



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



Laboratorio de Biomecánica
Práctica 4: Refuerzo del cable de un teleférico.

Catedrático: Yadira Moreno Vera

Martes V2

Brigada 214

Integrantes del Equipo #1	Matricula
Susana Rubio Medina	1798151
Melissa Lizeth Galindo Reyes	1856086
Oziel Alberto Torres Villarreal	1900260
Gabriel Eduardo Vázquez Ortega	1903060
José Manuel Reséndiz García	1907334
Miguel Ángel Martínez Villanueva	1934489

Semestre Agosto – Diciembre 2022

San Nicolás de los Garza, N.L.

01 de Noviembre de 2022

Definición

Es un medio de transporte que consiste en cabinas con capacidad para llevar un grupo de personas. Estas cabinas viajan suspendidas en el aire transportadas por uno o varios cables. La mayoría de estos medios de transporte son accionados por energía eléctrica. Este transporte se usa en zonas con grandes diferencias de altura, donde el acceso por carretera o ferrocarril resulta difícil.



El sistema de cada teleférico está compuesto por uno o más cables (dependiendo del tipo). El primer cable está fijo y sirve para sostener las cabinas, el segundo está conectado a un motor (ubicado en la estación) y hace mover las cabinas. Algunos teleféricos usan dos cabinas por tramo (trayecto entre estación y estación) a fin de crear un contrapeso. Otros sistemas más complejos tienen varias cabinas suspendidas simultáneamente en cada dirección

Estado del Arte

Las primeras civilizaciones que hicieron uso del transporte por cable fueron las orientales y la antigua civilización inca de Perú, en Europa no aparece este tipo de transporte hasta la Edad Media, a principios del siglo XVI, empleándose para la construcción de castillos y fortificaciones. En aquella época el cable estaba fabricado por cuerdas de cáñamo y para su funcionamiento se utilizaba la tracción animal o humana.

A partir del año 1500 comienza a utilizarse el cable de acero y desde el siglo XVI al XIX se va perfeccionando su trenzado de hilos, contribuyendo al desarrollo del transporte por cable, principalmente para su uso en las minas de carbón. A partir de los años 30 del siglo XX proliferan los teleféricos con fines turísticos en lugares de montaña, sustituyendo en muchos casos a los funiculares y los trenes cremallera. Sin embargo, es en las décadas de los 60 y los 70 cuando se produce el mayor desarrollo de este tipo de transporte, impulsado principalmente por el auge del esquí y otros deportes de montaña.

Desde entonces hasta ahora los teleféricos han experimentado una gran evolución, construyéndose, hoy, instalaciones cada vez más modernas. En su diseño, además de garantizar el buen funcionamiento y la seguridad, se da gran importancia a la innovación, la estética y la comodidad.

Los teleféricos actuales se utilizan principalmente con tres fines:

- Teleféricos destinados a los deportes de montaña, entre los que destacan los deportes de invierno
- Teleféricos turísticos en lugares con vistas panorámicas, con el fin de facilitar el disfrute de espacios inaccesibles con otros medios de transporte
- Transporte urbano en ciudades con grandes desniveles

Propuestas del diseño de la geometría.

El cable es el encargado de soportar y/o transportar los vehículos. El cable está dimensionado para cada instalación, para que soporte el peso de las cabinas con carga máxima.



El cable está formado por un conjunto de hilos de acero de este o diferente tamaño trenzados de tal forma que generan otros cables que a su vez se trenzan alrededor del alma formando el cable del teleférico.

El centro del cable se le denomina alma y está formado normalmente por un plástico flexible que se adapta a los

hilos del cable.

El cable de un teleférico es continuo, teniendo las dos puntas unidas en el empalme del cable. Este empalme se crea en diferentes nudos. Los empalmes de los cables pueden llegar a medir varias decenas de metros.

Los cables tienden a alargarse, sobre todo cuando son nuevos, cada instalación tiene un margen de alargado que al estar próximo o en el límite de estirado del cable se tiene que acortar el cable.

La forma en como el cable es colocando es con un cierto grado de inclinación le cual le facilita la movilidad para transportar las personas, siendo que el cable esta reforzado siendo que se compone de varios cables de acero entrelazados de la misma forma en como una cuerda está hecha, siendo que las limitaciones que el diseño puede tener es que la velocidad que tendrá el teleférico será lenta, pero esto se debe a que para recorrer el cable de subida es necesario de tensores que irán tensando el cable arrastrando el cable y atrayendo el teleférico de un punto a otro sin poder hacerlo de forma rápida para disminuir los riesgos de alta fricción que puedan desgastar el cable, por otro lado esta forma de utilizar la pendiente para transportar de un lado a otro es algo útil debido a que es la ruta más corta, pese a ser un movimiento relativamente lento es un trayecto más corto, siendo que el cable esta reforzado es seguro mientras no se exceda el límite de peso permitido para que así la tensión en el cable no sea extrema, de misma forma recomendando disminuir el movimiento dentro del teleférico para que estos no perturben el equilibrio del mismo.

DESARROLLO DEL CÓDIGO DE LA PRÁCTICA

Para realizar este ejercicio o problema propuesto tendremos que primero cambiar algunas cosas de nuestro código antes de correrlo con `topp1(40,80,0.2,3.0,5)`

Primeramente el vacío del código que ya teníamos se modificara a

```
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
```

Después de esto tendremos que modificar el vacío de otra optimización que está dentro del mismo código solo cambiando unas partes del mismo como estas:

```
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
```


MATLAB R2021a

HOME PLOTS APPS EDITOR PUBLISH VIEW

New Open Save Find Files Compare Print Find

Insert Comment Indent Breakpoints Pause Run and Advance Run and Time

Current Folder: D:\matlab2.0\bin

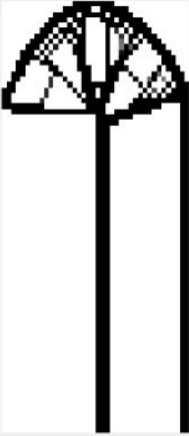
Workspace: topp4.m

Command Window

Editor - 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<21
9                 passive(ely,elx) = 1;
10            elseif elx>41
11                passive(ely,elx)=1;
12            else
13                passive(ely,elx) = 0;
14            end
15        end
16    end
17 end
18 x(find(passive))=0.001;
19 loop = 0; change = 1.;
20 % START ITERATION
21 while change > 0.01
22     loop = loop + 1;
23     xold = x;
24     % FE-ANALYSIS
25     [U]=FE(nelx,nely,x,penal);
26     % OBJECTIVE FUNCTION AND SENSITIVITY ANALYSIS
27     [KE] = 1k;
28     c = 0.;
29     for ely = 1:nely
30         for elx = 1:nelx
31             n1 = (nely+1)*(elx-1)+ely;
32             n2 = (nely+1)* elx +ely;
33             dc(ely,elx)=0.;
34         end
35     end
```

Figure 1



MATLAB R2021a

HOME PLOTS APPS EDITOR PUBLISH VIEW

New Open Save Find Files Compare Go To Find Indent Comment % Breakpoints Continue Step Step In Step Out Run to Cursor Function Call Stack: FE Quit Debugging

FILE NAVIGATE EDIT BREAKPOINTS DEBUG

Current Folder D:\matlab2.0\bin

Workspace Command Window

topp4.m

```

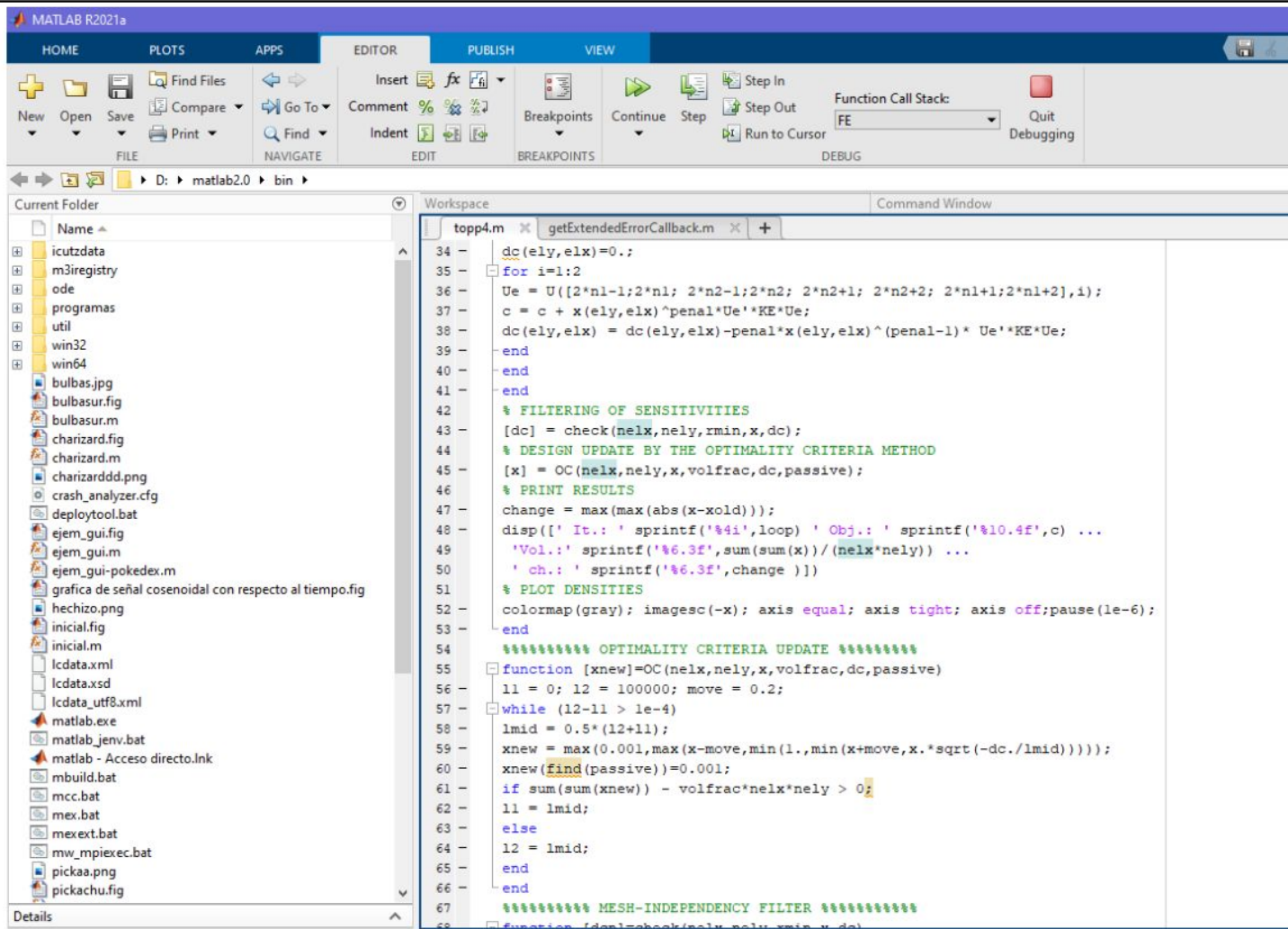
1  %%% A 99 LINE TOPOLOGY OPTIMIZATION CODE BY OLE SIGMUND, OCTOBER 1999 %%%
2  function new_pr42_f(nelx,nely,volfrac,penal,rmin)
3  % INITIALIZE
4  x(1:nely,1:nelx) = volfrac;
5  for ely = 1:nely
6  for elx = 1:nelx
7  if ely>21
8  if elx<21
9  passive(ely,elx) = 1;
10 elseif elx>41
11 passive(ely,elx)=1;
12 else
13 passive(ely,elx) = 0;
14 end
15 end
16 end
17 end
18 x(find(passive))=0.001;
19 loop = 0; change = 1.;
20 % START ITERATION
21 while change > 0.01
22 loop = loop + 1;
23 xold = x;
24 % FE-ANALYSIS
25 [U]=FE(nelx,nely,x,penal);
26 % OBJECTIVE FUNCTION AND SENSITIVITY ANALYSIS
27 [KE] = lk;
28 c = 0.;
29
30 for ely = 1:nely
31 for elx = 1:nelx
32 n1 = (nely+1)*(elx-1)+ely;
33 n2 = (nely+1)* elx +ely;
34 dc(ely,elx)=0.;
35

```

Details

8 occurrences of "nelx" found

UTF-8



MATLAB R2021a

HOME PLOTS APPS EDITOR PUBLISH VIEW

New Open Save Find Files Compare Go To Find Indent Comment % Breakpoints Continue Step Step In Step Out Run to Cursor Function Call Stack: FE Quit Debugging

FILE NAVIGATE EDIT BREAKPOINTS DEBUG

Current Folder: D:\matlab2.0\bin

Workspace: Command Window Editor - topp4.m

topp4.m

```

67 %***** MESH-INDEPENDENCY FILTER *****
68 function [dcn]=check(nelx,nely,rmin,x,dc)
69 dcn=zeros(nely,nelx);
70 for i = 1:nelx
71 for j = 1:nely
72 sum=0.0;
73 for k = max(i-round(rmin),1): min(i+round(rmin),nelx)
74 for l = max(j-round(rmin),1): min(j+round(rmin), nely)
75 fac = rmin-sqrt((i-k)^2+(j-l)^2);
76 sum = sum+max(0,fac);
77 dcn(j,i) = dcn(j,i) + max(0,fac)*x(1,k)*dc(1,k);
78 end
79 end
80 dcn(j,i) = dcn(j,i)/(x(j,i)*sum);
81 end
82 end
83 %***** FE-ANALYSIS *****
84 function [U]=FE(nelx,nely,x,penal)
85 [KE] = 1k;
86 K = sparse(2*(nelx+1)*(nely+1), 2*(nelx+1)*(nely+1));
87 F = sparse(2*(nely+1)*(nelx+1),2); U = sparse(2*(nely+1)*(nelx+1),2);
88 for ely = 1:nely
89 for elx = 1:nelx
90 n1 = (nely+1)*(elx-1)+ely;
91 n2 = (nely+1)* elx +ely;
92
93 edof = [2*n1-1; 2*n1; 2*n2-1; 2*n2; 2*n2+1; 2*n2+2; 2*n1+1; 2*n1+2];
94 K(edof,edof) = K(edof,edof) + x(ely,elx)^penal*KE;
95 end
96 end
97 % DEFINE LOADSAND SUPPORTS (HALF MBS-BEAM)
98 F(40,1) = -1.; F(9760,2)=1.;
99 fixeddofs = 2*(nely+1):2*(nely+1):2*(nelx+1)*(nely+1);
100 alldofs = [1:2*(nely+1)*(nelx+1)];
101 % U=FE(alldofs,fixeddofs,F,KE);

```

8 occurrences of "nelx" found

UTF-8 new_pr42_f Ln 6 Col 18

```

MATLAB R2021a
HOME PLOTS APPS EDITOR PUBLISH VIEW
New Open Save Find Files Compare Go To Find Indent Comment Insert Breakpoints Continue Step Step In Step Out Run to Cursor Function Call Stack: FE Quit Debugging
D:\matlab2.0\bin
Current Folder: D:\matlab2.0\bin
Workspace: Command Window
Name
icutzdata
m3iregistry
ode
programas
util
win32
win64
bulbas.jpg
bulbasur.fig
bulbasur.m
charizard.fig
charizard.m
charizarddd.png
crash_analyzer.cfg
deploytool.bat
ejem_gui.fig
ejem_gui.m
ejem_gui-pokedex.m
grafica de señal cosenoidal con respecto al tiempo.fig
hechizo.png
inicial.fig
inicial.m
lcddata.xml
lcddata.xsd
lcddata_utf8.xml
matlab.exe
matlab_jenv.bat
matlab - Acceso directo.lnk
mbuild.bat
mcc.bat
mex.bat
mexext.bat
mw_mpixec.bat
pickaa.png
pickachu.fig
topp4.m
getExtendedErrorCallback.m
65 [KE] = 1k;
66 K = sparse(2*(nelx+1)*(nely+1), 2*(nelx+1)*(nely+1));
67 F = sparse(2*(nely+1)*(nelx+1), 2); U = sparse(2*(nely+1)*(nelx+1), 2);
68 for ely = 1:nely
69 for elx = 1:nelx
70 n1 = (nely+1)*(elx-1)+ely;
71 n2 = (nely+1)* elx +ely;
72 g
73 edof = [2*n1-1; 2*n1; 2*n2-1; 2*n2; 2*n2+1; 2*n2+2; 2*n1+1; 2*n1+2];
74 K(edof,edof) = K(edof,edof) + x(ely,elx)^penal*KE;
75 end
76 end
77 % DEFINE LOADSAND SUPPORTS(HALF MBB-BEAM)
78 F(40,1) = -1.; F(9760,2)=1.;
79 fixeddofs = 2*(nely+1):2*(nely+1):2*(nelx+1)*(nely+1);
100 alldofs = [1:2*(nely+1)*(nelx+1)];
101 freedofs = setdiff(alldofs,fixeddofs);
102 % SOLVING
103 U(freedofs,:) = K(freedofs,freedofs) \F(freedofs,:);
104 U(fixeddofs,:) = 0;
105 %***** ELEMENT STIFFNESS MATRIX *****
106 function [KE]=1k
107 E = 1.;
108 nu = 0.3;
109 k=[ 1/2-nu/6 1/8+nu/8 -1/4-nu/12 -1/8+3*nu/8 ...
110 -1/4+nu/12 -1/8-nu/8 nu/6 1/8-3*nu/8];
111 KE = E/(1-nu^2)* [ k(1) k(2) k(3) k(4) k(5) k(6) k(7) k(8)
112 k(2) k(1) k(8) k(7) k(6) k(5) k(4) k(3)
113 k(3) k(8) k(1) k(6) k(7) k(4) k(5) k(2)
114 k(4) k(7) k(6) k(1) k(8) k(3) k(2) k(5)
115 k(5) k(6) k(7) k(8) k(1) k(2) k(3) k(4)
116 k(6) k(5) k(4) k(3) k(2) k(1) k(8) k(7)
117 k(7) k(4) k(5) k(2) k(3) k(8) k(1) k(6)
118 k(8) k(3) k(2) k(5) k(4) k(7) k(6) k(1)];

```

También se aplicó otro refuerzo de otra manera teniendo así dos optimizaciones, realmente no cambian mucho los códigos entre estas dos optimizaciones y tan solo es la instrucción que se le da al mismo código para que empiece a reforzar el cable. Esta segunda optimización resultaría algo así

MATLAB R2021a

HOME PLOTS APPS EDITOR PUBLISH VIEW

New Open Save Find Files Compare Go To Comment Indent Breakpoints Pause Step Step In Step Out Run to Cursor Function Call Stack: FE Quit Debugging

FILE NAVIGATE EDIT BREAKPOINTS DEBUG

Current Folder: D:\matlab2.0\bin

Workspace: topp4a.m

Command Window: Editor - topp4a.m

```

function u = zz(nelx,nely,x,penal)
82 - [KE] = 1k;
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)*
85 - for ely = 1:nely
86 - for elx = 1:nelx
87 -
88 -
89 - n1 = (nely+1)*(elx-1)+ely;
90 - n2 = (nely+1)* elx +ely;
91 - edof = [2*n1-1; 2*n1; 2*n2-1; 2*n2; 2*n2+1; 2*n2+2; 2*n1+1;
92 - K(edof,edof) = K(edof,edof) + x(ely,elx)^penal*KE;
93 - end
94 - end
95 - % DEFINE LOADSAND SUPPORTS(HALF MBB-BEAM)
96 - F(40,1) = -1;
97 - fixeddofs = 2*(nely+1):2*(nely+1):2*(nelx+1)*(nely+1);
98 - alldofs = [1:2*(nely+1)*(nelx+1)];
99 - freedofs = setdiff(alldofs,fixeddofs);
100 - % SOLVING
101 - U(freedofs,:) = K(freedofs,freedofs) \ F(freedofs,:);
102 - U(fixeddofs,:) = 0;
103 - %***** ELEMENT STIFFNESS MATRIX %*****
104 - function [KE]=1k
105 - E = 1.;
106 - nu = 0.3;
107 - k=[ 1/2-nu/6 1/8+nu/8 -1/4-nu/12 -1/8+3*nu/8 ...
108 - -1/4+nu/12 -1/8-nu/8 nu/6 1/8-3*nu/8];
109 - KE = E/(1-nu^2)* [ k(1) k(2) k(3) k(4) k(5) k(6) k(7) k(8)
110 - k(2) k(1) k(8) k(7) k(6) k(5) k(4) k(3)
111 - k(3) k(8) k(1) k(6) k(7) k(4) k(5) k(2)
112 - k(4) k(7) k(6) k(1) k(8) k(3) k(2) k(5)
113 - k(5) k(6) k(7) k(8) k(1) k(2) k(3) k(4)
114 - k(6) k(5) k(4) k(3) k(2) k(1) k(8) k(7)
115 - k(7) k(4) k(5) k(2) k(3) k(8) k(1) k(6)
116 - k(8) k(3) k(2) k(5) k(4) k(7) k(6) k(1)];
  
```

Figure 1

MATLAB R2021a

HOME PLOTS APPS EDITOR PUBLISH VIEW

New Open Save Find Files Compare Go To Find Insert Comment Indent Breakpoints Continue Step Step In Step Out Run to Cursor Function Call Stack: FE Quit Debugging

FILE NAVIGATE EDIT BREAKPOINTS DEBUG

Current Folder D:\matlab2.0\bin

Workspace Command Window

topp4a.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
19 while change > 0.01
20 loop = loop + 1;
21 xold = x;
22 % FE-ANALYSIS
23 [U]=FE(nelx,nely,x,penal);
24 % OBJECTIVE FUNCTION AND SENSITIVITY ANALYSIS
25 %
26 [KE] = lk;
27 c = 0.;
28 for ely = 1:nely
29 for elx = 1:nelx
30 n1 = (nely+1)*(elx-1)+ely;
31 n2 = (nely+1)* elx +ely;
32 dc(ely,elx)=0.;
33 for i=1:2
34 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);
35 %

```


MATLAB R2021a

HOME PLOTS APPS EDITOR PUBLISH VIEW

New Open Save Find Files Compare Print Find Go To Insert Comment Indent Breakpoints Continue Step Run to Cursor Function Call Stack: FE Quit Debugging

Current Folder: D:\matlab2.0\bin

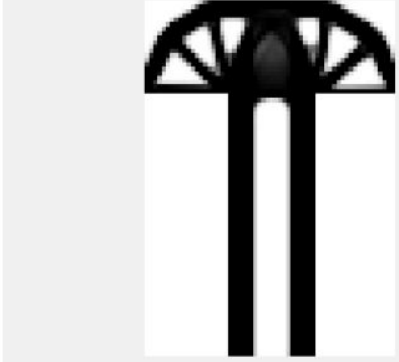
Workspace: Command Window: Editor - topp4s.m

File Edit View Insert Tools Desktop Window Help

```

34 - Ue = U([2*n1-1;2*n1; 2*n2-1;2*n2; 2*n2+1; 2*n2+2; 2*n1+1;2*n1+2],1);
35 - c = c + x(ely,elx)^penal*Ue'*KE*Ue;
36 - dc(ely,elx) = dc(ely,elx)-penal*x(ely,elx)^(penal-1)* Ue'*KE*Ue;
37 - end
38 - end
39 - end
40 - % FILTERING OF SENSITIVITIES
41 - [dc] = check(nelx,nely,rmin,x,dc);
42 - % DESIGN UPDATE BY THE OPTIMALITY CRITERIA METHOD
43 - [x] = OC(nelx,nely,x,volfrac,dc,passive);
44 - % PRINT RESULTS
45 - change = max(max(abs(x-xold)));
46 - disp([' It.: ' sprintf('%4i',loop) ' Obj.: ' sprintf('%10.4f',c) ...
47 - 'Vol.: ' sprintf('%6.3f',sum(sum(x))/(nelx*nely)) ...
48 - ' ch.: ' sprintf('%6.3f',change) ])
49 - % PLOT DENSITIES
50 - colormap(gray); imagesc(-x); axis equal; axis tight; axis off; pause(1e-6);
51 - end
52 - %~~~~~ OPTIMALITY CRITERIA UPDATE ~~~~~
53 - function [xnew]=OC(nelx,nely,x,volfrac,dc,passive)
54 - l1 = 0; l2 = 100000; move = 0.2;
55 - while (l2-l1 > 1e-4)
56 - lmid = 0.5*(l2+l1);
57 - xnew = max(0.001,max(x-move,min(1.,min(x+move,x.*sqrt(-dc./lmid))));
58 - xnew(find(passive))==0.001;
59 - if sum(sum(xnew)) - volfrac*nelx*nely > 0.2
60 - l1 = lmid;
61 - else
62 - l2 = lmid;
63 - end
64 - end
65 - %~~~~~ MESH-INDEPENDENCY FILTER ~~~~~
66 - function [dcn]=check(nelx,nely,rmin,x,dc)
67 - dcn=zeros(nely,nelx);
68 - for i = 1:nely

```



Conclusiones del equipo 1:

Susana Rubio Medina. – En esta práctica realizamos una propuesta de análisis de un cable de un teleférico además de la programación adecuada para realizar una optimización de esta. Es interesante ver cómo es que hemos ido avanzando de optimizar cosas cotidianas como la estructura de una bicicleta, hasta llegar a lo que realizamos que es el cable de un teleférico, el cual es una pieza importante y cualquier error puede llegar a afectar en accidentes graves e incluso costar vidas, es por eso que se realizan numerosas pruebas para tener una optimización correcta y poder mejorar día con día todo lo que sea posible y tengamos a nuestro alcance.

Melissa Lizeth Galindo Reyes. - De acuerdo con lo visto en esta práctica, podemos observar que utilizamos la misma metodología que las practicas anteriores donde definimos la forma geometría, estado del arte, alcances y limitaciones, en este caso enfocado en el refuerzo de un cable de un teleférico, aplicando la programación adecuada logramos con éxito el objetivo propuesto para esta práctica.

Oziel Alberto Torres Villarreal. – Pienso que un teleférico es una forma de transporte con bastantes riesgos y que es muy importante asegurarse que todo cumpla con lo debido ya que un error de cálculo pone directamente en peligro la vida de las personas. Para mi el cable es una de las partes más importantes del teleférico, incluso pienso que es una de las geometrías que se expone a un mayor número de variables como lo son la tensión, la temperatura, el peso del teleférico, el grosor del cable, etc. La práctica me pareció algo complicada porque la geometría ahora no era tan sencilla como las anteriores sin embargo los resultados nos indican que es posible realizar una optimización topológica en Matlab y así asegurarnos que todo este en su lugar.

Gabriel Eduardo Vázquez Ortega. – En esta práctica vimos lo que es la optimización topológica para la creación de un cable para sostener un teleférico, a diferencia de las otras prácticas, los objetos creados eran bastante sencillos o con una finalidad muy ligera, pero este primer objeto es ahora si muy importante, ya que estamos creando un objeto con el cual vamos a transportar decenas de personas, y en el caso de que este falle, sería una catástrofe, por eso hay que observar muy bien en su elaboración pero sobre todo el tipo de material con el cual se fabricara, para ver las diferentes propiedades, como la resistencia o elasticidad del material, en si la practica fue sencilla de elaborar y a la hora de simular se obtuvo la figura del cable de refuerzo.

José Manuel Reséndiz García. – En la práctica se vio del misma forma de trabajar, dónde se ve el mismo formato que estamos trabajando. La geometría no fue difícil y en el momento de realizarlo no se complica, ya que teníamos experiencia con los demás reporte. Por mi parte no tengo dudas al respecto

Miguel Ángel Martínez Villanueva. – En conclusión, a través de esta práctica logré cumplir de manera exitosa el propósito de esta, ya que se presentó una propuesta de análisis de forma con el propósito de diseño para reforzar el cable de un teleférico además de la programación realizada en el software Matlab para la ejecución de la optimización tomando así en cuenta las características de trabajo, las ventajas y desventajas de esta misma para un correcto diseño, análisis y simulación.

Bibliografía

- M., J. (2021, 29 octubre). *Autor de la entrada: Javi M.* TPC: Funiculares y teleféricos. Recuperado 17 de octubre de 2022, de <https://transporteporcable.com/como-se-fabrican-los-cables-de-los-funiculares-y-telefericos/>
- De Camino a La Cima TV. (2019, 6 enero). *Cambio y trenzado del cable de un telesilla o remonte.* [Vídeo]. YouTube. Recuperado 17 de octubre de 2022, de <https://www.youtube.com/watch?v=JaNbOYpkSGU>
- Deutsche Welle (www.dw.com). (s. f.). *Teleférico* / DW / 24.08.2022. DW.COM. Recuperado 17 de octubre de 2022, de <https://www.dw.com/es/telef%C3%A9rico/t-41334573>
- *Cable de alambre para teleféricos (vías de cuerda aérea/cuerdas para montañas/telesquí/vías para materiales).* (s. f.). Bekaert. Recuperado 17 de octubre de 2022, de <https://www.bekaert.com/es-MX/productos/productos-de-consumo/deporte-y-ocio/cable-de-alambre-para-telefericos-vias-de-cuerda-aerea-cuerdas-para-montanas-telesqui-vias-para-materiales>
- *Elementos de teleféricos - LEITNER.* (s. f.). Recuperado 17 de octubre de 2022, de <https://www.leitner.com/es/empresa/informaciones-utiles/elementos-de-telefericos/>