



UANL®



FIME

Universidad Autónoma de Nuevo León  
Facultad de Ingeniería Mecánica y Eléctrica

*Laboratorio de Biomecánica*

**Práctica 4: Reforzamiento de un cable de teleférico**

Docente: Dra. Yadira Moreno Vera

Día y hora: miércoles N4 Brigada: 309

Integrantes del equipo:

Matriculas	Nombre	Carrera
1806409	Eduardo Antonio Flores Ramírez	IMTC
1884328	Daniel Isaac Zaragoza Soto	IMTC
1900466	Denisse García Espinoza	IMTC
1897305	Kevin Alberto Flores Martínez	IMTC
1991814	Pedro Hazael Uriegas Peña	IMTC

Semestre agosto-diciembre 2022.

A 27 de noviembre de 2022.

#### Práctica #4: Refuerza el cable de un teleférico

Objetivo: El estudiante deberá presentar una propuesta de análisis de formas y de la programación para la ejecución de la optimización para calcular los esfuerzos y deformaciones en Matlab.

Estado del arte:

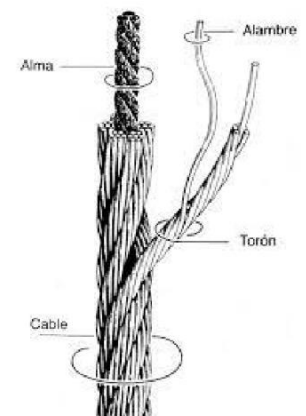
El teleférico es un ingenio que sirve para el transporte de personas y mercancías en zonas de alta montaña o con pronunciados desniveles. Entre sus características principales, destaca que dicho transporte se efectúa sobre vagonetas o cabinas que viajan suspendidas de un fuerte cable de acero.

Los teleféricos han supuesto un adelanto y una mejora para muchos habitantes de dichas zonas. Pero con el tiempo y en según qué parajes ha llegado a convertirse en un problema, debido a la masificación de turistas.

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:

- 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.
- El cable movable, que está enganchado a la cabina y la arrastra.
- El cable de freno, que aguanta la cabina en caso de que el cable movable se rompa.
- El cable de socorro, que, en dicha circunstancia de ruptura, lleva la cabina hasta la estación.

El cable utilizado en el teleférico está compuesto por un núcleo de acero o fibra y varias capas de acero trenzado, entrelazados de forma helicoidal.



Código utilizado:

```

%%Equipo 3, Brigada 309
function top(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<21
passive(ely,elx)=1;
elseif elx>41
passive(ely,elx)=1;
else
passive(ely,elx) = 0;
end
end
end
end
x(fin(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]=1k;
    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*n-1;2*n1; 2*n2-1; 2*n2; 2*n2+1; 2*n2+2; 2*n1+1;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-floor(rmin),1):min(i+floor(rmin),nelx)
            for l = max(j-floor(rmin),1):min(j+floor(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),1); U = zeros(2*(nely+1)*(nelx+1),1);
for elx = 1:nelx
    for ely = 1:nely
        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 LOADS AND SUPPORTS (HALF MBB-BEAM)
F(2,1) = -1;
fixeddofs = union([1:2:2*(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)];

```

## Resultados

Habr  dos casos a simular por lo que se cambiar n las fuerzas.

Caso 1:

```

79      % DEFINE LOADS AND SUPPORTS (HALF MBB-BEAM)
80 -    F(2,1) = -1;
81 -    fixeddofs = union([1:2:2*(nely+1)], [2*(nelx+1)*(nely+1)]);
82 -    alldofs   = [1:2*(nely+1)*(nelx+1)];
83 -    freedofs  = setdiff(alldofs, fixeddofs);

```

It.:	13	Obj.:	131.3801Vol.:	0.330	ch.:	0.133
It.:	14	Obj.:	130.6867Vol.:	0.330	ch.:	0.105
It.:	15	Obj.:	130.0695Vol.:	0.330	ch.:	0.101
It.:	16	Obj.:	129.4490Vol.:	0.330	ch.:	0.099
It.:	17	Obj.:	128.8378Vol.:	0.330	ch.:	0.101
It.:	18	Obj.:	128.6253Vol.:	0.330	ch.:	0.071
It.:	19	Obj.:	128.6258Vol.:	0.330	ch.:	0.060
It.:	20	Obj.:	128.6223Vol.:	0.330	ch.:	0.056
It.:	21	Obj.:	128.6022Vol.:	0.330	ch.:	0.052
It.:	22	Obj.:	128.5867Vol.:	0.330	ch.:	0.047
It.:	23	Obj.:	128.5709Vol.:	0.330	ch.:	0.043
It.:	24	Obj.:	128.5559Vol.:	0.330	ch.:	0.041
It.:	25	Obj.:	128.5439Vol.:	0.330	ch.:	0.040
It.:	26	Obj.:	128.5348Vol.:	0.330	ch.:	0.037
It.:	27	Obj.:	128.5284Vol.:	0.330	ch.:	0.035
It.:	28	Obj.:	128.5220Vol.:	0.330	ch.:	0.033
It.:	29	Obj.:	128.4976Vol.:	0.330	ch.:	0.032
It.:	30	Obj.:	128.5019Vol.:	0.330	ch.:	0.032
It.:	31	Obj.:	128.4973Vol.:	0.330	ch.:	0.031
It.:	32	Obj.:	128.4723Vol.:	0.330	ch.:	0.028
It.:	33	Obj.:	128.4768Vol.:	0.330	ch.:	0.027
It.:	34	Obj.:	128.4542Vol.:	0.330	ch.:	0.025
-----						
It.:	49	Obj.:	127.9311Vol.:	0.330	ch.:	0.024
It.:	50	Obj.:	127.9261Vol.:	0.330	ch.:	0.024
It.:	51	Obj.:	127.9309Vol.:	0.330	ch.:	0.025
It.:	52	Obj.:	127.9258Vol.:	0.330	ch.:	0.025
It.:	53	Obj.:	127.9151Vol.:	0.330	ch.:	0.023
It.:	54	Obj.:	127.9139Vol.:	0.330	ch.:	0.022
It.:	55	Obj.:	127.8997Vol.:	0.330	ch.:	0.020
It.:	56	Obj.:	127.9095Vol.:	0.330	ch.:	0.018
It.:	57	Obj.:	127.8966Vol.:	0.330	ch.:	0.016
It.:	58	Obj.:	127.8946Vol.:	0.330	ch.:	0.015
It.:	59	Obj.:	127.8929Vol.:	0.330	ch.:	0.013
It.:	60	Obj.:	127.8917Vol.:	0.330	ch.:	0.012
It.:	61	Obj.:	127.8907Vol.:	0.330	ch.:	0.011
It.:	62	Obj.:	127.8898Vol.:	0.330	ch.:	0.011
It.:	63	Obj.:	127.8899Vol.:	0.330	ch.:	0.010

Figure 1 × +



Caso 2:

```

96     F(40,1) = -1.; F(9760,2)=1.;
97     fixeddofs = 2*(nely+1):2*(nely+1):2*(nelx+1)*(nely+1);
98     alldofs = [1:2*(nely+1)*(nelx+1)];
99     freeddofs = setdiff(alldofs,fixeddofs);
100    % SOLVING
101    U(freedofs,:) = K(freedofs,freedofs) \ F(freedofs,:);
102    U(fixeddofs,:) = 0;

```

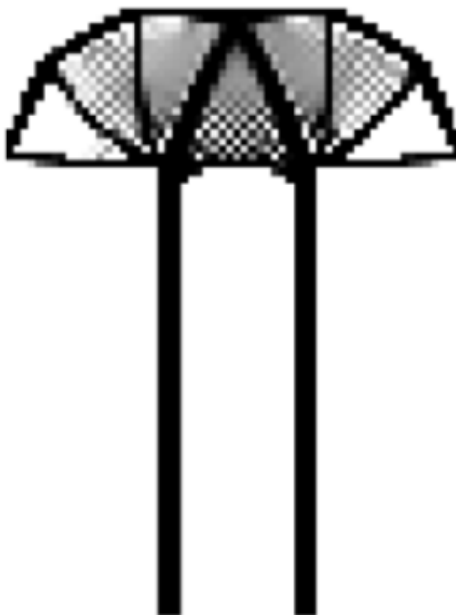
```

It.:   5 Obj.:  597.3313Vol.: 0.200 ch.: 0.200
It.:   6 Obj.:  481.3528Vol.: 0.200 ch.: 0.200
It.:   7 Obj.:  434.3080Vol.: 0.200 ch.: 0.200
It.:   8 Obj.:  399.8733Vol.: 0.200 ch.: 0.200
It.:   9 Obj.:  386.0501Vol.: 0.200 ch.: 0.200
It.:  10 Obj.:  377.1139Vol.: 0.200 ch.: 0.200
It.:  11 Obj.:  370.2076Vol.: 0.200 ch.: 0.200
It.:  12 Obj.:  365.0719Vol.: 0.200 ch.: 0.178
It.:  13 Obj.:  360.9552Vol.: 0.200 ch.: 0.181
It.:  14 Obj.:  357.3927Vol.: 0.200 ch.: 0.165
It.:  15 Obj.:  354.1529Vol.: 0.200 ch.: 0.183
It.:  16 Obj.:  350.9921Vol.: 0.200 ch.: 0.187
It.:  17 Obj.:  347.7993Vol.: 0.200 ch.: 0.179
It.:  18 Obj.:  344.5741Vol.: 0.200 ch.: 0.199

```

It.:	42	Obj.:	322.8136	Vol.:	0.200	ch.:	0.085
It.:	43	Obj.:	322.7960	Vol.:	0.200	ch.:	0.049
It.:	44	Obj.:	322.7907	Vol.:	0.200	ch.:	0.038
It.:	45	Obj.:	322.7876	Vol.:	0.200	ch.:	0.050
It.:	46	Obj.:	322.7856	Vol.:	0.200	ch.:	0.066
It.:	47	Obj.:	322.7798	Vol.:	0.200	ch.:	0.085
It.:	48	Obj.:	322.7741	Vol.:	0.200	ch.:	0.092
It.:	49	Obj.:	322.7614	Vol.:	0.200	ch.:	0.062
It.:	50	Obj.:	322.7526	Vol.:	0.200	ch.:	0.016
It.:	51	Obj.:	322.7506	Vol.:	0.200	ch.:	0.019
It.:	52	Obj.:	322.7482	Vol.:	0.200	ch.:	0.024
It.:	53	Obj.:	322.7510	Vol.:	0.200	ch.:	0.032
It.:	54	Obj.:	322.7489	Vol.:	0.200	ch.:	0.042
It.:	55	Obj.:	322.7435	Vol.:	0.200	ch.:	0.050
It.:	56	Obj.:	322.7368	Vol.:	0.200	ch.:	0.045
It.:	57	Obj.:	322.7331	Vol.:	0.200	ch.:	0.020
It.:	58	Obj.:	322.7286	Vol.:	0.200	ch.:	0.002

Figure 1 × +





Conclusiones:

Pedro Hazael Uriegas Peña 1991814:

Durante esta practica se conocieron los diferentes cableados para un teleférico y de igual manera se buscó optimizar la geometría establecida para reforzar le cable de un teleférico a través de un código anteriormente usado en matlab por lo que fue sencillo realizar la practica ya que se pudo verificar su reforzamiento a través de una simulación anteriormente realizada.

Denisse García Espinoza 1900466:

Se puede notar que el enfoque de esta práctica es muy similar a las anteriores, pues solo cambiamos la geometría, pero el objetivo sigue siendo el mismo, el cual es optimizar un gráfico para que funcione de manera mas eficiente. El programa creado en Matlab nos ayuda a verificar que el teleférico tenga el tamaño correcto durante la instalación, pues nos permite obtener las tensiones causadas por los principales elementos estructurales de dicho sistema.

Kevin Alberto Flores Martínez 1897305

Para este trabajo se nos propuso optimizar la geometría de diseño de un cable de teleférico, basándonos en la geometría que toma un cable cuando se le aplica peso, realizamos dos estudios y estos dieron resultados muy parecidos, para el caso de dos cargas aplicadas se puede apreciar que la pieza es simétrica en el eje Y.

Daniel Isaac zaragoza soto 1884328: Esta práctica nos ayudó a reforzar los temas vistos en la clase de biomecánica, en este caso vimos la tensión de un cable de teleférico y buscar principalmente una geometría establecida con el uso del programa de MATLAB y se buscó analizar y mejorar el grafico obtenido en el programa.

Eduardo Antonio Flores Ramírez 1806409:

En esta práctica llevamos a cabo aplicaciones de temas de esfuerzos, Matlab es un software muy útil para llevar a cabo dichos cálculos a través del uso de sus códigos. Además, se refuerzan los conocimientos de temas de ingeniería que aplican en la vida real, que en este caso fue un teleférico, un invento bastante útil pero que suele llevar consigo ciertos peligros por el uso masificado de turistas.

Referencias:

Ciencia, C. (2022, 3 febrero). *Qué es un teleférico*. CurioSfera Ciencia. <https://curiosfera-ciencia.com/que-es-un-teleferico-como-funciona/>