



Universidad Autónoma de Nuevo León

Facultad de Ingeniería Mecánica y Eléctrica

Laboratorio de biomecánica

**“PRÁCTICA 3. DISEÑO DE LA ESTRUCTURA
DE UN PANORÁMICO”**

Instructor(a): Ing. Isaac Estrada

Brigada: 109

Nombre	Matrícula	Carrera
Victor Emmanuel Cantú Corpus	1909659	IMC
Mauricio Julián Salazar Salzar	1906944	IMC
Brayan Orlando Belloc Castillo	1898242	IMC

Semestre Agosto – Diciembre 2022

Día 17 del mes Octubre del año 2022

Ciudad Universitaria, San Nicolás de los Garza, Nuevo León

Objetivo

El estudiante deberá presentar una propuesta de análisis de formas y de la programación para la ejecución de la optimización (descripción funcional) de características de trabajo específicas que presenta la(s) ventaja(s) (mencionar ventajas).

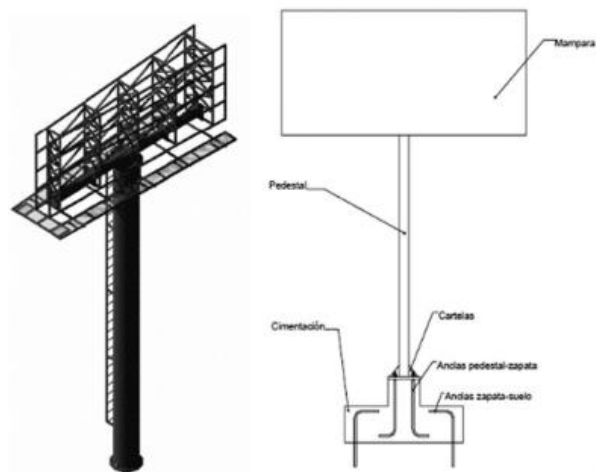
Marco teórico

Se refiere a aquellos panorámicos que se colocan en un edificio y que aprovechan la visibilidad y la altura de este para causar un mayor impacto.

Un espectacular es un anuncio publicitario de gran formato, normalmente se utiliza para la publicidad gráfica de cualquier ámbito, son de diversos tamaños y se colocan en puntos estratégicos de una ciudad para su mayor visibilidad.

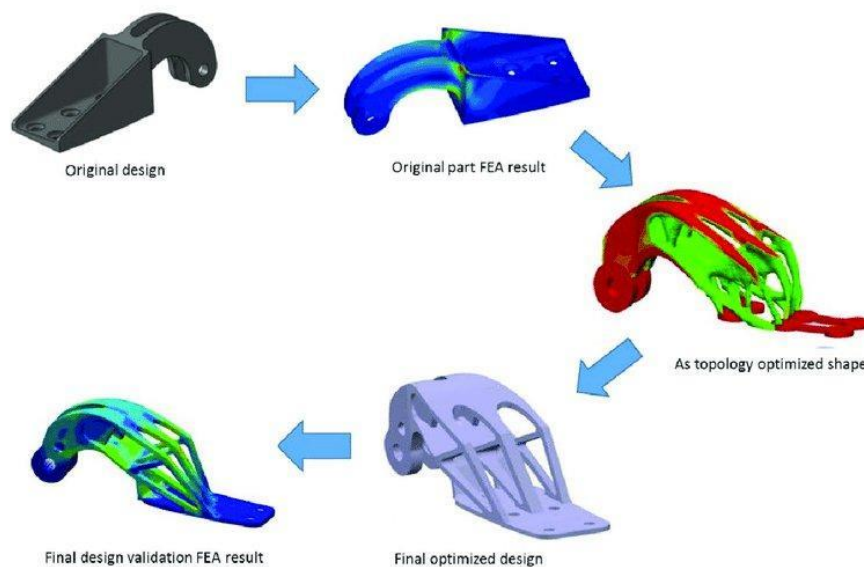
¿Cuál es la estructura de un espectacular?

- 1) Son sumamente visibles para un gran número de personas, y tal vez esa es su mayor virtud.
- 2) Son relativamente económicos en comparación con otros medios de comunicación.
- 3) Ofrece a las marcas la oportunidad de llegar tanto a su objetivo como a consumidores potenciales.



La optimización topológica suele tener lugar hacia el final del proceso de diseño, cuando la pieza deseada necesita tener un peso menor o usar menos materiales. El diseñador trabaja para descubrir ciertos parámetros preestablecidos, como las cargas aplicadas, el tipo de material, las limitaciones del modelo y su organización en el espacio.

En primer lugar, la optimización topológica estructural determina el espacio de diseño mínimo permisible que es necesario para optimizar la forma del producto. Después, de forma virtual, el software de optimización topológica aplica presión sobre el diseño desde distintos ángulos, pone a prueba su integridad estructural e identifica el material innecesario.



La técnica más común y práctica para realizar la optimización topológica es el método de elementos finitos (MEF). En primer lugar, el MEF tiene en cuenta el diseño geométrico para el espacio mínimo permitido (junto con otros factores) y divide el diseño en varias partes. A continuación, pone a prueba cada elemento finito para determinar su rigidez, su conformidad y el material redundante que contiene. Por último, el MEF vuelve a unir las piezas para finalizar el diseño completo.

Validar el diseño implica determinar un umbral para el campo de densidad de los elementos entre un valor de 0 y 1. Un valor de 0 convierte en vacío el material de una región designada de la estructura, mientras que un valor de 1 identifica la región designada como material sólido. Después, el diseñador puede eliminar todo el material

innecesario del modelo y finalizar la parte de optimización topológica del diseño.

Antes de la fabricación aditiva, los diseñadores descartaban muchas de las formas complejas que creaba la optimización topológica, ya que fabricarlas no era factible, y su potencial quedaba sin realizarse.

Procedimiento de la programación

```
1  %%%% A 99 LINE TOPOLOGY OPTIMIZATION CODE BY OLESIGMUND, JANUARY 2000%%%%
2  %%%% CODE MODIFIED FOR INCREASED SPEED, September 2002, BY OLE SIGMUND %%%
3  function topp3(nelx,nely,volfrac,penal,rmin);
4  % INITIALIZE
5  x(1:nely,1:nelx) = volfrac;
6  loop = 0;
7  %Declarando 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;
```

```
Editor - D:\Matlab\bin\topp3.m
topp3a.m x topp3.m +
22 - xold = x;
23 % FE-ANALYSIS
24 [U]=FE(nelx,nely,x,penal);
25 %13 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; %19
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;
35 2*n1+1;2*n1+2],1);
36 c = c + x(ely,elx)^penal*Ue'*KE*Ue;
37 dc(ely,elx) = dc(ely,elx)-penal*x(ely,elx)^(penal-1)*Ue'*KE*Ue;
38 end
39 end
40 end
41 %25 FILTERING OF SENSITIVITIES
42 [dc] = check(nelx,nely,rmin,x,dc);
43 %27 DESIGN UPDATE BY THE OPTIMALITY CRITERIA METHOD
```

```

Editor - D:\Matlab\bin\topp3.m
topp3a.m  topp3.m  +
43 %27 DESIGN UPDATE BY THE OPTIMALITY CRITERIA METHOD
44 [x] = OC(nelx,nely,x,volfrac,dc,passive);
45 %29 PRINT RESULTS
46 change = max(max(abs(x-xold)));
47 disp(['It.: ' sprintf('%4i',loop) 'Obj.: ' sprintf('%10.4f',c) ...
48 ' Vol.: ' sprintf('%6.3f',sum(sum(x))/(nelx*nely)) ...
49 ' ch.: ' sprintf('%6.3f',change )])
50 % PLOT DENSITIES
51 colormap(gray); imagesc(-x); axis equal; axis tight; axis off; pause(1e-6);
52 end
53 %40 %%%%%%%%% OPTIMALITY CRITERIA UPDATE %%%%%%%%%
54 function [xnew]=OC(nelx,nely,x,volfrac,dc,passive)
55 l1 = 0; l2 = 100000; move = 0.2;
56 while (l2-l1 > 1e-4)
57     lmid = 0.5*(l2+l1);
58     xnew = max(0.001,max(x-move,min(1.,min(x+move,x.*sqrt(-dc./lmid)))));
59     xnew(find(passive)) = 0.001;
60     if sum(sum(xnew)) - volfrac*nelx*nely > 0;
61         l1 = lmid;
62     else
63         l2 = lmid;
64     end

```

```

Editor - D:\Matlab\bin\topp3.m
topp3a.m  topp3.m  +
64 end
65 end
66 %%%%%%%%% MESH-INDEPENDENCY FILTER %%%%%%%%%
67 function [dcn]=check(nelx,nely,rmin,x,dc)
68 dcn=zeros(nely,nelx);
69 for i = 1:nelx
70     for j = 1:nely
71         sum=0.0;
72         for k = max(i-round(rmin),1):min(i+round(rmin),nelx)
73             for l = max(j-round(rmin),1):min(j+round(rmin), nely)
74                 fac = rmin-sqrt((i-k)^2+(j-l)^2);
75                 sum = sum+max(0,fac);
76                 dcn(j,i) = dcn(j,i) + max(0,fac)*x(l,k)*dc(l,k);
77             end
78         end
79         dcn(j,i) = dcn(j,i)/(x(j,i)*sum);
80     end
81 end
82 %65 %%%%%%%%% FE-ANALYSIS %%%%%%%%%
83 function [U]=FE(nelx,nely,x,penal)
84 [KE] = lk;
85 K = sparse(2*(nelx+1)*(nely+1), 2*(nelx+1)*(nely+1));

```

```

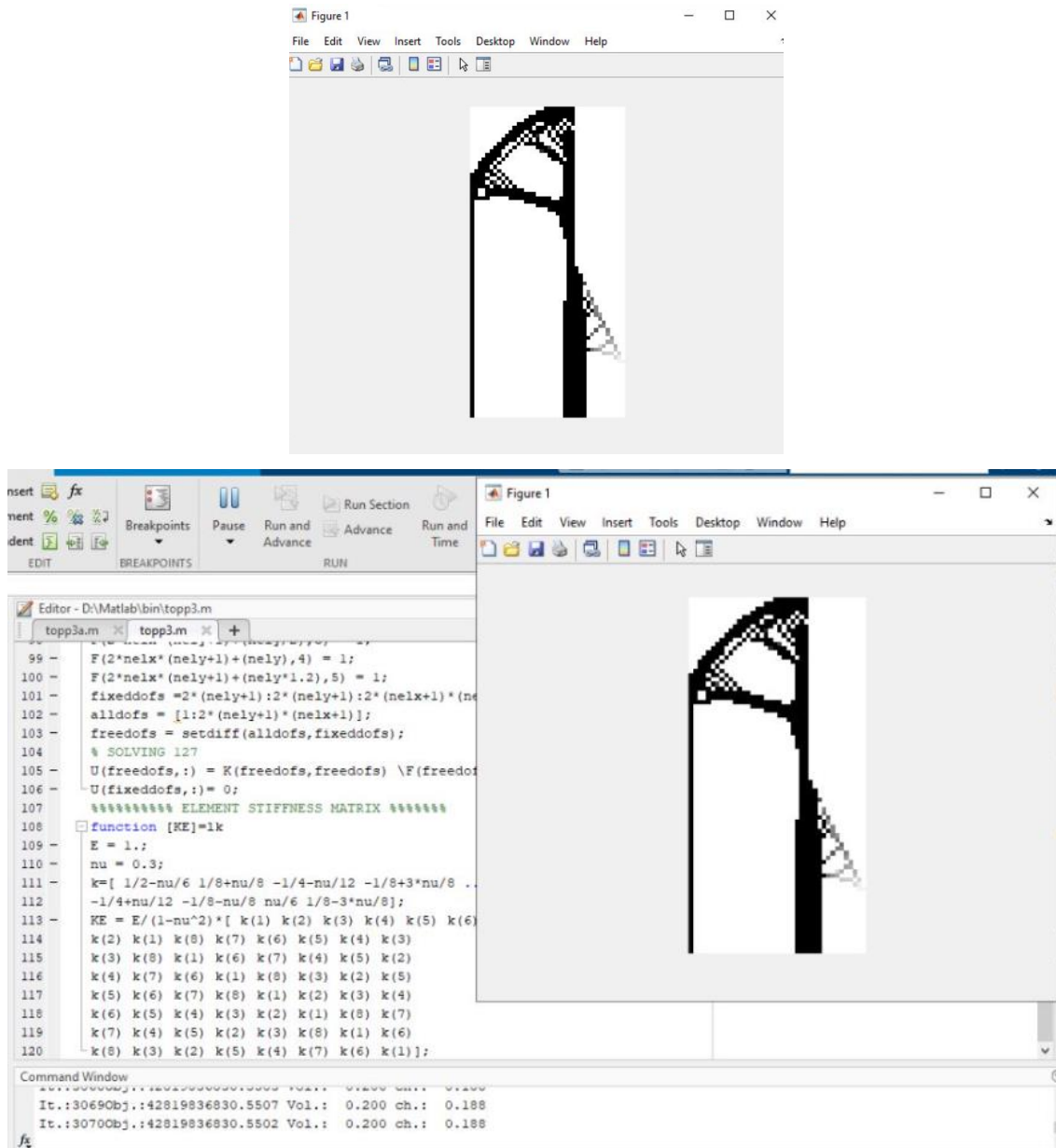
Editor - D:\Matlab\bin\topp3.m
topp3a.m  topp3.m  +
85 - K = sparse(2*(nelx+1)*(nely+1), 2*(nelx+1)*(nely+1));
86 - F = sparse(2*(nely+1)*(nelx+1),5); U = zeros(2*(nely+1)*(nelx+1),5);
87 - for ely = 1:nely
88 -     for elx = 1:nelx
89 -         n1 = (nely+1)*(elx-1)+ely;
90 -         n2 = (nely+1)* elx +ely;
91 -         edof = [2*n1-1; 2*n1; 2*n2-1; 2*n2; 2*n2+1;2*n2+2;2*n1+1; 2*n1+2];
92 -         K(edof,edof) = K(edof,edof) + x(ely,elx)^penal*KE;
93 -     end
94 - end
95 - % DEFINE LOADSAND SUPPORTS (HALF MBB-BEAM)
96 - F(2*nelx*(nely+1)+2,1) = 1;
97 - F(2*nelx*(nely+1)+(nely/4),2) = 1;
98 - F(2*nelx*(nely+1)+(nely/2),3) = 1;
99 - F(2*nelx*(nely+1)+(nely),4) = 1;
100 - F(2*nelx*(nely+1)+(nely*1.2),5) = 1;
101 - fixeddofs =2*(nely+1):2*(nely+1):2*(nelx+1)*(nely+1);
102 - alldofs = [1:2*(nely+1)*(nelx+1)];
103 - freedofs = setdiff(alldofs,fixeddofs);
104 - % SOLVING 127
105 - U(freedofs,:) = K(freedofs,freedofs) \F(freedofs,:);
106 - U(fixeddofs,:)= 0;

```

```

Editor - D:\Matlab\bin\topp3.m
topp3a.m  topp3.m  +
99 - F(2*nelx*(nely+1)+(nely),4) = 1;
100 - F(2*nelx*(nely+1)+(nely*1.2),5) = 1;
101 - fixeddofs =2*(nely+1):2*(nely+1):2*(nelx+1)*(nely+1);
102 - alldofs = [1:2*(nely+1)*(nelx+1)];
103 - freedofs = setdiff(alldofs,fixeddofs);
104 - % SOLVING 127
105 - U(freedofs,:) = K(freedofs,freedofs) \F(freedofs,:);
106 - U(fixeddofs,:)= 0;
107 - %%%%%%%%%% ELEMENT STIFFNESS MATRIX %%%%%%%%%%
108 - function [KE]=lk
109 -     E = 1.;
110 -     nu = 0.3;
111 -     k=[ 1/2-nu/6 1/8+nu/8 -1/4-nu/12 -1/8+3*nu/8 ...
112 -        -1/4+nu/12 -1/8-nu/8 nu/6 1/8-3*nu/8];
113 -     KE = E/(1-nu^2)*[ k(1) k(2) k(3) k(4) k(5) k(6) k(7) k(8)
114 -        k(2) k(1) k(8) k(7) k(6) k(5) k(4) k(3)
115 -        k(3) k(8) k(1) k(6) k(7) k(4) k(5) k(2)
116 -        k(4) k(7) k(6) k(1) k(8) k(3) k(2) k(5)
117 -        k(5) k(6) k(7) k(8) k(1) k(2) k(3) k(4)
118 -        k(6) k(5) k(4) k(3) k(2) k(1) k(8) k(7)
119 -        k(7) k(4) k(5) k(2) k(3) k(8) k(1) k(6)
120 -        k(8) k(3) k(2) k(5) k(4) k(7) k(6) k(1)];

```

Conclusión

Victor Cantú: Una vez realizada esta práctica pudimos conocer mejor sobre algunas propiedades y características de un anuncio denominado normalmente panorámico o espectacular, asimismo profundizamos aún más en el aspecto del estado del arte y la optimización topológica en cuanto el análisis de estructuras, dependiendo de la geometría que se asigne. Es importante conocer la estructura y geometría con la que se está trabajando para así tener un mejor resultado tanto de la programación como del diseño.

Mauricio Salazar: En conclusión, esta práctica es similar a las realizadas anteriormente, está siendo más compleja ya que se estudiaron diferentes casos, también en esta práctica se identificaron los elementos pasivos tomados para el diagrama y por último comentar la importancia que tienen los softwares para ayudarnos mediante sus diferentes herramientas a resolver diferentes tipos de situaciones como la de este caso la optimización de los diferentes casos planteado.

Brayan Belloc: La practica nos permitió observar el funcionamiento de la iteración en Matlab, durante la práctica se pudo observar cómo Matlab nos daba información acerca de la iteración en la que iba el código, el código usado duro más de una hora en realizar todas las iteraciones, haciendo que sea una práctica con mucha duración, los problemas surgidos fue adaptar el código pero en si solo era algunos cambios pues se usó el mismo código usado en la primer práctica, el código de 99 líneas, el mismo código usado en la segunda práctica, así se pudo observar la iteración dentro de Matlab en esta practica

Bibliografía

- ¿Qué es un anuncio panorámico? – Studio Apartment Hub. (2022, 7 abril). Recuperado 16 de octubre de 2022, de <https://studioapartmenthub.com/que-es-un-anuncio-panoramico/>
- Nociones básicas de optimización topológica: Cómo usar modelos algorítmicos para crear un diseño ligero. (s. f.). Formlabs. Recuperado 17 de octubre de 2022, de <https://formlabs.com/latam/blog/optimizacion-topologica/>