UNIVERSIDAD AUTONOMA DE NUEVO LEÓN

FACULTAD DE INGENIERIA MECÁNICA Y ELÉCTRICA

LAB. BIOMECANICA

PRACTICA #1

EQUIPO 5

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

Grupo: 202 Hora: N3

Maestro: Dra. Yadira Moreno Vera

Objetivo

El estudiante conocerá cada una de las secciones que integran el código de optimización topológica, como se debe de crea el archivo (.m) en MATLAB y como se ejecuta el análisis.

Marco teórico

Un problema clásico de la ingeniería consiste en determinar la configuración geométrica óptima de un cuerpo que minimice o maximice una cierta función objetivo, al mismo tiempo que satisface las restricciones o condiciones de contorno del problema. La solución de este problema puede ser planteada utilizando dos estrategias: como un problema de optimización de forma o de optimización de la topología. La optimización de forma consiste en modificar la geometría del dominio preservando su topología, es decir sin crear huecos o cavidades en su interior.

Este tipo de análisis es usualmente conocido como análisis de sensibilidad al cambio de forma y sus bases matemáticas se encuentran bien establecidas. El principal inconveniente del análisis de sensibilidad al cambio de forma es que sólo permite cambios en la frontera del dominio, lo que limita su campo de aplicación. Una manera más general de controlar un dominio es mediante modificaciones de su topología, lo que permite obtener la configuración deseada partiendo de una morfología inicial distante de la óptima. Los métodos de homogenización son posiblemente los más utilizados para la optimización topológica. Estos consisten en caracterizar la topología a través de su densidad, es decir, los huecos se identifican con regiones de densidad nula.

De esta forma la solución del programa resulta en una distribución ficticia de material. Matlab es un software matemático que ofrece un entorno de desarrollo integrado (IDE)con un lenguaje de programación propio (lenguaje M). Entre sus prestaciones básicas se hallan: la manipulación de matrices, la representación de datos y funciones, la implementación de algoritmos, la creación de interfaces de usuario (GUI) y la comunicación con programas en otros lenguajes y con otros dispositivos hardware.

La optimización topológica es una técnica englobada dentro del campo de análisis estructural. Se basa en el análisis mecánico de un componente o estructura. Su principal objetivo es el aligeramiento estructural manteniendo las funcionalidades mecánicas del componente objetivo.

A diferencia de otros tipos de optimización, la optimización topológica ofrece un nuevo concepto de diseño estructural enfocado a aquellas aplicaciones donde el peso del componente es crucial (por ejemplo, la industria aeroespacial). Gracias a los nuevos métodos computacionales, es posible llevar la optimización a un nivel más complejo de análisis a nivel estático, dinámico, plástico, modal o de impacto, entre otros, los cuales pueden considerarse durante el proceso de optimización.

El desarrollo de esta metodología tiene un amplio campo de aplicación para las tecnologías de fabricación aditiva, como por ejemplo la fabricación SLM (Selective Laser Melting), debido a las grandes posibilidades en términos de diseño (geometrías muy complejas).

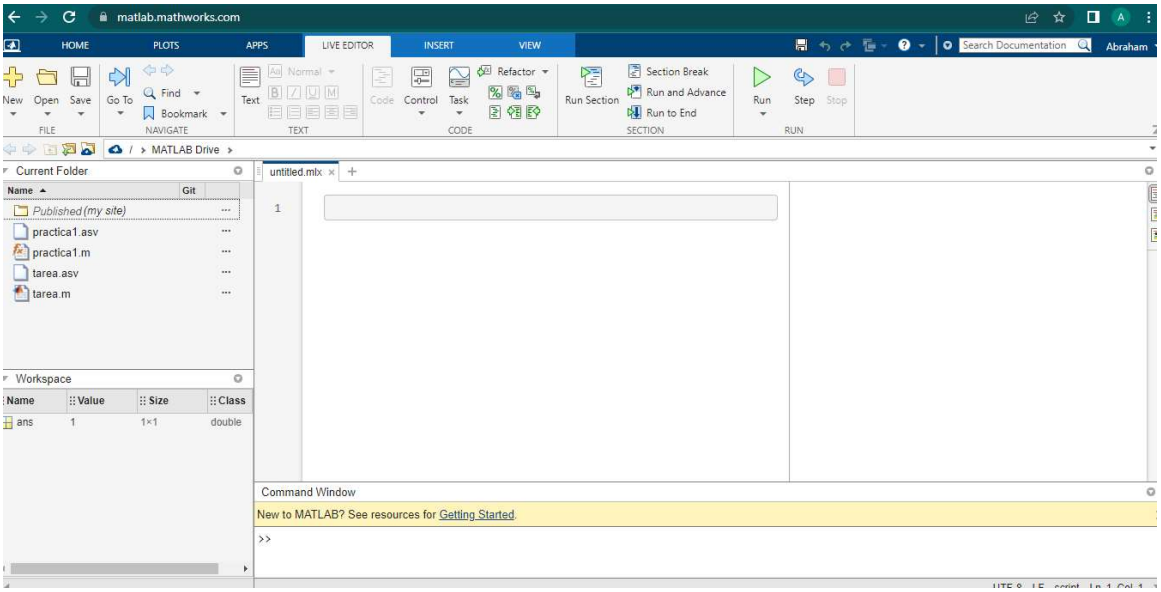
Estado del arte

En el ámbito del desarrollo de software, el estado del arte (state of the art) típicamente ha estado vinculado con la evolución de los lenguajes de computación. Uno de los primeros grandes saltos en esa dirección fue el desarrollo del primer lenguaje de alto nivel, FORTRAN.

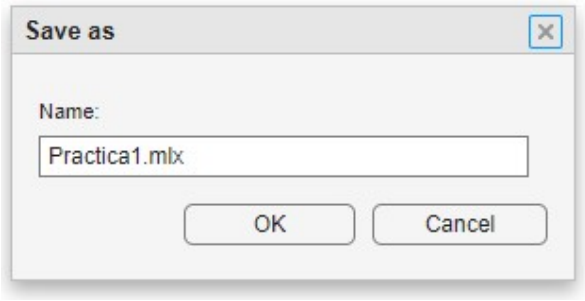
En sus inicios, el desarrollo de software en ese lenguaje era considerado “programación automática” porque requería menos conocimientos técnicos que los lenguajes ensambladores. Los detractores (gurús que programaban en lenguaje ensamblador) solían justificar su rechazo mostrando que los programas en FORTRAN eran significativamente más ineficientes comparados con los que ellos escribían. Hoy no solo tenemos lenguajes de programación de alto nivel, de “cuarta” y de “quinta generación”, funcionales u orientados a objetos; también lenguajes de documentación (Latex, HTML, etc.), de especificación (bison, flex, etc.), y otros.

Programación en Matlab

1) Abrimos el programa desde Matlab online



2) Asignamos un nombre al programa.



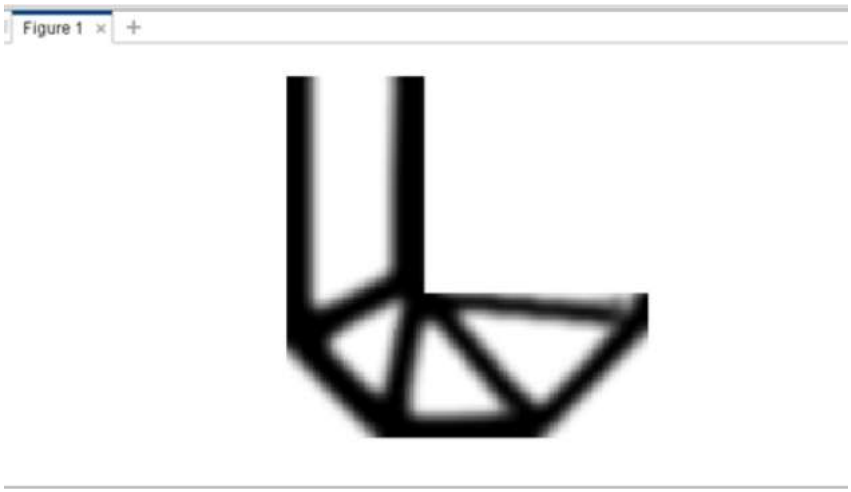
3) Agregamos el programa a Matlab de 99 líneas proporcionado.

```
function top(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-floor(rmin),1):min(i+floor(rmin),nelx)
      for l = max(j-floor(rmin),1):min(j+floor(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 = zeros(2*(nely+1)*(nelx+1),1);
for elx = 1:nelx
  for ely = 1:nely
    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 LOADS AND 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)];
```

4) Agregamos como parámetros iniciales top (60, 20, 0.5, 3.0, 1.5), obtuvimos el siguiente resultado.



Conclusiones

Abraham Guerra Carmona 1859521

En esta práctica se pudo ver un ejemplo y aprender más acerca de qué son y para qué sirven los algoritmos de optimización topológica, así también se vio un ejemplo de una estructura un código de Matlab de solo 99 líneas proporcionado por el libro, el cual se encargaba de optimizar una estructura para ciertas cargas definidas. La optimización topológica es importante en el área de la biomecánica, ya que siempre se busca prótesis o mecanismos lo más cómodos, resistentes y baratos para que estos puedan tener un buen desempeño, pero sin pesar demasiado, ya que puede complicarle al usuario el uso de dicha prótesis, este es un claro ejemplo en donde la optimización topológica tiene un gran campo pues se pueden analizar las estructuras a emplear en las prótesis para que éstas sean lo más ligeras posibles pero sin sacrificar su resistencia.

Javier Humberto Guia Martínez 1811170

En conclusión, acerca de la práctica se dio a conocer acerca de qué son y cuál es la utilidad de los algoritmos de optimización topológica, al igual que por medio de un ejemplo se mostró un código del software MATLAB, el cual su función era optimizar la estructura para cargas definidas. También se explico acerca de la optimización topológica y su importancia en la biomecánica, para él la innovación de prótesis tanto como su desempeño. Gracias a métodos computacionales innovadores, es posible llevar la optimización a un nivel más complejo de análisis a nivel estático, dinámico, plástico, modal o de impacto en los procesos de optimización.

Edgar Jair De La Cruz Méndez 1865707

La práctica tuvo complicaciones en un principio ya que se usó un software con el que tenía tiempo de no trabajar, sin embargo, el hecho de que ya lo conoció me ayudó a recordar poco a poco los comandos, desde la forma en poner los códigos hasta el cómo implementar en un caso de la vida real. En términos generales ví la práctica un poco introductoria al semestre y como repaso de lo ya conocido en materias anteriores.

Pablo Aaron Cruz Rentería 1853651

En esta práctica se vio por qué es importante controlar un dominio el cual como se mencionó en el reporte es mediante modificaciones de su topología, lo que nos hará obtener la configuración deseada partiendo de una morfología inicial distante de la óptima, después se realizó un ejercicio el cual se tuvo que desarrollar en Matlab en lo que a mi parecer esto nos enseñó a la vez 2 cosas la cuales son el uso de dicho software y ver la estructura de un código el cual como fin era optimizar la estructura para ciertas cargas definidas.

Alan Saim Lozano Espinosa 1866860

En la práctica correspondiente aprendimos el concepto y retomamos la importancia del estado del arte, así mismo se utilizó el software MATLAB para la realización del código como ejemplo de la optimización topológica, con el que la ida principalmente es el de optimizar una estructura en una carga definida, así como las aplicaciones que se tienen de este concepto. Me parece interesante todo lo que hemos podido lograr conocer al tener en cuenta la optimización de una estructura.

Bibliografías

W. Gibbs; “Software’s Chronic Crisis”, Scientific American, September 1994. http://bit.ly/sg30r1

Ramirez F, García H, López F & De la garza F. Optimización estructural del codo de una prensa mecánica mediante análisis topológico. Universidad Autónoma de Nuevo León. Facultad de Ingeniería Mecánica y Eléctrica. (pp 3-5).

https://mecheng.iisc.ac.in/suresh/me256/99linecodePaper.pd