



**Universidad Autónoma de Nuevo León**  
***Facultad de Ingeniería Mecánica y Eléctrica***



## ***Laboratorio de biomecánica***

### **Práctica 1**

<b>Nombre</b>	<b>Carrera</b>
Cristian Arturo Garza Cavazos	IMTC
Edelmiro Eugenio García Sánchez	IMTC
Jesús Alberto Funes Mendoza	IMTC

**Brigada: 309**

**Semestre: Agosto-diciembre 2022**

***Instructor: Ing. Isaac Estrada***

**San Nicolas de los Garza Nuevo León, 21 de septiembre del  
2022**

## **Objetivo**

El estudiante conocer 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

## **Marco teórico**

La optimización topológica es un método que forma parte del análisis estructural. Se basa en el análisis mecánico de componentes o estructuras. Su objetivo principal es reducir el peso de la estructura manteniendo la función mecánica de la pieza de destino. Se basa en el análisis mecánico de componentes o estructuras. Su objetivo principal es reducir el peso de la estructura manteniendo la función mecánica de la pieza de destino. A diferencia de otras formas de optimización, la optimización de topología proporciona un nuevo concepto para el diseño estructural que se enfoca en aquellas aplicaciones donde el peso de los componentes es crítico (por ejemplo, aeroespacial). Debido a las enormes posibilidades de diseño (geometrías muy complejas), el desarrollo de este método tiene amplias áreas de aplicación en técnicas de fabricación aditiva, como la fabricación SLM (selective laser melting).

Pasos Optimización Topológica:

- Dibujar o Importar geometría
- Simplificar la pieza y definir el espacio de diseño
- Establecer uniones, juntas y contactos
- Asignar materiales
- Definir los casos de carga
- Generar la optimización
- Refinar la geometría
- Exportar a CAD o generar STL

Hay varios aspectos a considerar en el proceso de optimización de topología; el espacio de diseño, las condiciones de carga a las que se someterán las piezas, los materiales y las tecnologías relevantes que se van a producir, la reducción de costes

mediante la minimización de soportes y el uso de latas impresas en el caso de tecnologías aditivas y mucho más.

Estado del arte.

El proceso de optimización es la búsqueda del valor mínimo de una función, la llamada función objetivo, que modela la característica física que se requiere optimizar. El punto óptimo es el conjunto de valores de las variables de diseño que minimiza la función objetivo. En este punto la curva es plana y la primera derivada de la función con respecto a las variables de diseño es igual a cero. Una función objetivo tiene varios mínimos globales, si la función objetivo tiene varios mínimos locales, el algoritmo tiende a quedarse estancado en alguno de ellos y nunca alcanzar el mínimo global. Determinar si el punto óptimo alcanzado es un mínimo local o el mínimo global es un problema matemático complicado y muy dependiente del tipo de problema. Sin embargo, muchos de estos mínimos locales llevan a soluciones útiles como es el caso en este trabajo.

Hay un sinfín de ejemplos de como podemos aplicar la optimización topológica en la ingeniería, ya que gracias a ella podemos tener en claro ejemplos tanto en el ámbito biomecánico como industrial.

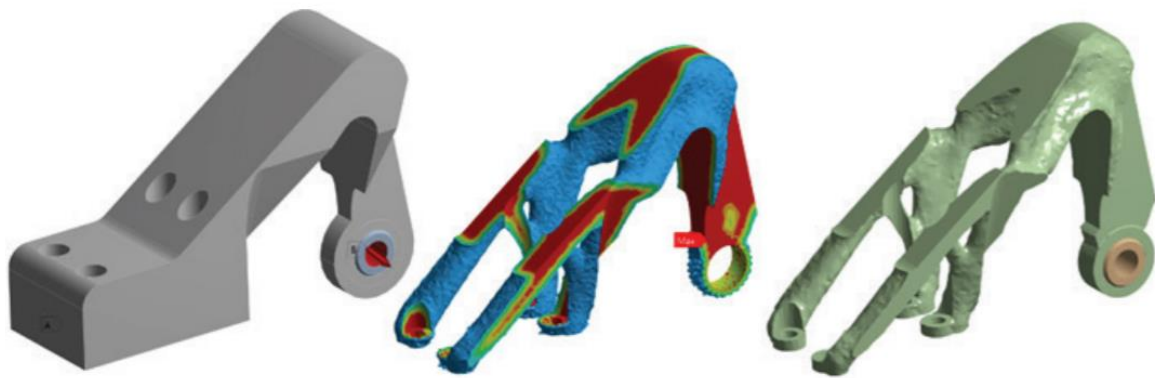


Imagen 1. En esta imagen podemos ver un ejemplo de la optimización topológica aplicada en la manufactura de piezas mecánicas.

Procedimiento de la programación.

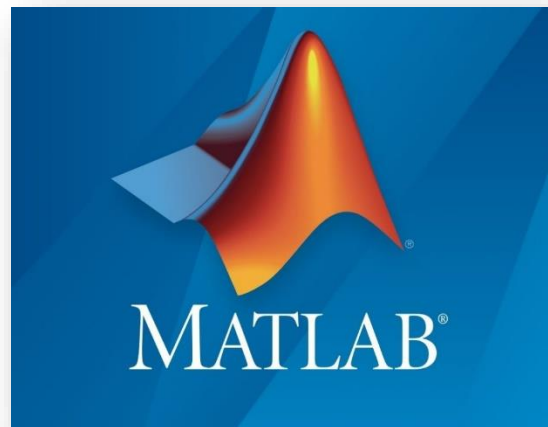
Matlab puede considerarse como un lenguaje de programación tal como C, Fortran, Java, etc. Algunas de las características de Matlab son:

- La programación es mucho más sencilla.

- Hay continuidad entre valores enteros, reales y complejos.
- La amplitud del intervalo y la exactitud de los números es mayor.
- Cuenta con una biblioteca matemática amplia.
- Abundantes herramientas gráficas, incluidas funciones de interfaz gráfica con el usuario.
- Capacidad de vincularse con los lenguajes de programación tradicionales.
- Transportabilidad de los programas.

Algunas de sus desventajas son

- Necesita de muchos recursos de sistema como son Memoria, tarjeta de videos, etc. para funcionar correctamente
- El tiempo de ejecución es lento.
- No genera código ejecutable.
- Es caro.



En el desarrollo de software, a menudo se utilizan técnicas de vanguardia.

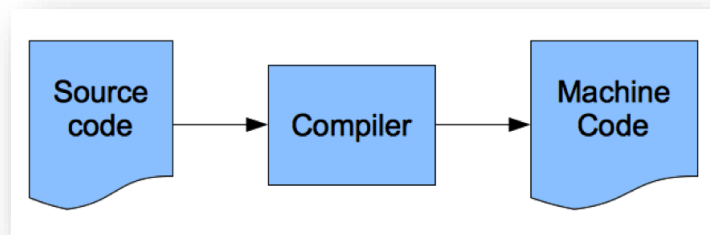
relacionadas con el desarrollo de lenguajes de programación. uno de ellos

El primer gran salto en esta dirección fue el desarrollo del primer lenguaje de alta calidad.

nivel, FORTRAN. Un aspecto del "estado del arte" tiene que ver con lo que decimos "Métodos formales" (MF).

En términos generales, los MF son técnicas mediante las cuales:

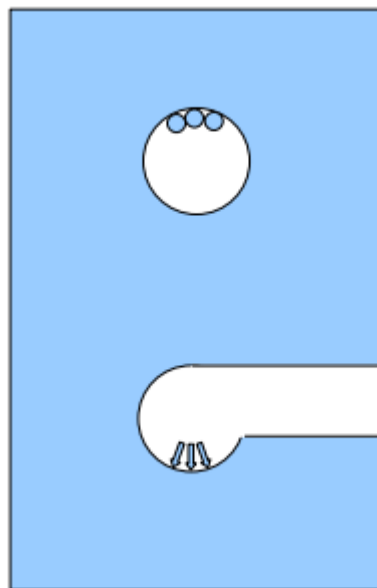
- Se escribe la especificación del sistema a desarrollar utilizando un lenguaje formal L1 (usualmente del paradigma declarativo), que luego es verificada con intervención humana y procesada mediante un compilador C1 para así generar el código en otro lenguaje formal L2 que representa un diseño de alto nivel.
- Este diseño escrito en el lenguaje L2 a su vez es verificado con intervención humana y procesado mediante un compilador C2 que genera el código en otro lenguaje formal L3 (usualmente del paradigma imperativo).
- Este código en el lenguaje L3 se procesa mediante un compilador C3 para obtener finalmente el sistema ejecutable.



Implementación de la programación en sus diferentes vistas.

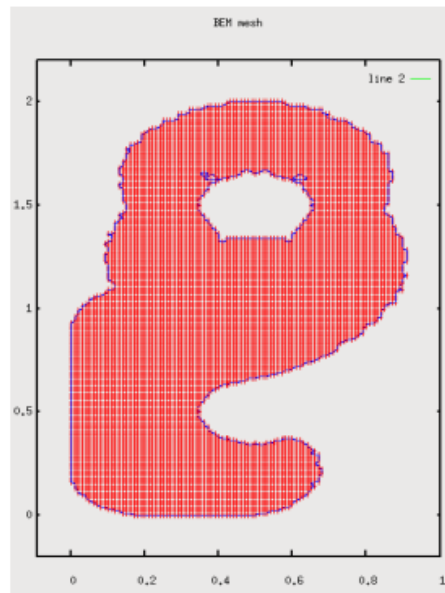
Ejemplo. Problema del gancho: en este ejemplo podremos ver como es aplicada la optimización topológica en un gancho de carga.

consiste en una placa rectangular de 2x1m con dos agujeros centrados simétricos y centrados en la chapa y una abertura para insertar el objeto a enganchar de la forma en la que se representa en la figura 2.0



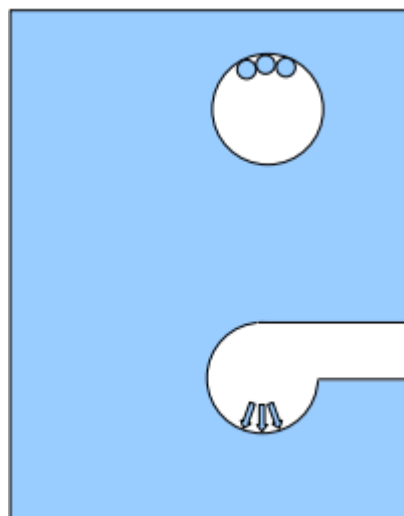
*Ilustración 1. figura 2.0*

Como solución inicial se obtuvo la siguiente forma.



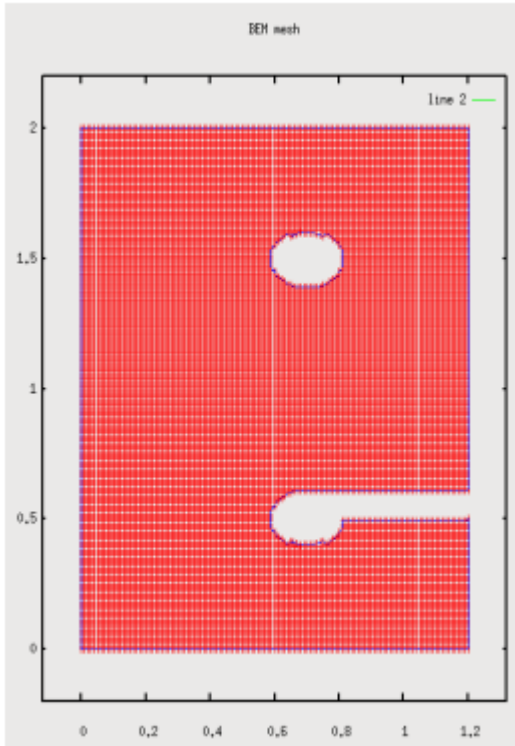
*Ilustración 2. figura 2.1*

En la figura 2.1 se puede observar como en la parte lateral izquierda hay una posible falta de material por lo que procedimos a cambiar la geometría inicial añadiendo material en dicha zona quedando el problema inicial de la siguiente forma (figura 2.0).

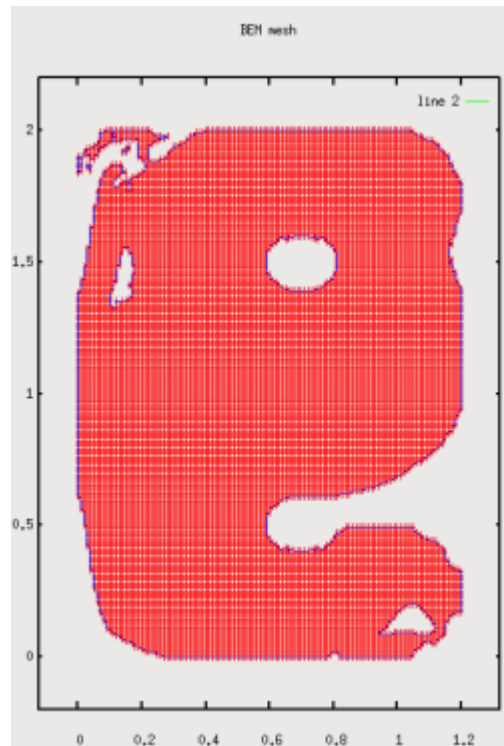


*Ilustración 3. figura 2.2*

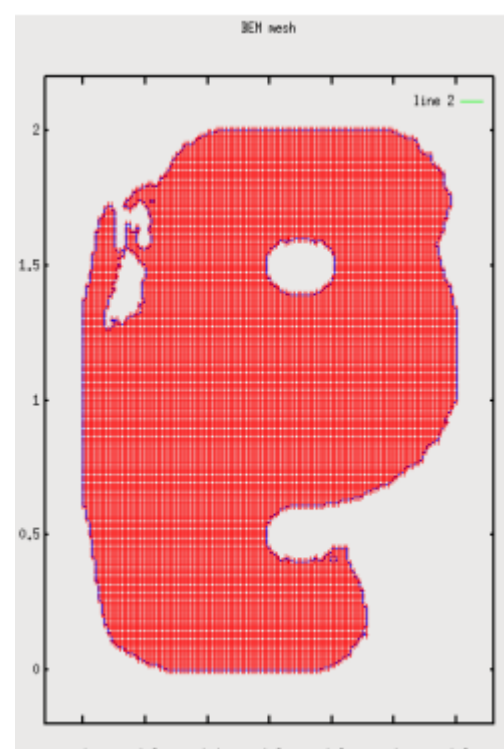
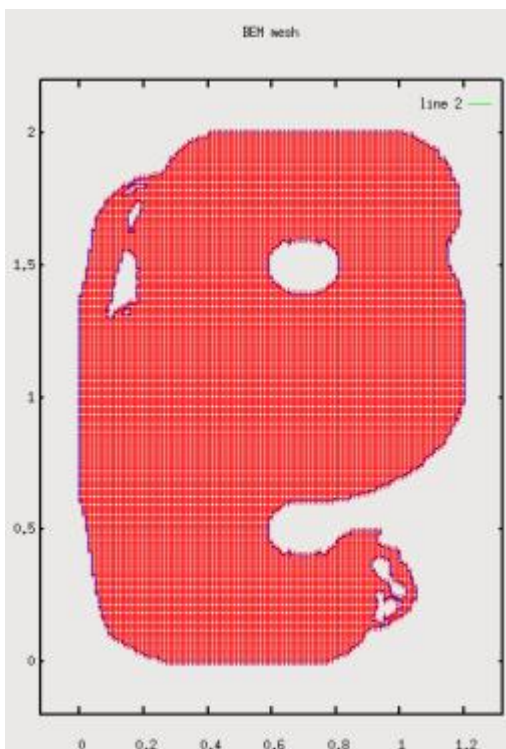
Una vez corrido el programa para la solución de la optimización topológica podemos ver los siguientes resultados.



*Ilustración 5. figura 2.3-iteración 1*



*Ilustración 4. figura 2.4- iteración 5*





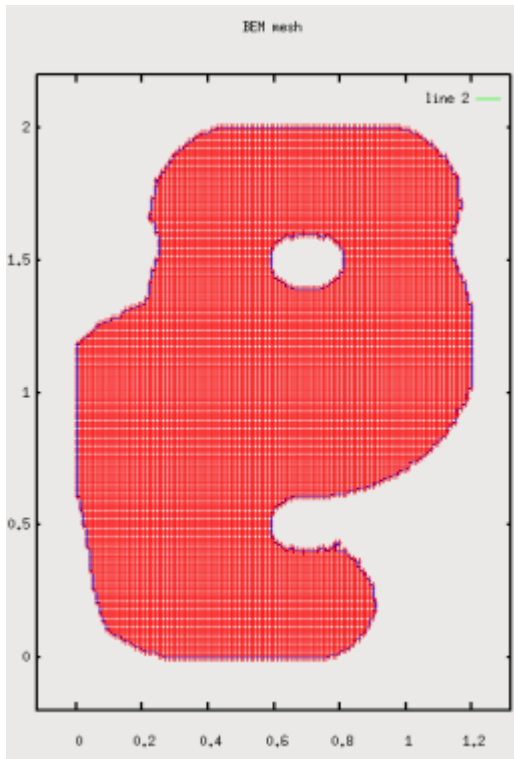


Ilustración 9. figura 2.7- iteración 20

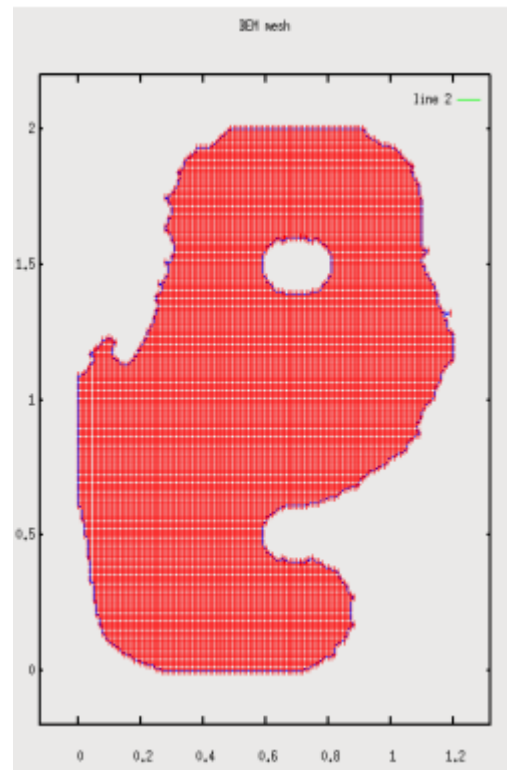


Ilustración 8. figura 2.8- iteración 25

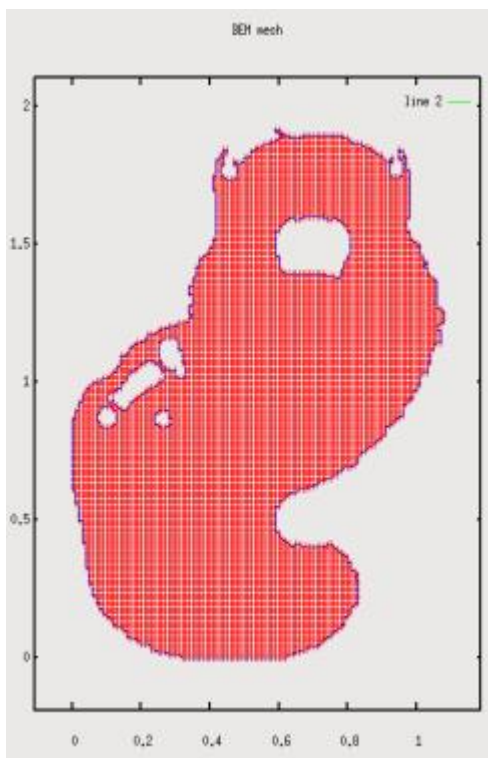


Ilustración 11. figura 2.9 – iteración 40

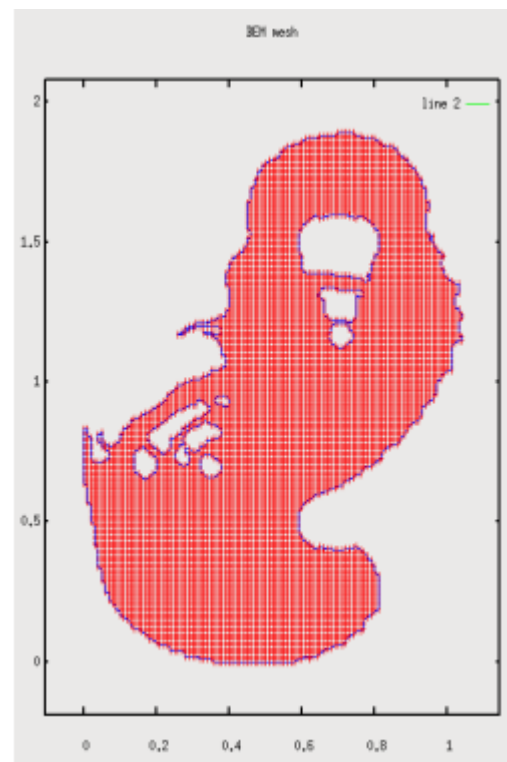


Ilustración 10. figura 2.10 – iteración 45





Ilustración 13. figura 2.11- iteración 60

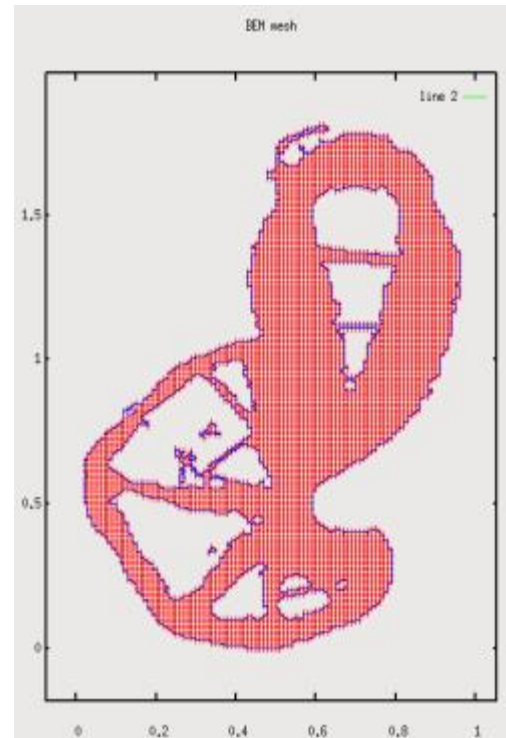


Ilustración 12. figura 2.12- iteración 65

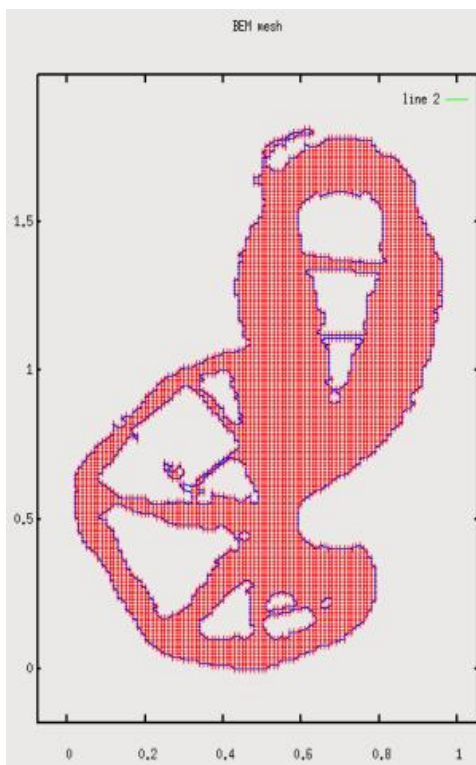
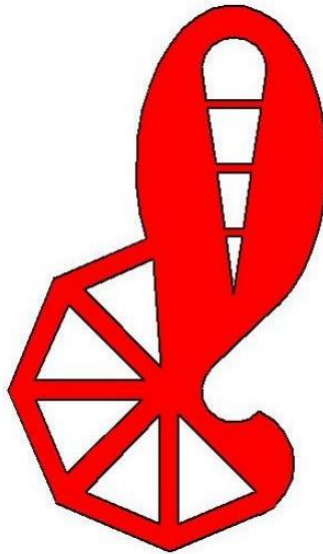


Ilustración 14. figura 2.13 iteración 70- solución final del gancho

En la figura 2.13 podemos observar la estructura final a la que llega el algoritmo del proyecto. Dicha estructura resultante podríamos dividirla en tres partes diferenciadas que serían:

- la correspondiente a la zona de agarre del gancho, zona superior, donde toma forma de elipse con una cuña interior con barras transversales y en donde se insertaría el gancho
- la correspondiente al cuerpo del gancho, zona inferior izquierda, que sería la encargada de soportar las mayores tensiones y que vemos que está formada por un anillo exterior unido con barras radiales al centro de la pieza o zona de la punta.
- la correspondiente a la punta, que el algoritmo no calcula y que se tendría que diseñar en función del objeto que quisiéramos colgar en ella.

Teniendo en cuenta las observaciones anteriores podríamos diseñar un gancho óptimo con una punta cualquiera como vemos en la figura 2.14.



*Ilustración 15. figura 2.14- diseño final del gancho*

Esto mismo puede llevarse a cabo en MATLAB ya que esta preparado para realizar la optimización topológica conociendo adecuadamente el procedimiento.

### Ejemplo en MATLAB.

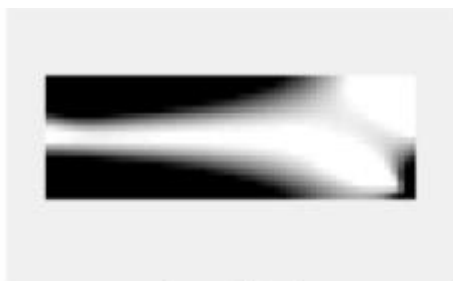
```
##### EJEMPLO DE OPTIMIZACIÓN TOPOLOGICA
##### PROGRAMA PRINCIPAL #####
function top(nelx,nely,volfrac,penal,rmin);
% INITIALIZE
x(1:nely,1:nelx) = volfrac;
loop = 0;
change = 1.;
% START ITERATION
while change > 0.01
    loop = loop + 1;
    xold = x;
% FE-ANALYSIS
    [U]=FE(nelx,nely,x,penal);
% 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
% FILTRO DE SENSIBILIDAD
[dc] = check(nelx,nely,rmin,x,dc);
% ACTUALIZACIÓN DEL DISEÑO POR EL MÉTODO DE CRITERIOS DE OPTIMALIDAD
[x] = OC(nelx,nely,x,volfrac,dc);
% IMPRESION DE RESULTADOS
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 )])
colormap(gray); imagesc(-x); axis equal; axis tight; axis off;pause(1e-
6);
end
##### CRITERIO DE OPTIMIZACIÓN #####
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
##### MALLADO #####
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
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
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
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);
% SOLVING
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-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)];

```

Resultados.



Iteración 1



Iteración 15



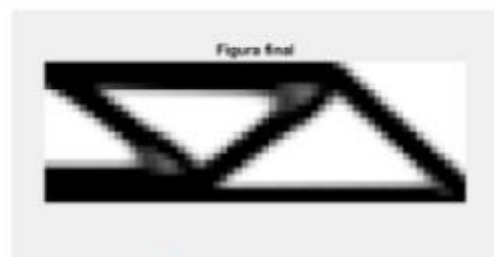
Iteración 27



Iteración 60



Iteración 80



Iteración 94

*Ilustración 16. figura 3.0- en esta imagen podemos ver como a una viga se le aplica el procedimiento antes visto en el gancho*

Conclusiones.

**Jesús Alberto Funes Mendoza** - en este trabajo nos enfrentamos a un reto ya que nunca habíamos visto con detenimiento lo que era la optimización topológica por lo que tuvimos que investigar mucho y además una vez entrando en MATLAB no habíamos visto como programar en este lenguaje para solucionar problemas de la optimización por lo cual también tuvimos que enfrentarnos a este.

Pero finalmente siento que en esta práctica aprendí mucho sobre cómo aplicar la optimización y para que nos puede servir a la hora de diseñar dispositivos.

**Cristian Arturo Garza Cavazos** - Al final de esta pasantía, uno puede comprender el concepto básico de optimización del terreno y cómo se usa ampliamente en varios campos de la ingeniería e incluso en la medicina moderna. Esto es en realidad la eliminación de la calidad innecesaria y, por lo tanto, abaratar el producto sin perder su carácter. Además de ejemplos de programación, este concepto nos brinda diferentes herramientas que podemos usar para crear diseños más optimizados para producir estas piezas de la mejor manera posible. Vimos un ejemplo en MATLAB, pero también se puede usar en una variedad de software, todos los cuales tienen el mismo propósito. En este ejemplo se ha elegido un determinado método de optimización, pero en realidad hay muchos otros, y dependerá de nuestro criterio saber cuál elegir.

**Edelmiro Eugenio Garcia Sanchez** -En esta practica tuvo como objetivo que era aprender sobre la optimización topológica que tuvimos que investigar para poder entender mucho mejor esta practica ya cuando tuvimos ejemplo claro vimos un ejemplo en diferentes programaciones pero en este caso usamos el programa de MATLAB que nos otorgo diferentes herramientas que podemos usar para crear diseños diferentes para la calidad de una pieza , para concluir este aprendemos diferentes tipos de programación diferente información de la optimización topológica