



**UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN**  
**FACULTAD DE INGENIERÍA MECÁNICA Y ELÉCTRICA**



**SEMESTRE AGOSTO-DICIEMBRE 2022**

**LABORATORIO DE BIOMECÁNICA**

**PRÁCTICA 4**

**PROFESORA: M.C. YADIRA MORENO VERA**

<b>INTEGRANTES EQUIPO #3</b>	<b>MATRÍCULA</b>
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 18 de octubre de 2022.**

# ÍNDICE

1. Objetivo.....	1
2. Definición de geometría .....	1
3. Marco teórico.....	3
4. Propuesta de diseño de la geometría, alcances y limitaciones. ....	5
5. Pasos del desarrollo de la programación.....	6
6. Resultados de la optimización .....	13
7. Conclusiones .....	17
8. Bibliografía .....	18
9. Anexo .....	18

# Práctica 4: Refuerzo del cable de un teleférico.

## 1. Objetivo

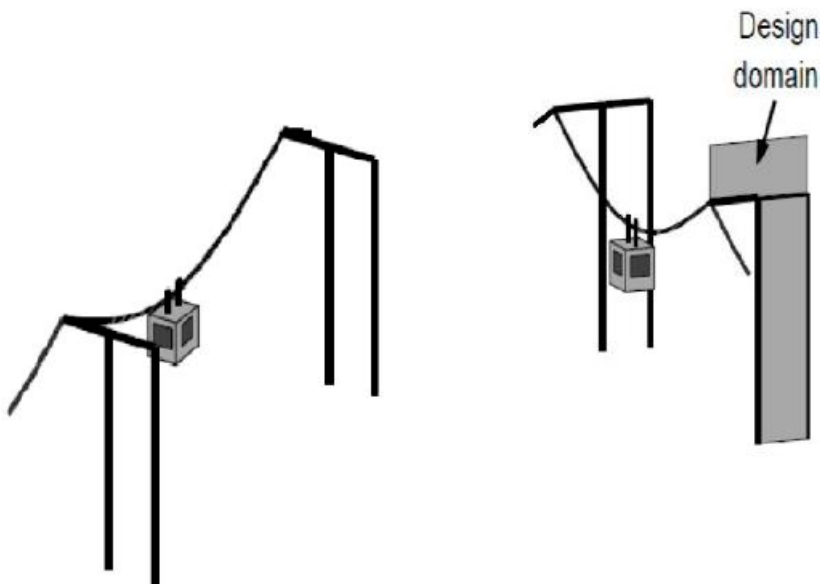
Desarrollar una propuesta de análisis de formas y programación para la ejecución de la optimización topológica para el refuerzo de un cable de un teleférico

## 2. Definición de geometría

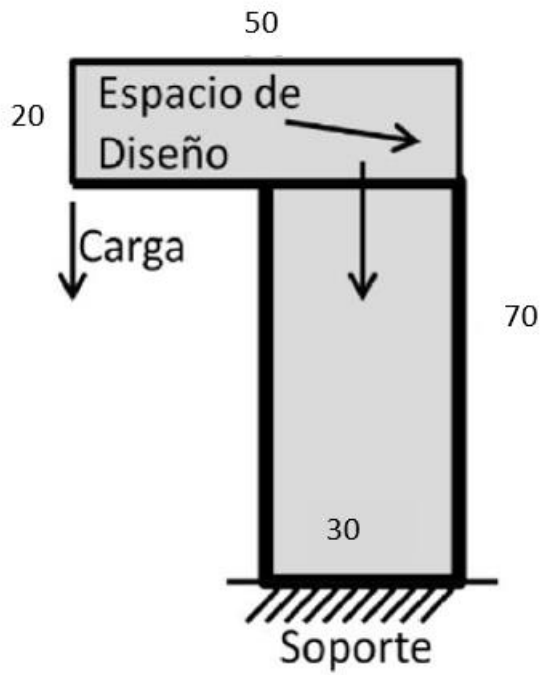
### Teleférico

Con la utilización de múltiples cargas, tenemos la realización del estudio el cual toma en consideración las implementaciones que este conlleva.

Cómo se observa en la imagen tenemos la implementación de un teleférico el cuál necesita por lo estudiado un refuerzo en su base con la información obtenida se sugerirá lo que es el refuerzo para este con la información de la segunda figura.

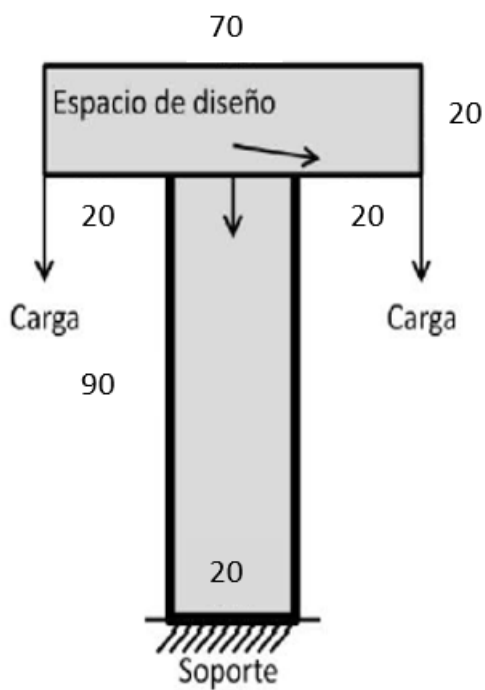


Espacio de diseño para 1 carga



También se quiere obtener mejoras para una estructura distinta la cual la cual está pueda llevar a su vez dos teleféricos cómo se muestra a continuación, con esta base se debe de considerar múltiples cargas.

Espacio de diseño para 2 cargas



### 3. Marco teórico

Los **teleféricos son un medio de transporte** utilizado desde hace muchos años. En algunos países como México se están instalando para facilitar la movilidad de los ciudadanos a lugares de difícil acceso para el transporte público. Entre sus **características principales**, destaca que dicho transporte se efectúa sobre vagonetas o cabinas que viajan suspendidas de un fuerte cable de acero.

Un teleférico consta de una estación de salida y otra de llegada, entre las que a veces hay otra estación intermedia. De una estación a la otra hay los siguientes cables:

- 1) El **cable portante**, que sostiene el peso de la cabina y que debe ser no sólo potente, sino sin juntas, es decir, de una sola pieza.
- 2) El **cable movable**, que está enganchado a la cabina y la arrastra.
- 3) El **cable de freno**, que aguanta la cabina en caso de que el cable movable se rompa.
- 4) El **cable de socorro**, que, en dicha circunstancia de ruptura, lleva la cabina hasta la estación.



Elementos de teleféricos

**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.



**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.

**Las pilonas:** Las pilonas deben soportar el peso del vehículo con los pasajeros y, por este motivo, su construcción es robusta. Los vehículos pueden circular por ambos lados de la pilona. Las pilonas individuales están formadas por una combinación de tubos de acero de diferente longitud, diámetro y espesor de la pared. En función de la morfología del terreno, las pilonas se transportarán hasta el área de la obra en helicóptero y se montarán allí. Las pilonas especiales son pilonas con una altura de más de 30 metros, construidas como pilonas tubulares divididas o como soportes en celosía.



**Los balancines:** Los balancines sirven para dirigir el cable portante a lo largo de la línea. Cada balancín está compuesto por una disposición de poleas. El número de poleas depende del peso que el cable debe transportar. Cada polea está compuesta por un cuerpo base, el anillo giratorio y la rueda con bridas.



**El cable:** El cable da nombre a todos los sistemas de transporte por cable, los teleféricos. Los cables de acero están compuestos de hilos de cable que se retuercen alrededor del núcleo del cable. Empresas especializadas fabrican los cables y los montan en el lugar.

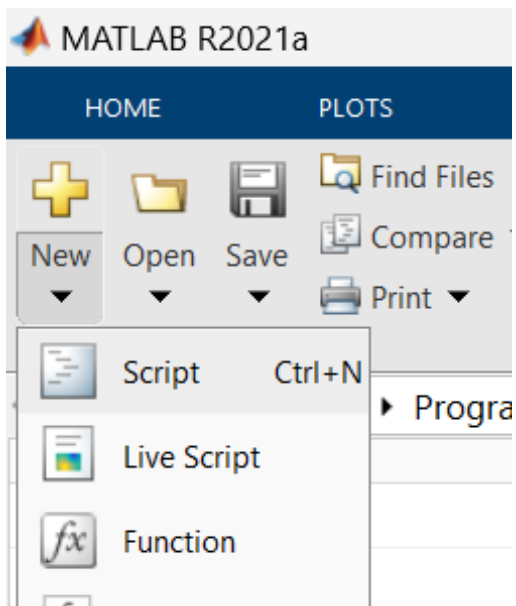
#### **4. Propuesta de diseño de la geometría, alcances y limitaciones.**

Para la propuesta de diseño de esta geometría, es necesario considerar sus puntos débiles o deficientes de la estructura y en estos centrarnos, analizarlos para optimizar las estructuras de un cable teleférico. De igual manera es importante considerar las circunstancias a las que está expuesto, como la cantidad de cargas que puede soportar, fenómenos naturales, cambios climáticos, etc.

De la mejor manera es necesario considerar que sea una estructura firme y simétrica como soporte de este cable.

## 5. Pasos del desarrollo de la programación.

5.1 En primera instancia, es necesario abrir un nuevo archivo dentro del software Matlab para el desarrollo de la optimización. Para ellos, hay que seleccionar la opción de “Script” en la pestaña de “New”.

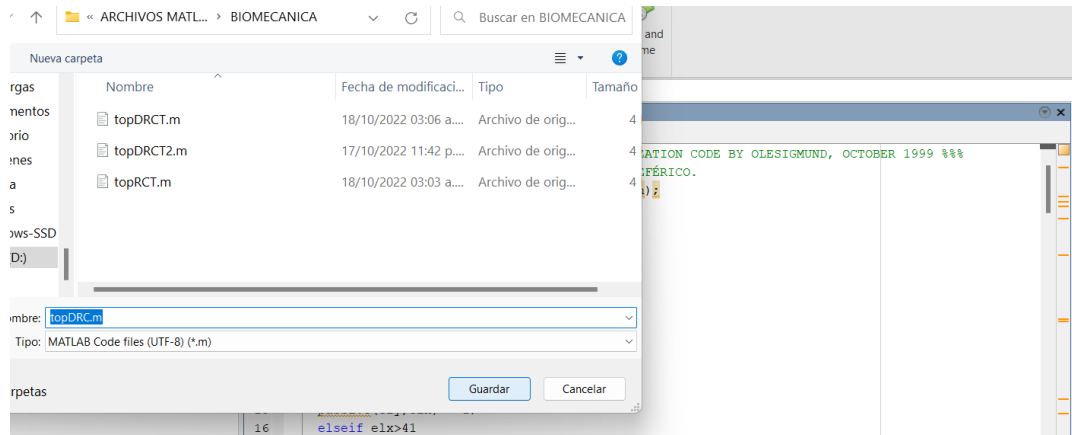


5.2 Se realiza la creación del programa principal donde se configuran los parámetros necesarios para la optimización respecto al refuerzo del cable del teleférico

```
1  %%% MODIFICACIÓN A 99 LINE TOPOLOGY OPTIMIZATION CODE BY OLESIGMUND, OCTOBER 1999 %%%
2  %%% PRÁCTICA 4.REFUERZO DEL CABLE DE UN TELEFÉRICO.
3  function topDRC(nelx,nely,volfrac,penal,rmin);
4  nelx=70;
5  nely=90;
6  volfrac=0.5;
7  penal=3.0;
8  rmin=1.5;
9  % INITIALIZE
10 x(1:nely,1:nelx) = volfrac;
11 for ely = 1:nely
12     for elx = 1:nelx
13         if ely>21
14             if elx<21
15                 passive(ely,elx) = 1;
16             elseif elx>41
17                 passive(ely,elx)=1;
18             else
19                 passive(ely,elx) = 0;
20             end
21         end
22     end
23 end
```



**5.3 Para que el software nos permita ejecutar el programa, y que a su vez, no corra riesgo de ser eliminado, resulta imprescindible guardarlo.**



**5.4 Del archivo original, se realizan las modificaciones pertinentes para la inicialización del programa, primero, con una sola carga.**

```

1  %%%% MODIFICACIÓN A 99 LINE TOPOLOGY OPTIMIZATION CODE BY OLESIGMUND, OCTOBER 1999 %%%
2  %%% PRÁCTICA 4.REFUERZO DEL CABLE DE UN TELEFÉRICO.
3  function topDRC(nelx,nely,volfrac,penal,rmin);
4  - nelx=70;
5  - nely=90;
6  - volfrac=0.5;
7  - penal=3.0;
8  - rmin=1.5;
9  % INITIALIZE
10 x(1:nely,1:nelx) = volfrac;
11 for ely = 1:nely
12 for elx = 1:nelx
13     if ely>21
14     if elx<21
15         passive(ely,elx) = 1;
16     elseif elx>41
17         passive(ely,elx)=1;
18     else
19         passive(ely,elx) = 0;
20     end
21     end
22 end
23 end
24 - x(find(passive))=0.001;
25 - loop = 0; change = 1.;
26 % START ITERATION

```

```

47 %25 FILTERING OF SENSITIVITIES
48 - [dc] = check(nelx,nely,rmin,x,dc);
49 %27 DESIGN UPDATE BY THE OPTIMALITY CRITERIA METHOD
50 - [x] = OC(nelx,nely,x,volfrac,dc,passive);
51 %29 PRINT RESULTS
52 - change = max(max(abs(x-xold)));
53 - disp([' It.: ' sprintf('%4i',loop) ' Obj.: ' sprintf('%10.4f',c) ...
54 'Vol.: ' sprintf('%6.3f',sum(sum(x))/(nelx*nely)) ...
55 ' ch.: ' sprintf('%6.3f',change )])
56 % PLOT DENSITIES
57 - colormap(gray); imagesc(-x); axis equal; axis tight; axis off; pause(1e-6);
58 - end
59 %40 %%%%%%%%%% OPTIMALITY CRITERIA UPDATE %%%%%%%%%%
60 - function [xnew]=OC(nelx,nely,x,volfrac,dc,passive)
61 - l1 = 0; l2 = 100000; move = 0.2;
62 - while (l2-l1 > 1e-4)
63 -     lmid = 0.5*(l2+l1);
64 -     xnew = max(0.001,max(x-move,min(1.,min(x+move,x.*sqrt(-dc./lmid)))));
65 -     xnew(find(passive)) = 0.001;
66 -     if sum(sum(xnew)) - volfrac*nelx*nely > 0;
67 -         l1 = lmid;
68 -     else
69 -         l2 = lmid;
70 -     end
71 - end

```

```

101 % DEFINE LOADSAND SUPPORTS (HALF MBB-BEAM)
102 - F(40,1) = -1.; F(9760,2)=1.;
103 - fixeddofs = 2*(nely+1):2*(nely+1):2*(nelx+1)*(nely+1);
104 - alldofs = [1:2*(nely+1)*(nelx+1)];
105 - freedofs = setdiff(alldofs,fixeddofs);
106 % SOLVING 127
107 - U(freedofs,:) = K(freedofs,freedofs) \ F(freedofs,:);
108 - U(fixeddofs,:)= 0;

```

**5.5** Ahora, como se solicita en el planteamiento, se realiza un análisis con 2 cargas, para ellos se considera el procedimiento para múltiples cargas. Y en base, al programa anterior, solo es necesario modificar los parámetros de las cargas

```

1      %%% A 99 LINE TOPOLOGY OPTIMIZATION CODE BY OLE SIGMUND, OCTOBER 1999 %%%
2      %%% Modificación de código para el Refuerzo del cable de un Teleférico lCarga
3      %%% topDRC1(50,70,0.33,3.0,1.2)
4      function topDRCT(nelx,nely,volfrac,penal,rmin);
5      - nelx=50;
6      - nely=70;
7      - volfrac=0.33;
8      - penal=3.0;
9      - rmin=1.2;
10     % INITIALIZE
11     x(1:nely,1:nelx) = volfrac;
12     for ely = 1:nely
13     for elx = 1:nelx
14     if ely>21
15     if elx<31
16     passive(ely,elx) = 1;
17     else
18     passive(ely,elx) = 0;
19     end
20     end
21     end
22     end
23     x(find(passive))=0.001;
24     loop = 0; change = 1.;

58     %%%%%%%%%% OPTIMALITY CRITERIA UPDATE %%%%%%%%%%
59     function [xnew]=OC(nelx,nely,x,volfrac,dc,passive)
60     - l1 = 0; l2 = 100000; move = 0.2;
61     while (l2-l1 > 1e-4)
62     - lmid = 0.5*(l2+l1);
63     - xnew = max(0.001,max(x-move,min(1.,min(x+move,x.*sqrt(-dc./lmid)))));
64     - xnew(find(passive))=0.001;
65     - if sum(sum(xnew)) - volfrac*nelx*nely > 0;
66     - l1 = lmid;
67     - else
68     - l2 = lmid;
69     - end
70     - end

```

```

101 % DEFINE LOADSAND SUPPORTS (HALF MBB-BEAM)
102 - F(40,1) = -1;
103 - fixeddofs = 2*(nely+1):2*(nely+1):2*(nelx+1)*(nely+1);
104 - alldofs = [1:2*(nely+1)*(nelx+1)];
105 - freedofs = setdiff(alldofs,fixeddofs);
106 % SOLVING
107 - U(freedofs,:) = K(freedofs,freedofs) \F(freedofs,:);
108 - U(fixeddofs,:)= 0;

```

5.6 Ahora, se verá de manera más detallada, el contenido del programa modificado para el refuerzo del teleférico. Primero se declara la función en MATLAB y se ejecuta el programa con el comando:

```

PANORAMICO%%
function topDP(nelx,nely,volfrac,penal,rmin);
-

```

5.7 Se encuentra primero, el inicio del programa. Después, se declara la iteración del programa para que comience a ejecutarse:

```

% 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
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);
-

```

5.8 Se declara la función objetivo y acto seguido, se realiza el análisis de sensibilidad:

```

% 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
-

```

**5.9 Luego, se coloca la optimización del modelo, se imprimen los resultados y se agrega el diseño topológico del trabajo:**

```

% 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
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
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

```

**5.10 Para esta fase, se coloca un filtro independiente para cada función. Aquí, se representa la implementación de Matlab:**

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% 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

```

**5.11** Analizamos nuestro modelo mediante el método de elementos finitos y se definen los soportes del mismo:

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% 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;

```

**5.12** Por último se obtiene la matriz de rigidez del modelo:

```

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

```

Figure 3-2

## 6. Resultados de la optimización

### Figura con 1 carga

Se obtiene el resultado a través de la simulación obtenida en Matlab, primeramente, se realizó y modifiqué el código de la práctica anterior, por lo tanto, al primer resultado se le aplica únicamente una carga, por ello el código es más corto y más eficiente. El código arrojaba advertencias, esto se debe a cambios internos del programa de Matlab, el cual solamente se hace caso omiso a las mismas, con esto dado por entendido, se arroja la simulación de la primera pieza con 1 carga.

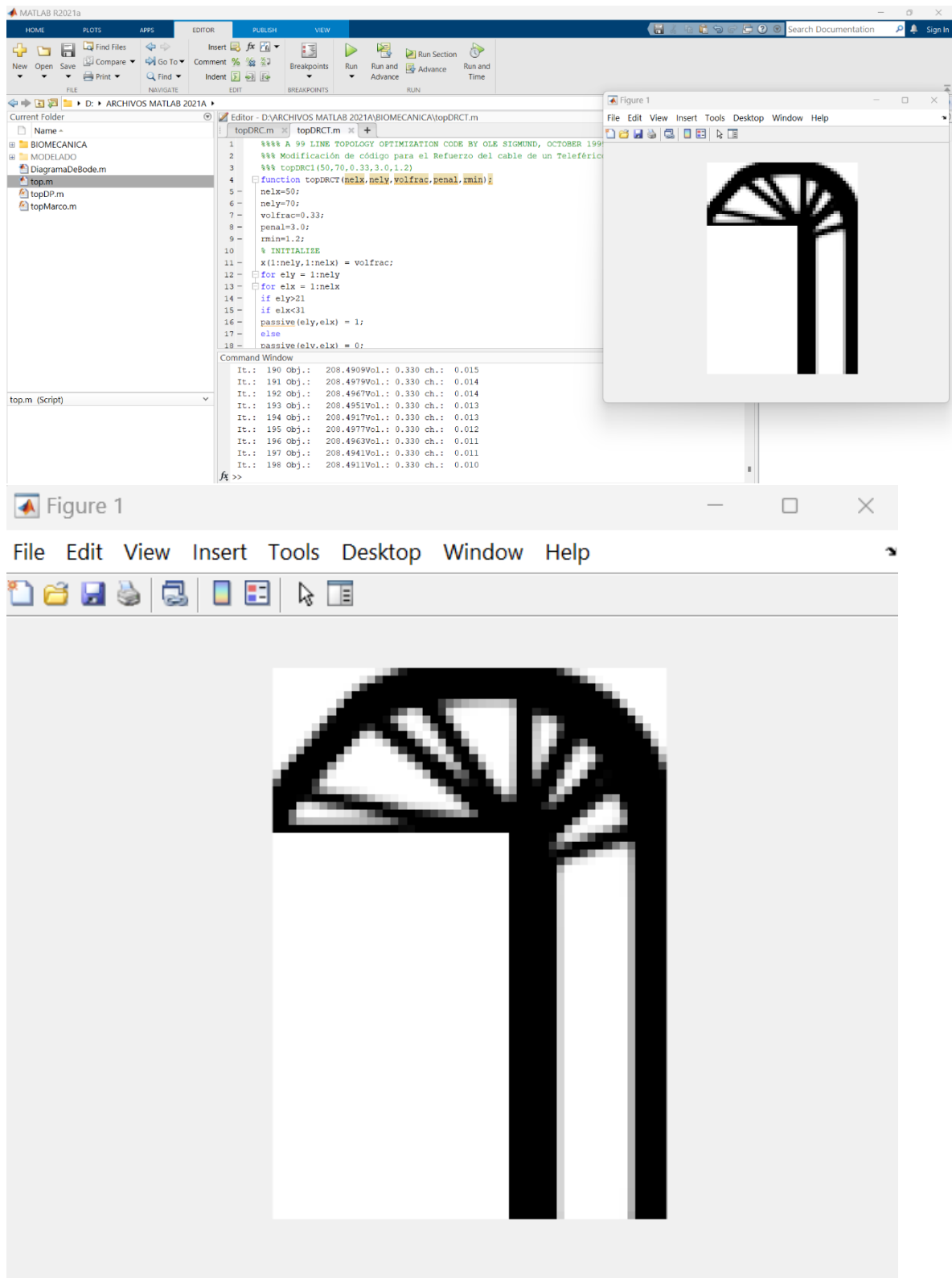
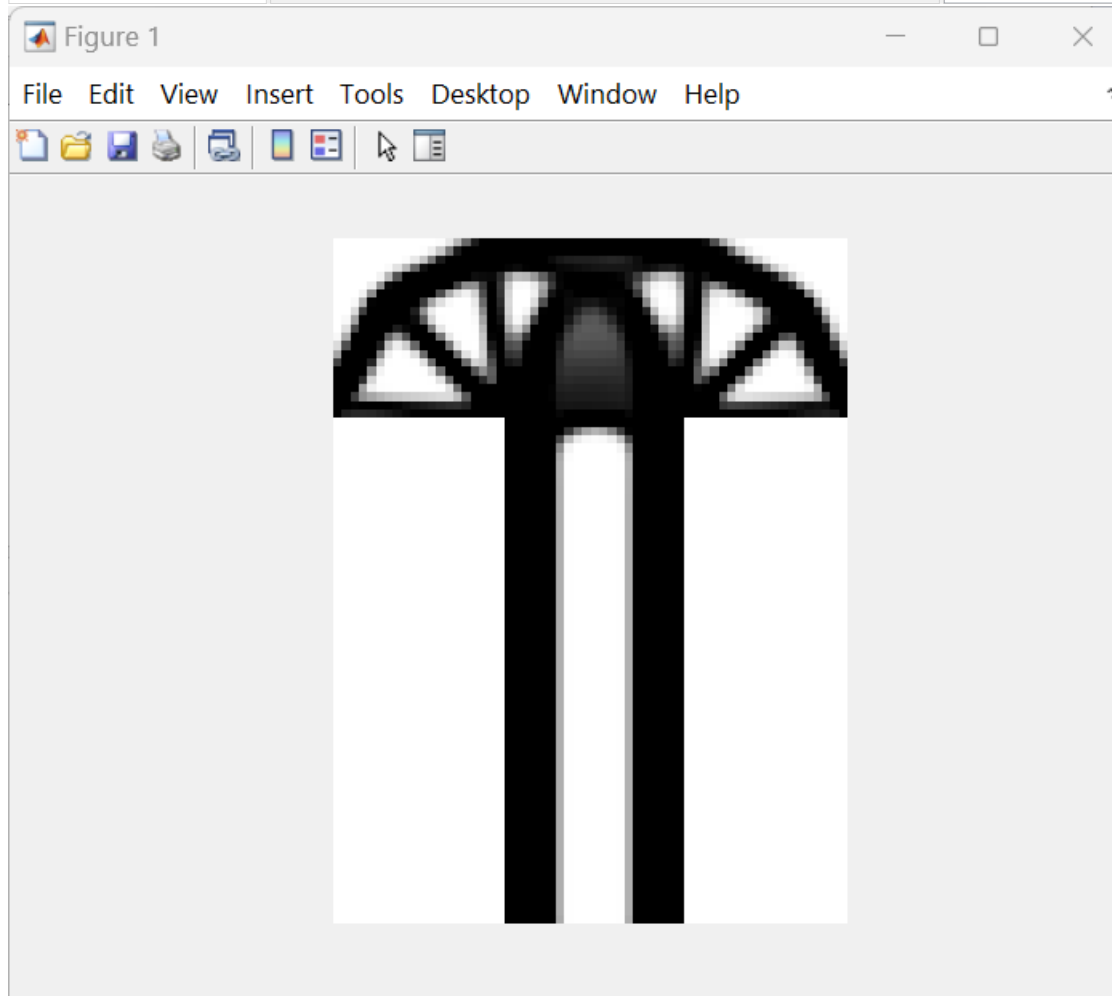
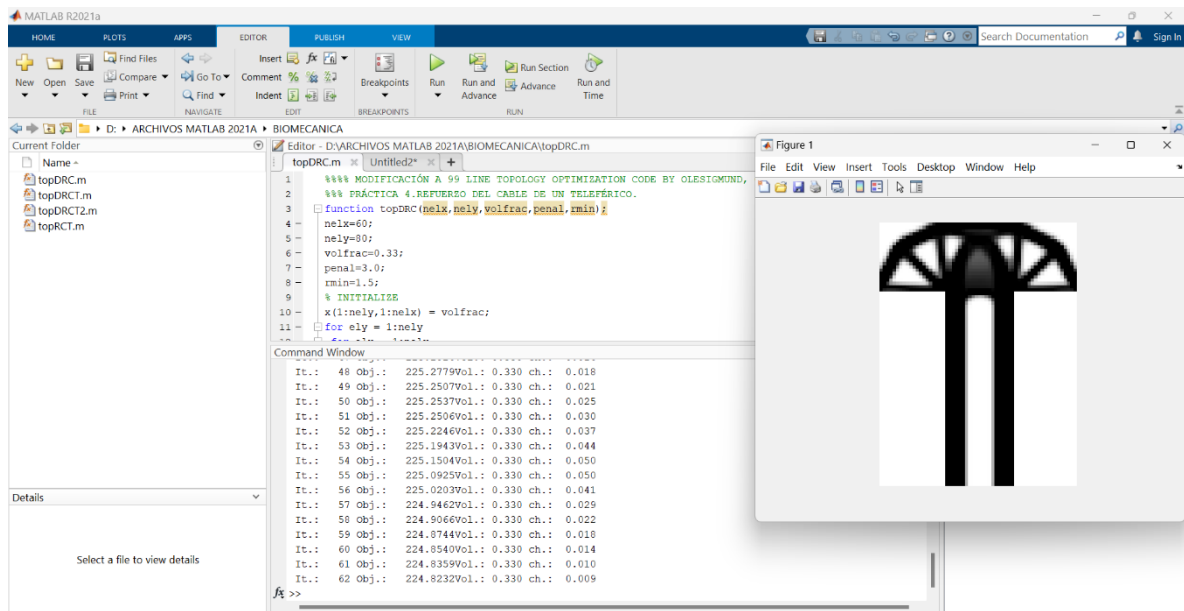


Figura con 2 cargas



Se muestra el código que se realizó para la figura con 1 carga, pero en este caso, se añadirá nuevamente la misma codificación para cumplir con el criterio de 2 cargas en la figura 2, de igual forma, nos arroja advertencias, en las cuales se hace caso omiso de ellas y nos muestra el resultado de la figura.

La diferencia de ambas figuras además de la fuerza es la optimización que nos arroja como resultado, ya que, las cargas aplicadas en el código su sumamente las mismas, pero se da que es aplicada una más en la figura 2, pero el resultado más optimo es el que se le aplican las 2 cargas, ya que soporta el doble de carga sin que llegue a tener una fractura por desgaste.



## 7. Conclusiones

### Diana Elisa Acosta Rodríguez:

Para esta práctica, se ha realizado un trabajo más complejo. Sin embargo, guarda cierta similitud con el panorámico debido a las cargas múltiples. Y se retoman conceptos que se han visto en prácticas anteriores. Pero, a pesar de ello, hubo algunas complicaciones, al momento de ejecutar el programa, debido al código tan extenso y, por ende, la posibilidad de equivocarse en algún carácter era mayor. De igual manera, la solicitud del planteamiento, la cual, era reforzar el cable de un teleférico, se puede realizar, reforzando las pilonas de acuerdo a su espacio de trabajo, haciendo uso del procedimiento de optimización topológica y el método de análisis de elementos finitos. De este modo, se logró llegar a un resultado óptimo.

### Joaquín Abdiel Coronado Vázquez:

La optimización de un cable teleférico es ampliamente importante, ya que se para la aplicación en la vida real, se deben de tener en cuenta varios factores para su diseño, en el cual no solo aplica la carga, sino también las tensiones y las masas de carga que puede aumentar o disminuir dependiendo de cada persona que suba, por ende es importante tomar en cuenta los 2 casos de esta práctica, porque se pueden demostrar las deformaciones y esfuerzos que son necesarios para que sea llevado a cabo en un caso real.

### Brenda Giselle Hinojosa:

En esta cuarta practica se realizó la optimización de un cable de teleférico, esto por medio de MatLab como en las practicas anteriores. Pudimos observar cómo al aplicar varias cargas que tiene el cable obteníamos ciertos esfuerzos y deformaciones, con esto se compararon los resultados para observar cómo se comportaban las cargas y así poder obtener la información necesaria para ver como interactúan con el teleférico, específicamente con el cable y con todos estos resultados poder mejorar el diseño y reforzarlo para así evitar problemas cuando se realice un diseño físico.

### Gloria Rosalía Domínguez Azueta:

Con la ayuda de Matlab, en esta práctica se logró realizar la optimización topológica para el reforzamiento del cableado de un teleférico, mejorando la configuración de la pieza y optimizando algunas características importantes para su desempeño final. De esta pieza, observé que el tiempo para realizarla fue mayor a las anteriores, debido a la optimización de los esfuerzos y a los elementos pasivos que fueron tomados en cuenta para el diseño. Por su parte con esta optimización se trató de obtener el menor peso posible aumentando

la rigidez y resistencia del cableado para el soporte del teleférico, con esta aplicación 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 el producto.

Andrea Anette Celestino Castillo:

Con la realización de esta práctica puedo mencionar que ha sido un poco más entendible su realización debido a que ya teníamos un conocimiento del programa como también de la base que se iba a realizar, puedo de igual modo mencionar que tuvimos unos pequeños problemas en la realización del programa de este pero pudimos darle solución a este para completar y finalizar a tiempo la práctica.

Ana Paulina Pérez Salazar:

En mi opinión esta práctica no hubo problema alguno para la realización de esta, fue muy interesante la elaboración de la optimización del cable de un teleférico ya que muestra cada uno de los análisis, es muy importante esto ya que nos ayudó para comprender los temas vistos en clase

Javier Cisneros Saldivar:

En esta ocasión, se puso en práctica la optimización de la forma con un panorámico, fue muy interesante ya que este método es muy utilizado para poder hacer la forma de nuestra modelo mucho más generativa que como lo era antes.

## **8. Bibliografía**

- *Elementos de teleféricos* - LEITNER. (s. f.). Recuperado 17 de octubre de 2022, de <https://www.leitner.com/es/empresa/informaciones-utiles/elementos-de-telefericos/>
- Ciencia, C. (2022, 3 febrero). *Qué es un teleférico*. CurioSfera Ciencia. Recuperado 17 de octubre de 2022, de <https://curiosfera-ciencia.com/que-es-un-teleferico-como-funciona/>
- 99 Line Topology Optimization Code – O. Sigmund, Department of Solid Mechanics, Building 404, Technical University of Denmark, DK-2800 Lyngby, Denmark.

## **9. Anexo**

**Código en Matlab para el diseño del refuerzo del cable de un teleférico.**

**Figura con 1 carga.**

```

%%% A 99 LINE TOPOLOGY OPTIMIZATION CODE BY OLE
SIGMUND, OCTOBER 1999 %%%
%%% Modificación de código para el Refuerzo del cable de
un Teleférico lCarga
%%% topDRC1(50,70,0.33,3.0,1.2)
function topDRCT(nelx,nely,volfrac,penal,rmin);
nelx=50;
nely=70;
volfrac=0.33;
penal=3.0;
rmin=1.2;
% 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
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
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) ];
```

### Figura con 2 cargas

```
%%% MODIFICACIÓN A 99 LINE TOPOLOGY OPTIMIZATION CODE
BY OLESIGMUND, OCTOBER 1999 %%%
%%% PRÁCTICA 4.REFUERZO DEL CABLE DE UN TELEFÉRICO.
function topDRC(nelx,nely,volfrac,penal,rmin);
nelx=60;
nely=80;
volfrac=0.33;
penal=3.0;
rmin=1.5;
% INITIALIZE
x(1:nely,1:nelx) = volfrac;
for ely = 1:nely
    for elx = 1:nelx
        if ely>21
            if elx<21
                passive(ely,elx) = 1;
            elseif elx>41
                passive(ely,elx)=1;
            else
                passive(ely,elx) = 0;
            end
        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.;
        end
    end
end
```



```

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
%25 FILTERING OF SENSITIVITIES
[dc] = check(nelx,nely,rmin,x,dc);
%27 DESIGN UPDATE BY THE OPTIMALITY CRITERIA METHOD
[x] = OC(nelx,nely,x,volfrac,dc,passive);
%29 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
%40 %%%%%%%%% 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
dcn(j,i) = dcn(j,i)/(x(j,i)*sum);
end
end
%65 %%%%%%%%% 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
=zeros(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(40,1) = -1.; F(9760,2)=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 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)];  
%topDRC(60,80,0.33,3.0,1.5)
```