

# DESARROLLO DE UNA INTERFAZ COMPUTACIONAL “ABAQUS-MATLAB” Y SU APLICACIÓN EN PROBLEMAS DE OPTIMIZACIÓN ESTRUCTURAL.

D. Decicco, A. Romero y A. Valdéz.

*Departamento de Ingeniería Mecánica. Universidad Tecnológica Nacional - Facultad Regional Córdoba.  
Maestro M. López esq. Cruz Roja Argentina. Ciudad Universitaria. C.P.A. X5016ZAA.  
Córdoba Capital. Córdoba. Argentina.*

[deciccodario@hotmail.com](mailto:deciccodario@hotmail.com), [daleakd@hotmail.com](mailto:daleakd@hotmail.com), [andres10valdez@hotmail.com](mailto:andres10valdez@hotmail.com)

Tutor: Dr. Ing. S.M. Giusti

*Departamento de Ingeniería Civil. Universidad Tecnológica Nacional - Facultad Regional Córdoba. CONICET  
Maestro M. López esq. Cruz Roja Argentina. Ciudad Universitaria. C.P.A. X5016ZAA.  
Córdoba Capital. Córdoba. Argentina. [sgiusti@civil.frc.utn.edu.ar](mailto:sgiusti@civil.frc.utn.edu.ar)*

## Introducción

Todo problema de ingeniería demanda plazos y recursos económicos y humanos; los cuales son naturalmente limitados y escasos. Ante esta situación, los ingenieros se encuentran en la necesidad de resolver estos problemas utilizando los recursos de la manera más eficaz posible, por lo que se hace imprescindible contar con herramientas cada vez más eficientes. En este sentido, las actuales herramientas de cálculo estructural poseen limitaciones siendo necesario utilizar más de una en un mismo proyecto.

Abaqus [1] es un programa basado en el Método de los Elementos Finitos [2] muy utilizado en el medio profesional, con la capacidad de resolver una gran cantidad de problemas de ingeniería con las más variadas condiciones de contorno, tipos de materiales y configuración de elementos finitos. Como desventaja se puede citar que no cuenta con un módulo de optimización estructural definido. Por otro lado, el software de cálculo MatLab cuenta con un paquete de elementos finitos diseñado especialmente para elementos lineales triangulares, lo que lo hace muy efectivo en tiempo y capacidad de cálculo. Así mismo es también su mayor desventaja ya que solo podemos utilizar este tipo de elemento para resolver los problemas pre-establecidos en su programación. Dado que el módulo de elementos finitos de Matlab no tiene la capacidad de resolver elementos de geometría arbitraria, ya que solo dispone de elementos denominados “triángulos de deformación constante”; la interfaz desarrollada le permite comunicarse con Abaqus, para resolver problemas utilizando la basta librería de elementos finitos, conjuntamente con los resolvedores y post-procesadores para cada tipo de elemento.

En respuesta a la necesidad constante de un mejor uso de los recursos disponibles, en este trabajo se presenta el desarrollo de una interfaz computacional entre estos dos programas comerciales, para aprovechar las ventajas individuales de cada uno a los fines de crear una herramienta de solución a problemas de ingeniería que no pueden ser resueltos al utilizarlos por separado. Esta interfaz es aplicada en un algoritmo de optimización de topología estructural basado en conceptos de derivada topológica.

## Desarrollo.

Para el diseño del problema y la resolución del mismo, Abaqus genera archivos de datos que contienen toda la información del modelo. Estos archivos se clasifican en dos grandes grupos: (i) archivos de entrada (\*.inp) y (ii) archivos de salida (\*.odb, \*.rpt). El primer grupo define las características del modelo a simular, como ser: elementos, materiales, condiciones de contorno y problema. El segundo grupo contiene toda la información obtenida de la simulación (resultados). La comunicación entre los programas se lleva a cabo mediante la manipulación, desde MatLab, de dichos registros. Se crearon, entonces, funciones que permiten la importación, el análisis y la exportación de los registros de datos.

La interfaz es, en concreto, un conjunto de funciones, programadas en MatLab, que permite el acceso a estos registros con la descripción del problema físico, la ejecución del mismo,

la exploración de la base de datos de salida (problema resuelto), y la reescritura del archivo de entrada correspondiente, con la actualización.

El acceso a la base de datos originada por Abaqus (durante la etapa de ejecución) se establece a través de comandos Python [1], extrayendo la información necesaria como el nombre de paso, los valores de tensión en los elementos, el nombre de instancia y el volumen de cada elemento. Además se crearon macros con el objetivo de generar reportes que proporcionen el valor de energía potencial total y las tensiones nodalizadas.

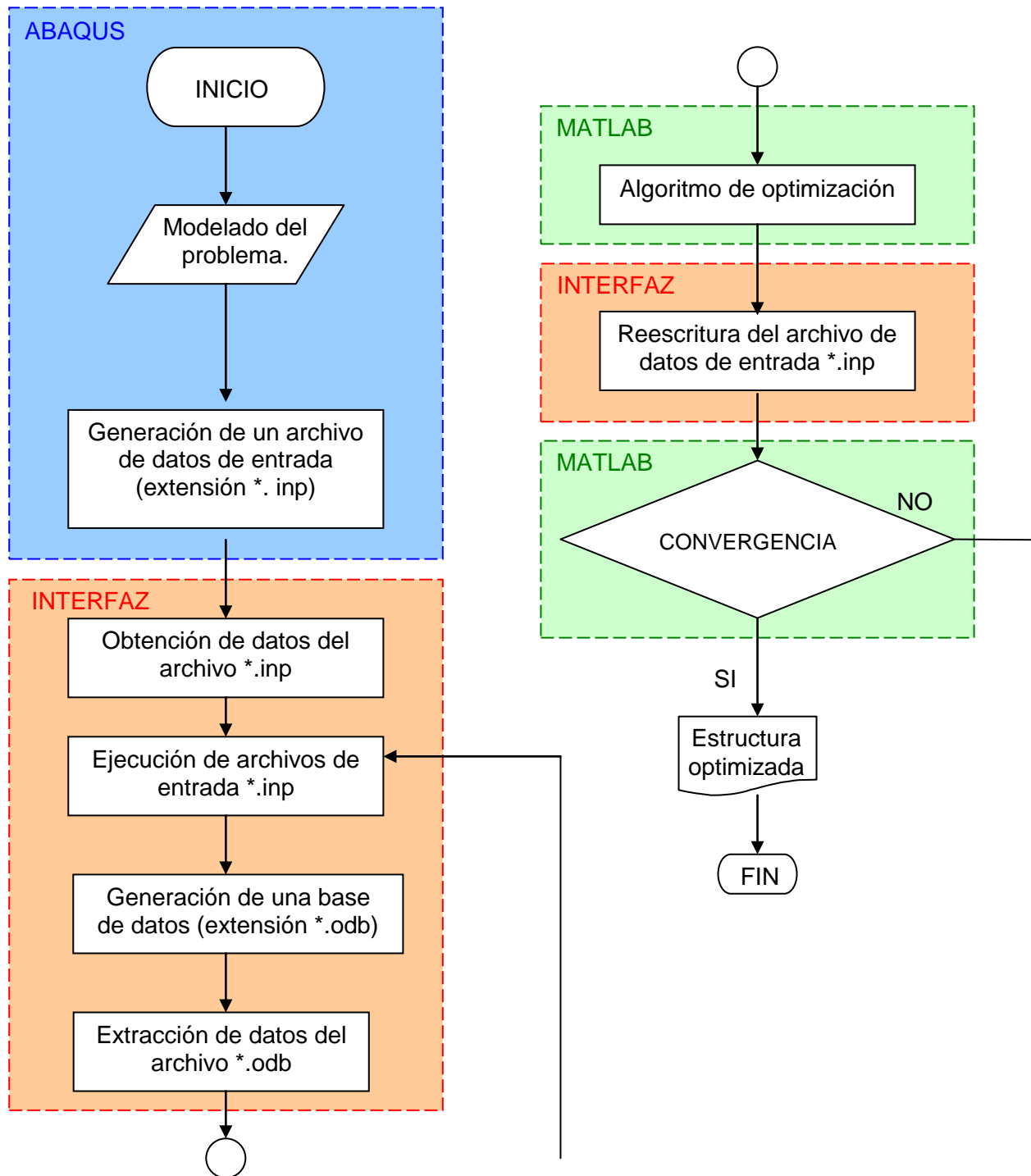


Figura 1. Diagrama de Flujo.

La función de optimización, administrada por MatLab, utiliza conceptos de derivada topológica [5,6] a los fines de establecer una distribución óptima del material en el dominio. La interfaz es capaz de distribuir, entonces, el material entre los elementos; siendo el optimizador quien gobierna esta acción. En la figura 1, se presenta el diagrama de flujo del proceso en el que podemos observar la importancia de la interfaz computacional como medio de enlace.

La interfaz creada cumple entonces con los siguientes propósitos:

- » Lectura del archivo de datos de entrada de Abaqus, de extensión \*.inp y extracción de información sobre el modelo.
- » Ejecución del archivo de datos \*.inp.
- » Acceso a la base de datos generada por Abaqus en la resolución del problema, \*.odb.
- » Creación de un reporte con las tensiones nodalizadas. \*.rpt.
- » Lectura del reporte, obtención de las tensiones promediadas y almacenamiento ordenado de las mismas.
- » Obtención de las tensiones en los elementos.
- » Almacenamiento de una matriz con las tensiones de cada elemento.
- » Cálculo de la energía potencial total.
- » Almacenamiento de energía potencial total en memoria.
- » Escritura de un nuevo \*.inp creando un inédito conjunto de elementos, cuya composición estará definida por el optimizador.

El algoritmo de optimización está basado en la representación geométrica del dominio a través de curvas de level-set. La dirección de descenso de la función a optimizar está guiada por la derivada topológica [5,6]. Una descripción detallada del algoritmo y su fundamentación matemática puede encontrarse en [3,4]. La ventaja de esta clase de algoritmo es un bajo costo computacional, ya que trabaja con una malla de elementos finitos fija y la función de level-set distribuye el material entre los elementos. Evitando el costo de crear una nueva malla de elementos finitos a cada cambio de topología durante el proceso de optimización.

Ejemplo de aplicación.

Se presenta seguidamente un ejemplo de aplicación resuelto mediante nuestra interfaz de enlace. Se trata de un problema de elasticidad lineal isotrópica. El dominio de análisis consta de un rectángulo de 70 cm x 45 cm, empotrado en su arista izquierda y libre en las demás, con una carga puntual de 10 N aplicada en el vértice superior derecho en dirección horizontal. El material está caracterizado por un módulo de Young de 210 GPa y un coeficiente de Poisson de 0,3. La siguiente imagen muestra la configuración inicial del problema, con sus condiciones de contorno. La función objetivo utilizada es la de energía potencial total y el problema de optimización consiste en minimizar la función objetivo sujeta a una restricción de volumen.

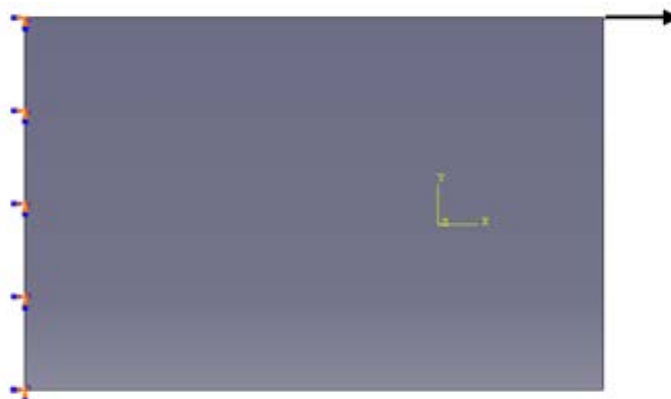


Figura 2. Configuración inicial del problema.

La malla de elementos finitos está caracterizada por el uso de elementos triangulares lineales de tensión plana (3 nodos), identificado por Abaqus con el código: "CPS3". La malla construida posee 1834 elementos con 976 nodos. El mallado del dominio se muestra en la figura siguiente.

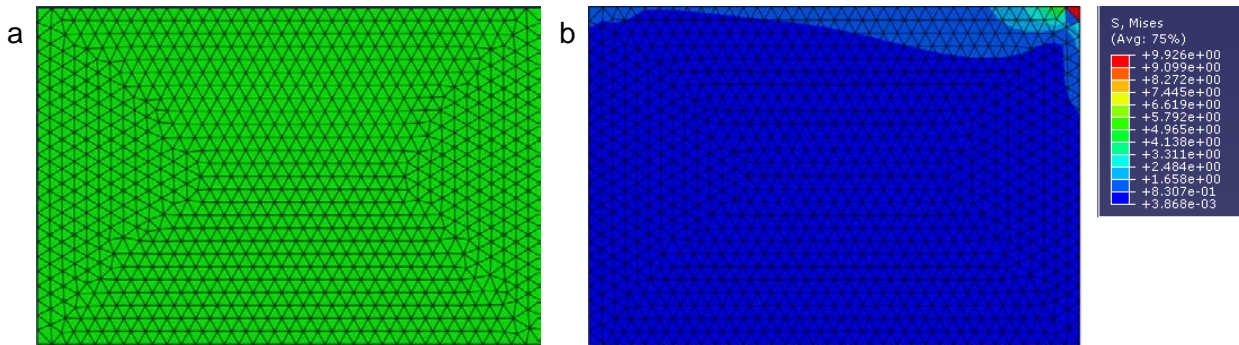


Figura 3. (a) Discretización del dominio (mallado). (b) Tensiones de von Mises.

Al aplicar el algoritmo de optimización, la estructura cuenta con dos grupos de elementos, cada uno con diferente material, eficientemente distribuidos. El material más débil (que se interpretará como un vacío) utilizado durante la secuencia de optimización posee un módulo de Young mil veces menor que el original y el mismo coeficiente de Poisson.

El resultado final que nos entrega el optimizador acoplado al software comercial, está reflejado en las siguientes imágenes, donde se distinguen gráficamente los dos grupos de elementos, siendo el gris oscuro el conjunto con material y el blanco el conjunto vacío. En el primer caso llegamos a tener un 65% del volumen inicial.

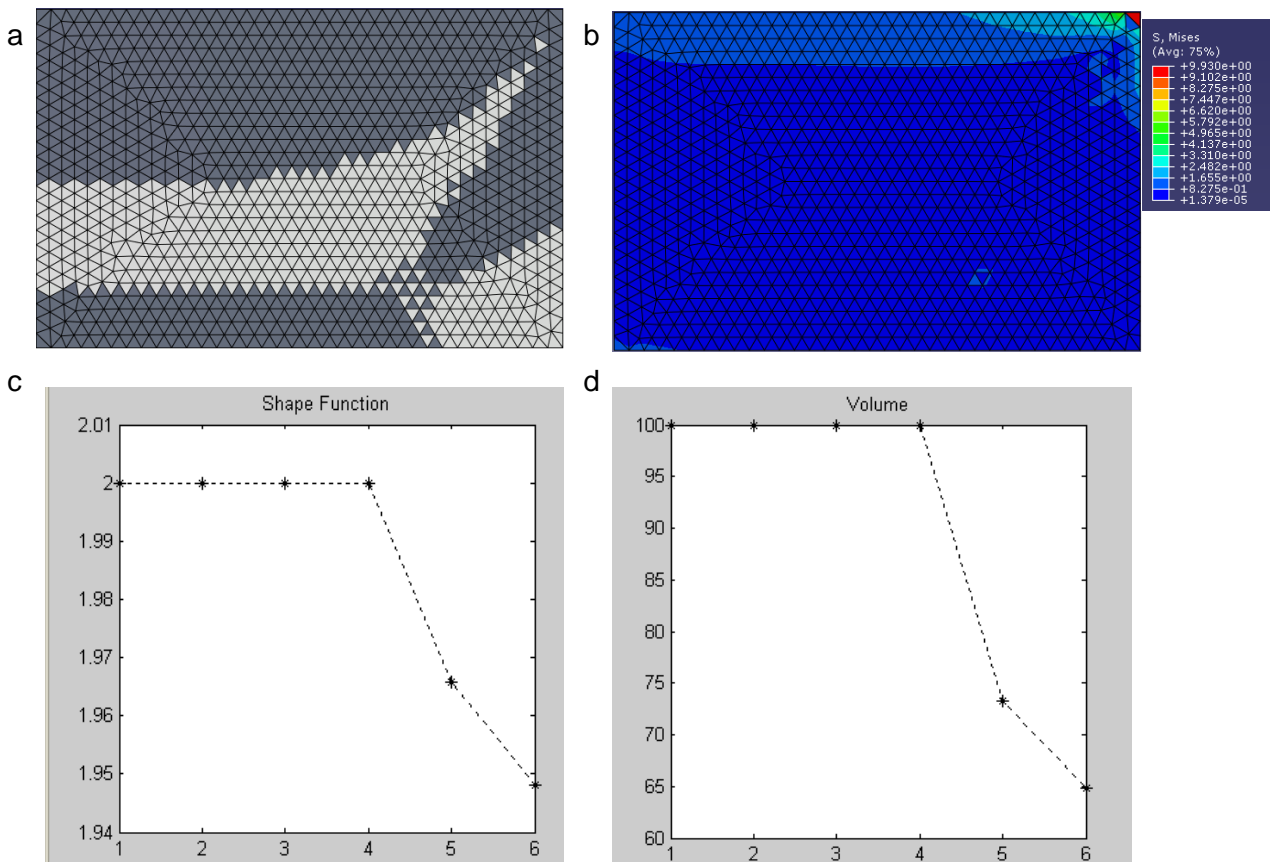


Figura 4. (a) Distribución optimizada de material. (b) Tensiones de von Mises. (c) Evolución de la función objetivo. (d) volumen de material duro vs. iteraciones.

En un segundo caso, obtenemos un volumen final de 24%. Los resultados son mostrados en la figura 5.

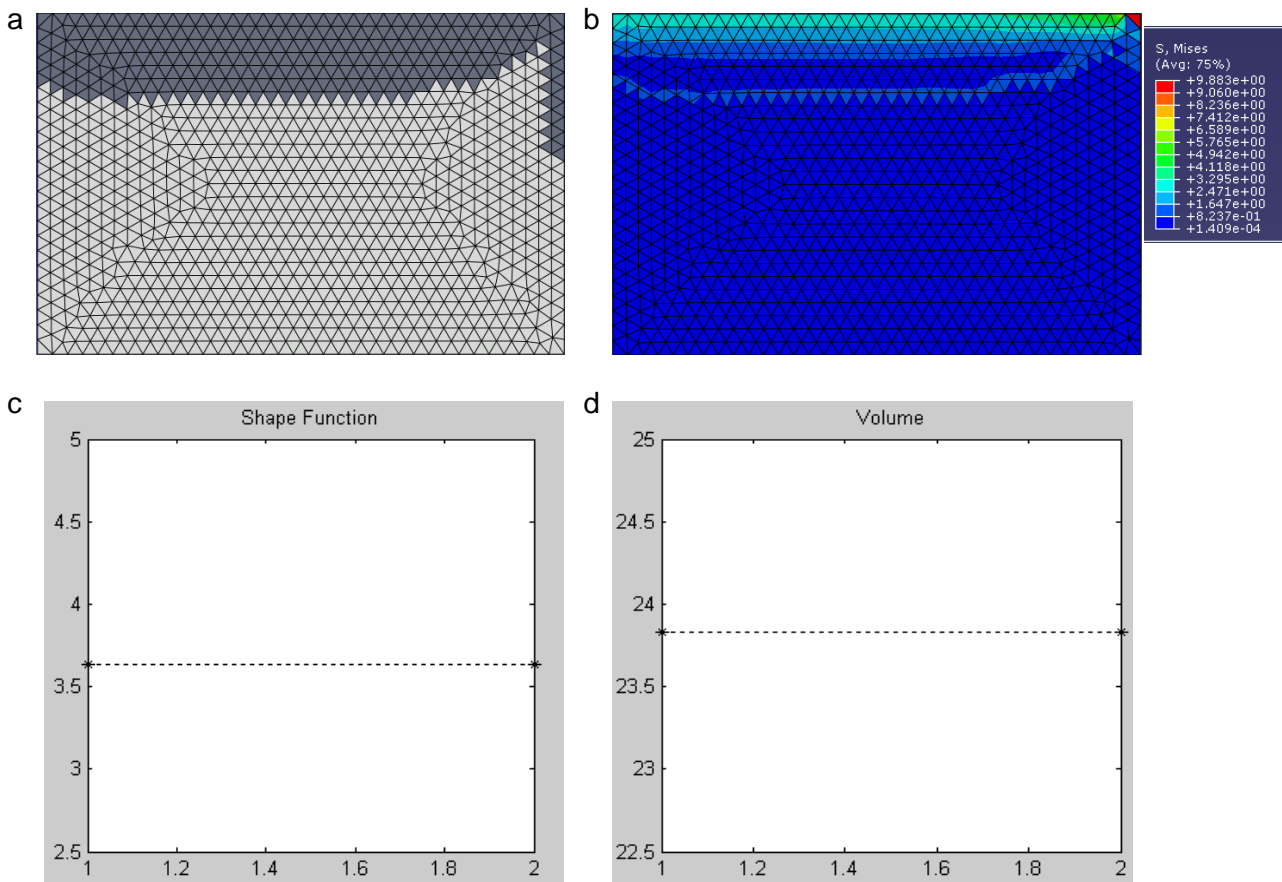


Figura 5. (a) Distribución optimizada de material. (b) Tensiones de von Mises. (c) Evolución de la función objetivo. (d) volumen de material duro vs. iteraciones.

Las figuras 4(c) y 5(c) muestran el comportamiento de la función de forma y la disminución de volumen en el transcurso de las iteraciones.

### Conclusión.

Mediante el desarrollo e implementación de la interfaz de enlace, se ha logrado la conjunción de las capacidades individuales de cada uno de estos programas comerciales, de amplia difusión en la resolución de problemas de ingeniería de nuestro medio.

Disponemos de una herramienta de diseño eficiente, amigable al usuario y capaz de racionalizar los recursos; a la cual sólo debemos ingresarle la información del modelo a optimizar, a través del archivo de datos de entrada (\*.inp), sin necesidad de conocer los fundamentos matemáticos del optimizador, o la programación por línea de comandos.

En particular, en el ejemplo mostrado observamos que habiendo reducido, en primera instancia un 35% el volumen y en segunda, un 75%; las tensiones medias y máximas se han conservado en un rango de variación del 1%. Esto indica que la estructura se ha saturado de tensiones, administrando de mejor modo todo el material disponible. Vemos que la estructura, inicialmente de geometría plana, luego de ser optimizada, se asemeja a un elemento unidimensional (barra). Finalmente, se destaca que el costo computacional de la interfaz y el algoritmo de optimización es bajo ya que se ha logrado convergencia a la topología óptima en reducidas iteraciones.

Actualmente la interfaz de enlace resuelve problemas bidimensionales de elasticidad lineal isotrópica, estando en etapa de desarrollo su extensión para problemas tridimensionales, y otros problemas del continuo a los cuales se pueda establecer un modelo discreto mediante el Método de los Elementos Finitos.

#### Bibliografía.

- [1] Abaqus documentation. Abaqus 6.8 Student Edition.
- [2] Hughes T.J.R. The finite element method. Linear static and dynamic finite element analysis. New Jersey: Prentice-Hall. 1987.
- [3] S. Amstutz & H. André. A new algorithm for topology optimization using a level-set method. *Journal of Computational Physics*, 216(2): 573-588, 2006.
- [4] S. Amstutz, S. M. Giusti, A. A. Novotny, E. A. de Souza Neto. Topological derivative for multi-scale linear elasticity models applied to the synthesis of microstructures. *International Journal for Numerical Methods in Engineering*, 84(6): 733-756, 2010.
- [5] A.A. Novotny, R.A. Feijóo, C. Padra & E. Taroco. Topological sensitivity analysis. *Computer Methods in Applied Mechanics and Engineering*, 192:803-829, 2003.
- [6] J. Sokolowski & A. Zochowski. On topological derivative in shape optimization. *SIAM Journal on Control and Optimization*, 37:1251-1272, 1999.