

Práctica 2

EQUIPO 6

Christopher Alejandro Monsivais Ramos 1899218

Alessandra González Torres 1895846

Angela Itzel Rodríguez Torres 1925284

Abdiel Isaac Rodríguez Pinales 1842044

Martin Alejandro Villanueva Cortes 1870820

Alejandro Sebastian Marin Cota 1814201

13 de septiembre de 2022

1. Marco Teórico

La geometría de una bici mide las longitudes de los tubos que la conforman, así como los ángulos que forman dichos tubos en la dirección y en el tubo de sillín principalmente. Los tubos se miden desde centro a centro y evidentemente no es necesario que la forma de los tubos sea convencional para medirlos. Lo que se mide es la longitud; no importa si el cuadro está realizado en algún tipo de monocasco o con tubería convencional o hidroformada. De este modo, además de la talla, que es el primer parámetro por el que elegimos una bici a nivel de medidas, la geometría es básica para que la bici se comporte de una manera u otra dependiendo del conjunto de medidas y ángulos.

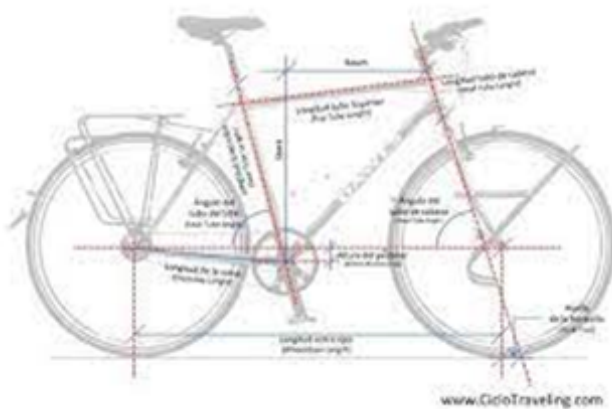


Figura 1: Geométrie de la bicicleta

El cuadro de bicicleta es la estructura rígida o semi-rígida 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.

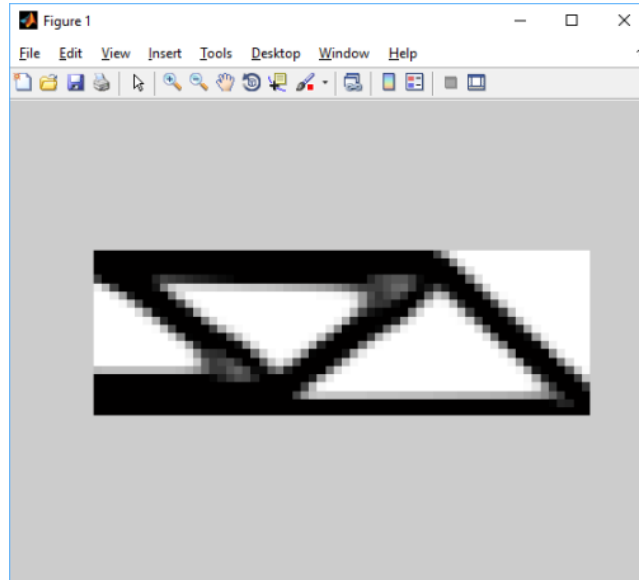


Figura 2: Ejemplo de la forma geométrica

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. A diferencia de otros tipos de optimización, la optimización topológica ofrece un nuevo concepto de diseño estructural enfocado a aquellas aplicaciones donde el peso del componente es crucial (por ejemplo, la industria aeroespacial).

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.[1]

2. Estado del arte

Hay diferencias entre las bicicletas de los hombres y de las mujeres y una de ellas es el diseño del cuadro. La bicicleta de las mujeres está diseñada para hacer más fácil de montar mientras que las bicicletas del hombre no. En esta práctica vamos a optimizar el diseño del marco para mejorarlo.[2]

El problema de diseño se ilustra en la figura a). Nuestra tarea es construir la parte frontal del marco para que la bicicleta sea lo más rígido posible. Esta parte se conecta con el manubrio y el asiento. El dominio de diseño se ilustra esquemáticamente en la figura b). Consideraciones:

- El manubrio produce una fuerza en dirección vertical.
- Bastidor trasero actúa como soporte.
- Tenemos que declarar una parte vacía del dominio de diseño para hacer espacio para la rueda delantera.

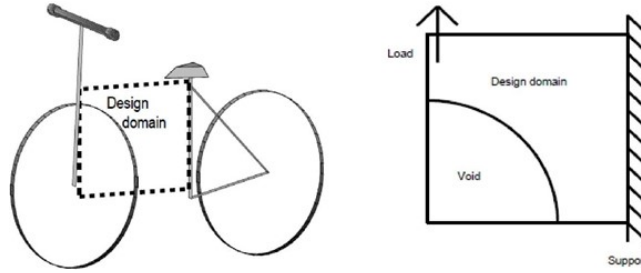


Figura 3: figura a) figura b)

3. Procedimiento de la programación

Codigo que se utiliza:

```

%%% A 99 LINE TOPOLOGY OPTIMIZATION CODE BY OLESIGMUND, OCTOBER 1999 %%%
function topp1(nelx,nely,volfrac,penal,rmin)
% INITIALIZE
x(1:nely,1:nelx) = volfrac;
loop = 0;
for ely = 1:nely
for elx = 1:nelx
if ((elx)^2+(ely-nely)^2) < (0.65*nelx)^2
passive(ely,elx) = 1;
else
passive(ely,elx) = 0;
end
end
end
x(find(passive))=0.001;
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,passive);
% 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,passive)
l1 = 0; l2 = 100000; move = 0.2;
while ((l2-l1)/l2 > 1e-4)
lmid = 0.5*(l2+l1);
xnew(find(passive))=0.001
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] = 1k;
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 LOADSAND SUPPORTS(HALF MBB-BEAM)
F(2,1) = 1;

```

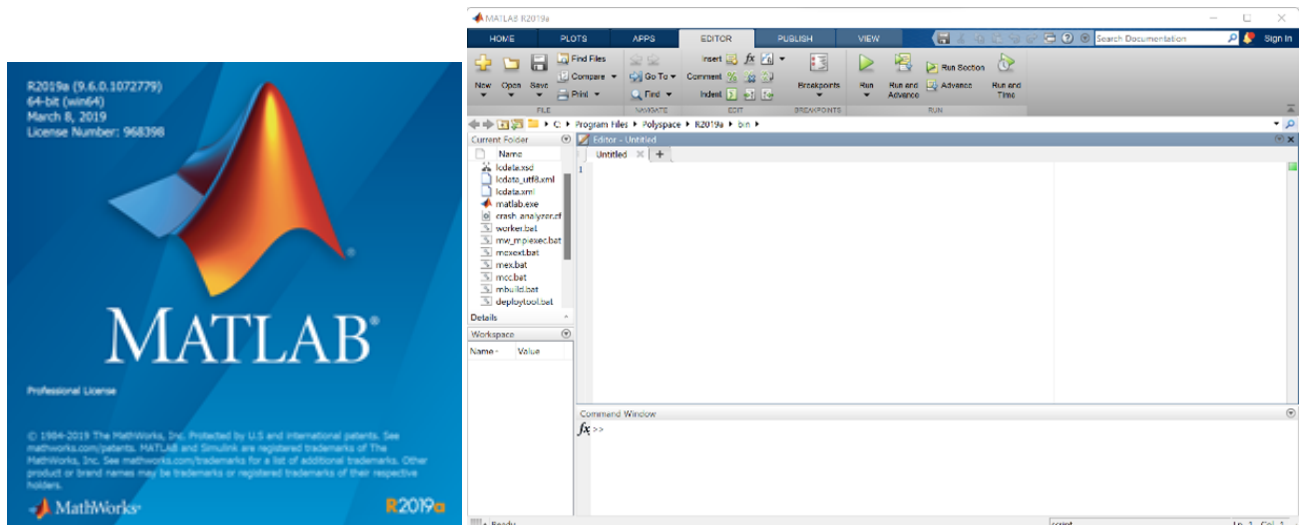
```

fixeddofs = 2*nelx*(nely+1)+1:2*(nelx+1)*(nely + 1);
alldofs = [1:2*(nely+1)*(nelx+1)];
freedofs = setdiff(alldofs,fixeddofs);
% SOLVING 127
U(freedofs,:) = K(freedofs,freedofs) \F(freedofs,:);
U(fixeddofs,:) = 0;
%%%%%%%%% ELEMENT STIFFNESS MATRIX %%%%%%%%%%
function [KE]=lk
E = 1.;
nu = 0.3;
k=[ 1/2-nu/6 1/8+nu/8 -1/4-nu/12 -1/8+3*nu/8 ...
-1/4+nu/12 -1/8-nu/8 nu/6 1/8-3*nu/8];
KE = E/(1-nu^2)*[ k(1) k(2) k(3) k(4) k(5) k(6) k(7) k(8)
k(2) k(1) k(8) k(7) k(6) k(5) k(4) k(3)
k(3) k(8) k(1) k(6) k(7) k(4) k(5) k(2)
k(4) k(7) k(6) k(1) k(8) k(3) k(2) k(5)
k(5) k(6) k(7) k(8) k(1) k(2) k(3) k(4)
k(6) k(5) k(4) k(3) k(2) k(1) k(8) k(7)
k(7) k(4) k(5) k(2) k(3) k(8) k(1) k(6)
k(8) k(3) k(2) k(5) k(4) k(7) k(6) k(1)];

```

4. Pasos que seguir para la programación en MATLAB

1. Abrir MATLAB y esperar a que éste se inicialice, y muestre su pantalla principal.



2. Una vez en la pantalla de inicio de MATLAB es necesario abrir un archivo nuevo en el editor de texto, dentro del cual será necesario escribir el código proporcionado.

```

Editor - Untitled*
Untitled* x +
1  %%% A 99 LINE TOPOLOGY OPTIMIZATION CODE BY OLESIGMUND, OCTOBER 1999 %%%
2  function topp1(nelx,nely,volfrac,penal,rmin)
3  % INITIALIZE
4  x(1:nely,1:nelx) = volfrac;
5  loop = 0;
6  for ely = 1:nely
7  for elx = 1:nelx
8  if ((elx)^2+(ely-nely)^2) < (0.65*nelx)^2
9  passive(ely,elx) = 1;
10 else
11 passive(ely,elx) = 0;
12 end
13 end
14 end
15 x(find(passive))=0.001;
16 change = 1.;
17 % START ITERATION
18 while change > 0.01
19 loop = loop + 1;
20 xold = x;
21 % FE-ANALYSIS
22 [U]=FE(nelx,nely,x,penal);
Command Window
fx>>

```

Figura 4: Código de Matlab P1

```

Editor - C:\Users\aleiss\Documents\UAN\7° Semestre\Biomecánica\Lab Biomecánica\topp1.m
topp1.m x +
86 K(edof,edof) = K(edof,edof) + x(ely,elx)^penal*KE;
87 end
88 end
89 % DEFINE LOADSAND SUPPORTS (HALF MBB-BEAM)
90 F(2,1) = 1;
91 fixeddofs = 2*nelx*(nely+1)+1:2*(nelx+1)*(nely + 1);
92 alldofs = [1:2*(nely+1)*(nelx+1)];
93 freeddofs = setdiff(alldofs,fixeddofs);
94 % SOLVING 127
95 U(freeddofs,:) = K(freeddofs,freeddofs) \ F(freeddofs,:);
96 U(fixeddofs,:) = 0;
97 %%%%%%%%% ELEMENT STIFFNESS MATRIX %%%%%%%%%
98 function [KE]=lk
99 E = 1.;
100 nu = 0.3;
101 k=[ 1/2-nu/6 1/8+nu/8 -1/4-nu/12 -1/8+3*nu/8 ...
102 -1/4+nu/12 -1/8-nu/8 nu/6 1/8-3*nu/8];
103 KE = E/(1-nu^2)*[ k(1) k(2) k(3) k(4) k(5) k(6) k(7) k(8)
104 k(2) k(1) k(8) k(7) k(6) k(5) k(4) k(3)
105 k(3) k(8) k(1) k(6) k(7) k(4) k(5) k(2)
106 k(4) k(7) k(6) k(1) k(8) k(3) k(2) k(5)
107 k(5) k(6) k(7) k(8) k(1) k(2) k(3) k(4)
108 k(6) k(5) k(4) k(3) k(2) k(1) k(8) k(7)
109 k(7) k(4) k(5) k(2) k(3) k(8) k(1) k(6)
110 k(8) k(3) k(2) k(5) k(4) k(7) k(6) k(1)];
Command Window

```

Figura 5: Código de Matlab P2

- El siguiente paso consiste en salvar el archivo. Se recomienda que el archivo se guarde en el directorio raíz de MATLAB que por default muestra. En caso de no ser así, debemos de navegar a “Mis Documentos MATLAB” y guardar ahí el script recién creado.

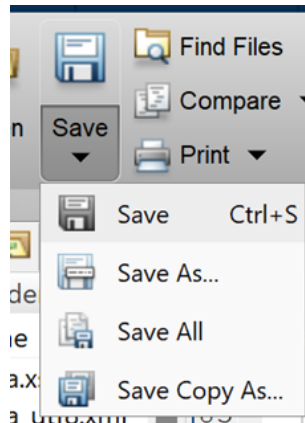


Figura 6: Como salvar un archivo

4. Una vez guardado el script en el directorio correcto, solo hace falta corroborar que el intérprete de MATLAB se encuentre en el mismo directorio. Esto se hace desde la pantalla principal de MATLAB. Para la versión R2019a del software, el directorio actual del intérprete se encuentra en la barra de herramientas superior.

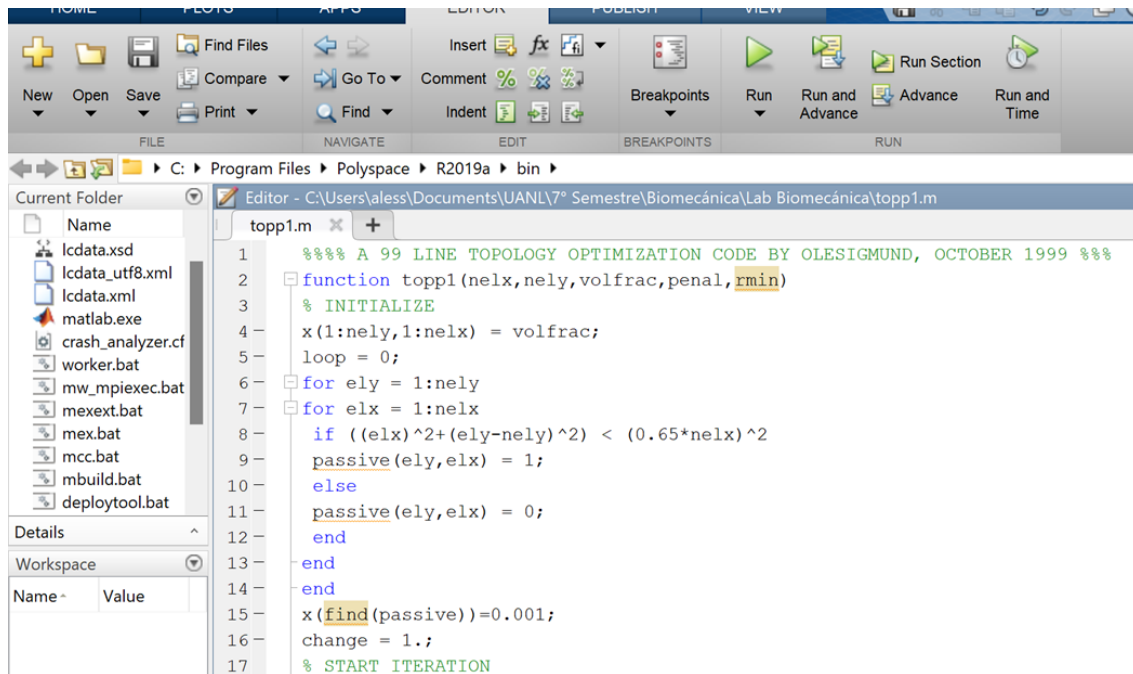


Figura 7: Barra de herramientas superior

5. Por último, hay que ejecutar el programa desde la ventana de comando de MATLAB. El código que se proporcionó viene preparado para optimizar un dominio de diseño, este caso en particular es evaluado y simulado cuando escribimos desde la línea de comando de MATLAB “topp1(12,12,0.33,3.0,0.9)”.

- nelx y nely: Son el número de elementos en las direcciones horizontales (x) y verticales (y).
- volfrac: Es la fracción de volumen.
- penal: Es el poder de penalización.
- rmi: Es el tamaño del filtro (dividido por el tamaño del elemento).

5. Resultados de la optimización

```
Editor - C:\Users\alees\Documents\UANL\7º Semestre\Biomecánica\Lab Biomecánica\topp1.m
topp1.m
99- E = 1.;
100- nu = 0.3;
101- k=[ 1/2-nu/6 1/8+nu/8 -1/4-nu/12 -1/8+3*nu/8 ...
102- -1/4+nu/12 -1/8-nu/8 nu/6 1/8-3*nu/8];
103- KE = E/(1-nu^2)*[ k(1) k(2) k(3) k(4) k(5) k(6) k(7) k(8)
104- k(2) k(1) k(8) k(7) k(6) k(5) k(4) k(3)
105- k(3) k(8) k(1) k(6) k(7) k(4) k(5) k(2)
106- k(4) k(7) k(6) k(1) k(8) k(3) k(2) k(5)
107- k(5) k(6) k(7) k(8) k(1) k(2) k(3) k(4)
108- k(6) k(5) k(4) k(3) k(2) k(1) k(8) k(7)
109- k(7) k(4) k(5) k(2) k(3) k(8) k(1) k(6)
110- k(8) k(3) k(2) k(5) k(4) k(7) k(6) k(1)];

Command Window
0.0010
0.0010
0.0010
0.0010
0.0010
0.0010
0.0010
0.0010
1.0000
1.0000
1.0000

It.: 15Obj.: 29.7510 Vol.: 0.330 ch.: 0.002
fx>> topp1(ne, nely, volfrac, penal, rmin)
```

Figura 8: Código en MatLab

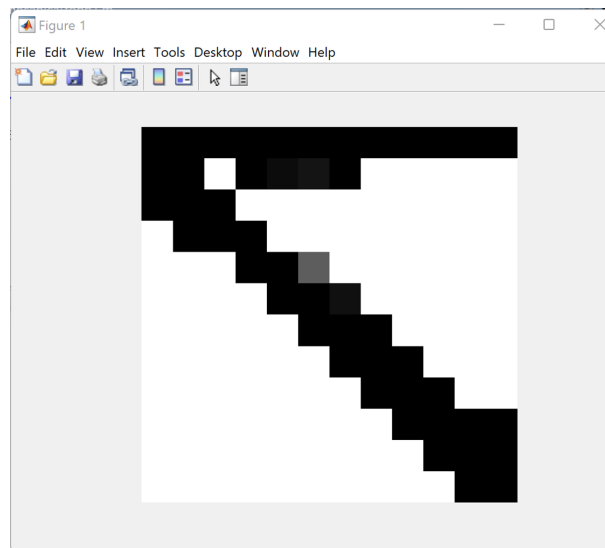


Figura 9: Figura que se crea en MatLab

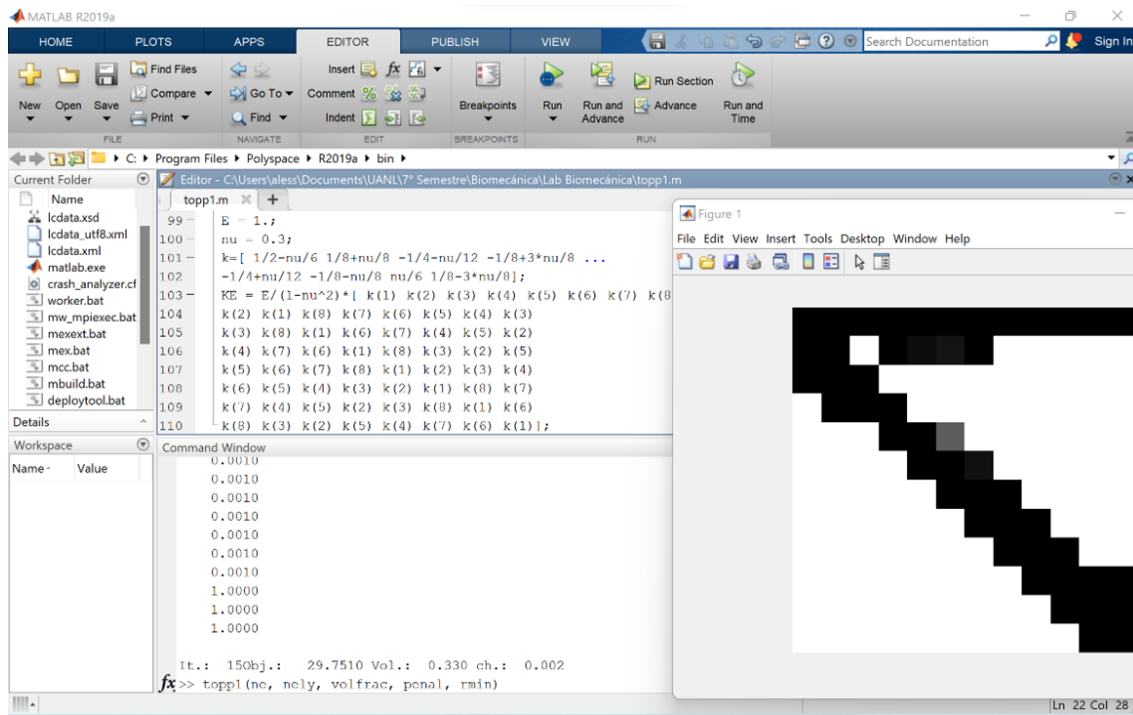


Figura 10: Pantalla de MatLab con código y figura

6. Conclusiones

Christopher Alejandro Monsivais Ramos:

Como conclusión a esta práctica puedo decir que nos sirvió mucho para seguir practicando este tipo de simulaciones sobre las optimizaciones topológicas, que en este caso se realizó a un marco de bicicleta. Esto nos ayuda mucho también para seguir practicando la programación de distintos tipos de simulaciones en Matlab como lo hemos ido haciendo durante la carrera.

Alessandra Gonzalez Torres:

En esta práctica seguimos basándonos en el código de la práctica anterior solo que se le hicieron algunas modificaciones y eso se puede ver en las líneas de código junto con los resultados obtenidos, ahora tuvimos una meta diferente y se creó una forma geométrica que parece (desde mi punto de vista) una parte de la rueda de una bicicleta.

Angela Itzel Rodriguez Torres:

En la realización de esta práctica utilizamos datos de la práctica anterior donde solo cambiamos la figura a una bicicleta y el uso de los comandos que nos ayuda en la creación de un modelo en el que se aplican diferentes cargas para una pieza y usamos el software Matlab para codificar y ejecutarlo, fue algo sencillo ya que algunos del equipo incluyéndome ya habíamos usado Matlab antes y con los pequeños conocimientos que cada quien tenía, pudimos complementarlo, errores que se corrigieron.

Abdiel Isaac Rodriguez Piñales:

Se concluye que los avances de tecnología han sido tales que se ha conseguido la optimización de los nuevos diseños. Como en el método de optimización topológica la cual sirve para reducir los materiales a utilizar los nuevos diseños. Esto ha sido útil en áreas como la ingeniería aeroespacial donde es crucial el volumen y materia de los componentes a utilizar. Todo es gracias al avance en el desarrollo de software.

Alejandro Sebastian Marin Cota:

En la primera instrucción que nos dio la práctica podemos ver que el diseño final depende en gran medida del

tamaño de la cuadrícula. Al exigir que el resultado no sea en blanco y negro, encontrará que el diseño pierde nitidez y se ve muy diferente. Finalmente, al aplicar un filtro, puede suavizar los bordes del resultado final del diseño y obtener una imagen más nítida en comparación con la anterior. Esto le permite ver que diferentes funciones en su código pueden hacer diferentes diseños.

Martín Alejandro Villanueva Cortes:

La realización de esta practica fue un algo parecida a la anterior ya que se nos fue solicitado hacer en esta ocasión la geometría de una bicicleta con la cual ya tuvimos que ver detalles como que tipo de bici puede ser o para quien la va a utilizar. Basándonos con estas características se opto por un diseño lo mas general posible a una bicicleta, lo cual nos puede dar una gran variedad de posibles códigos en MatLab para realizar cada una de las bicicletas que pueden haber.

Referencias

- [1] Optimización topológica, 2017.
- [2] O. Sigmund. 99 line topology optimization code, 1999.