Práctica 1

Adrián Meléndez Herrera

Juan José Prado Luna

Ana Karen Mendoza González

Daniel Soto Celis

José Sebastián Gálvez Campos

Adrián Sandoval Toscano

5 de septiembre de 2022

Resumen

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, con esto resolveremos un problema de ingeniería sobre la optimización topológica, un problema de ingeniería con el cual podemos optimizar diferentes procesos de construcción de piezas en diversas areas de la industria.

1. Introducción

El Método de Optimización Topológica (MOT) es una técnica computacional que permite diseñar estructuras óptimas, usando solamente una fracción de volumen del dominio de diseño total, sujeto a ciertas condiciones de contorno y cargas, distribuyendo dicha cantidad de material de forma óptima dentro de dicho dominio. Este método maximiza o minimiza una función objetivo iterativamente combinando técnicas de optimización lineal con el Método de Elementos Finitos (MEF)[2].

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)[3].

1.1. Aplicaciones

La industria automotriz abordó rápidamente este problema debido a la reducción de costes mediante el ahorro en materias primas asociadas con los tamaños de serie. De hecho, la reducción de unos pocos gramos por cada vehículo, en una producción de varios millones de unidades, representa toneladas de material ahorrado. Como ejemplo de esto tenemos el chasis impreso en 3D del Light Rider, una pieza que pesa solo 6 kilos gracias a una distribución óptima del material. Más recientemente, está la parte de suspensión del Fiat Chrysler Automóviles, que reúne más de 12 componentes diferentes en uno mismo. Al centrarse en la optimización topológica, los diseñadores redujeron su peso final en un 36 %.

La aeronáutica es sin duda otro sector interesado en la optimización topológica, con el objetivo de reducir costes indirectos. Un avión más ligero consume menos combustible, lo que, a la larga, genera importantes ahorros para una aerolínea. Esto es lo que el diseñador Andreas Bastian demostró con sus asientos de avión. Los diseñó un 54 % más ligeros, que, en su conjunto, supondría una reducción muy significativa del peso de un avión. Más allá del peso, la optimización topológica permite, especialmente al sector aeronáutico, imaginar formas mucho más complejas, ya que la industria se libera de las limitaciones impuestas por los moldes.

Por último, la medicina también está investigando este método de diseño, especialmente para la fabricación de implantes a medida. La optimización topológica permite imitar la densidad y rigidez de los huesos, al tiempo que reduce su peso total. De hecho, muchos implantes incorporan estructuras de celosía y siguen siendo tan sólidos como los diseñados tradicionalmente, o incluso más para algunos[1].

2. Desarrollo

2.1. Utilización de Matlab

1. Para esta práctica se utilizó MATLAB Online. Primeramente, se abre MATLAB y que se muestre su pantalla principal.

2. Después de seleccionar en la barra de herramientas, se selecciona New y luego Script.

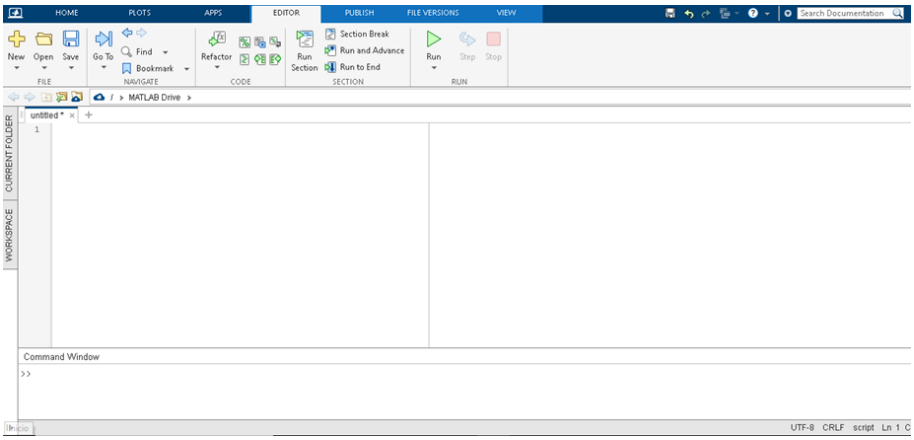

Figura 2: Pantalla principal de matlab

3. Una vez terminado el código, se guarda el archivo colocándole el nombre de “PRACTICA1”.

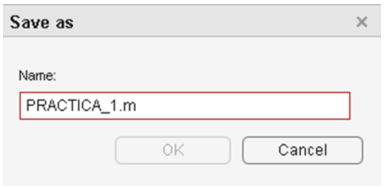

Figura 3: Ventana emergente de new script

4. Por último, hay que ejecutar el programa desde la ventana de comando de MATLAB. El código que se proporcionó, viene preparado para optimizar un dominio de diseño con cargas y restricciones, este caso en particular, es evaluado y simulado cuando escribimos desde la línea de comando de MATLAB “topp(60,20,0.5,3.0,1.5)”. El resultado de esta simulación se muestra en la figura 3.

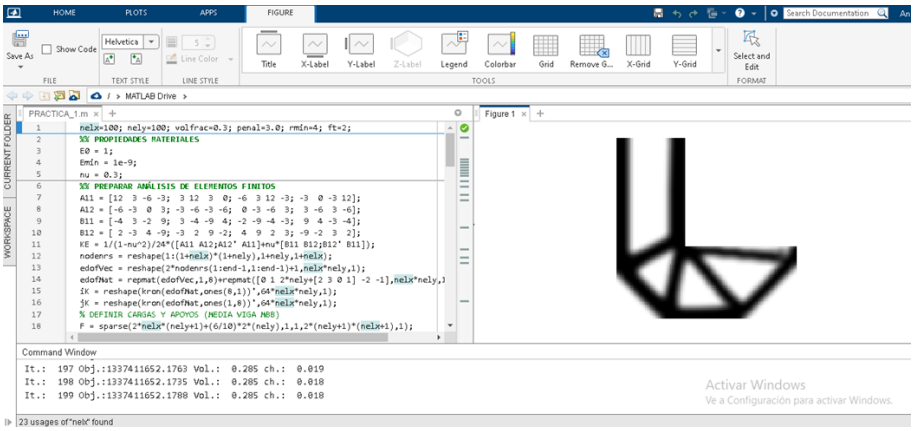

Figura 4: Resultado de la optimización de la geometría

2.2. Codigo desarrollado

```
%% VARIABLES DE ENTRADA
nelx=20; nely=20; volfrac=0.33; penal=3.0; rmin=1.5; ft=1;
%% PROPIEDADES MATERIALES
E0 = 1;
Emin = 1e-9;
nu = 0.3;
```

```
%% ANALISIS DE ELEMENTO FINITO
A11 = [12  3 -6 -3;  3 12  3  0; -6  3 12 -3; -3  0 -3 12];
A12 = [-6 -3  0  3; -3 -6 -3 -6;  0 -3 -6  3;  3 -6  3 -6];
B11 = [-4  3 -2  9;  3 -4 -9  4; -2 -9 -4 -3;  9  4 -3 -4];
B12 = [ 2 -3  4 -9; -3  2  9 -2;  4  9  2  3; -9 -2  3  2];
KE = 1/(1-nu^2)/24*([A11 A12;A12' A11]+nu*[B11 B12;B12' B11]);
nodenrs = reshape(1:(1+nelx)*(1+nely),1+nely,1+nelx);
edofVec = reshape(2*nodenrs(1:end-1,1:end-1)+1,nelx*nely,1);
edofMat = repmat(edofVec,1,8)+repmat([0 1 2*nely+[2 3 0 1] -2 -1],nelx*nely,1);
iK = reshape(kron(edofMat,ones(8,1))',64*nelx*nely,1);
jK = reshape(kron(edofMat,ones(1,8))',64*nelx*nely,1);
%% DEFINIR CARGAS Y APOYOS
F = sparse(2,1,1,2*(nely+1)*(nelx+1),1);
U = zeros(2*(nely+1)*(nelx+1),1);
fixeddofs = 2*(nelx)*(nely+1)+1:2*(nely+1)*(nelx+1);
alldofs = 1:2*(nely+1)*(nelx+1);
freedofs = setdiff(alldofs,fixeddofs);
passive = zeros(nely,nelx);
for i = 1:nelx
  for j = 1:nely
      if ((i)^2+(j-nely)^2) < (0.65*nelx)^2
          passive(j,i) = 1;
      end
  end
end
%% PREPARAR FILTROS
iH = ones(nelx*nely*(2*(ceil(rmin)-1)+1)^2,1);
jH = ones(size(iH));
sH = zeros(size(iH));
k = 0;
for i1 = 1:nelx
  for j1 = 1:nely
    e1 = (i1-1)*nely+j1;
    for i2 = max(i1-(ceil(rmin)-1),1):min(i1+(ceil(rmin)-1),nelx)
      for j2 = max(j1-(ceil(rmin)-1),1):min(j1+(ceil(rmin)-1),nely)
        e2 = (i2-1)*nely+j2;
        k = k+1;
        iH(k) = e1;
        jH(k) = e2;
        sH(k) = max(0,rmin-sqrt((i1-i2)^2+(j1-j2)^2));
      end
    end
  end
end
H = sparse(iH,jH,sH);
Hs = sum(H,2);
```

```
%% INICIALIZAR ITERACIÓN
x = repmat(volfrac,nely,nelx);
xPhys = x;
loop = 0;
change = 1;
obj = NaN(200,1);
changeplot = NaN(200,1);
volume = NaN(200,1);
%% INICIAR ITERACIÓN
while change > 0.01
  loop = loop + 1;
  %% FE-ANÁLISIS
  sK = reshape(KE(:)*(Emin+xPhys(:)'.^penal*(E0-Emin)),64*nelx*nely,1);
  K = sparse(iK,jK,sK); K = (K+K')/2;
  U(freedofs) = K(freedofs,freedofs)\F(freedofs);
  %% FUNCIÓN OBJETIVO Y ANÁLISIS DE SENSIBILIDAD
  ce = reshape(sum((U(edofMat)*KE).*U(edofMat),2),nely,nelx);
  c = sum(sum((Emin+xPhys.^penal*(E0-Emin)).*ce));
  dc = -penal*(E0-Emin)*xPhys.^(penal-1).*ce;
  dv = ones(nely,nelx);S
  obj(loop+1) = c;
  %% FILTRADO/MODIFICACIÓN DE SENSIBILIDADES
  if ft == 1
    dc(:) = H*(x(:).*dc(:))./Hs./max(1e-3,x(:));
  elseif ft == 2S
    dc(:) = H*(dc(:)./Hs);
    dv(:) = H*(dv(:)./Hs);
  end
  %% CRITERIOS DE OPTIMALIDAD ACTUALIZACIÓN DE LAS VARIABLES DE DISEÑO Y DENSIDADES FÍSICAS
  l1 = 0; l2 = 1e9; move = 0.2;
  while (l2-l1)/(l1+l2) > 1e-3
    lmid = 0.5*(l2+l1);
    xnew = max(0,max(x-move,min(1,min(x+move,x.*sqrt(-dc./dv/lmid)))));
    if ft == 1
      xPhys = xnew;
    elseif ft == 2
      xPhys(:) = (H*xnew(:))./Hs;
    end
    if sum(xPhys(:)) > volfrac*nelx*nely, l1 = lmid; else, l2 = lmid; end
    xPhys(passive==1) = 0;
    xPhys(passive==2) = 1;
  end
  change = max(abs(xnew(:)-x(:)));
  changeplot(loop+1) = change;
  volume(loop+1) = mean(xPhys(:));
  x = xnew;
```

```
99      %% IMPRIMIR RESULTADOS
100     fprintf(' It.:%5i Obj.:%11.4f Vol.:%7.3f ch.:%7.3f\n',loop,c, ...
101       mean(xPhys(:)),change);
102     if exist('OCTAVE_VERSION', 'builtin') ~= 0
103       fflush(stdout);
104     end
105      %% DENSIDADES GRAFICAS
106      colormap(gray); imagesc(1-xPhys); caxis([0 1]); axis equal; axis off; drawnow;
107   end
```

Figura 5: Código de 99 líneas para la optimización topológica

3. Conclusiones

La optimización topológica es una técnica utilizada para el análisis estructural. La optimización topológica implementada resultó ser confiable en sus resultados. La implementación de un programa de optimización topológica no requiere de recursos, adicionales a los de poder intervenir un programa de elementos finitos.

Se puede usar MATLAB para generar un análisis de elemento finito para objetos de ámbito simple y que se pueden usar para diferentes casos, además de generar un buen soporte que nos ayudará mucho en este caso. Donde se observo que los software de hoy en día nos apoyan mucho con cálculos e impresiones que nos facilitan el poder generar nuevas ideas e ir más rápido en nuestras investigaciones.

Por lo tanto, MATLAB es una gran herramienta que ha ayudado a generar diferentes soluciones en varios campos de aplicación por medio de métodos como la optimización topológica, que se basa principalmente en generar y mantener una estructura adecuada para diferentes piezas que son utilizadas en campos, por mencionar la automotriz manteniendo sus funcionalidades mecánicas. Cabe mencionar también que la generación del código fue complicada, ya que algunos comandos no eran comprendidos fácilmente, por eso mismo fue necesario el realizar prueba y error para lograr solucionar el error marcado.

Referencias

[1] L. C. La optimización topológica en la impresión 3d. 3dnatives, 2020. URL `https://www.3dnatives.com/es/optimizacion-topologica-10012017/`.

[2] Wilfredo Montealegre Rubio Fransisco Ramirez Gil, Estevban Sepulveda Orozco. Diseño de mecanismos flexibles mediante el método de optimización topológica. *Departamento de ingeniería mecánica, Facultad Minas, Universidad Nacional de Colombia*, 2013.

[3] O. Sigmund. 99 line topology optimization code. *Technical University of Denmark, DK-2800 Lyngby, Denmark*, 2017.