

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN
FACULTAD DE INGENIERÍA MECÁNICA Y ELÉCTRICA

Laboratorio de Biomecánica

Práctica 1

Ing. Yadira Moreno Vera

Equipo 1

IMC	Raúl Emiliano Soto Salas	1864359
IMC	Melissa Alejandra Jasso Maciel	1897337
IMC	Carlos Antonio Caballero Padilla	1900864
IMC	Omar Gerardo Ríos Gaytán	1902388
IMC	Edgar Alan Carrizales Treviño	1904406
IMC	Néstor Eliud Cano García	1909644

Clase Martes V1

Brigada: 204

Semestre: Agosto - Diciembre 2022

Ciudad Universitaria, San Nicolás de los Garza

6 Septiembre 2022

1) Programación a Realizar

” Desarrollar una herramienta computacional que realice la optimización topológica en estructuras de tres dimensiones usando elementos finitos tetraédricos de cuatro nudos, con un modelo de material lineal, elástico e isotrópico”. La herramienta computacional deberá ser capaz de generar la optimización topológica para estructuras o elementos tridimensionales, cumpliendo requisitos de acceso libre, escalabilidad y de fácil manejo.

2) Estado del Arte

La plataforma de MATLAB está optimizada para resolver problemas de ingeniería y científicos. El lenguaje de MATLAB, basado en matrices, es la forma más natural del mundo para expresar las matemáticas computacionales. Los gráficos integrados facilitan la visualización de los datos y la obtención de información a partir de ellos. Una vasta librería de toolboxes preinstaladas te permiten empezar a trabajar inmediatamente con algoritmos esenciales para tu dominio. El entorno de escritorio invita a experimentar, explorar y descubrir. Todas estas herramientas y prestaciones de MATLAB están probadas y diseñadas rigurosamente para trabajar juntas.

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 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).

La optimización topológica de una pieza es un proceso relacionado con el cálculo estructural en el cual se realiza un análisis mecánico del componente en cuestión que desemboca en el aligeramiento masivo de su estructura y material conservando sus propiedades mecánicas. Esto quiere decir reducir al máximo posible el peso de una pieza, dejando material sólo allí donde lo necesita para conservar su funcionamiento y propiedades.

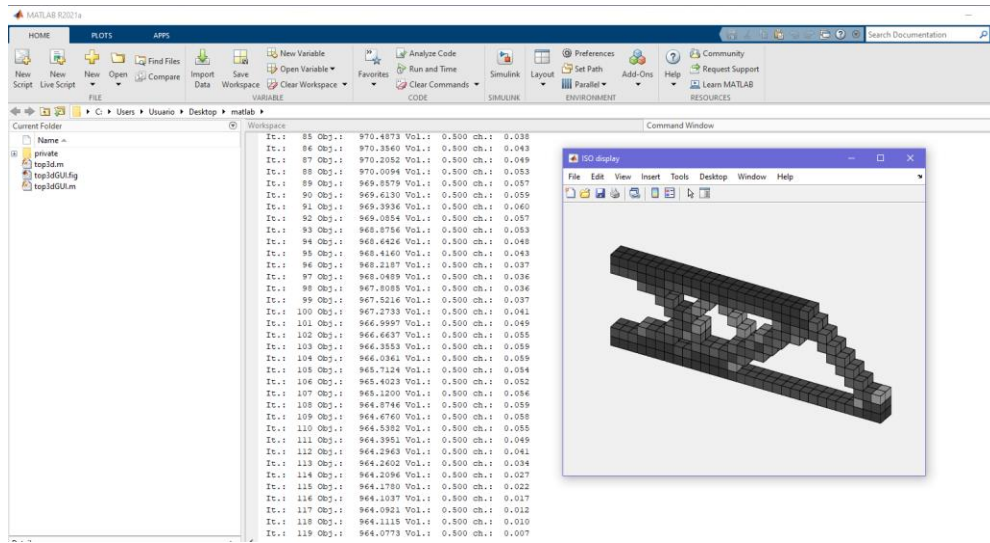
La teoría de la optimización matemática está constituida por un conjunto de resultados y métodos numéricos enfocados a encontrar el mejor candidato entre una colección de alternativas, sin tener que evaluar explícitamente todas esas alternativas. El concepto de optimización está en la base de la ingeniería, puesto que la función clásica del ingeniero es diseñar sistemas novedosos, mejores, más eficientes, y menos costosos. La potencia de los métodos de optimización para determinar el mejor diseño sin comprobar todos los posibles, se basa en la utilización de un nivel relativamente modesto de matemáticas y la tarea de realizar cálculos numéricos iterativos, utilizando procedimientos lógicos claramente definidos o algoritmos implementados en ordenadores.

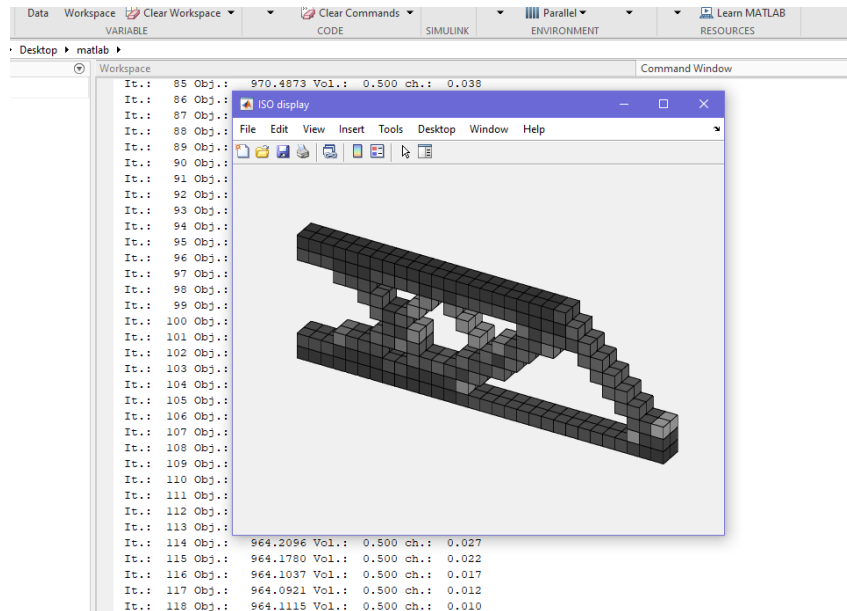
La optimización de forma consiste en modificar la geometría del dominio preservando su topología, es decir sin crear huecos o cavidades en su interior. Este tipo de análisis es usualmente conocido como análisis de sensibilidad al cambio de forma y sus bases matemáticas se encuentran bien establecidas. 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.

En general, un problema de optimización consiste en buscar ciertos valores para determinadas variables, de forma que, cumpliendo un conjunto de requisitos representados habitualmente por ecuaciones o inecuaciones algebraicas, nos proporcionan el mejor valor posible para una función que es utilizada para medir el rendimiento de nuestro diseño.

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.

3) Procedimiento de la programación





top3d(30, 10, 2, 0.5, 3.0, 1.2)

Con la línea de promoción anterior, está resolviendo el problema de la viga en voladizo predeterminada en el programa como se muestra abajo a la izquierda y el resultado de la optimización topológica debería ser similar al de abajo a la derecha.

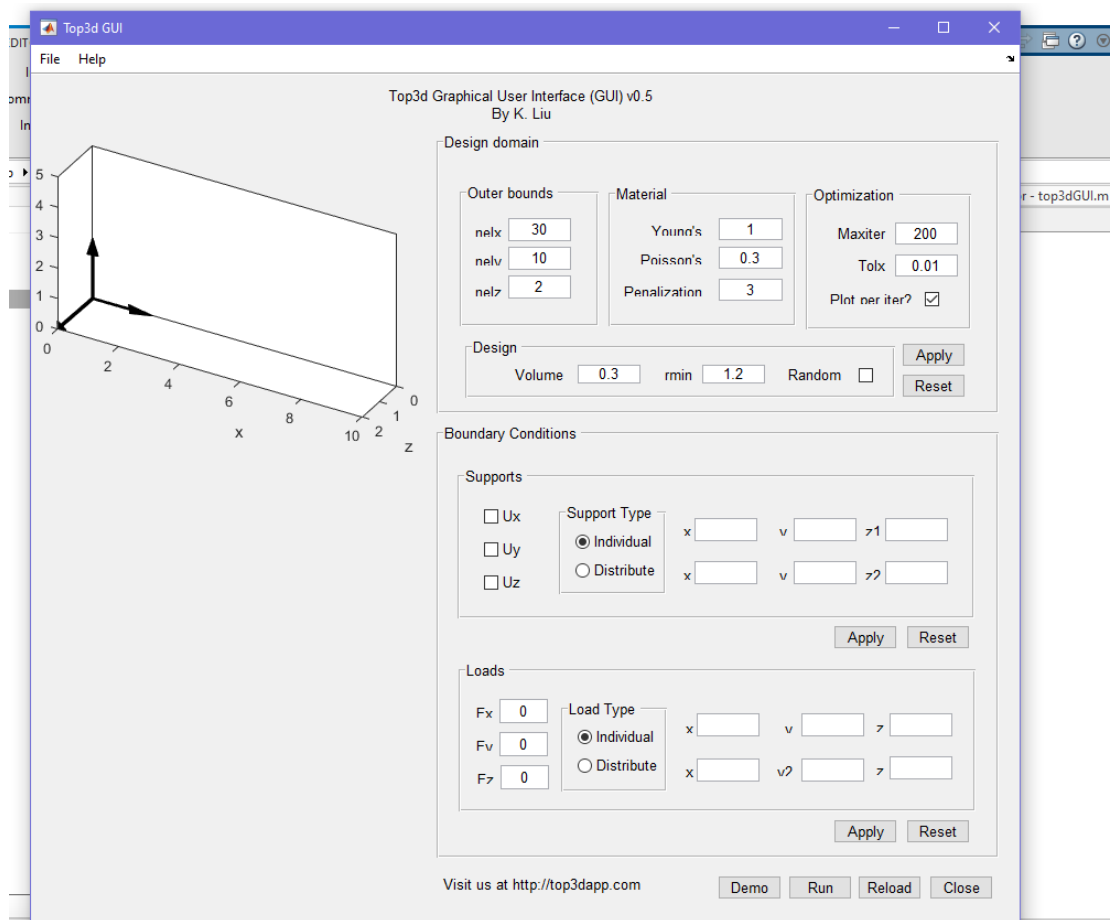
COMANDO GUI PARA INTERFAZ DE GEOMETRIA

```

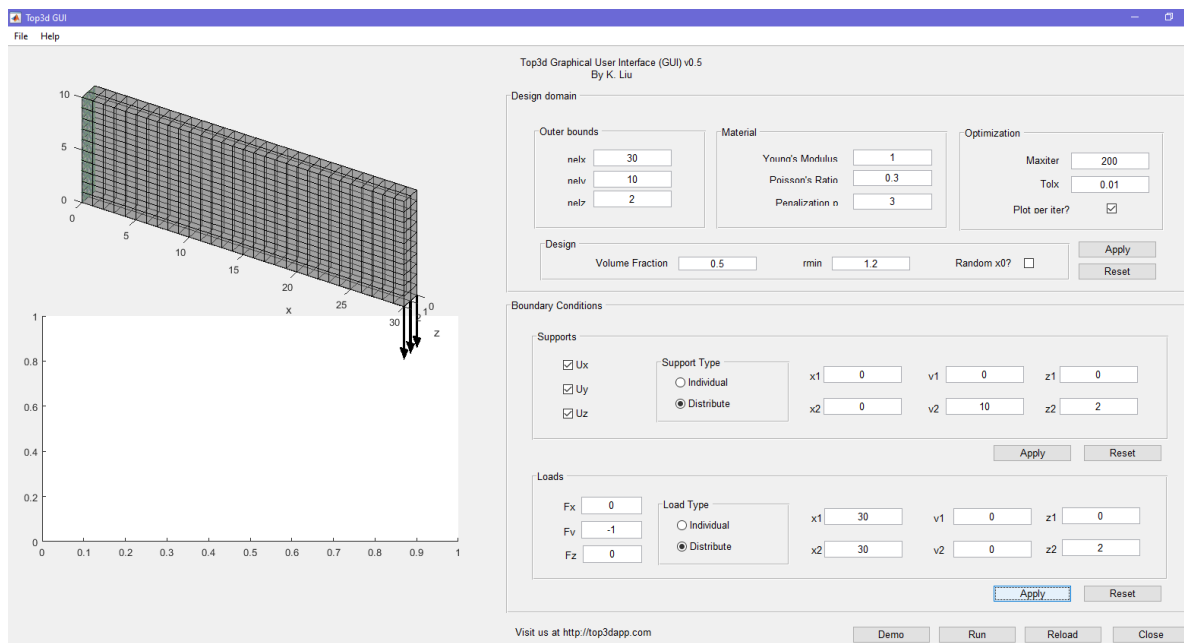
Workspace Command Window
top3dGUI.m
1 function varargout = top3dGUI(varargin)
2 % TOP3DGUI MATLAB code for top3dGUI.fig
3 % TOP3DGUI, by itself, creates a new TOP3DGUI or raises the existing
4 % singleton*.
5 %
6 % H = TOP3DGUI returns the handle to a new TOP3DGUI or the handle to
7 % the existing singleton*.
8 %
9 % TOP3DGUI('CALLBACK',hObject,eventData,handles,...) calls the local
10 % function named CALLBACK in TOP3DGUI.M with the given input arguments.
11 %
12 % TOP3DGUI('Property','Value',...) creates a new TOP3DGUI or raises the
13 % existing singleton*. Starting from the left, property value pairs are
14 % applied to the GUI before top3dGUI_OpeningFcn gets called. An
15 % unrecognized property name or invalid value makes property application
16 % stop. All inputs are passed to top3dGUI_OpeningFcn via varargin.
17 %
18 % *See GUI Options on GUIDE's Tools menu. Choose "GUI allows only one
19 % instance to run (singleton)".
20 %
21 % See also: GUIDE, GUIDATA, GUIHANDLES
22
23 % Edit the above text to modify the response to help top3dGUI
24
25 % Last Modified by GUIDE v2.5 08-Apr-2015 14:59:00
26
27 % Begin initialization code - DO NOT EDIT
28 gui_Singleton = 1;
29 gui_State = struct('gui_Name',       mfilename, ...
30 'gui_Singleton',   gui_Singleton, ...
31 'gui_OpeningFcn',  @top3dGUI_OpeningFcn, ...
32 'gui_OutputFcn',   @top3dGUI_OutputFcn, ...
33 'gui_LayoutFcn',   [], ...
34 'gui_Callback',    []);

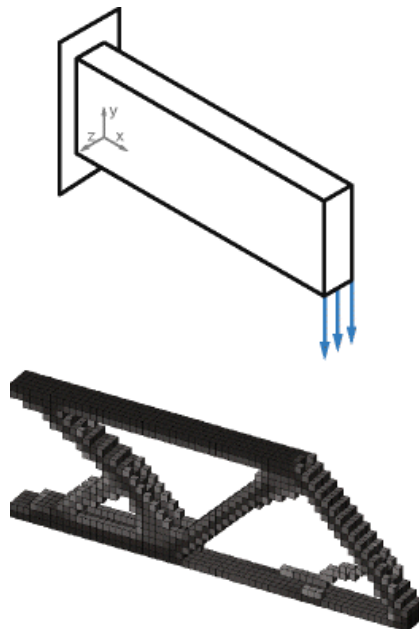
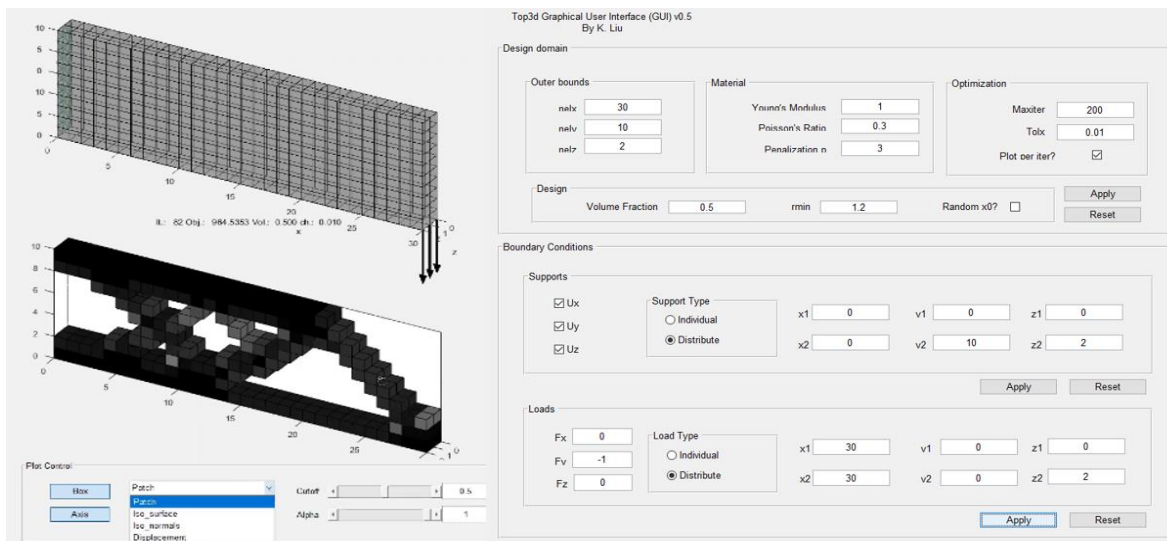
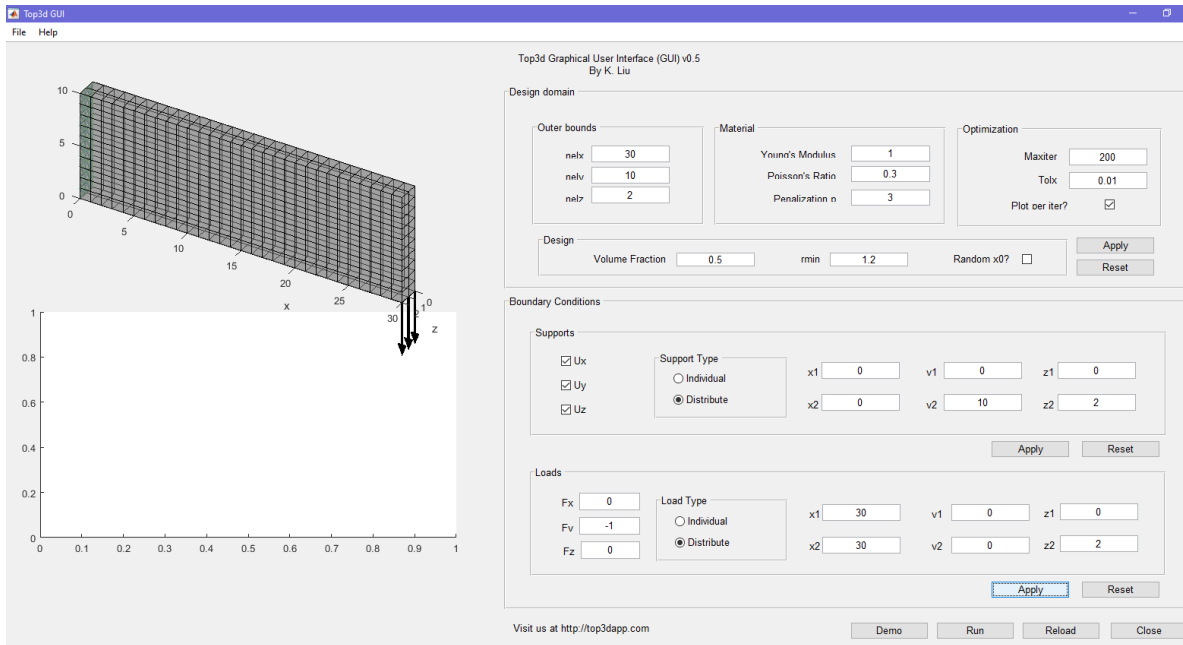
```

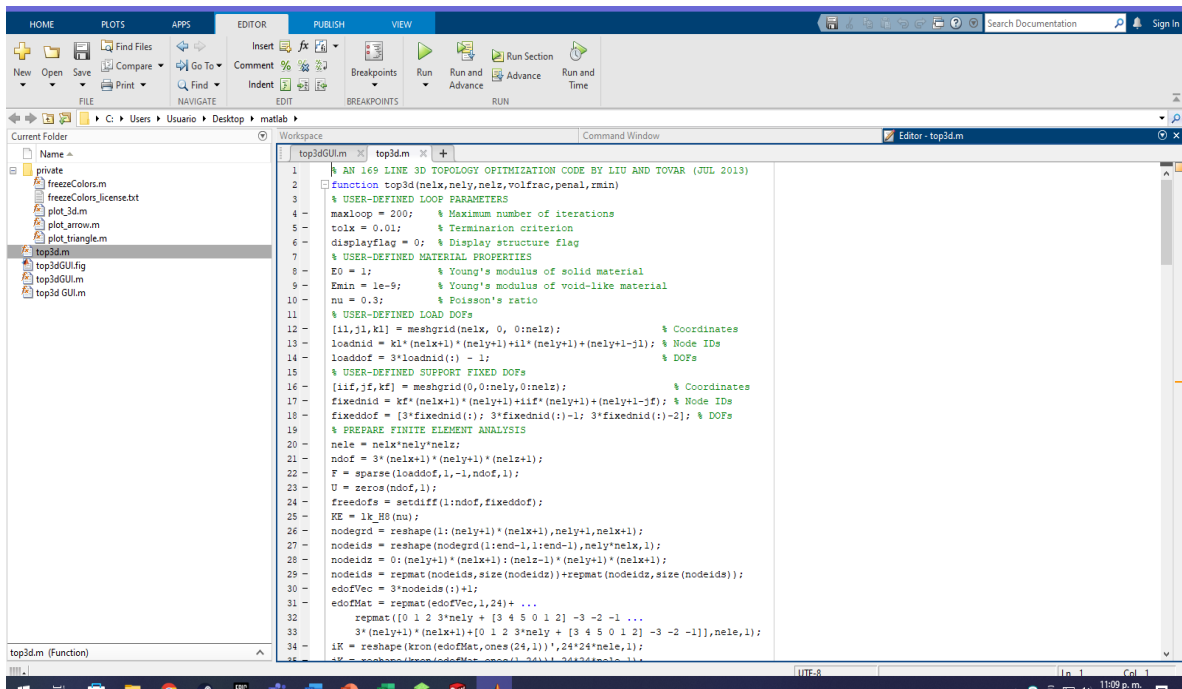
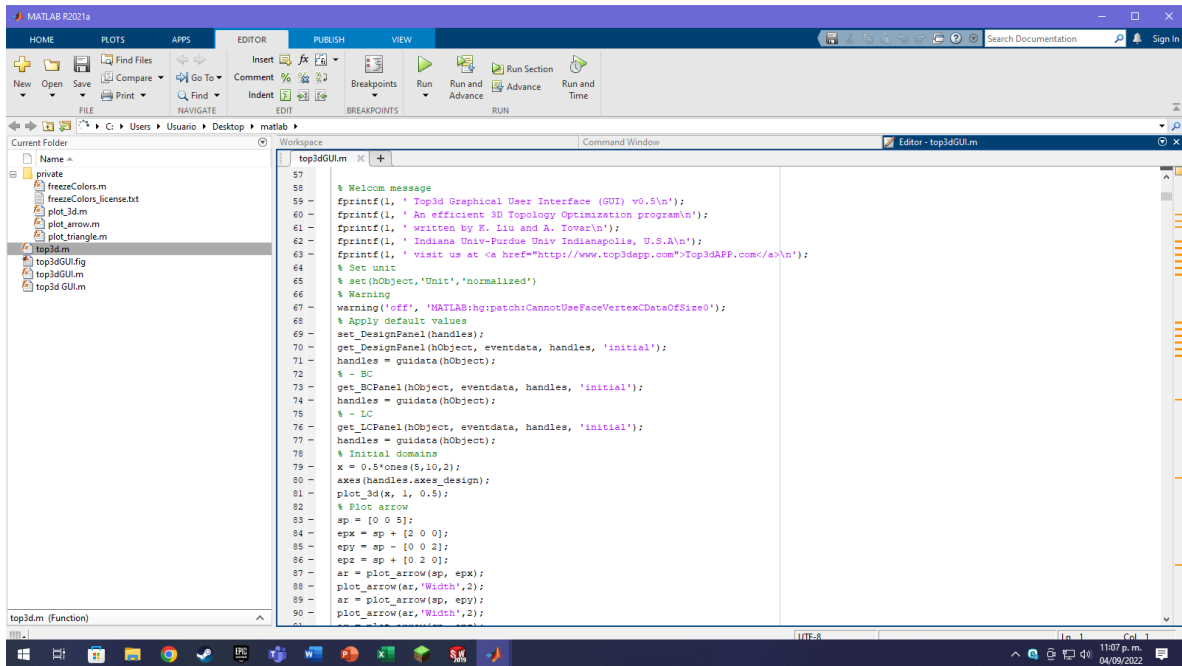
LO CORREMOS Y NOS ABRIRA LA INTERFAZ



Y configuramos los datos según el top3d anteriormente usado







4) Implementación o desarrollo de la programación en sus diferentes vistas

EXPLICACION GENERAL SOBRE EL CODIGO

UNA DE LAS POCAS CONTRIBUCIONES A LOS PROGRAMAS

MATLAB tridimensionales es presentada por Zhou y Wang (2005). Este código, denominado programa de 177 líneas, es el sucesor del programa de 99 líneas de Sigmund (2001) que hereda y amplifica los mismos inconvenientes. Nuestro artículo presenta un programa de 169 líneas denominado top3d que incorpora estrategias eficientes para la optimización topológica tridimensional. Este programa se puede usar de manera efectiva en computadoras personales para generar estructuras de tamaño considerable. Este documento explica el uso de top3d en problemas de optimización de topología de conducción de calor, mecanismo de cumplimiento y cumplimiento mínimo.

Por defecto, el código resuelve un problema de cumplimiento mínimo para la viga en voladizo. El dominio de diseño prismático está completamente restringido en un extremo y se aplica una carga vertical unitaria distribuida hacia abajo en el borde libre inferior. La Figura muestra los resultados de optimización de topología para resolver el problema de cumplimiento mínimo con las siguientes líneas de entrada de MATLAB:

```
top3d(60,20,4,0.3,3,1.5)
```

En primer lugar, cambiar las condiciones de carga

```
12 il = nelx/2; jl = 0 ; kl = nelz/2;  
13 loadnid = kl*(nelx+1)*(nely+1)+il*(nely  
    +1)+(nely+1-jl);  
14 loaddof = 3*loadnid(:) - 1;
```

En segundo lugar, definir las condiciones de contorno correspondientes

```
16 iif = [0 0 nelx nelx]; jf = [0 0 0 0];  
    kf = [0 nelz 0 nelz];  
17 fixednid = kf*(nelx+1)*(nely+1)+iif*(  
    nely+1)+(nely+1-jf);  
18 fixeddof = [3*fixednid(:); 3*fixednid(:)  
    -1; 3*fixednid(:) -2];
```

entonces el problema puede ser promovido por línea:

top3d(40,20,40,0.2,3.0,1.5)

Apêndice C: Programa MATLAB top3d

```

1 % A 169 LINE 3D TOPOLOGY OPTIMIZATION CODE BY LIU AND TOVAR (JUL 2013)
2 function top3d(nelx,nely,nelz,volfrac,penal,rmin)
3 % USER-DEFINED LOOP PARAMETERS
4 maxloop = 200; % Maximum number of iterations
5 tolx = 0.01; % Termination criterion
6 displayflag = 1; % Display structure flag
7 % USER-DEFINED MATERIAL PROPERTIES
8 E0 = 1; % Young's modulus of solid material
9 Emin = 1e-9; % Young's modulus of void-like material
10 nu = 0.3; % Poisson's ratio
11 % USER-DEFINED LOAD DOFs
12 il = nelx; jl = 0; kl = 0:nelz; % Coordinates
13 loadnid = kl*(nelx+1)*(nely+1)+il*(nely+1)+(nely+1-jl); % Node IDs
14 load dof = 3*loadnid(:) - 1; % DOFs
15 % USER-DEFINED SUPPORT FIXED DOFs
16 [jf,kf] = meshgrid(1:nely+1,1:nelz+1); % Coordinates
17 fixednid = (kf-1)*(nely+1)*(nelx+1)+jf; % Node IDs
18 fixeddof = [3*fixednid(:); 3*fixednid(:)-1; 3*fixednid(:)-2]; % DOFs
19 % PREPARE FINITE ELEMENT ANALYSIS
20 nele = nelx*nely*nelz;
21 ndof = 3*(nelx+1)*(nely+1)*(nelz+1);
22 F = sparse(loaddof,1,-1,ndof,1);
23 U = zeros(ndof,1);
24 freedofs = setdiff(1:ndof,fixeddof);
25 KE = lk_H8(nu);
26 nodegrd = reshape(1:(nely+1)*(nelx+1),nely+1,nelx+1);
27 nodeids = reshape(nodegrd(1:end-1,1:end-1),nely*nely,1);
28 nodeidz = 0:(nely+1)*(nelx+1):(nelx-1)*(nely+1)*(nelx+1);
29 nodeids = repmat(nodeids,size(nodeidz))+repmat(nodeidz,size(nodeids));
30 edofVec = 3*nodeids(:)+1;
31 edofMat = repmat(edofVec,1,24)+ ...
32 repmat([0 1 2 3*nely + [3 4 5 0 1 2] -3 -2 -1 ...
33 3*(nely+1)*(nelx+1)+[0 1 2 3*nely + [3 4 5 0 1 2] -3 -2 -1]],nele,1);
34 iK = kron(edofMat,ones(24,1))';
35 jK = kron(edofMat,ones(1,24))';
36 % PREPARE FILTER
37 iH = ones(nele*(2*(ceil(rmin)-1)+1)^2,1);
38 jH = ones(size(iH));
39 sH = zeros(size(iH));
40 k = 0;
41 for kl = 1:nelz

```

```

37 iH = ones(size(iH));
38 jH = ones(size(iH));
39 sH = zeros(size(iH));
40 k = 0;
41 for kl = 1:nelz
42     for il = 1:nelx
43         for jl = 1:nely
44             e1 = (kl-1)*nelx*nely + (il-1)*nely+jl;
45             for k2 = max(kl-(ceil(rmin)-1),1):min(kl+(ceil(rmin)-1),nelx)
46                 for i2 = max(il-(ceil(rmin)-1),1):min(il+(ceil(rmin)-1),nelx)
47                     for j2 = max(jl-(ceil(rmin)-1),1):min(jl+(ceil(rmin)-1),nely)
48                         e2 = (k2-1)*nelx*nely + (i2-1)*nely+j2;
49                         k = k+1;
50                         iH(k) = e1;
51                         jH(k) = e2;
52                         sH(k) = max(0,rmin-sqrt((i1-i2)^2+(j1-j2)^2+(k1-k2)^2));
53                     end
54                 end
55             end
56         end
57     end
58 end
59 H = sparse(iH,jH,sH);
60 Hs = sum(H,2);
61 % INITIALIZE ITERATION
62 x = repmat(volfrac,[nely,nelx,nelz]);
63 xPhys = x;
64 loop = 0;
65 change = 1;
66 % START ITERATION
67 while change > tolx && loop < maxloop
68     loop = loop+1;
69     % FE-ANALYSIS
70     sK = KE(:)*(Emin+xPhys(:)'.*penal*(E0-Emin));
71     K = sparse(iK(:),jK(:),sK(:)); K = (K+K')/2;
72     U(freedofs,:) = K(freedofs,freedofs)\F(freedofs,:);
73     % OBJECTIVE FUNCTION AND SENSITIVITY ANALYSIS
74     ce = reshape(sum((U(edofMat)*KE).*U(edofMat),2),[nely,nelx,nelz]);
75     c = sum(sum(sum((Emin+xPhys.*penal*(E0-Emin)).*ce)));
76     dc = -penal*(E0-Emin)*xPhys.*(penal-1).*ce;
77     dv = ones(nely,nelx,nelz);
78     % FILTERING AND MODIFICATION OF SENSITIVITIES
79     dc(:) = H*(dc(:)./Hs);
80     dv(:) = H*(dv(:)./Hs);
81     % OPTIMALITY CRITERIA UPDATE
82     l1 = 0; l2 = 1e9; move = 0.2;
83     while (l2-l1)/(l1+l2) > 1e-3

```

```

80 dv(:) = H*(dv(:)./Hs);
81 % OPTIMALITY CRITERIA UPDATE
82 l1 = 0; l2 = 1e9; move = 0.2;
83 while (l2-l1)/(l1+l2) > 1e-3
84     lmid = 0.5*(l2+l1);
85     xnew = max(0, max(x-move, min(1, min(x+move, x.*sqrt(-dc./dv/lmid)))));
86     xPhys(:) = (H*xnew(:))./Hs;
87     if sum(xPhys(:)) > volfrac*nele, l1 = lmid; else l2 = lmid; end
88 end
89 change = max(abs(xnew(:)-x(:)));
90 x = xnew;
91 % PRINT RESULTS
92 fprintf(' It.:%5i Obj.:%11.4f Vol.:%7.3f ch.:%7.3f\n', loop, c, mean(xPhys(:)));
93 % PLOT DENSITIES
94 if displayflag, clf; display_3D(xPhys); end
95 end
96 clf; display_3D(xPhys);
97 end
98 %===== AUXILIARY FUNCTIONS =====
99 % GENERATE ELEMENT STIFFNESS MATRIX
100 function [KE] = lk.H8(nu)
101 A = [32 6 -8 6 -6 4 3 -6 -10 3 -3 -3 -4 -8;
102     -48 0 0 -24 24 0 0 0 12 -12 0 12 12 12];
103 k = 1/72*A*[1; nu];
104 % GENERATE SIX SUB-MATRICES AND THEN GET KE MATRIX
105 K1 = [k(1) k(2) k(2) k(3) k(5) k(5);
106     k(2) k(1) k(2) k(4) k(6) k(7);
107     k(2) k(2) k(1) k(4) k(7) k(6);
108     k(3) k(4) k(4) k(1) k(8) k(8);
109     k(5) k(6) k(7) k(8) k(1) k(2);
110     k(5) k(7) k(6) k(8) k(2) k(1)];
111 K2 = [k(9) k(8) k(12) k(6) k(4) k(7);
112     k(8) k(9) k(12) k(5) k(3) k(5);
113     k(10) k(10) k(13) k(7) k(4) k(6);
114     k(6) k(5) k(11) k(9) k(2) k(10);
115     k(4) k(3) k(5) k(2) k(9) k(12);
116     k(11) k(4) k(6) k(12) k(10) k(13)];
117 K3 = [k(6) k(7) k(4) k(9) k(12) k(8);
118     k(7) k(6) k(4) k(10) k(13) k(10);
119     k(5) k(5) k(3) k(8) k(12) k(9);
120     k(9) k(10) k(2) k(6) k(11) k(5);
121     k(12) k(13) k(10) k(11) k(6) k(4);
122     k(2) k(12) k(9) k(4) k(5) k(3)];
123 K4 = [k(14) k(11) k(11) k(13) k(10) k(10);
124     k(11) k(14) k(11) k(12) k(9) k(8);
125     k(11) k(11) k(14) k(12) k(8) k(9);
126     k(13) k(12) k(12) k(14) k(7) k(7);
127     k(10) k(9) k(8) k(7) k(14) k(11);
128     k(10) k(8) k(9) k(7) k(11) k(14)];
129 K5 = [k(1) k(2) k(8) k(3) k(5) k(4);
130     k(2) k(1) k(8) k(4) k(6) k(11);
131     k(8) k(8) k(1) k(5) k(11) k(6);
132     k(3) k(4) k(5) k(1) k(8) k(2);
133     k(5) k(6) k(11) k(8) k(1) k(8);
134     k(4) k(11) k(6) k(2) k(8) k(1)];
135 K6 = [k(14) k(11) k(7) k(13) k(10) k(12);
136     k(11) k(14) k(7) k(12) k(9) k(2);
137     k(7) k(7) k(14) k(10) k(2) k(9);
138     k(13) k(12) k(10) k(14) k(7) k(11);
139     k(10) k(9) k(2) k(7) k(14) k(7);
140     k(12) k(2) k(9) k(11) k(7) k(14)];
141 KE = 1/((nu+1)*(1-2*nu))*...
142     [ K1 K2 K3 K4;
143       K2' K5 K6 K3';
144       K3' K6 K5' K2';
145       K4 K3 K2 K1'];
146 end
147 % DISPLAY 3D TOPOLOGY (ISO-VIEW)
148 function display_3D(rho)
149 [nely, nelx, nelz] = size(rho);
150 hx = 1; hy = 1; hz = 1; % User-defined unit element size
151 face = [1 2 3 4; 2 6 7 3; 4 3 7 8; 1 5 8 4; 1 2 6 5; 5 6 7 8];
152 set(gcf, 'Name', 'ISO display', 'NumberTitle', 'off');
153 for k = 1:nelz
154     z = (k-1)*hz;
155     for i = 1:nelx
156         x = (i-1)*hx;
157         for j = 1:nely
158             y = nely*hy - (j-1)*hy;
159             if (rho(j,i,k) > 0.5) % User-defined display density threshold
160                 vert = [x y z; x y-hx z; x+hx y-hx z; x+hx y z; x y z+hx; x y-hx z+hx; x+hx y-hx z+hx; x+hx y z+hx];
161                 vert(:, [2 3]) = vert(:, [3 2]); vert(:, 2,:) = -vert(:, 2,:);
162                 patch('Faces', face, 'Vertices', vert, 'FaceColor', [0.2+0.8*(1-rho(j,i,k)), 0.2+0.8*(1-rho(j,i,k))]);
163                 hold on;
164             end
165         end
166     end
167 end
168 axis equal; axis tight; axis off; box on; view([30 30]); pause(1e-6);
169 end
170 %=====
171 %===== This code was written by K Liu and A Tovar, Dept. of Mechanical
172 %===== Engineering, Indiana University-Purdue University Indianapolis,
173 %===== Indiana, United States of America
174 %=====
175 %===== Please send your suggestions and comments to: kailliu@iupui.edu
176 %=====
177 %===== The code as well as step-by-step tutorials can be found from the
178 %===== website: http://top3dapp.com
179 %=====
180 %=====
181 %=====
182 %=====
183 %=====
184 %===== Disclaimer:
185 %===== The authors reserves all rights for the program.
186 %===== The code may be distributed and used for educational purposes.
187 %===== The authors do not guarantee that the code is free from errors, and =
188 %===== they shall not be liable in any event caused by the use of the code.=
189 %=====

```

5) Conclusiones

Raúl Emiliano Soto Salas – 1864359

En conclusión con la práctica, el uso de matlab es algo que realmente nos ayuda a la hora de realizar trabajos y actividades, en este caso nos ayudó en la realización de un análisis topológico, personalmente no tenía idea de que en matlab se podía realizar este tipo de problemas y a la vez nos hace dar cuenta de como es que un software te puede facilitar tanto algo que haciendo por otros métodos te tardarías demasiado. Espero seguir aprendiendo más cosas sobre matlab y la biomecánica para poder realizar las siguientes actividades a lo largo del curso.

Melissa Alejandra Jasso Maciel 1897337

Al trabajar en esta práctica pude aprender sobre lo mucho que podemos hacer usando MATLAB, además de facilitarnos nuestros proyectos y ayudarnos con nuestros cálculos, también nos permite realizar simulaciones o más específicamente análisis de elemento finito y de una optimización topológica de algún sistema. Con respecto a los objetivos de la practica podemos decir que se cumplieron con éxito y fue muy interesante la realización de esta ya que es una función que desconocía, pero sé que utilizaré mucho para mis siguientes clases de la materia de biomecánica.

Carlos Antonio Caballero Padilla 1900864

En conclusión, a través de esta práctica #1 logré los objetivos planteados de la misma, ya que ahora conozco más a profundidad las secciones que integran el código de optimización topológica, además comprendí más acerca de Matlab así como del análisis obtenido de la ejecución de la geometría mediante su programación.

Omar Gerardo Ríos Gaytán 1902388

Después de realizar la anterior práctica, en la cual se debía de realizar un análisis topológico, me tome el tiempo de comprobar y estudiar algunas de las funciones que se me podía otorgar al usar el software de MatLab. Esto debido a que, aunque ya lo he usado antes, para lo único que lo usaba era para realizar funciones matemáticas sencillas como las derivaciones o integrales de ecuaciones y/o de las propias ecuaciones diferenciales; o bien, lo usaba para solucionar o comprobar de manera sencilla procedimientos con matrices de cualquier dimensión. Ahora, gracias a la práctica impartida y al realizar la conversión topológica gracias al software, pude entender una operación que jamás había usado en MatLab, y que me va a ayudar demasiado a lo largo de la clase ordinaria y del laboratorio impartido.

Edgar Alan Carrizales Treviño 1904406

Al realizar esta práctica pude entender más el cómo funciona Matlab dentro del área de la geometría que estábamos probando ya que anteriormente en semestres pasados habría usado Matlab, pero de manera más matemática sin necesidad de tratar de programar más a fondo y aunque hubo bastantes problemas al momento de programar esta tipología al final pudimos resolver la problemática y comprender la manera en la que se desea trabajar.

Néstor Eliud Cano García 1909644

La conclusión que puedo tener tras el desarrollo de esta práctica, es el cómo obtuve un mayor entendimiento sobre lo que es la optimización topológica y como está se puede utilizar en lo que es el software de Matlab, también logré tener un mayor entendimiento del funcionamiento de Matlab dentro de la materia de biomecánica, siendo que me recuerda un poco a mis clases de diseño de máquinas donde según la forma que pueda tener una viga podría modificar sus propiedades como lo sería su resistencia a la flexión o la presión, claro también tomando en cuenta cosas como el material usado, sin lugar a duda conocer el concepto de optimización topológica es algo interesante y útil si uno quiere dedicarse a la fabricación de prótesis mecánicas

Bibliografía

- Meza, C. A., Tamayo, F., & Franco, E. E. (2015). Optimización topológica aplicada al diseño de componentes estructurales mecánicos de peso reducido. *El hombre y la máquina*, (46), 72-79.
- Ochoa Sánchez, C. A. (2018). Optimización topológica en estructuras de tres dimensiones usando elementos finitos en el campo elástico lineal. Departamento de Ingeniería Civil y Agrícola.
- 99 Line Topology Optimization Code – O. Sigmund, Department of Solid Mechanics, Building 404, Technical University of Denmark, DK-2800 Lyngby, Denmark.