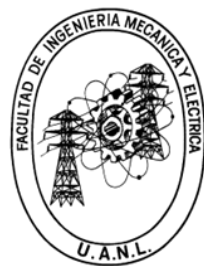




Universidad Autónoma de Nuevo León
Facultad de Ingeniería Mecánica y Eléctrica



Laboratorio de Biomecánica

Práctica #1:

Grupo: 109

Día: Lunes

Hora: N5

Equipo 7.

Nombre	Matrícula	Carrera
Raúl Delgado Saucedo	1796823	IMTC
Juan Alberto García Bautista	1895055	IMTC
Jorge Eduardo García Saavedra	1991794	IMTC

A 5 de septiembre de 2022.

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 ejecuta el análisis.

Programación y ejemplo de la geometría

La programación topológica es un método matemático que busca optimizar el diseño de un material dentro de un espacio de diseño dado, este es el volumen permitido en el cual se busca que exista el diseño, para un conjunto de condiciones como cargas, o situaciones del entorno, además de restricciones, con el fin de maximizar el rendimiento del sistema.

Existen varias formas de solucionar los problemas de autorización de forma eficiente, entre ellos técnicas como el criterio de optimalidad, el método de asíntotas móviles, y la programación lineal secuencial, cada uno con sus ventajas y desventajas. Por ejemplo, el primero te permite resolver rápidamente problemas sencillos, mientras que el último requiere más tiempo para ser procesado pero es útil para problemas más complejos.

En este caso, el autor utiliza el criterio de optimalidad en 12 líneas del programa, justo después de las 36 del programa principal.

El criterio de optimalidad determina la calidad o la presión de una solución, a partir de este criterio se obtiene la solución a problemas de optimización y esta solución se tiene cuando cuando las condiciones y restricciones del problema son satisfechas. En este código es utilizado el criterio de optimalidad a partir de los multiplicadores de Lagrange.

Después de las líneas de optimalidad le siguen 16 de filtro de mallado, para finalizar con 35 líneas para el código de elemento finito, y con esto se pueden resolver problemas bien planteados de optimización topológica.

Estado del arte

Encontrar la geometría del cuerpo ideal que minimice o maximice una función de costo específica mientras satisface las restricciones o condiciones límite de la tarea es un problema de ingeniería clásico. Hay dos formas de abordar la respuesta a este problema: como un desafío de optimización de forma o como un problema de optimización de topología.

El diseño implica designar dimensión a un componente donde se siguen reglas acorde a los procesos de manufactura. Considérese al diseño mecánico como la selección de materiales y geometría tales que satisfagan los requerimientos funcionales implícitos. Los ingenieros pueden plantear soluciones por el método canónico, pragmático, icónico (inspirado en estructuras semejantes del área ingenieril) o analógico (inspirado en formas externas de la ingeniería). En este contexto, el "diseño" de la estructura incluye información sobre la topología, la forma y el tamaño de la estructura y el método de distribución del material permite abordar los tres problemas simultáneamente.

Carretero Neches & Barroso Caro (2007) explican:

Una manera más general de controlar un dominio es mediante modificaciones de su topología, lo que permite obtener la configuración deseada partiendo de una morfología inicial distante de la óptima. Los métodos de homogeneización son posiblemente los más utilizados para la optimización topológica. Estos consisten en caracterizar la topología a través de su densidad, es decir, los huecos se identifican con regiones de densidad nula. De esta forma la solución del programa resulta en una distribución ficticia de material, siendo necesario en muchos casos utilizar métodos de penalización o filtros para obtener un resultado de utilidad ingenieril.

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 (EITA, 2019). El objetivo de la optimización topológica es identificar un diseño ideal en ausencia de conocimiento previo sobre la forma de la estructura ideal. A diferencia del caso de la optimización de forma tradicional, la topología de la estructura puede cambiar durante la optimización, como la adición de agujeros.

Desde que Bendsøe y Kikuchi desarrollaron los conceptos básicos en 1988, los problemas de optimización topológica se han planteado tradicionalmente mediante formulaciones de máxima rigidez. Con este tipo de planteamientos se pretende distribuir una cantidad predeterminada de material en un recinto de forma que se maximice la rigidez (se minimice la energía de deformación) de la pieza resultante para un determinado estado de carga .

El objetivo de la mejora topológica es minimizar la masa y maximizar la rigidez estructural donde la variables de diseño son los elementos finitos que alcanza su objetivo por medio de la iteración. Por otro lado, el objetivo de la optimización de forma es minimizar el esfuerzo máximo y homogeneizar los esfuerzos.

El problema de disposición que se definirá a continuación combina varias características de los problemas tradicionales 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 de la estructura que se va a construir y posiblemente algunas restricciones de diseño adicionales, como la ubicación y el tamaño de los huecos prescritos o las áreas sólidas.

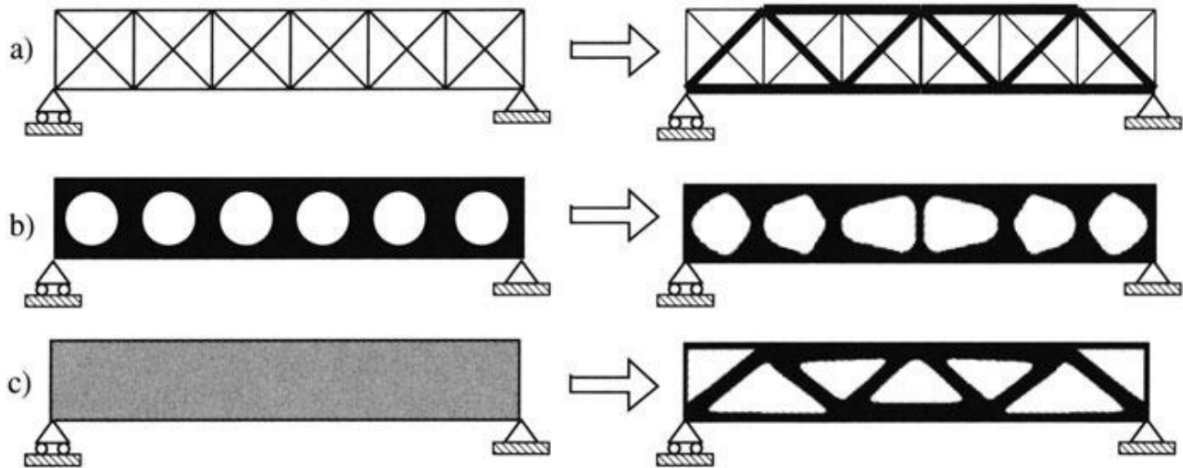


Figura 1.1. Tres categorías de optimización estructural. a) Optimización del tamaño de una estructura de truss, b) optimización de la forma y d) optimización de la topología. Los problemas iniciales se muestran en el lado izquierdo y las soluciones óptimas se muestran en el lado derecho (Bendsoe & Sigmund, 2013).

Matlab es una plataforma de programación diseñada específicamente para que ingenieros y científicos analicen y diseñen sistemas y productos que transforman nuestro mundo. El corazón de MATLAB es el lenguaje MATLAB, un lenguaje basado en matrices que permite la expresión más natural de las matemáticas computacionales.

Existe un artículo educacional del O. Sigmund llamado “A 99 line topology optimization code in matlab” donde explica un código de optimización topológica para la minimización de estructuras estáticas con carga. El código de optimización topológica de 99 líneas en Matlab que se utilizará en este laboratorio se divide en 36 líneas para la programación principal, 12 líneas para los criterios de optimización, 16 líneas para el filtro de mallado y 35 líneas para el código de elemento finito. De hecho, excluyendo las líneas de comentarios y líneas asociadas con la producción y el análisis de elementos finitos, el código resultante es de solo 49 líneas. Este código fue desarrollado por O. Sigmund, Department of Solid Mechanics, Building 404, Technical University of Denmark, DK-2800 Lyngby, Denmark. El código puede ser descargado desde la página del autor: <http://www.topopt.dtu.dk>.

En la literatura encontrada, se pueden encontrar múltiples formas de resolver problemas de optimización topológica. Una alternativa es llamada “power-law approach” o SIM (Solid Isotropic Material with Penalization). Aquí las propiedades del material se asumen como constantes dentro de cada elemento para discretizar el dominio del diseño y las variables son relativas a la densidad del elemento. Las propiedades del material se modelan como la densidad relativa del material elevada a alguna potencia multiplicada por las propiedades del material sólido. Este enfoque ha sido criticado ya que se argumentó que no existe ningún material físico con propiedades descritas por la interpolación de la ley de potencias. Aunque también hemos encontrado un artículo actualizado del 2010 del mismo autor de Implementacion de Matlab de Optimización de topología

Desarrollo.

El código es una función que tiene la siguiente estructura:

`op88(nelx,nely,volfrac,penal,rmin,ft)`

donde `nelx` y `nely` son el número de elementos en dirección horizontal y vertical, respectivamente, `volfrac` es la fracción de volumen prescrita f , `penal` es la potencia de penalización, `rmin` es el radio del filtro `rmin` (dividido por el tamaño del elemento), y el argumento adicional (en comparación con el código de 99 líneas) `ft` especifica si se debe utilizar el filtrado de sensibilidad ($ft = 1$) o el filtrado de densidad ($ft = 2$).

El viga MBB es un problema clásico en la optimización topológica. La viga MBB se usa aquí como ejemplo. El dominio de diseño, las condiciones de contorno y la carga externa para la viga MBB se muestran en la Fig. 1.2. El objetivo del problema de optimización es encontrar la distribución óptima del material, en términos de cumplimiento mínimo, con una restricción sobre la cantidad total de material.

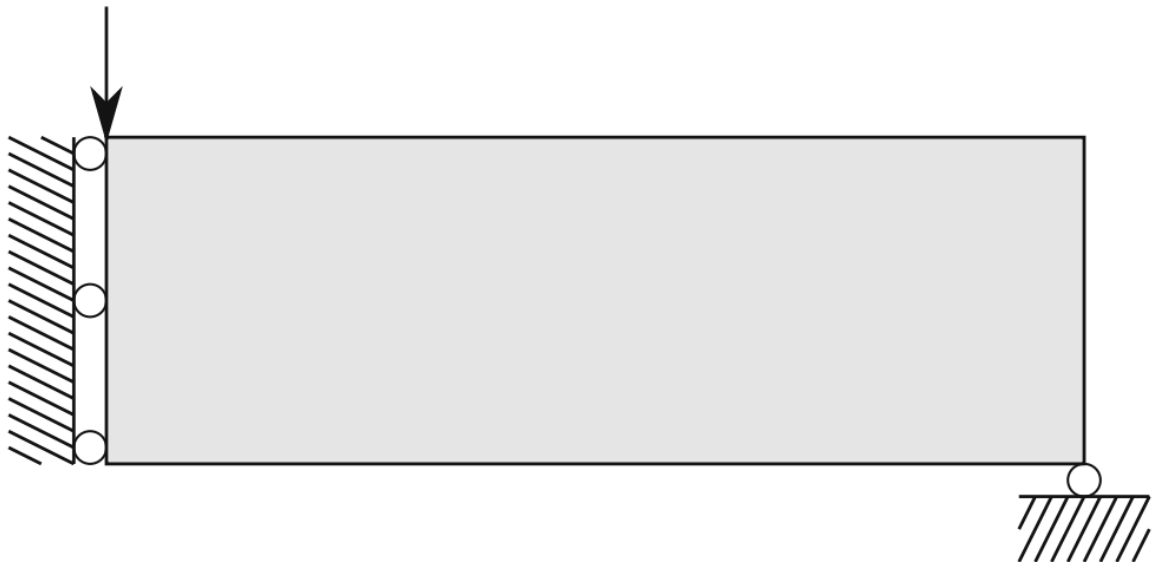


Fig. 1 El dominio de diseño, las condiciones de contorno y la carga externa para la optimización de una viga MBB simétrica.

El dominio de diseño se discretiza mediante elementos cuadrados finitos y se sigue un "enfoque basado en la densidad para la optimización de la topología".

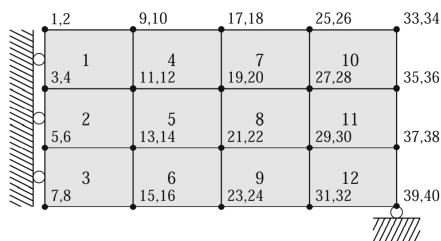
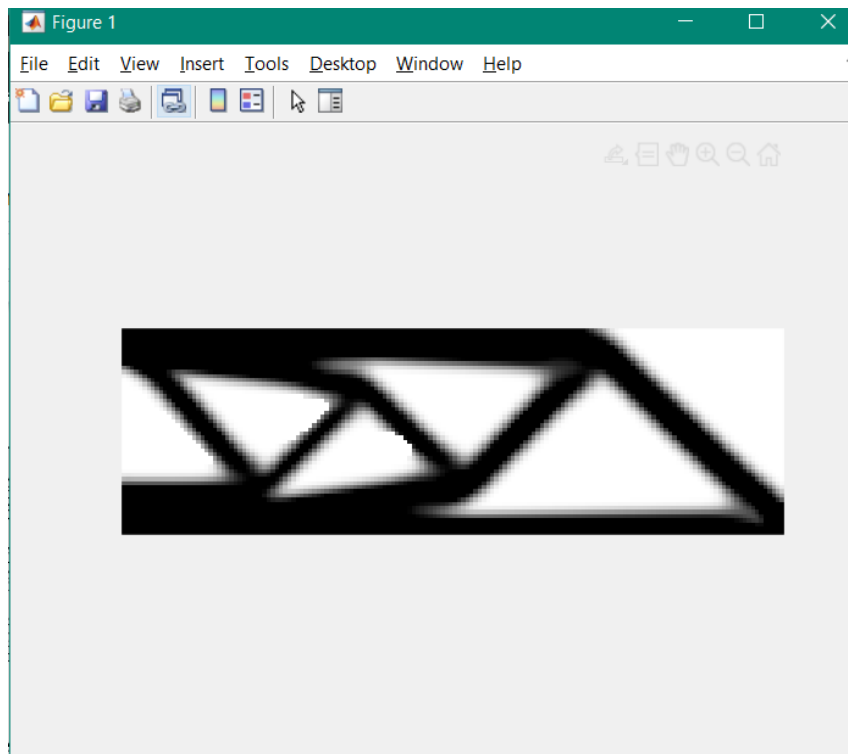


Fig. 1.3 El dominio de diseño con 12 elementos.
En nuestro caso usaremos los siguientes parámetros.
`top88(160,50,0.5,3.0,3.5,1)`

Se obtuvo la siguiente figura:



El código que se utilizó se muestra en la página siguiente:

```

%%% AN 88 LINE TOPOLOGY OPTIMIZATION CODE Nov, 2010 %%%
function top88(nelx,nely,volfrac,penal,rmin,ft)
%% MATERIAL PROPERTIES
E0 = 1;
Emin = 1e-9;
nu = 0.3;
%% PREPARE FINITE ELEMENT ANALYSIS
A11 = [12 3 -6 -3; 3 12 3 0; -6 3 12 -3; -3 0 -3 12];
A12 = [-6 -3 0 3; -3 -6 -3 -6; 0 -3 -6 3; 3 -6 3 -6];
B11 = [-4 3 -2 9; 3 -4 -9 4; -2 -9 -4 -3; 9 4 -3 -4];
B12 = [ 2 -3 4 -9; -3 2 9 -2; 4 9 2 3; -9 -2 3 2];
KE = 1/(1-nu^2)/24*([A11 A12;A12' A11]+nu*[B11 B12;B12' B11]);
nodenrs = reshape(1:(1+nelx)*(1+nely),1+nely,1+nelx);
edofVec = reshape(2*nodenrs(1:end-1,1:end-1)+1,nelx*nely,1);
edofMat = repmat(edofVec,1,8)+repmat([0 1 2*nely+[2 3 0 1] -2 -1],nelx*nely,1);
iK = reshape(kron(edofMat,ones(8,1))',64*nelx*nely,1);
jK = reshape(kron(edofMat,ones(1,8))',64*nelx*nely,1);
% DEFINE LOADS AND SUPPORTS (HALF MBB-BEAM)
F = sparse(2,1,-1,2*(nely+1)*(nelx+1),1);
U = zeros(2*(nely+1)*(nelx+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);
%% PREPARE FILTER
iH = ones(nelx*nely*(2*(ceil(rmin)-1)+1)^2,1);
jH = ones(size(iH));
sH = zeros(size(iH));
k = 0;
for i1 = 1:nelx
    for j1 = 1:nely
        e1 = (i1-1)*nely+j1;
        for i2 = max(i1-(ceil(rmin)-1),1):min(i1+(ceil(rmin)-1),nelx)
            for j2 = max(j1-(ceil(rmin)-1),1):min(j1+(ceil(rmin)-1),nely)
                e2 = (i2-1)*nely+j2;
                k = k+1;
                iH(k) = e1;
                jH(k) = e2;
                sH(k) = max(0,rmin-sqrt((i1-i2)^2+(j1-j2)^2));
            end
        end
    end
end
H = sparse(iH,jH,sH);
Hs = sum(H,2);
%% INITIALIZE ITERATION
x = repmat(volfrac,nely,nelx);
xPhys = x;
loop = 0;
change = 1;
%% START ITERATION
while change > 0.01
    loop = loop + 1;
    %% FE-ANALYSIS

```

```

sK = reshape(KE(:)*(Emin+xPhys(:)'.^penal*(E0-Emin)),64*nex*nely,1);
K = sparse(iK,jK,sK); K = (K+K')/2;
U(freedofs) = K(freedofs,freedofs)\F(freedofs);
%% OBJECTIVE FUNCTION AND SENSITIVITY ANALYSIS
ce = reshape(sum((U(edofMat)*KE).*U(edofMat),2),nely,nex);
c = sum(sum((Emin+xPhys.^penal*(E0-Emin)).*ce));
dc = -penal*(E0-Emin)*xPhys.^(penal-1).*ce;
dv = ones(nely,nex);
%% FILTERING/MODIFICATION OF SENSITIVITIES
if ft == 1
    dc(:) = H*(x(:).*dc(:))./Hs./max(1e-3,x(:));
elseif ft == 2
    dc(:) = H*(dc(:)./Hs);
    dv(:) = H*(dv(:)./Hs);
end
%% OPTIMALITY CRITERIA UPDATE OF DESIGN VARIABLES AND PHYSICAL DENSITIES
l1 = 0; l2 = 1e9; move = 0.2;
while (l2-l1)/(l1+l2) > 1e-3
    lmid = 0.5*(l2+l1);
    xnew = max(0,max(x-move,min(1,min(x+move,x.*sqrt(-dc./dv/lmid)))));
    if ft == 1
        xPhys = xnew;
    elseif ft == 2
        xPhys(:) = (H*xnew(:))./Hs;
    end
    if sum(xPhys(:)) > volfrac*nex*nely, l1 = lmid; else l2 = lmid; end
end
change = max(abs(xnew(:)-x(:)));
x = xnew;
%% PRINT RESULTS
fprintf(' It.:%5i Obj.:%11.4f Vol.:%7.3f ch.:%7.3f\n',loop,c, ...
    mean(xPhys(:)),change);
%% PLOT DENSITIES
colormap(gray); imagesc(1-xPhys); caxis([0 1]); axis equal; axis off; drawnow;
end
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% This Matlab code was written by E. Andreassen, A. Clausen, M. Schevenels,%
% B. S. Lazarov and O. Sigmund, Department of Solid Mechanics, %
% Technical University of Denmark, %
% DK-2800 Lyngby, Denmark. %
% Please sent your comments to: sigmund@fam.dtu.dk %
%
% The code is intended for educational purposes and theoretical details %
% are discussed in the paper %
% "Efficient topology optimization in MATLAB using 88 lines of code, %
% E. Andreassen, A. Clausen, M. Schevenels, %
% B. S. Lazarov and O. Sigmund, Struct Multidisc Optim, 2010 %
% This version is based on earlier 99-line code %
% by Ole Sigmund (2001), Structural and Multidisciplinary Optimization, %
% Vol 21, pp. 120--127. %
%
% The code as well as a postscript version of the paper can be %

```


Conclusión

El arte de la estructura es dónde poner los agujeros. La tecnología ahora está bien establecida y los diseños obtenidos con el uso de métodos de optimización de topología están en producción diariamente. El uso eficiente de los materiales es importante en muchos entornos diferentes. La industria aeroespacial y la industria automotriz, por ejemplo, aplican la optimización del tamaño y la forma al diseño de estructuras y elementos mecánicos. La optimización de la forma también se utiliza en el diseño de dispositivos electromagnéticos, electroquímicos y acústicos. En su configuración más general, la optimización de la forma de estructuras continuas debería consistir en una determinación para cada punto en el espacio si hay material en ese punto o no. Alternativamente, para una discretización FEM, cada elemento es un vacío potencial o un miembro estructural. En este escenario, la topología de la estructura no está fijada a priori, y la formulación general debe permitir la predicción del diseño de una estructura. El campo de la optimización estructural combina la mecánica, el cálculo variacional y la programación matemática para obtener mejores diseños de estructuras. En este trabajo no se trató el tema de las fórmulas que le dan vida al código de matlab, sin embargo se pueden ver dentro del código y nos demuestran que los softwares de hoy en día nos apoyan mucho con cálculos e impresiones que nos facilitan el poder generar nuevas ideas e ir más rápido en nuestras investigaciones.

Referencias

EITA. (2019). *Optimización Topológica - Estudio de Ingeniería y Tecnología Avanzada S.L.*

Estudio de Ingeniería Y Tecnología Avanzada S.L.

<https://eitaingenieros.com/optimizacion>

Carretero Neches, L. & Barroso Caro, A. (2007). Optimización topológica de problemas elásticos planos utilizando el método de los elementos de contorno [proyecto fin de carrera]. s.n.

Bendsoe, M. P., & Sigmund, O. (2013). *Topology Optimization*. Springer Publishing.

Sigmund, O. (2001). A 99 line topology optimization code written in Matlab. *Structural and Multidisciplinary Optimization*, 21(2), 120–127. <https://doi.org/10.1007/s001580050176>

