

Práctica 3 Diseño de la estructura de un panoramico

Uriel G. Ernesto A. Marcela O. Ana L. Diego O.

17 de octubre de 2022

Resumen

En esta práctica seguiremos trabajando con los códigos topológico y que sucede cuando lo optimizamos, así mismo se mostrará un ejemplo de cómo se vería un código después de la optimización.

1. Introducción

La idea de esta práctica es desarrollar en el estudiante la capacidad de análisis, implementación y solución del problema propuesto, mediante la optimización de códigos para la realización de análisis más detallados en la elaboración de proyectos.[1].

2. Instrucciones

El estudiante deberá presentar una propuesta de un análisis de formas y de la programación para la ejecución de la optimización (descripción funcional) de características de trabajo específicas que presenta la ventaja.

3. Nombre y definición de la forma Geometria

Para el desarrollo de la práctica de laboratorio. Con el fin de optimizar su rendimiento, usaremos una figura libre, como la práctica anterior que se optimizó un código muy similar, utilizando la lógica del código de Matlab para colocación de cargas, apoyos y fuerzas dentro de un espacio de diseño propuesto. Usaremos un letrero sobre dos bases.

4. 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. Existen diversos materiales que las agencias especializadas en la creación y diseño para estas estructuras. Usualmente los postes panorámicos están contruidos 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 acrílico. En cuanto al tamaño de la cartelera fija donde se colocará el anuncio, los tamaños y medidas son varios. Las estructuras pueden ser desde 4 x 6 metros, hasta 8 x 6 metros, ubicadas al ras del piso. Algunas medidas habituales de los espectaculares pueden ser entre 320x200cm, 400x300 cm, 800x300 cm y 1200x400 cm. El espacio de diseño a evaluar será de 2 dimensiones, las cargas y los apoyos de observan en la figura 3.1.

5. Propuesta de diseño de la geometría, alcance y limitaciones

El espacio de diseño que se va a evaluar en esta practica consta de dos dimensiones, en las cuales se observan a continuación las cargas y apoyos de dicho espacio:

Así mismo se muestra a continuación el espacio de diseño que se usara para esta práctica, en el cual se espera una fracción volumétrica aproximada de un 0.2

Para resolver esta practica se toma en consideración para la solución cinco cargas horizontales, además de que todos los apoyos tendrán restricciones en los ejes X y Y, por lo que el espacio de estados se puede representar de la siguiente manera:



Figura 1: Parte 1 del código

Por lo que para resolver esta práctica se hará uso de la herramienta de programación Matlab utilizando el mismo código que se uso en la practica anterior delimitando los siguientes parámetros: $P3(40,80,0.2,3,0.5)$. El código se muestra a continuación:

Hay limitaciones como son los materiales, tenemos que muchos de los materiales pueden variar por que desde que pueden ser estructuras de concreto hasta estructuras metalicas, por lo tanto simularlo tendra esta limitación.

6. Código de programación

```
P3.m x +
1  %%%% A 99 LINE TOPOLOGY OPTIMIZATION CODE BY OLESIGMUND, OCTOBER 1999 %%%
2  function P3(nelx,nely,volfrac,penal,rmin)
3  % INITIALIZE
4      x(1:nely,1:nelx) = volfrac;
5      loop = 0;
6      %DECLARANDO VACIO
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)>(1+nelx)*0.6666
10                 passive(ely,elx) = 1;
11             else
12                 passive(ely,elx) = 0;
13             end
14         end
15     end
16     x(find(passive))=0.001;
17     change = 1.;
18     % START ITERATION
19     while change > 0.01
20         loop = loop + 1;
21         xold = x;
22         % FE-ANALYSIS
23         [U]=FE(nelx,nely,x,penal);
24         % OBJECTIVE FUNCTION AND SENSITIVITY ANALYSIS
25         [KE] = lk;
26         c = 0.;
27         for ely = 1:nely
28             for elx = 1:nelx
```

Figura 2: Parte 1 del código

```

P3.m x +
28     for elx = 1:nelx
29         n1 = (nely+1)*(elx-1)+ely;
30         n2 = (nely+1)* elx +ely; %19
31         dc(ely,elx) = 0.;
32         for i =1:5
33             Ue = U([2*n1-1;2*n1; 2*n2-1;2*n2; 2*n2+1; 2*n2+2; ...
34                 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
40 % FILTERING OF SENSITIVITIES
41 [dc] = check(nelx,nely,rmin,x,dc);
42 % DESIGN UPDATE BY THE OPTIMALITY CRITERIA METHOD
43 [x] = OC(nelx,nely,x,volfrac,dc,passive);
44 % PRINT RESULTS
45 change = max(max(abs(x-xold)));
46 disp(['It.: ' sprintf('%4i',loop) 'Obj.: ' sprintf('%10.4f',c) ...
47     ' Vol.: ' sprintf('%6.3f',sum(sum(x))/(nelx*nely)) ...
48     ' ch.: ' sprintf('%6.3f',change )])
49 % PLOT DENSITIES
50 colormap(gray); imagesc(-x); axis equal; axis tight; axis off; pause(1e-6);
51 end
52 end
53 %%%%%%%%%% OPTIMALITY CRITERIA UPDATE %%%%%%%%%%
54 function [xnew]=OC(nelx,nely,x,volfrac,dc,passive)
55     l1 = 0; l2 = 100000; move = 0.2;

```

Figura 3: Parte 2 del codigo

```

P3.m x +
56 while (l2-l1 > 1e-4)
57     lmid = 0.5*(l2+l1);
58     xnew = max(0.001,max(x-move,min(1.,min(x+move,x.*sqrt(-dc./lmid)))));
59     xnew(find(passive)) = 0.001;
60     if sum(sum(xnew)) - volfrac*nex*nely > 0;
61         l1 = lmid;
62     else
63         l2 = lmid;
64     end
65 end
66 end
67 %%%%%%%%% MESH-INDEPENDENCY FILTER %%%%%%%%%
68 function [dcn]=check(nex,nely,rmin,x,dc)
69     dcn=zeros(nely,nex);
70     for i = 1:nex
71         for j = 1:nely
72             sum=0.0;
73             for k = max(i-round(rmin),1):min(i+round(rmin),nex)
74                 for l = max(j-round(rmin),1):min(j+round(rmin), nely)
75                     fac = rmin-sqrt((i-k)^2+(j-l)^2);
76                     sum = sum+max(0,fac);
77                     dcn(j,i) = dcn(j,i) + max(0,fac)*x(l,k)*dc(l,k);
78                 end
79             end
80             dcn(j,i) = dcn(j,i)/(x(j,i)*sum);
81         end
82     end
83 end

```

Figura 4: Parte 3 del código

```

P3.m x +
84 %%%%%%%%%%% FE-ANALYSIS %%%%%%%%%%%
85 function [U]=FE(nelx,nely,x,penal)
86     [KE] = lk;
87     K = sparse(2*(nelx+1)*(nely+1), 2*(nelx+1)*(nely+1));
88     F = sparse(2*(nely+1)*(nelx+1),5); U = sparse(2*(nely+1)*(nelx+1),5);
89     for ely = 1:nely
90         for elx = 1:nelx
91             n1 = (nely+1)*(elx-1)+ely;
92             n2 = (nely+1)* elx +ely;
93             edof = [2*n1-1; 2*n1; 2*n2-1; 2*n2; 2*n2+1;2*n2+2;2*n1+1; 2*n1+2];
94             K(edof,edof) = K(edof,edof) + x(ely,elx)^penal*KE;
95         end
96     end
97     % DEFINE LOADSAND SUPPORTS(HALF MBB-BEAM)
98     F(2*nelx*(nely+1)+2,1) = 1;
99     F(2*nelx*(nely+1)+(nely/4),2) = 1;
100    F(2*nelx*(nely+1)+(nely/2),3) = 1;
101    F(2*nelx*(nely+1)+(nely),4) = 1;
102    F(2*nelx*(nely+1)+(nely*1.2),5) = 1;
103    fixeddofs = 2*(nely+1):2*(nely+1):2*(nelx+1)*(nely+1);
104    alldofs = [1:2*(nely+1)*(nelx+1)];
105    freedofs = setdiff(alldofs,fixeddofs);
106    % SOLVING 127
107    U(freedofs,:) = K(freedofs,freedofs) \F(freedofs,:);
108    U(fixeddofs,:)= 0;
109    end

```

Figura 5: Parte 4 del código

```

110 %%%%%%%%%%% ELEMENT STIFFNESS MATRIX %%%%%%%%%%%
111 function [KE]=lk
112     E = 1.;
113     nu = 0.3;
114     k=[ 1/2-nu/6 1/8+nu/8 -1/4-nu/12 -1/8+3*nu/8 ...
115       -1/4+nu/12 -1/8-nu/8 nu/6 1/8-3*nu/8];
116     KE = E/(1-nu^2)*[ k(1) k(2) k(3) k(4) k(5) k(6) k(7) k(8)
117       k(2) k(1) k(8) k(7) k(6) k(5) k(4) k(3)
118       k(3) k(8) k(1) k(6) k(7) k(4) k(5) k(2)
119       k(4) k(7) k(6) k(1) k(8) k(3) k(2) k(5)
120       k(5) k(6) k(7) k(8) k(1) k(2) k(3) k(4)
121       k(6) k(5) k(4) k(3) k(2) k(1) k(8) k(7)
122       k(7) k(4) k(5) k(2) k(3) k(8) k(1) k(6)
123       k(8) k(3) k(2) k(5) k(4) k(7) k(6) k(1)];
124     end

```

Figura 6: Parte 5 del código

7. Resultado de la optimización

Para la primera prueba se utilizarán los valores $P1(30,10,0.5,3,1.5)$, los resultados fueron los siguientes.

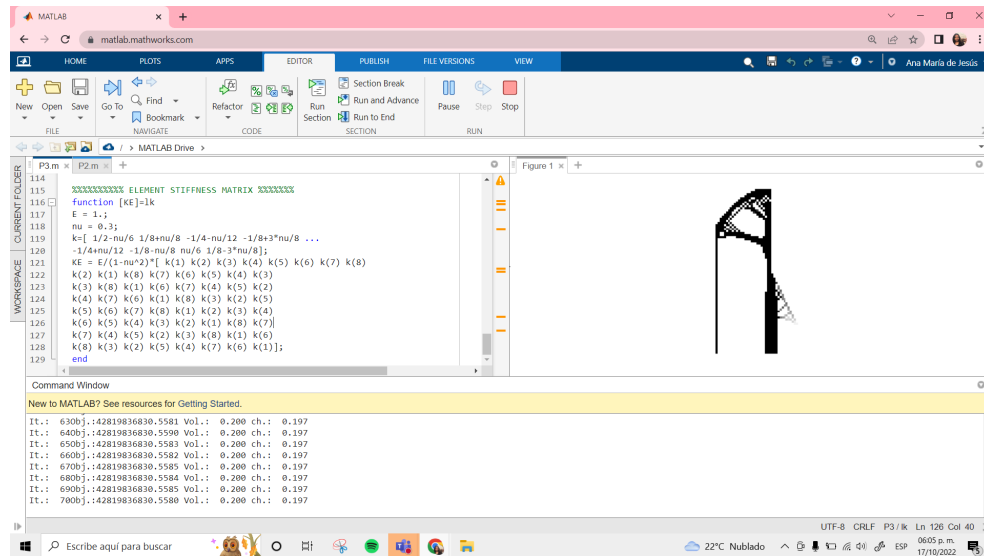


Figura 7: Resultados

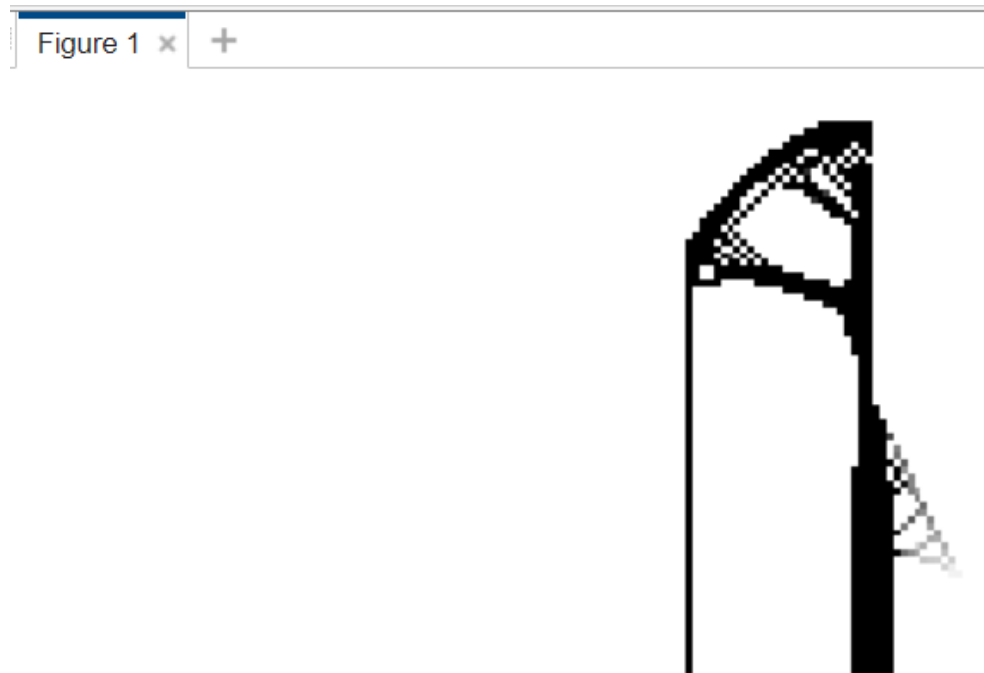


Figura 8: Resultado de la optimización

8. Conclusiones

■ Diego O.

Al termino de esta practica se puede concluir acerca de la importancia que tienen los diferentes lenguajes de programación, en este caso Matlab, que a la hora de generarles un código para realizar una actividad en específico y así poder simplificar procedimientos matemáticos que nos llevaría tiempo resolverlos, tal es el caso de esta practica en la cual nos ahorramos el realizar diferentes procedimientos algebraicos o matemáticos correspondientes al modelado y simulación, por lo que al correr el código podemos concluir que se cumplió el objetivo de la practica el cual era desarrollar la capacidad de análisis, implementación y solución del problema propuesto.

■ Ana L.

Para esta practica se utilizó el código que ya se venía utilizando desde la primera practica, la diferencia es que el proposito en esta practica fue simular un soporte para un panoramico, es decir, un anuncio publicitario, el cual comunmente esta sujeto a distintas fuerzas del medio, por lo que en esta practica se presento una solucion numerica computacional por medio del software de MATLAB para simular el desempeño mecanico de los componentes mecanicos. Gracias a esta practica conoci como representar esto en MATLAB por medio del codigo de 99 lineas y poco a poco entiendo las funciones y sus propositos dentro del codigo y su importancia para la programacion de elementos de indole biomecanico.

■ Uriel G.

Al termino de esta practica podemos analizar que para la creacion de panorámicos es muy importante tener presentes demasiados factores ya que es necesario conocer cada una de las propiedades de los materiales, ademas del lugar donde sera colocado ya que como conocemos existen areas donde cada factor climatico es diferente. La herramienta de matlab nos ayuda demasiado ya que nos muestra lo que puede ocurrir en la estructura del panoramico, tambien es necesario mencionar que el codigo nos genero un poco de dificultad al momento de realizarlo pero se logro generarlo y así entenderlo.

■ Marcela O.

En esta práctica aprendimos a realizar y definir la estructura de un panorámico, esta herramienta la vemos en nuestro día a día en los anuncios publicitarios. Esto nos ayuda a encontrar el sentido de la estructura de lo que vemos en la ciudad. Debemos tener en cuenta muchos puntos para que los cálculos, el material y la fabricación de la estructura panorámica sea segura. Utilizamos el software de MATLAB para el cuál programamos un código topológico, parecido al de la práctica anterior. Esto nos ayudó a reforzar nuestros conocimientos sobre el tema y a encontrar otra manera de aplicarlos.

■ Ernesto A.

Para esta práctica seguimos con el uso de la optimización de codigos para realizar figuras o proyectos, para ello se nos dejo elegir el objeto a realizar, a pesar de esto hay que tener muchos factores en cuenta para trabajar en la optimizacion, siendo aun asi algo que facilita muchas actividades, en esta practica fue un cartel publicitario, con esto se podra ir en un futuro desarrollar algun componente o proyecto respecto a la biomecanica, como son la protesis o demas, por medio de un código de MATLAB, el cuál colocaremos los datos para su simulacion y asi optener los resultados sin tener que realizarlo de una forma fisica.

Referencias

[1] Anonimo. ¿cuánto cuesta construir la estructura de un panorámico?