



**UANL**

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN®



**FIME**

FACULTAD DE INGENIERÍA MECÁNICA Y ELÉCTRICA

## **LABORATORIO DE BIOMECÁNICA**

**PERIODO ESCOLAR AGOSTO – DICIEMBRE 2022**

**Nombre de la actividad**

**Práctica 3**

<b>Nombre del alumno</b>	<b>Matricula</b>
<b>Leonardo Daniel De Leon Fuentes</b>	<b>1991978</b>
<b>Fatima Montserrat Castro Nuñez</b>	<b>1991834</b>
<b>Brandon Geovanny Espinosa Alcocer</b>	<b>1938292</b>
<b>Erick Eduardo Landa Gonzalez</b>	<b>1992037</b>
<b>Susana Regina Macías de Luna</b>	<b>1992013</b>

<b>Grupo</b>	<b>Salon</b>	<b>Dia clase</b>	<b>Hora</b>
<b>309</b>	<b>12BMC</b>	<b>Miércoles</b>	<b>N5</b>

**Nombre del profesor: Yadira Moreno Vera**

**Fecha de entrega:**

**26/10/2022**

## **Objetivo:**

**Presentar una propuesta de análisis de formas y de la programación para la ejecución de la optimización de características de trabajo específicas.**

### **1) Nombre y definición de la forma GEOMETRÍA**

Diseño de la estructura de un panorámico por medio de código del software MATLAB tomando en cuenta las diferentes cargas que se implementan, incluyendo los apoyos y esfuerzos que forman parte del diseño del panorámico. La estructura panorámica es el soporte en el cual se coloca un anuncio de publicidad, este puede ser de una cara o de incluso hasta tres caras.

Los panorámicos generalmente se posicionan en medio de las ciudades para publicitar algún producto, servicio o simplemente transmitir un mensaje a las personas que observen el anuncio.

Hay diferentes materiales que se utilizan para la realización de las estructuras de los panorámicos, los más usados son metal y acero ya que cuentan con las propiedades más óptimas respecto a la resistencia de clima, lluvia o algún otro fenómeno natural que pueda dañar la estructura del panorámico.

### **2) Estado del arte**

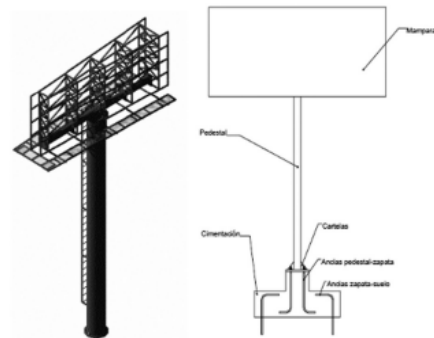
Las estructuras panorámicas son soportes en donde pueden ir ubicados diferentes tipos de anuncios publicitarios. La ventaja de estas estructuras es que pueden ir ubicadas en diferentes paisajes urbanos con el fin de promocionar un producto o servicio.

Los materiales adecuados para la construcción de Panorámicos pueden ser elaborados en plástico, lona, telas, PVC, acrílico o metal. Aunque también, puede optar por panorámicos electrónicos y agregar elementos como luces, e incluso música.

Este tipo de estructura aparentemente es muy sencilla porque cuenta solamente con tres partes principales que son: la mampara, el pedestal y la cimentación.

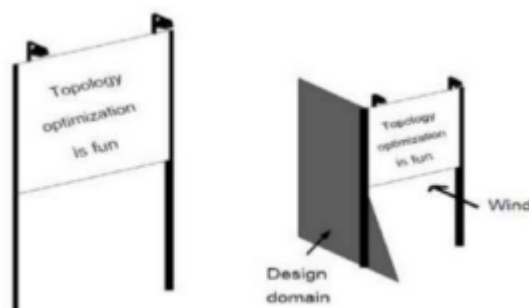
Sin embargo, vista en forma minuciosa, una mampara consta de varios componentes y accesorios que hacen que esta estructura sea realmente muy compleja tanto en su diseño estructural, como en su construcción y también en su comportamiento sobre todo ante viento como el producido por huracán o tromba. Tanto la cimentación como el pedestal y la mampara elevada pueden constar de

diversos elementos tales como: anclas suelo–zapata, vigas estabilizadoras, anclas pedestal–zapata, lastres, placas-base, el tubo del pedestal, escaleras externas e internas, andamios, placas de conexión pedestal-mampara, travesaño principal de la mampara, placas verticales del travesaño, armaduras, pernos de sujeción, láminas de la mampara, accesorios de iluminación, ganchos o argollas de sujeción, travesaños secundarios; además, se tienen diversos elementos de sujeción o conexión tales como pernos, tornillos, remaches, soldaduras, etc.



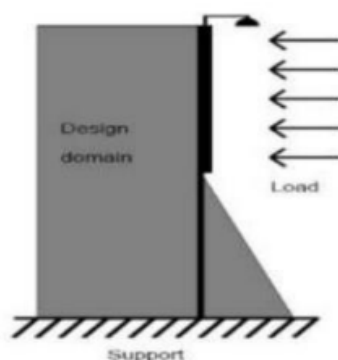
*Partes de un panorámico usual.*

En la figura 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:



*Espacio de diseño*

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.



Se debe planear una geometría para poder realizar los cálculos necesarios como tubo de pedestal, donde se requiere calcular la relación de esbeltez:

$$R_E = \frac{kl}{r}$$

donde:

k = factor de longitud efectiva

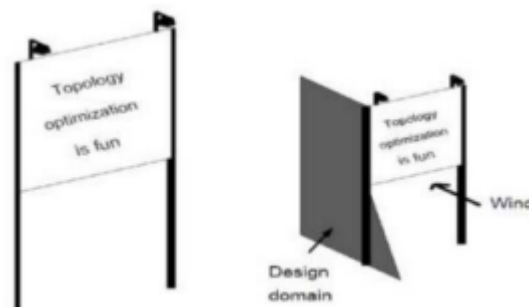
l = longitud del elemento

RE = Relación de Esbeltez

r = radio de giro

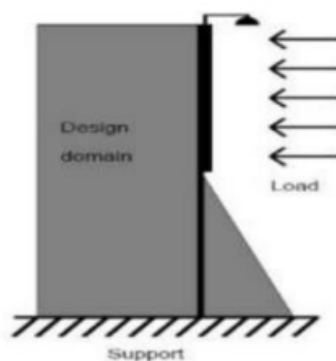
### 3) Propuesta de diseño de la geometría, alcances y limitaciones

En la figura 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:



*Espacio de diseño*

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.



#### 4) Pasos del desarrollo de la programación

Se utiliza el código de la práctica pasada como base para esta optimización.

1) Cambios en % INITIALIZE (fig. 4.1): en esta sección se agrega la definición de un ciclo igual a 0 (`loop = 0`), además de que se cambia las especificaciones de la condición `if` por

```
if ((ely-(nely*0.5)<(2*elx)-(1.36*nely)) || (ely
    <(1+nely*0.5))) &&(elx...>(1+nely)*0.6666))
```

```
4      % INITIALIZE
5      x(1:nely,1:nely) = volfrac;
6      loop = 0;
7      for ely = 1:nely
8          for elx = 1:nely
9              if ((ely-(nely*0.5)<(2*elx)-(1.36*nely)) || (ely <(1+nely*0.5))) &&(elx...
10                 >(1+nely)*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.;
```

Figura 4.1) Primera modificación.

2) Cambios en % OBJECTIVE FUNCTION AND SENSITIVITY ANALYSIS (fig. 4.2):

Se define `dc(ely,elx) = 0.`; se agrega un ciclo `for i = 1:5` y dentro de él se mueven los valores de `Ue`, `c` y `dc` que ya se tenían; se modifica la definición `dc` por `dc(ely,elx) = dc(ely,elx)-penal*x(ely,elx)^(penal-1)*Ue'*KE*Ue`

```
25      % OBJECTIVE FUNCTION AND SENSITIVITY ANALYSIS
26      [KE] = lk;
27      c = 0.;
28      for ely = 1:nely
29          for elx = 1:nely
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) = dc(ely,elx)-penal*x(ely,elx)^(penal-1)*Ue'*KE*Ue;
37              end
38          end
39      end
```

Figura 4.2) Segundo cambio

3) Cambios en OPTIMALITY CRITERIA UPDATE (fig. 4.3): El único cambio que se hace aquí es volver a definir el ciclo while como `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 = max(0.001,max(x-move,min(1.,min(x+move,x.*sqrt(-dc./lmid)))));
58          xnew(find(passive)) = 0.001;
59          if sum(sum(xnew)) - volfrac*nelx*nely > 0
60              l1 = lmid;
61          else
62              l2 = lmid;
63          end
64      end

```

Figura 4.3) Tercera modificación.

4) Cambios en FE-ANALYSIS (fig. 4.4): En esta sección se cambia la definición de `F` por `F = sparse(2*(nely+1)*(nelx+1),5);` y también se agrega la definición de `U = zeros(2*(nely+1)*(nelx+1),5);`.

```

81      %%%%%%%%%%% FE-ANALYSIS %%%%%%%%%%%
82      function [U]=FE(nelx,nely,x,penal)
83          [KE] = 1k;
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 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

```

Figura 4.4) Cuarta modificación.

5) Cambios en DEFINE LOADS AND SUPPORTS(HALF MBB-BEAM (fig. 4.5): En esta sección se agregan las 5 cargas que se suponen afectan a un panorámico, y se cambia el soporte por `fixeddofs = 2*(nely+1):2*(nely+1):2*(nelx+1)*(nely+1)`.

```

95      % DEFINE LOADS AND 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),5) = 1;

```

Figura 4.5) Quinta modificación.

**Codigo 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;
```

```
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

% 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

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% OPTIMALITY CRITERIA UPDATE %%%%%%%%%
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 = max(0.001,max(x-move,min(1.,min(x+move,x.*sqrt(-dc./lmid)))));

    xnew(find(passive)) = 0.001;

    if sum(sum(xnew)) - volfrac*nelx*nely > 0

        l1 = lmid;

    else

        l2 = lmid;

    end

end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% MESH-INDEPENDENCY FILTER %%%%%%%%%
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

    end

end

```

```

end

dcn(j,i) = dcn(j,i)/(x(j,i)*sum);

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 ely = 1:nely

    for elx = 1:nelx

        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*nelx*(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):2*(nely+1):2*(nelx+1)*(nely+1);

alldofs = 1:2*(nely+1)*(nelx+1);

```

```

freedofs = setdiff(alldofs,fixeddofs);

% SOLVING 127

U(freedofs,:) = K(freedofs,freedofs) \F(freedofs,:);

U(fixeddofs,:)= 0;

%%%%%%%%%%%%% ELEMENT STIFFNESS MATRIX %%%%%%%%%%%%%%

function [KE]=lk

E = 1.;

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)];

```

## 5) Resultados de la optimización

Se obtuvo el diseño de la estructura panorámica como se muestra en la figura 5



Figura 5. Resultado de la optimización

## 6) Conclusiones

**Erick Eduardo Landa Gonzalez 1992037**

Para esta práctica como la metodología en las actividades anteriores, por medio de la codificación en el software de MATLAB se obtiene la optimización de los esfuerzos en este caso para la estructura de un panorámico, es importante hacer mención que los espacios vacíos de la estructura también son importantes ya que estos resultan ser elementos pasivos de la estructura, por lo tanto se adquiere el conocimiento de que también los elementos pasivos son importantes al momento de llevar a cabo un diseño teniendo la misma importancia que los demás elementos que lo conforman.

**Fatima Montserrat Castro Nuñez 1991834**

En esta práctica volvimos a editar el código original de la primera práctica, nosotros utilizamos el código de la segunda practica para empezar a realizar las debidas modificaciones para cumplir con el diseño del panoramico considerando las furzas y el soporte del mismo. Considero que esta práctica nos sirve para ampliar nuestra visalización de las diferentes aplicaciones que le podemos dar a MATLAB para optimizar topologias. Esto es muy

**Leonardo Daniel De Leon Fuentes 1991978**

En esta práctica hicimos uso del código para aplicarlo en una geometría que se definió, pudimos ver como cambia su forma gracias a las cargas que se le aplican, haciéndola de esta forma más resistente a este tipo de cargas, aplicamos lo visto en prácticas anteriores.

**Susana Regina Macías de Luna 1992013**

En esta práctica número 3, su objetivo era el presentar una propuesta de análisis de formas donde también se ve la programación para la ejecución de la optimización del mismo. Donde se editó también el código de la primera práctica utilizando la ayuda del software Matlab.

**Brandon Geovanny Espinosa Alcocer 1938292**

En esta práctica número 3 al igual que los demás usamos matlab paravpoder sacar todos los datos de como se va a construir lo que se pidió así como la imagen de formas la cual es la geometría.

