

# PRÁCTICA 4

## REFUERZA EL CABLE DE UN TELEFÉRICO

### INTEGRANTES:

OSIRIS ACOSTA CISNEROS	1992234	IMTC
VÍCTOR HUGO PUENTE ÁLVAREZ	1929757	IMTC
ALEJANDRO HERNÁNDEZ NAVARRO	1923272	IMTC
DIEGO ÁVILA GONZÁLEZ	1853396	IMTC
ANEGLA RODRÍGUEZ FLORES	1896624	IMTC

# 1) Nombre y definición de la geometría

---

- El objetivo de esta práctica es presentar una propuesta de análisis de formas y de la programación para la ejecución de características de trabajo específicas que presenta las ventajas.
- La geometría que se propuso a analizar y optimizar fue el cable de un teleférico, donde haciendo uso del algoritmo de optimización topológica se buscará la estructura óptima que refuerce el cable lo máximo posible para su uso en la vida real.



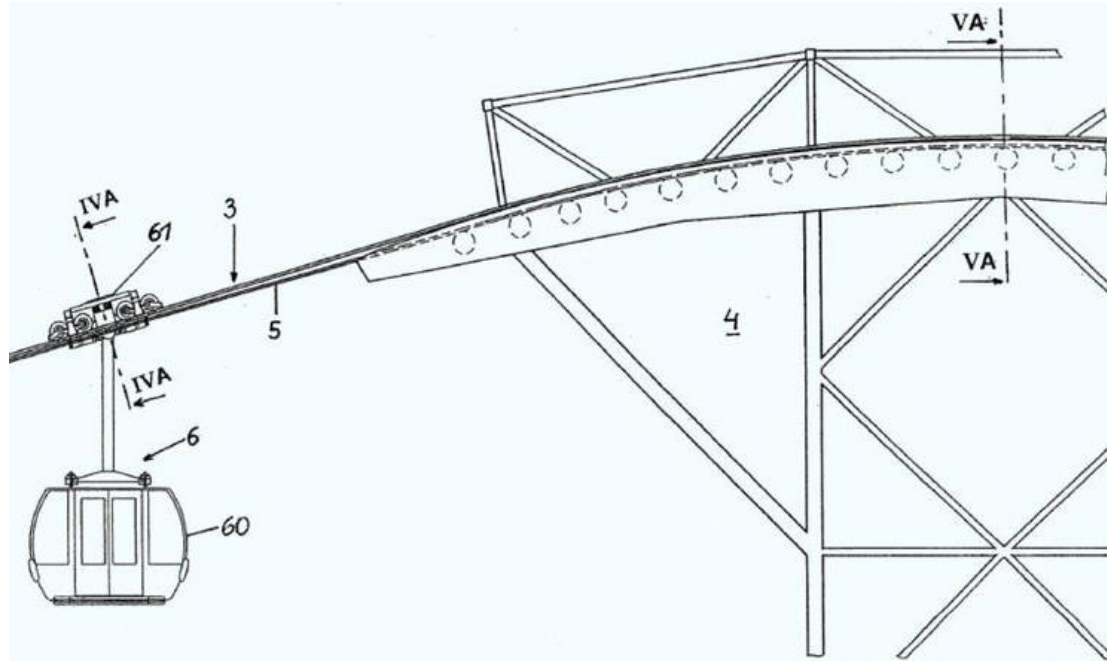
## 2) Estado del arte

---

El teleférico es un sistema de transporte no tripulado aéreo constituido por cabinas colgadas de una serie de cables que se encargan de hacer avanzar a las unidades a través de las estaciones. Cuando las cabinas van por tierra se denomina funicular.



# Teleferico



G.2

Un teleférico es un transporte aéreo mediante góndolas o cabinas colgadas de cables, los cuales a su vez son sostenidos por torres o postes. Este tipo de transporte es tradicionalmente asociado al esquí o como atractivo turístico, pero en años recientes se ha explorado la posibilidad de usarlos para mejorar la movilidad urbana como medio de transporte público. Algunas ciudades del mundo, además de las aquí listadas, que han explorado esta opción son: Portland, Isla Roosevelt (Nueva York), Hong Kong, Lagos, Constantina, Coblenza y Taipéi (Taiwán).



# LAS ESTACIONES

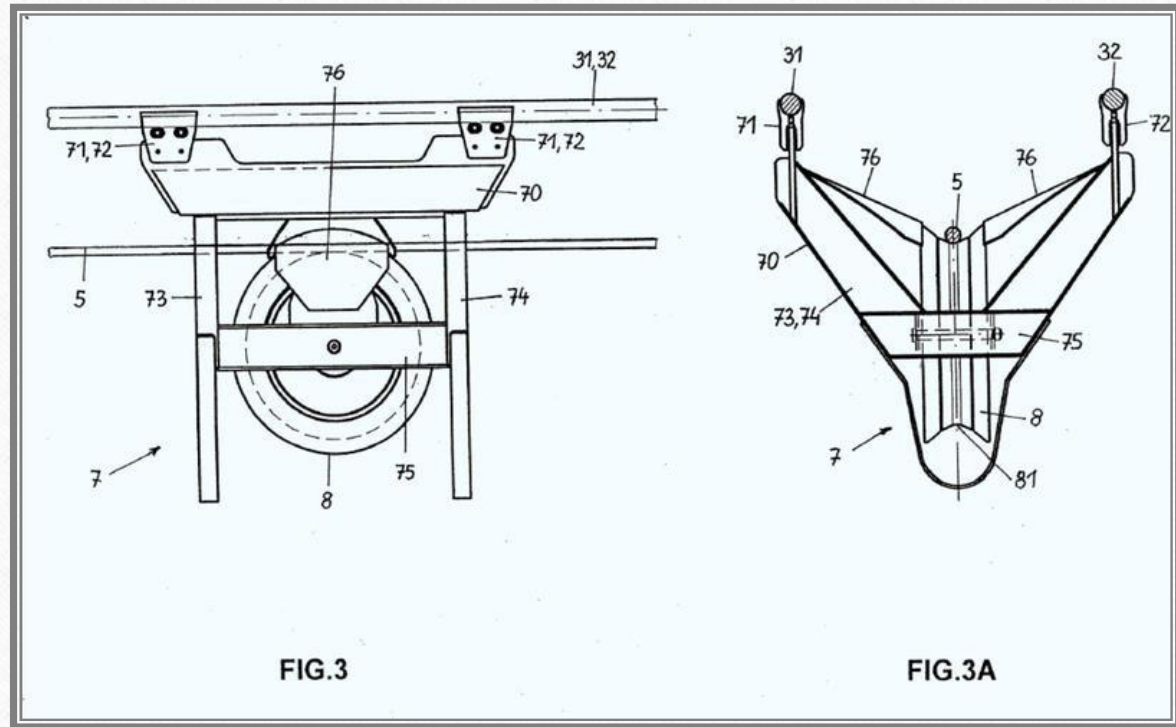
---

Los sistemas de teleférico requieren como mínimo dos estaciones. Una estación se construye en el punto de inicio del teleférico y el otro en su punto final. En cuanto a los aspectos técnicos de un teleférico, el motor, los frenos de servicio o el panel de mando no se sitúan en el vehículo sino directamente en la estación.



## EL GRUPO MOTOR

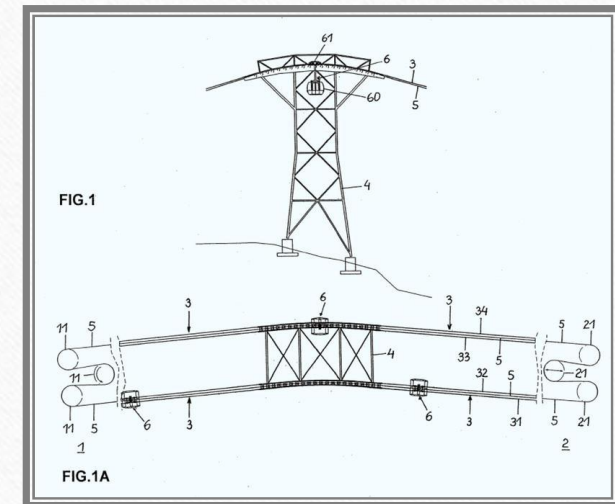
Dependiendo de las necesidades específicas del proyecto, la motriz se puede ubicar tanto en la estación inferior o superior, en configuración enterrada o aérea, en las variantes motriz fija o motriz tensora. Un grupo motor clásico está formado al menos por un motor, un freno de servicio, un freno de emergencia y el reductor de velocidad.





# LA LÍNEA

Ningún otro elemento como la morfología del terreno es capaz de influir tan claramente en las características de la línea de un teleférico. En consecuencia, es importante el desarrollo de todos los aspectos de los componentes y su correcto funcionamiento, los cuales ofrecen a los pasajeros el máximo confort y seguridad. La línea tiene diferentes componentes, como las pilonas, los balancines y el cable.



# SISTEMAS DE PINZA FIJA

---

- **Telesquís:** Los telesquís tienen una larga historia y son el tipo de instalación más común en todo el mundo. Pueden funcionar prácticamente en cualquier condición meteorológica y también se distinguen por tener unos bajos costes de mantenimiento.
- **Telesillas de pinza fija:** Este tipo de telesillas también están en uso desde hace décadas y, gracias a su robustez y su larga vida operativa, junto con unos costes de inversión relativamente bajos, representan un tipo de sistema de elevación altamente solicitado.



# INSTALACIONES DESEMBRAGABLES

---

- **Telesillas desembragables:** Los telesillas desembragables son los sistemas más populares, cómodos y rápidos para los seguidores de los deportes de invierno. Pueden disponer de cúpulas que protegen de las inclemencias del tiempo, las cuales se pueden elevar durante el trayecto.
- **Telecabinas desembragables monocable:** Los telecabinas desembragables monocable disponen de un único cable que actúa a la vez como cable portador y tractor. Las cabinas están disponibles en diferentes diseños, siempre con una estructura de aluminio.

# INSTALACIONES DESEMBRAGABLES

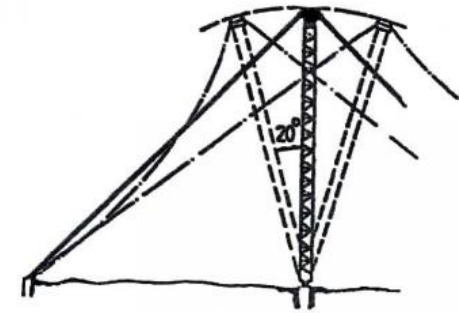
---

- **Telemix: Combinación de telesilla y telecabina desembragable:** Se conoce como sistema Telemix a la combinación de telesilla y telecabina. Estas instalaciones incluyen las ventajas de ambos sistemas. A los esquiadores expertos les atraen sobre todo las cómodas sillas, ya que no tienen que quitarse los esquís, mientras que las cabinas son más adecuadas para principiantes, niños o incluso para las personas que no esquían.
- **elecabines bicable y tricable:** Estos teleféricos desembragables deben su nombre al número de cables que tienen. Una instalación bicable (2S, del alemán) se sostiene sobre un cable portador y un cable tractor. Una instalación tricable (3S) se sostiene sobre un cable tractor y dos cables portadores.

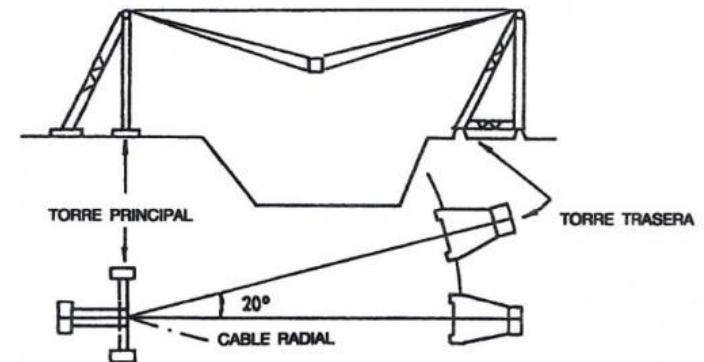


### 3) Propuesta de diseño de la geometría, alcances y limitaciones

- Se distinguen los siguientes cables en el blondín:
  - Cable vía o cable carril: cable atirantado sobre el que se desplaza el carretón o bicicleta. Está fijo a dos mástiles o torres, actuando a modo de dintel de pórtico.
  - Cable tractor o de vaivén: es el que desplaza al carretón.
  - Cable elevador: sirve para el izado de la carga, fijando la posición vertical del gancho

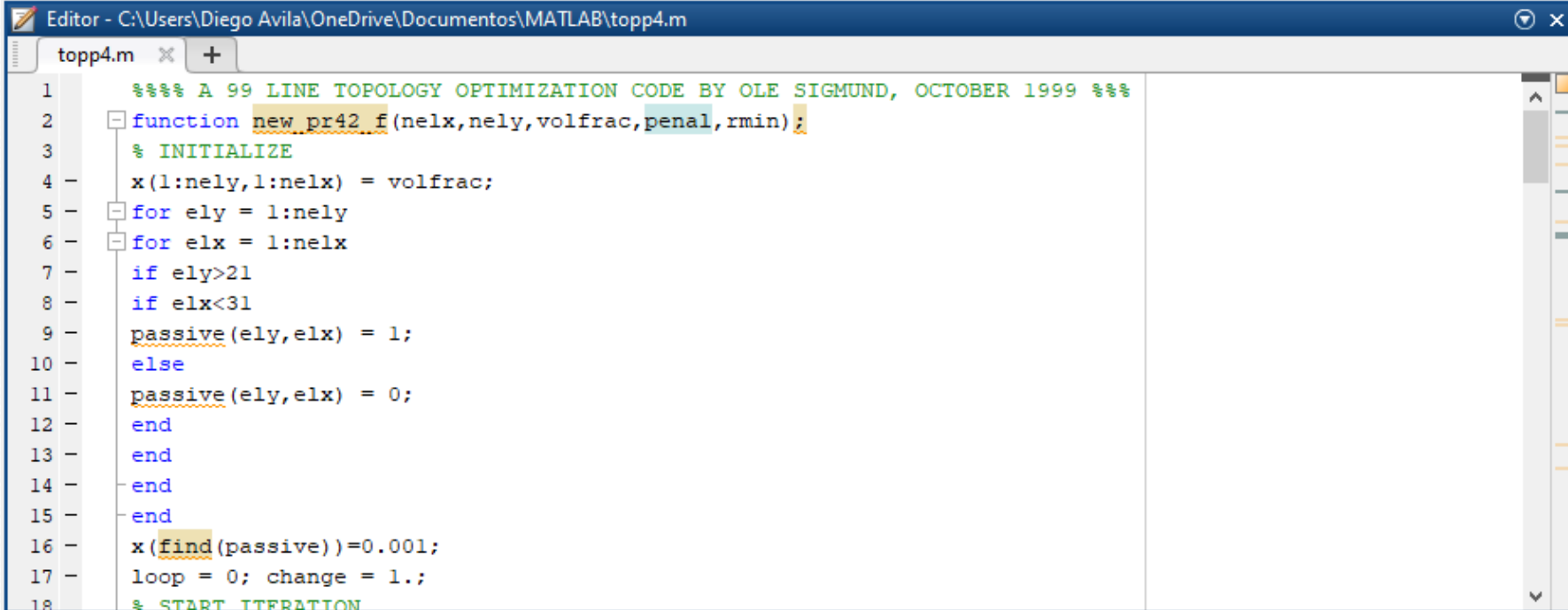


*Blondín de cable basculante*



## 4) Pasos del desarrollo de la programación

1.-



```
Editor - C:\Users\Diego Avila\OneDrive\Documentos\MATLAB\topp4.m
topp4.m
1  %%%% A 99 LINE TOPOLOGY OPTIMIZATION CODE BY OLE SIGMUND, OCTOBER 1999 %%%
2  function new_pr42_f(nelx,nely,volfrac,penal,rmin);
3  % INITIALIZE
4  x(1:nely,1:nelx) = volfrac;
5  for ely = 1:nely
6  for elx = 1:nelx
7  if ely>21
8  if elx<31
9  passive(ely,elx) = 1;
10 else
11 passive(ely,elx) = 0;
12 end
13 end
14 end
15 end
16 x(find(passive))=0.001;
17 loop = 0; change = 1.;
18 % START ITERATION
```



2.-



The screenshot shows a MATLAB Command Window with a blue title bar. A yellow banner at the top reads "New to MATLAB? See resources for [Getting Started](#)." Below this, the command prompt shows the execution of the `topp4` function. The first attempt, `>> topp4`, results in a red error message: "Not enough input arguments." The second attempt, `fx >> topp4(60,80,0.33,3.0,1.5);`, is shown on the next line but has not yet produced an output.

```
Command Window
New to MATLAB? See resources for Getting Started.
>> topp4
Not enough input arguments.

Error in topp4 (line 4)
x(1:nely,1:nelx) = volfrac;

fx >> topp4(60,80,0.33,3.0,1.5);
```

`topp4(60,80,0.33,3.0,1.5);`

3.-

MATLAB R2019a

HOME PLOTS APPS EDITOR PUBLISH VIEW

File Edit Breakpoints Debug

Current Folder: C:\Program Files\Polyspace\R2019a\bin

Editor: C:\Users\Diego Avila\OneDrive\Documentos\MATLAB\topp4.m

```

83 - K = sparse(2*(nelx+1)*(nely+1), 2*(nelx+1)*(nely+1));
84 - F = sparse(2*(nely+1)*(nelx+1), 2); U = sparse(2*(nely+1)*(nelx+1), 2);
85 - for ely = 1:nely
86 -     for elx = 1:nelx
87 -         n1 = (nely+1)*(elx-1)+ely;
88 -         n2 = (nely+1)* elx +ely;
89 -         edof = [2*n1-1; 2*n1; 2*n2-1; 2*n2; 2*n2+1; 2*n2+2; 2*n1+1; 2*n1+2];
90 -         K(edof,edof) = K(edof,edof) + x(ely,elx)^penal*KE;
91 -     end
92 - end
93 - % DEFINE LOADSAND SUPPORTS (HALF MBB-BEAM)
94 - F(40,1) = -1;
95 - fixeddofs = 2*(nely+1):2*(nely+1):2*(nelx+1)*(nely+1);
96 - alldofs = [1:2*(nely+1)*(nelx+1)];
97 - freedofs = setdiff(alldofs,fixeddofs);
98 - % SOLVING
99 - U(freedofs,:) = K(freedofs,freedofs) \F(freedofs,:);

```

Command Window

New to MATLAB? See resources for [Getting Started](#).

It.	Obj.	128.2926Vol.	0.330	ch.	0.028
42	Obj.	128.2926Vol.	0.330	ch.	0.028
43	Obj.	128.2494Vol.	0.330	ch.	0.029
44	Obj.	128.2143Vol.	0.330	ch.	0.028
45	Obj.	128.1593Vol.	0.330	ch.	0.028
46	Obj.	128.1102Vol.	0.330	ch.	0.027
47	Obj.	128.0554Vol.	0.330	ch.	0.025

new\_pr42\_f / FE

Ln 91 Col 1



4.-

MATLAB R2019a

HOME PLOTS APPS EDITOR PUBLISH VIEW

New Open Save Find Files Compare Go To Find Insert Comment Indent Breakpoints Run Run and Advance Run Section Advance

FILE NAVIGATE EDIT BREAKPOINTS RUN

C:\Program Files\Polyspace\R2019a\bin

Current Folder

- Name
- m3registry
- registry
- util
- win32
- win64
- crash\_analyzer.cfg
- deploytool.bat
- lclata.xml
- lclata.xsd
- lclata\_utf8.xml
- matlab.exe
- mbuild.bat
- mcc.bat
- mex.bat
- mexext.bat
- mw\_mpiexec.bat
- worker.bat

Editor - C:\Users\Diego Avila\OneDrive\Documents\MATLAB\topp4.m

```
topp4.m
83 - K = sparse(2*(nelx+1)*(nely+1), 2*(nelx+1)*(nely+1));
84 - F = sparse(2*(nely+1)*(nelx+1), 2); U = sparse(2*(nely+1)*(nelx+1), 2);
85 - for ely = 1:nely
86 -   for elx = 1:nelx
87 -     n1 = (nely+1)*(elx-1)+ely;
88 -     n2 = (nely+1)*elx + ely;
89 -     edof = [2*n1-1; 2*n1; 2*n2-1; 2*n2; 2*n2+1; 2*n2+2];
90 -     K(edof,edof) = K(edof,edof) + x(ely,elx)^pena;
91 -   end
92 - end
93 - % DEFINE LOADSAND SUPPORTS (HALF MBB-BEAM)
94 - F(40,1) = -1;
95 - fixeddofs = 2*(nely+1):2*(nely+1):2*(nelx+1)*nely;
96 - alldofs = [1:2*(nely+1)*(nelx+1)];
97 - freedofs = setdiff(alldofs,fixeddofs);
98 - % SOLVING
99 - U(freedofs,:) = K(freedofs,freedofs) \ F(freedofs,:);
```

Command Window

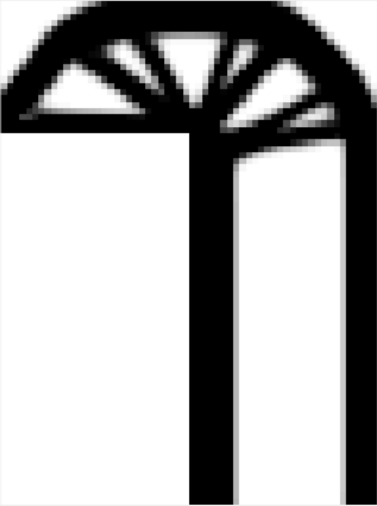
New to MATLAB? See resources for [Getting Started](#).

It.:	58	Obj.:	127.8946	Vol.:	0.330	ch.:	0.015
It.:	59	Obj.:	127.8929	Vol.:	0.330	ch.:	0.013
It.:	60	Obj.:	127.8917	Vol.:	0.330	ch.:	0.012
It.:	61	Obj.:	127.8907	Vol.:	0.330	ch.:	0.011
It.:	62	Obj.:	127.8898	Vol.:	0.330	ch.:	0.011
It.:	63	Obj.:	127.8899	Vol.:	0.330	ch.:	0.010

fx >>

Figure 1

File Edit View Insert Tools Desktop Window Help



new\_pr42\_f / FE Ln 91 Col 1

# CÓDIGO

```
%%%% A 99 LINE TOPOLOGY OPTIMIZATION CODE BY OLE
SIGMUND, OCTOBER 1999 %%%
function new_pr42_f(nelx,nely,volfrac,penal,rmin);
% INITIALIZE
x(1:nely,1:nelx) = volfrac;
for ely = 1:nely
for elx = 1:nelx
if ely>21
if elx<31
passive(ely,elx) = 1;
else
passive(ely,elx) = 0;
end
end
end
x(find(passive))=0.001;
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=1:2
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,passive);
% 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)))));
xnew(find(passive))=0.001;

```

```

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
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),2); U =
sparse(2*(nely+1)*(nelx+1),2);
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(40,1) = -1;
fixeddofs = 2*(nely+1):2*(nely+1):2*(nelx+1)*(nely+1);

```

```

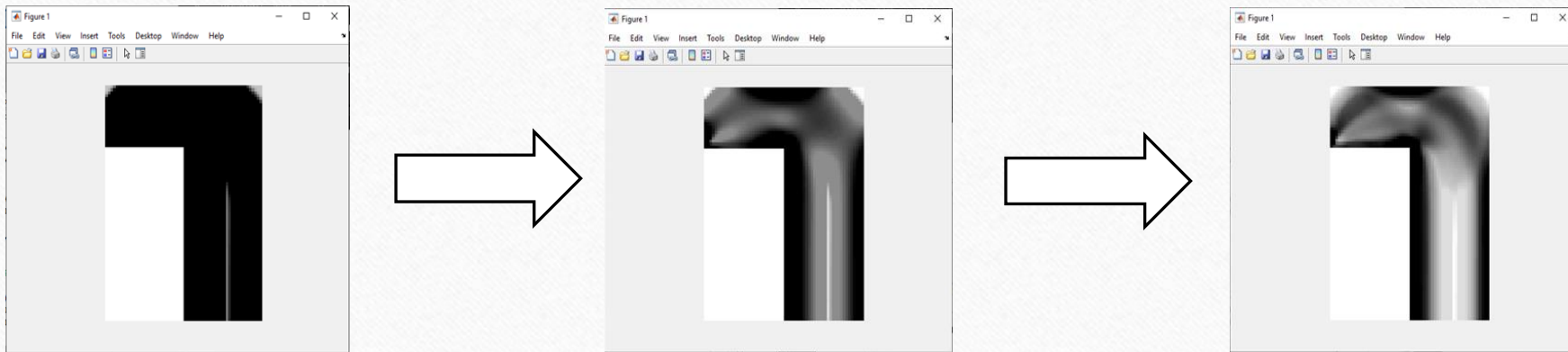
alldofs = [1:2*(nely+1)*(nelx+1)];
freedofs = setdiff(alldofs,fixeddofs);
% SOLVING
U(freedofs,:) = K(freedofs,freedofs) \F(freedofs,:);
U(fixeddofs,:)=0;
%%%%%%%%%% ELEMENT STIFFNESS MATRIX %%%%%%%%%%
function [KE]=lk
E = 1.;
nu = 0.3;
k=[ 1/2-nu/6 1/8+nu/8 -1/4-nu/12 -1/8+3*nu/8 ...
-1/4+nu/12 -1/8-nu/8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

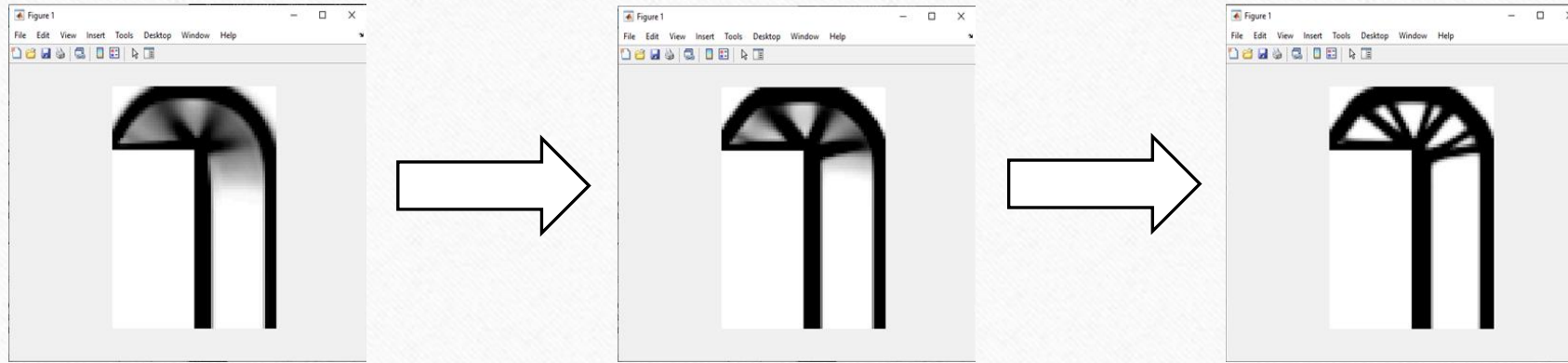
```



## 5) Resultados de la optimización

A partir de la propuesta de diseño de la geometría se realizó el código de Matlab de optimización topológica considerando los alcances y limitaciones del diseño para llegar a un resultado final, a continuación se muestran una serie de imágenes del proceso de la generación del diseño final.





Finalmente se tiene un diseño final obtenido mediante el proceso de optimización topológica utilizando el código de Matlab.

La geometría particular que se obtuvo es debido a las consideraciones de diseño que se plantearon durante la etapa de la propuesta de diseño.



## 6) Conclusiones

---

- Diego Ávila González

En esta actividad aplicamos una vez más el programa de 99 lenas de la optimización topológica en MATLAB pero ahora con el caso del refuerzo del cable de un teleférico, aquí las variables cambian ya que se muestra que la estructura debe de aguantar las cargas que genera un teleférico y también ver que este se debe de generar con el menor material posible por lo tanto se generó una estructura la cual se puede ver representada como vigas que soportan al teleférico.

- Alejandro Hernández Navarro

Durante esta práctica se propuso una geometría para reforzar el cable de un teleférico con la finalidad de aplicar la optimización topológica y poder optimizar la geometría propuesta, reduciendo la masa y manteniendo la rigidez de la pieza. Para lograr esto se utilizaron conocimientos previos respecto al código de optimización topológica en Matlab para adaptarlo al problema que se está trabajando, para esto primeramente se propuso el diseño con sus respectivas restricciones y condiciones frontera, después se pasó al código de Matlab de optimización topológica para así obtener el diseño final.



- Victor Hugo Puente Alvarez

Por medio de esta práctica logramos reforzar el cable de un teleférico, logramos calcular los esfuerzos y deformaciones que podría presentar el cable del teleférico, esto con ayuda desde un programa en Matlab, por medio de la clase ordinaria a realizar la optimización topologica en la pieza para hacer este cable la mejor opción y la más segura al momento de usarlo, no crear un accidente y esto es lo que se hizo en la práctica y lo logramos plasmar en la simulación de Matlab como de un diseño se aplica dicho optimización y tenemos un mejor diseño que tiene reforzados los mejores enfoques de esfuerzo.

- Osiris Acosta Cisneros

En esta práctica hicimos uso del algoritmo que ya hemos venido utilizando a lo largo del desarrollo de las demás práctica, es objetivo de esta práctica es optimizar topológicamente un cable de un teleférico, con el fin de obtener el diseño mejor apto para la vida real en términos de resistencia y durabilidad. Se logró obtener el resultado y se puede concluir que cumple con los requisitos.



## 7) Bibliografías

---

- Elementos de teleféricos - LEITNER. (2021). Leitner. Recuperado 16 de octubre de 2021, de <https://www.leitner.com/es/empresa/informaciones-utiles/elementos-de-telefericos/>
- Etiqueta: Andarivel. El blog de Vctor Yepes. (n.d.). Recuperado 16 de octubre de 2021, de <https://victoryepes.blogs.upv.es/tag/andarivel/>

