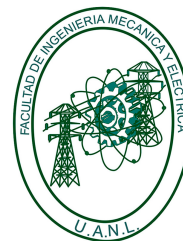




Universidad Autónoma de Nuevo León

Facultad de Ingeniería Mecánica y Eléctrica



Laboratorio de Biomecánica

Práctica 1:

Catedrático: M.C. Yadira Moreno Vera

Equipo

Datos del equipo:

<u>Nombre</u>	<u>Matrícula</u>	<u>Carrera</u>
Luis Alberto Pinzon Garcia	1640007	IMC
Luis Carlos Gómez Espinoza	1926227	IMC
César Mauricio Alvarez Olguín	1910330	IMC
Mauro Alberto Hernández Saldaña	1925229	IMC
Mauricio Martinez Tovar	1862673	IMC
Eduardo Rodriguez Montalvo	1991965	IMC

Fecha de entrega: **06 de Septiembre de 2022**

Hora: V2 Brigada: 214

Semestre: Agosto - Diciembre 2022

Objetivo:

El estudiante conocerá 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.

1) Nombre y definición de la programación mencionar un ejemplo de forma de la GEOMETRÍA.

Un problema clásico de la ingeniería consiste en determinar la configuración geométrica óptima de un cuerpo que minimice o maximice una cierta función objetivo, al mismo tiempo que satisface las restricciones o condiciones de contorno del problema. La solución de este problema puede ser planteada utilizando dos estrategias: como un problema de optimización de forma o de optimización de la topología.

DEFINICIÓN DE LA PROGRAMACIÓN

La optimización topológica es conocida como una técnica que se encuentra englobada dentro del campo de análisis estructural, éste se basa en el análisis mecánico de un componente o estructura, su objetivo es el aligeramiento estructural manteniendo las funcionalidades mecánicas de componente. En este tipo de optimización se ofrece un nuevo concepto de diseño estructural enfocado a aquellas aplicaciones donde el peso del componente es crucial.

Gracias a los nuevos métodos computacionales, es posible llevar la optimización a un nivel más complejo de análisis a nivel estático, dinámico, plástico, modal o de impacto, entre otros, los cuales pueden considerarse durante el proceso de optimización.

La optimización topológica, por tanto, 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.

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 (una presión sobre las lengüetas de sujeción, por ejemplo). Después, el software se encarga de calcular todas las tensiones aplicadas.

La geometría final, que cumple con los requisitos mecánicos y de diseño, se puede obtener finalmente después de alisar la pieza.

En el artículo consultado, se presenta una implementación compacta en matlab de un código de optimización de topología de 99 líneas, el objetivo de éste es la minimización del cumplimiento de estructuras cargadas estáticamente.

El número total de líneas de entrada de matlab es 99, aquí se incluye el optimizador y la subrutina de elementos finitos. asimismo las 99 líneas son divididas en:

- 36 líneas para el programa principal.
- 12 líneas para el optimizador basado en criterios de optimización.
- 16 líneas para un filtro de independencia de malla.
- 35 líneas para el código de elemento finito.

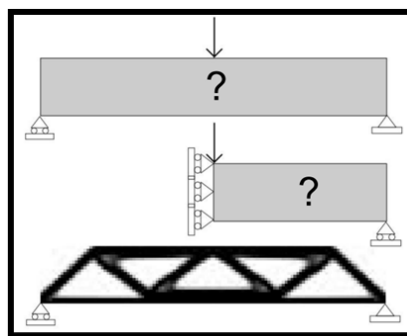
El código se utiliza para optimización estructural, en casos múltiples de carga, esquemas alternativos de independencia de malla, áreas pasivas, etc.

FORMA DE LA GEOMETRÍA (EJEMPLO)

Un ejemplo de la implementación del código de matlab sobre la optimización topológica, es el siguiente:

El código de Matlab se construye como un código de optimización de topología estándar. El programa principal es llamado desde el indicador de Matlab por la línea top (nelx, nely, volfrac, penal, rmin) donde nelx y nely son el número de elementos en las 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).

En la siguiente figura se muestra la distribución de densidad resultante obtenida por el código dado en el Apéndice llamado con la línea de entrada.



En la figura se tiene la optimización topológica de MBB-beam.

- **Parte Superior:** Se muestra el dominio del diseño,
- **Parte de Enmedio:** Se muestra medio dominio de diseño con simetría y condiciones de contorno
- **Parte Inferior:** Se muestra la topología resultante en ambas mitades.

Las condiciones de contorno por defecto corresponden a la mitad de la “MBB-Beam”. La carga se aplica verticalmente en la esquina superior izquierda y hay un límite simétrico condiciones a lo largo del borde izquierdo y la estructura se apoya horizontalmente en la esquina inferior derecha.

2) Estado del Arte.

A 99 Line Topology Optimization Code Written in Matlab

Resumen del Marco Teórico

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.

La optimización topológica es una técnica englobada dentro del campo de análisis estructural. Se basa en el análisis mecánico de un componente o estructura. Su principal objetivo es el aligeramiento estructural manteniendo las funcionalidades mecánicas del componente objetivo. Gracias a los nuevos métodos computacionales, es posible llevar la optimización a un nivel más complejo.

El artículo que se consultó, presenta una implementación compacta en matlab, en base a un código de optimización de topología de 99 líneas, donde su objetivo es la minimización del cumplimiento de estructuras cargadas estáticamente.

El código de Matlab se construye como un código de optimización de topología estándar, el programa principal es llamado desde el indicador de Matlab por la línea top (nelx, nely, volfrac, penal, rmin).

Las 99 líneas están divididas en 36 líneas para el programa principal, 12 de ellas, son para el optimizador basado en criterios de optimización, 16, para un filtro de dependencia de malla y 35, para el código de elementos finitos.

Excluyendo las líneas de comentario, así como las asociadas a la salida y al análisis de elementos finitos, se concluye que únicamente se necesitan 49 líneas de entrada de Matlab, para poder dar resolución a un problema de optimización topológica.

PALABRAS CLAVE

- Líneas
- Optimización Topológica
- Código Matlab
- top

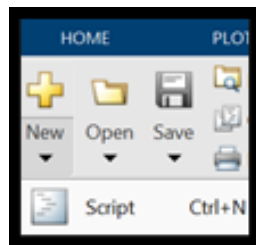
3) Procedimiento de la programación.

Como se comentó anteriormente, el código implementado en Matlab está compuesto como un código de optimización topológica estándar, donde para llevar a cabo dicho código, hay que seguir la siguiente metodología:

- 1) Inicializar el programa de Matlab.



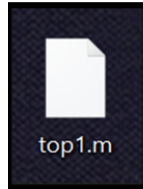
- 2) Crear un nuevo archivo haciendo clic en New, para posteriormente, seleccionar la opción de Script.



- 3) Implementar el código de optimización de topología de 99 líneas en el espacio que se generó al realizar el paso anterior.

```
1  %%% A 99 LINE TOPOLOGY OPTIMIZATION CODE BY OLE SIGMUND, JANUARY 2000 %%%
2  %%% CODE MODIFIED FOR INCREASED SPEED, September 2002, BY OLE SIGMUND %%%
3  function top1(nelx,nely,volfrac,penal,rmin);
4  nelx=60;
5  nely=20;
6  volfrac=0.5;
7  penal=3.0;
8  rmin=1.5;
9  % INITIALIZE
10 x(1:nely,1:nelx) = volfrac;
11 loop = 0;
12 change = 1.;
13 % START ITERATION
14 while change > 0.01
15     loop = loop + 1;
16     xold = x;
```

- 4) Una vez escrito todo el código, se procede a guardarlo, prestando atención en la ubicación donde se va a guardar el script, así como en el nombre que se le va a asignar al archivo. En este caso, se le guardará con el nombre de top1, donde su extensión por default será de .m.



- 5) Finalmente, se correrá el programa haciendo uso del botón de “Run”, en caso de existir algún error, analizar el código y solucionarlo, para que el programa pueda correr sin dificultad alguna.



4) Implementación o desarrollo de la programación en sus diferentes vistas.

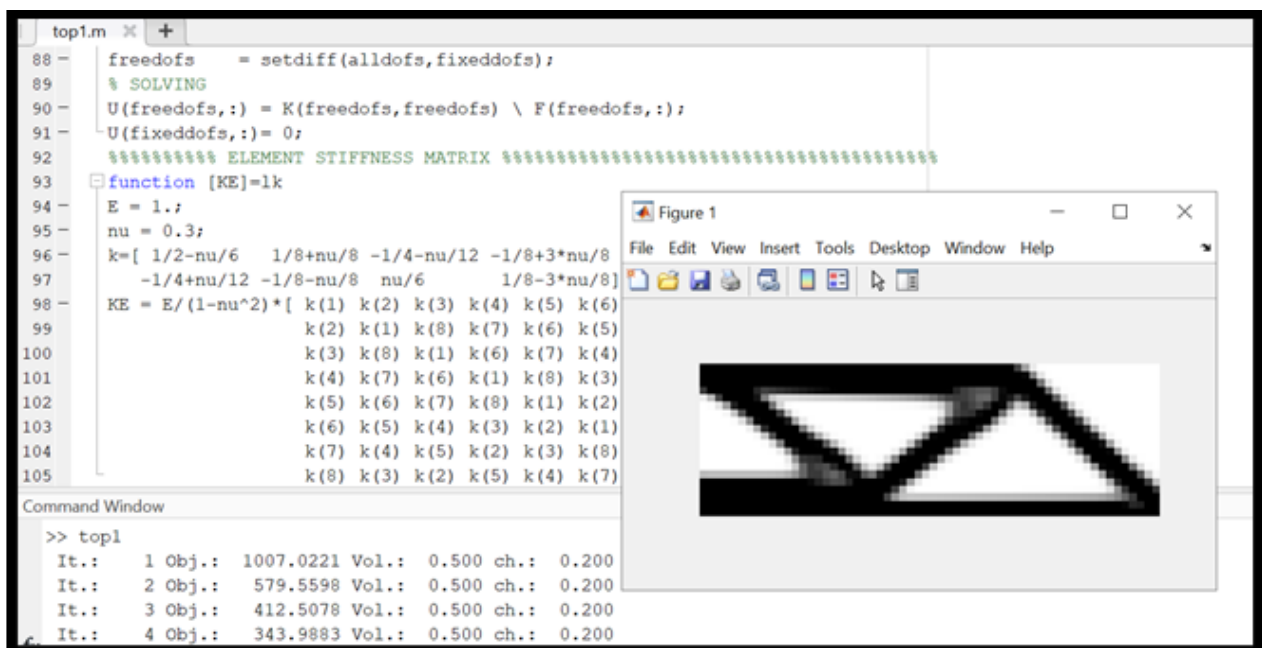
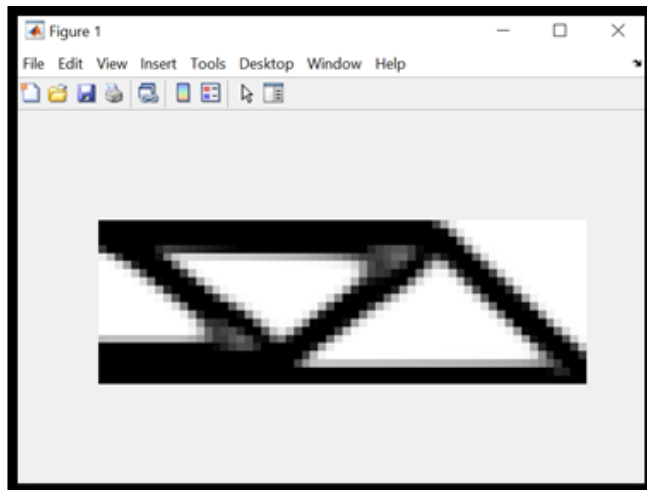
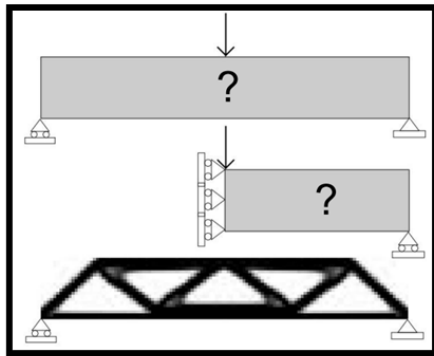
Código de 99 líneas

```
top1.m  X  +
1  |%%% A 99 LINE TOPOLOGY OPTIMIZATION CODE BY OLE SIGMUND, JANUARY 2000 %%%
2  |%%% CODE MODIFIED FOR INCREASED SPEED, September 2002, BY OLE SIGMUND %%%
3  |function top1(nelx,nely,volfrac,penal,rmin);
4  |    nelx=60;
5  |    nely=20;
6  |    volfrac=0.5;
7  |    penal=3.0;
8  |    rmin=1.5;
9  |    % INITIALIZE
10 |    x(1:nely,1:nelx) = volfrac;
11 |    loop = 0;
12 |    change = 1.;
13 |    % START ITERATION
14 |    while change > 0.01
15 |        loop = loop + 1;
16 |        xold = x;
17 |
18 |    % FE-ANALYSIS
19 |    [U]=FE(nelx,nely,x,penal);
20 |    % OBJECTIVE FUNCTION AND SENSITIVITY ANALYSIS
21 |    [KE] = lk;
22 |    c = 0.;
23 |    for ely = 1:nely
24 |        for elx = 1:nelx
25 |            n1 = (nely+1)*(elx-1)+ely;
26 |            n2 = (nely+1)* elx +ely;
27 |            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);
28 |            c = c + x(ely,elx)^penal*Ue'*KE*Ue;
29 |            dc(ely,elx) = -penal*x(ely,elx)^(penal-1)*Ue'*KE*Ue;
30 |        end
31 |    end
32 |    % FILTERING OF SENSITIVITIES
33 |    [dc] = check(nelx,nely,rmin,x,dc);
```

```

33 % DESIGN UPDATE BY THE OPTIMALITY CRITERIA METHOD
34 [x] = OC(nelx,nely,x,volfrac,dc);
35 % PRINT RESULTS
36 change = max(max(abs(x-xold)));
37 disp([' It.: ' sprintf('%4i',loop) ' Obj.: ' sprintf('%10.4f',c) ...
38 ' Vol.: ' sprintf('%6.3f',sum(sum(x))/(nelx*nely)) ...
39 ' ch.: ' sprintf('%6.3f',change )])
40 % PLOT DENSITIES
41 colormap(gray); imagesc(-x); axis equal; axis tight; axis off; pause(1e-6);
42 end
43 %***** OPTIMALITY CRITERIA UPDATE *****
44 function [xnew]=OC(nelx,nely,x,volfrac,dc)
45 l1 = 0; l2 = 100000; move = 0.2;
46 while (l2-l1 > 1e-4)
47     lmid = 0.5*(l2+l1);
48     xnew = max(0.001,max(x-move,min(1.,min(x+move,x.*sqrt(-dc./lmid)))));
49     if sum(sum(xnew)) - volfrac*nelx*nely > 0;
50         l1 = lmid;
51     else
52         l2 = lmid;
53     end
54 end
55 %***** MESH-INDEPENDENCY FILTER *****
56 function [dcn]=check(nelx,nely,rmin,x,dc)
57 dcn=zeros(nely,nelx);
58 for i = 1:nelx
59     for j = 1:nely
60         sum=0.0;
61         for k = max(i-floor(rmin),1):min(i+floor(rmin),nelx)
62             for l = max(j-floor(rmin),1):min(j+floor(rmin),nely)
63                 fac = rmin-sqrt((i-k)^2+(j-l)^2);
64                 sum = sum+max(0,fac);
65                 dcn(j,i) = dcn(j,i) + max(0,fac)*x(l,k)*dc(l,k);
66             end
67         end
68         dcn(j,i) = dcn(j,i)/(x(j,i)*sum);
69     end
70 end
71 %***** FE-ANALYSIS *****
72 function [U]=FE(nelx,nely,x,penal)
73 [KE] = lk;
74 K = sparse(2*(nelx+1)*(nely+1), 2*(nelx+1)*(nely+1));
75 F = sparse(2*(nely+1)*(nelx+1),1); U = zeros(2*(nely+1)*(nelx+1),1);
76 for elx = 1:nelx
77     for ely = 1:nely
78         n1 = (nely+1)*(elx-1)+ely;
79         n2 = (nely+1)* elx +ely;
80         edof = [2*n1-1; 2*n1; 2*n2-1; 2*n2; 2*n2+1; 2*n2+2; 2*n1+1; 2*n1+2];
81         K(edof,edof) = K(edof,edof) + x(ely,elx)^penal*KE;
82     end
83 end
84 % DEFINE LOADS AND SUPPORTS (HALF MBB-BEAM)
85 F(2,1) = -1;
86 fixeddofs = union([1:2*(nely+1)], [2*(nelx+1)*(nely+1)]);
87 alldofs = [1:2*(nely+1)*(nelx+1)];
88 freedofs = setdiff(alldofs,fixeddofs);
89 % SOLVING
90 U(freedofs,:) = K(freedofs,freedofs) \ F(freedofs,:);
91 U(fixeddofs,:)= 0;
92 %***** ELEMENT STIFFNESS MATRIX *****
93 function [KE]=lk
94 E = 1.;
95 nu = 0.3;
96 k=[ 1/2-nu/6 1/8+nu/8 -1/4-nu/12 -1/8+3*nu/8 ...
97 -1/4+nu/12 -1/8-nu/8 nu/6 1/8-3*nu/8];
98 KE = E/(1-nu^2)*[ k(1) k(2) k(3) k(4) k(5) k(6) k(7) k(8)
99 k(2) k(1) k(8) k(7) k(6) k(5) k(4) k(3)
100 k(3) k(8) k(1) k(6) k(7) k(4) k(5) k(2)
101 k(4) k(7) k(6) k(1) k(8) k(3) k(2) k(5)
102 k(5) k(6) k(7) k(8) k(1) k(2) k(3) k(4)
103 k(6) k(5) k(4) k(3) k(2) k(1) k(8) k(7)
104 k(7) k(4) k(5) k(2) k(3) k(8) k(1) k(6)
105 k(8) k(3) k(2) k(5) k(4) k(7) k(6) k(1)];

```



5) Conclusiones por cada autor.

☐ **Luis Carlos Gomez Espinoza 1926227**

Como conclusión en la práctica uno del laboratorio, comprendí lo que es la optimización topológica, la importancia de esta y cómo puede ser implementada por medio de programas como Matlab, donde para realizarla hay que tener en cuenta que es por medio de un código de 99 líneas, éste tiene la finalidad de lograr un aligeramiento estructural siempre manteniendo las funcionalidades mecánicas del objeto a optimizar.

☐ **César Mauricio Alvarez Olguín 1910330**

Esta primera práctica a pesar de ser bastante corto y a su vez sencilla, debido a que genere un error al momento de correr el código, pero revisando línea por línea, detectamos pequeños detalles los cuales eran los que hacían ruido en el programa, a su vez, logre adquirir un poco más de conocimientos, principalmente en el concepto de “optimización topológica”, así como lo que es el “estado del arte” de un tema en cuestión. Finalmente se llevó a cabo y comprendió el código de 99 líneas, el cual fue el objeto de estudio en esta primera práctica.

☐ **Mauricio Martinez Tovar**

En la práctica uno del laboratorio, comprendí lo que es la optimización topológica, la importancia de esta y cómo puede ser implementada por medio de programas como Matlab. Es la segunda vez que trabajo con matlab y aunque al inicio es complicado con la práctica y estudiando los códigos se logró realizar el código. Sin duda se reforzaron los conocimientos utilizando este software tan práctico.

☐ **Mauro Alberto Hernández Saldaña**

En esta primera práctica, desarrollamos e investigamos sobre lo que es la optimización topológica, para que sirve y cómo es que podemos darnos un ejemplo utilizando matlab, donde realizamos un código para poder entenderlo y ayudar a la práctica, aunque fue corta dentro de lo que cabe, se le debe dedicar su debido tiempo para cumplir con lo deseado.

☐ **Eduardo Rodriguez Montalvo**

En esta primera práctica, se desarrollaron los principios de optimización topológica, las distintas aplicaciones y visualizaciones en la vida cotidiana y cómo es que se pueden representar a través del software matlab y como es que en esta ocasión por medio del código implementado pudimos comprender mas a detalle como es que se desarrolla de manera óptima y precisa para mas adelante aplicarlo en futuras situaciones que requieran este tipo de optimizaciones.

☐ **Luis Alberto Pinzon Garcia**

En esta primera actividad del laboratorio de biomecánica se encargó de realizar un ejemplo de una forma de la geometría y nosotros la realizamos y comprendimos de la forma de la optimización topológica, se nos presentaron varios errores al momento de implementarlo en el software pero al final logramos corregirlo y comprenderlo que fue el código de 99 líneas y también se explico el estado de arte, y como final logramos como finalidad el funcionamiento del objeto de la optimización.

❖ Fuentes bibliográficas

- 99 Line Topology Optimization Code - O. Sigmund, Department of Solid Mechanics, Building 404, Technical University of Denmark, DK-2800 Lyngby, Denmark.
- Centro Avanzado de Tecnologías Aeroespaciales. (2019). Optimización Topológica. 24/08/2022, de CATEC Sitio web: <http://www.catec.aero/es/materiales-y-procesos/linea-de-investigacion-topologica-optimizacion-topologica#:~:text=La%20optimizacion%20topologica%20es%20una,funcionalidades%20mec%C3%A1nicas%20del%20componente%20objetivo>
- *Optimización Topológica*. (2019, 9 enero). Estudio de Ingeniería y Tecnología Avanzada S.L. Recuperado 3 de septiembre de 2022, de <https://eitaingenieros.com/optimizacion/>