreaTridef=...
        (MaxTridef(:,1)-MinTridef(:,1))* ...
        (MaxTridef(:,2)-MinTridef(:,2))/2;
    AreaElmdef=AreaElmdef+AreaTridef;

```

DOCUMENTO III

```
        patch(Tridef(:,1),Tridef(:,2),ErrorValue, ...
            'EdgeColor', 'black');
    end
end
end

ActElsPlot=CalcMesh(iMesh).ActiveElements;
for iElm=ActElsPlot
    Nodes=Elm.Top(iElm,[1 2 3 4]');
    XYZ=Nod.XYZ(Nodes,:);
    patch(XYZ(1,:),XYZ(2,:),0*ones(1,NNodes),...
        'FaceColor','none','EdgeColor','red');
end
axis equal
```

2.3. sa_SetDesVar_Nurbs_1var

Esta función se utiliza a la hora de combinar más de un parámetro en una sola variable de diseño. Se relacionan en la variable matricial **ActivationFlags** y se define una relación entre ellas con una matriz de coeficientes **Coefs**.

```
function sa_SetDesVar_Nurbs_1var(DesignParameters)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% In this function we manually introduce the new coordinates of the
% points of geometry that we want to modify to find the velocity
% field on the boundary by finite differences.
% The process is repeated for each design variable.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Version:                                     Date:
24/02/2011
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

global Geo SensAnaGlob

%% Modification of geometry with design variables

% Number of Design Variables
SensAnaGlob.Param.ExactNorm=0.0558156294;
SensAnaGlob.Param.ExactSens=-5.08239878e-4;

%=====
% QUARTER OF CYLINDER PROBLEM (10bm_Cylinder)
%=====

if nargin==0
    SensAnaGlob.Param.NDesVar=1;
    for iDesVar=1:SensAnaGlob.Param.NDesVar
        % Design Variable increment
        SensAnaGlob.Param.DesVarInc(iDesVar)=1;
        SensAnaGlob.VelField.GeoPt(iDesVar)=Geo.Pt;
        Inc=SensAnaGlob.Param.DesVarInc(iDesVar);
        SensAnaGlob.VelField.GeoPt(iDesVar).XYZ(1,3)=...
        SensAnaGlob.VelField.GeoPt(iDesVar).XYZ(1,3)+(Inc*cosd(0));
        SensAnaGlob.VelField.GeoPt(iDesVar).XYZ(2,3)=...
        SensAnaGlob.VelField.GeoPt(iDesVar).XYZ(2,3)+(Inc*sind(0));
        SensAnaGlob.VelField.GeoPt(iDesVar).XYZ(1,4)=...
        SensAnaGlob.VelField.GeoPt(iDesVar).XYZ(1,4)+(Inc*cosd(90));
        SensAnaGlob.VelField.GeoPt(iDesVar).XYZ(2,4)=...
        SensAnaGlob.VelField.GeoPt(iDesVar).XYZ(2,4)+(Inc*sind(90));
    end
else
    % Nparameters
    ActivationFlags=[1 1 1 1]; % Ndesvar
end
```

DOCUMENTO III

```
SensAnaGlob.Param.NDesVar=size(ActivationFlags,1);
SensAnaGlob.Param.DesVarInc=0.01*ones(1,2);
%      Constant  Coef.a1  Coef.a2  Coef.a3  Coef.a4
Coefs=[  0      0      0      0      0;... % X Pt. 1
        0      0      0      0      0;... % Y Pt. 1
        5      0      0      0      0;... % X Pt. 2
        0      0      0      0      0;... % Y Pt. 2
        0      1      0      0      0;... % X Pt. 3
        0      0      0      0      0;... % Y Pt. 3
        0      0      0      0      0;... % X Pt. 4
        0      0      0      0      1;... % Y Pt. 4
        0      0      0      0      0;... % X Pt. 5
        5      0      0      0      0;... % Y Pt. 5
        0      0  1/(2^(1/2))  0      0;... % X Pt. 6
        0      0      0  1/(2^(1/2))  0];... % Y Pt. 6
for iDesVar=1:SensAnaGlob.Param.NDesVar
    SensAnaGlob.VelField.GeoPt(iDesVar)=Geo.Pt;
    Basis=...
        [1 DesignParameters+...

DesignParameters.*(SensAnaGlob.Param.DesVarInc(iDesVar).* ...
ActivationFlags(iDesVar,:))];

SensAnaGlob.VelField.GeoPt(iDesVar).XYZ=reshape(Coefs*Basis,2,[]);
end
end
```

Este ejemplo se basa en el caso de la tubería. Está definida por 6 puntos. Con la matriz de coeficientes se identifica que variable de diseño afecta a cierto parámetro geométrico, las coordenadas x e y de los puntos. La matriz se multiplica al vector de variables de diseño $\mathbf{a} = (1, a_1, a_2, a_3, a_4)^t$ siendo el primer valor la constante. Si un parámetro es constante, como la coordenada x del punto 2, se multiplica su valor por la constante 1. En cambio, si un parámetro si está afectado por una variable, como la coordenada y del punto 6 que se ve afectada por la variable 3, se pondrá su influencia en la cuarta columna en la fila correspondiente (la 12). Si la relación es proporcional el coeficiente es 1, o como en el ejemplo la relación es $1/\sqrt{2}$ para hacer una sola variable de diseño y esta relación es necesaria. Si no hay relación el coeficiente es 0.

2.4. sa_StressVM

Esta función calcula la tensión de Von Mises y su derivada como se ha explicado en el trabajo.

```
function [ StressVM, dStressVM ] = sa_StressVM( iMesh,iDesVar )
%sa_StressVM Calculates the Von Mises Stress at Nodes
% and his sensitivity using the stress values obtained from
% the FEM analysis

global SensAnaIter Iter SensAnaGlob Reference...
    CalcMesh Prob PlaneStress Geo BatchInfo

nu=Prob.nu;
LCMatrix=Geo.Loads.Cases.Matrix;
FEMLoadCase=find(LCMatrix(:,9)==1);
iRecov=300;
if iDesVar==0
    nDesVar=1:SensAnaGlob.Param.NDesVar;
else
    nDesVar=iDesVar;
end

ContEls=CalcMesh(iMesh).OnContour';
IntEls=CalcMesh(iMesh).Internal';
nIntElm=length(IntEls);
nContElm=length(ContEls);
NumEls=CalcMesh(iMesh).NumEle;

if BatchInfo.AnalysisMode2D==PlaneStress
    Cepsilon=[1 -0.5 0; -0.5 1 0; 0 0 3];
else
    Cepsilon=[1-nu+nu^2 -0.5-nu+nu^2 0;-0.5-nu+nu^2 1-nu+nu^2 0;0 0
3];
end
long=[];

for iVar=nDesVar
    long(iVar)=...
max(SensAnaIter(iMesh,iVar).ErrorAnalysis.ContourStresses.Length_N)
;
    StressVar=zeros(long(iVar)/3,NumEls);
    dStressVM(:,:,iVar)=StressVar;
end
StressVMmax=0;
StressVM=[];
for i=nDesVar

    for iIntElm=1:nIntElm
        sizeInt=...
size...
(SensAnaIter(iMesh,i).ErrorAnalysis.InternalStresses.Recovered_N);
        longInt=sizeInt(1);
        nNInt=longInt/3;
        for iNInt=1:nNInt
```

DOCUMENTO III

```
        sigma_x=...
    Iter(iMesh).ErrorAnalysis(1,FEMLoadCase).InternalStresses...
        (iRecov,1).Recovered_N(iNInt*3-2,iIntElm);
        sigma_y=...
    Iter(iMesh).ErrorAnalysis(1,FEMLoadCase).InternalStresses...
        (iRecov,1).Recovered_N(iNInt*3-1,iIntElm);
        tau_xy=...
    Iter(iMesh).ErrorAnalysis(1,FEMLoadCase).InternalStresses...
        (iRecov,1).Recovered_N(iNInt*3,iIntElm);
        sigma=[sigma_x sigma_y tau_xy];
        if i==1
            StressVM(iNInt,IntEls(iIntElm))=...
                (sigma*Cepsilon*sigma')^(1/2);
            if StressVM(iNInt,IntEls(iIntElm))>StressVMmax
                StressVMmax=StressVM(iNInt,IntEls(iIntElm));
                column=IntEls(iIntElm);
                row=iNInt;
            end
        end
    end
    dsigma_x=...
SensAnaIter(iMesh,i).ErrorAnalysis.InternalStresses.Recovered_N...
    (iNInt*3-2,iIntElm);
    dsigma_y=...
SensAnaIter(iMesh,i).ErrorAnalysis.InternalStresses.Recovered_N...
    (iNInt*3-1,iIntElm);
    dtau_xy=...
SensAnaIter(iMesh,i).ErrorAnalysis.InternalStresses.Recovered_N...
    (iNInt*3,iIntElm);
    dsigma=[dsigma_x dsigma_y dtau_xy];
    InvStressVM=1/StressVM(iNInt,IntEls(iIntElm));
    dStressVM(iNInt,IntEls(iIntElm),i)=...
        (InvStressVM*dsigma*Cepsilon*sigma');

end

end

for iContElm=1:nContElm
    longCont=...
    SensAnaIter(iMesh,i).ErrorAnalysis.ContourStresses.Length_N...
        (iContElm);
    nNCont=longCont/3;
    for iNCont=1:nNCont
        sigma_x=...

Iter(iMesh).ErrorAnalysis(1,FEMLoadCase).ContourStresses...
    (iRecov,1).Recovered_N(iNCont*3-2,iContElm);
    sigma_y=...
    Iter(iMesh).ErrorAnalysis(1,FEMLoadCase).ContourStresses...
        (iRecov,1).Recovered_N(iNCont*3-1,iContElm);
        tau_xy=...
    Iter(iMesh).ErrorAnalysis(1,FEMLoadCase).ContourStresses...
        (iRecov,1).Recovered_N(iNCont*3,iContElm);
        sigma=[sigma_x sigma_y tau_xy];
        if i==1
            StressVM(iNCont,ContEls(iContElm))=...
                (sigma*Cepsilon*sigma')^(1/2);
            if StressVM(iNCont,ContEls(iContElm))>StressVMmax
                StressVMmax=StressVM(iNCont,ContEls(iContElm));
```

```

        column=ContEls (iContElm);
        row=iNCont;
    end
end
    dsigmax=...
SensAnaIter (iMesh,i).ErrorAnalysis.ContourStresses.Recovered_N...
    (iNCont*3-2,iContElm);
    dsigmay=...
SensAnaIter (iMesh,i).ErrorAnalysis.ContourStresses.Recovered_N...
    (iNCont*3-1,iContElm);
    dtauxy=...
SensAnaIter (iMesh,i).ErrorAnalysis.ContourStresses.Recovered_N...
    (iNCont*3,iContElm);
    dsigma=[dsigmax dsigmay dtauxy];
    InvStressVM=1/StressVM (iNCont,ContEls (iContElm));
    dStressVM (iNCont,ContEls (iContElm),i)=...
        InvStressVM*dsigma*Cepsilon*sigma';

    end

end

end
Reference.StressVM=StressVM;
Reference.dStressVM=dStressVM;
% deltaVM=0;
for i=nDesVar
Reference.dStressOfMaxStressVM (i)=Reference.dStressVM (row,column,i)
;
end
Reference.MaxStressVM=StressVM (row,column);

end

```