



UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN
FACULTAD DE INGENIERÍA MECÁNICA Y ELÉCTRICA



SEMESTRE AGOSTO-DICIEMBRE 2022

LABORATORIO DE BIOMECÁNICA

PRÁCTICA 5

PROFESORA: M.C. YADIRA MORENO VERA

INTEGRANTES EQUIPO #3	MATRÍCULA
Diana Elisa Acosta Rodríguez	1914714
Joaquin Abdiel Coronado Vázquez	1924388
Brenda Giselle Hinojosa	1910669
Gloria Rosalía Domínguez Azueta	1913999
Andrea Anette Celestino Castillo	1925924
Ana Paulina Pérez Salazar	1865049
Javier Cisneros Saldivar	1856682

BRIGADA: 204 AULA: 12BMC DÍA: MARTES HORA: V1

San Nicolás de los Garza, N.L. a 31 de octubre de 2022.

ÍNDICE

Objetivo	1
Estado del arte.....	2
Propuesta de diseño de la geometría, alcances y limitaciones.	4
Pasos del desarrollo de la programación.....	5
Conclusiones.....	13
Bibliografía	16
Anexo	16

Práctica 5: Optimización de una prótesis de pie

1. Objetivo

Presentar una propuesta de análisis de un pie para las diferentes fases de la marcha humana y la programación para la ejecución de la optimización de características de trabajo específicas que presenta la optimización del pie con Matlab.

2. Definición de geometría

En el diseño de optimización de prótesis de pie y modelamiento de éstas se busca una curvatura que permite un almacenamiento máximo de energía. Esto debido a que los pies son de las extremidades más importantes y básicas no solo para poder caminar, sino simplemente para sostenerse de pie y poseer equilibrio, por lo tanto, esta prótesis de pie es de las más avanzadas en la medicina para casos de amputación, accidentes enfermedades, etc.

Esta prótesis puede suplantar desde unos cuantos dedos (prótesis de dedos de pie) el antepié, pie completo y pie con tobillo (todo depende del grado y lugar de la amputación).

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.

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

- 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, bioeléctricos y biomecánicos para otorgar movimientos especiales en caso de ser necesario.

3. Estado del arte

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.

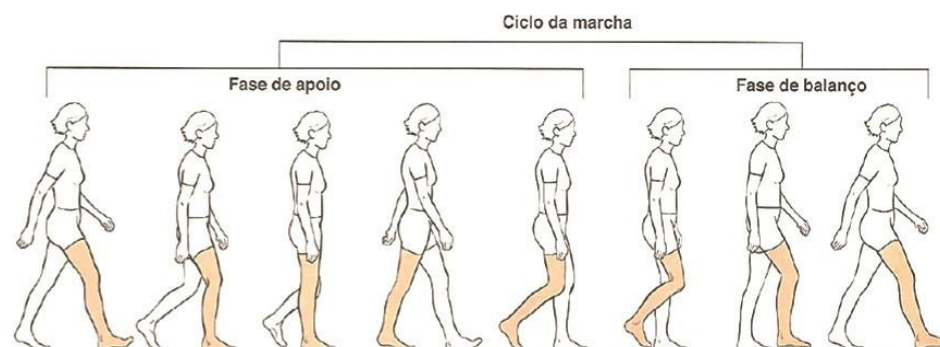


Figura 1. Fases del ciclo de la marcha

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).

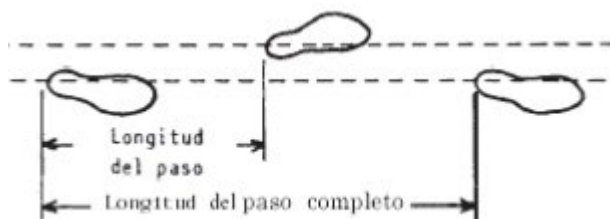


Figura 2. Longitud de paso

¿Qué son las Prótesis de Pie?

Las prótesis de pie se tratan de aparatos médicos destinados a sustituir de forma artificial un pie, tobillo o parte del pie faltante en el cuerpo de un paciente a causa de una amputación total o parcial del miembro.

Estos aparatos protéticos tienen tanto finalidades funcionales para la rehabilitación del paciente como funciones estéticas sumamente avanzadas para imitar de forma bastante realista la parte del cuerpo faltante.

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, por lo que las prótesis de pie y tobillo son de las más importantes y avanzadas en casos de amputación por accidentes, enfermedades o agenesia.

¿Cómo funciona una prótesis de pie?

Un pie protésico se trata de un aparato médico diseñado justo a la medida de cada paciente para suplantar estética y funcionalmente la parte faltante de la extremidad a causa de una amputación.

Esta tiene la función tanto de aparentar que existe la sección faltante como de ser un punto de apoyo para que el paciente pueda caminar, utilizar calzado y desempeñar sus labores de forma normal o lo más normal posible.

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

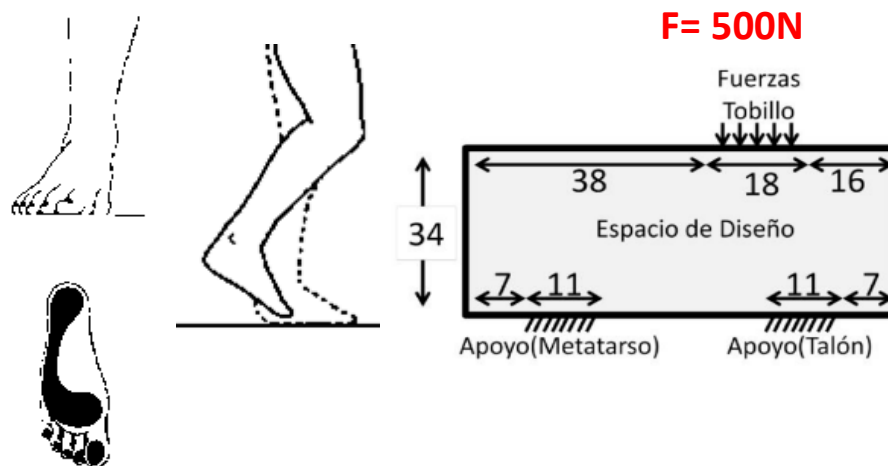
movimiento necesario para que el paciente pueda sostener de pie, caminar e inclusive correr y desempeñar labores o deportes de alto rendimiento.

4. Propuesta de diseño de la geometría, alcances y limitaciones.

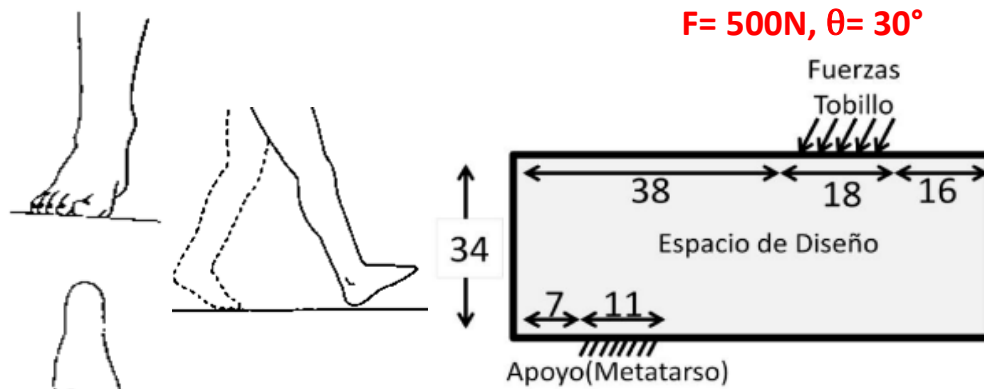
Como propuesta de este diseño es necesario analizar y realizar una selección del pie protésico la cual está basada en las características, propiedades y la absorción de energía durante el contacto inicial en la respuesta a la carga a medida que el amputado transfiere el peso del cuerpo al pie protésico. Es importante tener en cuenta que el material aporte flexibilidad y resistencia garantizando que la deflexión de la parte anterior del pie desde el apoyo plantar medio al apoyo plantar final sea proporcional al peso del usuario y al nivel de impacto.

Como estudio de lo que implica el movimiento, durante el tiempo en que el pie está en contacto con el suelo, el pie rueda desde el talón a la punta del pie, de manera análoga a como rueda un neumático sobre el suelo. Esto es analogado a que la pierna se comporta como un péndulo invertido cuyo centro cambia de posición desde el talón, al tobillo y desde el tobillo a la parte delantera del pie. Esta procesión de pivotes debe ser pensada en términos de que funcionan de manera traslapada y conjunta, no como entes discretos y sucesivos en el tiempo. Esta procesión transforma los pivotes discretos en una superficie de rodadura, forma que se asume que adopta el pie al caminar con respecto al tobillo.

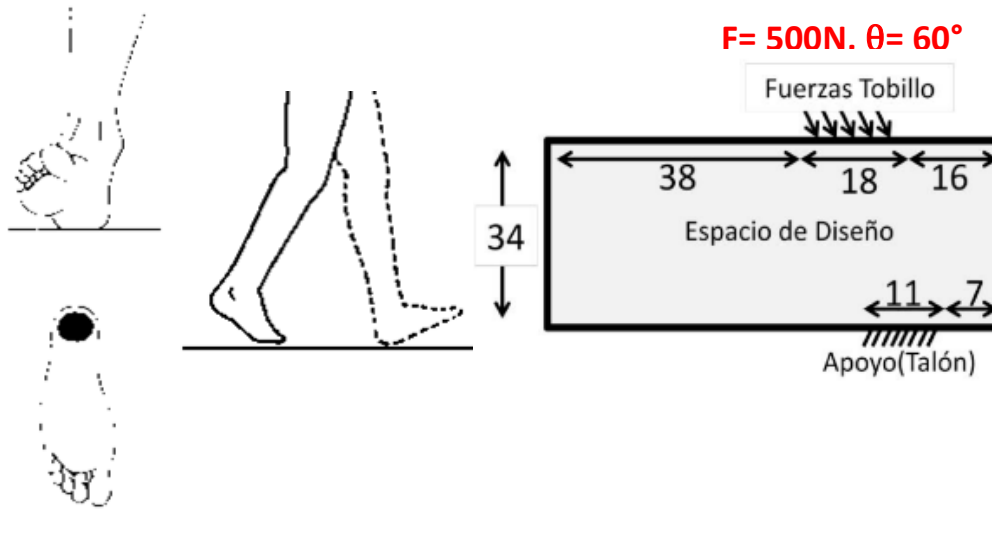
Normal



Despegue

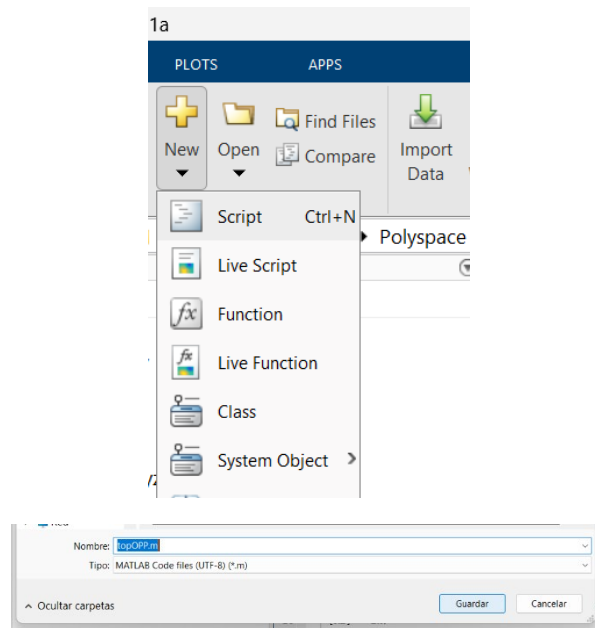


Apoyo



5. Pasos del desarrollo de la programación.

1. Primero abrimos un archivo nuevo en nuestro software MATLAB para poder hacer un nuevo programa:



2. Después escribimos el código en MATLAB con todos los pasos que hemos realizado anteriormente para optimizar las estructuras:

The image shows the MATLAB Editor window with the file 'D:\ARCHIVOS MATLAB 2021A\BIOMECANICA\topOPP.m' open. The script content is as follows:

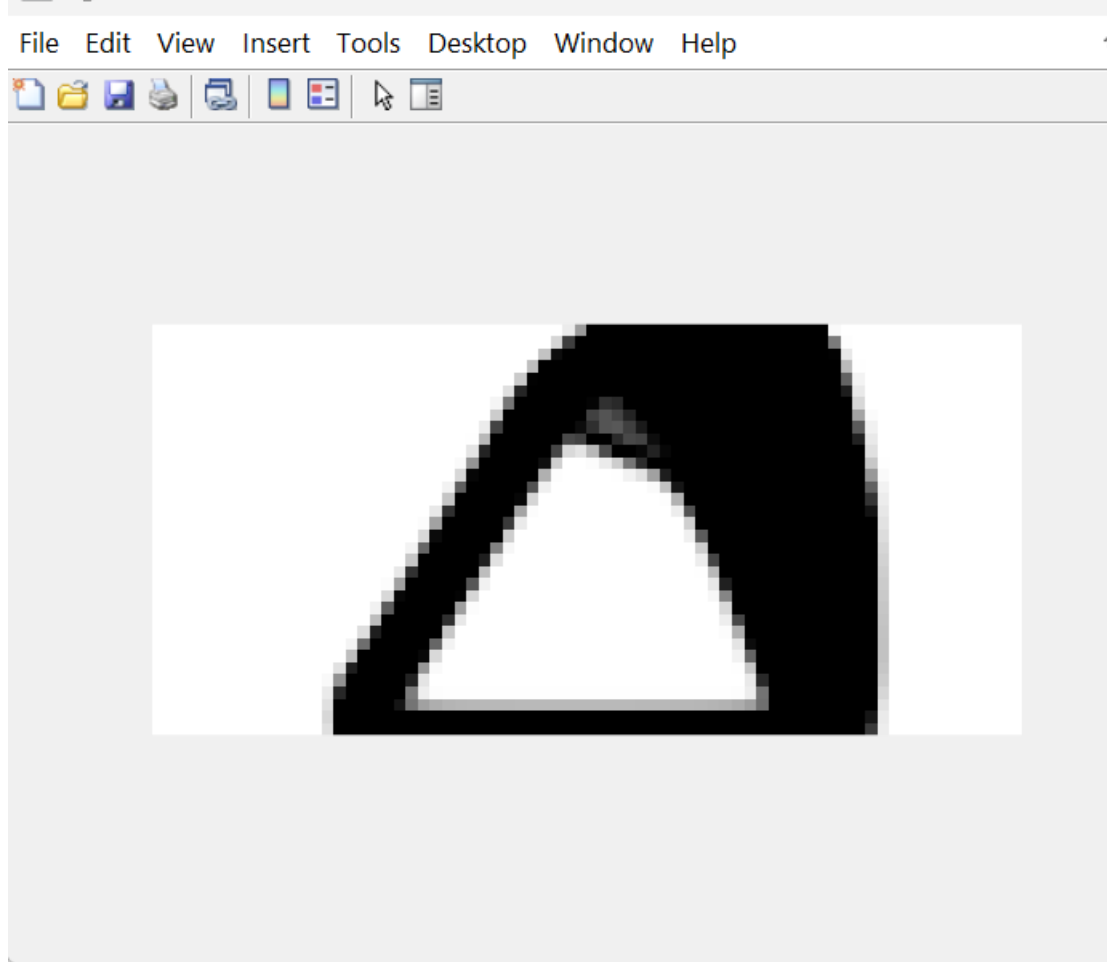
```

1  %%% A 99 LINE TOPOLOGY OPTIMIZATION CODE BY OLESIGMUND, OCTOBER 1999 %%%
2  %%% FASE NORMAL
3  %%% topOPP(72,34,0.33,3.0,1.2)
4  function topOPP(nelx,nely,volfrac,penal,rmin);
5  % INITIALIZE
6  x(1:nely,1:nelx) = volfrac;
7  loop = 0;
8  change = 1.;
9  % START ITERATION
10 while change > 0.01
11     loop = loop + 1;
12     xold = x;
13     % FE-ANALYSIS
14     [U]=FE(nelx,nely,x,penal);
15     % OBJECTIVE FUNCTION AND SENSITIVITY ANALYSIS
  
```

The status bar at the bottom indicates 'UTF-8' encoding, the file name 'topOPP', and the cursor position 'Ln 2 Col 10'.

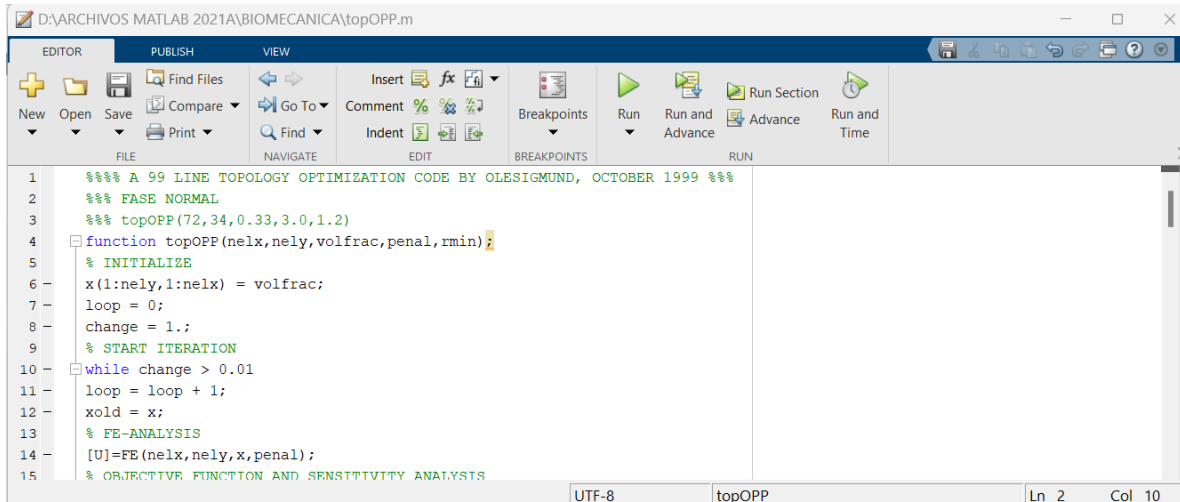
3. Cambiamos algunos parámetros del programa original para poder hacer una prótesis de pie más optima. Con esto, se hace referencia a la definición de las cargas y soportes:


```
topOPP.m x +
78 - n2 = (nely+1)* elx +ely;
79 - edof = [2*n1-1; 2*n1; 2*n2-1; 2*n2; 2*n2+1;2*n2+2;2*n1+1; 2*n1+2];
80 - K(edof,edof) = K(edof,edof) + x(ely,elx)^penal*KE;
81 - end
82 - end
83 - % DEFINE LOADSAND SUPPORTS (HALF MBB-BEAM)
84 - F(3222,1) = -1;
85 - F(3782,2) = -1;
86 - F(2662,3) = -1;
87 - F(2942,4) = -1;
88 - F(3502,5) = -1;
89 - fixeddofs = union([560:2*(nely+1):1260],[3920:2*(nely+1):4620]);
90 - alldofs = [1:2*(nely+1)*(nelx+1)];
91 - freedofs = setdiff(alldofs,fixeddofs);
92 - % SOLVING
93 - U(freedofs,:) = K(freedofs,freedofs) \F(freedofs,:);
94 - U(fixeddofs,:)= 0;
95 - %%%%%%%%%% ELEMENT STIFFNESS MATRIX %%%%%%%%%%
96 - function [KE]=lk
97 - E = 1;
98 - nu = 0.3;
```



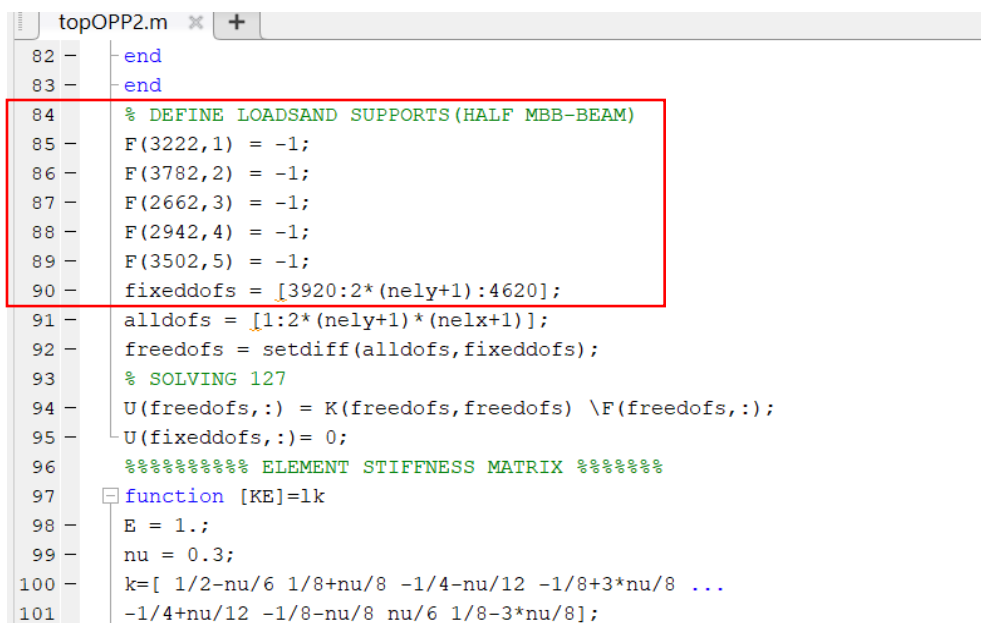
Fase Despegue

4. En esta fase también llamada de balanceo, el pie se dispone a hacer un esfuerzo especial para dejar el peso de la carga al otro pie, ya que éste perderá contacto con el suelo. El código principal, permanece intacto al anterior, ya que es la inicialización del proceso de optimización topológica.



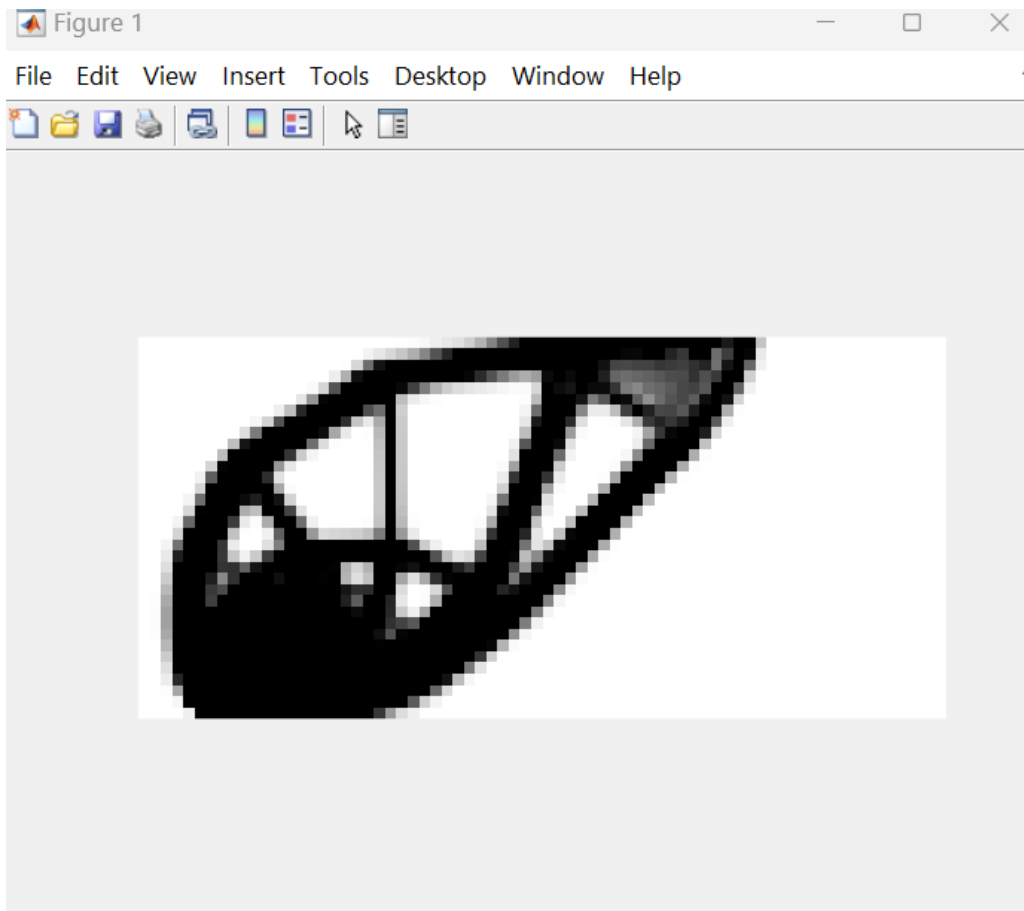
```
1  %%% A 99 LINE TOPOLOGY OPTIMIZATION CODE BY OLESIGMUND, OCTOBER 1999 %%%
2  %%% FASE NORMAL
3  %%% topOPP(72,34,0.33,3.0,1.2)
4  function topOPP(nelx,nely,volfrac,penal,rmin)
5  % INITIALIZE
6  x(1:nely,1:nelx) = volfrac;
7  loop = 0;
8  change = 1.;
9  % START ITERATION
10 while change > 0.01
11     loop = loop + 1;
12     xold = x;
13     % FE-ANALYSIS
14     [U]=FE(nelx,nely,x,penal);
15     % OBJECTIVE FUNCTION AND SENSITIVITY ANALYSIS
```

5. De acuerdo a la propuesta de diseño, los alcances y limitaciones, se pueden observar las cargas ejercidas en el metatarso, debido a la fase en que se encuentra. Con un ángulo de 30° respecto al eje normal. Para ello, se realizan las modificaciones pertinentes en la definición de cargas y soportes. Colocando también los DOF (Grados de libertad) fijos.



```
82 - end
83 - end
84 - % DEFINE LOADSAND SUPPORTS (HALF MBB-BEAM)
85 - F(3222,1) = -1;
86 - F(3782,2) = -1;
87 - F(2662,3) = -1;
88 - F(2942,4) = -1;
89 - F(3502,5) = -1;
90 - fixeddofs = [3920:2*(nely+1):4620];
91 - allldofs = [1:2*(nely+1)*(nelx+1)];
92 - freedofs = setdiff(allldofs, fixeddofs);
93 - % SOLVING 127
94 - U(freedofs,:) = K(freedofs, freedofs) \ F(freedofs, :);
95 - U(fixeddofs,:) = 0;
96 - %%%%%%%%%% ELEMENT STIFFNESS MATRIX %%%%%%%%%%
97 - function [KE]=lk
98 - E = 1.;
99 - nu = 0.3;
100 - k=[ 1/2-nu/6 1/8+nu/8 -1/4-nu/12 -1/8+3*nu/8 ...
101 - -1/4+nu/12 -1/8-nu/8 nu/6 1/8-3*nu/8];
```

6. Como resultado, se obtiene la siguiente forma de la prótesis optimizada, en la fase de despegue, ubicada en la parte del metatarso que es donde se concentran las cargas.



Fase apoyo

7. Finalmente, se encuentra la fase de apoyo, en la que el pie vuelve a tener contacto con el suelo. Como se observó anteriormente, se ejecuta el programa principal en donde se inicializan las variables y parámetros para la optimización topológica.

```

1  % A 99 LINE TOPOLOGY OPTIMIZATION CODE BY OLESIGMUND, OCTOBER 1999 %
2  % FASE APOYO
3  % topOPP3(72,34,0.33,3.0,1.2)
4  function topOPP3(nelx,nely,volfrac,penal,rmin);
5  % INITIALIZE
6  x(1:nely,1:nelx) = volfrac;
7  loop = 0;
8  change = 1.;
9  % START ITERATION
10 while change > 0.01
11     loop = loop + 1;
12     xold = x;
13     % FE-ANALYSIS
14     [U]=FE(nelx,nely,x,penal);

```

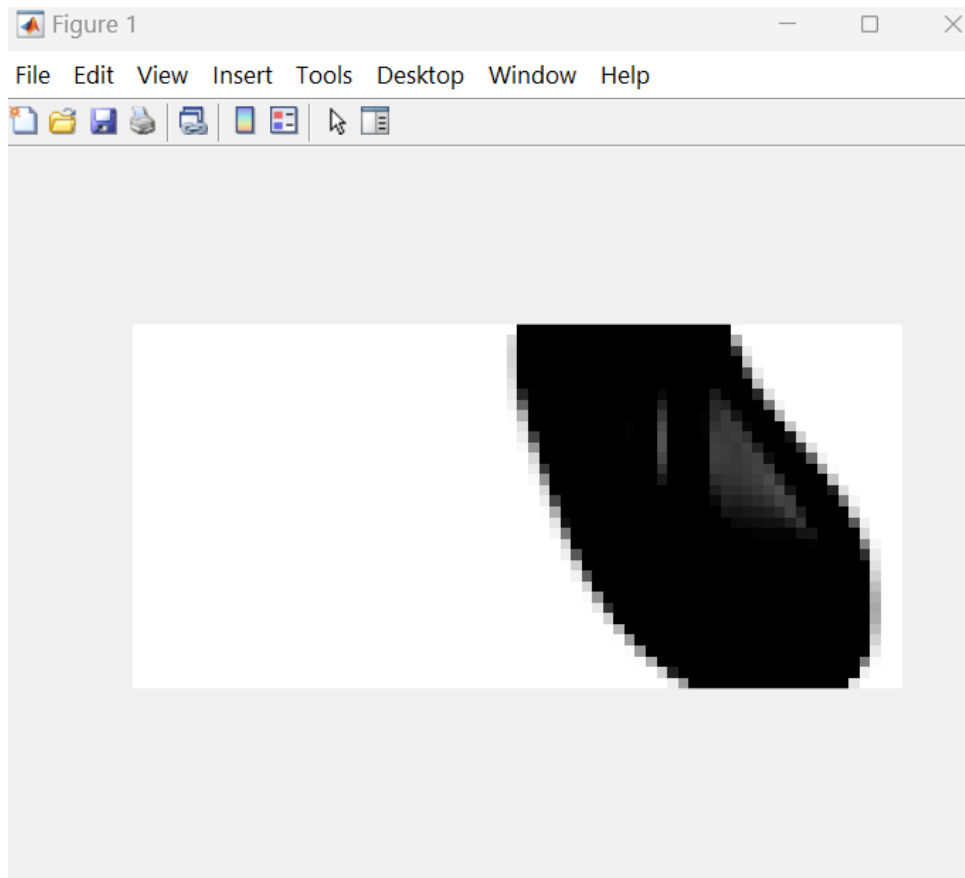
8. Y en la sección encargada de la definición de cargas, se establecen los valores de las cargas de acuerdo a las dimensiones y dirección de las mismas. En esta fase, las cargas son ejercidas en el talón. La fuerza tiene una magnitud de 500 N con un ángulo de 60° de su eje vertical, cuando se encuentra en fase normal. También se configuran los grados de libertad fijo como en los casos anteriores.

```

82 - end
83 - end
84 % DEFINE LOADSAND SUPPORTS (HALF MBB-BEAM)
85 F(3222,1) = -1;
86 F(3782,2) = -1;
87 F(2662,3) = -1;
88 F(2942,4) = -1;
89 F(3502,5) = -1;
90 fixeddofs = [560:2*(nely+1):1260];
91 alldofs = [1:2*(nely+1)*(nelx+1)];
92 freedofs = setdiff(alldofs,fixeddofs);
93 % SOLVING 12/
94 U(freedofs,:) = K(freedofs,freedofs) \ F(freedofs,:);
95 U(fixeddofs,:) = 0;
96 %%%%%%%%% ELEMENT STIFFNESS MATRIX %%%%%%%%%
97 function [KE]=lk
98 E = 1.;
99 nu = 0.3;
100 k=[ 1/2-nu/6 1/8+nu/8 -1/4-nu/12 -1/8+3*nu/8 ...
101    -1/4+nu/12 -1/8-nu/8 nu/6 1/8-3*nu/8];

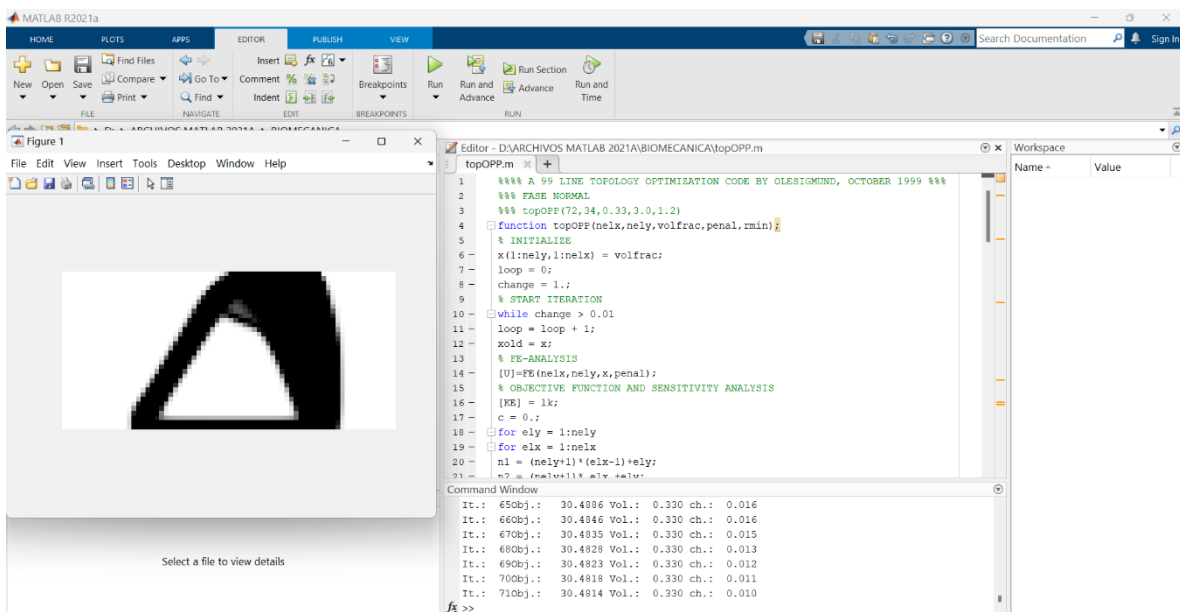
```

9. Por último, se obtiene la figura optimizada de la prótesis en base a la fase de apoyo. Que muestra el cambio en la ubicación de la concentración de las cargas que se encuentra en el talón.

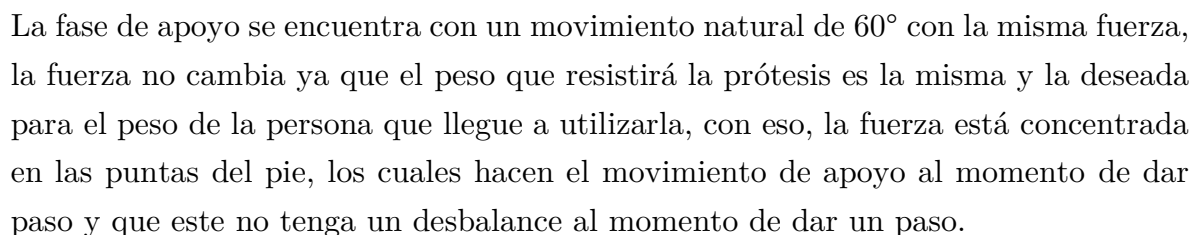


Resultados de las optimizaciones

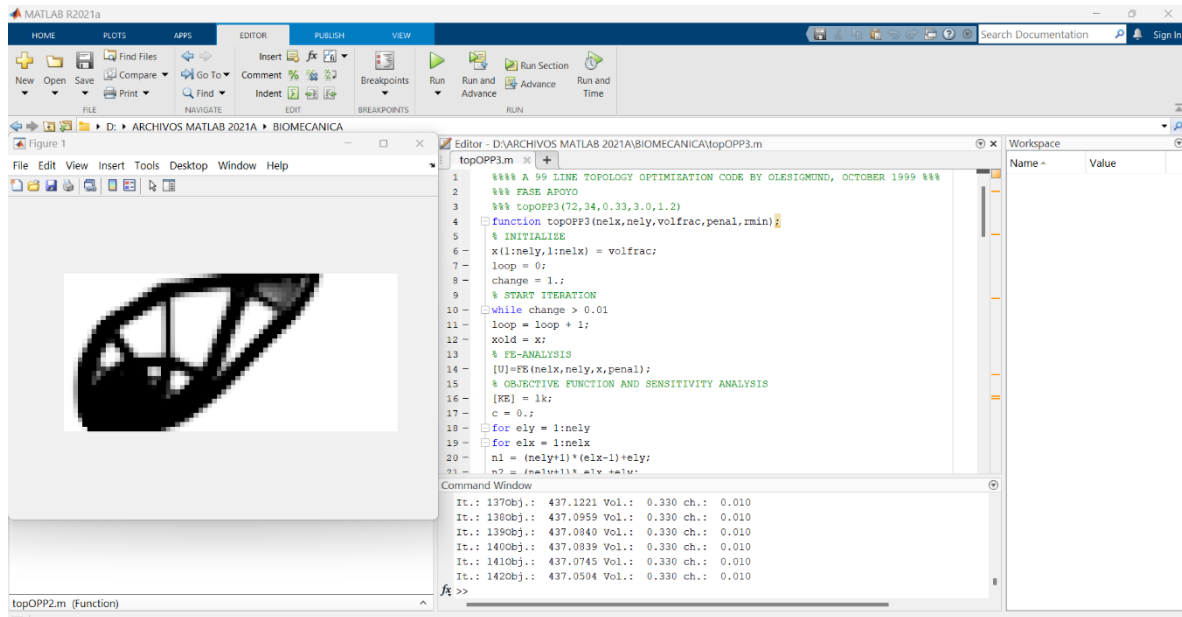
Fase Normal



Fase apoyo



Fase despegue



Por último, la fase de despegue, se muestra la optimización del diseño generativo, el cual nos da la mejor forma de crear la prótesis en esta área, esta parte sirve para recibir toda la carga y fuerza a 30° , ya que el talón es un punto de equilibrio cuando nuestro pie va a tocar suelo después de realizar un paso, por ello, el diseño que nos arroja es el óptimo para esta área de trabajo de la prótesis, es así donde se ve un mejor procedimiento, donde las cargas y fuerzas iniciales no cambian.

6. Conclusiones

Diana Elisa Acosta Rodríguez:

La optimización topológica es un proceso muy importante, ya que como se pudo observar a lo largo de las prácticas, permite generar un mejor diseño de las piezas para eliminar material innecesario y colocar estratégicamente más material en la zona donde se concentran los esfuerzos de una pieza. Esto, para mejorar la resistencia mecánica también. Cada vez el grado de dificultad era mayor, por las cargas y espacio de trabajo a considerar de acuerdo al uso y funciones que realizan las piezas a optimizar.

En esta ocasión se realizó el análisis mecánico de la prótesis de pie y el objetivo fue completado al realizar la optimización topológica en todas las fases del ciclo de

marcha (normal, despegue y apoyo) que es importante contemplar para las cargas y la concentración de esfuerzos.

Joaquín Abdiel Coronado Vázquez:

En esta última práctica, se vio una optimización topológica más complicada, ya que se muestra el tema visto en clase, diseño generativo, ya que, no solo vemos algo estructural en la vida diaria, sino que trabajamos en la parte de prótesis humanas, las cuales son más complejas para realizar, porque al igual que lo anterior visto en prácticas pasadas, es importante ver la mejor optimización, pero para prótesis se ocupa ver el mejor diseño generativo, para reducir costos y ver el material con el que se realizara la misma.

Brenda Giselle Hinojosa:

Al termino de esta práctica y de este laboratorio pudimos ver de diferentes maneras la optimización topológica, conforme avanzábamos hacíamos la optimización en piezas más completas y complejas. Creo que todo esto nos ha ayudado para entender mejor el tema, así como para saber porque es necesario realizar las optimizaciones y como estas nos ayudan a obtener un mejor trabajo.

Con esta práctica también vimos como la movilidad de las articulaciones y el trabajo que se realizan en los músculos son importantes para tomarlos en cuenta en nuestro análisis, así mismo como la falta de ciertas funciones nos lleva a sustituirlas por otras para así obtener estabilidad y progresión.

Gloria Rosalía Domínguez Azueta:

En esta práctica se logró realizar la optimización topológica para las diferentes fases de un pie, mejorando la configuración de la pieza y optimizando algunas características importantes para obtener un mejor resultado. De esta pieza, se realizaron distintas modificaciones en las fuerzas aplicadas según las distintas geometrías del comportamiento de un pie dentro de las tres fases de la marcha humana. Por su parte con esta optimización en la aplicación médica en conjunto con la ingeniería, afirme la importancia que tiene la optimización en la vida real, pues ello, permite reducir los costos del material y con ello hacer más factible las piezas

realizadas para una aplicación en específico. En cuanto al laboratorio en general, puedo mencionar que fue un gran aporte de conocimientos como método para la optimización de piezas para diversas áreas y para aplicaciones en futuros proyectos.

Andrea Anette Celestino Castillo:

Con la realización de esta práctica puedo mencionar que teniendo en claro lo que es la geometría y definición de nuestro proyecto, se puede abordar con más sencillas la elaboración de la práctica. Aunque es muy importante tener en claro cuál es el objetivo principal de esta práctica y como es que se llevara a cabo para poder lograrlo, ya que la realización de una optimalización para la prótesis de un pie no es sencilla y no se entienden con claridad los objetivos y las necesidades esenciales. Asimismo, con la elaboración de la práctica logramos realizar el análisis del movimiento de la prótesis con nuestro software, logrando así la optimización topológica para las diferentes fases de un pie.

Ana Paulina Pérez Salazar:

La realización de esta práctica no fue tan compleja debido a que tenemos los conocimientos necesarios para analizar lo requerido en Matlab, de igual manera creo tener un poco de más bases debido a que en clase ordinaria me correspondió este proyecto y en específico la optimización del diseño de prótesis, sin embargo es un poco complicado hablando de la geometría si se compara con las prácticas anteriores. Esta práctica nos ayudó a comprender lo que es necesario para la realización de una optimización más compleja.

Javier Cisneros Saldivar:

En esta práctica aprendimos a hacer más optima nuestra prótesis de pie, a diferencia de las practicas anteriores en esta práctica se trabajó con una estructura más complicada ya que no se trataba de un panorámico o una viga, esta estructura resulto más complicada de lo que se pensaba, pero al final se pudo optimizar nuestra prótesis, también aprendimos las bases que se deben considerar para la construcción de una prótesis de un pie, parámetros como movilidad, grados de libertad, entre otros.

Bibliografía

- 99 Line Topology Optimization Code – O. Sigmund, Department of Solid Mechanics, Building 404, Technical University of Denmark, DK-2800 Lyngby, Denmark.
- Lemus, L. E. (2021, 15 agosto). ▷ *Prótesis de Pie - Tipos, Funciones y Precios* - Conoce más aquí. Mi Protesis de Pierna. <https://miprotesisdepierna.mx/protesis-de-pie/>
- Nogueras, M. A. (1999, 1 enero). *Fases de la marcha humana* / *Revista Iberoamericana de Fisioterapia y Kinesiología*. <https://www.elsevier.es/es-revista-revista-iberoamericana-fisioterapia-kinesiologia-176-articulo-fases-marcha-humana-13012714>

Anexo

Código en Matlab para la optimización de una prótesis de pie

Código 1-FASE NORMAL

```
%% A 99 LINE TOPOLOGY OPTIMIZATION CODE BY OLESIGMUND,
OCTOBER 1999 %%
%% FASE NORMAL
%% topOPP(72,34,0.33,3.0,1.2)
function topOPP(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 =
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 2-FASE DESPEGUE

```

%%% A 99 LINE TOPOLOGY OPTIMIZATION CODE BY OLESIGMUND,
OCTOBER 1999 %%%
%%% FASE DESPEGUE
%%% topOPP2(72,34,0.33,3.0,1.2)
function topOPP2(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 3-FASE APOYO

```
%% A 99 LINE TOPOLOGY OPTIMIZATION CODE BY OLESIGMUND,
OCTOBER 1999 %%
%% FASE APOYO
%% topOPP3(72,34,0.33,3.0,1.2)
function topOPP3(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)];

```