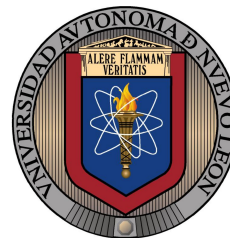




Facultad de Ingeniería Mecánica y Eléctrica

UANL

MECATRÓNICA



Laboratorio de Biomecánica

Grupo:309

Diseño de la estructura de un panorámico Práctica 3

FECHA DE ENTREGA: 26 OCTUBRE 2022

Profesor:

Dra. Yadira Moreno Vera

Nombre	Matricula	Carrera
Betsaida Alejandra Ruedas Vázquez	1730437	IMTC
Miguel Rodrigo Aguilar Moreno	1801380	IMTC
Erick Daniel Esquivel Arguelles	1826021	IMTC
Omar Isaí Moreno Cruz	1849630	IMTC
Saul Moises Mendoza Cida	1942534	IMTC

Índice

1. Objetivo:	3
2. Nombre y definición de la gemoetría:	3
3. Estado del arte:	3
4. Marco teórico:	4
5. Desarrollo:	5
6. Conclusiones:	11

1. 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).

2. Nombre y definición de la gemoetría:

¿Qué es el anuncio panorámico?

La publicidad en anuncio panorámico es un espacio publicitario exterior de gran tamaño destinado a captar la atención de los transeúntes. Por lo general, se coloca en áreas de alto tráfico, cerca de las rutas de tráfico y autopistas.

El anuncio debe ser grande para cumplir con su propósito de ser llamativo y vistoso para llamar la atención de los transeúntes, por lo general la estructura sobre la cual se monta la lona o el vinil publicitario es de alguna aleación de acero.

Existen muchos factores que pueden llegar a alterar la estructura del anuncio panorámico y generar una falla mecánica en su estructura, por eso se considera de suma importancia hacer un estudio sobre su diseño con la finalidad de hacerlo más seguro.

3. Estado del arte:

Anuncio panorámico

Un anuncio panorámico tiene que ser atractivo visualmente hablando, estos anuncios son de gran altura y lo suficientemente grandes como para ser observados desde grandes distancias, con el paso del tiempo la creatividad de estos anuncios ha ido cambiando de tal forma que se han creado ciertos tipos de anuncios publicitarios.

Estos pueden clasificar los soportes para anuncios panorámicos en 2 principales categorías:

*Monopostes: Es la estructura en la cual nos centraremos en optimizar, dicha estructura solo cuenta con un soporte cilíndrico rígido que soporta el espacio publicitario.

*Bipostes: Es la estructura que contine 2 soportes, y estos soportan el espacio publicitario, por lo general el espacio de estos anuncios es mas grande y por tanto mas vistoso, pero no es el tema a tratar en el desarrollo de este reporte.

Los monopostes al igual que los bipostes estan hechos de aceros estructurales los cuales cumplen con las características mecánicas ideales para su construcción y aplicación.

Las partes de un anuncio panorámico son:

***Mampara:** Es el area que abarca la zona u area donde se instalará el vinil del anuncio publicitario.

***Pedestal:** Es la sección cilíndrica del monoposte que soporta la mampara.

***Cartela:** Es un elemento que refuerza el pedestal en el punto de selección sobre sección transversal exterior.

***Anclas pedestal-zapata:** Elementos que refuerzan el anclaje de la cartela fijándolo con la cimentación.

***Cimentación:** Estructura rígida que se contruye para dar soporte estructural al monoposte.

***Ancla zapata-suelo:** Estructura que fija la cimentación y el suelo.

4. Marco teórico:

Para que el anuncio sea seguro y duradero se realiza un estudio de ingeniería y en donde se deben de considerar los factores que podrían generar una alteración en la estructura los cuales son:

***El viento que golpea el anuncio:** Como el clima es diferente según la región, se mantiene como variable el viento que golpea al anuncio, en ese momento se produce una fuerza que genera cierto pandeo o deformación del pedestal, por lo que el monoposte debe ser lo suficientemente rígido como para soportar tal esfuerzo.

***Impactos en el pedestal:** Algunas veces se da la situación en donde algún vehículo impacta la estructura y esto produce perturbaciones en la misma que podrían generar algún fallo estructural.

***Dimensiones del anuncio:** Para un diámetro "n" del pedestal debe de existir un límite en las dimensiones del anuncio publicitario, ya que la parte superior de la estructura genera un esfuerzo de compresión sobre la sección transversal del pedestal, por ello el pedestal debe ser lo suficientemente ancho para soportar dicho esfuerzo.

***Factor de seguridad:** El factor de seguridad es un cálculo matemático donde se consideran todos los esfuerzos que interactúan en la estructura con la finalidad de determinar si será estable o colapsará (relación entre el esfuerzo máximo y el esfuerzo de diseño), recordad que el factor de seguridad en una estructura grande siempre debe ser mucho mayor a 1, en dado caso que sea 1 se concluye que apenas la estructura llegue a su esfuerzo máximo la estructura colapsará de forma inmediata.

La finalidad de este reporte es usar la optimización topológica es una técnica englobada dentro del campo de análisis estructural. Se basa en el análisis mecánico de un componente o estructura. Su principal objetivo es el aligeramiento estructural manteniendo las funcionalidades mecánicas del componente objetivo, esto con el fin de ser lo más práctico posible.

Por ello usando de forma correcta la optimización topologica sobre una estructura se debe de conseguir el mismo factor de seguridad.

La optimización puede ser de forma, tamaño o topológica (de forma y tamaño).

El desarrollo de la optimización se puede realizar con la ayuda de un software de cálculo, en donde se ingresa un codigo que obedece un determinado modelo matemático y los parámetros ingresados determinan el resultado de la optimización.

5. Desarrollo:

Desarrollo de código

1) El primer cambio se realiza entre las líneas 5 y 6, es necesario agregar un ciclo igual a cero y realizar un cambio en la función if de la línea 9.

```
6 loop =0;
9 if (((ely-(nely*0.5)<(2*elx)-(1.36*nely)) || (ely <(1+nely*0.5))) && (elx...
10     >(1+nely)*0.6666))
```

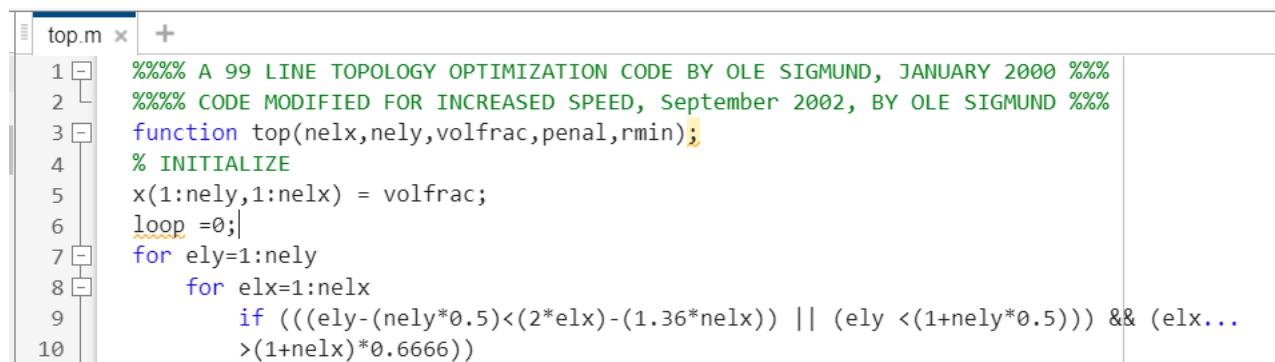


Figura 1: Líneas del código 6,9 y 10 modificadas

2) El segundo cambio se realiza entre las líneas 32 y 33 donde es necesario agregar la línea de código mostrada a continuación, de igual forma es necesario agregar parte de esta función en otra parte del código, en la línea 37.

```

33 dc(ely,elx)=0;
37 dc(ely,elx) = dc(ely,elx) -penal*x(ely,elx)^(penal-1)*Ue'*KE*Ue;

```

```

33      dc(ely,elx)=0;
34      for i = 1:5
35          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);
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;

```

Figura 2: Lineas del código 33 y 37 modificadas

3) Para el tercer cambio es necesario redefinir el ciclo while de la línea 55.

```

55 while (l2-l1 ≥ 1e-4)

```

```

52      %%%%%%%%%% OPTIMALITY CRITERIA UPDATE %%%%%%%%%%
53      function [xnew]=OC(nelx,nely,x,volfrac,dc,passive)
54      l1 = 0; l2 = 100000; move = 0.2;
55      while (l2-l1 > 1e-4)
56          lmid = 0.5*(l2+l1);
57          xnew(find(passive))=0.001;
58          xnew = max(0.001,max(x-move,min(1,min(x+move,x.*sqrt(-dc./lmid)))));

```

Figura 3: Lineas del código 55 modificadas

4) En el cuarto cambio es necesario redefinir la función F y realizar el mismo cambio en la función U, cambiando un 1 por un 5, para cambiar el número de cargas, estas funciones se encuentran en las líneas 85 y 85 respectivamente.

```

85 F = sparse(2*(nely+1)*(nelx+1),5);

```

```

86 U = zeros(2*(nely+1)*(nelx+1),5);

```

```

81      %%%%%%%%%% FE-ANALYSIS %%%%%%%%%%
82      function [U]=FE(nelx,nely,x,penal)
83      [KE] = lk;
84      K = sparse(2*(nelx+1)*(nely+1), 2*(nelx+1)*(nely+1));
85      F = sparse(2*(nely+1)*(nelx+1),5);
86      U = zeros(2*(nely+1)*(nelx+1),5);
87      for elx = 1:nelx

```

Figura 4: Lineas del código 85 y 86 modificadas

5) Finalmente, para el ultimo cambio es necesario definir las 5 cargas (a partir de la línea 96 y realizar un cambio en la función fixeddofs, quedando de la siguiente manera.

```

96 F(2*nekhx(nely+1)+2,1) = 1;

97 F(2*nely(nely+1)+(nely/4),2) = 1;

98 F(2*nely*(nely+1)+(nely/2),3) = 1;

99 F(2*nely*(nely+1)+(nely),4) = 1;

100 F(2*nely(nely+1)+(nely*1.2),5) = 1;

101 fixeddofs = 2*(nely+1)+1:2*(nelx+1):2*(nely+1)*(nely+1);

```

```

95 % DEFINE LOADS AND SUPPORTS (HALF MBB-BEAM)
96 F(2*nekhx(nely+1)+2,1) = 1;
97 F(2*nely(nely+1)+(nely/4),2) = 1;
98 F(2*nely*(nely+1)+(nely/2),3) = 1;
99 F(2*nely*(nely+1)+(nely),4) = 1;
100 F(2*nely(nely+1)+(nely*1.2),5) = 1;
101 fixeddofs = 2*(nely+1)+1:2*(nelx+1):2*(nely+1)*(nely+1);
102 alldofs    = [1:2*(nely+1)*(nelx+1)];
103 freedofs   = setdiff(alldofs,fixeddofs);

```

Figura 5: Líneas del código 96 a 101 modificadas

Finalmente, para obtener la optimización del panorámico, es suficiente correr el código especificando las siguientes variables en la ventana de comandos, con esto obtendremos el resultado de la siguiente figura. `top(40,80,0.2,3.0,0.5)`

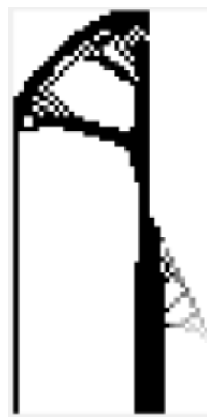


Figura 6: Panorámico optimizado

Código completo

```

%%% A 99 LINE TOPOLOGY OPTIMIZATION CODE BY OLE SIGMUND, JANUARY 2000 %%%
%%% CODE MODIFIED FOR INCREASED SPEED, September 2002, BY OLE SIGMUND %%%
function top(nelx,nely,volfrac,penal,rmin);
% INITIALIZE
x(1:nely,1:nelx) = volfrac;
loop =0;
for ely=1:nely
    for elx=1:nelx
        if (((ely-(nely*0.5)<(2*elx)-(1.36*nelx)) || (ely <(1+nely*0.5))) && (elx...
            >(1+nelx)*0.6666))
            passive(ely,elx)=1;
        else
            passive(ely,elx)=0;
        end
    end
end
x(find(passive))=0.001
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;
        dc(ely,elx)=0;
        for i = 1:5
            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) = dc(ely,elx) -penal*x(ely,elx)^(penal-1)*Ue'*KE*Ue;
        end
    end
end
end

```



```

% FILTERING OF SENSITIVITIES
[dc] = check(nelx,nely,rmin,x,dc);
% DESIGN UPDATE BY THE OPTIMALITY CRITERIA METHOD
[x] = OC(nelx,nely,x,volfrac,dc,passive);
% PRINT RESULTS
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 )])
% PLOT DENSITIES
colormap(gray); imagesc(-x); axis equal; axis tight; axis off; pause(1e-6);
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [xnew]=OC(nelx,nely,x,volfrac,dc,passive)
l1 = 0; l2 = 100000; move = 0.2;
while (l2-l1 > 1e-4)
    lmid = 0.5*(l2+l1);
    xnew(find(passive))=0.001;
    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
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
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-floor(rmin),1):min(i+floor(rmin),nelx)
            for l = max(j-floor(rmin),1):min(j+floor(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
end

```

```

%%%%%%%%%% FE-ANALYSIS %%%%%%%%%%%
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),5);
U = zeros(2*(nely+1)*(nelx+1),5);
for elx = 1:nelx
    for ely = 1:nely
        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
% DEFINE LOADS AND SUPPORTS (HALF MBB-BEAM)
F(2*nekhx(nely+1)+2,1) = 1;
F(2*nelx(nely+1)+(nely/4),2) = 1;
F(2*nelx*(nely+1)+(nely/2),3) = 1;
F(2*nelx*(nely+1)+(nely),4) = 1;
F(2*nelx(nely+1)+(nely*1.2),5) = 1;
fixeddofs =2*(nely+1)+1:2*(nelx+1):2*(nely+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;
%%%%%%%%%% ELEMENT STIFFNESS MATRIX %%%%%%%%%%%
function [KE]=lk
E = 2*10^11;
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)];
%

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% This Matlab code was written by Ole Sigmund, Department of Solid %
% Mechanics, Technical University of Denmark, DK-2800 Lyngby, Denmark. %
% Please sent your comments to the author: sigmund@fam.dtu.dk %
% %
% The code is intended for educational purposes and theoretical details %
% are discussed in the paper %
% "A 99 line topology optimization code written in Matlab" %
% 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 %
% downloaded from the web-site: http://www.topopt.dtu.dk %
% %
% Disclaimer: %
% The author reserves all rights but does not guaranty that the code is %
% free from errors. Furthermore, he shall not be liable in any event %
% caused by the use of the program. %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

6. Conclusiones:

Betsaida Alejandra Ruedas Vázquez:

Al haber concluido la practica sobre el diseño de la estructura de un panorámico pude desarrollar propuestas para el análisis de formas así mismo de la programación para la ejecución de la optimización con sus características y ventajas.

Entiendo que 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.

Miguel Rodrigo Aguilar Moreno:

El desarrollo de una estructura panorámica resultó ser un análisis interesante, ya que se requieren considerar muchos factores externos para su correcta construcción.

Además de los factores físicos, los factores digitales al momento de realizar su optimización son considerables y hay que tener su debida precaución en ello.

Erick Daniel Esquivel Arguelles:

Fue un poco complicado asimilar como funciona el código del programa, pero a medida que avanzó la práctica entendí que es más fácil analizar el código por partes, y no completo, entendiendo que las optimizaciones en la estructura no se hacen de forma simultánea, sino que existe todo un proceso que se compone funciones que analizan los parámetros de entrada que resulta en un análisis de la estructura y en una propuesta de optimización de la misma.

Omar Isaí Moreno Cruz:

A lo largo de las prácticas hemos utilizado el mismo código haciendo modificaciones de algunas líneas de éste para así obtener nuevas optimizaciones. En la práctica anterior fue un análisis hacia la estructura de una bicicleta, en este caso, el de un anuncio panorámico para el cual se agregaron ciclos, se redefinieron funciones y cargas. Al final se corrió el código obteniéndose la optimización del panorámico.

Saul Moises Mendoza Cida:

Esta práctica fue un poco más sencilla de realizar, gracias a que ya se conocía el programa, siendo este el mismo que se ha utilizado en las prácticas anteriores, con esto solo se tuvieron que realizar los cambios correspondientes para que el código pudiera optimizar el panorámico. Gracias a todas estas aplicaciones que se han estado trabajando, cada vez, se comprueba más la versatilidad de este código y que sus funciones de optimización son aplicables en varias áreas de la ingeniería siendo bastante útil tener un código que permita, a pesar de no tener un gran detalle, optimizar estructuras cotidianas, las cuales vemos o usamos todos los días.

Referencias

- [1] Elliott, C., Hintermüller, M., Leugering, G., and Sokolowski, J. (2011). Special issue on advances in shape and topology optimization: theory, numerics and new applications areas. *Optim. Methods Softw.*, 26(4-5):511–512.
- [2] Fujii, D., Ejima, S., and Kikuchi, N. (2000). Topology optimization of compliant mechanisms using the homogenization design method. *J. Struct. Constr. Eng.*, 65(528):99–105.
- [3] Katsuno, E. T., Dantas, J. L. D., and Silva, E. C. N. (2022). Topology optimization of low-friction painting distribution on a marine propeller. *Struct. Multidiscipl. Optim.*, 65(9).
- [4] Kikuchi, N., N. S. F. L. S. O. . S. E. C. N. (1998). Design optimization method for compliant mechanisms microstructure. *Methods Appl. Mech. Engrg.*, 151(4-5):401 –417.
- [5] Kota, S., J. J. L. Z. R. S. M. . S. J. (2001). Design of compliant mechanisms: Applications to mems. *Analog Integrated Circuits and Signal Processing*, 29:7 –15.
- [6] Larsen, U. D., Signund, O., and Bouwsta, S. (1997). Design and fabrication of compliant micro-mechanisms and structures with negative poisson’s ratio. *J. Microelectromech. Syst.*, 6(2):99–106.
- [7] Lin, J., Luo, Z., and Tong, L. (2010). A new multi-objective programming scheme for topology optimization of compliant mechanisms. *Struct. Multidiscipl. Optim.*, 40(1-6):241–255.
- [8] Maute, K. Allen, M. (2004). . conceptual design of aeroelastic structures by topology optimization. *Struct Multidisc Optim*, 27(27 - 42).
- [9] Pingen, G., Evgrafov, A., and Maute, K. (2007). Topology optimization of flow domains using the lattice boltzmann method. *Struct. Multidiscipl. Optim.*, 34(6):507–524.
- [10] Sigmund, O. and Clausen, P. M. (2007). Topology optimization using a mixed formulation: An alternative way to solve pressure load problems. *Comput. Methods Appl. Mech. Eng.*, 196(13-16):1874–1889.

Los créditos de las fotografías pertenecen a sus respectivos autores.