

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN
FACULTAD DE INGENIERÍA MECÁNICA Y ELÉCTRICA

Laboratorio de Biomecánica

Práctica 3

DOCENTE: YADIRA MORENO VERA
MARTES N3 BRIGADA 202

Equipo #5

Matricula	Nombre
1859521	Abraham Guerra Carmona
1853651	Pablo Aarón Cruz Rentería
1811170	Javier Humberto Guia Martínez
1866860	Alan Saim Lozano Espinosa
1865707	Edgar Jair De La Cruz Mendez

Objetivo

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

Marco Teórico

El análisis de formas es la disciplina que se ocupa del procesamiento de las figuras geométricas para posibilitar su caracterización. Está relacionado con el análisis estadístico de formas, y su propósito es posibilitar la determinación de la coincidencia entre formas y facilitar su reconocimiento. Se aplica exclusivamente a la geometría de un objeto, pero no a su análisis estructural.

El análisis de formas permite la clasificación de formas geométricas, por ejemplo, usando un ordenador para detectar objetos de forma parecida a partir de una base de datos o partes que encajan entre sí. Para que una máquina analice y procese automáticamente formas geométricas, los objetos deben representarse en forma digital. Más comúnmente, se utilizan representaciones de contorno para describir el objeto por sus límites. Sin embargo, otras representaciones basadas en el volumen o representaciones basadas en puntos se pueden usar para representar las formas.

Una vez recibidas las representaciones de los objetos en forma numérica, ya sea mediante modelado, escaneado o extrayendo la forma de imágenes 2D o 3D, se deben simplificar antes de poder ser comparadas. La representación simplificada a menudo se denomina descriptor de forma. Estas representaciones simplificadas pretenden almacenar la mayor parte de la información relevante, a la vez que son más fáciles de manejar, almacenar y comparar que las propias formas directamente.

En el pasado, muchos de los procesos de diseño fueron realizados por la experiencia e intuición del diseñador en vez de una aplicación intensiva de la teoría de optimización. Recientemente esta forma de pensar ha cambiado debido a la importancia que ha tomado el campo de la optimización estructural en el diseño, ya que mediante su aplicación se logra reducir costos, materiales y tiempo en los procesos de diseño realizados por los ingenieros.

El propósito de aplicar los conceptos de diseño óptimo a la ingeniería estructural es el de obtener una solución a un problema de ingeniería que cumpla con todas las limitaciones y restricciones impuestas, y que a la vez resulte ser la mejor en cuanto a uno o varios criterios de diseño previamente establecidos.

El incremento dramático en la capacidad y velocidad de proceso de los computadores modernos en las últimas décadas y los avances realizados en el desarrollo de nuevos métodos numéricos para el análisis de medios continuos en conjunción con los algoritmos tradicionales de optimización, tales como programación cuadrática secuencial, programación no lineal han hecho posible automatizar el diseño y el proceso de optimización de formas.

En este tipo de diseños las técnicas de optimización numérica son usadas en un intento de obtener la mejor forma y, adicionalmente, el uso más efectivo de los materiales para una determinada estructura bajo las restricciones impuestas por las condiciones de diseño.

Programación y geometría

El éxito de cualquier metodología de optimización descansa sobre su habilidad para tratar problemas complejos, como es el caso del diseño de optimización de formas. Resolver eficientemente problemas de tipo no lineal es desafiante. Por ello, se ha hecho práctica común que los métodos sean modificados, combinados y extendidos a fin de construir algoritmos que combinen las mejores características de varios de ellos en una propuesta que de cierta manera se adapte al problema en particular y permita resolver el problema en un contexto general que de otra manera cada método por separado no podría resolver por sí solo.

Por estas razones, la metodología de optimización de formas presentada se basa en el fundamento de varios métodos su combinación y la escogencia más apropiada de ellos. En este trabajo, entre las numerosas técnicas evolucionarias existentes, se han seleccionados los Algoritmos Genéticos debido al hecho de que teórica y experimentalmente se ha demostrado que proveen una búsqueda robusta en espacios complejos de diseño. Estos algoritmos tienen muchas ventajas sobre los métodos tradicionales de búsqueda. Entre otras consideraciones, no necesitan de información especializada de la función a optimizar si no los valores de la función objetivo o de mérito.

Los algoritmos genéticos son algoritmos de búsqueda basados en los mecanismos de la selección natural y la genética, fueron desarrollados en un intento de simular algunos de los procesos observados en la evolución natural, como la selección de parejas, los procesos de recombinación de información y mutación, con el fin de generar mecanismos de búsqueda que convergen rápidamente a la estructura óptima o casi-óptima del problema con un esfuerzo mínimo y probando solo una pequeña fracción del espacio de diseño.

Recientemente, los Algoritmos Evolutivos (AE) se han usado extensivamente para encontrar soluciones casi-óptimas u óptimas al problema de optimización estructural de formas debido a su robustez, eficiencia y eficacia. Entre sus ventajas se encuentra el hecho de no requerir información especializada para obtener el óptimo buscado además de no necesitar de un análisis de sensibilidades durante este proceso. Todas estas ventajas hacen a AE un buen candidato para resolver problemas de optimización en general, sin embargo, algunas de sus desventajas, tales como la carencia de diversidad en la población durante el proceso de evolución y su excesivo tiempo de cálculo debido al gran número de funciones de evaluación, han llamado la atención de la comunidad especializada a fin de solventar estos inconvenientes. En este sentido, el objetivo de este artículo es proponer una metodología de optimización basada en un Algoritmo Genético Distribuido de carácter flexible que permita la optimización estructural de la forma de modelos de elementos finitos o elementos de contorno modelados con elementos de diseño geométrico. El carácter distribuido y el manejo flexible de las variables y parámetros permiten al algoritmo solventar los inconvenientes antes mencionados. Finalmente, un ejemplo numérico es presentado y discutido a fin de mostrar la habilidad de la metodología propuesta para optimizar este tipo de problemas.

DESARROLLO DE LA PROGRACIÓN

Para esta práctica lo primero que debemos hacer es editar nuestro código que fue antes proporcionado y para así ingresar los datos de las fuerzas que nos pide la práctica para lo cual son 5 fuerzas y que se requieren en esta práctica, para cambiar el anclaje del espacio de diseño a otra posición se tiene que cambiar la línea de código "fixeddofs". Para poder hacer el empotramiento diagonal, Consultamos el código de 90 líneas en el cual encontramos una sección donde se explican los elementos pasivos los cuales sirven de ayuda para determinar un espacio en blanco, el cual se ocupó para hacer el empotramiento de la parte inferior en nuestra pieza.

El código implementado para la simulación del programa el siguiente:

```
function topp1(nelx,nely,volfrac,penal,rmin)
% INITIALIZE
x(1:nely,1:nelx) = volfrac;
loop = 0;
```

```

for ely = 1:nely
for elx = 1:nelx
if ((elx)^2+(ely-nely)^2) < (0.65*nelx)^2
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 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,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)/l2 > 1e-4)
lmid = 0.5*(l2+l1);
xnew(find(passive))=0.001
xnew = max(0.001,max(x-move,min(1.,min(x+move,x.*sqrt(-dc./lmid)))));

```

```

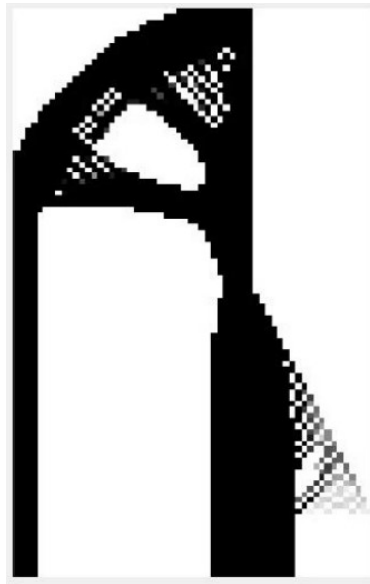
if sum(sum(xnew)) - volfrac*nex*nely > 0;
l1 = lmid;
else
l2 = lmid;
end
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% MESH-INDEPENDENCY FILTER %%%%%%%%%%
function [dcn]=check(nex,nely,rmin,x,dc)
dcn=zeros(nely,nex);
for i = 1:nex
for j = 1:nely
sum=0.0;
for k = max(i-round(rmin),1):min(i+round(rmin),nex)
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(nex,nely,x,penal)
[KE] = lk;
K = sparse(2*(nex+1)*(nely+1), 2*(nex+1)*(nely+1));
F = sparse(2*(nely+1)*(nex+1),1); U =sparse(2*(nely+1)*(nex+1),1);
for ely = 1:nely
for elx = 1:nex
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 =2*nex*(nely+1)+1:2*(nex+1)*(nely + 1);
alldofs = [1:2*(nely+1)*(nex+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)];

```



RESULTADO OBTENIDO

CONCLUSIONES

Abraham Guerra Carmona 1859521

En la realización de esta práctica se pudo observar el algoritmo que se insertó en Matlab para crear un diseño optimizado o generativo de una estructura, para lo cual es un tema bastante interesante ya que gracias a estos algoritmos se pueden crear estas estructuras que además de lucir bien físicamente son totalmente funcionales al momento de darles su aplicación ya que tienen en su estructura lo necesario para poder funcionar y aguantar las cargas para las que se hayan diseñado, todo esto también se aplica en programas de CAD en los cuales te da la opción para optimizar tu pieza diseñada, en estos tiempos con la tecnología de impresión 3d es mucho más fácil volver realidad todas aquellas estructuras optimizadas.

Pablo Aarón Cruz Rentería 1853651

Tenemos entonces que para este trabajo se seleccionaron los algoritmos genéticos ya que fueron desarrollados con el fin de generar mecanismos de búsqueda que nos lleven rápidamente a una estructura óptima el cual es el objetivo de esta práctica generar a través del código o algoritmo que nuestra pieza, aunque esté sujeta a esfuerzos en este caso 5 por que son 5 fuerzas, optimizarla y llevarla al esfuerzo mínimo.

Javier Humberto Guia Martínez 1811170

En la práctica se insertó el algoritmo en Matlab para crear un diseño optimizado o generativo de una estructura, estos pueden crear estas estructuras funcionales enfocadas a la aplicación ya que poseen lo necesario para funcionar y resistir cargas. Esto al igual se pueden aplicar en programas de CAD o semejantes en los cuales se optimiza y se obtienen más a detalle propiedades de las estructuras, las cuales pueden hacerse físicamente mediante su impresión 3D con distintos tipos de materiales.

Alan Saim Lozano Espinosa 1866860

En la presente práctica, vimos la solución de una pieza utilizando la optimización de la pieza, con esto, aprendimos que aparte de las herramientas CAD que pueden ser usadas, también se hace uso de estos softwares que calculan la mejor opción para ser fabricada una pieza en base a condiciones optimas de desarrollo que la hagan

más ligera y resistente ante cargas externas. Estos algoritmos crean dichas estructuras logrando que las formaciones cumplan una necesidad específica.

BIBLIOGRAFÍA

99 Line Topology Optimization Code – O. Sigmund, Department of Solid Mechanics, Building 404, Technical University of Denmark, DK-2800 Lyngby, Denmark.

IMME. (2007). *Una metodología para la optimización estructural de formas usando principios de evolución flexible distribuida...*
http://ve.scielo.org/scielo.php?script=sci_arttext&pid=S0376-723X2007000100002