Universidad Autónoma de Nuevo León

Facultad de Ingeniería Mecánica y Eléctrica

LABORATORIO DE BIOMECANICA

PRÁCTICA 5

Optimización de una prótesis de pie

Docente: Dra. Yadira Moreno Vera

Día: Martes

Hora: V2

Brigada: 214 ***Equipo***: 2

Nombre	Matricula	Carrera
María de los Ángeles Puente Peña	1905238	IMTC
Viviana Nathalie Tienda Téllez	1919910	IMTC
Mauricio Guerrero Hernández	1905306	IMTC
Karla Gabriela Torres García	1910427	IMTC
Brayan Uriel Grimaldo Salazar	1908530	IMTC
Alina Martínez Escobedo	1912818	IMTC

Fecha de entrega: 08-noviembre-2022

OBJETIVO

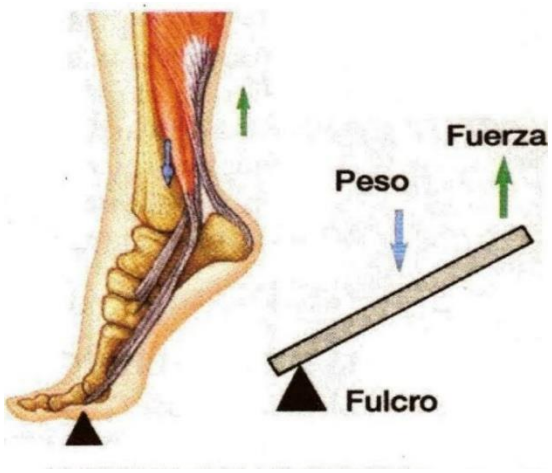
El estudiante deberá presentar una propuesta de análisis de formas y de la programación para la ejecución de la optimización (descripción funcional) de características de trabajo específicas que presenta las ventajas.

INTRODUCCION

En el siguiente trabajo presentaremos una propuesta de la optimización de una prótesis de pie utilizando la programación de 99 líneas que se utilizó en la optimización topológica, aquí se modificaran ciertos parámetros y al código para poder presentar la geometría necesaria.
Al tener que analizar una prótesis de pie debemos de crear distintas simulaciones, para esto realizamos 3 distintos casos que son las formas esenciales la primera son en la posición normal del pie en reposo, la segunda es cuando se hace un despegue del pie y en la tercera es cuando se hace el apoyo en el suelo, por lo que a partir de estos distintos casos lograremos tener desde distintas perspectivas la optimización de la prótesis de pie.

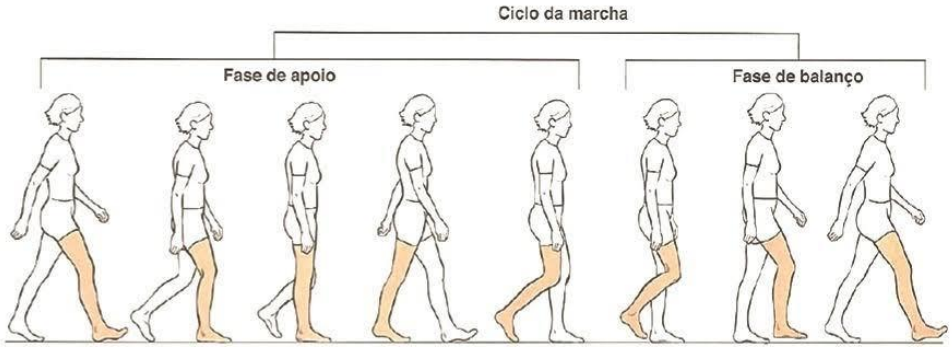
NOMBRE Y DEFINICIÓN DE LA FORMA GEOMETRICA

Esta práctica consta de realizar el diseño y análisis de las fuerzas que se aplican en un pie, utilizando la programación que Matlab ofrece y la optimización analizada en la práctica anterior.



En esta ocasión el pie constara de un rectángulo simple (el cual se presentara mas adelante), al cual se le aplicaran distintas fuerzas y apoyos para ver el cambio en la optimización.
El primer caso es el pie en estado normal, donde se aplica una fuerza en el tobillo (parte superior) y 2 apoyos, que son en el metatarso y talón (parte inferior).
En el segundo caso se aplica una fuerza en el tobillo y un solo apoyo en el metatarso.
El tercer caso es igual al segundo con la diferencia que el apoyo es únicamente en el talón.

Esto tiene relación con algunos pasos del proceso de marcha humana, ya que este es un proceso de locomoción en el cual el cuerpo humano, en posición erecta generalmente, se mueve hacia delante, siendo su peso soportado alternativamente por ambos miembros inferiores. Se caracteriza a la vez que se distingue de la carrera por el contacto permanente del individuo con el suelo a través de al menos uno de sus pies.



ESTADO DEL ARTE

Los pies son una de las extremidades más importantes y básicas no sólo para poder caminar, sino para simplemente poder sostenerse parado y tener el equilibrio necesario para todo tipo de actividades. Las prótesis de pie son dispositivos médicos empleados para ayudar a pacientes que han sufrido la amputación total o parcial de su pie a causa de un accidente, enfermedad o agenesia. Estas prótesis sustituyen de forma artificial la parte faltante o la totalidad del pie, imitando las formas y dimensiones de la sección original.

¿Cómo funciona una prótesis de pie?
Una prótesis de pie, también conocidas como pie protésico, son un dispositivo con finalidades funcionales y estéticas, diseñado justo a la medida de cada paciente para sustituir la parte faltante de la extremidad a causa de una amputación total o parcial del pie.
Esta prótesis tiene la función de asemejar la presencia de la extremidad faltante, ademas de brindar el soporte y movimiento necesario para que el paciente pueda sostenerse de pie, caminar e inclusive correr y desempeñar distintos trabajos o deportes de alto rendimiento

Todas las prótesis de pie cuentan con una serie de mecanismos diseñados para poder adaptarse al muñón de forma adecuada y firme, además de brindar el soporte y movimiento necesario para que el paciente pueda sostener de pie, caminar e inclusive correr y desempeñar labores o deportes de alto rendimiento.

Tipos de prótesis

Con el pasar de los años y el avance tecnológico la mayor parte de las prótesis contienen rasgos más humanos, obteniendo múltiples ventajas para proporcionar una segunda oportunidad a personas que han perdido una de sus extremidades. Algunos tipos de prótesis que existen son las siguientes:

- Prótesis Biónica
- Prótesis Sach
- The Walking MP
- Prótesis Pasiva
- Prótesis DER
- Prótesis Ankle-Foot
- Prótesis CERV
- Prótesis with Control of Plantar flexion and Inversion-Eversion Torque
- Prótesis de Knee

Figura 1: Tipos de prótesis para pie.

Partes de las prótesis

Dependiendo del tipo de amputación, la prótesis podrá estar compuesta de diversos elementos, como lo pueden ser:

- Socket o encaje con el muñón.
- Plantillas.
- Almohadillas.
- Estabilizadores para el talón.
- Elementos estéticos.

De igual manera, existen prótesis mucho más avanzadas que pueden tener dispositivos y elementos robóticos, mecánicos, mioeléctricos y biomecánicos para otorgar movimientos especiales en caso de ser necesario.

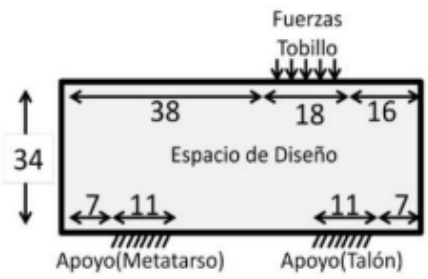
Materiales de las prótesis de pie

El pie prostético puede ser fabricado de uno o más materiales, los cuales son seleccionados para brindar el soporte, agarre y diseño necesario para suplantar la parte faltante del cuerpo humano. Por lo general de utiliza:

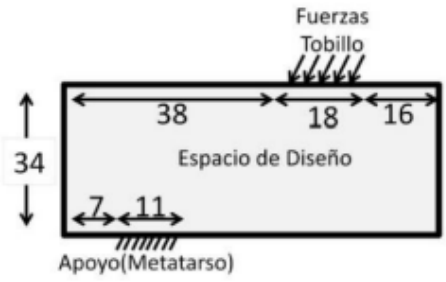
- Silicona médica.
- Plantillas de carbono.
- Rellenos de materiales elásticos.
- Aluminio.
- Titanio.
- Acero inoxidable.
- Plásticos.

PROPUESTA DE DISEÑO DE LA GEOMETRIA, ALCANCES Y LIMITACIONES

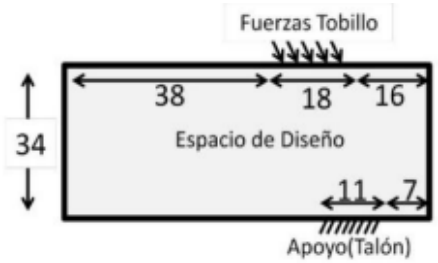
Para la optimización de una prótesis de pie, será necesario el realizar distintas simulaciones debido a que el pie puede presentar distintos estados en el que se ejercen los esfuerzos, por lo que será necesario el realizar 3 simulaciones distintas correspondiente a cada uno de los casos, de los cuales el "Caso 1" con el pie en una posición normal, en el que el talón y el área metatarsial son los apoyos, la fuerza que se aplica sobre el tobillo es de 500N; posteriormente el "Caso2" en el que se produce el despegue del pie, donde el área metatarsial es el apoyo, aplicando una fuerza de 500N con un ángulo de 30°; y por el ultimo tenemos el "Caso 3" en cual se trata del apoyo del pie después del despegue, en donde el área del talón es el apoyo, aplicando una fuerza de 500N sobre el tobillo con un ángulo de 60°.



-Caso 1



-Caso 2



-Caso 3

Alcances y Limitaciones:

La movilidad libre de las articulaciones y el trabajo que desempeñan los músculos es importante para el correcto funcionamiento del pie. Es por eso por lo que las articulaciones y los músculos deben actuar en el momento preciso y con la intensidad necesaria.

La falta de ciertas acciones durante la marcha debe ser sustituidas por otras, todo esto para mantener la estabilidad y la progresión deseada.

En este análisis nuestros alcances serán poder optimizar la forma teniendo en cuenta las fuerzas que se aplican en el pie, con las limitaciones de que las faces de la marcha humana en las que se involucran los momentos realizados por la rodilla no serán tomadas en cuenta.

PASOS DEL DESARROLLO DE PROGRAMACION

Para el desarrollo de la programación de esta práctica se usó el mismo código de optimización topológica de 99 líneas al igual que la práctica 1, por lo que a partir de este código únicamente fue necesario el establecer distintos parámetros para que fuese una estructura distinta al momento de ejecutar el programa, y de igual manera se realizaron ciertas modificaciones al código para que este pueda asemejarse a la geometría propuesta para el diseño de la optimización de la prótesis de pie. Dada las geometrías propuestas para los distintos casos, se optó por la siguiente topología:
top51(72,34,0.33,3.0,1.5)
Una vez teniendo los parámetros de nuestra figura geométrica base a la cual se le realizara la optimización, lo que se prosiguió a realizar fue el cambiar ciertas líneas del código para que cumpliese con lo deseado en la práctica:

Casos 1,2 y 3

Para los tres casos el código de Matlab se construye como un código de optimización de topología estándar. El programa principal es llamado desde el indicador de Matlab por la línea

```
2    function top52(nelx,nely,volfrac,penal,rmin);
```

donde nelx y nely son el número de elementos en el direcciones horizontal y vertical, respectivamente, volfrac es la fracción de volumen, penal es el poder de penalización y rmin es el tamaño del filtro (dividido por el tamaño del elemento). Otras variables, así como las condiciones de contorno se definen en el propio código de Matlab y se puede editar si es necesario. Para cada iteración en el ciclo de optimización de topología, el código genera una imagen de la distribución de densidad actual.

El programa principal comienza distribuyendo el uniformemente en el dominio del diseño. Después de algunas otras inicializaciones, el ciclo principal comienza con una llamada a la subrutina de elementos finitos que devuelve el vector de desplazamiento U.

```
11      [U]=FE(nelx,nely,x,penal);
```

Dado que la matriz de rigidez del elemento para el material sólido es el mismo para todos los elementos, la subrutina de matriz de rigidez del elemento se llama solo una vez.

```
13      [KE] = lk;
```

Después de esto, un ciclo sobre todos los elementos determina la función objetivo y las sensibilidades. Las variables n1 y n2 denotan arriba a la izquierda y a la derecha
números de nodo de elementos en números de nodo globales y son
utilizado para extraer el vector de desplazamiento del elemento Ue de
el vector de desplazamiento global U.
El cumplimiento actual, así como otros parámetros se imprimen y se traza la distribución de densidad resultante. El bucle principal se termina.

```
34      % PLOT DENSITIES
35      colormap(gray); imagesc(-x); axis equal; axis tight; axis off;pause(1e-6);
36    end
```

El optimizador encuentra las variables de diseño actualizadas sabiendo que el volumen material es una función monótonamente decreciente del multiplicador de Lagrange (retraso), el valor del Multiplicador lagrangiano que satisface la restricción de volumen se puede encontrar mediante un algoritmo de bisección. El intervalo que limita el multiplicador lagrangiano se divide repetidamente por la mitad hasta que su tamaño es menos que los criterios de convergencia

```
37    %%%%%%%%%% OPTIMALITY CRITERIA UPDATE %%%%%%%%%
38    function [xnew]=OC(nelx,nely,x,volfrac,dc,passive)
39      l1 = 0; l2 = 100000; move = 0.2;
40      while (l2-l1 > 1e-4)
41      lmid = 0.5*(l2+l1);
42      xnew = max(0.001,max(x-move,min(1.,min(x+move,x.*sqrt(-dc./lmid)))));
43      if sum(sum(xnew)) - volfrac*nelx*nely > 0;
44      l1 = lmid;
45      else
46      l2 = lmid;
47      end
48      end
```

Filtrado de independencia de malla
Representan la implementación de Matlab para el filtrado de independencia de malla hay que tomar en cuenta que no todos los elementos en el dominio de diseño son buscados para encontrar los elementos que se encuentran dentro el radio rmin pero solo aquellos dentro de

un cuadrado con lado longitudes dos veces redondas (rmin) alrededor de la considerada elemento. Seleccionando rmin menos de uno en la llamada del
rutina, las sensibilidades filtradas serán iguales a las sensibilidades originales haciendo que el filtro esté inactivo.

```
49      %%%%%%%%%% MESH-INDEPENDENCY FILTER %%%%%%%%%%
50      function [dcn]=check(nelx,nely,rmin,x,dc)
51      dcn=zeros(nely,nelx);
52      for i = 1:nelx
53      for j = 1:nely
54      sum=0.0;
55      for k = max(i-round(rmin),1): min(i+round(rmin),nelx)
56      for l = max(j-round(rmin),1): min(j+round(rmin), nely)
57      fac = rmin-sqrt((i-k)^2+(j-l)^2);
58      sum = sum+max(0,fac);
59      dcn(j,i) = dcn(j,i) + max(0,fac)*x(l,k)*dc(l,k);
60      end
61      end
62      dcn(j,i) = dcn(j,i)/(x(j,i)*sum);
63      end
64      end
```

Caso I. Código de elementos finitos (líneas 66 a 103)

El código de elementos finitos en el apartado de FE. ANALYSIS.

La matriz de rigidez global está formada por un bucle sobre todos los elementos, las variables n1 y n2 indican la parte superior izquierda y derecha números de nodo de elementos en números de nodo globales y son se utiliza para insertar la matriz de rigidez del elemento a la derecha lugares en la matriz de rigidez global.

La matriz de rigidez del elemento se calcula en las líneas 91.

```
65      %%%%%%%%% FE-ANALYSIS %%%%%%%%%%%%
66      function [U]=FE(nelx,nely,x,penal)
67      [KE] = lk;
68      K = sparse(2*(nelx+1)*(nely+1), 2*(nelx+1)*(nely+1));
69      F = sparse(2*(nely+1)*(nelx+1),5); U = sparse(2*(nely+1)*(nelx+1),5);
70      for ely = 1:nely
71      for elx = 1:nelx
72      n1 = (nely+1)*(elx-1)+ely;
73      n2 = (nely+1)* elx +ely;
74      edof = [2*n1-1; 2*n1; 2*n2-1; 2*n2; 2*n2+1; 2*n2+2;2*n1+1; 2*n1+2];
75      K(edof,edof) = K(edof,edof) + x(ely,elx)^penal*KE;
76      end
77      end
78      % DEFINE LOADSAND SUPPORTS(HALF MBB-BEAM)
79      F(3222,1) = -1;
80      F(3782,2) = -1;
81      F(2662,3) = -1;
82      F(2942,4) = -1;
83      F(3502,5) = -1;
84      fixeddofs = [3920:2*(nely+1):4620];
85      alldofs = [1:2*(nely+1)*(nelx+1)];
86      freedofs = setdiff(alldofs,fixeddofs);
87      % SOLVING
88      U(freedofs,:) = K(freedofs,freedofs) \F(freedofs,:);
89      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)];
```

Caso II. Código de elementos finitos

```
%%%%%%%%%% 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
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)];
```

Caso III. Código de elementos finitos

```
%%%%%%%%%% 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
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)];
```

RESULTADOS DE LA OPTIMIZACION

Caso1:

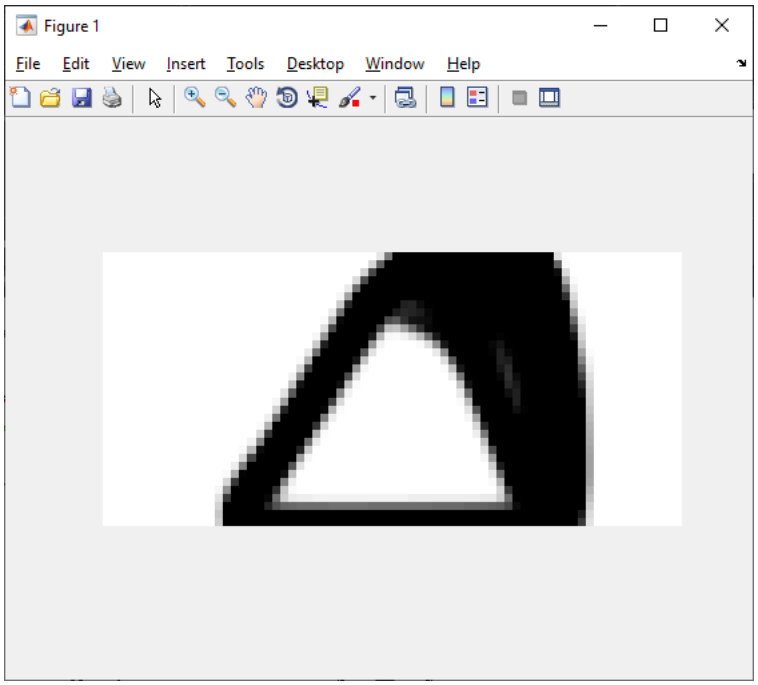
```
%%%% BIOMECANICA PROTESIS PIE E2 B214 %%%
function top51(nelx,nely,volfrac,penal,rmin);
% INITIALIZE
x(1:nely,1:nelx) = volfrac;
loop = 0; change = 1.;
% START ITERATION
while change > 0.01
loop = loop + 1;
xold = x;
% FE-ANALYSIS
[U]=FE(nelx,nely,x,penal);
% OBJECTIVE FUNCTION AND SENSITIVITY ANALYSIS
[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,passive)
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
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 para el Caso 1 con el pie en poyo Normal

Resultados de la optimización:



-Resultados de la optimización del Caso 1 con el pie en poyo Normal

Caso 2:

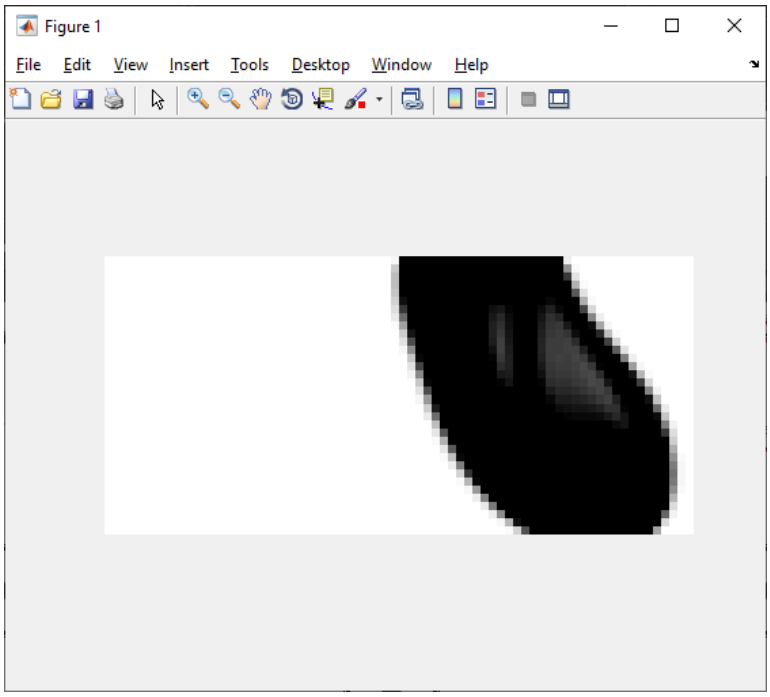
```
%%%% BIOMECANICA PROTESIS PIE E2 B214 %%%
function top52(nelx,nely,volfrac,penal,rmin);
% INITIALIZE
x(1:nely,1:nelx) = volfrac;
loop = 0; change = 1.;
% START ITERATION
while change > 0.01
loop = loop + 1;
xold = x;
% FE-ANALYSIS
[U]=FE(nelx,nely,x,penal);
% OBJECTIVE FUNCTION AND SENSITIVITY ANALYSIS
[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,passive)
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
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 para el Caso 2 con el despegue del pie

Resultados de la optimización:



-Resultados de la optimización del Caso 2 con el despegue del pie

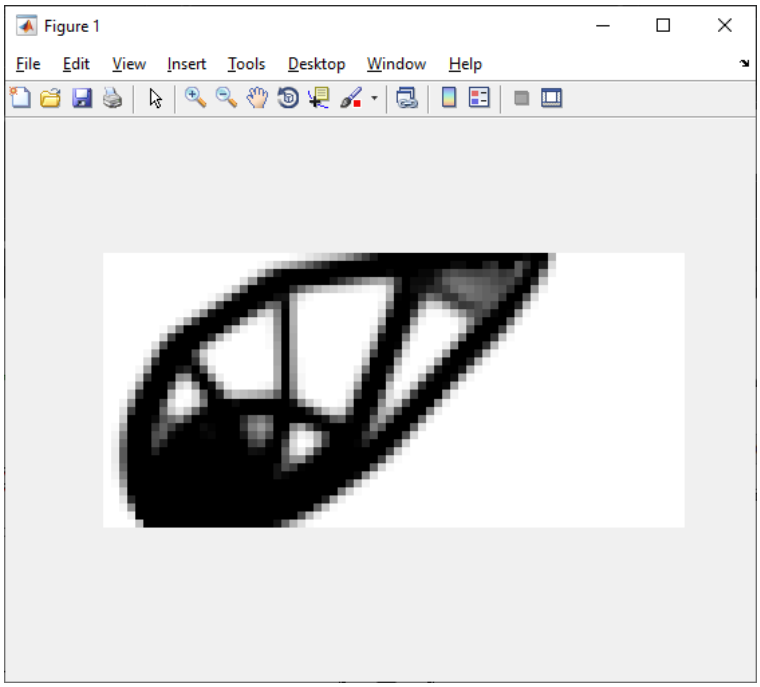
Caso 3:

```
%%%% BIOMECANICA PROTESIS PIE E2 B214 %%%
function top53(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,passive)
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
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 para el Caso 3 con el apoyo del pie

Resultados de la optimización:



-Resultados de la optimización del Caso 3 con el apoyo del pie

CONCLUSIONES

María de los Ángeles Puente Peña:

Para la realización de esta práctica volvimos a hacer uso de la optimización topológica, gracias a la cual cumplimos con el objetivo de presentar una propuesta de análisis de formas y de la programación para la ejecución de la optimización de características de trabajo específicas de las fuerzas que se aplican en diferentes momentos en un pie. Pudimos observar 3 momentos diferentes, donde la fuerza se aplicaba dependiendo la posición, ya fuera de descanso, apoyándose con la punta del pie o con el talón. También con esto podemos observar cómo cada parte del programa se vuelve más complejo debido a que es una estructura cada vez más compleja de analizar y por consecuencia también toma más tiempo.

Mauricio Guerrero Hernández:

En base a esta práctica de laboratorio de biomecánica, pudimos observar el comportamiento en este caso de una prótesis y de pie, pudiendo así notar el como las distintas cargas o las partes del cuerpo, se es necesario un tipo de prótesis, tal es el caso que tuvimos distintos puntos en el cual abordamos el problema y las distintas optimizaciones obtenidas. De igual manera se pudieron reforzar los conocimientos acerca de la optimización topológica en

Matlab por medio de la programación, mas sin embargo ya aplicado a el área médica el cual nos será de mucha ayuda en dado caso que nos queramos enfocar en dicha rama de nuestra carrera.

Brayan Uriel Grimaldo Salazar:

En conclusión, a esta práctica, logramos el objetivo planteado al inicio el cual era presentar una propuesta de análisis de forma y la correcta programación para la ejecución de la optimización. Se logró observar el comportamiento de una prótesis de pie, la cual se observó las distintas partes del cuerpo, y esto por medio de tres optimizaciones realizadas en el Software Matlab que se llevaron a cabo las optimizaciones topológicas gracias a ello podemos notar como afecta una prótesis y como poder analizarla para ver como afectará en un futuro

Karla Gabriela Torres García:

En conclusión, pienso que es importante conocer un poco más acerca de las prótesis y poder observar y/o analizar cómo es que puede ser optimizada una prótesis para pie en el software de Matlab.

Se requirió conocimiento de prácticas anteriores, puesto que seguimos enfocados en el uso de la optimización topológica y realmente opino que realizar este tipo de optimizaciones son útiles para observar cómo podríamos mejorar una pieza, en este caso una prótesis, reduciendo el material, pero sin perder la resistencia que debería de tener.

Conocer más acerca de que tipos de prótesis existen o materiales que pueden utilizar es interesante y creo que la ingeniería se mezcla perfectamente bien con la medicina y de ahí se pueden crear mejoras para personas que lo necesiten.

Viviana Nathalie Tienda Tellez:

En esta práctica, al igual que en lo que hemos trabajado durante el semestre, realizamos un análisis de optimización topológica en este caso de una prótesis de pie, para realizarla establecimos las fuerzas involucradas al estar caminando, hay fuerzas en el tobillo, y en el metatarso y talón que funcionan como apoyos. Los análisis de optimización topológica son de gran utilidad en la industria ya que significan primeramente como su nombre lo dice una optimización en la forma de la estructura así como de su funcionamiento, de igual manera aportan un gran ahorro ya que se utilizaría menos materia prima, sin embargo en la mayoría de los casos las formas resultantes tienen una topología difícil de manufacturar por lo que hay ocasiones en las que el resultado de la optimización después es usado como inspiración para el diseño pero facilitando su maquinado.

Alina Martínez Escobedo:

Gracias a esta práctica de laboratorio de biomecánica pudimos analizar una prótesis de pie y cómo son las cargas en ella. Para lograr el objetivo de está práctica realizamos 3 distintas

optimizaciones con el programa de 99 líneas y se logró observar dichas optimizaciones al utilizar el software Matlab. Con este trabajo pudimos notar como afecta una prótesis y cómo podemos analizarla para ver cómo afectará.

REFERENCIAS

- DAVID, J. (2016, September 7). Tipos y mecanismos de prótesis de pies. Retrieved November 7, 2022, from Monografias.com website: https://www.monografias.com/docs110/protesis-pie/protesis-pie
- miprotesis.mx. (2022, May 6). ▷ Prótesis de Pie en México - Tipos y Precios - Miprotesis.mx. Retrieved November 7, 2022, from Especialistas en Prótesis website: https://miprotesis.mx/protesis-de-pie/#Protesis_de_Pie_%C2%BFQue_son_y_como_funcionan
- LTF Erick Lemus. (2021, August 15). ▷ Prótesis de Pie - Tipos, Funciones y Precios - Conoce más aquí. Retrieved November 7, 2022, from Mi Protesis de Pierna website: https://miprotesisdepierna.mx/protesis-de-pie/