UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN

FACULTAD DE INGENIERÍA MECÁNICA Y ELÉCTRICA

Agosto - Diciembre 2022

Unidad de Aprendizaje: Laboratorio de Biomecánica

Practica No. 5

Nombre	Matricula
Gabryiel Bailon Avila	1869828
Carlos Eduardo Rivera López	1897545
Víctor Adrián Higuera Vázquez	1876474
Andrés Anaya Hernández	1914471
César Armando Luna Zapata	1844920

Hora: Martes V2

San Nicolás de los Garza, Nuevo León fecha: 06-septiembre-2022

Estado del arte.

Para encontrar las primeras piezas protésicas que se conocen, debemos remontarnos a los años 950 y 750 a.C. en el antiguo Egipto. Estas piezas, encontradas en una momia enterrada en Egipto, estaban hechas de madera y cuero y simulaban un dedo gordo del pie. Gracias a esta invención, los egipcios, además de mejorar la estética del pie, podían caminar con normalidad y sin perder el equilibrio.

El siguiente hallazgo más antiguo se encontraba en la ciudad de Capua, en Italia. Databa del año 300 a.C. y constituía la prótesis de una pierna realizada en madera y recubiertacon hierro y bronce. Lamentablemente, la pieza original se destruyó durante los bombardeos de la Segunda Guerra Mundial.

En la Edad Media, la protésica, al igual que muchas otras ciencias, tuvo un desarrollo prácticamente nulo. Solo destaca el gancho de mano y la pierna de madera y metal, pero era algo más decorativo que funcional. Estas prótesis normalmente solo se utilizaban en las batallas cuando algún soldado perdía una extremidad.

En el año 1800, James Potts creó una pierna de madera que incluía la articulación de la rodilla compuesta de acero y un pie también articulado por tendones de tripa de gato.

Nombre y definición de la forma geométrica

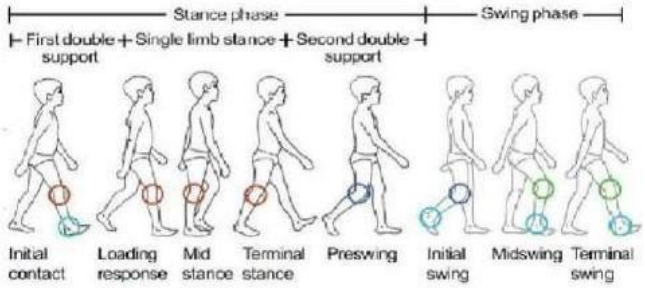
El conocimiento de la locomoción humana normal es la base del tratamiento sistemático y del manejo de la marcha patológica, especialmente cuando se usan prótesis y ortesis.

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.

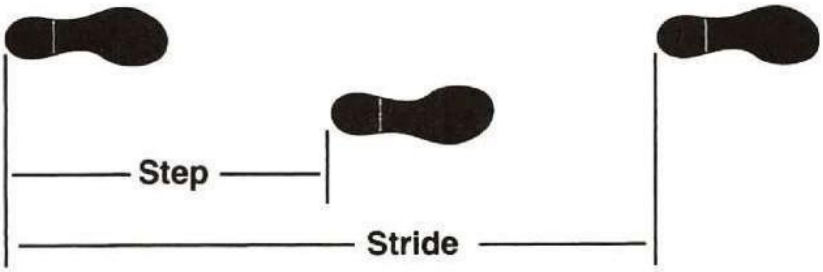
La movilidad libre de las articulaciones y el trabajo que desempeñan los músculoses importante para el éxito de esta tarea. Estos últimos deben actuar en el momento preciso y con la intensidad necesaria. La falta de ciertas acciones durante la marcha debe ser sustituida por otras, con el fin de mantener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

Ciclo de la marcha

El ciclo de la marcha comienza cuando el pie contacta con el suelo y termina conel 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 5.1).
Una pierna está en fase de apoyo cuando está en contacto con el suelo y está enfase de balanceo cuando no contacta con el suelo



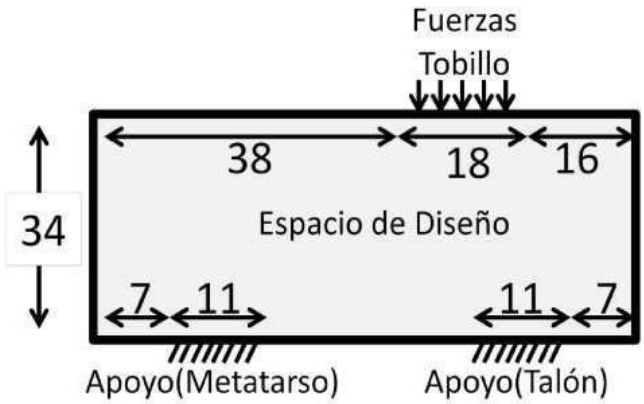
La longitud del paso completo es la distancia lineal entre los sucesivos puntos de contacto del talón del mismo pie. Longitud del paso es la distancia lineal en el plano de progresión entre los puntos de contacto de un pie y el otro pie (Figura 5.2).



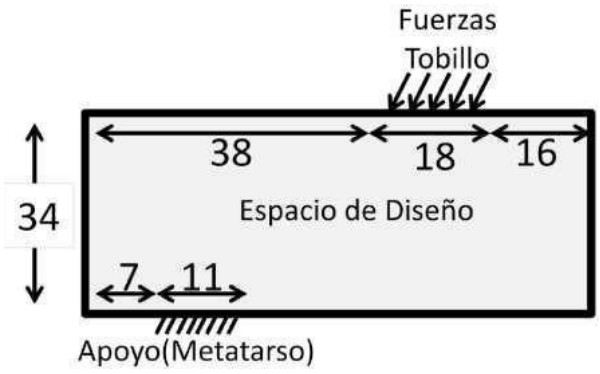
Propuesta de diseño de la geometría, alcances ylimitaciones

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

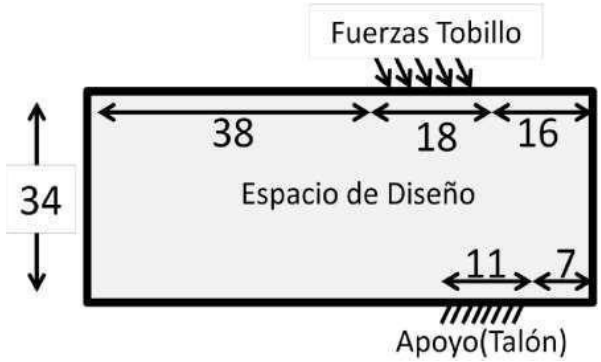
- Normal (El talón y área metatarsial son los apoyos, la fuerza se aplica sobre el tobillo con una fuerza de 500N)



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



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



Programación

Normal

```
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;
en
d
en
d
en
d
% 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;
```

Despegue

```
[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;
en
d
en
d
en
d
% 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)];
```

Apoyo

```
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;
en
d
en
d
en
d
% 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
=
l
m
id
;
e
n
d
end
%%%%%%%%%% MESH-INDEPENDENCY FILTER %%%%%%%%%%
function
[dcn]=check(nelx,nely,rmin,x,dc)
dcn=zeros(nely,nelx);
for i =
1:nel
xfor  j
=
1:nel
y
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;
e
n
d
e
n
d
% 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]=lkE
= 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)];
```

```
%%%% 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;
    e
    n
    d
    e
    n
    d
    e
    n
    d
    % 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
```

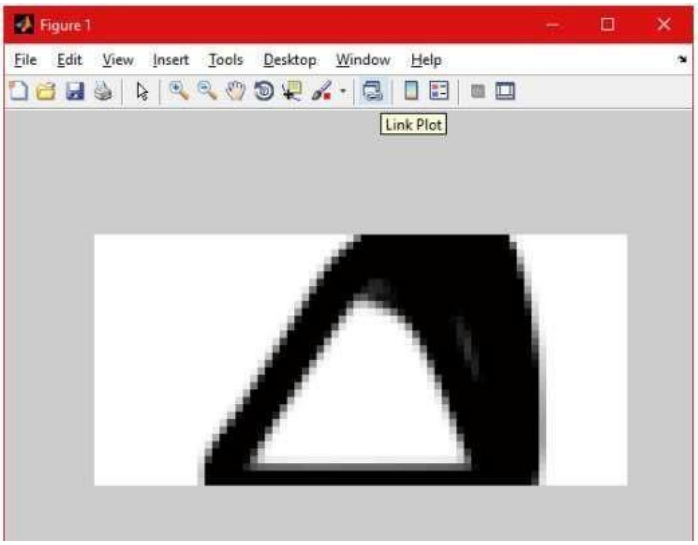
```
change = max(max(abs(x-xold)));
disp(['It.:' sprintf('%4i',loop) 'Obj.:' sprintf('%10.4f',c) ...
' Vol.: '
sprintf('%6.3f',sum(sum(x))/(nelx*nely)) ...
' ch.: ' sprintf('%6.3f',change )])
% PLOT DENSITIES
colormap(gray); imagesc(-x); axis equal; axis tight; axis
off;pause(1e 6);end
%%%%%%%%%% OPTIMALITY 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
=
l
m
id
;
el
s
e
l2
=
l
m
id
;
e
n
d
end
%%%%%%%%%% MESH-INDEPENDENCY FILTER %%%%%%%%%%
function
[dcn]=check(nelx,nely,rmin,x,dc)
dcn=zeros(nely,nelx);
for i =
1:nel
xfor  j
```

```
=
1:nel
y
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;
e
n
d
e
n
d
% 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];
```

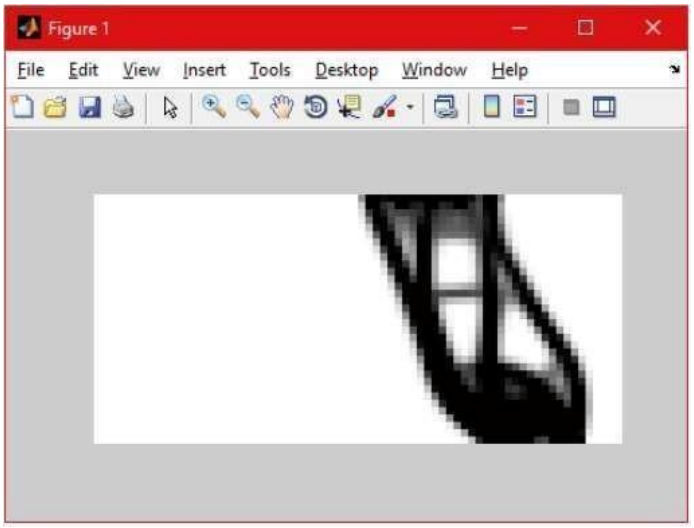
```
alldofs =
[1:2*(nely+1)*(nelx+1)];
freedofs =
setdiff(alldofs,fixeddofs);
% SOLVING 127
U(freedofs,:) = K(freedofs,freedofs) \F(freedofs,:);
U(fixeddofs,:)= 0;
%%%%%%%%%% ELEMENT STIFFNESS MATRIX %%%%%%%%
function
[KE]=lkE
= 1.;
nu = 0.3;
k=[ 1/2-nu/6 1/8+nu/8 -1/4-nu/12 -1/8+3*nu/8 ...
-1/4+nu/12 -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 optimización

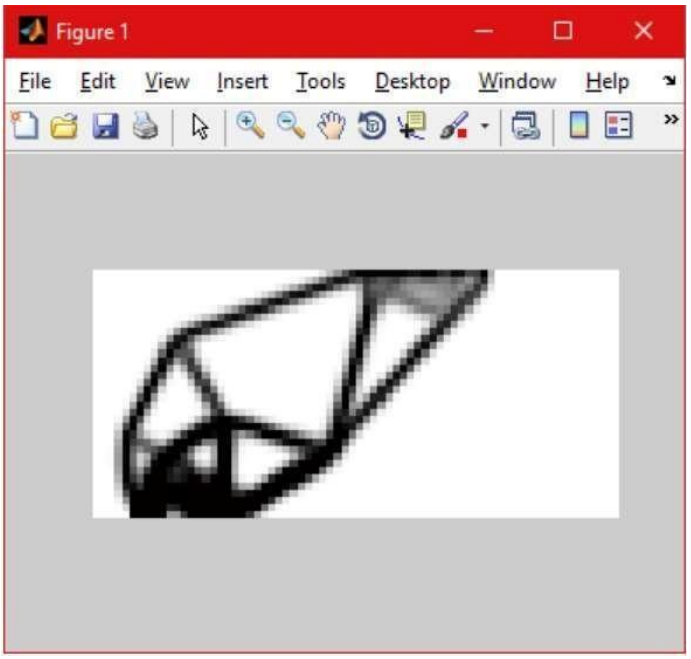
Normal



Despegue



Apoyo



Conclusiones

César Armando Luna Zapata 1844920 IMTC

Como conclusión en base a la comparación me pude dar cuenta que como tienenla misma carga, pero a diferentes ángulos, por lo que la simulación muestra en cuanto esta normal después en despegue y ya en apoyo, ya que las fuerzas se dividen de diferente manera.

Gabryiel Bailon Avila 1869828 IMTC

En esta práctica aprendí que en el apoyo normal se le aplica una carga de 500N, en la de despegue igual se aplica una carga de 500N, pero con un

ángulo de 30° y en la de apoyo al igual una carga de 500N, pero a un ángulo de 60°. Con lo anterior aprendí acerca de cómo introducir en Matlab lo anteriormente mencionadomodificando el código que siempre utilizamos de 99 líneas. Al igual manera me di cuenta de la simulación de cada una de ellas.

Víctor Adrián Higuera Vázquez 1876474 IMTC

La implementación de prótesis ha mejorado notablemente la vida de personas con amputaciones totales o parciales de miembros por lo cual significa un gran avancetanto medico como mecánico.

Carlos Eduardo Rivera López 1897545 IMTC

La implementación de las prótesis en el mundo se ha venido dando desde épocas muy antiguas por lo cual este viene a tener un progreso exponencial según la época por la cual se desarrolle. Además, el uso deeste código puede ayudar a visualizar los apoyos diferentes que tiene esta pieza y trabajar en conjunto para realizar una prótesis solo se necesita juntar sus diferentes vistas.

Andrés Anaya Hernández 1914471 IMTC

No solo he aprendido a hacer un diseño biomecánico si no que también he aprendido junto con mis compañeros como optimizar el trabajo como por ejemplo, haciendo uso de la programación e implementarla al trabajo para fines prácticos.

Referencias

- 99 Line Topology Optimization Code – O. Sigmund, Department of SolidMechanics, Building 404, Technical University of Denmark, DK-2800 Lyngby, Denmark.

O. Sigmund, Department of Solid Mechanics, Building 404, Technical University of Denmark, DK-2800 Lyngby, Denmark. El código puede serdescargado desde la página del autor: http://www.topopt.dtu.dk