



UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN
FACULTAD DE INGENIERÍA MECÁNICA Y ELÉCTRICA



Agosto - Diciembre 2022

Unidad de Aprendizaje: Laboratorio de Biomecánica

Practica No. 1

Nombre	Matricula
Gabryiel Bailon Avila	1869828
Carlos Eduardo Rivera López	1897545
Víctor Adrián Higuera Vázquez	1876474
Andrés Anaya Hernández	1914471
César Armando Luna Zapata	1844920

Hora: Martes V2

San Nicolás de los Garza, Nuevo León fecha: 06-septiembre-2022

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 como se ejecuta el análisis.

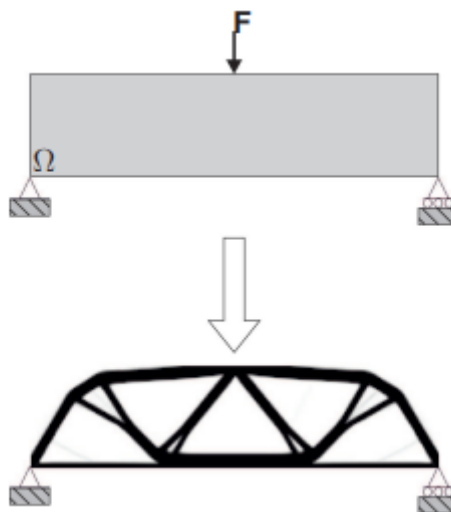
Nombre y definición del programa

La optimización topológica (OT) es una herramienta matemática que le permite al diseñador sintetizar topologías óptimas. En ingeniería mecánica se entiende como topología óptima a una pieza o parte mecánica diseñada especialmente para maximizar o minimizar alguna característica deseada.

La OT es un campo de investigación de rápido crecimiento, donde intervienen distintas áreas como son las matemáticas, la mecánica y las ciencias computacionales, y que cuenta con importantes aplicaciones prácticas en la industria y en el sector de manufactura. En la actualidad, la optimización topológica es usada en las industrias aeroespacial, automotriz, de obras civiles, entre otras. Además tiene un papel muy importante en el campo de las micro y nanotecnologías, principalmente, en el diseño de mecanismos flexibles.

La Figura 1 muestra un dominio bidimensional con restricciones (apoyos) en las esquinas inferiores y una carga (F) en la mitad del lado superior. Entonces, el enunciado de un problema de máxima rigidez (o mínima flexibilidad) es:

¿Cuál es la distribución de material en el dominio que proporciona la máxima rigidez para el estado de carga y la restricción de volumen determinados?



ESTADO DEL ARTE

La optimización topológica es una técnica que soluciona numéricamente el problema, por medio de algoritmos iterativos que maximizan o minimizan una función objetivo (o de costo).

Existen muchas otras aplicaciones de la OT en la ingeniería mecánica, como son: diseño de mecanismos flexibles y micromecanismos, diseño de MEMs (Sistemas Micro-Electro-Mecánicos), diseño de metamateriales (materiales con coeficiente de Poisson o coeficiente de expansión térmica negativos), resonadores, aletas para intercambio de calor, entre otras aplicaciones. Además, están comenzando a aparecer trabajos sobre aplicación de la técnica en la dinámica de fluidos, tales como diseño de mezcladores, muros de contención de represas, perfiles de ala de avión, entre otros.

Muchas industrias que fabrican piezas mecánicas, tales como la automotriz, aeroespacial, naval, de obras civiles, entre otras, requieren el diseño de piezas con el menor peso posible pero asegurando su resistencia mecánica. Actualmente, la reducción de peso se debe a criterios del diseñador, quien retira material de una manera casi empírica y comprueba si el resultado es aceptable por medio del análisis de resistencia mecánica. La técnica de la optimización topológica permite diseñar estructuras óptimas, con el menor peso posible para un determinado estado de carga y una determinada restricción de volumen.

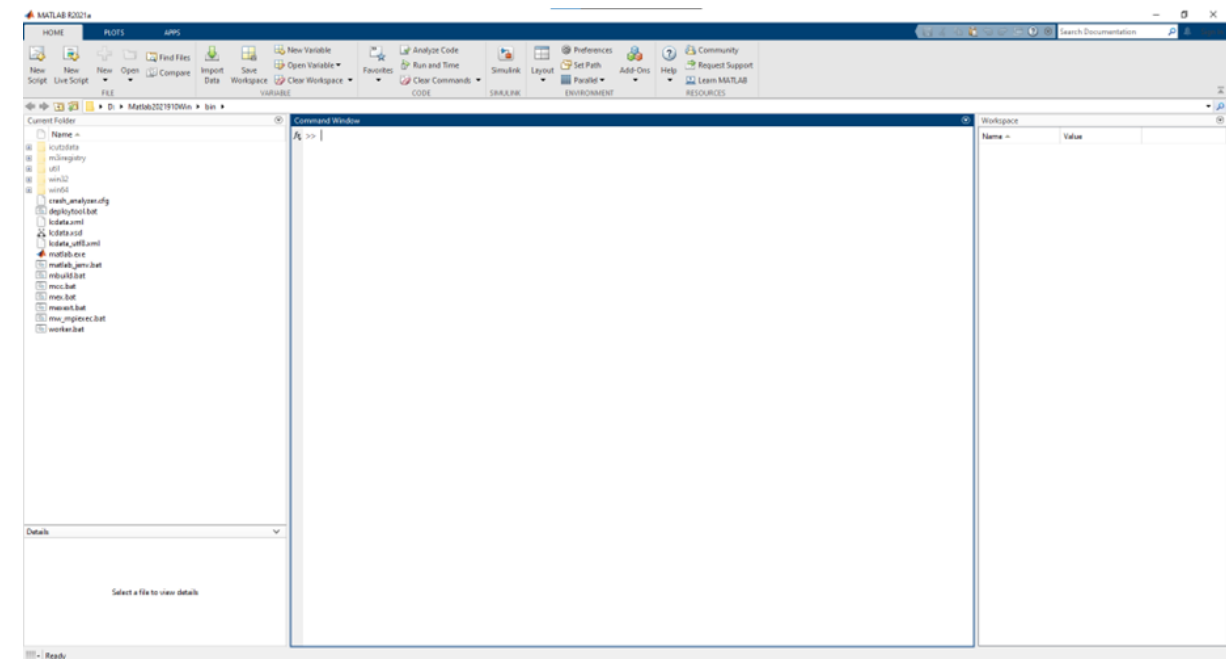
La OT es una técnica relativamente nueva. La primera publicación *The limits of economy of material in frame structures* de A. G. Michell apareció en 1904. En este trabajo se estudiaron los criterios de optimización para la reducción de peso en estructuras de barras articuladas, aplicando los resultados al análisis de una especie de viga en voladizo llamada "Viga de Michell". El problema de la viga de Michell ha sido un punto de referencia y ejemplo básico en el estudio de los métodos de optimización estructural.

En la década de 1970, con la evolución en la computación digital y los métodos numéricos, se despertó un nuevo interés en el tema. Autores como Mröz y Pradera estudiaron la optimización analítica exacta en problemas estructurales y se mostraron las importantes ventajas de los métodos numéricos.

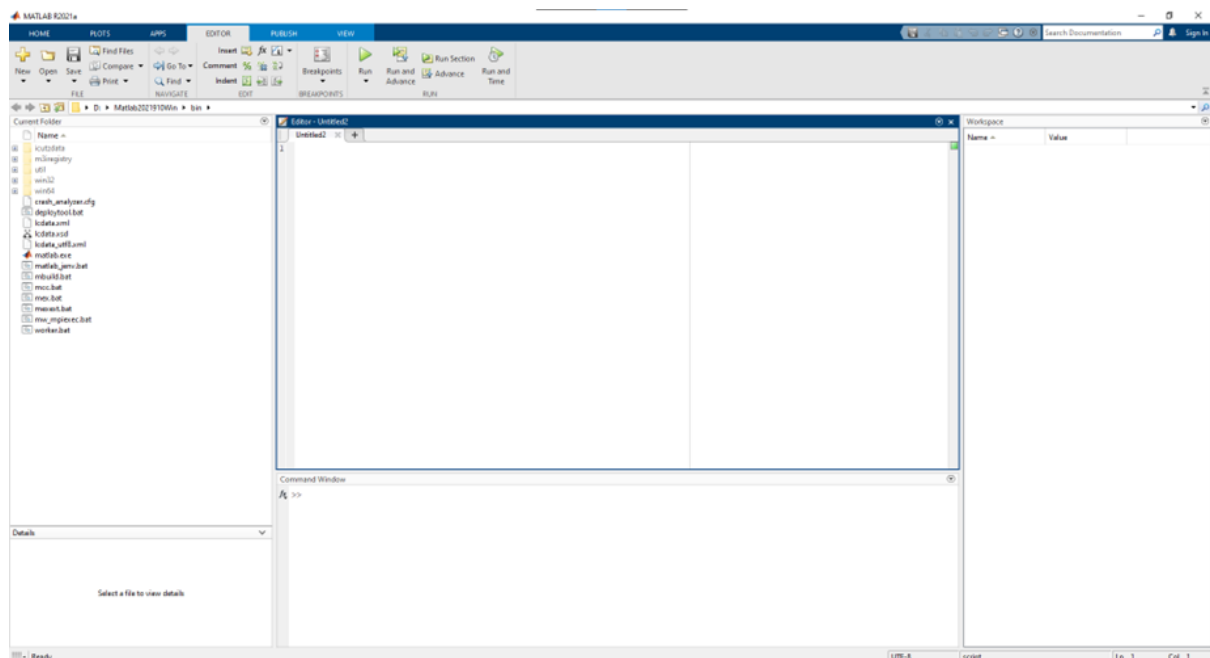
PROCEDIMIENTO DE LA PROGRAMACIÓN

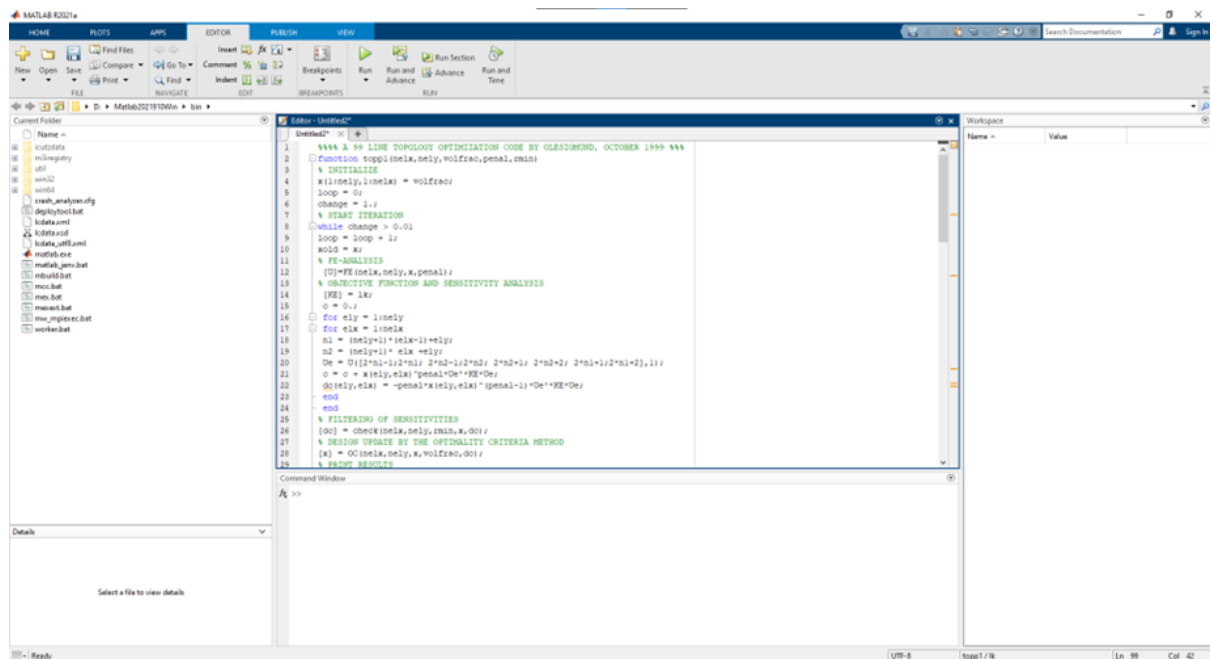
El código de Matlab está compuesto como un código de optimización topológica estándar, el cual está listo para ser interpretado por MATLAB luego de llevar a cabo la siguiente serie de sencillos pasos:

1) Tenemos que abrir MATLAB, esperar a que éste se inicie y muestre su pantalla principal.

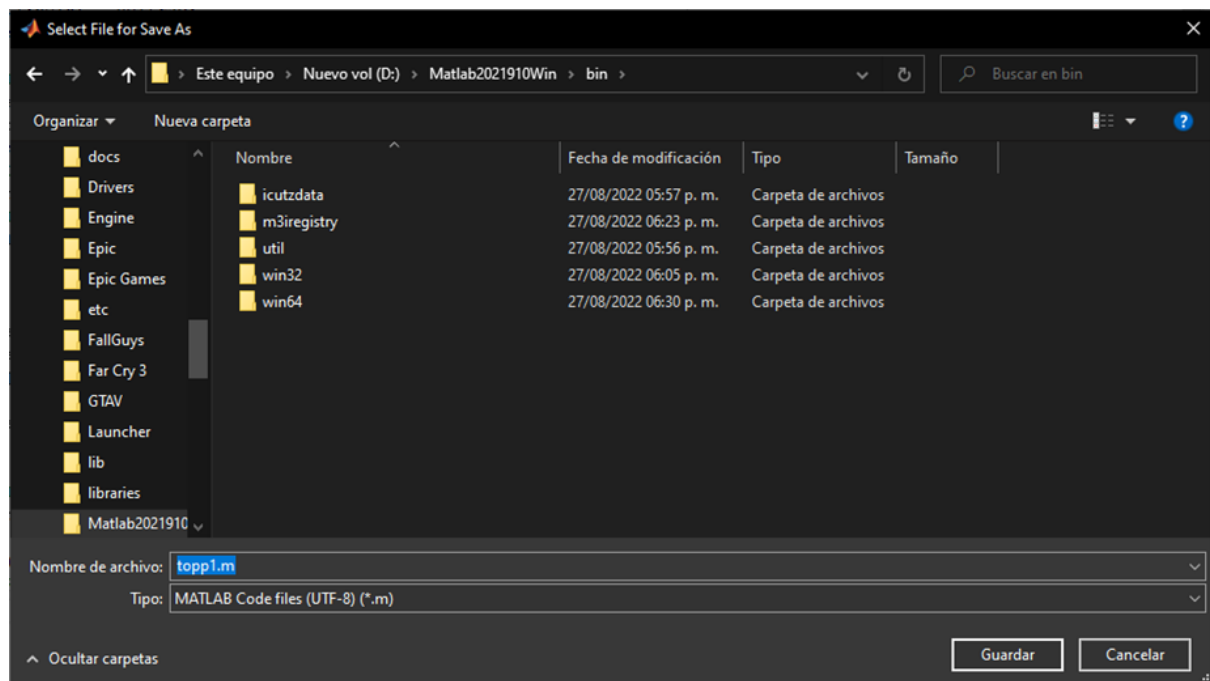


2) Una vez en la pantalla de inicio de MATLAB es necesario generar un script nuevo, con lo que se abre un editor de texto, dentro del cual escribiremos el código proporcionado.



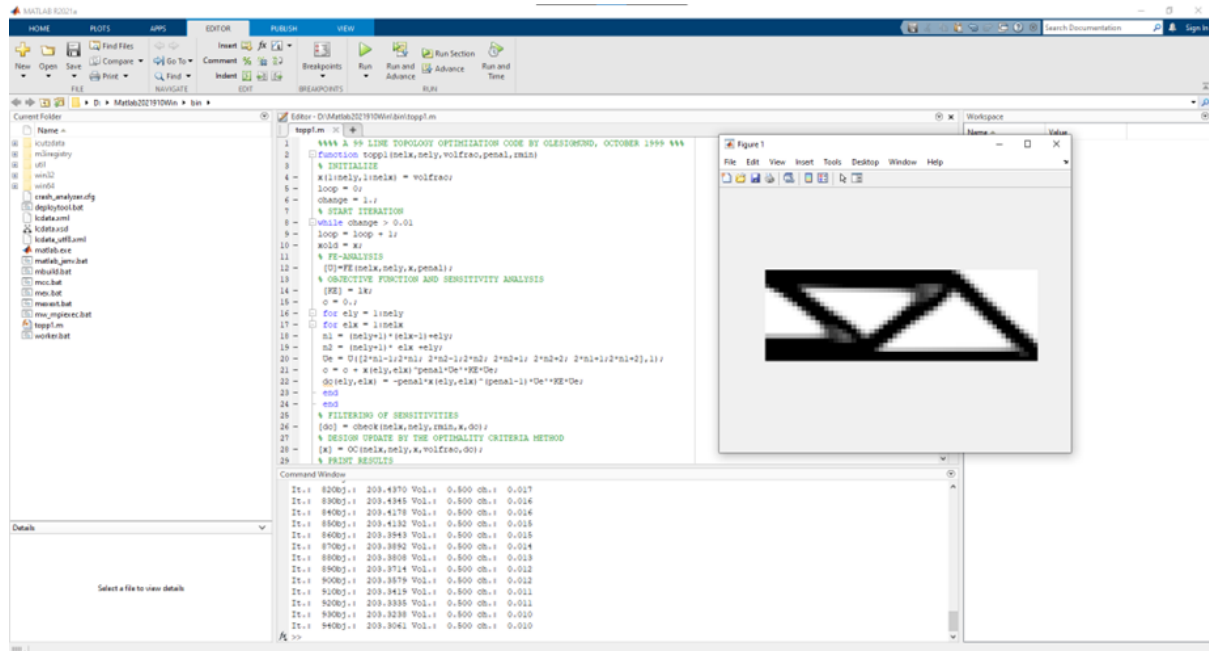


3) Una vez con el código completamente escrito en el editor de texto, se guardará el archivo, teniendo en cuenta la ubicación donde se va a guardar y el nombre que se le va a asignar.



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. El directorio actual del intérprete se encuentra en la barra de herramientas superior.

5) Por último, se ejecuta el programa desde la ventana de comando de MATLAB. El código que se proporcionó viene preparado para optimizar un dominio de diseño con cargas y restricciones, para ello escribimos “topp1(60,20,0.5,3.0,1.5)” en la línea de comando de MATLAB.



Codigo de 99 líneas

```
%% 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;
```

```
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;

```

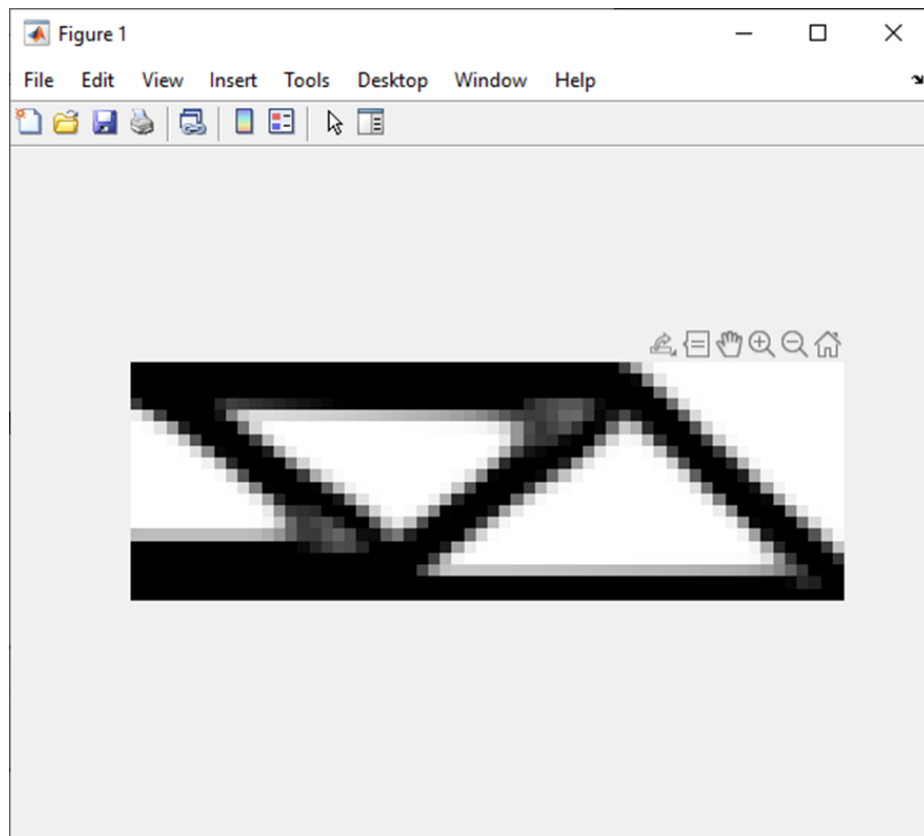
```

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 LOADSAND SUPPORTS(HALF MBB-BEAM)
F(2,1) = -1;
fixeddofs =union([1:2*(nely+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)];

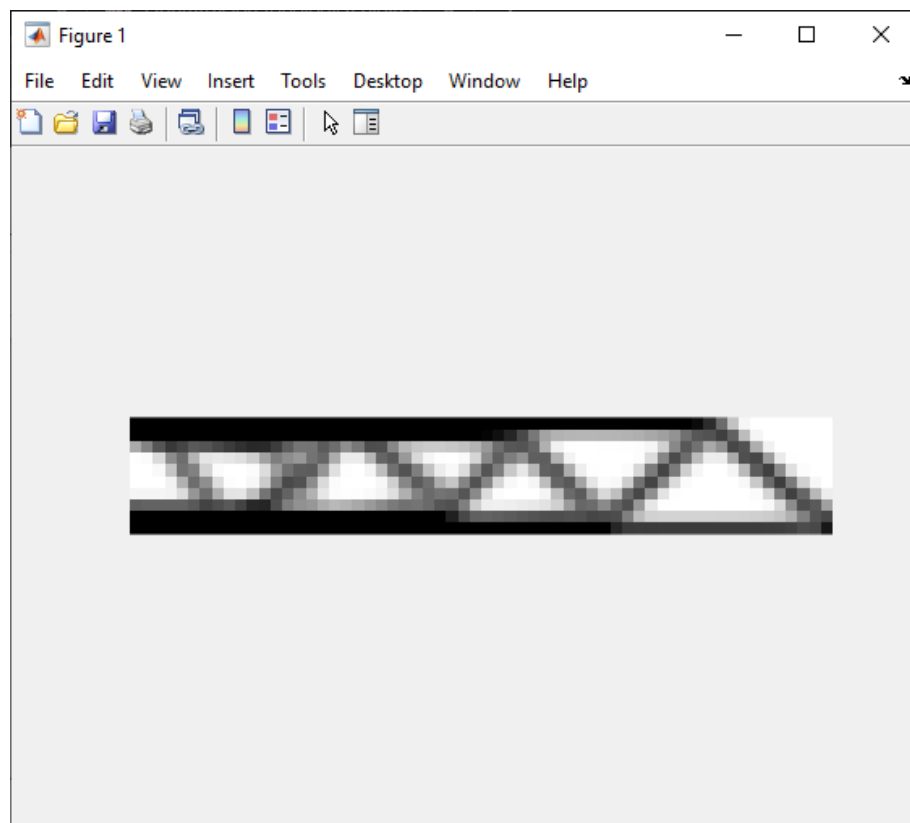
```

Implementacion en diferentes vistas

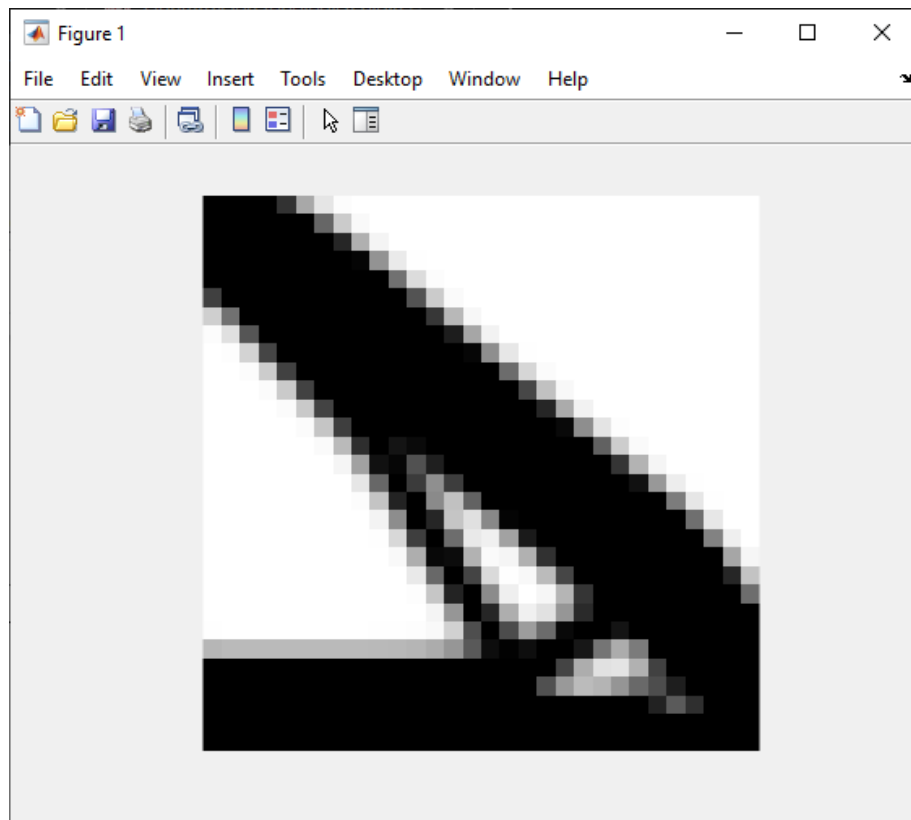
>> topp1(60,20,0.5,3.0,1.5)



>> topp1(60,10,0.5,3.0,1.5)



>> topp1(30,30,0.5,3.0,1.5)



CONCLUSIONES

Gabryiel Bailon 1869828

Se observó que la optimización topológica es un proceso relativamente nuevo e innovador en el campo del diseño, no solo en software sino tambien al momento de la realización de piezas y ensambles, ya que al momento de optimizar los materiales con los cuales estan hechas las piezas.

El uso del software MATLAB es primordial para la correcta interpretación del código de programación para llevar a cabo dicha tarea.

Carlos Eduardo Rivera López 1897545

Gracias a esta práctica número 1 pudimos conocer y comprender más acerca de la optimización topológica, para esta práctica utilizamos el software Matlab para emplear el código ya previamente visto, una vez puesto el código se analizó y se obtuvieron los datos necesarios, nos dimos cuenta que esto es importante para el desarrollo de piezas, la técnica de optimización y el análisis de elemento finito nos puede ayudar mucho en los procesos ingenieriles.

Víctor Adrián Higuera Vázquez 1876474

Conocer, entender y comprender la optimización topológica es muy importante porque ayuda mucho en un diseño mecánico por el análisis detallado que da al componente o estructura a evaluar para hacer más ligero la pieza mecánica manteniendo las principales funcionalidades. Hace más compacto el mecanismo sin modificar su funcionalidad, y así siendo importante para el área de micro y nanotecnología.

César Armando Luna Zapata 1844920

Buscando la evolución descubrimos un proceso de optimización topológica siendo este un proceso nuevo e innovador. Gracias a esta práctica aprendimos que podemos utilizar matlab para generar un análisis de elemento finitos.

Andrés Anaya Hernández 1914471

Al realizar esta práctica aprendimos a emplear funciones de matlab para poder llevar a cabo las operaciones correspondientes a la optimización topológica, usando como herramienta un código de 99 líneas y conocimos las funciones que tiene la optimización topológica a la hora del desarrollo de piezas mecánicas y así llegar a mejores construcciones.

