

# Práctica 1

1845056 Aurora Nahomy Martínez Pérez  
1854324 Kevin Orlando Huerta Jaramillo  
1896681 Jovanny Daniel Alvarado Ramírez  
1909876 Fernando Herrera Garza  
1991876 Bernardo Gil Villarreal

9 de septiembre de 2022

## 1. 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 ejecutara el análisis.

## 2. Introducción

En este reporte, se realizará la implementación del código de 99 líneas para la optimización topológica de una pieza, en nuestro caso una viga. Esto se hará en el software de Matlab que nos permite realizar este procedimiento, aunque tambien se pueden utilizar otros programas como Python u otro lenguajes de programación.

## 3. Marco Teórico

### 3.1. Optimización topológica

La optimización topológica es una técnica que pertenece al análisis estructural, y consiste, básicamente, en analizar un componente o estructura y, en función de como 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.

Pasos de la optimización topológica

1. Dibujar o Importar geometría
2. Simplificar la pieza y definir el espacio de diseño
3. Establecer uniones, juntas y contactos
4. Asignar materiales
5. Definir los casos de carga
6. Generar la optimización

7. Refinar la geometría
8. Exportar a CAD o generar STL
9. Verificar el rendimiento
10. Fabricar

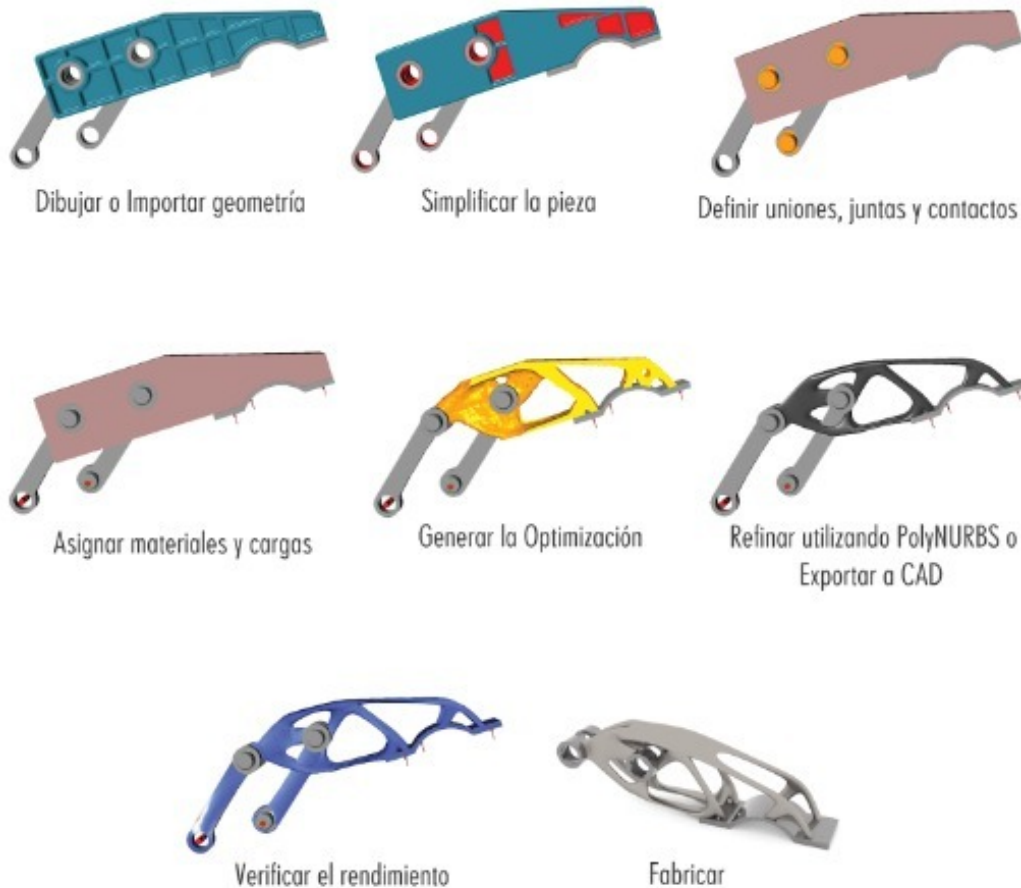


Figura 1: Pasos de la optimización topológica

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.

El desarrollo de esta metodología tiene un amplio campo de aplicación para las tecnologías de fabricación aditiva, como por ejemplo la fabricación SLM (Selective Laser Melting), debido a las grandes posibilidades en términos de diseño (geometrías muy complejas).

[1]

### 3.2. Matlab

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

## Características de Matlab

- Lenguaje de alto nivel para cálculos científicos y de ingeniería
- Entorno de escritorio optimizado para la exploración iterativa, el diseño y la solución de problemas
- Gráficas para visualizar datos y herramientas para crear diagramas personalizados
- Aplicaciones para ajustar curvas, clasificar datos, analizar señales, ajustar sistemas de control y muchas otras tareas
- Toolboxes complementarias para una amplia variedad de aplicaciones científicas y de ingeniería
- Herramientas para crear aplicaciones con interfaces de usuario personalizadas
- Interfaces para C/C++, Java®, .NET, Python, SQL, Hadoop y Microsoft® Excel®
- Opciones de implementación libres de derechos para compartir programas de MATLAB con los usuarios finales

## Ventajas

- Alta precisión
- Alto soporte de funciones
- Rápido prototipado
- Integración con dispositivos hardware.

## Desventajas

- La construcción de modelos requiere un tratamiento especial.
- Los resultados de la simulación pueden ser difíciles de interpretar.
- Un modelamiento del sistema y un análisis de los datos pueden consumir mucho tiempo y muchos recursos.

[2]

## 4. Desarrollo

### 4.1. Programa a realizar

Se realizó la optimización topológica de una viga, usando el código de optimización de 99 líneas en Matlab. La viga se modeló mediante elementos finitos de viga, y en los resultados se ven las deformaciones hechas.

### 4.2. Estado de arte

En la literatura, se pueden encontrar multitud de enfoques para la resolución de problemas de optimización de la topología. El documento original de Bendsoe y Kikuchi (1988) utilizó un enfoque basado en la microestructura u homogeneización basado en estudios de existencia de soluciones. El enfoque basado en la homogeneización se adaptó en muchos trabajos, pero tiene la desventaja de que la determinación y evaluación de las microestructuras y sus orientaciones es engorroso, si no se resuelve, y además, las estructuras resultantes no pueden construirse, ya que no hay una escala de longitud definida, ya que es asociada a las microestructuras. Mas sin embargo, el enfoque de homogeneización de la optimización topológica sigue siendo importante en el sentido de que puede proporcionar límites en el rendimiento teórico de las estructuras.

La fabricación aditiva, a diferencia de las técnicas de mecanizado tradicionales, permite producir piezas con geometrías complejas. El peso total se puede optimizar mediante un método digital denominado optimización por lógica. Esto también maximiza la resistencia mecánica de la pieza creada, la optimización topográfica es de hecho un campo del diseño digital que permite encontrar fórmulas matemáticas de la distribución óptima de materiales en un volumen determinado sometido a una tensión mecánica más o menos significativa.

El diseño implica designar dimensiones a un componente donde sea acorde a los procesos de manufactura. Considere el diseño mecánico como la selección de materiales y geometría tales que satisfagan los requerimientos funcionales implícitos. En este contexto el diseño de la estructura incluye información sobre la topología, la forma o el tamaño de la sutura y el método de distribución del material permite abordar los tres problemas simultáneamente.

El problema de disposición que se definirá a continuación de varias características del problema se dice una vez en la optimización del diseño estructural. El propósito de la optimización topológica es encontrar el diseño óptimo de una estructura dentro de una región específica. Las únicas cantidades conocidas en el problema son las cargas aplicadas, las posibles condiciones de apoyo, el volumen estructural y posiblemente algunas restricciones de diseño adicionales como por ejemplo la ubicación del tamaño de los huecos o las áreas.

La optimización topológica de estructura continua, es cuando se considera la optimización topológica a partir de un medio continuo, lo que se busca es la óptima configuración de la estructura, donde el dominio se discretiza en elementos finitos.

Las técnicas de solución de problemas a medida que ha ido evolucionando la optimización topológica, Su implementación en problemas con un mayor grado de complejidad son más comunes. Esto ha requerido el desarrollo de nuevas técnicas de solución. Dentro de las técnicas de solución se pueden mencionar el criterio de optimalidad OC, el método de la asíntota móvil MMA y el método de programación lineal PLS, entre otros.

### 4.3. Procedimiento de programación

#### 1. Ejecutar Matlab

Para poder ejecutar Matlab, buscamos nuestro acceso directo, que generalmente está ubicado en el escritorio de nuestra computadora. Para abrir Matlab, basta con dar doble clic en el icono seleccionado mostrado en la imagen.



Figura 2: Ejecutando Matlab

#### 2. Crear un nuevo script

Para realizar nuestra programación es necesario hacer un nuevo escrito o “script”. Para ello, al abrir Matlab, buscamos el apartado “FILE” o “ARCHIVO”, y damos un clic en el icono que dice “New script”, y al hacer esto se nos abrirá un nuevo espacio en el área de trabajo para hacer la programación.

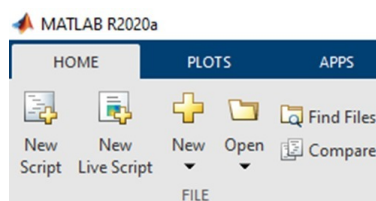


Figura 3: Creando un nuevo script

### 3. Redacción de nuestro código dentro del script

Ahora, bastará con copiar el código que se mostrará a continuación en nuestro nuevo script de Matlab.

```
function Practica1(nelx,nely,volfrac,penal,rmin);
x(1:20,1:60) = volfrac;
loop = 0;
change = 1.;
while change > 0.01
loop = loop + 1;
xold = x;
[U]=FE(nelx,nely,x,penal);
[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
[dc] = check(nelx,nely,rmin,x,dc);
[x] = OC(nelx,nely,x,volfrac,dc);
change = max(max(abs(x-xold)));
disp([' It.: ' sprintf('
' Vol.: ' sprintf('
' ch.: ' sprintf('
colormap(gray); imagesc(-x); axis equal; axis tight; axis
off;pause(1e-6);
end
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
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-floor(rmin),1):min(i+floor(rmin),nelx)
for l = max(j-floor(rmin),1):min(j+floor(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
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 = zeros(2*(nely+1)*(nelx+1),1);
for elx = 1:nelx
for ely = 1:nely
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
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);
U(freedofs,:) = K(freedofs,freedofs) \ F(freedofs,:);
U(fixeddofs,:)= 0;
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-nu2) * [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. Guardar nuestro script con la extensión (.m) correspondiente a matlab y compilar

Ahora es necesario guardar nuestro programa con un nombre en específico, asegurándonos que el nombre con el que vayamos a guardar el script y el nombre de la primera función de nuestro código sean iguales.

```

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 Practical(nelx,nely,volfrac,penal,rmin);
4  % INITIALIZE
5  x(1:20,1:60) = volfrac;
6  loop = 0;

```

Figura 4: Ejemplo sobre la situación del nombre

Como vemos en la imagen, el nombre de la primera función del script es “Practical1”, por lo tanto vamos a guardar el script con este mismo nombre, con la extensión (.m). Para guardar el script, vamos a la pestaña “EDITOR”.

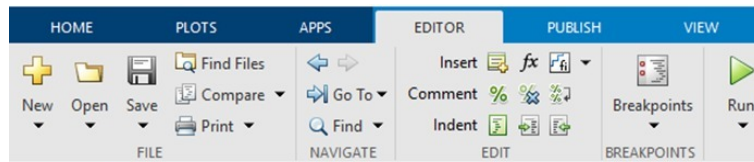


Figura 5: Se guarda con la extensión .m

Aquí vamos al apartado de “FILE” y seleccionamos el botón llamado “SAVE”. Una vez en esta ventana, escribimos el nombre “Practical1” y nos aseguramos que en “Tipo” esté la extensión (.m) seleccionada, y simplemente damos un clic en “Guardar”.

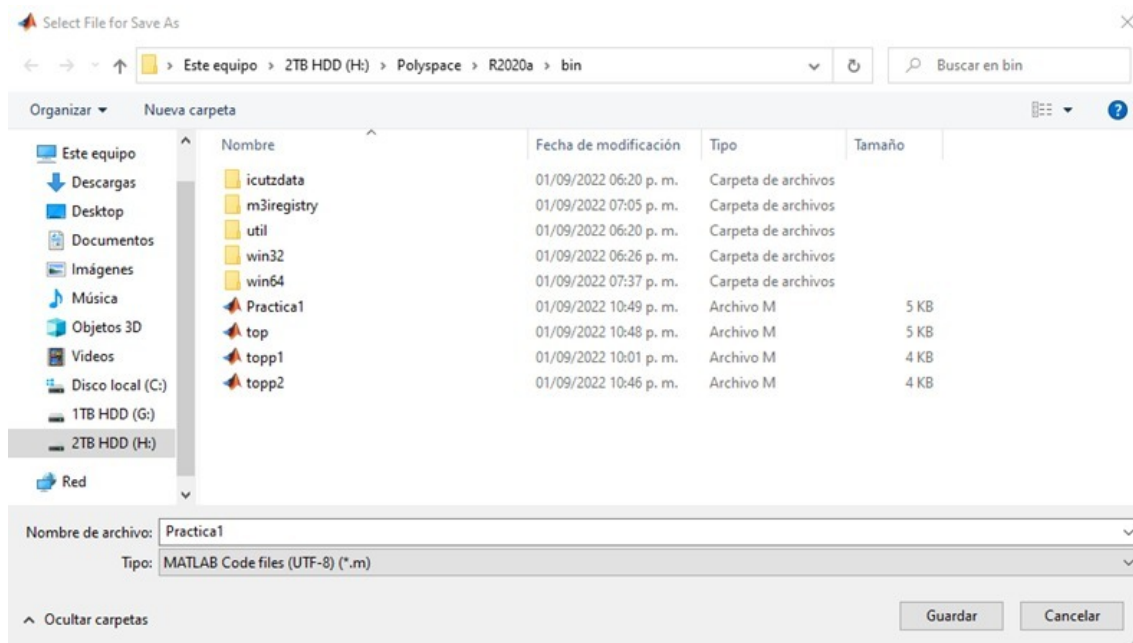


Figura 6: Guardando el archivo de código Matlab

Ahora podremos usar nuestro código para realizar el proceso de optimización en Matlab.

#### 4.4. Implementación

Para realizar la implementación del código en Matlab, usaremos los siguientes valores para los parámetros de la función Practica1, que son los siguientes. 'Practica1(60,20,0.5,3.0,1.5)'. Ahora escribiremos estos datos en una línea de comando que se encuentra abajo del código.

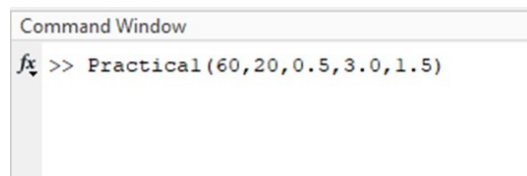


Figura 7: Parámetros de la función 'Práctica1'

Ahora solo tendremos que presionar la tecla Enter, y observar el procedimiento. En las siguientes imágenes vemos el inicio y el resultado final.



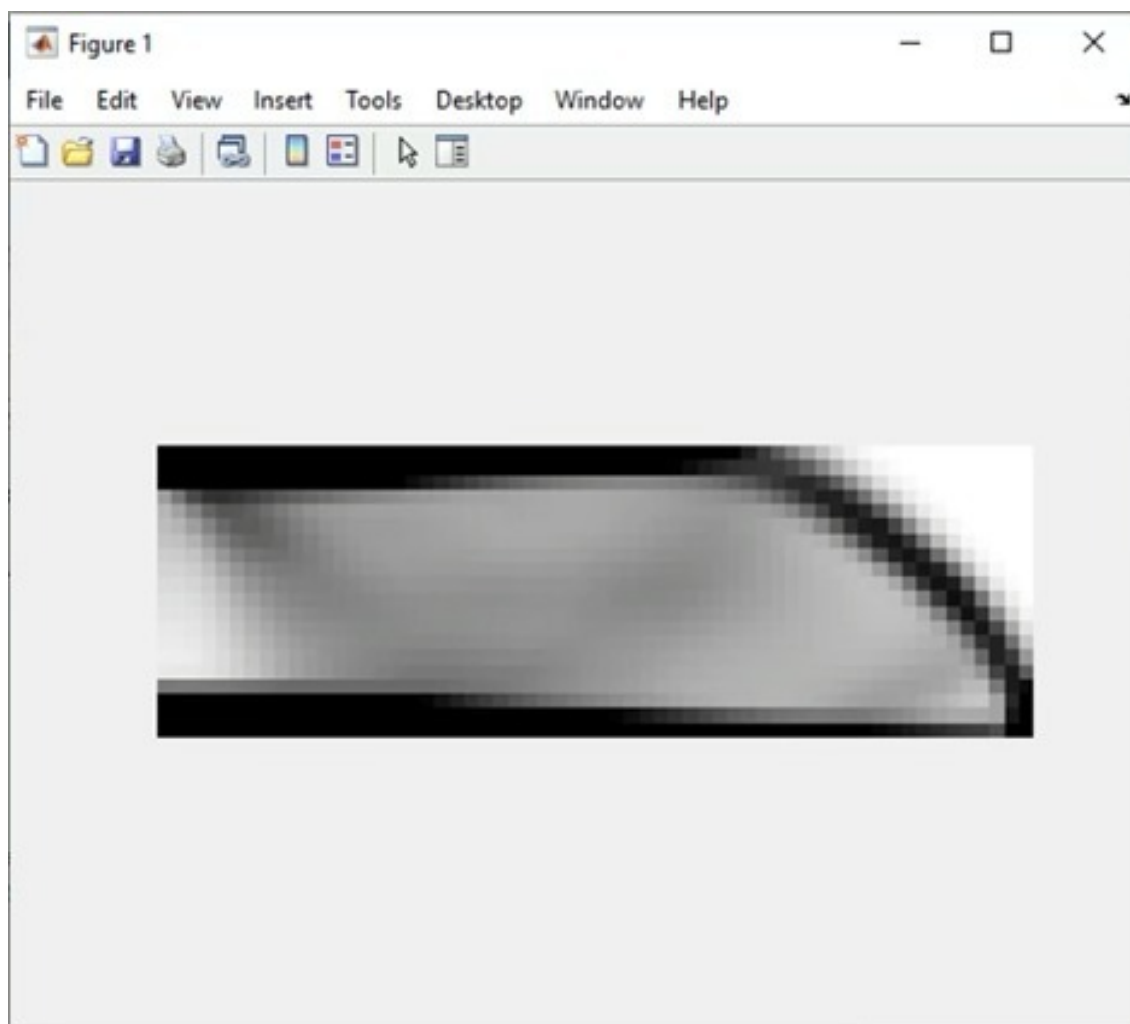


Figura 8: Antes de la optimización

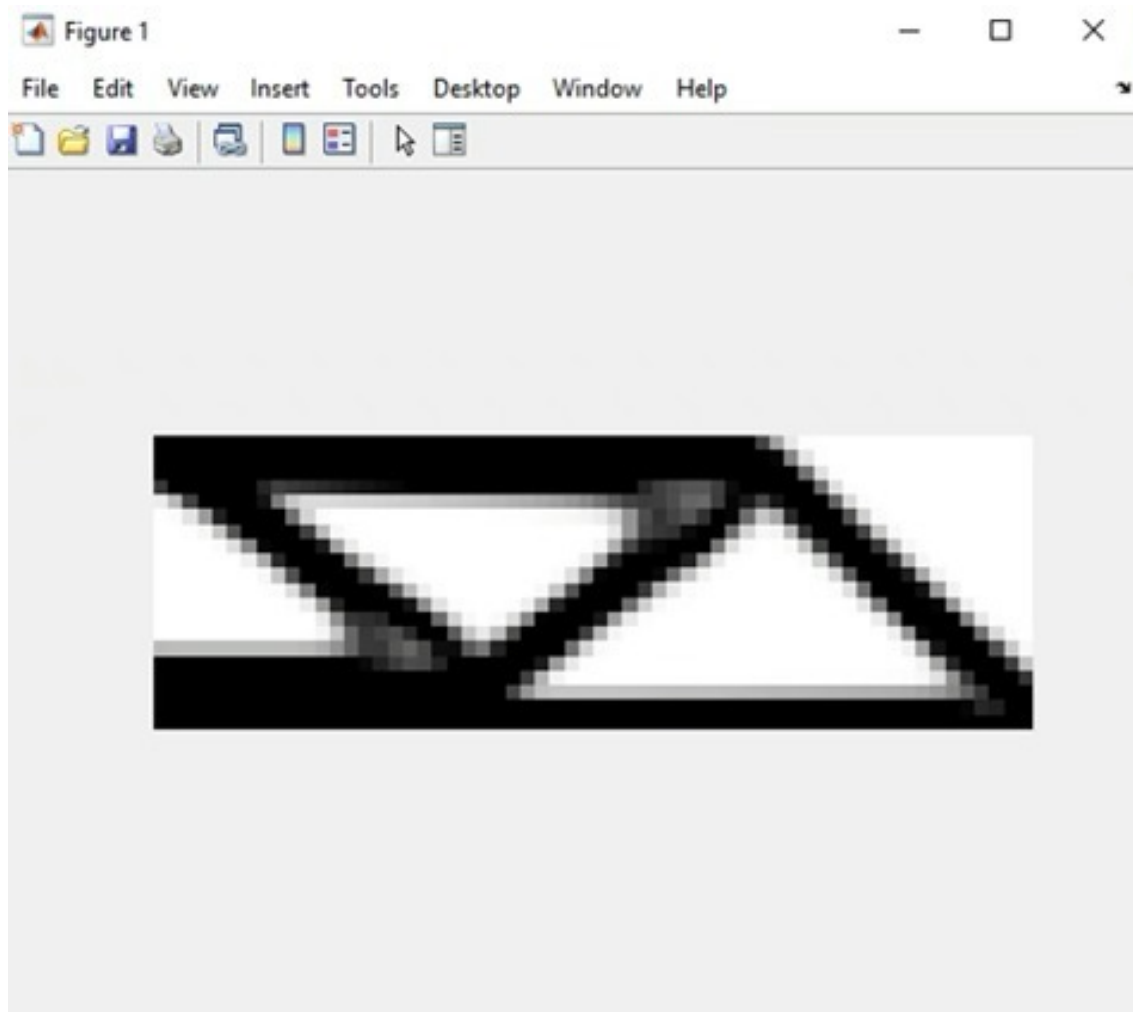


Figura 9: Después de la optimización

## 5. Conclusiones

- Bernardo Gil Villarreal 1991876

Gracias a la elaboración de esta práctica y en la realización del código en MATLAB, podemos observar cuales fueron las variables utilizadas en esta práctica de la cual nos permite la implementación de un código de optimización en dicho programa, el resultado de dicha codificación nos arroja una representación gráfica de la cual nos permitió comprobar los resultados esperados. Y gracias al conocimiento previo de esta práctica podemos desarrollar más piezas y observar si es viable realizar cierta pieza y poder llevarlo a lo práctico o más bien a lo físico, pudimos aplicar conocimientos de análisis de elemento finito y de programación de matlab para realizar la práctica y cumplir dicho objetivo.

- Jovanny Daniel Alvarado Ramírez 1896681

Un aspecto a tomar en cuenta del diseño mecánico es su forma. Gracias a esta práctica observamos lo que se refiere a optimización topológica, mediante el ejemplo realizado en Matlab notamos como el diseño inicial se eficientiza para lograr un diseño de una pieza final con menor cantidad de material y en teoría igual o más rígida. Esta elaboración de la optimización fue posible gracias al empleo de Matlab, pues el método que se emplea, siendo el de análisis de elementos finitos, no es posible desarrollarlo por el humano a gran detalle como lo hacen estos tipos de programas. Los métodos de diseño están en constante crecimiento, pues el diseño

generativo demanda cada vez más eficacia, por lo que el manejo de softwares como Matlab son obligatorios, y aquí recae la importancia de conocer el procedimiento para la ejecución de una optimización topológica.

- Fernando Herrera Garza 1909876

Como vimos, con el código en Matlab, se pudo realizar la optimización topológica de la pieza vista, que es nuestro caso fue una viga, generando un análisis de elemento finito para objetos que pueden son de utilidad en algunas areas de la ingeniería, y este mismo código se puede implementar para muchas otras piezas mas, cambiando los parámetros de la funcion del código. La realización de este reporte usando Matlab no fue complicada, solamente hay que conocer un poco de este software para comprenderlo bien y saber en que zonas de el programa hay que hacer las cosas.

- Kevin Orlando Huerta Jaramillo 1854324

En esta practica lo que se realizo fue mediante la utilización del software de Matlab, se pudo observar la optimización topológica de la pieza que se eligió en el equipo. Con la ayuda del software se observó como es que se puede hace eficiente el diseño de la pieza, optimizándola para poder elaborarla con la menor cantidad de material posible, y que pueda cumplir con su funcionamiento sin deformarse. La implementación de Matlab para el análisis de piezas en muy importante ya que se puede analizar cada propiedad de la pieza que se quiera realizar.

- Aurora Nahomy Martinez Perez 1845056

Durante esta práctica desarrollamos una pieza mediante el software matlab, en el cual pudimos observar el concepto de optimizacion topologica de una viga el cual fue modelado mediante elementos finitos por lo que se puede ver las deformaciones producidas en ella. En general considero que matlab es un software amigable y funcional para realizar cualquier tarea que lo requiera.

## Referencias

- [1] Estudio de ingenieria y tecnologia avanzado. optimizacion topologica.
- [2] Lenguaje Matlab. Matlab and simulink.