Universidad Autónoma de Nuevo León

Facultad de Ingeniería Mecánica y Eléctrica

Laboratorio de Biomecánica

Práctica 3

Diseño de la estructura de un panorámico

1904701	Bernardo Canul Aguilar	IMTC
1904820	Sylaid Pérez Oviedo	IMTC
1910351	Daniel Tudón González	IMTC
1991843	Javier Rangel Elizondo	IMTC
1992120	Francisco Adrián Castillo Herrera	IMTC

Hora: N5

Brigada: 509

Fecha: 27 octubre 2022

Ciudad Universitaria, San Nicolás de la Garza, N.L

Práctica #3: Diseño de la estructura de un panorámico

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).

Marco Teórico

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 con el fin de promocionar un producto o servicio.

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 un huracán.

Tanto la cimentación como el pedestal y la mampara elevada pueden constar de diversos elementos tales como: anclas suelo-zapata, vigas estabilizadoras, anclas pedestal-zapara, lastres, placas-base, acartelamientos, tubo del pedestal, entre otras.

Estado del arte

Título del documento	“Diseño de la estructura de un panoramico”
Fuente Bibliográfica	
Objetivo	El código de 99 líneas de optimización nos va a ayudar a mejorar las caracteristicas de un panorámico

Contenido	Se presenta el código de 99 líneas ejecutado en Matlab, adecuado a poder hacer un análisis de formas, y mejorar especificaciones; presenta código principal, optimizador y código de elementos finitos.
Palabras Clave	Estructura de un panorámico, Matlab, análisis de formas, optimización.
Conclusión	Este documento da una propuesta con el ejemplo de optimización, se utiliza el código de 99 líneas de Matlab.

Procedimiento de la programación

Para el desarrollo del código, se utilizó el mismo de la práctica #1 para darle unos cambios y hacer el caso del panorámico.

El código original es el siguiente:

```
%%% A 99 LINE TOPOLOGY OPTIMIZATION CODE BY OLESIGMUND, OCTOBER 1999 %%%
function plabbiom(nelx,nely,volfrac,penal,rmin)
% INITIALIZE
x(1:nely,1:nelx) = volfrac;
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;
    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) = -penal*x(ely,elx)^(penal-1)*Ue *KE*Ue;
    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);
  % 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)
  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)))));
    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
  %%%%%%%%%% 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),1); U =sparse(2*(nely+1)*(nelx+1),1);
  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(2,1) = -1;
  fixeddofs = union((1:2:2*(nely+1)),(2*(nelx+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 = 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)];
```

-*Cambios fuerzas múltiples*

Se cambió el script para ingresar las fuerzas que son necesarias ya que se observa que son 5, se cambia el anclaje del espacio de diseño a otra posición usando la instrucción de *fixeddofs*.

-Cambios Empotramiento diagonal

Para poder crear el empotramiento diagonal o el espacio blanco en la parte inferior derecha, se necesita modificar el código para recrear el espacio conocido.

El código final con las modificaciones es el siguiente:

```
%Práctica 3
%Eq5
%Biomecánica 508
%Sem Agosto-Diciembre 2022

function practi3(nelx,nely,volfrac,penal,rmin)
% INITIALIZE
x(1:nely,1:nelx) = volfrac;
loop = 0;
% DECLARACIÓN 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);
%oBJECTIVE FUNCTION AND SENSIBILYTI 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)=-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-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

%%%%%%%%%% 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 = sparse(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
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)];
```

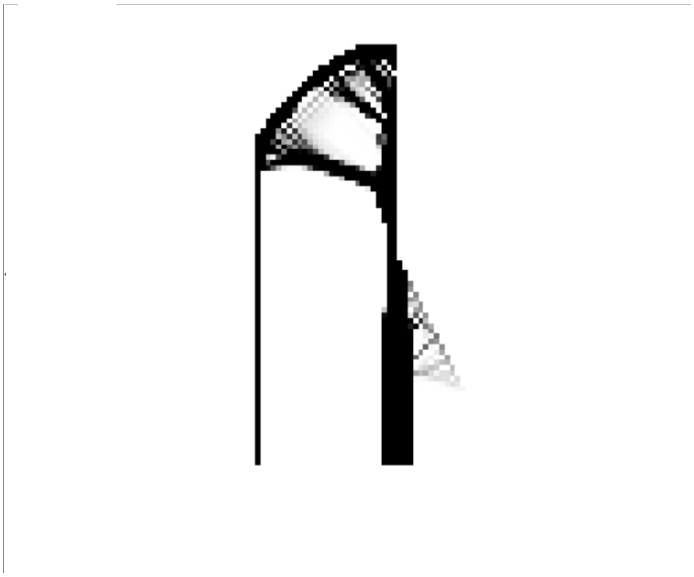
Implementación del programa

Se le dieron un valor a practi3 de (40,80,0.2,3.0,0.5) en la ventana de comando de Matlab, como se observa:

Command Window

```
>> practi3(40,80,0.2,0.5)
```

Se observó la siguiente figura como resultado final:



Conclusiones

-Sylaid Pérez Oviedo

Gracias a esta práctica conocí las fuerzas y resistencias que poseen los panorámicos, a partir de que se piensa en donde se colocará, los materiales de la composición, entre otras cosas que intervienen; en el programa de Matlab se hizo la simulación del manejo mecánico de elementos en el que observamos la investigación mediante componente limitado.

-Francisco Adrián Castillo Herrera

En el presente documento se hizo un análisis estructural de un panorámico. En base a los parámetros ajustados se obtuvo un elemento similar a un poste con soportes. Se observa que el resultado presenta zonas difuminadas, las cuales presentan un área de oportunidad relacionada con las iteraciones (posiblemente) y el factor de penalización, dado a que este último debería corregir esas zonas de densidad intermedia y patrones “checker board”.

-Daniel Tudón Gonzalez

Con esta práctica de la estructura de un panorámico, se puede observar que la optimización que se generó, se obtiene un beneficio de utilización de menor masa pero manteniendo el objetivo de poder soportar la carga del panorámico, la cual se idealiza para poder realizar la simulación en el software, que para efectos de esta práctica se usó MATLab. Además de que esta optimización se puede extrapolar para otras dimensiones de panorámicos que el cambio en las dimensiones puede cambiar debido a los reglamentos municipales, estatales o federales y sigue manteniendo su optimización sin afectar nada en el aspecto mecánico, por lo que

sigue manteniendo sus propiedades siempre y cuando el material que se utilice para el panorámico cumpla con un comportamiento lineal, elástico e isotrópico, para que se pueda considerar el material dentro de la optimización del código.

-Javier Rangel Elizondo

En resumen, en esta actividad puse volver a ver la versatilidad de utilizar este código, iniciamos viendo una viga, luego un marco de bici, y ahora la composición de un panorámico, el poder hacer esta clase de simulaciones, nos posibilita evaluar ciertos materiales y maneras que se usen físicamente, empero con virtud de que no se gastan muchos recursos para ello.

-Bernardo Canul Aguilar

En esta práctica hemos podido observar como el proceso de optimización es generalizable, no importa si se aplica a una viga, a un panorámico o incluso a un marco de bicicleta, el proceso es el mismo, y utilizando el código obtenido en la práctica #1, es sumamente sencillo, solo es cuestión de cambiar un par de parámetros, sin duda alguna esta práctica ha sido de gran utilidad para mi desarrollo como ingeniero. Además de esto se puede observar que la optimización ha dado como resultado que exista mayor masa en la base, puesto que ésta sostiene el peso completo del panorámico, mientras que en otras partes no están sometidos a una carga tan grande.