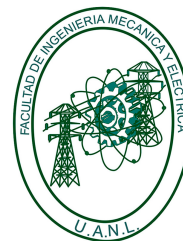




Universidad Autónoma de Nuevo León

Facultad de Ingeniería Mecánica y Eléctrica



Laboratorio de Biomecánica

Práctica 2:

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: **20 de Septiembre de 2022**

Hora: V2 Brigada: 214

Semestre: Agosto - Diciembre 2022

Objetivo:

El estudiante deberá presentar una propuesta de análisis de formas y de la programación para la ejecución de la optimización (descripción funcional) de características de trabajo específicas que presenta la(s) ventaja(s).

1) Nombre y definición de la forma GEOMETRÍA.

Una bicicleta es un vehículo de dos ruedas movido por una persona, provisto de un manubrio en la parte delantera, un asiento para el conductor y dos pedales que transmiten el movimiento de las piernas a la rueda trasera mediante una cadena y un piñón.

Las principales características que comprenden un diseño óptimo de una buena bicicleta, son las siguientes:

1.- Cuadro Rígido

Una bicicleta completamente cargada (con alforjas traseras y delanteras) se siente muy diferente que cuando se pedalea sin carga. El peso extra puede hacer que el cuadro se flexione, lo que combinado con la velocidad puede producir un bamboleo.

2.- Geometría Adecuada

Hay que buscar que la geometría permita una postura cómoda y relajada, con la espalda semi erguida y con los brazos relajados. Una bicicleta diseñada especialmente para viajar también tendrá vainas más largas, esto contribuye a aumentar la distancia entre ejes, lo que la hace más estable.

3.- Puntos de Sujeción

Una bicicleta diseñada para viajes ligeros solo contará con puntos de sujeción en las vainas superiores para fijar el rack trasero. En cambio, una bicicleta pensada para viajes largos contará con monturas también en la tijera/horquilla para poder instalar un rack delantero.

4.- Rango de Velocidades

Es recomendable buscar una con 3 platos (26-36-48, más o menos).

5.- Ruedas Fuertes

Es importante que una bici para viaje se equipe con un buen par de ruedas incluyendo una buena maza, mínimo 36 rayos/radios, aros de doble pared y unas buenas llantas con protección anti pinchazo.

Info. Adicional: El diseño y la configuración básica de la bicicleta han cambiado poco desde la primera transmisión de cadena desarrollada alrededor del año 1885.

DEFINICIÓN DE LA PROGRAMACIÓN

La optimización topológica es una técnica que se encuentra englobada dentro del campo de análisis estructural. Su objetivo es el aligeramiento estructural manteniendo las funcionalidades mecánicas de componente. A su vez, ofrece un nuevo concepto de diseño estructural enfocado a aquellas aplicaciones donde el peso del componente es crucial.

En otras palabras, esta optimización consiste en utilizar un software concreto para poder eliminar el material que no posee los soportes. 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. Seguido a esto, 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.

Como se utilizó el mismo código de optimización implementado en la práctica anterior (práctica 1), se reciclo algunos datos importantes para explicar cómo funciona este tipo de código:

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

La principal diferencia que existe entre las bicicletas diseñadas para el uso masculino y aquellas que son elaboradas para el uso femenino, es el diseño del cuadro. Puesto que, es bien sabido que aquellas bicicletas que utilizan las mujeres, poseen un diseño en particular, lo cual les permite una mayor facilidad al momento de montarla. Por otro lado, el diseño correspondiente a las bicicletas de los hombres, es un poco más complejo.

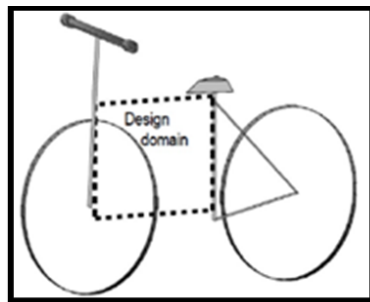
En esta segunda práctica, lo que se busca es poder optimizar el diseño del marco de una bicicleta, con la finalidad de poder mejorarlo. En otras palabras, se va a construir la parte frontal del marco de una bicicleta, para que esta, sea lo más rígida posible.

❖ **NOTA:**

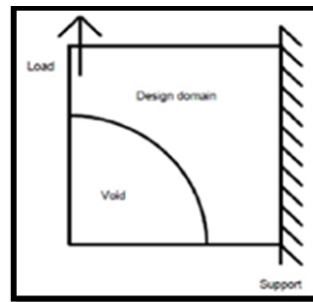
El marco de una bicicleta se encuentra conectado al manubrio y el asiento.

Se tiene las siguientes consideraciones:

- ☐ El manubrio produce una fuerza en dirección vertical.
- ☐ El bastidor trasero actúa como soporte.
- ☐ Se debe declarar una parte vacía del dominio de diseño para hacer espacio para la rueda delantera.



“Problema de Diseño”

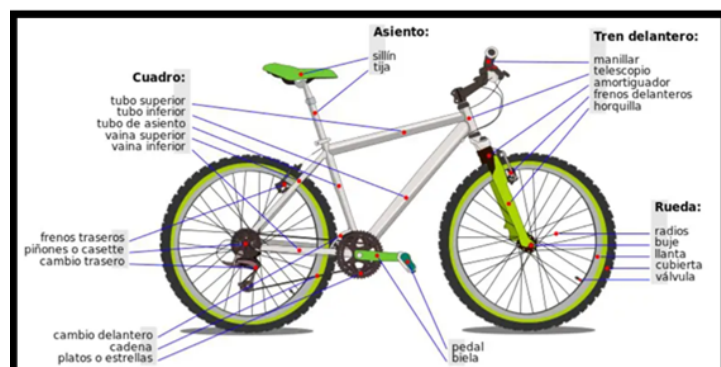


“Vista Esquemática”

2) Estado del Arte.

Los componentes de una bicicleta son las diferentes partes que la forman. Algunos de estos componentes varían en función del tipo de bicicleta, pero en general todas comparten las siguientes partes:

- Cuadro
- Horquilla
- Suspensión
- Ruedas
- Manillar
- Frenos
- Transmisión
- Sillín
- Sistema eléctrico



Como esta práctica consiste en optimizar el diseño del marco de una bicicleta, el enfoque estará orientado a esta parte de la bicicleta, por lo tanto, tenemos la siguiente información:

El cuadro de bicicleta es la estructura rígida o semirrígida que une todos los componentes de la bicicleta. Al cuadro se fija el manillar, la horquilla, el sillín, la rueda trasera, la transmisión y los frenos. En caso de ser una bicicleta eléctrica, también se fija el sistema eléctrico completo o algunas partes de él.

Los cuadros de bicicleta son de materiales resistentes como el aluminio, el acero, el carbono o el magnesio. Están diseñados para ofrecer al ciclista una geometría de conducción optimizada para el uso que se le va a dar a ese tipo de bicicleta.

Así, una bicicleta de carretera tiene una geometría pensada en una posición de pedaleo aerodinámica. En cambio, una bicicleta de paseo tiene una geometría para que el ciclista se pueda sentir lo más cómodo posible.

El material del cuadro tiene una relación directa con tres factores importantes: el peso, la resistencia y el precio. Cuanto menos pese el cuadro, menos peso deberá arrastrar el ciclista y, por tanto, menos esfuerzo deberá realizar.

Los materiales más ligeros y resistentes son el magnesio y el carbono pero también son los más caros. El acero es el material más barato pero también el más pesado. El aluminio es el material más común en los cuadros porque tiene una relación peso-resistencia-coste más equilibrada.

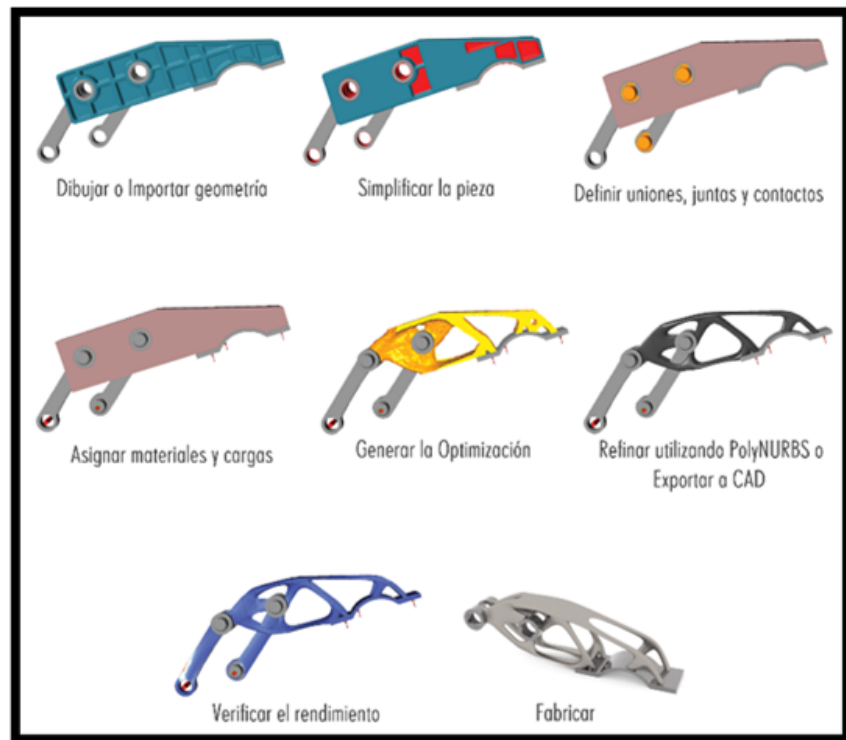


“Ejemplos de marcos de bicicletas”

La optimización topológica es una técnica que pertenece al análisis estructural, y consiste, básicamente, en analizar un componente o estructura y, en función de cómo se cargue, eliminar material ahí donde no es necesario.

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.

A continuación, se expone de manera visual y representativa, los pasos para llevar a cabo una “Optimización Topológica” adecuada:



“Pasos para una Optimización Topológica”

Los principales beneficios que este tipo de optimización nos brinda, son los siguientes:

- 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

El número total de líneas de entrada de Matlab es de 99, incluyendo el optimizador y la subrutina de elementos finitos.

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

Excluyendo las líneas de comentario y las asociadas a la salida y al análisis de elementos finitos, se observa que sólo se necesitan 49 líneas de entrada, de Matlab para resolver un problema de optimización topológica bien planteado.

Añadiendo tres líneas adicionales, el programa puede resolver problemas con múltiples casos de carga. El código está pensado para fines educativos.

3) Propuesta de diseño de la geometría, alcances y limitaciones.

La propuesta de diseño de geometría para esta práctica es realizar el marco de la rueda de una bicicleta, dónde se utilizará el programa que se realizó en la primera práctica de laboratorio para la optimización topológica con el fin de obtener sus alcances y limitaciones.

La geometría a realizar en programación en el software de Matlab, es la siguiente:

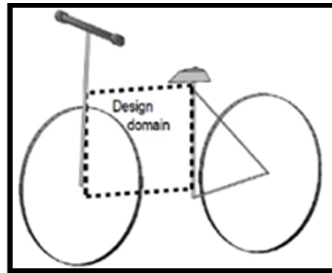


Figura 1: Propuesta de diseño bicicleta.

4) Pasos del desarrollo de la programación.

Se utilizó el código implementado en Matlab de la práctica uno como base, el cual está compuesto por un código de optimización topológica estándar, donde para realizar las modificaciones a dicho código, Se realizó de la siguiente metodología:

- 1) Inicializar el programa de Matlab.

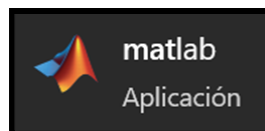


Figura 2: ícono de Matlab.

- 2) Abrir el código utilizado en la práctica uno, por medio de la opción Open de Matlab.

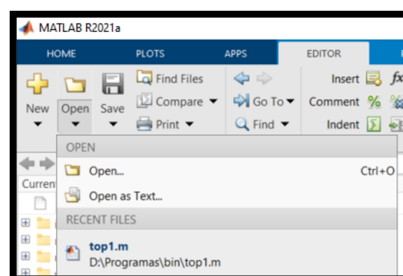
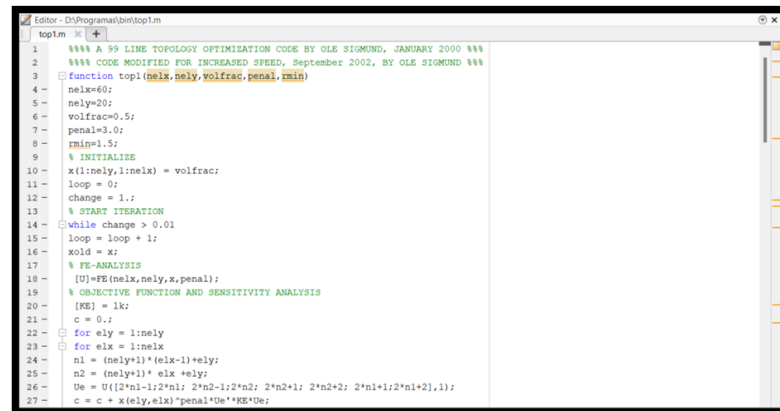


Figura 3: Botón de Open en Matlab.

- 3) Implementar y modificar el código de optimización de topología de 99 líneas utilizado en la práctica uno 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(l: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 % FE-ANALYSIS
18 [U]=FE(nelx,nely,x,penal);
19 % OBJECTIVE FUNCTION AND SENSITIVITY ANALYSIS
20 [KE] = lk;
21 c = 0.;
22 for ely = 1:nely
23 for elx = 1:nelx
24 n1 = (nely+1)*(elx-1)+ely;
25 n2 = (nely+1)*elx +ely;
26 Ue = U((2*n1-1):2*n1; 2*n2-1:2*n2; 2*n2+1; 2*n1+1:2*n1+2),1);
27 c = c + x(ely,elx)*penal*Ue'*KE*Ue;
```

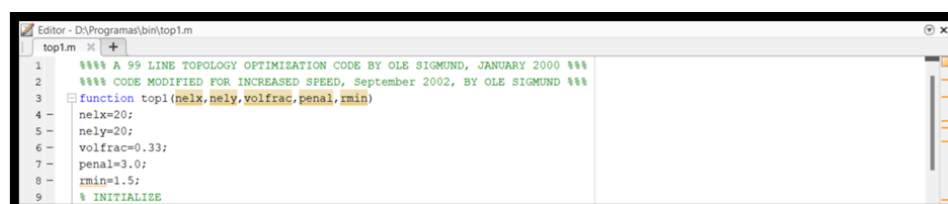
Figura 4: Código de la práctica 1.

- 4) Se realizaron las siguientes modificaciones:

-Se modificaron los parámetros nelx, nely, volfrac, penal y rmin para realizar la optimización que se mostrará en la figura, donde la funcion de cada uno de estos está dada por:

- nelx: Es el número de elementos finitos en el eje x.
- nely: Es el número de elementos finitos en el eje y.
- volfrac: Corresponde a la fracción de volumen en el dominio de diseño.
- penal: Es la penalización de las densidades intermedias. En odne una penalización considerada como alta hará la solución en blanco y negro, por lo tanto los elementos finitos estarán llenos o vacíos, en cambio, una penalización igual a 1 significa que no hay penalización de las densidades intermedias.
- rmin: Es el radio de filtro que hace el diseño de malla-independiente.

Una vez conocidas las funciones de cada uno, estos se implementaron de la siguiente manera:



```
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=20;
5 nely=20;
6 volfrac=0.33;
7 penal=3.0;
8 rmin=1.5;
9 % INITIALIZE
```

Figura 5: Modificación de los parámetros nelx, nely, volfrac, penal y rmin de la práctica 2.

- Antes de iniciar la iteración, entre las líneas 12 a 21 se le agregó al código las líneas mostradas de la figura 5, se realizó debido a que no se deja un espacio libre para el marco de la bicicleta, para poder hacer este espacio se implementó una matriz con ceros.

```

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(l:nely,1:nelx) = volfrac;
11 loop = 0;
12 for ely = 1:nely
13     for elx = 1:nelx
14         if ((elx)^2*(ely-nely)^2) < (0.65*nelx)^2
15             passive(ely,elx) = 1;
16         else
17             passive(ely,elx) = 0;
18         end
19     end
20 end
21 x(find(passive))=0.001;
22 change = 1.;
23 % START ITERATION

```

Figura 6: Líneas de código 12 a 21 de la práctica 2.

Posteriormente, se modificó la línea 44 para realizar la optimización topológica de forma correcta y poder modificar las líneas 46, 54 y 58 del código.

```

40 end
41 % FILTERING OF SENSITIVITIES
42 % [dc] = check(nelx,nely,rmin,x,dc);
43 % DESIGN UPDATE BY THE OPTIMALITY CRITERIA METHOD
44 % [x] = OC(nelx,nely,x,volfrac,dc,passive);
45 % PRINT RESULTS
46 change = max(max(abs(x-xold)));
47 disp(['It.: ' sprintf('%4i',loop) 'Obj.: ' sprintf('%10.4f',c) ...
48 ' Vol.: ' sprintf('%6.3f',sum(sum(x))/(nelx*nely)) ...
49 ' ch.: ' sprintf('%6.3f',change) ])
50 % PLOT DENSITIES
51 colormap(gray); imagesc(-x); axis equal; axis tight; axis off; pause(1e-6);
52 end
53 %%%%%%%%% OPTIMALITY CRITERIA UPDATE %%%%%%%%%

```

Figura 7: Línea de código 44 de la práctica 2.

-Se definió el tamaño de los elementos finitos a una vez por una unidad en la línea 46, esto se hace para realizar la correcta optimización topológica, de esta manera es posible ejecutar las correcciones de escala.

```

43 %%%%%%%%% OPTIMALITY CRITERIA UPDATE %%%%%%%%%
44 function [xnew]=OC(nelx,nely,x,volfrac,dc,passive)
45 l1 = 0; l2 = 100000; move = 0.2;
46 while ((l2-l1)/l2 > 1e-4)
47     lmid = 0.5*(l2+l1);
48     xnew(find(passive))=0.001
49     xnew = max(0.001,max(x-move,min(1.,min(x+move,x.*sqrt(-dc./lmid)))));
50     if sum(sum(xnew)) - volfrac*nelx*nely > 0
51         l1 = lmid;
52     else
53         l2 = lmid;
54     end
55 end

```

Figura 8: Línea de código 46 que define el tamaño de los elementos de la práctica.
2.

-Se modificaron las líneas 54 y 58 añadiendo el comando que empieza los elementos de la zona vacía con el valor de 0.001 que corresponde a $x_{new}(\text{find}(\text{passive}))=0.001$ para realizar la optimización topológica .

```

52 - end
53 - %%%%%%%%%% OPTIMALITY CRITERIA UPDATE %%%%%%%%%%
54 - function [xnew]=OC(nelx,nely,x,volfrac,dc,passive)
55 - l1 = 0; l2 = 100000; move = 0.2;
56 - while ((l2-l1)/l2 > 1e-4)
57 -     lmid = 0.5*(l2+l1);
58 -     xnew(find(passive))=0.001
59 -     xnew = max(0.001,max(x-move,min(1.,min(x+move,x.*sqrt(-dc./lmid)))));
60 -     if sum(sum(xnew)) - volfrac*nelx*nely > 0;
61 -         l1 = lmid;
62 -     else
63 -         l2 = lmid;
64 -     end
65 - end
66 - %%%%%%%%%% MESH-INDEPENDENCY FILTER %%%%%%%%%%

```

Figura 9: Modificación en las líneas de código 54 y 58 de la práctica 2.

-Por medio de las líneas 86 y 87 del código se considera solo la carga y el apoyo de la bicicleta para la optimización topológica.

```

81 - edof = [2*n1-1; 2*n1; 2*n2-1; 2*n2; 2*n2+1; 2*n2+2; 2*n1+1; 2*n1+2];
82 - K(edof,edof) = K(edof,edof) + x(ely,elx)*penal*KE;
83 - end
84 - end
85 - % DEFINE LOADSAND SUPPORTS (HALF MBB-BEAM)
86 - F(2,1) = 1;
87 - fixeddofs = 2*nex*(nely+1)+1:2*(nex+1)*(nely + 1);
88 - alldofs = [1:2*(nely+1)*(nex+1)];
89 - freedofs = setdiff(alldofs,fixeddofs);
90 - % SOLVING 127
91 - U(freedofs,:) = K(freedofs,freedofs) \ F(freedofs,:);
92 - U(fixeddofs,:) = 0;
93 - %%%%%%%%%% ELEMENT STIFFNESS MATRIX %%%%%%%%%%

```

Figura 10: Definición de la optimización topológica por medio de las líneas 86 y 87 de la práctica 2.

5) Se muestra el código completo.

```

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=20;
5  nely=20;
6  volfrac=0.33;
7  penal=3.0;
8  rmin=1.5;
9  % INITIALIZE
10 x(1:nely,1:nelx) = volfrac;
11 loop = 0;
12 for ely = 1:nely
13     for elx = 1:nelx
14         if ((elx)^2+(ely-nely)^2) < (0.65*nex)^2
15             passive(ely,elx) = 1;
16         else
17             passive(ely,elx) = 0;
18         end
19     end
20 end
21 x(find(passive))=0.001;
22 change = 1.;
23 % START ITERATION
24 while change > 0.01
25     loop = loop + 1;
26     xold = x;
27     % FE-ANALYSIS

```

```

Editor - D:\Programas\bin\top1.m
top1.m
27 % FE-ANALYSIS
28 [U]=FE(nelx,nely,x,penal);
29 % OBJECTIVE FUNCTION AND SENSITIVITY ANALYSIS
30 [KE] = lk;
31 c = 0.;
32 for ely = 1:nely
33 for elx = 1:nelx
34 n1 = (nely+1)*(elx-1)+ely;
35 n2 = (nely+1)* elx +ely;
36 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);
37 c = c + x(ely,elx)^penal*Ue'*KE*Ue;
38 dc(ely,elx) = -penal*x(ely,elx)^(penal-1)*Ue'*KE*Ue;
39 end
40 end
41 % FILTERING OF SENSITIVITIES
42 %[dc] = check(nelx,nely,rmin,x,dc);
43 % DESIGN UPDATE BY THE OPTIMALITY CRITERIA METHOD
44 [x] = OC(nelx,nely,x,volfrac,dc,passive);
45 % PRINT RESULTS
46 change = max(max(abs(x-xold)));
47 disp(['It.: ' sprintf('%4i',loop) 'Obj.: ' sprintf('%10.4f',c) ...
48 ' Vol.: ' sprintf('%6.3f',sum(sum(x))/(nelx*nely)) ...
49 ' ch.: ' sprintf('%6.3f',change )])
50 % PLOT DENSITIES
51 colormap(gray); imagesc(~x); axis equal; axis tight; axis off; pause(1e-6);
52 end
53 %***** OPTIMALITY CRITERIA UPDATE *****

```

```

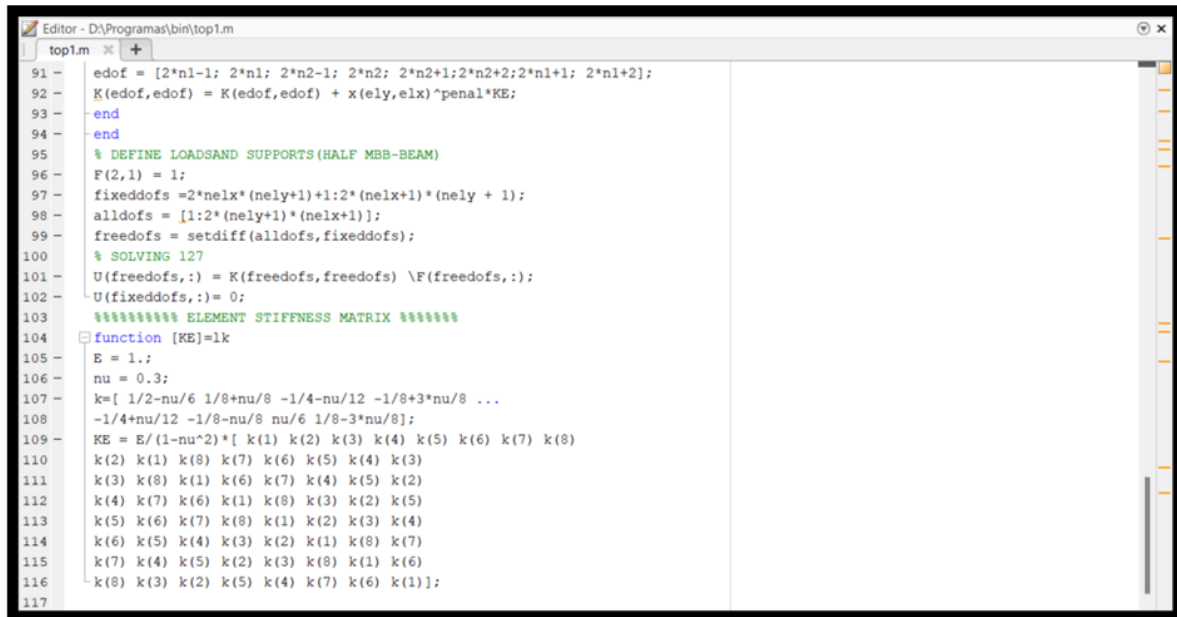
Editor - D:\Programas\bin\top1.m
top1.m
53 %***** OPTIMALITY CRITERIA UPDATE *****
54 function [xnew]=OC(nelx,nely,x,volfrac,dc,passive)
55 l1 = 0; l2 = 100000; move = 0.2;
56 while ((l2-l1)/l2 > 1e-4)
57 lmid = 0.5*(l2+l1);
58 xnew(find(passive))=0.001
59 xnew = max(0.001,max(x-move,min(1.,min(x+move,x.*sqrt(-dc./lmid))));
60 if sum(sum(xnew)) - volfrac*nelx*nely > 0;
61 l1 = lmid;
62 else
63 l2 = lmid;
64 end
65 end
66 %***** MESH-INDEPENDENCY FILTER *****
67 function [dcn]=check(nelx,nely,rmin,x,dc)
68 dcn=zeros(nely,nelx);
69 for i = 1:nelx
70 for j = 1:nely
71 sum=0.0;
72 for k = max(i-round(rmin),1):min(i+round(rmin),nelx)
73 for l = max(j-round(rmin),1):min(j+round(rmin), nely)
74 fac = rmin-sqrt((i-k)^2+(j-l)^2);
75 sum = sum+max(0,fac);
76 dcn(j,i) = dcn(j,i) + max(0,fac)*x(l,k)*dc(l,k);
77 end
78 end
79 dcn(j,i) = dcn(j,i)/(x(j,i)*sum);

```

```

Editor - D:\Programas\bin\top1.m
top1.m
79 dcn(j,i) = dcn(j,i)/(x(j,i)*sum);
80 end
81 end
82 %***** FE-ANALYSIS *****
83 function [U]=FE(nelx,nely,x,penal)
84 [KE] = lk;
85 K = sparse(2*(nelx+1)*(nely+1), 2*(nelx+1)*(nely+1));
86 F = sparse(2*(nely+1)*(nelx+1),1); U = sparse(2*(nely+1)*(nelx+1),1);
87 for ely = 1:nely
88 for elx = 1:nelx
89 n1 = (nely+1)*(elx-1)+ely;
90 n2 = (nely+1)* elx +ely;
91 edof = [2*n1-1; 2*n1; 2*n2-1; 2*n2; 2*n2+1;2*n2+2;2*n1+1; 2*n1+2];
92 K(edof,edof) = K(edof,edof) + x(ely,elx)^penal*KE;
93 end
94 end
95 % DEFINE LOADSAND SUPPORTS (HALF MBB-BEAM)
96 F(2,1) = 1;
97 fixeddofs = 2*nelx*(nely+1)+1:2*(nelx+1)*(nely + 1);
98 alldofs = [1:2*(nely+1)*(nelx+1)];
99 freedofs = setdiff(alldofs,fixeddofs);
100 % SOLVING 127
101 U(freedofs,:) = K(freedofs,freedofs) \ F(freedofs,:);
102 U(fixeddofs,:) = 0;
103 %***** ELEMENT STIFFNESS MATRIX *****
104 function [KE]=lk
105 E = 1.;

```



```
Editor - D:\Programas\bin\top1.m
top1.m
91 - edof = [2*n1-1; 2*n1; 2*n2-1; 2*n2; 2*n2+1; 2*n2+2; 2*n1+1; 2*n1+2];
92 - K(edof,edof) = K(edof,edof) + x(ely,elx)^penal*KE;
93 - end
94 - end
95 - % DEFINE LOADSAND SUPPORTS (HALF MBB-BEAM)
96 - F(2,1) = 1;
97 - fixeddofs = 2*n1*(nely+1)+1:2*(n1+1)*(nely + 1);
98 - alldofs = [1:2*(nely+1)*(n1+1)];
99 - freedofs = setdiff(alldofs,fixeddofs);
100 - % SOLVING 127
101 - U(freedofs,:) = K(freedofs,freedofs) \ F(freedofs,:);
102 - U(fixeddofs,:) = 0;
103 - % ELEMNT STIFFNESS MATRIX %
104 - function [KE]=1k
105 - E = 1.;
106 - nu = 0.3;
107 - k=[ 1/2-nu/6 1/8+nu/8 -1/4-nu/12 -1/8+3*nu/8 ...
108 - 1/4+nu/12 -1/8-nu/8 nu/6 1/8-3*nu/8];
109 - KE = E/(1-nu^2)*[ k(1) k(2) k(3) k(4) k(5) k(6) k(7) k(8)
110 - k(2) k(1) k(8) k(7) k(6) k(5) k(4) k(3)
111 - k(3) k(8) k(1) k(6) k(7) k(4) k(5) k(2)
112 - k(4) k(7) k(6) k(1) k(8) k(3) k(2) k(5)
113 - k(5) k(6) k(7) k(8) k(1) k(2) k(3) k(4)
114 - k(6) k(5) k(4) k(3) k(2) k(1) k(8) k(7)
115 - k(7) k(4) k(5) k(2) k(3) k(8) k(1) k(6)
116 - k(8) k(3) k(2) k(5) k(4) k(7) k(6) k(1)];
117
```

6) Una vez escrito todo el código, se guardó, prestando atención en la ubicación donde se guardó el script, así como en el nombre que se le asignó al archivo. En este caso, otra vez se le guardó con el nombre de top1, donde su extensión por default es de .m.

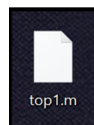


Figura 11: Extensión del programa en Matlab.

7) Finalmente, se ejecutó el programa haciendo uso del botón de “Run”.

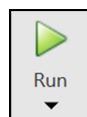
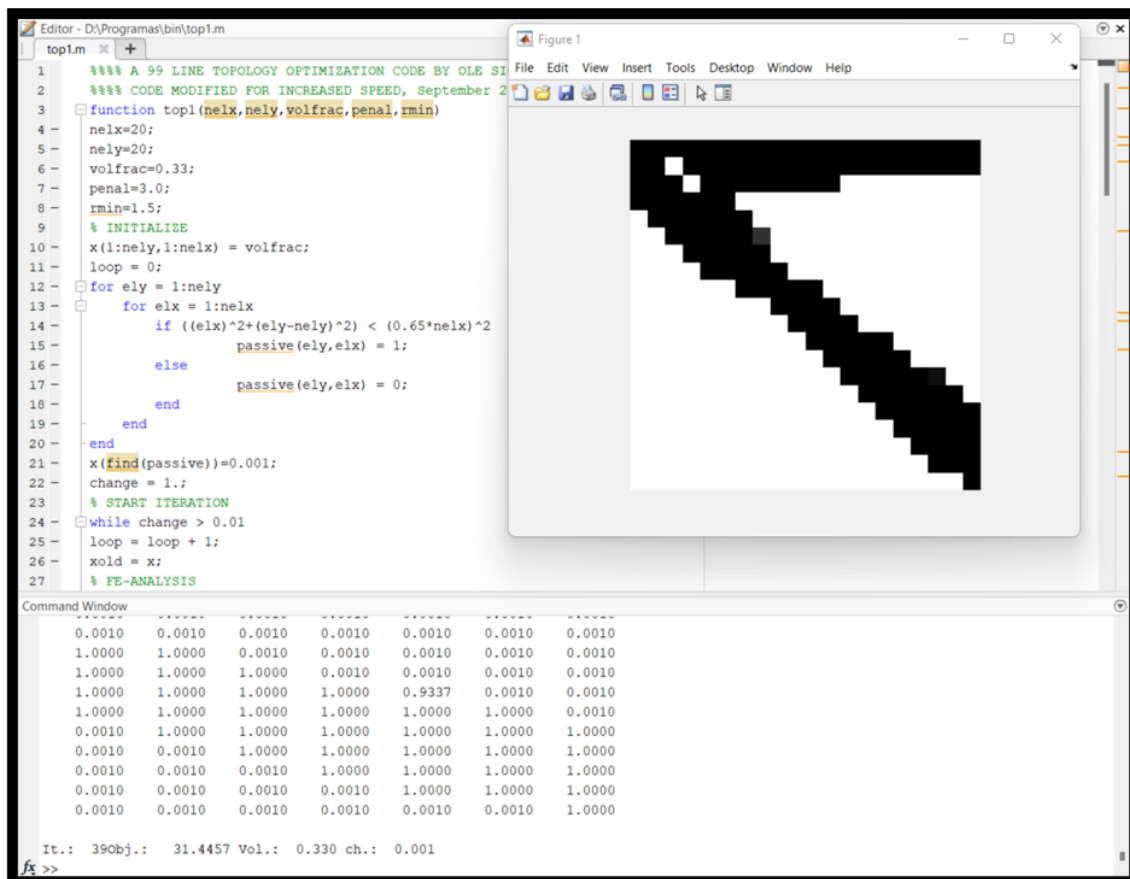


Figura 12: Botón de ejecución de programa en Matlab.

5) Resultados de la optimización.

Resultado de la optimización del código de 99 líneas modificado



6) Conclusiones por cada autor.

☐ Luis Carlos Gomez Espinoza 1926227

Como conclusión en la práctica dos del laboratorio, una vez comprendido cómo funciona la optimización topológica, por medio de la modificación del código de 99 líneas de la práctica uno, fue posible realizar la aplicación de este a un marco de una bicicleta, solo considerando para esta práctica la carga y el apoyo de la misma, realizando cambios en los parámetros del código, principalmente en nelx, nely, volfrac, penal y rmin donde nelx es el número de elementos finitos en el eje x, nely es el número de elementos finitos en y, volfrac es la fracción de volumen en el dominio de diseño, penal se encarga de la penalización de las densidades intermedias y por último, rmin es un radio de filtro que hace el diseño de malla-independiente.

Es importante destacar que la optimización topológica es una gran herramienta para el diseño, asimismo, programas como Matlab nos ayudan a realizar las

optimizaciones de una manera sencilla y eficaz, permitiendo obtener grandes conocimientos de aplicación en el laboratorio de biomecánica.

☐ **César Mauricio Alvarez Olguín 1910330**

Esta segunda práctica hizo uso del código de optimización topológica implementado en la práctica anterior, es este caso, para poder desarrollar el marco de una bicicleta, es gracias a esto, que el desarrollo del código mostrado con anterioridad, resultó ser bastante sencillo tanto de utilizar, así como de comprender su funcionamiento. Uno de los cambios más sobresalientes realizados al código original, fueron los que se hicieron en los parámetros nelx, nely, volfrac, penal y rmin, esto con el objetivo de llevar a cabo la optimización que se obtuvo como resultado.

A su vez, se modificaron diversas líneas del código, donde cabe destacar el tamaño de los elementos finitos a una vez por una unidad en la línea 46, lo cual se hizo para poder realizar correctamente la optimización, puesto que de esta manera, fue posible ejecutar las correcciones de escala.

Finalmente, por medio de las líneas 86 y 87 del código, se considera única y exclusivamente la carga y el apoyo de la bicicleta para la optimización.

☐ **Mauricio Martinez Tovar**

Para esta práctica se realizó un código de optimización topológica, con esto se puede utilizar para una amplia investigación en nuestro caso fue desarrollar un marco de bicicleta para implementar un mejor diseño y que sea 100% funcional tanto en estética como en peso y aerodinámico. Sin duda una tarea algo compleja pero con lo realizado en la práctica anterior se complementó.

☐ **Mauro Alberto Hernández Saldaña**

En esta segunda utilizamos lo que fue la optimización topológica que vimos en la práctica 11, pero ahora de manera más implementaría, para poder desarrollar un diseño de marco para una bicicleta, generando un diseño que tuviere la mejor resistencia y optimización posible, fuera de la modificación de algunas partes del código, la práctica no contó con una dificultad muy elevada, lo que hizo que aprender de esta práctica y comprender cómo funcionaba fuera más fácil, por lo que este tema quedó comprendido muy bien y es de un gran apoyo.

☐ **Eduardo Rodriguez Montalvo**

En esta segunda práctica, comprendimos y analizamos más a detalle el proceso de optimización topológica, todo esto aplicado a una propuesta de análisis y modificación a un marco de bicicleta como es mencionado en el nombre de la actividad. Todo esto se vio plasmado en la realización del código en la aplicación matlab misma que nos permitió apreciar todo esto en forma gráfica. Indagamos un

poco en nuevos comandos de modificación como lo fueron nel x y y, penal, entre otros. A lo largo del desarrollo de la práctica, lo más destacable fue el tener cada detalle en orden en las distintas líneas de código, mismas que nos arrojaron un resultado satisfactorio.

☐ **Luis Alberto Pinzon Garcia**

En esta Practica 2 que lleva por nombre “Diseño del marco de una bicicleta” se nos pidió que generamos un propuesta de cómo analizar y de programar para poder ejecutarlo en una optimización topológica y como lo dice la practica asi se aplicó, y pues llegamos a la formulación del código como se muestra y resultó ser de una manera sencilla para que otros lo puedan utilizar y que la comprensión de cómo funciona sea razonable. Y como podemos observar se hicieron otros parámetros los cuales fueron nelx, nely, volfrac, penal y rmin,y cómo los podemos comprobar logramos llegar a un resultado de poder optimizar nuestro diseño. También hicimos modificaciones a otras líneas del mismo código para diferenciar los elementos finitos, y así realizamos la optimización de buena forma y ya así poder ejecutar.

❖ Fuentes bibliográficas

- 99 Line Topology Optimization Code - O. Sigmund, Department of Solid Mechanics, Building 404, Technical University of Denmark, DK-2800 Lyngby, Denmark.
- *Optimización Topológica*. (2019, 9 enero). Estudio de Ingeniería y Tecnología Avanzada S.L. Recuperado 16 de septiembre de 2022, de <https://eitaingenieros.com/optimizacion/>
- Pedrero, J. C. (2019, 17 enero). *GEOMETRÍA DE UN CUADRO DE BICICLETA*. Escuela La Bicicleta. Recuperado 16 de septiembre de 2022, de <https://labicicleta.net/escuela/geometria-de-un-cuadro-de-bicicleta/>