

Laboratorio de Biomecánica

Práctica 5: Optimización de una prótesis de pie

Nombre	Matrícula	Carrera
Víctor Hugo Puente Álvarez	1929757	IMTC
Alejandro Hernández Navarro	1923272	IMTC
Angela Rodríguez Flores	1896624	IMTC
Diego Ávila González	1853396	IMTC
Osiris Acosta Cisneros	1992234	IMTC

Día y Hora: Martes V1

Brigada: 204

Docente: Isaac Estrada

Fecha de entrega: de 2022

1) Nombre y definición de la geometría

El objetivo de esta práctica es presentar una propuesta de análisis de formas y de la programación para la ejecución de la optimización (descripción funcional) de características de trabajo específicas y sus ventaja(s).

La geometría que se propuso a analizar y optimizar fue la de una prótesis de pie haciendo uso del algoritmo de optimización topológica con el fin de conseguir una optimización lo más apta posible de su estructura para un uso real, que soporte las cargas y distribuya el peso de la manera más óptima posible.

2) Estado del arte

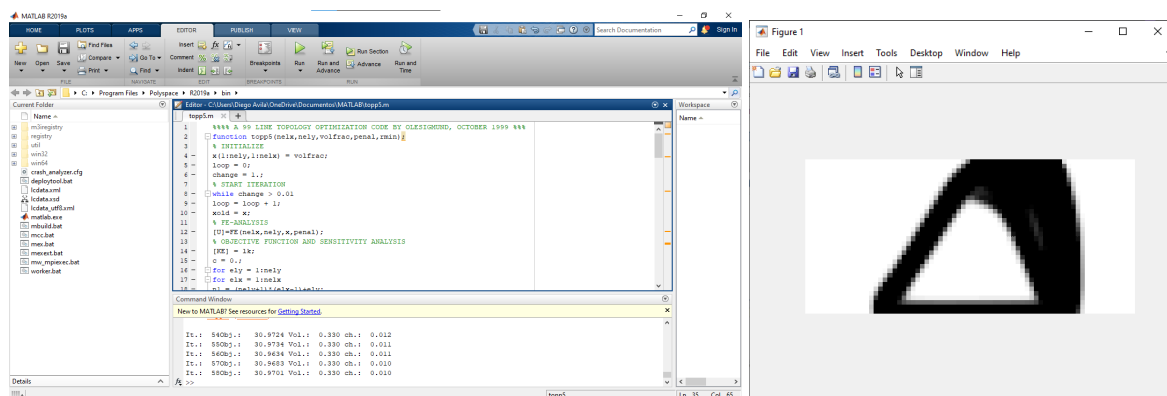
Una prótesis es un dispositivo de extensión adaptable al organismo que reemplaza una parte del cuerpo que no se encuentra. Esta permite cumplir la función que ese segmento o estructura cumplía anteriormente.

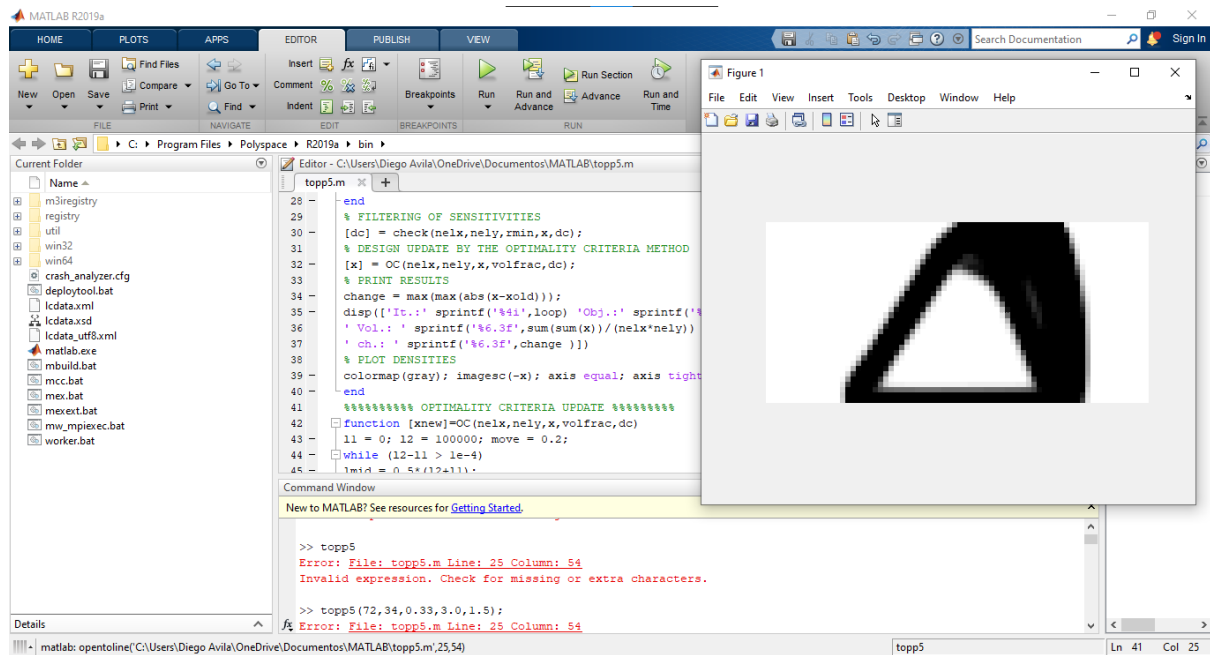
Las prótesis se utilizan desde la antigüedad para reemplazar las estructuras faltantes en el cuerpo y de esta forma devolverle la funcionalidad luego de una amputación o malformación congénita, además del sentido estético a ese segmento afectado.

Es un elemento artificial que se adosa al cuerpo para brindar la función que anteriormente realizaba en el cuerpo, devolviendo la independencia y autonomía en las diferentes actividades de la vida diaria en caso de haber afectado esta área.



Imagen 5.1. Ejemplo de prótesis.





topp5(72,34,0.33,3.0,1.5);

Código 1

```
%% A 99 LINE TOPOLOGY OPTIMIZATION CODE BY
% OLESIGMUND, OCTOBER 1999 %
function topp5(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;
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],i);
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
% 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./lmi
d)))));
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-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
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% 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),5); U
=sparse(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 LOADSAND SUPPORTS(HALF MBB-BEAM)
F(3222,1) = -1;
F(3782,2) = -1;
F(2662,3) = -1;

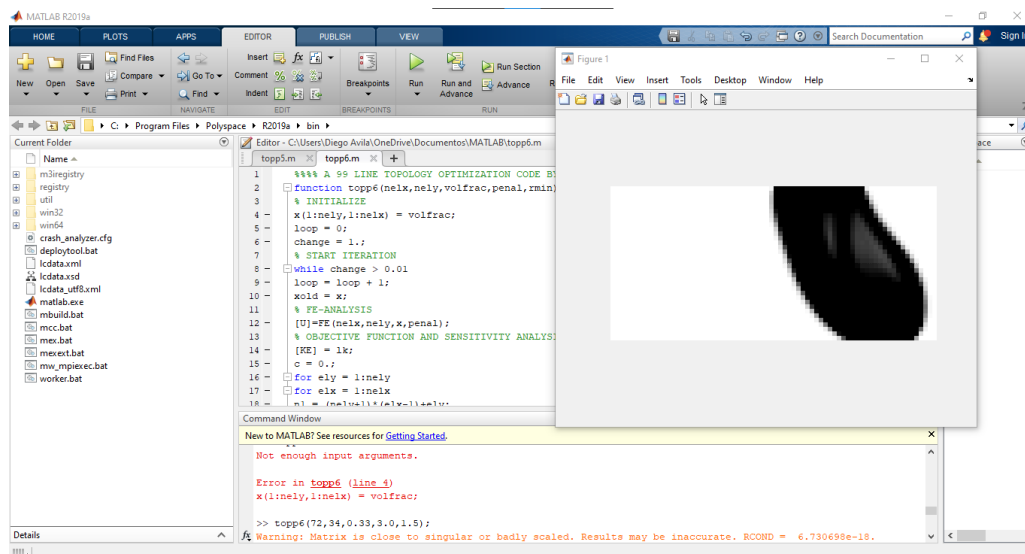
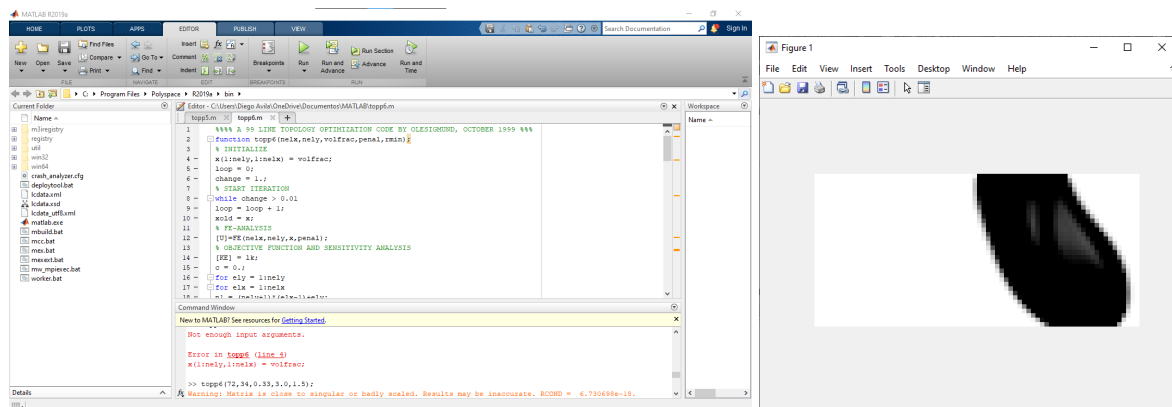
```

```

F(2942,4) = -1;
F(3502,5) = -1;
fixeddofs
union([560:2*(nely+1):1260],[3920:2*(nely+1):4620]);
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)];

```

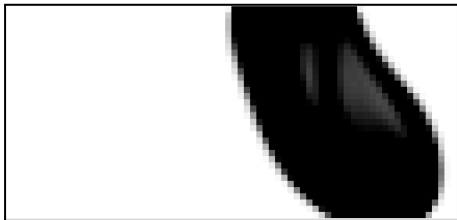
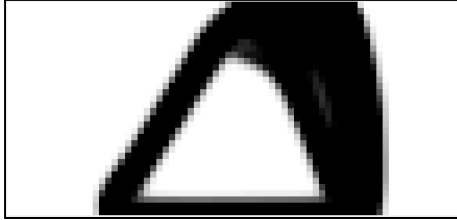
Parte 2



```
topp6(72,34,0.33,3.0,1.5);
```

5) Resultados

Ambas partes unidas:



Código 2

```
%%%%%%%% A 99 LINE TOPOLOGY OPTIMIZATION
CODE BY OLESIGMUND, OCTOBER 1999 %%%%
function topp6(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;
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],i);
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
% 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(['!t.: ' 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-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
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% 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),5); U
=sparse(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 LOADSAND SUPPORTS(HALF
MBB-BEAM)
F(3222,1) = -1;
F(3782,2) = -1;

F(2662,3) = -1;
F(2942,4) = -1;
F(3502,5) = -1;
fixeddofs = [3920:2*(nely+1):4620];
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)];

```

6) Conclusiones

Alejandro Hernandez Navarro

Durante esta práctica se propuso una geometría para una prótesis de pie con la finalidad de aplicar la optimización topológica y poder optimizar la geometría propuesta, reduciendo la masa y manteniendo la rigidez de la pieza. Para lograr esto se utilizaron conocimientos previos respecto al código de optimización topológica en Matlab para adaptarlo al problema que se está trabajando, una vez que se propuso el diseño con sus respectivas restricciones y condiciones frontera, se adaptó el código de Matlab de optimización topológica para así obtener el diseño final.

Diego Avila González

Esta práctica es la última que realizaremos del laboratorio de biomecánica y por lo tanto como su nombre lo indica podremos realizar un diseño de un cuerpo biológico y mecánico que tienen movimiento, nada mejor que realizar el diseño de un pie humano, para esto pondremos en práctica lo que hemos aprendido en las pasadas prácticas ya que este es un diseño más completo y complejo, por lo tanto influenciaron más cosas para poderlo realizar correctamente, con esta práctica

concluimos que con la optimización topológica si podemos realizar usos en la biomecánica y podremos realizar prótesis de varias partes del cuerpo.

Victor Hugo Puente Alvarez

Por medio de esta practica se puede ver como diseñar piezas en diferentes etapas, para que al momento de hacer un ensamble tener como resultado una que en este caso fue una protesis para un pie, se utilizo de tal forma los conocimientos vistos en las practicas anteriores para optimizar dichas piezas y minimizar su masa y maximizar su rigidez para tener el mejor resultado posible.

Osiris Acosta Cisneros

En esta práctica vimos como diseñar un pie humano, dónde pudimos a prueba todo lo que hemos aprendido hasta el momento en el laboratorio. Pudimos aplicar el conocimiento de manera efectiva y se puede concluir que la optimización topológica también se puede utilizar es parte biológicas.

Referencias bibliográficas

- FisioOnline. (s.f.). *Definición - Qué es prótesis*. FisioOnline. Recuperado 24 de octubre de 2022, de <https://www.fisioterapia-online.com/glosario/protesis>

