Práctica 3

Adrián Meléndez Herrera

Juan José Prado Luna

Ana Karen Mendoza González

Daniel Soto Celis

José Sebastián Gálvez Campos

Adrián Sandoval Toscano

18 de octubre de 2022

Resumen

Realización de un panorámico, aplicando fuerzas y un cambio de posición del anclaje, además de la creación un empotramiento diagonal.

1. Nombre y definición de la geometría

Marco de bicicleta

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:

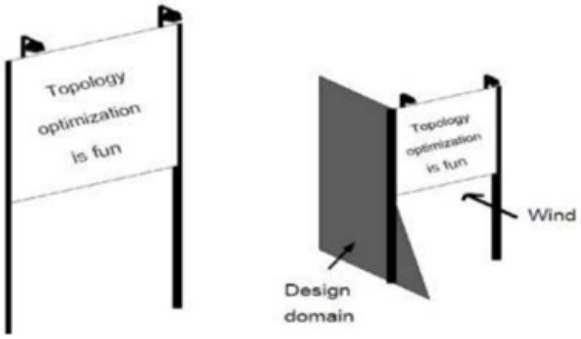

Figura 1. Imagen del panorámico.

Figura 2: Marco de bicicleta a definir

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 rigido 1, y sus patas son del mismo material que el marco.

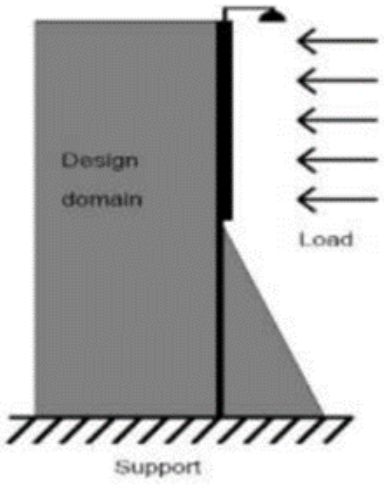

Figura 2. Espacio de diseño

Figura 3: Marco de bicicleta a definir

2. Estado del arte

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 rigida para soportar estas fuerzas.

Existen diversos materiales que las agencias especializadas usan en la creación y diseño para estas estructuras. Usualmente los postes panorámicos están construidos 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 acrilico. También existen espectaculares digitales o electrónicos que tienen luces, pantallas eléctricas y música.

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

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

Figura 4: propuesta de diseño

4. Pasos del desarrollo de la programación

1. Para empezar tenemos que editar nuestro script topp, se tiene guardado como topp3, para poder ingresar las fuerzas que requerimos, si observamos nos encontramos con 5 y para cambiar el anclaje del espacio de diseño a otra posición se tiene que cambiar la línea con la instrucción fixeddofs, para esto se modificaran las siguientes líneas:

```
%13 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; %19
 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
```

Figura 5: Código modificado para cargas y soportes

```
% DEFINE LOADSAND 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;
```

Figura 6: Código modificado para cargas y soportes

2. Después, Para crear el empotramiento diagonal, o crear el espacio en blanco para recrear el empotramiento en la parte inferior derecha; en el archivo del uso del código de 99 líneas existe una sección donde se habla de elementos pasivos el cual sirve de ayuda para determinar un espacio en blanco, en el ejemplo del archivo viene como hacer un circulo, y nosotros necesitamos un rectángulo y un triángulo para esto se modificaron y/o agregaron las siguientes líneas:

```
%Declarando vacio
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)*0.6
  passive(ely,elx) = 1;
  else
  passive(ely,elx) = 0;
  end
 end
end
x(find(passive))=0.001;
```

Figura 7: Código modificado para la zona pasiva

3. Sin embargo, aun con estas dos modificaciones queda poner las penalizaciones que contendrá la geometría durante nuestro código donde la sintaxis de la función es: top(nelx,nely,volfrac,penal,rmin) por lo tanto, la penalización es top(40,80,0.2,3.0,0.5).

4. Por último, hay que ejecutar el programa desde la ventana de comando de MATLAB.

```
>> topp3(40,80,0.2,3.0,0.5);
It.:    1Obj.:42819847281.0602 Vol.:  0.200 ch.:  0.200
Warning: MATLAB has disabled some
advanced graphics rendering features by
switching to software OpenGL. For more
information, click here.
It.:    2Obj.:42819839234.6959 Vol.:  0.200 ch.:  0.200
It.:    3Obj.:42819838251.5864 Vol.:  0.200 ch.:  0.200
It.:    4Obj.:42819837687.0633 Vol.:  0.200 ch.:  0.200
It.:    5Obj.:42819837275.8039 Vol.:  0.200 ch.:  0.200
It.:    6Obj.:42819837054.2414 Vol.:  0.200 ch.:  0.200
It.:    7Obj.:42819836942.4498 Vol.:  0.200 ch.:  0.200
It.:    8Obj.:42819836891.8488 Vol.:  0.200 ch.:  0.200
It.:    9Obj.:42819836869.1915 Vol.:  0.200 ch.:  0.200
It.:   10Obj.:42819836854.7881 Vol.:  0.200 ch.:  0.200
It.:   11Obj.:42819836847.1769 Vol.:  0.200 ch.:  0.200
It.:   12Obj.:42819836842.3217 Vol.:  0.200 ch.:  0.200
It.:   13Obj.:42819836838.5312 Vol.:  0.200 ch.:  0.200
It.:   14Obj.:42819836835.9110 Vol.:  0.200 ch.:  0.200
It.:   15Obj.:42819836834.3136 Vol.:  0.200 ch.:  0.200
```

Figura 8: Pantalla de comando de MATLAB

5. Resultados de la optimización

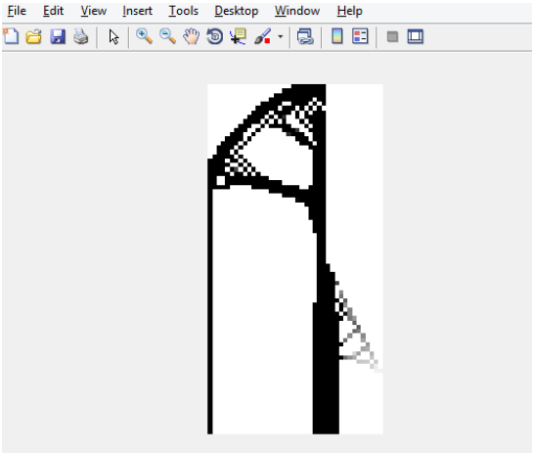

Figura 9: Resultado de geometría final

5.1. Código final

```
%%% A 99 LINE TOPOLOGY OPTIMIZATION CODE BY OLESIGMUND, OCTOBER 1999 %%%
function top(nelx,nely,volfrac,penal,rmin);
% INITIALIZE
x(1:nely,1:nelx) = volfrac;
loop = 0;
%Declarando vacio
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);
%13 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; %19
 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
%25 FILTERING OF SENSITIVITIES
```

```
[dc] = check(nelx,nely,rmin,x,dc);
%27 DESIGN UPDATE BY THE OPTIMALITY CRITERIA METHOD
[x] = OC(nelx,nely,x,volfrac,dc,passive);
%29 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(1e6);
end
%40 %%%%%%%%% 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-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
%65 %%%%%%%%% 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 LOADSAND 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)];
```

6. Conclusiones

En esta actividad se observó el comportamiento estático de un panorámico y al cambiar ciertas partes del código, esta muestra ahora una representación del ya mencionado panorámico, con esto se analiza más a fondo dicho código y como con solo cambiar algunos valores o datos puede crear algo nuevo. También se pude volver a apreciar la versatilidad de utilizar este código, iniciamos viendo una viga, después el marco de una bicicleta y ahora vemos la estructura de un panorámico, el poder hacer simulaciones nos permite evaluar los materiales y formas que utilizaríamos físicamente, pero con la ventaja de que no gastamos tantos recursos para ello.

A través de este ejercicio se aprendio sobre la durabilidad y resistencia de lo panorámicos en cuanto a su ubicación, material de construcción y otros factores perturbadores. Utilizando el software Matlab se simularon las propiedades mecánicas de la pieza, a través de un análisis de elemento finito. Este tipo de simulaciones se nos permiten evaluar los materiales y formas que en se usan físicamente, pero gracias a este tipo de simulaciones se tiene la ventaja de que no se gastan muchos recursos en ello[1].

Referencias

[1] O. Sigmund. 99 line topology optimization code. *Technical University of Denmark, DK-2800 Lyngby, Denmark*, 2017.