



# UANL

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN



# FIME

FACULTAD DE INGENIERÍA MECÁNICA Y ELÉCTRICA

## LABORATORIO DE BIOMECÁNICA

PERIODO ESCOLAR AGOSTO – DICIEMBRE 2022

Nombre de la actividad

Práctica 5

Nombre del alumno	Matricula
Leonardo Daniel De Leon Fuentes	1991978
Fatima Montserrat Castro Nuñez	1991834
Brandon Geovanny Espinosa Alcocer	1938292
Erick Eduardo Landa Gonzalez	1992037
Susana Regina Macías de Luna	1992013

Grupo	Salon	Dia clase	Hora
309	12BMC	Miércoles	N5

Nombre del profesor: Yadira Moreno Vera

Fecha de entrega:

02/11/2022

## **Objetivo**

El estudiante deberá presentar una propuesta de análisis de forma y de la programación para la ejecución de la optimización (descripción funcional) de características de trabajo específicas que presenta la(s) ventaja(s) (mencionar ventajas)

### **1) Nombre y definición de la forma GEOMETRICA**

En esta práctica se optimizará la prótesis de un pie, el objetivo de esta optimización será reducir la masa de la prótesis y reducir el costo de esta.

Existen diferentes componentes de prótesis para pierna y describiremos a grandes rasgos en qué consiste una prótesis de pierna básica.

Los componentes para una prótesis de pierna arriba de rodilla son tres:

- 1.- El socket es el componente de prótesis para pierna que va en contacto con el muñón.
- 2.- La rodilla de prótesis que es la que permite el movimiento de flexión extensión de la rodilla de prótesis para pierna.
- 3.- El pie de prótesis para pierna que es el que contacta con el suelo para dar el siguiente paso.

El socket es un componente fundamental para prótesis de pierna arriba de rodilla como para la prótesis de pierna abajo de rodilla que estará en contacto directo con el muñón del paciente amputado y permitirán realizar la contención de interface entre la prótesis de pierna que adquiera el paciente y el muñón y este socket será distinto dependiendo de la necesidad de cada paciente.

El pie de prótesis de pierna básica es el componente que me va a permitir amortiguar el primer contacto con el talón, para posteriormente trasladar el peso del cuerpo a la punta del pie de la prótesis para pierna y despegar el peso del cuerpo del suelo. JJ



## 2) Estado del arte

Existen diferentes tipos de prótesis: pasiva, semi-activa y activa. Sin embargo las dos primeras no realizan movimientos motores deseables (caminar, correr o realizar actividades recreativas); es por ello que actualmente se estudian y desarrollan prótesis activas, las cuales tienen actuadores, sensores y microcontroladores para determinar el modo de caminar al sensar los estados del paciente. Entre las prótesis activas están aquellas eléctricas y las electrohidráulicas.

Las prótesis de pierna pueden variar de acuerdo con sus componentes y nivel de amputación; pues no es lo mismo para un paciente que tienen una amputación debajo de rodilla a un paciente al que se le realizó la amputación arriba de rodilla y que requerirá de una cantidad mayor de componentes, para sustituir la articulación perdida.

Muchas investigaciones están enfocadas en el estudio de la geometría del implante, dejando a un lado la optimización topológica, y cómo cargar un peso mayor al de los elementos humanos, puede llegar a no ser funcional.

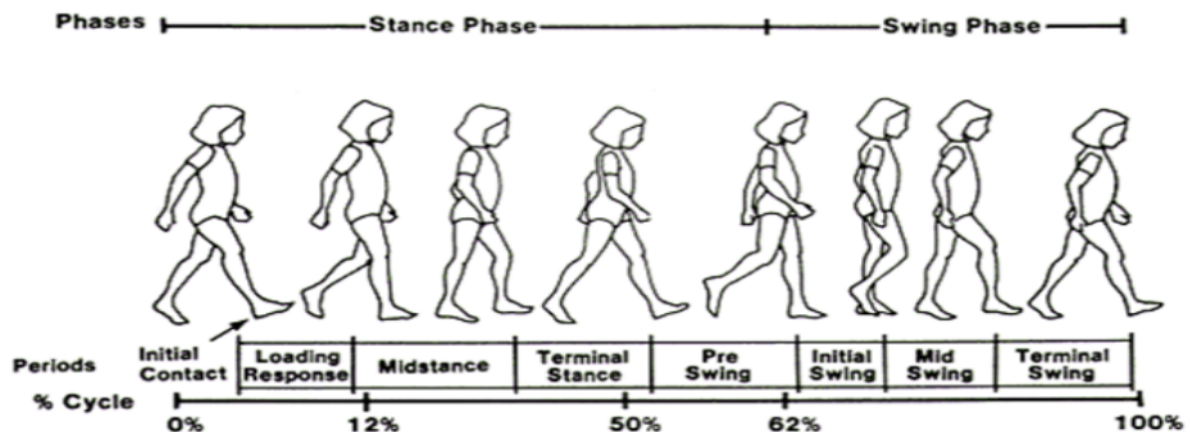
## 3) Propuesta de diseño de la geometría, alcances y limitaciones

Para la propuesta de diseño de esta práctica se comprenderá y analizará el comportamiento de un pie, en 3 fases que tiene el ser humano a la hora de marchar.

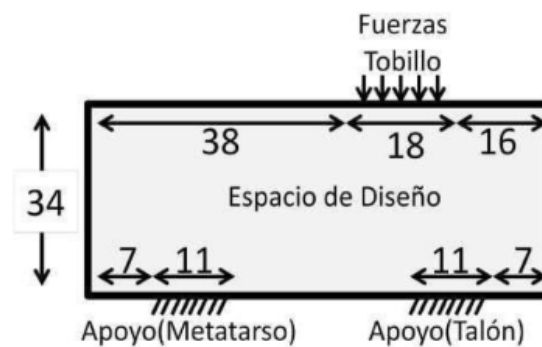
El desarrollo de la marcha es un proceso altamente complejo que se adquiere durante las primeras etapas de la vida, en donde los sistemas motor y perceptivo se tienen que desarrollar en los niños de forma independiente, por medio de la

exploración y control de los sistemas nervioso, muscular, esquelético y propioceptivo.

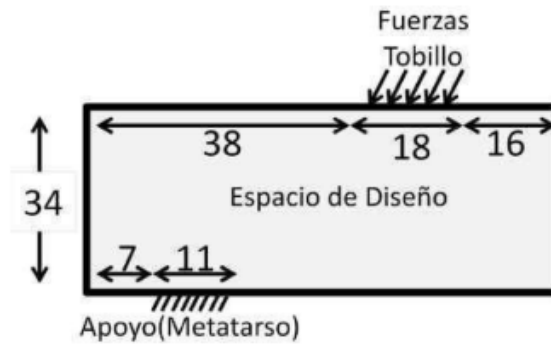
Cuando alguno de los sistemas sufre alguna alteración dan como resultado factores limitantes para el desarrollo de un patrón de marcha correcto y adaptable.



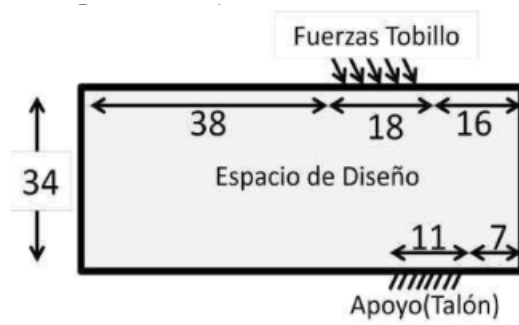
**Normal:** El talón y el área metatarsal son los apoyos, la fuerza que se aplica sobre el tobillo es de 500N.



**Despegue:** El área metatarsal es el apoyo, en este caso la fuerza es de 500N que se aplica sobre el tobillo a un ángulo de 30°.



**Apoyo:** El área del talón es el apoyo, la fuerza es de 500N de igual manera se aplica sobre el tobillo a un ángulo de  $60^\circ$ .



#### 4) Pasos del desarrollo de la programación

Para la programación de los 3 casos nos basaremos en el código de nuestra primera práctica, del cual vamos a editar solo dos secciones:

La primera sección que se tiene que editar es la de “% OBJECTIVE FUNCTION AND SENSITIVITY ANALYSIS”, la cual será igual para los 3 casos, ya que se trata del espacio completo de un pie, en donde se aplicarán las cargas y estarán los apoyos (sección que se edita después). El código con el que se reemplaza al original es el que se muestra en la figura 4.1, que define el espacio de diseño.

```
14 % OBJECTIVE FUNCTION AND SENSITIVITY ANALYSIS
15 - [KE] = lk;
16 - c = 0.;
17 - for ely = 1:nely
18 -   for elx = 1:nelx
19 -     n1 = (nely+1)*(elx-1)+ely;
20 -     n2 = (nely+1)* elx +ely;
21 -     dc(ely,elx)=0.;
22 -     for i=1:5
23 -       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);
24 -       c = c + x(ely,elx)^penal*Ue'*KE*Ue;
25 -       dc(ely,elx) = dc(ely,elx)-penal*x(ely,elx)^(penal-1)*Ue'*KE*Ue;
26 -     end
27 -   end
28 - end
```

Figura 4.1: Código para definir el espacio de diseño.

La otra sección que se tiene que editar es la de “% DEFINE LOADSAND SUPPORTS(HALF MBB-BEAM)”, que es donde se definirán las cargas y los apoyos que se tiene. Para todos los casos las cargas serán iguales (fig. 4.2), pero los apoyos serán diferentes para cada caso, ya que el estudio de cada caso se basa en donde se apoyará el pie, por lo que a continuación se muestra cada código que se tiene que implementar.

```
82 % DEFINE LOADSAND SUPPORTS (HALF MBB-BEAM)
83 - F(3222,1) = -1;
84 - F(3782,2) = -1;
85 - F(2662,3) = -1;
86 - F(2942,4) = -1;
87 - F(3502,5) = -1;
```

Figura 4.2: Cargas definidas.

##### a. Pie “completamente” apoyado:

El siguiente código (fig 4.2a) es para definir un apoyo “completo” del pie, pero que se basa en en apoyo del pie en el talón y en el metatarso, ya que así es normalmente como se apoya el pie al estar “completamente” apoyado.

```

88 - fixeddofs = union([560:2*(nely+1):1260],[3920:2*(nely+1):4620]);
89 - alldofs = [1:2*(nely+1)*(nelx+1)];
90 - freedofs = setdiff(alldofs,fixeddofs);

```

Figura 4.2a: Código para los apoyos en el metatarso y en el talón del pie.

## CODIGO COMPLETO

```

%%% A 99 LINE TOPOLOGY OPTIMIZATION CODE BY OLESIGMUND %%%
%%% PRÁCTICA 5 - PRIMER CASO: APOYO COMPLETO %%%
function topp5a(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./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),5);
U = sparse(2*(nely+1)*(nex+1),5);
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(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)*(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)];

```

## b. Pie apoyado en el talón:

El código de la figura 4.2b es para definir que el apoyo está ubicado en el talón.

```

88 - fixeddofs = [3920:2*(nely+1):4620];
89 - alldofs = [1:2*(nely+1)*(nelx+1)];
90 - freedofs = setdiff(alldofs,fixeddofs);

```

Figura 4.2b: Código para el apoyo en el talón del pie.

## CODIGO COMPLETO

```

%%% A 99 LINE TOPOLOGY OPTIMIZATION CODE BY OLESIGMUND %%%
%%% PRACTICA 5 - SEGUNDO CASO: APOYO EN TALÓN %%%
function topp5b(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
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
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
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
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(1,k)*dc(1,k);
end
end
dcn(j,i) = dcn(j,i)/(x(j,i)*sum);
end
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
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)];

```

### c. Pie apoyado del metatarso:

El código de la figura 4.2c es para definir el apoyo en el metatarso del pie.

```

88 - fixeddofs = [560:2*(nely+1):1260];
89 - alldofs = [1:2*(nely+1)*(nelx+1)];
90 - freedofs = setdiff(alldofs,fixeddofs);

```

Figura 4.2c: Código para el apoyo en el metatarso del pie.

## CODIGO COMPLETO

```

%%% A 99 LINE TOPOLOGY OPTIMIZATION CODE BY OLESIGMUND %%%
%%% PRACTICA 5 - TERCER CASO: APOYO EN EL METATARSO %%%
function topp5c(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./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
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);
= 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 = [560:2*(nely+1):1260];
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)];

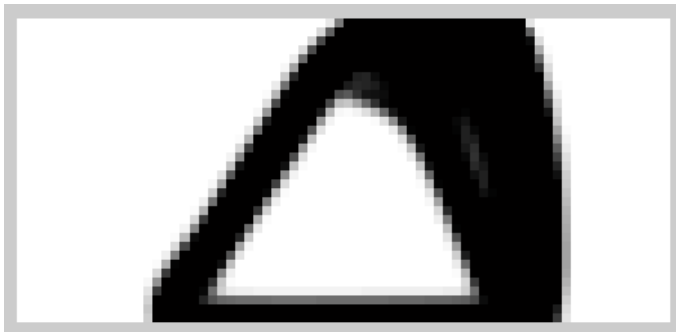
```

## 5) Resultados de la optimización

Para ver los resultados de la optimización en los tres casos debemos de asignar los siguientes parámetros:

nelx = 72 (Largo del espacio de diseño),  
nely = 34 (Ancho del espacio de diseño) y  
volfrac = 0.33, penal = 3.0, rmin = 1.5

**topp5a (72,34,0.33,3.0,1.5)**



**topp5b (72,34,0.33,3.0,1.5)**



**topp5c (72,34,0.33,3.0,1.5)**



## **6) Conclusiones**

### **Leonardo Daniel De León Fuentes 1991978**

En esta práctica se propuso la geometría de una prótesis de un pie y se realizó la optimización para reducir la masa y costos, aprendimos partes de las prótesis y los puntos donde se generan mayor cargas, donde se necesita una mayor rigidez y donde no es tan necesaria.

### **Fatima Montserrat Castro Nuñez 1991834**

Esta práctica es muy interesante ya que por fin podemos usar la optimización topológica que habíamos estado estudiando para otros casos, en un caso de biomecánica, que es el tema de nuestro interés. Me parece muy útil que hayamos estado trabajando con diferentes optimizaciones para al final poder implementarlo en el estudio de una prótesis de pie, ya que son de las prótesis que más se necesitan y es importante ver su estudio y además optimizar su topología. Me gustó mucho utilizar el código en matlab, ya que con pocas modificaciones hemos podido aplicarlo para diferentes estudios de diversos objetos. Definitivamente obtuvimos grandes conocimientos en este laboratorio que podemos aprovechar después.

### **Erick Eduardo Landa Gonzalez 1992037**

En esta práctica se me hizo algo curiosa por el hecho de hacer una optimización topológica ya que estábamos trabajando desde las anteriores prácticas. Esto nos ayudó a poder aplicarlo en algo más grande en este caso es la prótesis de un pie, hoy en día se ocupa mucho esta prótesis ya que es la que más se requiere y por medio de matlab pudimos resolverlo y aplicarlo en otros casos.

### **Brandon Geovanny Espinosa Alcocer 1938292**

En esta práctica pudimos observar la gran utilidad que tiene este software respecto a la optimización de diseños , en este caso nos enfocamos en prótesis para piernas ; observamos que en este software podemos ir desarrollando etapa por etapa los diseños que requiramos hacer . Después de poder utilizar este software , podemos tomarlo en cuenta para futuros diseños de cualquier estructura , además su fácil utilidad lo hace apto para la manipulación de diversos compañeros . El desarrollo de prótesis de diversas partes del cuerpo humano ha beneficiado grandemente la calidad de vida de las personas con falta de extremidades , totales o parciales , esto significa un gran avance tanto médico como científico y mecánico . La implementación de prótesis se viene utilizando desde hace muchas épocas , sin embargo no siempre en su forma óptima , es por ello que esta actividad nos muestra que simplemente es cuestión de cálculos , y los diseños se pueden definir de mejor manera , pensando primeramente en la mejora hacia los consumidores .

### **Referencias:**

<https://mediprax.mx/elementos-de-una-protesis-para-pierna-basicas-la-mejor-opcion/>