

Práctica 1

Optimización topológica con Matlab

Equipo 4

Omar Gutierrez, Pietro Castañeda, Julio Adrian, Víctor Santiago, Lauro Garcia

21 de septiembre de 2022

Resumen

Este trabajo propone la aplicación de un código de 99 líneas en la optimización topológica de una pieza con el uso del criterio de optimalidad estándar, este código fue realizado en Matlab por el profesor Ole Sigmund, así mismo se aborda la teoría necesaria sobre la optimización topológica y el ambiente de programación en Matlab.

1. Introducción

El problema de Optimización Topológica (OT) consiste en buscar una distribución óptima de material en un dominio de diseño que satisfaga las solicitaciones y las condiciones de borde definidas. Generalmente, la OT ha sido formulada en términos de minimizar la energía de deformación de la estructura analizada.

En la mayoría de trabajos sobre OT reportados se encuentra que los métodos de optimización comúnmente empleados son: el criterio de Optimalidad Estándar (Optimality Criteria), el Método de la Curva de Nivel (Level Set Method) y la eficiencia de Pareto (Pareto Optimal Tracing), entre otros. De forma general, todas estas técnicas exploran el espacio de búsqueda de una manera controlada y tienen la ventaja de no depender del cálculo de derivadas para llevar a cabo el proceso de optimización. Entre estos, con el programa de 99 líneas que se selecciono para el estudio de una pieza definida, se empleó el criterio de Optimalidad Estándar (Optimality Criteria).

En su primera parte este trabajo presenta el estado del arte necesario para abordar por primera vez la optimización topológica y el ambiente de programación en matlab, también la descripción del problema de OT y los resultados que se obtuvieron de la implementación del código.

2. Objetivo

El estudiante conocerá cada una de las secciones que integran el código de optimización topológica, como se debe crear un archivo (.m) en MATLAB y cómo se ejecuta el análisis.

El estudiante deberá presentar una propuesta de análisis de formas de programación, de características de trabajo específicas (programación) que presenta. La metodología que se utilizará para la propuesta se compondrá en los siguientes pasos:

- Nombre y definición de la programación mencionar un ejemplo de forma de la GEOMETRIA
- Estado del arte
- Procedimiento de la programación
- Implementación o desarrollo de la programación en sus diferentes vistas
- Conclusiones por cada autor.

3. Estado del arte

3.1. Optimización topológica

Este método derivado de las matemáticas fue claramente definido, explicado y hecho utilizable para la mecánica en la década de 2000, en particular con el artículo fundador de Ole Sigmund.

El software de optimización topológica cada vez más sofisticado permite a los ingenieros guardar el material posible para un objeto mientras mantienen o mejoran su fuerza o flexibilidad (si es necesario) y teniendo en cuenta las restricciones que se colocarán en él. Anteriormente, se basa en la intuición, el método de prueba y error y / o el genio de los creadores y / o ingenieros de fabricación [2].

3.1.1. ¿Qué es la Optimización Topológica?

La optimización de la topología (TO) es un método matemático que optimiza el diseño del material dentro de un espacio de diseño dado, para un conjunto dado de cargas, condiciones de contorno y restricciones con el objetivo de maximizar el rendimiento del sistema. TO es diferente de la optimización de la forma y la optimización del tamaño en el sentido de que el diseño puede alcanzar cualquier forma dentro del espacio de diseño, en lugar de tratar con configuraciones predefinidas.

La formulación convencional de TO utiliza un método de elementos finitos [FEM] para evaluar el desempeño del diseño [3]. El diseño se optimiza utilizando técnicas de programación matemática basadas en gradientes, como el algoritmo de criterios de optimalidad y el método de mover asíntotas o algoritmos no basados en gradientes, como los algoritmos genéticos.

3.1.2. ¿Cómo se hace Optimización Topológica?

En el proceso de optimización topológica, se deben de tener en cuenta varios aspectos; el espacio de diseño, el o los casos de carga que va a sufrir la pieza en cuestión, el material y la tecnología con que se va a realizar su fabricación, la reducción de costes mediante la minimización de soportes y aprovechamiento de la cuba de impresión, en caso de utilizar tecnologías aditivas, y muchos más [2].

Pasos Optimización Topológica:

1. Dibujar o Importar geometría
2. Simplificar la pieza y definir el espacio de diseño
3. Establecer uniones, juntas y contactos
4. Asignar materiales
5. Definir los casos de carga
6. Generar la optimización
7. Refinar la geometría
8. Exportar a CAD o generar STL
9. Verificar el rendimiento
10. Fabricar

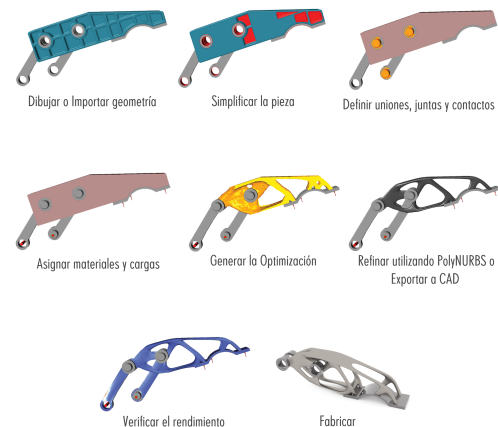


Figura 1: Flujo de trabajo

3.1.3. Beneficios

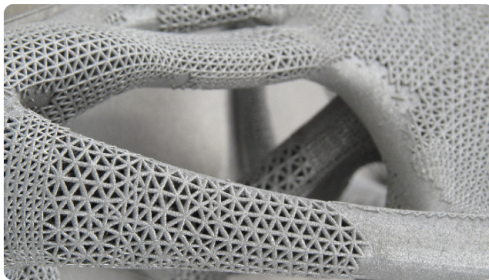
Estos son los principales beneficios:

- Reducción de peso y volumen
- Maximizar la resistencia
- Aprovechamiento de la cuba de fabricación
- Reducción de costes
- Minimiza los tiempos de fabricación
- Diseños orgánicos y más atractivos



Figura 2: Beneficios

Debido a la enorme red de miembros estructurales, los diseños optimizados de lattice tienden a exhibir una mejor estabilidad y un mejor comportamiento térmico. También cuentan con excelentes características de peso y se utilizan como una solución para la reducción de peso. Los diseños de lattice se adaptan particularmente bien a las aplicaciones biomédicas, como implantes, ya que la naturaleza porosa de la estructura facilita el crecimiento de los huesos y del tejido [2].



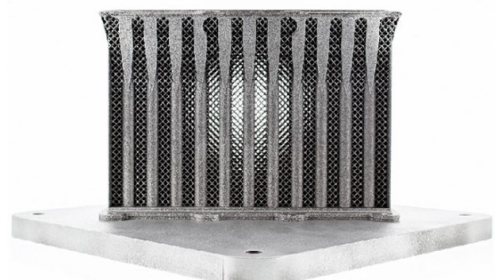
(a) Reducción de peso



(b) Mejor estabilidad



(c) Aplicaciones biomédicas



(d) Mejor comportamiento térmico

Figura 3: Optimización lattice

3.2. Matlab

MatLab es una plataforma de programación diseñada por la compañía MathWorks, con fines ingenieriles y científicos. MatLab utiliza su propio lenguaje de programación basado en matrices, permitiendo una expresión más natural de matemáticas computacionales. Usando MatLab, se pueden implementar y diseñar algoritmos, resolver ecuaciones, realizar operaciones con matrices, representar información gráficamente, realizar simulaciones de modelos matemáticos, analizar señales, ajustar sistemas de control, programar aplicaciones, etc [1]. MatLab también es capaz de realizar operaciones de optimización de topología, y es un software muy robusto y completo para esto.

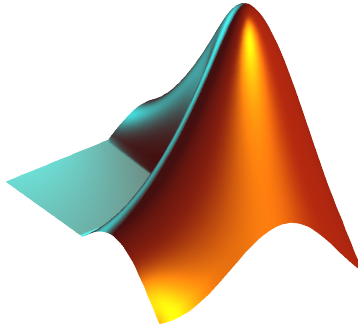


Figura 4: Logo Matlab

Como se mencionó, también es capaz de realizar operaciones de optimización de topología, sin embargo, esta no es soportada oficialmente y se tiene que recurrir a códigos hechos por el usuario. Existen variaciones de código utilizando varios métodos, como el de BESO (Bidirectional Evolutionary Structural Optimization) y TOBS (Topology Optimization of Binary Structures). Dichos códigos pueden variar entre las 88 a 100 líneas de código y están bastante documentados y accesibles en internet.

3.2.1. Ventajas y desventajas de uso

Algunas de las ventajas que ofrece Matlab son:

- Lenguaje de programación accesible y fácil de aprender.
- Grandes capacidades de personalización, con módulos adicionales que brindan más funcionalidad a Matlab.
- Es un programa poderoso y ligero que permite realizar una gran cantidad de operaciones matemáticas rápidamente.
- Acceso a grandes bibliotecas de paquetes matemáticos para cualquier campo de matemáticas, desde operaciones básicas hasta avanzadas.
- Sistema de gráficos Simulink, que permite visualización de datos bidimensionales y tridimensionales, procesamiento de imágenes, presentación de gráficos y animaciones.

Desventajas de usar Matlab:

- La interfaz se basa en trabajos de línea de comandos, puede resultar difícil de navegar y no existen menús de acceso rápido a procedimientos de uso frecuente.
- Necesidad de conocer matemáticas para comprender el uso de funciones y procedimientos avanzados.

4. Ejemplo de optimización topológica

Para comenzar el proceso, se toma en cuenta el diseño, la carga que se aplicara y en que punto se aplicara, el material a utilizar, entre muchos otros. El proceso básicamente consiste en diseñar una pieza la cual buscaremos optimizar, para esto hay que definir la forma, material, uniones, puntos de contacto y restricciones de la pieza y posteriormente se aplicarán cargas simuladas a esta pieza y se evaluará las partes donde exista un mayor esfuerzo, y también las partes en las cuales el esfuerzo sea menor.

Posteriormente el software se encarga de generar una nueva geometría para la pieza la cual tendrá una menor cantidad de material, pero buscará cumplir con todas las especificaciones mecánicas requeridas. Por último se hace una verificación para probar que la pieza generada cumple con las características especificadas y si es así se completa el proceso de diseño.

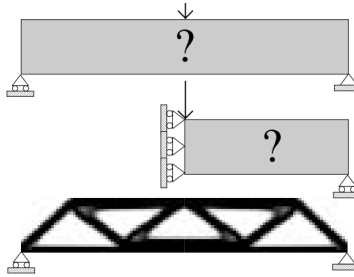


Figura 5: Ejemplo de optimización topológica

4.1. Procedimiento de programación

A continuación se muestran los pasos correspondientes para la aplicación del programa de 99 líneas de optimización topológica realizado por Ole Sigmund.

1. Dirigirse a la pagina: <https://www.topopt.mek.dtu.dk/apps-and-software/a-99-line-topology-optimization-code-written-in-matlab> y descargar el archivo .m dando click en lo encerrado.

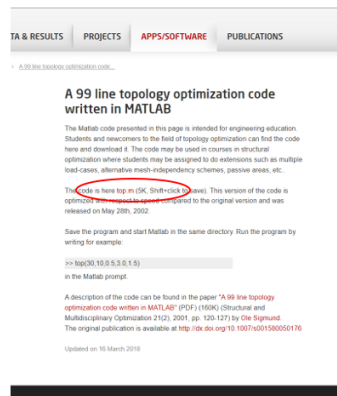


Figura 6: Descargar top.m

2. Abrir Matlab, hacer la combinación Ctrl + O, o seleccionar la carpeta con nombre Open.

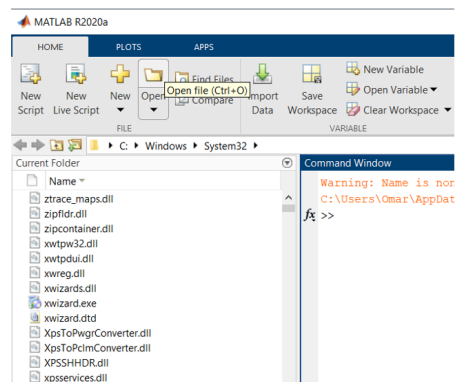


Figura 7: Seleccionar Open.m

3. Seleccionar la carpeta en donde se descargó el archivo y dar doble click en este para abrirlo.

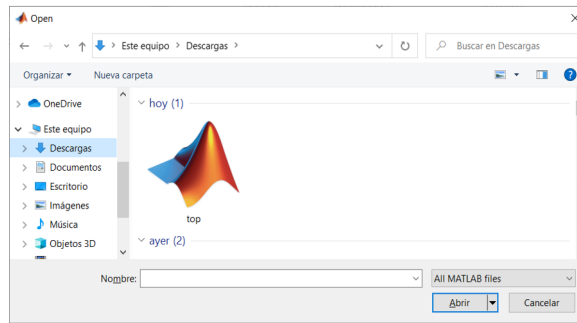


Figura 8: Abrir archivo top.m

Con solo estos tres pasos tenemos el código para realizar la optimización topológica de cualquier pieza básica.

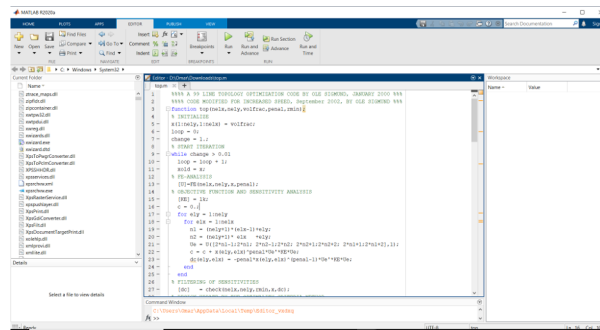


Figura 9: Código de optimización topológica

4.2. Geometría a optimizar

Una vez comprobamos que el código se paso correctamente, se planteó una geometría para analizar, se eligió una sencilla ya que creemos que es la mejor forma de comprender el funcionamiento del programa. La geometria planteada se muestra en la figura 10:

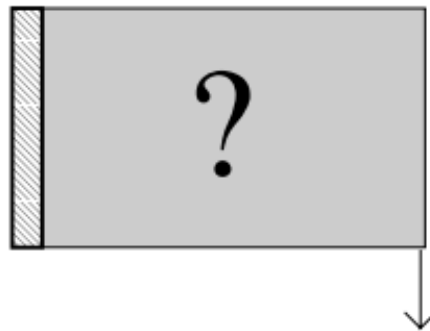


Figura 10: Código de optimización topológica

4.2.1. Funcionamiento del programa

- Programa Principal (1-36)

El programa principal comienza distribuyendo el material uniformemente en el dominio de diseño (línea 4). Después se presentan las iteraciones que permitan que la optimización de la pieza sea cada vez mejor y más visible [4]. El ciclo principal comienza con una llamada a la subrutina de elementos finitos (línea 12) que devuelve el vector de desplazamiento U . Dado que la matriz de rigidez del elemento para material sólido es la misma para todos los elementos, la subrutina de matriz de rigidez del elemento se llama una sola vez (línea 14).

Después de esto, un bucle sobre todos los elementos (líneas 16 a 24) determina la función objetivo y las sensibilidades (4). Las variables $n1$ y $n2$ denotan números de nodo de elemento superior izquierdo y derecho en números de nodo global y se utilizan para extraer el vector de desplazamiento de elemento U_e del vector de desplazamiento global U . El análisis de sensibilidad es seguido por una llamada al filtro de independencia de malla (línea 26) y el optimizador (línea 28). El cumplimiento actual, así como otros parámetros, se imprimen en las líneas 30 a 33 y se traza la distribución de densidad resultante (línea 35).

- Criterio de Optimalidad Estándar (37-48)

El optimizador encuentra las variables de diseño actualizadas (líneas 37-48). Sabiendo que el volumen material ($\text{sum}(\text{sum}(x_{\text{new}}))$) es una función monótonamente decreciente del multiplicador de Lagrange (lag), el valor del multiplicador de Lagrange que satisface la restricción de volumen se puede encontrar mediante un algoritmo de bisección (líneas 40-48).

El algoritmo de bisección se inicializa preguntando un límite inferior $l1$ y un límite superior $l2$ para el multiplicador lagrangiano (línea 39). El intervalo que delimita el multiplicador lagrangiano se divide repetidamente por la mitad hasta que su tamaño es menor que el criterio de convergencia (línea 40).

- Filtrado de independencia de malla (49 - 64)

Tenga en cuenta que no se buscan todos los elementos en el dominio de diseño para encontrar los elementos que se encuentran dentro del radio r_{min} , sino solo aquellos dentro de un cuadrado con longitudes de lado dos veces redondas (r_{min}) alrededor del elemento considerado. Seleccionando r_{min} menor a uno en la llamada de la rutina, las sensibilidades filtradas serán iguales a las sensibilidades originales haciendo el filtro inactivo.

- Código de elemento finito (65 - 99)

Hay que tener en cuenta, como menciona el artículo, que el solucionador utiliza la opción escasa en Matlab. La matriz de rigidez global está formada por un bucle sobre todos los elementos (líneas 70-77). Como era el caso en el programa principal, las variables $n1$ y $n2$ denotan números de nodo de elementos superiores izquierdo y derecho en números de nodos globales y se utilizan para insertar la matriz de rigidez del elemento en los lugares correctos en la matriz de rigidez global [4].

Como se mencionó anteriormente, tanto los nodos como los elementos son numerados en columna de izquierda a derecha. Es más, cada nodo tiene dos grados de libertad (horizontal y vertical), por lo que el comando $F(2,1)=-1$. (línea 79) se aplica una fuerza de fuerza unitaria vertical en la esquina superior izquierda.

4.2.2. Resultados

Una vez teníamos la geometría a optimizar, solo restaba utilizar el script proporcionado las medidas de la geometría y los demás parametros fueron elegidos por aproximación a lo que venía por defecto en el programa:

- nel x = 32
- nel y = 20
- volfrac = 0.4
- penal = 3.0
- rmin = 1.2
- top(32,20,0.4,3.0,1.2)

Fueron cambiados los grados de libertad y la posición de la fuerza, ya que este se encuentra empotrado y con una fuerza en la esquina derecha:

- 79. $F(2*(nelx+1)*(nely+1),1) = -1;$
- 80. $fixeddofs = [1:2*(nely+1)];$

Para iniciar el código es necesario dar click en Run, ingresar top(32,20,0.4,3.0,1.2) y presionar Enter.

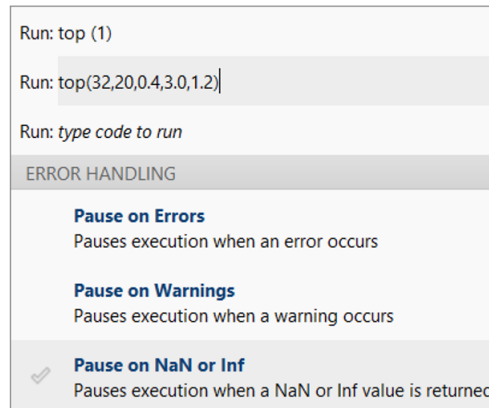
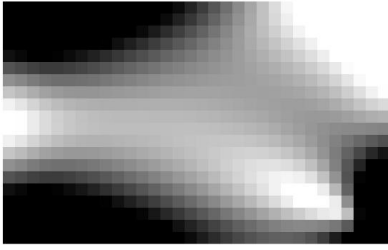


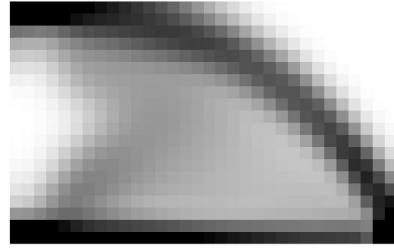
Figura 11: Inicialización de código

Para entender el comportamiento del programa se probará la pieza con diferentes iteraciones para observar la optimización topológica, esto se muestra a continuación en la figura 12

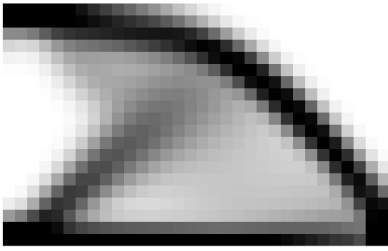
En esta se observa que en base a cada iteración el material se va distribuyendo o borrando de las zonas en las cuales no se ven afectadas por la carga y manteniendo las esenciales, esto se logra por el criterio de optimalidad, que va reduciendo el volumen de la pieza hasta una restricción que se complementa con la parte de elemento finito que va modificando la matriz de rigidez y el filtrado de malla que permite mantener las uniones con un rmin para obtener una mejor optimización de la geometría.



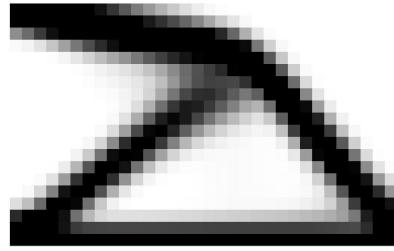
(a) 1 iteración



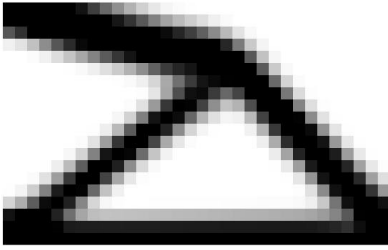
(b) 2 iteraciones



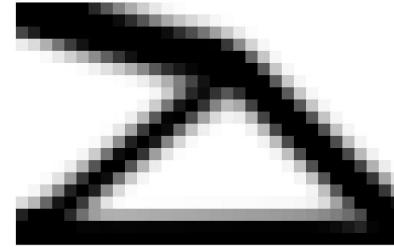
(c) 8 iteraciones



(d) 13 iteraciones



(e) 45 iteraciones



(f) 143 iteraciones

Figura 12: Optimización topológica en base a iteraciones

En caso de querer trabajar con otras condiciones, ya sea de grados de libertad o ahora aplicar en diferentes direcciones las cargas, se considera relativamente fácil, ya que, si se desea cambiar el cómo estará sujeta, los cambios ocurrirían en la línea 79 modificando $nelx$ y $nely$, línea 80 modificando los grados de libertad que presente la geometría y si se desea agregar otras cargas a la geometría, hay que modificar los vectores de fuerza y desplazamiento en la línea 69, por ejemplo en caso de agregar dos cargas en sentidos opuestos quedaría:

$F = \text{sparse}(2*(nely+1)*(nelx+1),2); U = \text{zeros}(2*(nely+1)*(nelx+1),2);$

Cambiando simplemente el ultimo valor que son el número de fuerzas y desplazamientos .

5. Conclusiones

- Omar Alejandro Gutierrez Carrizales :

La realización de esta práctica me permitió conocer que la optimización topológica no solo se realiza en los CAD, si no que también se puede realizar desde Matlab importando una figura o programando las condiciones de una figura inicial.

Además con esto queda me claro que la optimización topológica es de gran ayuda ya que elimina partes de la geometría que no están sometidas a esfuerzos y por lo tanto no aportan mucho a la estructura más que un gasto, espacio y quitan un poco la estética, dejando solo así las partes fundamentales para que la geometría pueda seguir realizando la acción para la que fue realizada con poco o casi nulo cambio.

- Pietro Augusto Castañeda Moreno:

La optimización topológica, de entre otras tantas, resulta vital para la industria contemporánea, ya que permite la creación de productos con una menor cantidad de recursos. De esta manera, no se limita la producción de piezas, y se disminuye el impacto en el medio ambiente.

El proceso de optimización requiere de una serie de cálculos, por lo que se auxilia de procedimientos de álgebra lineal, así como de recursos como las matrices, y el Lagrangiano. Se entiende la presencia de las matrices porque se encuentran en el Análisis de Elemento Finito, se entiende la presencia del lagrangiano ya que se utiliza en todo tipo de sistemas físicos para expresar la evolución de un sistema físico.

- Víctor Christopher Santiago Martínez:

En conclusión con la práctica anterior, la optimización topológica es en parte útil indispensable para un sin número de procesos dentro de la industria, ya que, debido como su nombre lo dice, optimiza y se puede llegar a un uso reducido de recursos en cuanto a la aplicación dada. Este mismo proceso está dado y desarrollado con una gran cantidad de cálculos vistos en la práctica. Finalmente, concluyo con que es de suma importancia al punto de ser vital para una variedad de procesos.

- Luis Lauro García Hernández:

En esta práctica la optimización topológica es usada para la optimización de un material, esta forma de usarlo es muy eficaz y facilita el diseño y disminuye el uso de recursos. Todo esto se hace mediante cálculos, los cuales se hacen en forma de matrices, en este caso se usó el programa que se llama MATLAB, el cual es un programa de programación para fines de ingeniería o científicos y en este caso se usó para implementar la optimización topológica y demostrar como se usa.

- Julio Adrián Torres Cisneros:

En esta práctica se pudo ver un ejemplo y aprender más acerca de qué son y para qué sirven los algoritmos de optimización topológica, así también se vio un ejemplo de una estructura un código de Matlab de solo 99 líneas, el cual se encargaba de optimizar una estructura para ciertas cargas definidas.

La optimización topológica es importante en el área de la biomecánica, ya que siempre se busca prótesis o mecanismos lo más cómodos, resistentes y baratos para que estos puedan tener un buen desempeño, pero sin pesar demasiado, ya que puede complicarle al usuario el uso de dicha prótesis, este es un claro ejemplo en donde la optimización topológica tiene un gran campo pues se pueden analizar las estructuras a emplear en las prótesis para que éstas sean lo más ligeras posibles pero sin sacrificar su resistencia.

Referencias

- [1] Matlab - el lenguaje del cálculo técnico. <https://la.mathworks.com/products/matlab.html>. Acceso: 18-09-2021.
- [2] Optimización topológica. <https://eitaingenieros.com/optimizacion/>. Acceso: 18-09-2021.
- [3] O. Begambre Carrillo C. Millán Páramo. Solución de problemas de optimización topológica empleando el algoritmo simulated annealing modificado. Recuperado de: <https://www.sciencedirect.com/science/article/pii/S0213131515000127>, 2016.
- [4] Ole Sigmund. A 99 line topology optimization code written in matlab. Recuperado de: <https://www.topopt.mek.dtu.dk/apps-and-software/a-99-line-topology-optimization-code-written-in-matlab>, September 2002.