

Práctica 3: Diseño de la estructura de un panorámico

Brigada: 109

Emanuel Salas de León 1854873
Jason Méndez Muñoz 1889560
Jesús Mario Duarte Salinas 1907630
Lorena Alejandra Campos Carrazco 1909279
Yair Obed Morales Ortiz 1992266

17 de octubre de 2022

1. Marco Teórico

El proceso de diseño generativo es una forma de inteligencia de diseño que responde a un problema de ingeniería con una serie de métodos para elegir, que se pueden modificar según sea necesario, a diferencia de los métodos tradicionales, donde el proceso comienza con un modelo basado en conocimientos del ingeniero, el diseño generativo comienza unos parámetros de diseño y utiliza una IA para crear el modelo. Básicamente, le está diciendo al algoritmo de diseño generativo que no sabe cuál es la respuesta, pero sabe lo que se necesita.

Es como diseñar en sentido inverso: se establecen las diferentes características que debe cumplir el diseño y se procura llegar al mejor resultado posible, en lugar del método tradicional de probar prototipos con métodos operativos para ir ajustándose después. Es un proceso creativo que se puede utilizar para todo, desde problemas simples hasta soluciones complejas.

En el área de diseño, la principal ventaja del diseño generativo es que ofrece varias opciones de diseño que cumplen con los requisitos especificados, lo que aumenta en gran medida la eficiencia de todo el proceso.

2. Desarrollo

2.1. Nombre y definición de la forma (Geometría)

En esta práctica se pretende diseñar un panorámico publicitario con ayuda de alguna herramienta para el diseño generativo, un panorámico es un anuncio publicitario de gran formato, normalmente se utiliza para la publicidad gráfica de cualquier ámbito, son de diversos tamaños y se colocan en puntos estratégicos de una ciudad para su mayor visibilidad.

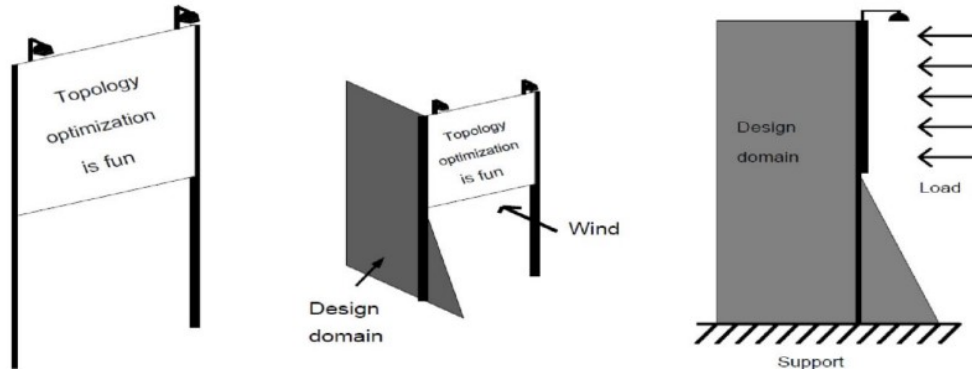


Figura 1: *Panóramico publicitario*

2.2. Estado del arte

El método de elementos finitos (FEM), o análisis de elementos finitos (FEA), se basa en la idea de construir un objeto complicado con formas simples o dividir un objeto complicado en piezas más pequeñas y manejables. La aplicación de esta idea simple se puede encontrar en todas partes en la vida cotidiana, así como en la ingeniería, en esta actividad se hace uso de el elemento finito en una viga. Porque la geometría de la pieza, sometida a cargas y restricciones, se subdivide en partes más pequeñas, conocidas como elementos que representan el dominio continuo del problema. La división de la geometría en pequeños elementos resuelve un problema complejo, al subdividirlo en problemas más simples.

Los elementos finitos están conectados entre sí por puntos, que se llaman nodos o puntos nodales. Al conjunto de todos estos ítems se lo denomina malla, que este proceso se hace de manera automática en Fusion 360. Debido a las subdivisiones de la geometría, las ecuaciones matemáticas que rigen el comportamiento físico, no se resolverán de una manera exacta, sino aproximada por este método numérico. La precisión de los Métodos dos Elementos Finitos depende de la cantidad de nodos y elementos, del tamaño y de los tipos de elementos de la malla. Por lo tanto, cuanto menor sea el tamaño y mayor el número de elementos en una malla, más precisos serán los resultados de las análisis.

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

El diseño se planea realizar por medio de Matlab teniendo como referencia el código de 99 líneas. Mediante la implementación de un código de optimización adecuado para la resolución de esta pieza, podremos observar el resultado de la geometría final y obtener un diseño con las mismas características y propiedades del diseño original, pero con un peso menor. Este método no cuenta con limitaciones, se pueden adecuar sinfín de diseños y obtener optimizaciones deseadas de acuerdo a las necesidades que tengamos. La optimización topológica es una técnica englobada dentro del campo de análisis estructural, basada en el análisis mecánico de un componente, sistema o estructura.

2.4. Pasos del desarrollo de la programación

1. Ejecutar Matlab

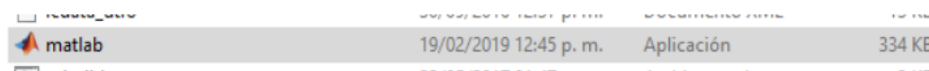


Figura 2: *Ejecución de Matlab*

2. Crear un nuevo script

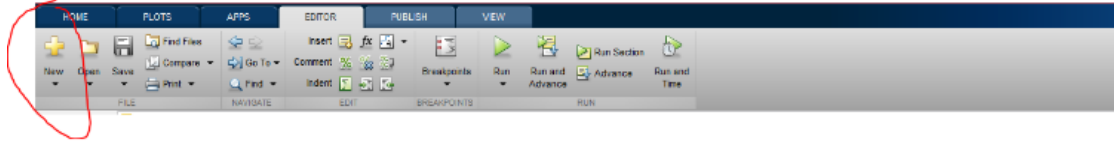


Figura 3: Script

3. Escribir nuestro código dentro del script

```

%%% A 99 LINE TOPOLOGY OPTIMIZATION CODE BY OLESIGMUND, JANUARY 2000%%%
%%%% CODE MODIFIED FOR INCREASED SPEED, September 2002, BY OLE SIGMUND %%%

function topp3(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 ((1+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

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)];

[x] = OC(nelx,nely,x,volfrac,dc,passive);
%29 PRINT RESULTS
change = max(max(abs(x-xold)));
disp('1c.1' sprintf('%4s',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
%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;
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
%45 %%%%%%%%%% 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*(nelx+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 LOADAND SUPPORTS (HALF MB-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;

```

Figura 4: Código

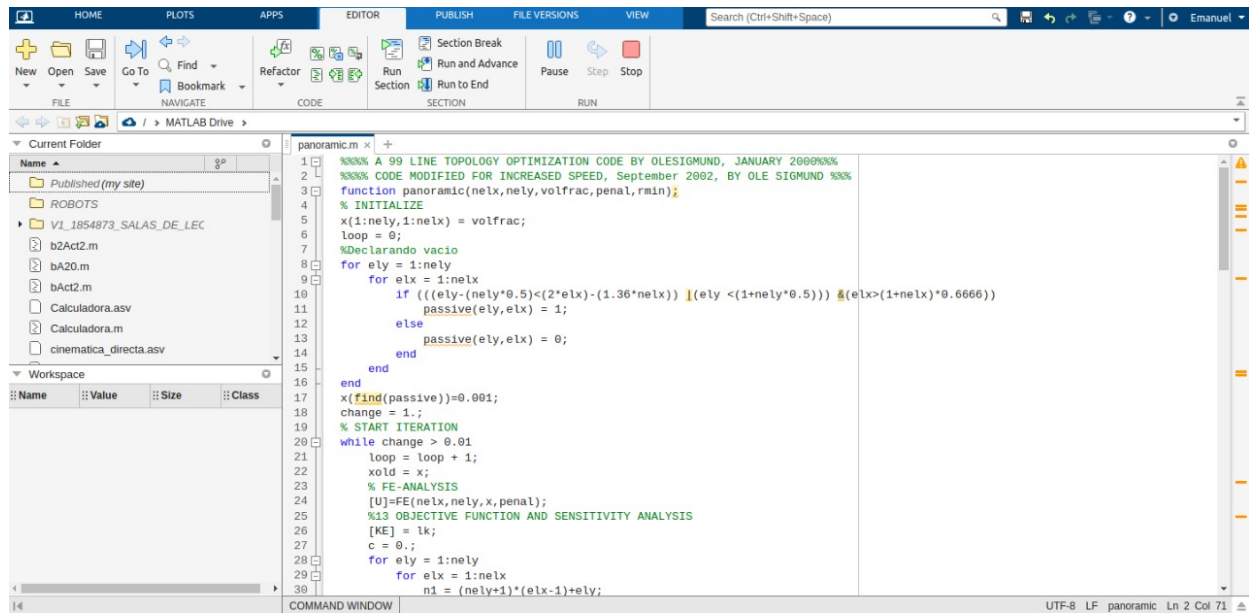


Figura 5: Código en Matlab

En este caso se utilizó el mismo código que la práctica pasada solo que cambiando ciertos parámetros que nos permitieron obtener los resultados obtenidos.

4. Guardar nuestro script con la extensión (.m) correspondiente a matlab y compilar

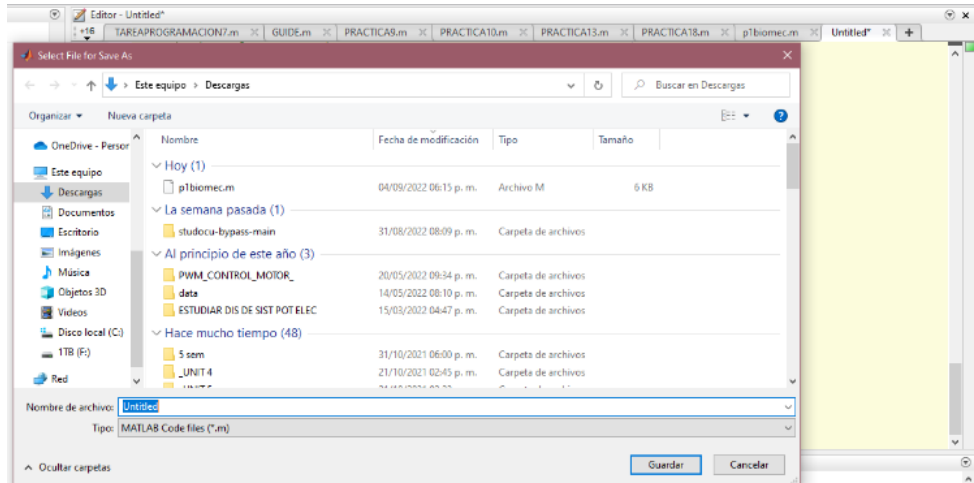


Figura 6: Guardar código

2.5. Resultados de la optimización

Después de haber corrido el código en Matlab se obtuvo la siguiente imagen que nos muestra la optimización del panorámico.

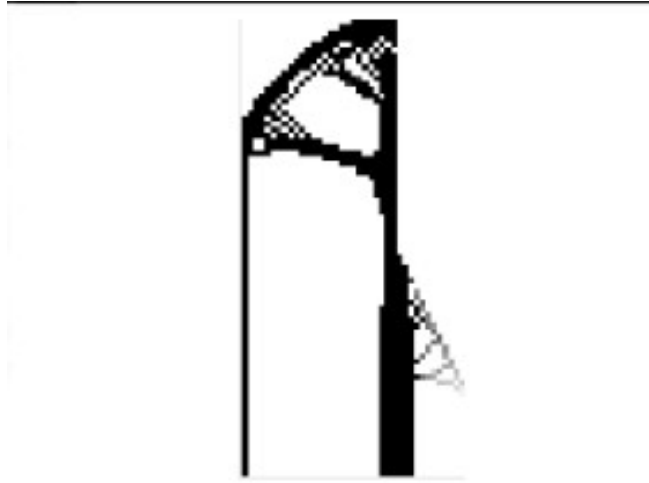


Figura 7: *Resultado de la optimización*

3. Conclusiones de cada autor

■ Emanuel Salas de Leon 1854873

Durante esta práctica se aplicó todo lo aprendido en las prácticas anteriores, para el diseño de un anuncio publicitario, específicamente para un cartel panorámico, este tipo de anuncio está comúnmente expuesto a fuerzas externas. Por esta razón que para el diseño de este se consideraron estas fuerzas a la hora de generar el diseño, de tal forma que al final se obtuvo una geometría particular para soportar dichas fuerzas.

■ Jason Méndez Muñoz 1889560

Esta practica a servido para seguir profundizando en los conocimientos adquiridos durante este laboratorio. Yendo mas allá de la teoría, de nuevo buscamos la solución a un problema real, analizando la estructura más optima para un cartel panorámico, usando la optimización topológica para buscar una estructura resistente a todas las fuerzas a las que este tipo de estructuras están normalmente expuestas.

■ Jesús Mario Duarte Salinas 1907630

Por medio de esta práctica se reafirmó lo que nos puede ofrecer la optimización topológica, en esta ocasión fue de un anuncio panorámico y volvimos a utilizar el código que hemos estado utilizado anteriormente. Además se tomaron en cuenta algunas fuerzas externas y se definió la forma geométrica de la pieza. Por medio de la simulación en Matlab obtuvimos un diseño similar a lo que buscábamos y solo se necesitó modificar algunas de las líneas del código antes mencionado.

- **Lorena Alejandra Campos Carrazco 1909279** Al realizar esta actividad se volvió a ver la optimización topológica, que bien sabemos es un método para poder optimizar el diseño de un elemento o estructura dada. En esta actividad se vio la optimización topológica de un anuncio panorámico, esto fue posible gracias al software de Matlab donde se escribió un código el cual nos permitió hacer la optimización. Como ya se sabe el método de optimización topológica se centra en encontrar la topología óptima de un producto o estructura. El objetivo es minimizar el peso del producto o estructura pero que siga cumpliendo los mismos criterios y especificaciones.

■ **Yair Obed Morales Ortiz 1992266**

Al realizar esta práctica puedo concluir que, fue una práctica un poco similar a las anteriores, utilizando Matlab, para la elaboración de esta práctica se presentó y se aplicó una propuesta de análisis de formas, con el fin de optimizar el diseño, para ello se tomó en cuenta lo que se vio en las prácticas anteriores donde se definió la forma geométrica de la pieza, el cual vendría siendo un panorámico, y posteriormente se declararon las variables a tomar en cuenta, como lo es el espacio del diseño y las fuerzas aplicadas. Ejecutando el código de programa en el software se pueden observar los resultados obtenidos en la figura.

Referencias

- [1] Menéndez, L. G., Somonte, M. D., and Escudero, M. d. M. E. (2020). Diseño generativo: el estado del arte. *Técnica Industrial*, 327:44–49.
- [2] Miles, Dan, C., Lodhia, B., and Jonnalagadda, S. (2022). ¿qué es el diseño generativo y cómo puede aprovecharse en la fabricación?
- [3] Velasco Pérez, R. (2015). *Estudio de la aplicación del diseño generativo al diseño conceptual arquitectónico*. PhD thesis, Universitat Politècnica de València.