



Universidad Autónoma de Nuevo León
Facultad de Ingeniería Mecánica y Eléctrica



Laboratorio de Biomecánica

Práctica 5: optimizacion de la prótesis de un pie.

Docente: Dra. Yadira Moreno Vera

Dia y hora: miércoles N4 Brigada: 309

Integrantes del equipo:

Matriculas	Nombre	Carrera
1806409	Eduardo Antonio Flores Ramírez	IMTC
1884328	Daniel Isaac Zaragoza Soto	IMTC
1900466	Denisse García Espinoza	IMTC
1897305	Kevin Alberto Flores Martínez	IMTC
1991814	Pedro Hazael Uriegas Peña	IMTC

Semestre agosto-diciembre 2022.

PRACTICA #5

Optimización de una Prótesis de Pie

Contenido

Objetivo	3
Estado del Arte	3
Propuesta de diseño de la geometría, alcances y limitaciones	6
Pasos del desarrollo de la programación	7
Código del ejercicio 1	7
Código del ejercicio 2	9
Código del ejercicio 3	10
CONCLUSIONES.....	14
BIBLIOGRAFIA.....	14

Objetivo:

Proponer el diseño de una prótesis de pie el optimice su funcionamiento.

Estado del Arte

Los pies protésicos van desde los básicos (fijos), articulados (con movimiento en una o más direcciones) o de respuesta dinámica (acumulan y devuelven energía al caminar, dando una sensación de “empuje”, algo muy parecido a lo que hace el pie humano). Hoy en día, los pies protésicos pueden tener resortes en la zona correspondiente a los dedos y al talón con el fin de permitir un mayor movimiento del tobillo, la regulación del talón a diversas alturas y la absorción del impacto.

Pie No Articulado

Consiste en un pie sin articulación, de tobillo sólido y talón almohadillado también conocido como SACH por sus siglas en inglés (Solid Ankle Cushion Heel), compuesto por una quilla rígida de diversos materiales que constituye el tobillo y el empeine, una cuña de material viscoelástico que puede tener diferentes

densidades en el talón y un antepié elástico modelado para adquirir la forma de un pie.

La elasticidad del talón depende de las diferentes durezas de la cuña. Debido a la deformación del material elástico amortigua el impacto y disipa la energía del choque simulando cierta flexión plantar del tobillo en el apoyo del talón. Durante la fase de despegue, la flexibilidad del antepié protésico permite una adaptación al terreno ayudando al impulso del pie protésico. En la fase de apoyo simula un cierto grado de flexión plantar, produciendo una absorción de la energía del impacto que queda disipada. Durante la última fase del apoyo la flexibilidad de la punta del pie permite una adaptación de esta a la carga.

Pie articulado

Consta de una quilla rígida de diferentes materiales. Lleva una articulación de un eje a nivel de la articulación anatómica del tobillo, cuya movilidad es controlada por uno o dos topes de goma de distinta densidad. Todo va recubierto de una capa de tejido elástico moldeado con la forma del pie. Precisa de un sistema de adaptación que se une al resto de la prótesis.

En función y respuesta de la velocidad de la marcha y del peso corporal, los topes cambian la flexión plantar o dorsal del tobillo protésico para proporcionar el soporte adecuado. Si además lleva topes de goma laterales, permite cierto grado de inversión y eversión para adaptarse mejor ante las irregularidades del terreno.

La locomoción proviene del fenómeno físico conocido como movimiento. Así, el movimiento siempre significa un cambio de posición en el espacio. Es el movimiento que permite que el sujeto (ya sea una persona o una máquina) se desplace y, además de adquirir otra posición, cambie de lugar. Es una posibilidad que sólo tienen los seres vivos y algunas máquinas o aparatos creados por el ser humano que, de todas maneras, deben contar con algún método de propulsión como motores o energía.

El conocimiento de la locomoción humana normal es la base del tratamiento sistemático y del manejo de la marcha patológica, especialmente cuando se usan prótesis y ortesis.

El caminar o andar de una persona, se define como la repetición de una serie de movimientos simultáneos, que desplazan el cuerpo sobre una línea de progresión

deseada. Y al mismo tiempo mantienen una postura estable, soportando el peso corporal.

La movilidad libre de las articulaciones y el trabajo que desempeñan los músculos es importante para el éxito de esta tarea. Estos últimos deben actuar en el momento preciso y con la intensidad necesaria. La falta de ciertas acciones durante la marcha debe ser sustituida por otras, con el fin de mantener la estabilidad y la progresión deseada.

Ciclo de Marcha

El ciclo de la marcha comienza cuando el pie contacta con el suelo y termina con el siguiente contacto con el suelo del mismo pie. Los dos mayores componentes del ciclo de la marcha son: la fase de apoyo y la fase de balanceo (Figura 1). Una pierna está en fase de apoyo cuando está en contacto con el suelo y está en fase de balanceo cuando no contacta con el suelo.

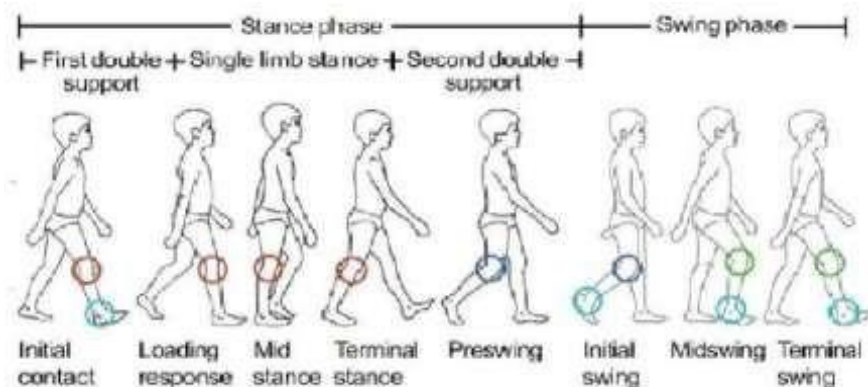


Figura 1. Fases en la Marcha Humana.

La longitud del paso completo es la distancia lineal entre los sucesivos puntos de contacto del talón del mismo pie. Longitud del paso es la distancia lineal en el plano de progresión entre los puntos de contacto de un pie y el otro pie (Figura 2).

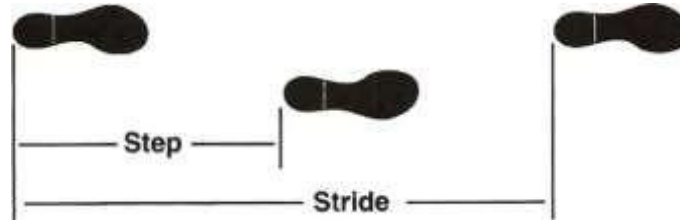
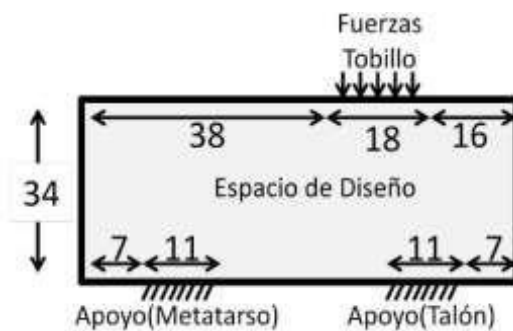


Figura 2. Longitud de Paso.

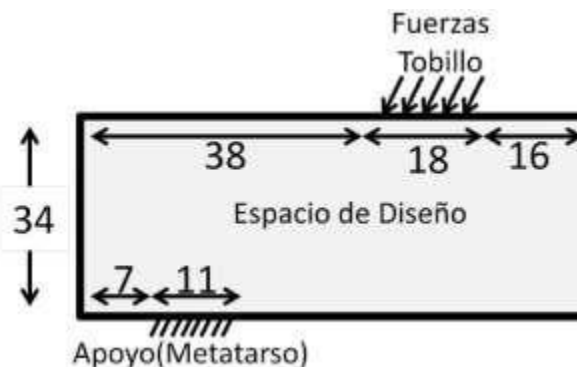
Propuesta de diseño de la geometría, alcances y limitaciones.

Para la realización de esta práctica se analizará el comportamiento de un solo pie dentro de las 3 fases de la marcha humana:

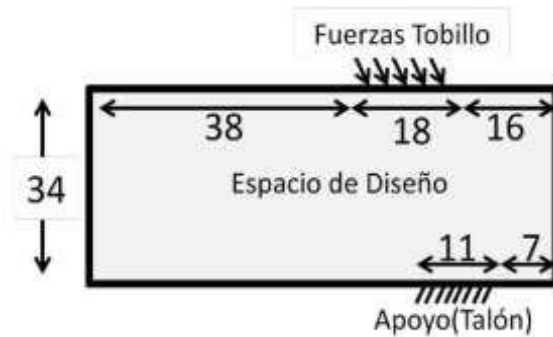
- Normal (El talón y área metatarsial son los apoyos, la fuerza se aplica sobre el tobillo con una fuerza de 500N)



- Despegue (El área metatarsial es el apoyo, la fuerza de 500N se aplica sobre el tobillo con un ángulo de 30°)



- Apoyo (El área del talón es el apoyo, la fuerza de 500N se aplica sobre el tobillo con un ángulo de 60°)



Pasos del desarrollo de la programación

Los 3 códigos con sus modificaciones realizadas.

Código del ejercicio 1

```

%%% A 99 LINE TOPOLOGY OPTIMIZATION CODE BY OLESIGMUND, OCTOBER 1999 %%%
function topp(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
% 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
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)];

```

Código del ejercicio 2

```

%%% A 99 LINE TOPOLOGY OPTIMIZATION CODE BY OLESIGMUND, OCTOBER 1999 %%%
function topp(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
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)];

```

Código del ejercicio 3

```

%%% A 99 LINE TOPOLOGY OPTIMIZATION CODE BY OLESIGMUND, OCTOBER 1999 %%%
function topp(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
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));

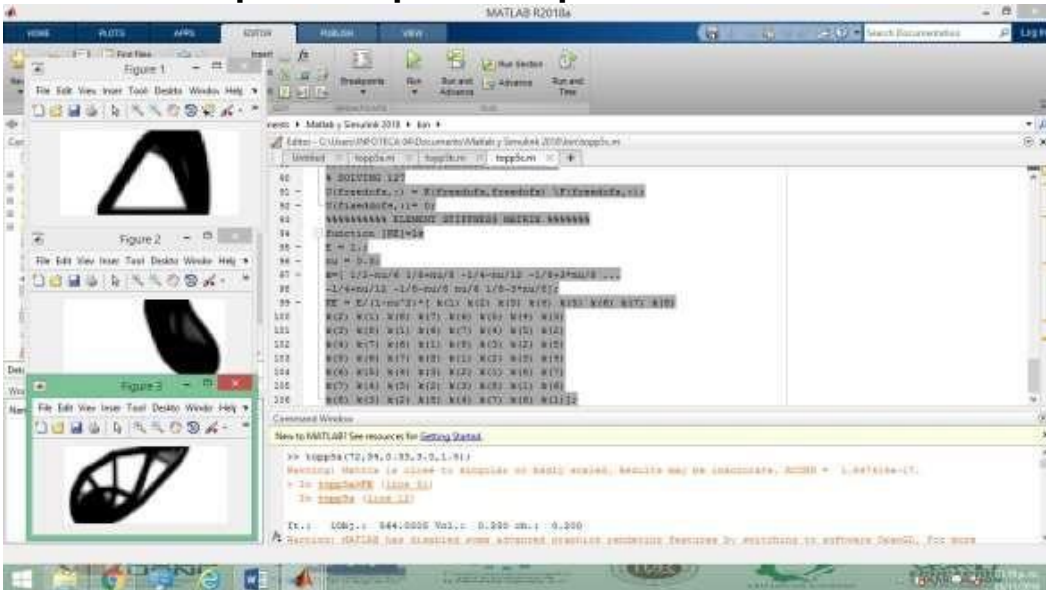
```

```

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 = [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)];

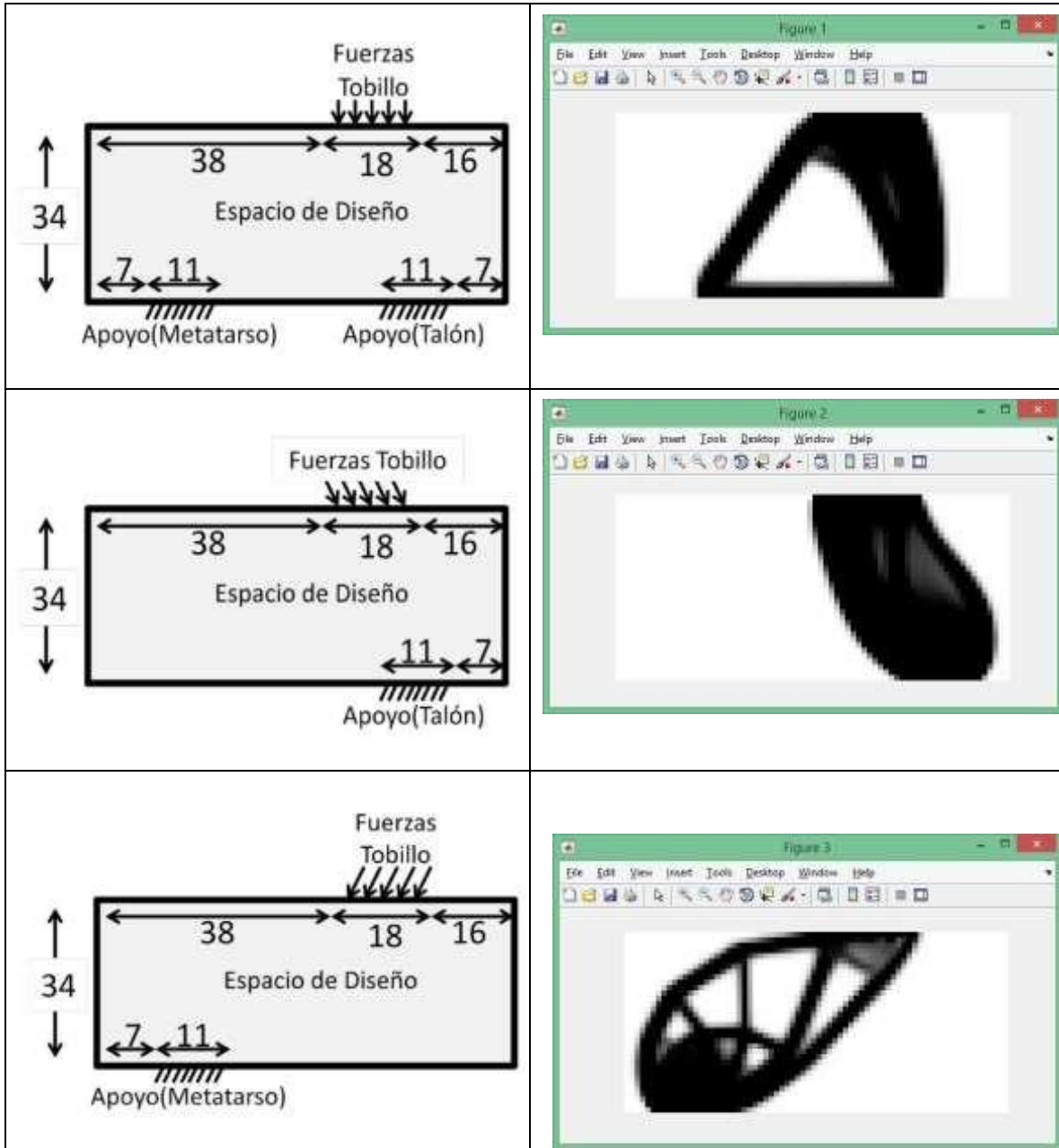
```

Captura de pantalla que muestra el desarrollo de la práctica.



Resultados de la optimización

A continuación, le mostraremos las simulaciones para los tres estados de las marchas.



CONCLUSIONES

Daniel isaac zaragoza soto 1884328: En esta práctica tuvo como objetivo proponer diseño el cual optimiza el funcionamiento de la prótesis de un pie. Sobre la marcha nos percatamos de las posiciones al realizar un movimiento simple. El uso de software llamado "Matlab" nos brinda las herramientas necesarias para realizar un código por el cual se va a ser un análisis de elemento finito y analizar nuestro diseño desde diferentes perspectivas.

Denisse García Espinoza 1900466: durante esta práctica se trabajó con un código con el cual se diseñó una prótesis de pie, sabemos el impacto que tienen estos elementos en la vida cotidiana para los que lo necesitan, por eso se analizaron varios casos para esta misma prótesis para observar mejor su estructura y funcionamiento. Gracias a matlab se puede realizar un mejor análisis de diferentes tipos de estructuras.

Kevin Alberto Flores Martínez 1897305: Para esta práctica nos propusieron optimizar el diseño de geometría de una prótesis de pie, introduciendo diferentes puntos de presión representantes de un pie normal como lo son el talón y el metatarso al código. Se realizaron tres ejercicios cada uno con diferentes puntos, ángulos y cantidad de presión. De nuevo el software de Matlab nos propuso un diseño basado en estos datos donde se pueden apreciar diferentes diseños que a la vez que ahorran recursos, son perfectamente capaces de realizar su trabajo.

Pedro Hazael Uriegas Peña 1991814: Con esta práctica se llevó al cabo la optimización de una prótesis de un pie, esto a través del código utilizado en prácticas pasada con el software Matlab, se analizó la estructura de la prótesis y se calcularon según las fuerzas de la figura, un reforzamiento en la protesis propuesta.

Eduardo Antonio Flores Ramírez 1806409: Con esta práctica se tuvo un mejor entendimiento sobre el movimiento del cuerpo humano, entender la locomoción ayuda a tener una base de conocimientos para el diseño de futuras prótesis. Además en esta práctica, se hizo uso de softwares para el diseño de estas, aplicándolo a temas de Biomecánica.

BIBLIOGRAFIA

- 99 Line Topology Optimization Code – O. Sigmund, Department of Solid Mechanics, Building 404, Technical University of Denmark, DK-2800 Lyngby, Denmark.
- 3D Natives (2017). La optimización topológica en la impresión 3D. Recuperado de <https://www.3dnatives.com/es/optimizacion-topologica10012017/>
- http://es.mathworks.com/products/matlab/index.html?s_tid=gn_loc_drophttp://mmc2.geofisica.unam.mx/cursos/tfs/TFSPresentaciones/Presentacion21Aaron.pdf
- <https://www.ecured.cu/Locomoción>

