

Práctica 3

Diseño de la estructura de un panorámico

1845056 Aurora Nahomy Martínez Pérez
1854324 Kevin Orlando Huerta Jaramillo
1896681 Jovanny Daniel Alvarado Ramírez
1909876 Fernando Herrera Garza
1991876 Bernardo Gil Villarreal

21 de octubre de 2022

1. Objetivo

Desarrollar en el estudiante la capacidad de análisis, implementación y solución de un problema propuesto. Presentar una propuesta de análisis de formas y de la programación para la ejecución de la optimización de características de trabajo específicas que presenta las ventajas.

2. Introducción

En esta práctica se desarrollará el diseño para una estructura de un panorámico, en el cual se tomarán en cuenta las fuerzas y las cargas a las que se expone el panorámico, tales como el viento, para poder desarrollar el diseño adecuado para el problema.

3. Marco Teórico

3.1. Panorámicos

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

Se menciona a aquellos panorámicos que se pondrán en un 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 los soportes para los anuncios publicitarios. Las estructuras panorámicas se exponen a altas ráfagas de viento, por lo que su estructura ocupa ser rígida para soportar estas fuerzas.

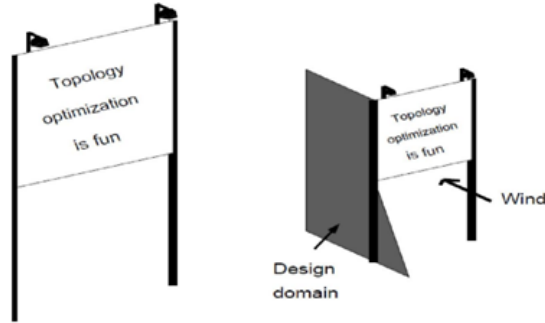


Figura 1: Panorámico

Este tipo de estructura aparentemente es muy sencilla porque cuenta solamente con tres partes principales que son: la mampara, el pedestal y la cimentación; en la Figura anterior se muestran esquemáticamente estas tres partes. 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, acartelamientos, el tubo del pedestal, escaleras externas e internas, andamios, placas de conexión pedestal-mampara, travesano principal de la mampara, placas verticales del travesano, armaduras, pernos de sujeción, láminas de la mampara, accesorios de iluminación, ganchos o argollas de sujeción, travesanos secundarios; además, se tienen diversos elementos de sujeción o conexión tales como pernos, tornillos, remaches, soldaduras, etc.

[1] [2]

4. Metodología

4.1. Propuesta de diseño de la geometría, alcances y limitaciones

Los panorámicos se exponen a altas ráfagas de viento, por lo que su estructura ocupa ser muy rígida para soportar estas fuerzas. El espacio de diseño a evaluar será de 2 dimensiones, las cargas y los apoyos.

En la práctica se busca realizar un diseño de la estructura de un panorámico utilizando la optimización topológica para hacer este diseño. Los panorámicos se enfrentan a distintas o diferentes cargas como las cargas de viento, de sismos, muertas o vivas.

La característica principal de estos panorámicos hace hincapié en la forma en que se conectan los dichos elementos que no conforman para darle una estabilidad al sistema, a su vez dándole forma en que debe fabricarse e instalarse la estructura.

Lo más común de donde se ubican dichas estructuras es en vías públicas donde se pueden encontrar la mayor parte de la población dependiendo dicho sector de la ciudad, teniendo en cuenta esto, se debe garantizar su seguridad porque su colapso podría afectar a muchas personas. Los rótulos deben ser diseñados de una forma conservadora para proteger a las personas para evitar una catástrofe, como así también evitar pérdidas económicas directamente, como por ejemplo los costos de la estructura misma y daños secundarios.

Gracias a la optimización topológica se busca la reducción del volumen de la estructura del panorámico y que soporte las cargas necesarios que esta será sometida como por ejemplo la carga del viento mediante movimientos horizontales de masas de aire provocado por las diferentes unidades de presión en los distintos sectores de la atmósfera y teniendo en cuenta la rotación de la tierra, para estos movimientos que ocurren habitualmente es esencial tomar en cuenta la velocidad y presión de los vientos para llevar a cabo el diseño y la sumatoria de las fuerzas que se estarían aplicando en dicha estructura.

La optimización topológica estructural determina el espacio de diseño mínimo permisible que es necesario para optimizar la forma del producto. Después entramos en detalle para la forma virtual y con un software de optimización para comprobar y analizar la aplicación de las cargas y esfuerzos que contará la estructura desde distintos ángulos y áreas para la comprobación de que material utilizaremos.

5. Desarrollo

5.1. Propuesta

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:



Figura 2: Espacio de diseño

5.2. Ejercicio con el espacio de diseño propuesto

Para realizar esta práctica, tenemos que hacer algunos cambios a las líneas de código de nuestro script que hemos usado en las últimas prácticas. Para empezar la modificación, tenemos que usar 5 fuerzas y cambiarlas de posición. Para ello tenemos que modificar instrucción fixeddofs, para esto se modificarán las siguientes líneas:

Líneas originales

```

76  %***** FE-ANALYSIS *****
77  function [U]=FE(nelx,nely,x,penal)
78  [KE] = lk;
79  K = sparse(2*(nelx+1)*(nely+1), 2*(nelx+1)*(nely+1));
80  F = sparse(2*(nely+1)*(nelx+1),1); U = sparse(2*(nely+1)*(nelx+1),1);
..

```

Líneas modificadas

```

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); U = zeros(2*(nely+1)*(nelx+1),5);
86  for ely = 1:nely
87  for elx = 1:nelx

```

Líneas originales

```

26 - for ely = 1:nely
27 -     for elx = 1:nelx
28 -         n1 = (nely+1)*(elx-1)+ely;
29 -         n2 = (nely+1)* elx +ely;
30 -         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);
31 -         c = c + x(ely,elx)^penal*Ue'*KE*Ue;
32 -         dc(ely,elx) = -penal*x(ely,elx)^(penal-1)*Ue'*KE*Ue;

```

Líneas modificadas

```

25 - % OBJECTIVE FUNCTION AND SENSITIVITY ANALYSIS
26 - [KE] = lk;
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; 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

```

Líneas originales

```

87 - % DEFINE LOADSAND SUPPORTS (HALF MBB-BEAM)
88 - F(2,1) = 1;
89 - fixeddofs = 2*nelx*(nely+1)+1:2*(nelx+1)*(nely + 1);
90 - alldofs = [1:2*(nely+1)*(nelx+1)];
91 - freedofs = setdiff(alldofs,fixeddofs);

```

Líneas modificadas

```

87 - % DEFINE LOADSAND SUPPORTS (HALF MBB-BEAM)
88 - F(2*(nelx)*(nely+1)+2,1)=1;
89 - F(2*(nelx)*(nely+1)+20,1)=1;
90 - F(2*(nelx)*(nely+1)+40,1)=1;
91 - F(2*(nelx)*(nely+1)+60,1)=1;
92 - F(2*(nelx)*(nely+1)+80,1)=1;
93 - fixeddofs = 2*nelx*(nely+1)+1:2*(nelx+1)*(nely + 1);
94 - alldofs = [1:2*(nely+1)*(nelx+1)];
95 - freedofs = setdiff(alldofs,fixeddofs);

```

Ahora, el siguiente paso es realizar el empotramiento diagonal, y esto se realizará usando una sección del código que nos habla sobre los elementos pasivos el cual sirve de ayuda para determinar un espacio en blanco. Tendremos que realizar las siguientes modificaciones al código:

```

2 - function Practica3(nelx,nely,volfrac,penal,rmin)
3 - % INITIALIZE
4 - x(1:nely,1:nelx) = volfrac;
5 - loop = 0;
6 - %Se declara el espacio en blanco
7 - for ely = 1:nely
8 -     for elx = 1:nelx
9 -         if (((ely-(nely*0.5)<(2*elx)-(1.36*nelx)) & (ely < (1+nely*0.5))) & (elx
10 -             > (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.;

```

Líneas originales

```

36 | % DESIGN UPDATE BY THE OPTIMALITY CRITERIA METHOD
37 - | [x] = OC(nelx,nely,x,volfrac,dc);

46 | %%%%%%%%%% OPTIMALITY CRITERIA UPDATE %%%%%%%%%%
47 | function [xnew]=OC(nelx,nely,x,volfrac,dc)

```

Líneas modificadas

```

36 | % DESIGN UPDATE BY THE OPTIMALITY CRITERIA METHOD
37 - | [x] = OC(nelx,nely,x,volfrac,dc,passive);

46 | %%%%%%%%%% OPTIMALITY CRITERIA UPDATE %%%%%%%%%%
47 | function [xnew]=OC(nelx,nely,x,volfrac,dc,passive)

```

Líneas originales

```

47 | function [xnew]=OC(nelx,nely,x,volfrac,dc,passive)
48 - | l1 = 0; l2 = 100000; move = 0.2;
49 - | while ((l2-l1)/l2 > 1e-4)
50 - |     lmid = 0.5*(l2+l1);
51 - |     xnew = max(0.001,max(x-move,min(1.,min(x+move,x.*sqrt(-dc./lmid)))));
52 - |     if sum(sum(xnew)) - volfrac*nelx*nely > 0;

```

Líneas modificadas

```

47 | function [xnew]=OC(nelx,nely,x,volfrac,dc,passive)
48 - | l1 = 0; l2 = 100000; move = 0.2;
49 - | while ((l2-l1)/l2 > 1e-4)
50 - |     lmid = 0.5*(l2+l1);
51 - |     xnew(find(passive))=0.001
52 - |     xnew = max(0.001,max(x-move,min(1.,min(x+move,x.*sqrt(-dc./lmid)))));
53 - |     if sum(sum(xnew)) - volfrac*nelx*nely > 0;

```

5.3. Código final implementado

```

function Practica3(nelx,nely,volfrac,penal,rmin);
x(1:nely,1:nelx) = volfrac;
loop = 0;
change = 1.;
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.;
while change > 0.01
loop = loop + 1;
xold = x;
[U]=FE(nelx,nely,x,penal);
[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
[dc] = check(nelx,nely,rmin,x,dc);
[x] = OC(nelx,nely,x,volfrac,dc,passive);
change = max(max(abs(x-xold)));
disp(['It.: ' sprintf('
' Vol.: ' sprintf('
' ch.: ' sprintf('
colormap(gray); imagesc(-x); axis equal; axis tight; axis off; pause(1e6);
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 = 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
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-round(rmin),1):min(i+round(rmin),nelx)
for l = max(j-round(rmin),1):min(j+round(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
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
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);

```

```

U(freedofs,:) = K(freedofs,freedofs) (freedofs,:);
U(fixeddofs,:)= 0;
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)];

```

6. Resultado

Primera optimización obtenida

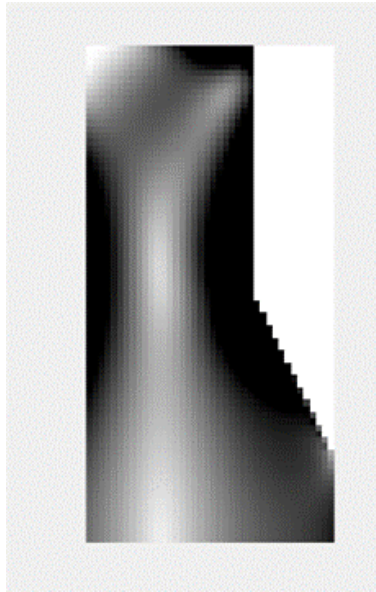


Figura 3: Primer resultado

Resultado final

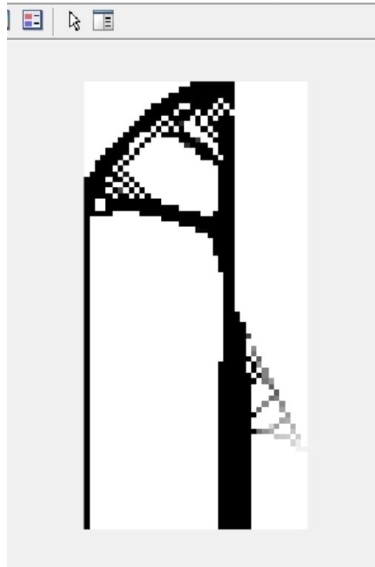


Figura 4: Resultado final

7. Conclusiones

Bernardo Gil Villarreal 1991876

Gracias a la realización de esta práctica hicimos uso de análisis de elemento finito para objetos, y la finalidad de esto es que además esto nos representa una importante herramienta para la exploración y la extracción de conocimiento teniendo como objetivo de saber tomar decisiones o hacer predicciones de que será mejor para la creación de objetos, esto se logró gracias al software de matlab para la optimización panorámico mediante el código utilizado.

Jovanny Daniel Alvarado Ramírez 1896681

Sabemos que la optimización de formas es un procedimiento innovador y relativamente nuevo, por lo que su estudio es de gran utilidad ya que es muy probable que se aplique en el campo laboral. Ahora, hay que recalcar que el uso de este gran procedimiento implica una mayor cantidad de recursos utilizados, así que dependiendo de la optimización que se pretenda realizar, habría que estimar los costos del proceso para lograr un balance entre el resultado que se busca y el obtenido. En este caso se propuso una forma simple sobre la cual trabajar, comento que nuestro resultado es bastante satisfactorio debido a que utilizamos un software sencillo y útil como lo es MATLAB para realizar el proceso, esto resultó en una optimización bastante buena en comparación con la propuesta.

Aurora Nahomy Martínez Pérez 1845056

Con esta práctica, vimos más elementos que podemos aplicar para la realización de la optimización topológica, como los elementos pasivos, que nos permiten hacer algunas estructuras con diferentes partes, modificando algunas secciones específicas de nuestro código, también tomando en cuenta las cargas que necesitamos aplicar a la estructura y demás aspectos relacionados a la optimización. La utilización de Matlab con el código de optimización topológica, resultó sencilla, solamente conociendo las partes del código que necesitamos cambiar.

Kevin Orlando Huerta Jaramillo 1854324

En la realización de esta práctica se pudo observar el comportamiento de la estructura del panorámico al cambiar al momento de aplicar el código, pudiendo observar mediante la aplicación del método de elemento finito las zonas de más carga en el panorámico y así poder determinar las zonas que se optimizarán.

Fernando Herrera Garza 1909876

El tiempo que llevo la realización de esta práctica fue mucho mayor al tiempo que tomo realizar las practicas anteriores, debido a que el proceso fue mucho más pesado, y por ello Matlab se demoro más en el proceso, también

haciendo posible que el programa se llegue a congelar si no se cuenta con un equipo de computo lo suficientemente bueno. Como se trató de una estructura más compleja, los elementos pasivos nos pueden llegar a afectar el resultado de la optimización.

Referencias

- [1] Louis F Geschwindner Lawrence G Griffis, John L Gross Jerome F Hajjar, Patrick M Hassett Tony C Hazel, Mark V Holland Ronald J Janowiak, Richard C Kaehler Lawrence A Kloiber, Lawrence F Kruth Jay W Larson, Roberto T Leon James O Malley, and Sanjeev R Malushte. specification-for-structural-steel-buildings-360-10.pdf.
- [2] NTC and DC. Normas técnicas complementarias para diseño y construcción de estructuras de concreto. *Gaceta oficial de la Ciudad de México*, 2017.