

Práctica 1

Brigada: 109

Emanuel Salas de León 1854873
Jason Méndez Muñoz 1889560
Jesús Mario Duarte Salinas 1907630
Lorena Alejandra Campos Carrazco 1909279
Yair Obed Morales Ortiz 1992266

5 de septiembre de 2022

1. Marco Teórico

La optimización topológica es un método matemático que optimiza el diseño y la estructura del material dentro de un espacio de diseño geométrico 3D dado, para un conjunto definido de reglas establecidas por el diseñador. El objetivo es maximizar el rendimiento de la pieza mediante el modelado matemático y la optimización de factores como fuerzas externas, condiciones de carga, condiciones de contorno, restricciones y propiedades del material dentro del diseño.

La optimización topológica convencional utiliza el análisis de elementos finitos para evaluar el rendimiento del diseño y producir estructuras para satisfacer objetivos como los siguientes:

- Relación rigidez-peso reducida
- Mejor relación entre tensión y peso
- Volumen de material reducido en función del factor de seguridad,
- Frecuencia natural en función al peso

Este 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, biomédica, entre otras. Además, tiene un papel muy importante en el campo de las micro y nanotecnologías, principalmente, en el diseño de mecanismos flexibles.

Para esta práctica vamos a realizar un ejercicio como el siguiente. La figura de abajo muestra una viga de dominio bidimensional (Ω) 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) sería: ¿Cuál es la distribución de material en el dominio (Ω) que proporciona la máxima rigidez (o mínima flexibilidad) para el estado de carga y la restricción de volumen determinados?

Por medio de la optimización topológica podemos solucionar numéricamente el problema, por medio de algoritmos iterativos que maximizan o minimizan una función objetivo (o de costo). La parte inferior de la Figura muestra la topología óptima obtenida para el estado de carga mostrado y el volumen final de la estructura igual al 40 del volumen inicial. El resultado mostrado en la Figura es una estructura familiar. Muchos puentes de barras articuladas presentan esta geometría, llamada viga Messerschmitt-Bölkow-Blohm (MBB).

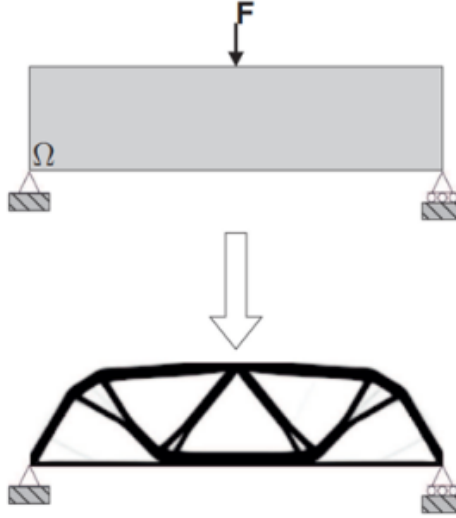


Figura 1: *Problema máxima rigidez con extensión de volumen*

2. Desarrollo

2.1. Programa a realizar

Optimización topológica de una viga sobre 2 soportes sujeta a una fuerza central, mediante el código optimización topológica de 88 líneas

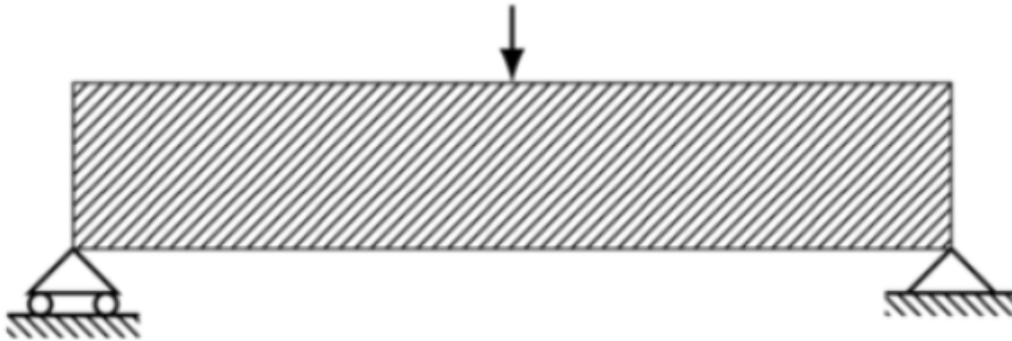


Figura 2: *Problema a realizar*

2.2. Estado del arte

Sigmund, O. A 99 line topology optimization code written in Matlab. En este artículo educativo el autor nos presenta una implementación en Matlab de un código de optimización topológica para la minimización de la conformidad de estructuras cargadas estáticamente. Dicho código está compuesto por 99 líneas donde están divididas en 36 líneas para el programa principal, 12 líneas para el optimizador basado en criterios de optimización, 16 líneas para un filtro de dependencia de malla y 35 líneas para el código de elementos finitos.

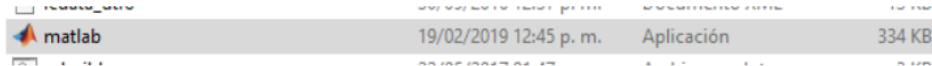
En la literatura, se pueden encontrar multitud de enfoques para la resolución de problemas de optimización de la topología. En el documento original de Bendsøe y Kikuchi (1988) se utilizó un enfoque basado en la microestructura u homogeneización basado en estudios de existencia de soluciones. El enfoque basado en la homogeneización se ha

adoptado en muchos trabajos, pero tiene la desventaja de que la determinación y evaluación de las microestructuras óptimas y sus orientaciones es engorroso, si no se resuelve (para problemas de incumplimiento) y, además, las estructuras resultantes no pueden construirse, ya que no hay una escala de longitud definida está asociada a las microestructuras. 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.

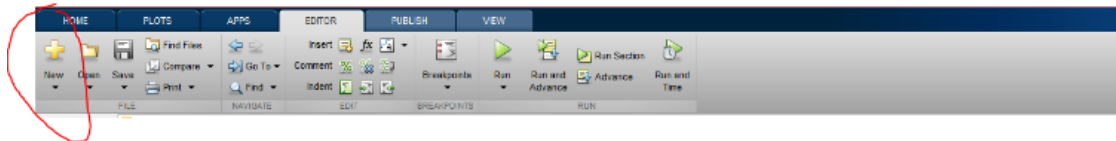
Abad R. Implementación en Matlab de un procedimiento de optimización topológica para estructuras con cargas térmicas. En este trabajo el autor realizó un estudio de la técnica de optimización topológica utilizada en el diseño de estructuras. Este estudio se centra en estructuras sometidas a cargas térmicas uniformes, por lo que lo que se busca es la topología que llegue a ofrecer una conducción óptima del calor. El autor implementa el problema en un código del programa de Matlab, para la resolución de estructuras, y realiza un amplio análisis de los resultados obtenidos mediante este programa para diferentes condiciones. Nos menciona que la metodología empleada para el cálculo se basa en el Método de los Elementos Finitos y utiliza un Criterio de Optimalidad para optimizar la conducción de calor.

2.3. Procedimiento de la programación

1. Ejecutar Matlab



2. Crear un nuevo script



3. Escribir nuestro código dentro del script

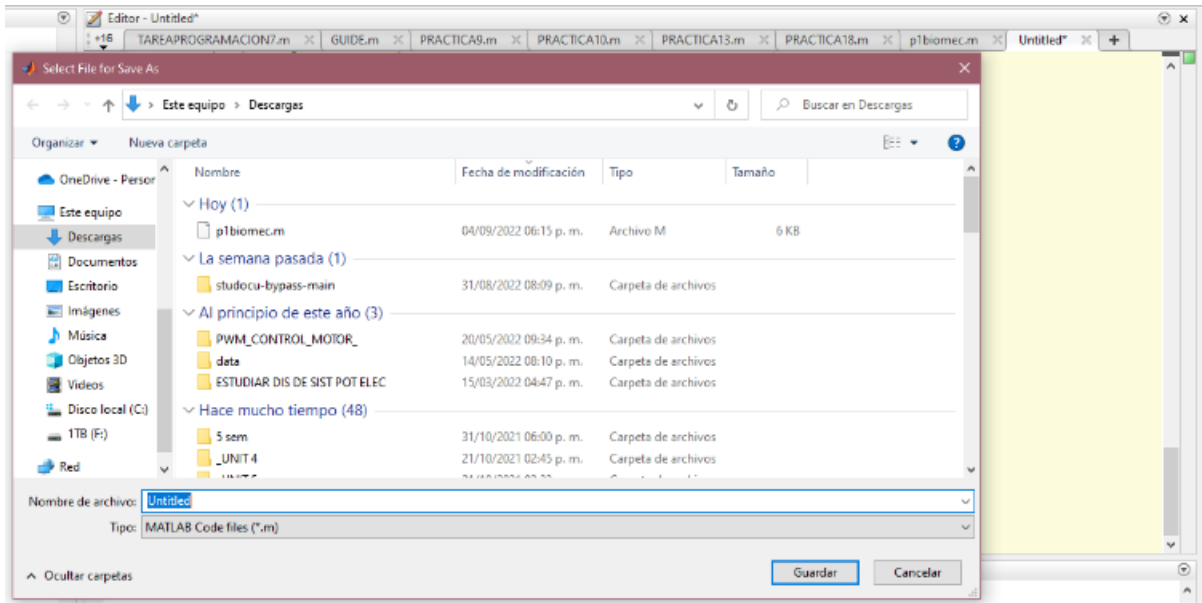
```

Editor - C:\Users\Vair Obed\Downloads\p1biomec.m
TAREA PROGRAMACION 7.m GUIDE.m PRACTICA 9.m PRACTICA 10.m PRACTICA 13.m PRACTICA 18.m p1biomec.m Untitled

1  %%% AN 88 LINE TOPOLOGY OPTIMIZATION CODE Nov, 2010 %%%
2  %function top88(nelx,nely,volfrac,penal,rmin,ft)
3  nelx=120; nely=20; volfrac=0.5; penal=3.0; rmin=1.5; ft=1;
4  %% MATERIAL PROPERTIES
5  E0 = 1;
6  Emin = 1e-9;
7  nu = 0.3;
8  %% PREPARE FINITE ELEMENT ANALYSIS
9  A11 = [12 3 -6 -3; 3 12 3 0; -6 3 12 -3; -3 0 -3 12];
10 A12 = [-6 -3 0 3; -3 -6 -3 -6; 0 -3 -6 3; 3 -6 3 -6];
11 B11 = [-4 3 -2 9; 3 -4 -9 4; -2 -9 -4 -3; 9 4 -3 -4];
12 B12 = [ 2 -3 4 -9; -3 2 9 -2; 4 9 2 3; -9 -2 3 2];
13 KE = 1/(1-nu^2)/24*( [A11 A12; A12' A11]+nu*[B11 B12; B12' B11]);
14 noderms = reshape(1:(1+nelx)*(1+nely),1+nely,1+nelx);
15 edofVec = reshape(2*noderms(1:end-1,1:end-1)+1,nelx*nely,1);
16 edofMat = repmat(edofVec,1,8)+repmat([0 1 2*nely+[2 3 0 1] -2 -1],nelx*nely,1);
17 iK = reshape(kron(edofMat,ones(8,1))',64*nelx*nely,1);
18 jK = reshape(kron(edofMat,ones(1,8))',64*nelx*nely,1);
19 % DEFINE LOADS AND SUPPORTS (HALF MBB-BEAM)
20 F = sparse(nelx*(nely+1)+2,1,-1,2*(nely+1)*(nelx+1),1);
21 U = zeros(2*(nely+1)*(nelx+1),1);
22 fixeddofs = union(2*nely+2,2*(nelx+1)*(nely+1));
23 alldofs = 1:2*(nely+1)*(nelx+1);
24 freedofs = setdiff(alldofs,fixeddofs);
25 %% PREPARE FILTER
26 iH = ones(nelx*nely*(2*(ceil(rmin)-1)+1)^2,1);
27 jH = ones(size(iH));

```

4. Guardar nuestro script con la extensión (.m) correspondiente a matlab y compilar



2.4. Implementación o desarrollo de la programación en sus diferentes vistas

```

1 %%% AN 88 LINE TOPOLOGY OPTIMIZATION CODE Nov, 2010 %%%
2 function top88(nelx,nely,volfrac,penal,rmin,ft);
3 nelx=120; nely=20; volfrac=0.5; penal=3.0; rmin=1.5; ft=1;
4 %%% MATERIAL PROPERTIES
5 E0 = 1;
6 Emin = 1e-9;
7 nu = 0.3;
8 %%% PREPARE FINITE ELEMENT ANALYSIS
9 A11 = [12 3 -6 -3; 3 12 3 0; -6 3 12 -3; -3 0 -3 12];
10 A12 = [-6 -3 0 3; -3 -6 -3 -6; 0 -3 -6 3; 3 -6 3 -6];
11 B11 = [-4 3 -2 9; 3 -4 -9 4; -2 -9 -4 -3; 9 4 -3 -8];
12 B12 = [ 2 -3 4 -9; -3 2 9 -2; 4 9 2 3; -9 -2 3 2];
13 KE = 1/(1-nu^2)/24*( [A11 A12; A12' A11]+nu*[B11 B12; B12' B11]);
14 nodenrs = reshape(1:(nelx)*(nely),nelx,nely,1);
15 edofnrc = reshape(2*nodenrs(1:end-1,1:end-1),nelx*nely,1);
16 edofmat = repmat(edofnrc,1,8)*repmat([0 1 2*nely/[2 3 0 1] -2 -1],nelx*nely,1);
17 Ik = reshape(8ron(edofmat,ones(8,1)),64*nely*nely,1);
18 Jk = reshape(8ron(edofmat,ones(8,1)),64*nely*nely,1);
19 % DEFINE LOADS AND SUPPORTS (HALF RB-BEAM)
20 F = sparse(nelx*(nely+1)+1,1,1,2*(nely+1)*(nelx+1),1);
21 U = zeros(2*(nely+1)*(nelx+1),1);
22 fixedofs = union(2*nely+2,2*(nelx+1)*(nely+1));
23 alldots = 1:2*(nely+1)*(nelx+1);
24 freedofs = setdiff(alldots,fixedofs);
25 %%% PREPARE FILTER
26 Ih = ones(nelx*nely*(2*(ceil(rmin)-1)+1)^2,1);
27 Jh = ones(size(Ih));
28 sh = zeros(size(Ih));
29 k = 0;
30 for i1 = 1:nelx
31     for j1 = 1:nely
32         e1 = (i1-1)*nely+j1;
33         for i2 = max(i1-(ceil(rmin)-1),1):min(i1+(ceil(rmin)-1),nelx)
34             for j2 = max(j1-(ceil(rmin)-1),1):min(j1+(ceil(rmin)-1),nely)
35                 e2 = (i2-1)*nely+j2;
36                 k = k+1;
37                 Ih(k) = e1;
38                 Jh(k) = e2;
39                 sh(k) = max(0,rmin-sqrt((i1-i2)^2+(j1-j2)^2));
40             end
41         end
42     end
43 end
44 H = sparse(Ih,sh,H);
45 Hs = sum(H,2);
46 %%% INITIALIZE ITERATION
47 x = repmat(volfrac,nely,nelx);
48 xPhys = x;
49 loop = 0;
50 change = 1;
51 obj = NaN(200,1);
52 changeplot = NaN(200,1);
53 volume = NaN(200,1);
54 %%% START ITERATION
55 while change > 0.01
56     loop = loop + 1;
57     %%% FE-ANALYSIS
58     sk = reshape(x(:))*(Emin+shPhys(:)).*penal*(E0-Emin),64*nely*nely,1);
59     K = sparse(1k,1k,K); K = (K+K')/2;
60     U(freedofs) = K(freedofs,freedofs)\V(freedofs);
61     %%% OBJECTIVE FUNCTION AND SENSITIVITY ANALYSIS
62     ce = reshape(sum(U(edofmat)*KE),n(edofmat),2,nely,nelx);
63     c = sum(sum(E0ndofPhys.*penal*(E0-Emin)).*ce);
64     dc = -penal*(E0-Emin)*shPhys.*(penal-1).*ce;
65     dv = ones(nely,nelx);
66     obj(loop+1) = c;
67     %%% FILTERING/MODIFICATION OF SENSITIVITIES
68     if ft == 1
69         dc(:) = H*(dc(:))./dc(:)./Ms./max(1e-3,u(:));
70     elseif ft == 2
71         dc(:) = H*(dc(:)./Ms);
72     else
73         dc(:) = H*(dc(:)./Ms);
74     end
75     %%% OPTIMALITY CRITERIA UPDATE OF DESIGN VARIABLES AND PHYSICAL DENSITIES
76     l1 = 0; l2 = 1e9; move = 0.2;
77     while (l2-l1)/(l1+l2) > 1e-3
78         lmid = 0.5*(l2+l1);
79         xnew = min(0,max(x-move,min(1,xin[xmove,x.*sqrt(-dc./dv/lmid)])));
80         if ft == 1
81             xPhys = xnew;
82         elseif ft == 2
83             xPhys(:) = (H*xnew(:))./Ms;
84         end
85         if sum(xPhys(:)) > volfrac*nely*nely, l1 = lmid; else, l2 = lmid; end
86     end
87     change = max(abs(xnew(:)-x(:)));
88     changeplot(loop+1) = change;
89     volume(loop+1) = mean(xPhys(:));
90     x = xnew;
91     %%% PRINT RESULTS
92     fprintf(' It.:%5i Obj.:%11.4f Vol.:%7.3f ch.:%7.3f\n',loop,c, ...
93           mean(xPhys(:),change);
94     if exist('OCTAVE_VERSION','builtin') == 0
95         fflush(stdout);
96     end
97     %%% PLOT DENSITIES
98     colormap(gray); imagesc(1-xPhys); caxis([0 1]); axis equal; axis off; draw
99 end
100 %%% EXTRA PLOTS
101 Subj(isnan(obj)) = [];
102 Xchangeplot(isnan(changeplot)) = [];
103 Xvolume(isnan(volume)) = [];
104 %
105 Xfigure;
106 Xax1 = subplot(3,1,1);
107 Xplot(obj,'r');
108 Xtitle(ax1,'Objective function')
109 %
110 Xax2 = subplot(3,1,2);
111 Xplot(volume,'r');
112 Xtitle(ax2,'Volume')
113 %
114 Xax3 = subplot(3,1,3);
115 Xplot(changeplot,'r');
116 Xtitle(ax3,'Change of volume')
117 %
118 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
119 % This Matlab code was written by E. Andreassen, A. Clausen, M. Schevenels,
120 % B. S. Lazarov and O. Sigmund, Department of Solid Mechanics,
121 % Technical University of Denmark,
122 % DK-2800 Lyngby, Denmark.
123 % Please sent your comments to: sigmund@fam.dtu.dk
124 %
125 % The code is intended for educational purposes and theoretical details

```

Figura 3: Código de Matlab

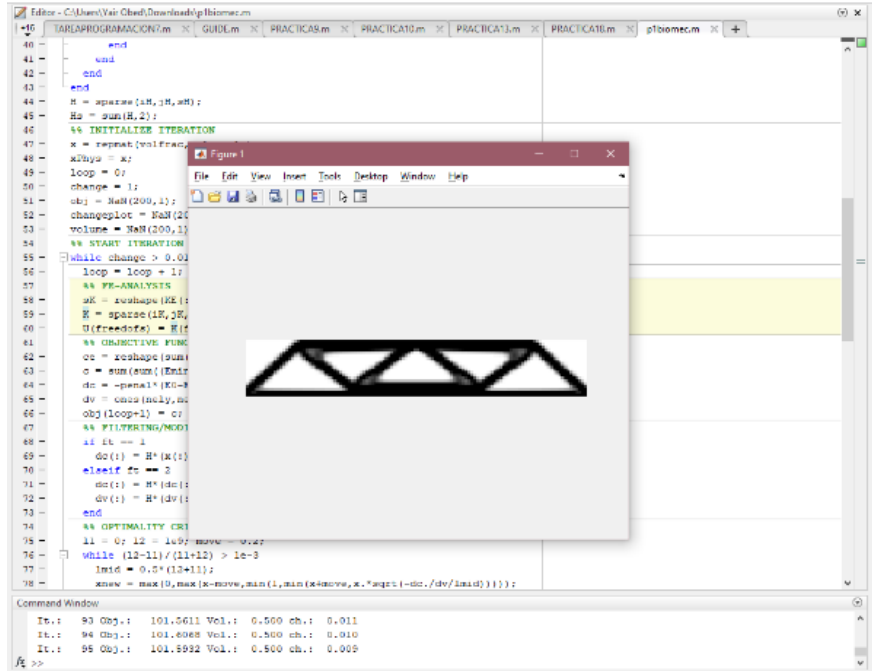


Figura 4: Imagen del diseño optimizado

3. Conclusiones de cada autor

■ Emanuel Salas de Leon 1854873

En esta actividad hablamos de la optimización topológica, como bien sabemos es un método de optimización que se utiliza para encontrar el mejor diseño posible para una estructura dada. Este método se basa en el estudio de las relaciones entre los elementos de un sistema, y el objetivo es encontrar la topología óptima que minimizará el costo total del sistema. En esta primera práctica realizamos la programación de una optimización topológica de una viga, en el software de matlab, como bien se puede observar en la imagen final se muestra el diseño resultante, que si queremos minimizar costos y tener una topología óptima, sería el indicado.

■ Jason Méndez Muñoz 1889560

Con esta actividad cada uno de los integrantes de nuestro equipo se familiarizó con la definición de Optimización topológica y sus áreas de aplicación, así como la forma de implementar esta actividad usando el software Matlab, nos familiarizamos con los comando necesarios y la estructura del código así como su ejecución, donde tomamos como ejemplo una viga sencilla sometida a una fuerza cortante, la cual pudimos optimizar dando una pieza resultante capaz de soportar la misma carga pero como una considerable disminución de material.

■ Jesús Mario Duarte Salinas 1907630

Por medio de esta práctica pudimos entender el concepto de optimización topológica y utilizamos un código de programación para implementarlo en Matlab. En mi caso yo no tenía mucho conocimiento acerca de este concepto de optimización, pero tras una investigación acerca de este ahora comprendo que nos sirve de gran ayuda para sintetizar las estructuras con valores óptimos de uno varios de sus parámetros. Y por medio del código implementado se mostró un ejemplo de una optimización topológica para una viga.

■ Lorena Alejandra Campos Carrasco 1909279

Con la realización de esta práctica se buscaba entender mejor el concepto de optimización topológica e implementar un ejemplo de un código en Matlab. Para ello se realizó una investigación que nos ayudaría a entender que es la optimización topológica, la cual nos ayuda a sintetizar estructuras de manera óptima según varios factores, una vez que ya se entendió el concepto se buscó un ejemplo de caso donde se pudiera implementar.

Nosotros escogimos una optimización topológica para una viga sencilla sometida a una fuerza. Esta pieza se optimizó mediante un código en Matlab dando como resultado la pieza capaz de soportar la carga pero optimizando el material.

■ **Yair Obed Morales Ortiz 1992266**

Al realizar esta práctica puede concluir que no conocía el concepto de optimización topológica y aprendimos cómo se lleva a cabo mediante una programación y simulación por el software numérico Matlab. Nos familiarizamos un poco mejor con el concepto y como se lleva a cabo en la programación, tanto las partes de la programación como las funciones que se utilizan, de ejemplo pusimos la optimización topológica de una viga sujeta a una carga central, donde se aprecia la animación de la estructura normal y como se optimiza.

Referencias

- [1] Abad Castro, R. (2018). Implementación en matlab de un procedimiento de optimización topológica para estructuras con cargas térmicas.
- [2] Meza, C. A., Tamayo, F., and Franco, E. E. (2015). Optimización topológica aplicada al diseño de componentes estructurales mecánicos de peso reducido. *El hombre y la máquina*, (46):72–79.
- [3] Sigmund, O. (2001). A 99 line topology optimization code written in matlab. *Structural and multidisciplinary optimization*, 21(2):120–127.