UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN

FACULTAD DE INGENIERÍA MECÁNICA Y ELÉCTRICA

SEMESTRE AGOSTO-DICIEMBRE 2022

LABORATORIO DE BIOMECÁNICA

PRÁCTICA 1

PROFESORA: M.C. YADIRA MORENO VERA

NOMBRE	MATRÍCULA
Diana Elisa Acosta Rodríguez	1914714
Joaquin Abdiel Coronado Vázquez	1924388
Brenda Giselle Hinojosa	1910669
Gloria Rosalía Domínguez Azueta	1913999
Andrea Anette Celestino Castillo	1925924
Ana Paulina Pérez Salazar	1865049
Javier Cisneros Saldivar	1856682

BRIGADA: 204 **AULA:** 12BMC **DÍA:** MARTES **HORA:** V1

San Nicolás de los Garza, N.L. a 5 de septiembre de 2022.

ÍNDICE

PRÁCTICA 1:

DESCRIPCIÓN Y USO DEL CÓDIGO DE OPTIMIZACIÓN TOPOLÓGICA DE 99 LINEA EN MATLAB.

1. Objetivo

Conocer cada una de las secciones que integran el código de optimización topológica, como se debe de crear el archivo (.m) en MATLAB y como se ejecuta el análisis de una pieza simulada.

2. Marco teórico

MATLAB es una plataforma de programación y cálculo numérico utilizada por millones de ingenieros y científicos para analizar datos, desarrollar algoritmos y crear modelos.

Este software combina un entorno de escritorio perfeccionado para el análisis iterativo y los procesos de diseño con un lenguaje de programación que expresa las matemáticas de matrices y arrays directamente.

El programa: MATLAB (abreviatura de MATrix LABoratory, «laboratorio de matrices»), es un sistema de cómputo numérico que ofrece un entorno de desarrollo integrado (IDE) con un lenguaje de programación propio (lenguaje M).

La optimización topológica consiste en utilizar un software concreto para «eliminar» el material que no posee los soportes. Entre los programas más conocidos se encuentran las soluciones Ansys Discovery, Tosca de Dassault Systèmes, Within Labs de Autodesk, Inspire de SolidThinking, Netfabb y Simufact Additive.

Con la optimización topológica, las cargas mecánicas son los datos de entrada que permitirán al software proponer una nueva geometría de la pieza. Así, en principio hay menos iteraciones, lo que reduce considerablemente los tiempos de diseño y fabricación.

La optimización topológica comienza con la creación de un modelo 3D en la fase de borrador, en el que se aplicaran las diferentes cargas o fuerzas para la pieza. Después, el software se encarga de calcular todas las tensiones aplicadas.9

En este nivel, se puede realizar un corte de la pieza con el fin de retirar las partes no sometidas a las fuerzas. La geometría final, que cumple con los requisitos mecánicos y de diseño, se puede obtener finalmente después de alisar la pieza. De esta forma, la

optimización topológica responde a la necesidad de reducción de masa además del aumento de la resistencia mecánica de la pieza.

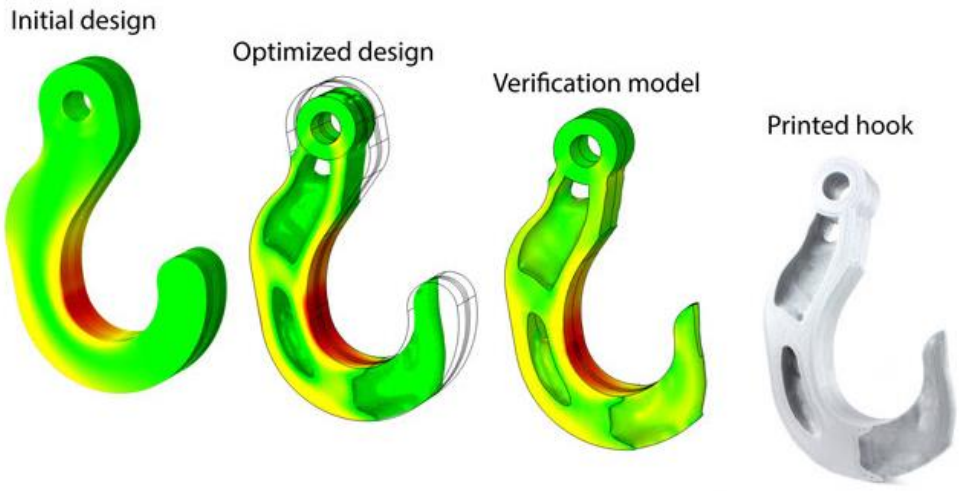


Figura 1. Optimización topológica.

3. Contenido

1) Geometría

El código presentado, se trata de la *optimización de tipología de 99 líneas* escrito en Matlab, destinado especialmente al aprendizaje. Dentro de la ingeniería existen distintos tipos de problemas como la configuración geométrica optima de un cuerpo que llegue a satisfacer tanto las restricciones como las condiciones del contorno de un problema. Asimismo, para poder dar solución a este problema podemos establecer dos tipos de soluciones, abordarlo como un problema de optimalización de forma o de optimalización de topología.

La optimalización de topología consiste básicamente en la *utilización de un software concreto para así "eliminar" el material que no posee los soportes*. La optimalización de topología comienza con lo que es la creación de un modelo 3D con un borrador, en donde se aplicarán las distintas cargas o fuerzas que ejercerán sobre la pieza ya con esto el software se encargara de calcular todas las tensiones aplicadas. Y ya la geometría final, si cumple con los requisitos mecánicos y el diseño, podemos obtenerla al alisar la pieza.

Para la implementación en ***Matlab***, se realizan en el código simplificaciones. Se supone que el dominio de diseño es rectangular y discretizada por elementos cuadrados finitos. De esta manera, la numeración de elementos y nodos es simple.

La ***relación de aspecto*** de la estructura está dada por la relación de elementos en la dirección horizontal (***nelx***) y vertical (***nely***).

El programa en Matlab, es construido como un código de optimización de topología estándar. Lo que es el programa principal, se llama desde el editor de Matlab con la siguiente línea:

top(nelx,nely,volfrac,penal,rmin)

En donde ***nelx*** y ***nely*** son el número de elementos en el direcciones horizontal y vertical, respectivamente, ***volfrac*** es la fracción de volumen, penal es el poder de penalización y ***rmin*** es el tamaño del filtro (dividido por el tamaño del elemento). Otras variables, así como las condiciones de contorno se definen en el propio código de Matlab y se puede editar si es necesario.

El ***programa principal*** (líneas 1 a 36) comienza distribuyendo el uniformemente en el dominio del diseño (línea 4). Después de algunas otras inicializaciones, 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 el material sólido es el mismo para todos los elementos, el elemento subrutina matriz de rigidez ent se llama sólo una vez (línea 14). Después de esto, un ciclo sobre todos los elementos (líneas 16–24) determina función objetivo y sensibilidades (4).

Las variables *n1* y *n2* denotan arriba a la izquierda y a la derecha números de nodo de elementos en números de nodo globales y son utilizado para extraer el vector de desplazamiento del elemento *Ue* de el vector de desplazamiento global *U*.

El análisis de sensibilidad va seguido de una llamada al filtro de independencia de malla (línea 26) y al optimizador de Criterios de Optimalidad (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).

El ciclo principal se termina si el cambio en las variables de diseño (cambio determinado en línea 30) es inferior al 1 por ciento. De lo contrario, los pasos anteriores son repetido.

La programación tiene como objetivo diseñar utilizando los distintos comandos que tiene Matlab para diseñar nuestro objeto a analizar que en este caso es una viga, muchas veces nos topamos con el clásico problema de optimización en la ingeniería.

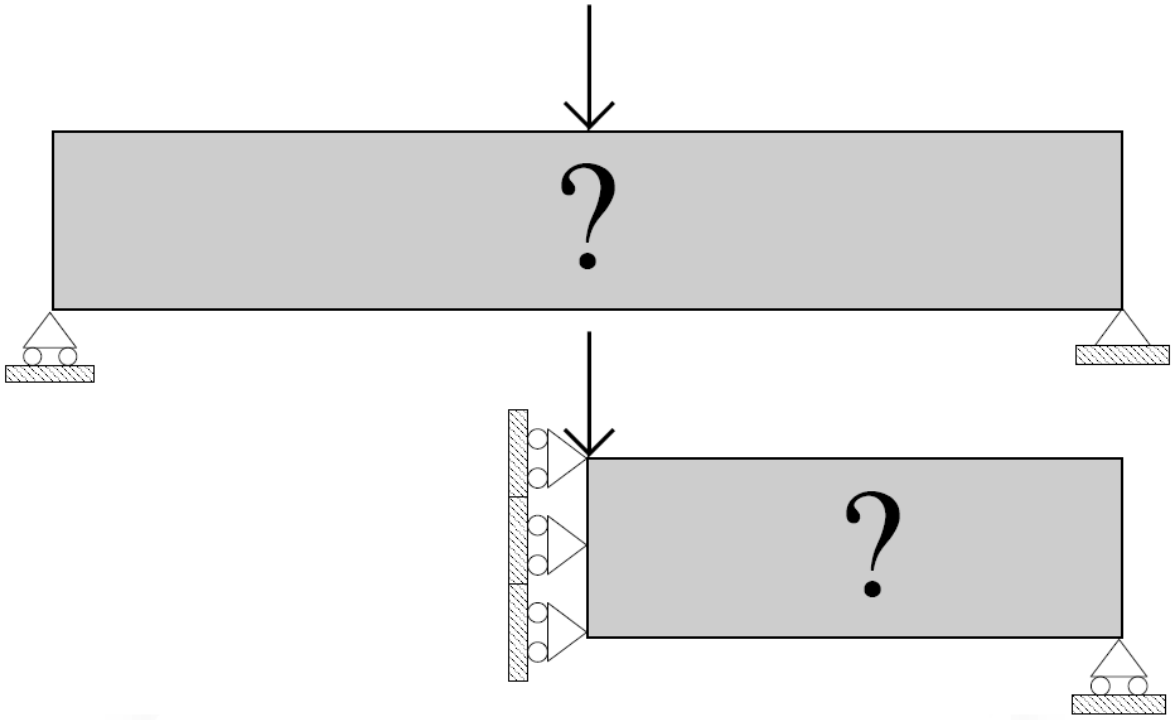


Figura 2. Dominio de diseño con condiciones de simetría.

Figura 3. Referencia de topología resultante de la viga optimizada.

2) Estado del arte

2.1 Optimización estructural

La optimización de productos y procesos busca la minimización o maximización de una o varias variables sujetas a restricciones. Es así, que un proceso de optimización se puede expresar como:

$$\begin{aligned} \min_{\mathbf{x}} \quad & f(\mathbf{x}) \\ s.a. \quad & \mathrm{g}(\mathbf{x}) \leq 0 \\ & \mathrm{h}(\mathbf{x}) = 0 \end{aligned}$$

Por tanto, la optimización estructural se utiliza para diseñar componentes con altas calidades (óptimas) de resistencia, mínimo volumen y bajos costos.

2.2 Algoritmos de optimización

Para la solución de problemas de optimización estructural se pueden observar 4 tipos principales de métodos: Algoritmos de programación no lineal, métodos de optimización metaheurística, métodos de optimización robusta y de confiabilidad y métodos heurísticos especiales (Saitou, 2005).

Los algoritmos de programación no lineal buscan resolver estos problemas de forma tal, que permitan obtener una solución óptima al problema.

Los métodos de optimización metaheurística imitan fenómenos naturales y han permitido el análisis de estructuras complejas. Este tipo de algoritmos permiten a diferencia de los métodos de programación matemática, la búsqueda de óptimos globales en problemas de optimización no lineal.

Los métodos de optimización robusta y de confiabilidad, permiten tener en cuenta las incertidumbres que se presentan en el diseño. Los métodos de optimización robusta permiten que la función objetivo de un problema sea insensible a variaciones de las variables de diseño.

Los métodos heurísticos permiten obtener el escenario óptimo de una estructura usando reglas locales simples, lo cual genera bajo costo computacional. Por lo anterior han sido empleados en la solución de problemas de diseño estructural de gran complejidad y de gran escala

2.3 Optimización de ensambles

Las investigaciones relacionadas con el diseño de ensambles o estructuras multicuerpo se dividen en tres áreas principales que son: Optimización de estructuras con uniones flexibles, optimización en la distribución óptima, y síntesis del ensamble basado en la descomposición.

Optimización de estructuras con uniones flexibles: optimiza los costos de producción con el análisis de las conexiones o uniones semirrígidas entre vigas y columnas de estructuras civiles.

El método de síntesis de ensambles basado en la descomposición: consiste en un método sistemático donde la geometría total de un producto es descompuesta en componentes y uniones de acuerdo con los requerimientos específicos de diseño y producción.

2.4 Análisis por elementos finitos para uniones flexibles

El análisis por elementos finitos a una estructura permite la verificación de la deformación de los elementos de la estructura (componentes y uniones) de acuerdo con las condiciones

iníciales impuestas en el problema. En la mayoría de los análisis de estructuras, las uniones son asumidas como conexiones articuladas o conexiones rígidas, las cuales conllevan respectivamente al análisis de estructuras como armaduras o marcos lo cual son empleadas en ingeniería para simplificar el procedimiento de análisis.

Para el análisis de estructuras que cumplan con estas especificaciones, la rigidez y resistencia de deben ser verificadas, e incorporadas en los programas computaciones. Para la determinación de los desplazamientos generados en la estructura bidimensional, incorporando la flexibilidad de las uniones, cada uno de los elementos que la constituyen se definen como elementos tipo viga con conexiones semirrígidas en sus extremos.

2.5 Diseño para la manufacturabilidad y ensamblabilidad

Dentro de las diferentes guías de diseño dadas por estas metodologías para el desarrollo de estructuras de gran tamaño y complejidad se encuentran:

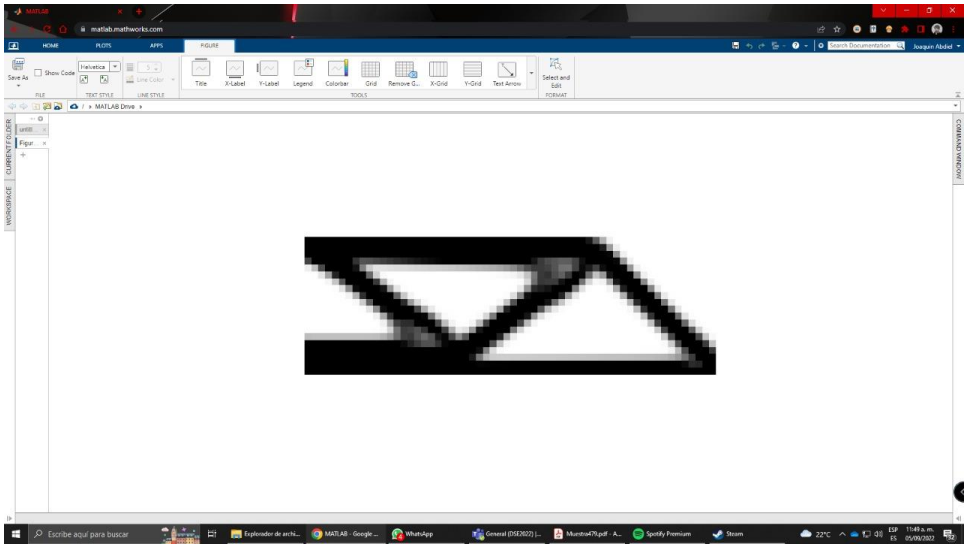
- Maximizar la estandarización de parámetros de diseño, tales como: material, componentes, modularización de conjuntos, herramientas, uniones, entre otros.
- Seleccionar soluciones que permitan la simplificación de la manufactura (simplificación de formas, reducción de operaciones, aplicación de operaciones de menor costo y tiempo de ejecución).
- Eliminar y reducir procesos de unión entre componentes, los cuales puedan inducir errores de fabricación.
- Seleccionar soluciones que permitan el mejoramiento de la uniformidad y el paralelismo.
- Y minimizar el número de recursos empleados.

3) Procedimiento de la programación

Lo primero que hicimos fue abrir el Matlab y transcribir el código que se había mostrado, una vez que pusimos el código lo ejecutamos, y nos arrojó la figura que queremos analizar.

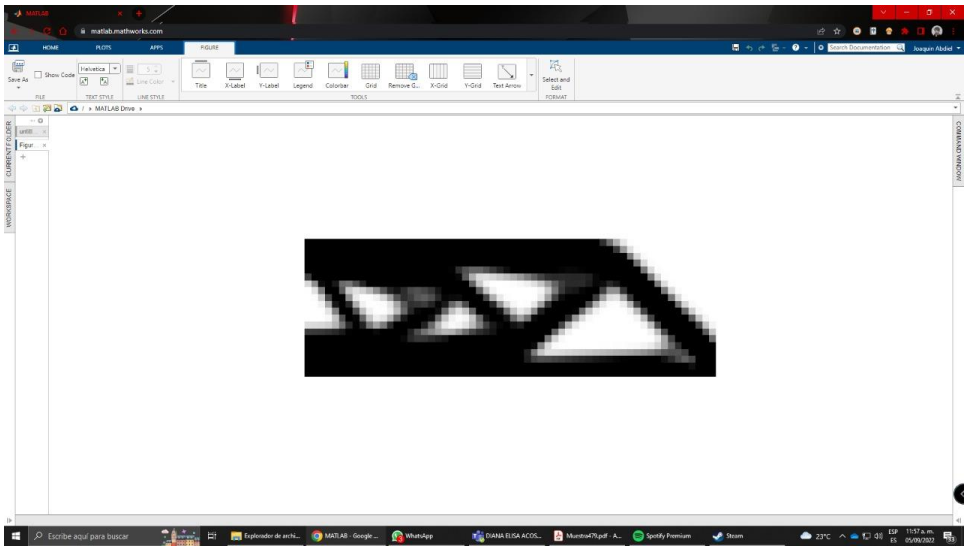
```
top.m
1  %%%% A 99 LINE TOPOLOGY OPTIMIZATION CODE BY OLESIGMU
2  function top(nelx,nely,volfrac,penal,rmin);
3  % INITIALIZE
4  x(1:nely,1:nelx) = volfrac;
5  loop = 0;
6  change = 1.;
7  % START ITERATION
8  while change > 0.01
9  loop = loop + 1;
10 xold = x;
11 % FE-ANALYSIS
12 [U]=FE(nelx,nely,x,penal);
13 % OBJECTIVE FUNCTION AND SENSITIVITY ANALYSIS
14 [KE] = lk;
15 c = 0.;
16 for ely = 1:nely
17 for elx = 1:nelx
18 n1 = (nely+1)*(elx-1)+ely;
19
```

Comenzamos a analizarlo para poder optimizar lo mejor posible nuestra estructura, cambiamos algunos valores del programa y al volverlo a ejecutar nos apareció la misma geometría con algunas diferencias:



Como se puede observar en la imagen anterior, se comienzan a notar unos huecos que se forman dónde empieza la unión de las vigas, así que seguimos cambiando algunos parámetros, los cambiamos y volvimos a ejecutar el programa de nuevo:

Como se puede observar en la imagen anterior se ve cada vez menos visible, por lo que esos parámetros no son óptimos así que seguimos cambiando los valores de los parámetros:



Como se puede observar en la imagen anterior, la estructura que tiene nuestra figura no se ve bien definida, todo se ve muy amontonado, por lo tanto, no es muy optimo.

```
1 %%% A 99 LINE TOPOLOGY OPTIMIZATION CODE BY OLE
SIGMUND, OCTOBER 1999 %%%
2 function top(nelx,nely,volfrac,penal,rmin);
3 % INITIALIZE
4 x(1:nely,1:nelx) = volfrac;
5 loop = 0;
6 change = 1.;
```

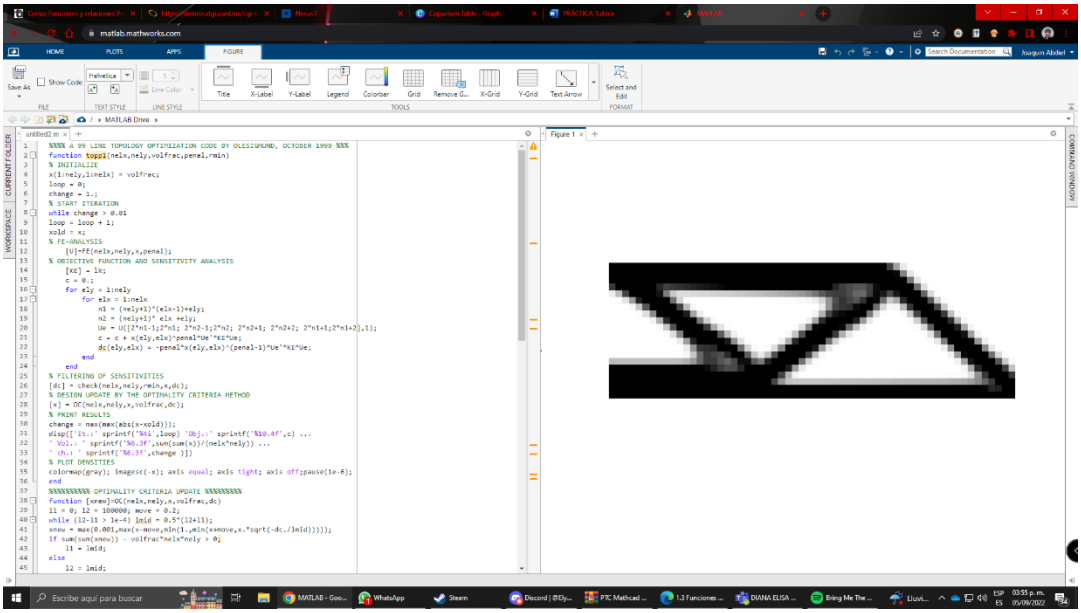
```
% START ITERATION
while change > 0.01
  loop = loop + 1;
  xold = x;
%FE-ANALYSIS
  [U]= FE(nelx,nely,x,penal);
%OBJECTIVE FUNCTION AND SENSITIVITY ANALYSIS
  [KE] = lk;
  c = 0.;
  for ely = 1:nely
     for elx = 1:nelx
         n1 = (nely+1)*(elx-1)+ely;
         n2 = (nely+1)*elx+ely;
         Ue = U([2*n1-1;2*n1;2*n2-1;2*n2;2*n2+1;
2*n2+2;2*n1+1;2*n1+2],1);
         c = c + x(ely,elx)^penal*Ue'*KE*Ue;
         dc(ely,elx) = -penal*x(ely,elx)^(penal-1)*
Ue'*KE*Ue;
     end
end
% FILTERING OF SENSITIVITIES
  [dc] = check(nelx,nely,rmin,x,dc);
% DESIGN UPDATE BY THE OPTIMALITY CRITERIA METHOD
  [x] = OC(nelx,nely,x,volfrac,dc);
% PRINT RESULTS
  change = max(max(abs(x-xold)));
  disp(['It.:'sprintf('%4i',loop) 'Obj.:'
sprintf('%10.4f',c)...
       'Vol.:'sprintf('%6.3f',sum(sum(x))/ (nelx*nely))…
       'ch.:'sprintf('%6.3f',change)])
%PLOT DENSITIES
  colormap(gray);imagesc(-x);axis equal;axis tight;axis
off;pause(1e-6);
  end
%%%%%%%%%% OPTIMALITY CRITERIA UPDATE %%%%%%%%%
function[xnew]=OC(nelx,nely,x,volfrac,dc)
l1 = 0; l2 = 100000; move = 0.2;
while (l2-l1 > 1e-4)
lmid = 0.5 * (l2+l1);
xnew = max(0.001,max(x-move,min(1.,min(x+move,x.*sqrt(-
dc./lmid)))));
if sum(sum(xnew))-volfrac*nelx*nely > 0;
    l1 = lmid;
else
    l2 = lmid;
end
end
```

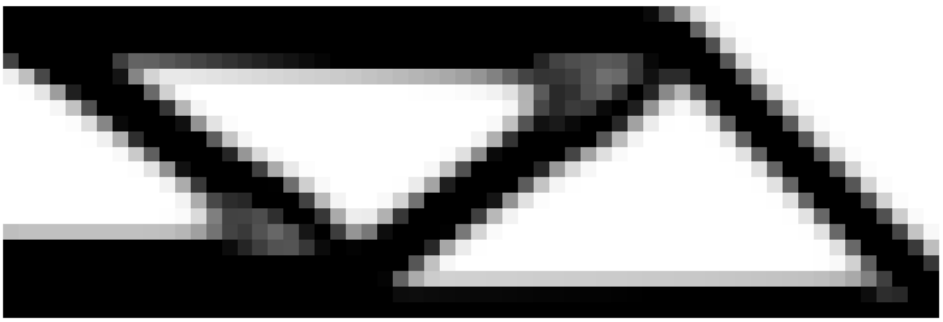
```
%%%%%%%%%% MESH-INDEPENDENCY FILTER %%%%%%%%%%
function [dcn] = check(nelx,nely,rmin,x,dc)
dcn = zeros(nely,nelx);
for i = 1:nelx
  for j = 1:nely
    sum = 0.0;
    for k = max(i-round(rmin),1):
min(i+round(rmin),nelx)
        for l = max(j-round(rmin),1):
min(j+round(rmin),nely)
            fac = rmin-sqrt((i-k)^2+(j-l)^2);
            sum = sum+max(0,fac);
            dcn(j,i) = dcn(j,i)+max(0,fac)*x(l,k)
*dc(l,k);
        end
      end
    dcn(j,i) = dcn(j,i)/(x(j,i)*sum);
  end
end
%%%%%%%%%% FE-ANALYSIS %%%%%%%%%%%
function[U] = FE(nelx,nely,x,penal)
[KE] = lk;
K = sparse(2*(nelx+1)*(nely+1),2*(nelx+1)* (nely+1));
F = sparse(2*(nely+1)*(nelx+1),1);U=
sparse(2*(nely+1)*(nelx+1),1);
for ely = 1:nely
for elx = 1:nelx
    n1 = (nely+1)*(elx-1)+ely;
    n2 = (nely+1)*elx+ely;
    edof = [2*n1-1;2*n1;2*n2-1;2*n2;2*n2+1;
2*n2+2;2*n1+1;2*n1+2];
    K(edof,edof) = K(edof,edof)+ x(ely,elx)^penal*KE;
    end
end
% DEFINE LOADS AND SUPPORTS (HALF MBB-BEAM)
F(2,1) = -1;
fixeddofs = union([1:2:2*(nely+1)],
[2*(nelx+1)*(nely+1)]);
alldofs = [1:2*(nely+1)*(nelx+1)];
freedofs = setdiff(alldofs,fixeddofs);
% SOLVING
U(freedofs,:) = K(freedofs,freedofs) \ F(freedofs,:);
U(fixeddofs,:)= 0;
%%%%%%%%%% ELEMENT STIFFNESS MATRIX %%%%%%%
function [KE]=lk
E = 1.;
un = 0.3;
```

```
90 k=[ 1/2-nu/6 1/8+nu/8 -1/4-nu/12 -1/8+3*nu/8 …
91 -1/4+nu/12 -1/8-nu/8 nu/6 1/8-3*nu/8];
92 KE = E/(1-nu^2)*[ k(1) k(2) k(3) k(4) k(5) k(6) k(7)
k(8)
93 k(2) k(1) k(8) k(7) k(6) k(5) k(4) k(3)
94 k(3) k(8) k(1) k(6) k(7) k(4) k(5) k(2)
95 k(4) k(7) k(6) k(1) k(8) k(3) k(2) k(5)
96 k(5) k(6) k(7) k(8) k(1) k(2) k(3) k(4)
97 k(6) k(5) k(4) k(3) k(2) k(1) k(8) k(7)
98 k(7) k(4) k(5) k(2) k(3) k(8) k(1) k(6)
99 k(8) k(3) k(2) k(5) k(4) k(7) k(6) k(1)];
```

top(60,20,0.5,3.0,1.5)

4) Desarrollo de la programación en sus diferentes vistas.





5) Conclusiones

Diana Elisa Acosta Rodríguez:

La elaboración de la práctica ha sido muy útil en el sentido de la introducción a los conceptos fundamentales, que involucra la optimización topológica, la cual, es muy eficaz como método de reducción de material que no tiene algún soporte, y si permite aumentar su resistencia. Todo, esto pudimos llevarlo a cabo, cumpliendo con los requisitos de la pieza, obteniendo así mismo la geometría indicada. Para ello, se realizó un modelo tridimensional de la pieza deseada (en este caso, una viga), con ayuda del código generado en el software de Matlab. En el código, se implementaron diferentes procesos, como el uso de filtros y elementos finitos. También, se optó por manipular los parámetros, pero esto generaba deformaciones e impedían el buen resultado de la pieza final. Sin embargo, analizando e investigando, se logró llegar a elaborar la pieza indicada, de acuerdo a la optimización topológica, en 99 líneas de código.

Joaquín Abdiel Coronado Vázquez:

Al concluir con la práctica pudimos visualizar y aprender sobre el diseño a través del código en Matlab todo esto debido a los análisis por elementos finitos, también aprendimos sobre lo que es el estado del arte que esto se basa en varias estructuras, varios puntos a estudiar, también vimos sobre la optimización topológica, la cual pues consiste en eliminar el material que no poseen los soportes dentro de una pieza. con eso se pueden llegar a ver las cargas mecánicas que son permitidas dentro de la pieza a través del software y también se puede ayudar a ver sobre la creación de un modelo 3D. siendo así vimos un ejemplo de cómo desarrollarla a través del código de Matlab y también concluimos que al modificarle los parámetros a la pieza puede haber una deformación dentro de la misma ya sea

aumentada o disminuida y esto puede afectar a lo que sería nuestra construcción final de la misma.

Brenda Giselle Hinojosa:

Para realizar esta práctica fue necesario hacer una investigación sobre la optimización topológica siendo este concepto algo nuevo que no se había escuchado, por lo tanto, primero se entró en contexto, viendo de que trataba esto y para que era necesario; al momento de aplicar el código en Matlab se pudo comprender mejor como era que funcionaba, este código fue sacado de internet con ayuda de una página proporcionada por el instructor y consistía en un código de 99 líneas. La optimización topológica nos permite aligerar las estructuras manteniendo las funcionalidades mecánicas del componente objetivo y así mismo aumentando su resistencia. Se estuvieron modificando los parámetros en Matlab para llegar a los que fueran más óptimos y así obtener la mejor estructura posible; al modificar estos datos se obtenían deformaciones y con esto podemos asegurar que nuestra estructura deja de ser segura para su uso.

Gloria Rosalía Domínguez Azueta:

El resultado del algoritmo de optimización topológica con la ayuda de Matlab resulto como se esperaba, este código de 99 líneas fue implementado para optimizar, filtrar y utilizar código de elementos finitos manteniendo un número total de líneas por debajo de 100. Al inicio no comprendía tanto el concepto de optimización topológica, una vez realizada la investigación comprendí que está permite la eliminación o reducción de material que no posee algún soporte o estructura para aumentar la resistencia de la pieza, aplicando inicialmente las cargas a la estructura. A pesar de los inconvenientes en el código, al final se optimizo la estructura en Matlab como se esperaba, logramos determinar la configuración geométrica óptima del cuerpo geométrico, obteniendo los parámetros muy cerca a lo que se buscaba, pues nos percatamos que, al modificar los parámetros topológicos, ya sea aumentando o disminuyendo, permitían que la estructura tuviera una distribución de densidad distinta, afectando el resultado, aun así, se logró llegar a la distribución de la imagen tomada como referencia.

Andrea Anette Celestino Castillo:

Para terminar con la practica podemos mencionar que investigamos un poco a cerca de lo que es la optimalización topológica, y varios tipos más de esta, así como es que esta se implemente y su uso, quedando así un poco más claro el tema. Asimismo, también vimos lo que es un ejemplo de esta, desarrollándola en un software que en nuestro caso fue Matlab en donde realizamos un código para poder comprender mejor la práctica, aunque tuvimos un poco de problemas en la realización del código, se pudieron resolver y pudimos observar la implementación y el desarrollo del programa.

Ana Paulina Pérez Salazar:

En la elaboración de esta práctica pudimos analizar y comprender sobre la optimización topológica que para mí era un concepto nuevo, por medio de la investigación se pudo enterarnos a detalle de lo que teníamos que desarrollar y sus conceptos básicos, como se mencionó antes la optimización topológica consiste en utilizar un software concreto para eliminar el material que no posee soporte y hay infinidad de ellos para solucionar dicho problema, esto nos ayuda a como su nombre lo dice optimizar el tiempo de diseño y fabricación.

Javier Cisneros Saldivar:

La optimización es un aspecto muy importante en cualquier proceso o diseño realizado ya que nos permite ahorrar ya sea dinero, tiempo, esfuerzo, entre otros aspectos, fue muy interesante ver como a medida que nosotros cambiábamos los parámetros las estructuras se iba modificando visualmente hasta casi ver solamente los puros pixeles.

Bibliografía

- Carlos Alberto Meza Valencia. (3 octubre 2012). Optimización topológica en el diseño de elementos estructurales mecánicos. 5 septiembre 2022, De Universidad Autónoma de occidente Base de datos.
- Carlos Julio Camacho Flores. (febrero 2011). Optimización topológica estructural de ensambles. 5 septiembre 2022, de Universidad Nacional de Colombia Base de datos.
- 99 Line Topology Optimization Code – O. Sigmud, Department of Solid Mechanics, Building 404, Technical University of Denmark, DK-2800 Lyngby, Denmark.