

Unicersidad Autónoma de Nuevo León

Facultad de Ingeniería Mecánica y Eléctrica

PRÁCTICA #3

Docente: Yadira Moreno Vera

Brigada: 109 Hora: N5 Día: lunes

Materia: Laboratorio de Biomecánica

Integrantes del equipo:

Nombre	Matricula:	Carrera:
Aarón Lozano Aguilar	1844469	IMTC
Montserrat Granados Salinas	1817165	IMTC
Eunice Carolina Méndez Sosa	1851345	IMTC
Eimie Carolina Pereda Sánchez	1915035	IMTC
Aida Mata Moreno	1734743	IMTC

Semestre: Agosto-Diciembre 2022

Práctica #3: Diseño de la estructura de un panorámico.

Objetivo:

El estudiante deberá presentar un reporte con la solución numérica computacional del problema de la simulación del desempeño mecánico de componentes mecánicos por medio del método de MATLAB, esto para desarrollar en el estudiante la capacidad de análisis, implementación y solución de un problema propuesto.

Marco teórico:

Una estructura panorámica es el soporte sobre el cual se posicionará un anuncio publicitario, ya sea de una cara o de tres caras.

Estas estructuras usualmente se encuentran en medio de diversos paisajes urbanos y sostienen diseños publicitarios con el objetivo de promocionar un producto, servicio o transmitir un mensaje.

Cada país tiene ciertas normativas en cuanto a dónde es apropiado o no colocar estos soportes para anuncios publicitarios. En algunos no está permitido que se construyan estructuras panorámicas a los lados de autopistas porque estos pueden distraer a los conductores.

Los panorámicos se exponen a altas ráfagas de viento, por lo que su estructura ocupa ser muy rígida para soportar estas fuerzas.

En la figura 1 se muestra el panorámico que será el espacio de diseño a evaluar, éste será de 2 dimensiones, con cargas y apoyos como se muestra a continuación:

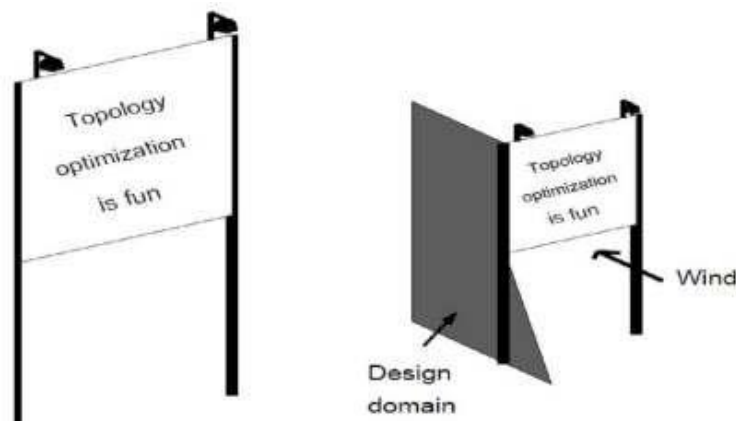


Figura 1. Imagen del panorámico.

Existen diversos materiales que las agencias especializadas usan en la creación y diseño para estas estructuras. Usualmente los postes panorámicos están contruidos de metal y acero para que sean lo suficientemente resistentes al clima, lluvias y cualquier otro fenómeno de la naturaleza.

Por otra parte, el panorámico en sí mismo son hechos de lona, vallas de PVC, plástico, tela, metal o acrílico. También existen espectaculares digitales o electrónicos que tienen luces, pantallas eléctricas y música.

En la figura 2 se puede ver el espacio de diseño para esta práctica. Se espera una fracción volumétrica aproximada de 0.20% del espacio de diseño. Supongamos que el panorámico es muy rígido 1, y sus patas son del mismo material que el marco.

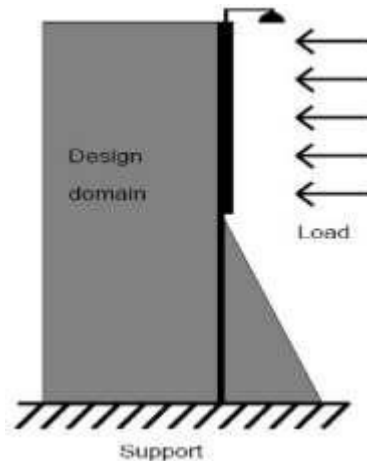
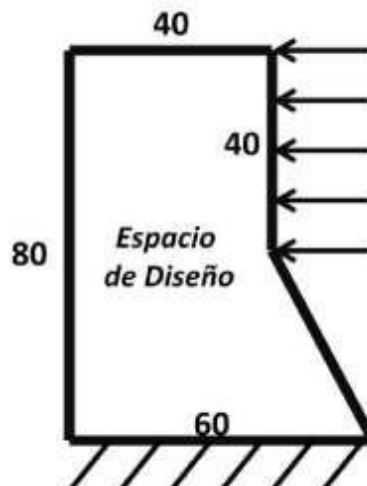


Figura 2. Espacio de diseño

Desarrollo

Se tomarán ciertas consideraciones para la solución de esta práctica: 5 cargas, los apoyos tendrán restricciones en "X", "Y" y el espacio de diseño para esta práctica será de:



Estado del arte

Título del documento	“A 99 line topology optimization code written in Matlab”
Fuente Bibliográfica	99 Line Topology Optimization Code – O. Sigmund, Department of Solid Mechanics, Building 404, Technical University of Denmark, DK-2800 Lyngby, Denmark.
Objetivo	Dar a conocer cada una de las secciones que integran el código de optimización topológica de 99 líneas en Matlab, saber ejecutar el análisis del código y observar los resultados obtenidos.
Contenido	<p>El artículo presenta una implementación compacta en Matlab de un código de optimización topológica para la minimización de la conformidad de estructuras cargadas estáticamente. El número total de líneas de entrada de Matlab es de 99, incluyendo el optimizador y la subrutina de elementos finitos. Las 99 líneas están</p> <p>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.</p> <p>De hecho, excluyendo las líneas de comentario y las asociadas a la salida y al análisis de elementos finitos, se observa que sólo se necesitan 49 líneas de entrada, de Matlab para resolver un problema de optimización topológica bien planteado. Añadiendo tres líneas adicionales, el programa puede resolver problemas con múltiples casos de carga. El código está pensado para fines educativos.</p>
Palabras Clave	Optimización Topológica, criterios de optimización, web mundial, código Matlab
Conclusión	En este artículo investigado sobre la optimización topológica y su programación nos habla de cómo está constituido dicho código de 99 líneas y cuál es la idea principal que este debe cumplir y que es lo que debe demostrar al ejecutarlo.

Código

```
3 - function toppract(nelx,nely,volfrac,penal,rmin)
4 - % INITIALIZE
5 - x(1:nely,1:nelx) = volfrac;
6 - loop = 0;
7 - %DECLARACION DE VACIO
8 - for ely = 1:nely
9 -     for elx = 1:nelx
10 -         if (((ely-(nely*0.5)<(2*elx)-(1.36*nelx)) && (ely<(1+nely*0.5))) && (elx > (1+nelx)*0.6666))
11 -             passive(ely,elx)=1;
12 -         else
13 -             passive(ely,elx) = 0;
14 -         end
15 -     end
16 - end
17 - x(find(passive))=0.001;
18 - change = 1.;
19 - % START ITERATION
20 - while change > 0.01
21 -     loop = loop + 1;
22 -     xold = x;
23 -     % FE-ANALYSIS
24 -     [U]=FE(nelx,nely,x,penal);
25 -     % OBJECTIVE FUNCTION AND SENSITIVITY ANALYSIS
26 -     [KE] = 1k;

27 -     c = 0.;
28 -     for ely = 1:nely
29 -         for elx = 1:nelx
30 -             n1 = (nely+1)*(elx-1)+ely;
31 -             n2 = (nely+1)* elx +ely;
32 -             dc(ely,elx) = 0.;
33 -             for i = 1:5
34 -                 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);
35 -                 c = c + x(ely,elx)^penal*Ue'*KE*Ue;
36 -                 dc(ely,elx) = -penal*x(ely,elx)^(penal-1)*Ue'*KE*Ue;
37 -             end
38 -         end
39 -     end
40 -     % FILTERING OF SENSITIVITIES
41 -     [dc] = check(nelx,nely,rmin,x,dc);
42 -     % DESIGN UPDATE BY THE OPTIMALITY CRITERIA METHOD
43 -     [x] = OC(nelx,nely,x,volfrac,dc,passive);
44 -     % PRINT RESULTS
45 -     change = max(max(abs(x-xold)));
46 -     disp(['It.: ' sprintf('%4i',loop) 'Obj.: ' sprintf('%10.4f',c) ...
47 -         'Vol.: ' sprintf('%6.3f',sum(sum(x))/(nelx*nely)) ...
48 -         'ch.: ' sprintf('%6.3f',change )])
49 -     % PLOT DENSITIES
50 -     colormap(gray); imagesc(-x); axis equal; axis tight; axis off; pause(1e-6);
51 - end
```

```

52
53 %%%%%%%%%%% OPTIMALITY CRITERIA UPDATE %%%%%%%%%%%
54 function [xnew]=OC(nelx,nely,x,volfrac,dc,passive)
55 -     l1 = 0;
56 -     l2 = 100000;
57 -     move = 0.2;
58 -     while (l2-l1 > 1e-4)
59 -         lmid = 0.5*(l2+l1);
60 -         xnew = max(0.001,max(x-move,min(1.,min(x+move,x.*sqrt(-dc./lmid)))));
61 -         xnew(find(passive)) = 0.001;
62 -         if sum(sum(xnew)) - volfrac*nelx*nely > 0;
63 -             l1 = lmid;
64 -         else
65 -             l2 = lmid;
66 -         end
67 -     end
68
69 %%%%%%%%%%% MESH-INDEPENDENCY FILTER %%%%%%%%%%%
70 function [dcn]=check(nelx,nely,rmin,x,dc)
71 -     dcn=zeros(nely,nelx);
72 -     for i = 1:nelx
73 -         for j = 1:nely
74 -             sum=0.0;
75 -             for k = max(i-round(rmin),1):min(i+round(rmin),nelx)
76 -                 for l = max(j-round(rmin),1):min(j+round(rmin), nely)
77 -                     fac = rmin-sqrt((i-k)^2+(j-l)^2);
78 -
79 -                     sum = sum+max(0,fac);
80 -                     dcn(j,i) = dcn(j,i) + max(0,fac)*x(l,k)*dc(l,k);
81 -                 end
82 -             end
83 -             dcn(j,i) = dcn(j,i)/(x(j,i)*sum);
84 -         end
85 -     end
86
87 %%%%%%%%%%% FE-ANALYSIS %%%%%%%%%%%
88 function [U]=FE(nelx,nely,x,penal)
89 -     [KE] = 1k;
90 -     K = sparse(2*(nelx+1)*(nely+1),2*(nelx+1)*(nely+1));
91 -     F = sparse(2*(nely+1)*(nelx+1),5);
92 -     U = sparse(2*(nely+1)*(nelx+1),5);
93 -     for ely = 1:nely
94 -         for elx = 1:nelx
95 -             n1 = (nely+1)*(elx-1)+ely;
96 -             n2 = (nely+1)*elx+ely;
97 -             edof = [2*n1-1; 2*n1; 2*n2-1; 2*n2; 2*n2+1; 2*n2+2; 2*n1+1; 2*n1+2];
98 -             K(edof,edof) = K(edof,edof)+x(ely,elx)^penal*KE;
99 -         end
100 -     end
101 -     % DEFINE LOADSAND SUPPORTS (HALF MBB-BEAM)
102 -     F(2*(nelx)*(nely+1)+2,1) = 1;
103 -     F(2*(nelx)*(nely+1)+(nely/4),2) = 1;
104 -     F(2*(nelx)*(nely+1)+(nely/2),3) = 1;

```

```

104 - F(2*(nelx)*(nely+1)+(nely),4) = 1;
105 - F(2*(nelx)*(nely+1)+(nely*1.2),5) = 1;
106 - fixeddofs = 2*(nely+1):2*(nely+1):2*(nelx+1)*(nely+1);
107 - alldofs = [1:2*(nely+1)*(nelx+1)];
108 - freedofs = setdiff(alldofs,fixeddofs);
109 - % SOLVING
110 - U(freedofs,:) = K(freedofs,freedofs) \ F(freedofs,:);
111 - U(fixeddofs,:)= 0;
112
113 - %%%%%%%%%%% ELEMENT STIFFNESS MATRIX %%%%%%%%%%%
114 - function [KE]=lk
115 - E = 1.;
116 - nu = 0.3;
117 - k=[ 1/2-nu/6 1/8+nu/8 -1/4-nu/12 -1/8+3*nu/8 ...
118 - -1/4+nu/12 -1/8-nu/8 nu/6 1/8-3*nu/8];
119 - KE = E/(1-nu^2)*[ k(1) k(2) k(3) k(4) k(5) k(6) k(7) k(8)
120 - k(2) k(1) k(8) k(7) k(6) k(5) k(4) k(3)
121 - k(3) k(8) k(1) k(6) k(7) k(4) k(5) k(2)
122 - k(4) k(7) k(6) k(1) k(8) k(3) k(2) k(5)
123 - k(5) k(6) k(7) k(8) k(1) k(2) k(3) k(4)
124 - k(6) k(5) k(4) k(3) k(2) k(1) k(8) k(7)
125 - k(7) k(4) k(5) k(2) k(3) k(8) k(1) k(6)
126 - k(8) k(3) k(2) k(5) k(4) k(7) k(6) k(1)];

```

Después de tener los cambios necesarios en el código se optó por darle un valor a toppract de toppract (40,80,0.2,3.0,0.5) en la ventana de comando de Matlab. Llegando a nuestra respuesta final la cual es esta:

```

Command Window
>>
>> toppract(40,80,0.2,3.0,0.5)
It.: 10bj.:42819847281.0592Vol.: 0.200ch.: 0.200
Warning: MATLAB has disabled some advanced graphics rendering features by switching to software OpenGL. For more
information, click here.
It.: 20bj.:42819839234.7122Vol.: 0.200ch.: 0.200
It.: 30bj.:42819838251.6167Vol.: 0.200ch.: 0.200
It.: 40bj.:42819837687.0867Vol.: 0.200ch.: 0.200
It.: 50bj.:42819837275.7362Vol.: 0.200ch.: 0.200
It.: 60bj.:42819837054.2473Vol.: 0.200ch.: 0.200
It.: 70bj.:42819836942.4612Vol.: 0.200ch.: 0.200

```

Fig.16 Ventana de comando con el inicio de implementación del diseño con los parámetros dados [toppract(40,80,0.2,3.0,0.5)]

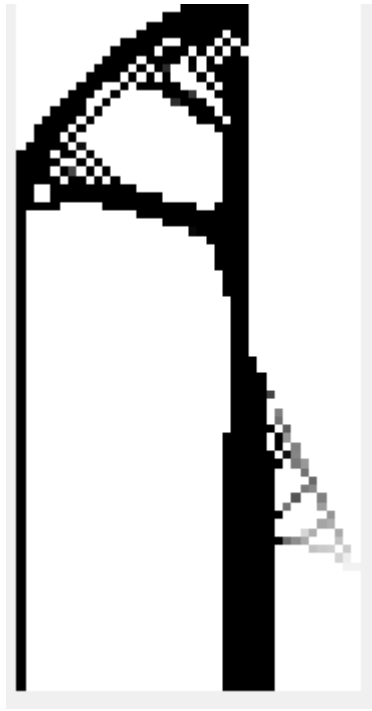


Fig.17 Imagen del diseño resultante con los parámetros dados en la ventana de comando en Matlab.

Conclusiones

Aarón Lozano Aguilar 1844469 IMTC

Concluyo qué esta práctica fue muy interesante y de mucho aprendizaje al ponernos a analizar de una manera crítica la representación gráfica del panorámico, gracias a esta práctica se aprendió a analizar el comportamiento de un panorámico y así poder simularlo en Matlab y representarlo gráficamente

Montserrat Granados Salinas 1817165 IMTC

A lo largo de esta práctica pude apreciar la versatilidad que presenta este código de optimización para el análisis de elemento finito; esto debido a que al realizar unas cuantas modificaciones al código este nos permite obtener la optimización de una valla publicitaria. Me ayudó a poder entender las maneras en las que podemos aplicar y en donde este tipo de programa y como nos ayuda de una manera muy versátil cumpliendo las necesidades que tengamos o nos pidan.

Eunice Carolina Méndez Sosa 1851345 IMTC

Esta práctica la cual se trató sobre el diseño de una estructura de un panorámico pudimos desarrollar más nuestra capacidad de análisis y resolución de un problema, para esta actividad nos apoyamos con el software de Matlab en el cual pudimos analizar el comportamiento del panorámico y realizar la representación gráfica de este.

Eimie Carolina Pereda Sánchez 1915035 IMTC

En esta práctica se trato acerca de la optimización y diseño de un panorámico, para la investigación encontramos el estado de arte y estructura de dicho objeto y así armamos el código basándonos en el de prácticas anteriores solo cambiando ciertos valores necesarios para que se basara en el tema de esta práctica, me gustó mucho la forma en que en cada practica vamos diseñando y creando nuevos códigos en MATLAB, nos hace practicar mucho y empezar a desarrollar ese pensamiento analítico y además a relacionarnos más con el programa ya que a veces puede llegar a ser algo tedioso.

Aida Mata Moreno 1734743 IMTC

En esta práctica aprendí sobre las estructuras de un panorámico y el cómo poder codificarlo en matlab, esta actividad es muy importante, ya que pudimos ampliar más nuestros conocimientos en la plataforma de programación de matlab.

Referencias Bibliográficas

- 99 Line Topology Optimization Code – O. Sigmund, Department of Solid Mechanics, Building 404, Technical University of Denmark, DK-2800 Lyngby, Denmark.

Blu Cactus. (2021). Estructura de un panorámico. octubre 2021, de Blu Cactus Digital Marketing Sitio web: <https://www.blucactus.com.mx/cuanto-cuesta-estructura-panoramico/>

