

Práctica 5

Optimización de un prótesis de pie

1845056 Aurora Nahomy Martínez Pérez
1854324 Kevin Orlando Huerta Jaramillo
1896681 Jovanny Daniel Alvarado Ramírez
1909876 Fernando Herrera Garza
1991876 Bernardo Gil Villarreal

4 de noviembre de 2022

1. Objetivo

El estudiante deberá presentar una propuesta de análisis de formas y de la programación para la ejecución de la optimización de características de trabajo específicas que presenta las ventajas.

2. Introducción

En esta práctica lo que se realizará es la optimización de una prótesis de un pie, por lo cual se realizará una investigación sobre cómo es que funciona el movimiento que compone el caminar de una persona, para así poder desarrollar y poder observar los puntos de apoyo del pie y ver las fuerzas que se ejercen para así poder optimizar las superficie estudiada.

3. Marco Teórico

3.1. Locomoción

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 alguien método de propulsión como motores o energía. El conocimiento de la locomoción humana normal es la base del tratamiento sistémico 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.

3.1.1. Ciclo de la 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.

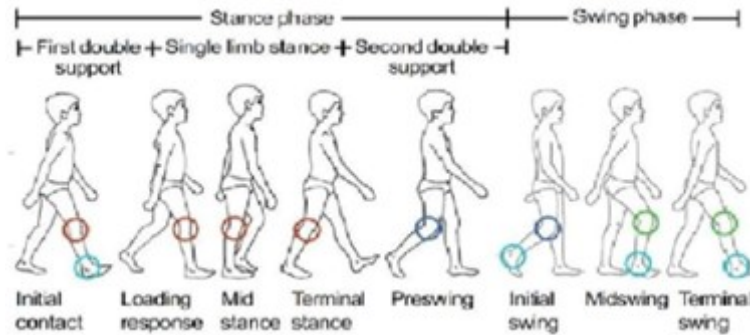


Figura 1: Fases en la marcha humana

La longitud del paso completo es la distancia lineal entre los sucesivos puntos de contacto del taloén 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).

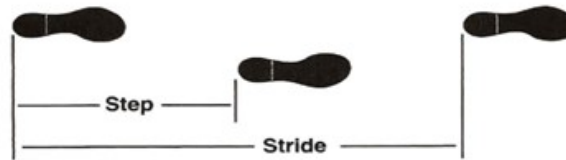


Figura 2: Longitud de paso

[1] [2] [3]

4. Metodología

4.1. Estado del arte

Dentro del área médica, los médicos y cirujanos están conscientes de los rápidos avances tecnológicos existentes y que se siguen desarrollando en el mundo. Debido a esto, contemplan adoptar nuevas técnicas para solucionar sus problemas clínicos. De cualquier modo, el desarrollo de soluciones requiere individuos especializados para aplicar los conocimientos y tecnología existente de manera efectiva y apropiada. Sin embargo, es prácticamente imposible para un individuo dominar todos los conocimientos intrínsecos en el área de la Biomecánica. Donde son claramente evidentes las diferencias existentes entre las materias involucradas, como son:

- Cirugía ortopédica.
- Rehabilitación.
- Simulación numérica.

- Fisiología.
- Estática.
- Dinámica.
- Diseño Mecánico.
- Manufactura.

Huiskes [1989] señala que el diseño de este tipo de prótesis considera los siguientes factores para corregir el ajuste por interferencia que se realiza en los vástagos no cementados y estos son:

- La forma geométrica.
- El recubrimiento del que se encuentra constituido.
- Los micro movimientos.
- La transferencia de las cargas
- Los esfuerzos en el hueso.

4.2. Nombre y definición de la forma Geométrica

Se realizará la optimización de la prótesis de un pie, dentro de sus etapas de movimiento que componen la marcha que tenemos los humanos, que sería en una posición normal, en el despegue y el apoyo.

4.3. Desarrollo

Fase: Normal

En esta fase, el talón y el área metatarsal son los que funcionan como apoyos, y la fuerza se aplica sobre lo que es el tobillo. Se ve representado en la siguiente figura:

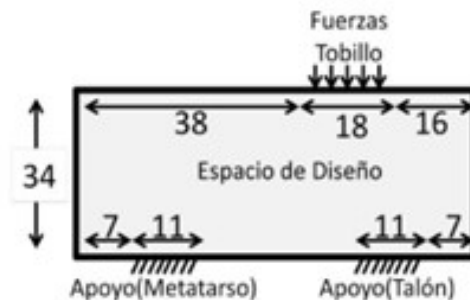


Figura 3: Fase normal

Como vemos, se colocará una fuerza de 500 N sobre la parte del tobillo.

Fase: Despegue

Aquí, el apoyo es únicamente el área metatarsal y se sigue aplicando la fuerza en el tobillo, y no tomamos en cuenta la parte del talón.

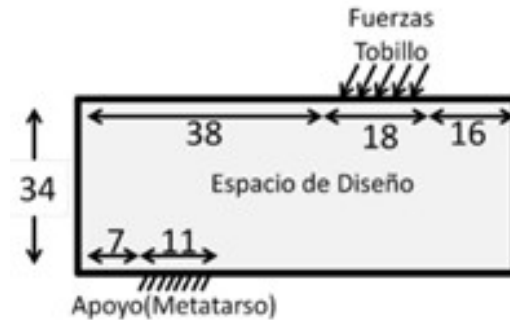


Figura 4: Fase de despegue

Fase: Apoyo

El área del talón funciona como el apoyo, y la fuerza se aplica sobre el tobillo con un ángulo deseado de 60°.

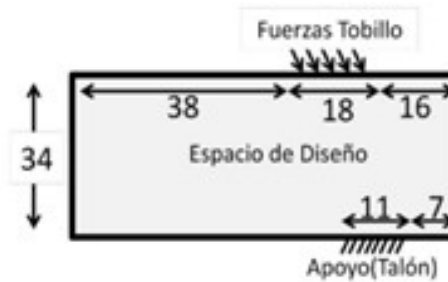


Figura 5: Fase de apoyo

4.4. Código final implementado

Como vemos, se propusieron 3 ejercicios distintos y por ende, cada uno llevará su código modificado propiamente dependiendo de donde se ubiquen los apoyos y las fuerzas. Primero realizaremos la optimización de la fase normal.

■ Fase: Normal

```
%%% A 99 LINE TOPOLOGY OPTIMIZATION CODE BY OLESIGMUND, OCTOBER 1999 %%%
function Practica5Ej1(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(1e6);
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);
            end
            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

```

```

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)];

```

■ Fase: Despliegue

```

%%% A 99 LINE TOPOLOGY OPTIMIZATION CODE BY OLESIGMUND, OCTOBER 1999 %%%
function Practica5Ej2(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(1e6);
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);
            end
            dcn(j,i) = dcn(j,i) + max(0,fac)*x(l,k)*dc(l,k);
        end
    end
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); 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)];

```

■ Fase: Apoyo

```

%%% A 99 LINE TOPOLOGY OPTIMIZATION CODE BY OLESIGMUND, OCTOBER 1999 %%%
function Practica5Ej3(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(1e6);
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)];

```

5. Resultado

Fase 1 Normal

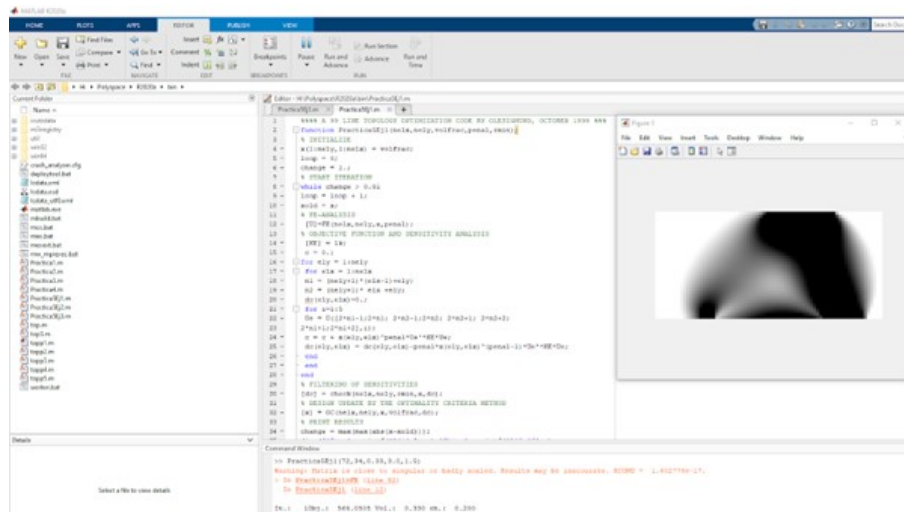


Figura 6: Inicio

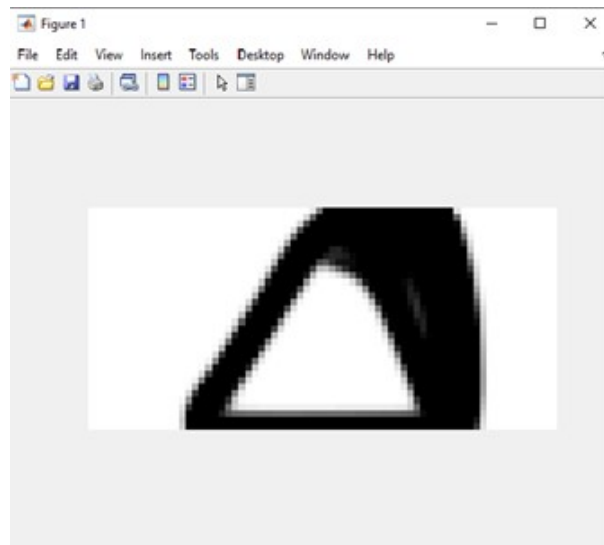


Figura 7: Final

Fase 2 Despegue

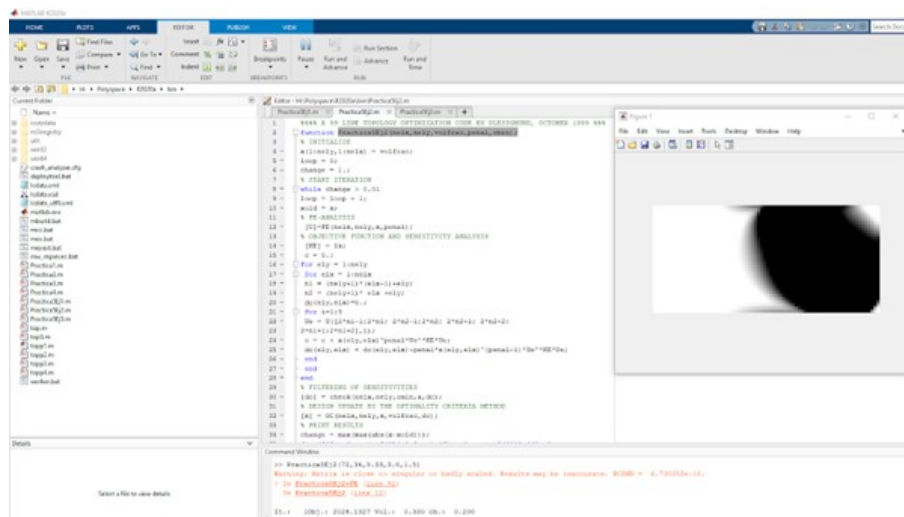


Figura 8: Inicio

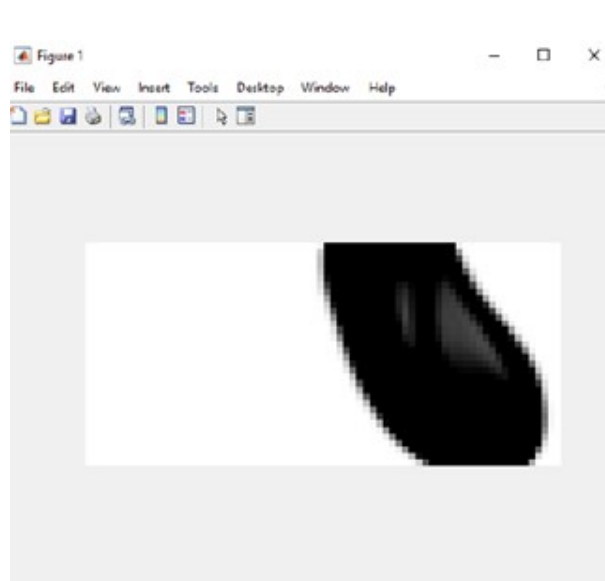


Figura 9: Final

Fase 3 Apoyo

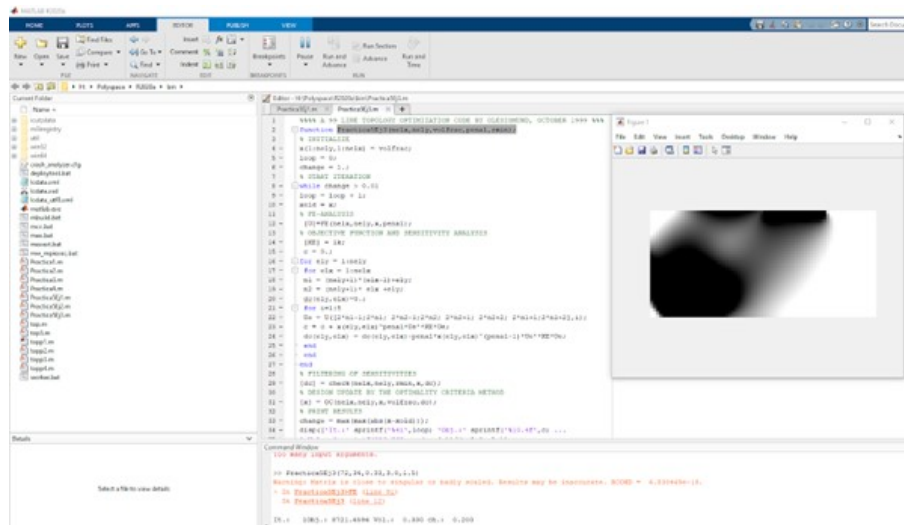


Figura 10: Inicio

Esta etapa no se logró finalizar en MATLAB debido a que estaba tomando mucho tiempo, y constantemente se congelaba el programa, por lo cual no se logró obtener la optimización completa de esta fase.

6. Conclusiones

■ Bernardo Gil Villarreal 1991876

A lo largo de este reporte pudimos observar y nos percatamos que se puede usar Matlab para generar un análisis de elemento finito para objetos de ámbito simple o más profesional y que se pueden usar para diferentes casos, además de generar un buen soporte que nos ayudara mucho en este caso. A partir de lo que aprendimos, nos damos cuenta que los softwares 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. Así mismo, es importante destacar que la movilidad libre de las articulaciones y el trabajo que desempeñan los músculos es importante para lograr el mejor diseño posible. La falta de ciertas acciones, debe ser sustituida por otras, con el fin de mantener la estabilidad y la progresión. El uso de estas herramientas nos permite analizar nuestro diseño desde diferentes perspectivas.

■ Jovanny Daniel Alvarado Ramírez 1896681

Se trabajó de nuevo en base a una propuesta que contenía diversas especificaciones que cumplir en el nuevo proceso de optimización que en este caso consistía en la optimización de una prótesis de pie. Obtuvimos los resultados esperados y puedo concluir concretamente en que es posible generar nuevas y mejores geometrías de elementos físicos estructurales de mayor importancia como lo son estas prótesis. De igual manera, menciono que dados los recursos a nuestro alcance la optimización en Matlab es buena y algo tardada; aunque el proceso fue sencillo dado que solo fue necesario modificar un poco el código de programación que ya teníamos.

■ Aurora Nahomy Martínez Pérez 1845056

En esta práctica se pudo realizar la optimización de pie donde se tuvo que analizar el movimiento de la pierna mediante el uso de matlab la optimización en estos casos debe realizarse en distintas etapas dependiendo del mismo movimiento que realice la extremidad.

- Kevin Orlando Huerta Jaramillo 1854324

Para poder realizar la optimización de la prótesis se tiene que analizar el movimiento de la pierna al momento de caminar para poder determinar las fuerzas que se ejercen al dar el paso y poder ver cuales son las partes de mayor esfuerzo, para después mediante el código que se tienen en Matlab para obtener la optimización.

- Fernando Herrera Garza 1909876

Como se logró observar, para el caso de las prótesis del cuerpo, se necesitan de varias fases o etapas de optimización, dependiendo de los movimientos que realicen estas partes del cuerpo, y es necesario tener en consideración todas estas fases, para un mejor resultado. Al igual que con las anteriores prácticas, cada fase tomó un largo periodo de procesado en MATLAB, y desafortunadamente no se logró obtener la optimización de la última fase, pero sería solamente de esperar a que MATLAB logre realizar el proceso y ver su resultado final.

Referencias

- [1] Byron J Lambert, Laura E Leigh Perkins, and Richard J Rapoza. Endoprótesis coronarias bioabsorbibles. *Revista Española de Cardiología Suplementos*, 13:29–39, 2013.
- [2] Adrià Casinos Pardos. *La locomoción*. LOS LIBROS DE LA CATARATA, 2018.
- [3] Lisandro Puglisi and Héctor Moreno. Prótesis robóticas. *Revista del Departamento de Automática e Informática Industrial*, pages 1–2, 2006.