UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN

FACULTAD DE INGENIERÍA MECÁNICA Y ELÉCTRICA

SEMESTRE AGOSTO-DICIEMBRE 2022

LABORATORIO DE BIOMECÁNICA

PRÁCTICA 2

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 12 de septiembre de 2022.

ÍNDICE

Práctica 2: Diseño del marco de una bicicleta

1. Objetivo

Presentar una propuesta de análisis de formas y de la programación para la ejecución de la optimización (descripción funcional) de características de trabajo específicas que presenta la(s) ventaja(s) (mencionar ventajas).

3. Contenido

Para la elaboración de esta práctica vamos a optimizar lo que es el diseño del marco esto para hacer una mejora, ya que en la actualidad existen algunas bicicletas las cuales están optimizadas pero los diseños suelen ser algo incómodos en algunas ocasiones causándoles a los consumidores algunas molestias, así como también estos diseños suelen ser muy angostos dejando de fuera a un cierto número de compradores, el mal diseño de estos marcos suele ser muy complejos y muy difíciles de montar.

El problema de diseño se ilustra en la figura 2.1. muestra lo que es nuestra tare, básicamente es construir la parte frontal del marco para que la bicicleta sea lo más rígido posible. Esta parte se conecta con el manubrio y el asiento.

Asimismo, debemos de tener en cuenta algunas consideraciones como:

- El manubrio produce una fuerza en dirección vertical.
- Bastidor trasero actúa como soporte.
- Tenemos que declarar una parte vacía del domino de diseño para hacer espacio para la rueda delantera.

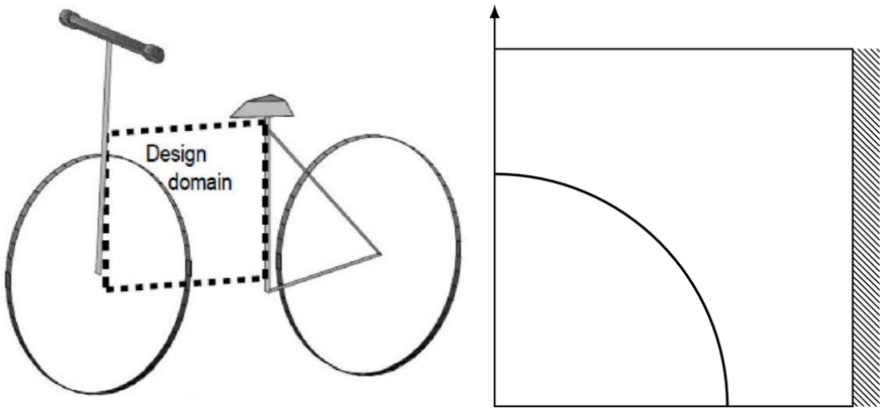

Figura 1. *Problema de diseño*

1) Estado del arte

El cuadro, bastidor, chasis o marco de bicicleta, dependiendo de región o nacionalidad, es la pieza básica de una bicicleta, en la cual se fijan los otros componentes como la horquilla, las ruedas, el sillín, el manillar, etc.

El cuadro de la bicicleta es el elemento central de la bicicleta, y donde van enganchados el resto de los componentes. La mayoría de los cuadros de bicicleta son de aluminio, de fierro o de fibra de carbono los más modernos. Los tubos frontales, se conocen como tubo superior y tubo inferior, los cuales están unidos en la parte del frente por el telescopio; mientras que aquellos de la parte posterior, se denominan vainas, superior e inferior. También incluye el tubo de asiento, que es el elemento vertical donde se conectará el sillín.

Figura 2. *Bicicleta*

Si observas la imagen, podrás distinguir dos "triángulos". El frontal, que no es precisamente un triángulo; se conforma por el tubo del asiento, el tubo superior, el tubo inferior y el tubo de la dirección. Por su parte, el triángulo trasero; está formado por las vainas superiores e inferiores y el tubo del asiento.

El diseño de los cuadros de las bicicletas está destinado a dar una mejor respuesta a los diferentes requerimientos que los usuarios puedan darle. En este aspecto, el material, pero sobre todo la geometría que esté presente dará respuesta a las diferentes necesidades del mercado. De ahí la variedad de modelos, en busca de tener un manejo óptimo para cada tipo de cliente. La estética y las modas harán el resto.

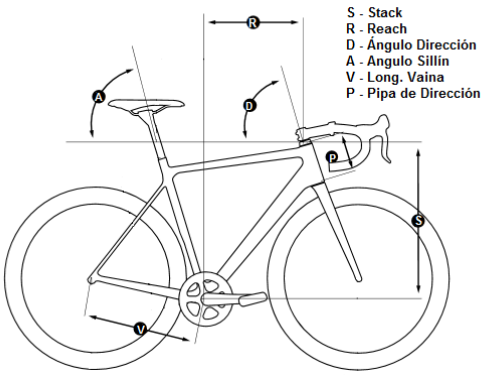

Figura 3. *Diseño de bicicleta*

Así mismo, para un buen diseño del cuadro de la bicicleta es necesario aplicar la optimización topológica vista la practica pasada. Esta optimización topológica consiste en lo siguiente.

La optimización topológica es una técnica englobada dentro del campo de análisis estructural. Se basa en el análisis mecánico de un componente o estructura. Su principal objetivo es el aligeramiento estructural manteniendo las funcionalidades mecánicas del componente objetivo.

A diferencia de otros tipos de optimización, la optimización topológica ofrece un nuevo concepto de diseño estructural enfocado a aquellas aplicaciones donde el peso del componente es crucial.

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

Como propuesta para el diseño de la geometría, se puede seguir exactamente el diseño cuadrado que se plantea, pero con la diferencia de contar con un pequeño soporte en diagonal para así poder realizar correctamente o para apreciar alguna mejora dentro de dicho diseño y así ver posteriormente, al momento de realizar el análisis numérico, obtener resultados mejores o altos respecto a lo que se tenía antes, lo cual es el objetivo principal dentro de este problema, tener resultados cuantificables buenos y que por ende esto haga que el diseño tenga un buen desempeño mecánico.

La siguiente figura muestra una propuesta aproximada la cual serviría, como se mencionó antes, obtener un mejor desempeño mecánico:

Figura 4. *Resultado esperado de optimización topológica.*

Algunas limitaciones que puede tener el diseño son al momento de la modificación de los parámetros de la optimización topológica, pues estos pueden hacer que la estructura sea más o menos rígida según lo establecido en los parámetros.

4) Desarrollo de la programación.

a. 1era optimización

Primero abrimos el MATLAB y creamos un script, el cual guardamos como archivo .m para poder hacer nuestro programa, una vez modificado el programa de 99 líneas realizado en la práctica anterior, lo corremos y nos muestra el marco de la bicicleta que estamos buscando, después pasamos a cambiar los parámetros para poder optimizar nuestro marco de bicicleta.

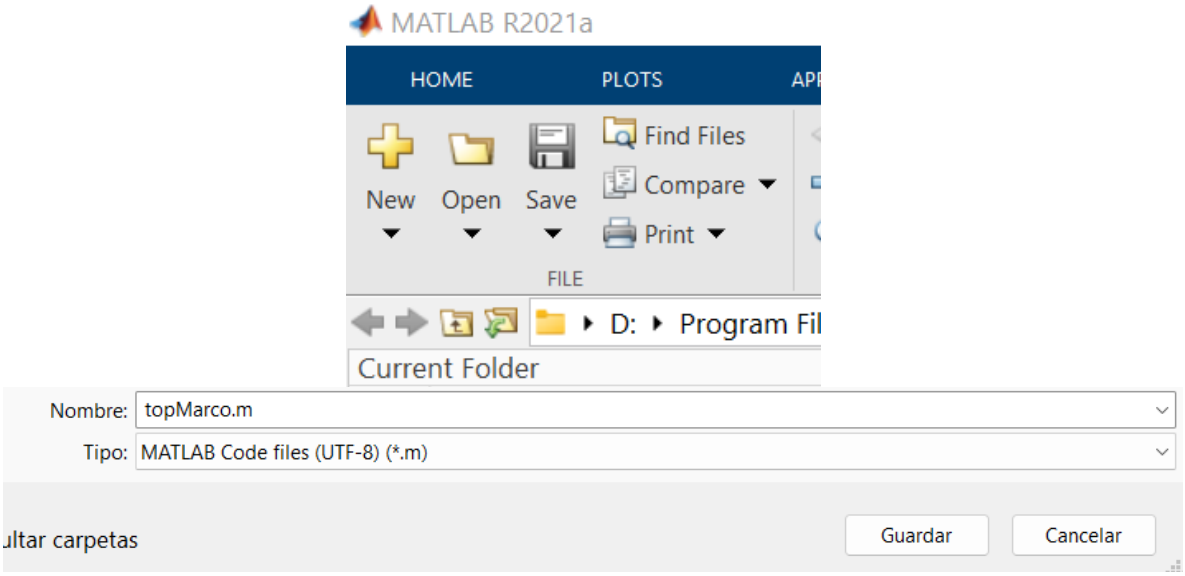

Figura 5. *Creación de archivo en Matlab.*

Nuestra primera optimización es el topMarco(20,20,0.33,3.0,1.5) de la Figura 6 y 7, esta optimización nos brinda una imagen borrosa del marco de la bicicleta, por lo que no es lo que estamos buscando realmente, luego cambiamos los parámetros a topMarco(12,12,0.33,3.0,0.9) y esta nos brinda una imagen mucho más clara de la que tuvo la primera optimización solamente tiene un pixel el cual se ve borroso a diferencia de los demás.

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

topMarco.m

```
1
2     %%%% A 99 LINE TOPOLOGY OPTIMIZATION CODE BY OLESIGMUND, OCTOBER 1999 %%%
3     function topMarco(nelx,nely,volfrac,penal,rmin)
4     % INITIALIZE
5 -   x(1:nely,1:nelx) = volfrac;
6 -   loop = 0;
7 -   change = 1.;
8     % START ITERATION
9 -   while change > 0.01
10 -  loop = loop + 1;
11 -  xold = x;
12    % FE-ANALYSIS
13 -  [U]=FE(nelx,nely,x,penal);
14    % OBJECTIVE FUNCTION AND SENSITIVITY ANALYSIS
15 -  [KE] = lk;
16 -  c = 0.;
17 -  for ely = 1:nely
18 -  for elx = 1:nelx
19 -  n1 = (nely+1)*(elx-1)+ely;
```

Command Window

Figura 6. *Código modificado para primera optimización.*

```
Command Window
>> topMarco(20,20,0.33,3.0,1.5)
It.:    1Obj.:  372.2150 Vol.:  0.330 ch.:  0.200
It.:    2Obj.:  186.0788 Vol.:  0.330 ch.:  0.200
It.:    3Obj.:  122.6779 Vol.:  0.330 ch.:  0.200
It.:    4Obj.:   87.5816 Vol.:  0.330 ch.:  0.200
It.:    5Obj.:   66.0079 Vol.:  0.330 ch.:  0.200
It.:    6Obj.:   52.4132 Vol.:  0.330 ch.:  0.197
It.:    7Obj.:   43.8460 Vol.:  0.330 ch.:  0.156
It.:    8Obj.:   39.3067 Vol.:  0.330 ch.:  0.119
```

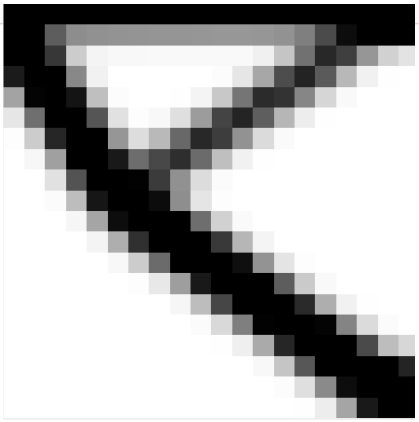

Figura 7. *Resultado de primera optimización.*

b. 2da optimización

Se muestra ahora, un diseño que conserva el espacio para la rueda delantera de la bicicleta, mostrando la eficiencia del análisis como se realizó anteriormente. Esto, haciendo uso de los elementos finitos y creando una matriz con ceros en elementos libres. El resultado obtenido se parece más al marco de bicicleta de un hombre, ya que no está hecho para ser tan sencillo al momento montar (Fig. 8 y 9.). Se realizo con la misma optimización topológica de topMarco(20,20,0.33,3.0,1.5)), solamente realizando algunas modificaciones al código (Fig. 8 y 9) y le aplicamos un filtro de mallado, el resultado que obtenemos es un marco de bicicleta bien definido, al principio teníamos un pixel que estaba un poco borroso a comparación de los demás pero con ayuda del filtro de mayado se ve mucho más claro que en la primera imagen. (Fig. 10).

```
47      %%%%%%%%%%37 OPTIMALITY CRITERIA UPDATE %%%%%%%%%
48      function [xnew]=OC(nelx,nely,x,volfrac,dc,passive)
49 -    l1 = 0; l2 = 100000; move = 0.2;
50 -    while ((l2-l1)/l2 > 1e-4)
51 -    lmid = 0.5*(l2+l1);
52 -    xnew = max(0.001,max(x-move,min(1.,min(x+move,x.*sqrt(-dc./lmid)))));
53 -    xnew(find(passive)) = 0.001;
54 -    if sum(sum(xnew)) - volfrac*nelx*nely > 0
55 -    l1 = lmid;
56 -    else
57 -    l2 = lmid;
58 -    end
 3      %3 INITIALIZE
 4 -    x(1:nely,1:nelx) = volfrac;
 5 -    for ely = 1:nely
 6 -    for elx = 1:nelx
 7 -    if ((elx)^2+(ely-nely)^2) < (0.65*nelx)^2
 8 -    passive(ely,elx) = 1;
 9 -    else
10 -    passive(ely,elx) = 0;
11 -    end
12 -    end
13 -    end
14 -    x(find(passive))=0.001;
15 -    loop = 0;
16 -    change = 1.;
```

Figura 6. *Modificación de código.*

```
34 -    end
35      %25 FILTERING OF SENSITIVITIES
36 -    [dc] = check(nelx,nely,rmin,x,dc);
37      %27 DESIGN UPDATE BY THE OPTIMALITY CRITERIA METHOD
38 -    [x] = OC(nelx,nely,x,volfrac,dc,passive);
39      %29 PRINT RESULTS
40 -    change = max(max(abs(x-xold)));
41 -    disp(['It.:' sprintf('%4i',loop) 'Obj.:' sprintf('%10.4f',c) ...
42      ' Vol.: ' sprintf('%6.3f',sum(sum(x))/(nelx*nely)) ...
43      ' ch.: ' sprintf('%6.3f',change )])
44      %34 PLOT DENSITIES
45 -    colormap(gray); imagesc(-x); axis equal; axis tight; axis off;pause(1e-6);
46 -    end
47      %%%%%%%%%%37 OPTIMALITY CRITERIA UPDATE %%%%%%%%%
48      function [xnew]=OC(nelx,nely,x,volfrac,dc,passive)
```

```
44      %34 PLOT DENSITIES
45 -    colormap(gray); imagesc(-x); axis equal; axis tight; axis off;pause(1e-6);
46 -    end
47      %%%%%%%%%%37 OPTIMALITY CRITERIA UPDATE %%%%%%%%%
48      function [xnew]=OC(nelx,nely,x,volfrac,dc,passive)
49 -    l1 = 0; l2 = 100000; move = 0.2;
50 -    while ((l2-l1)/l2 > 1e-4)
51 -    lmid = 0.5*(l2+l1);
52 -    xnew = max(0.001,max(x-move,min(1.,min(x+move,x.*sqrt(-dc./lmid)))));
53 -    xnew(find(passive)) = 0.001;
54 -    if sum(sum(xnew)) - volfrac*nelx*nely > 0
55 -    l1 = lmid;
56 -    else
```

```
49 -    l1 = 0; l2 = 100000; move = 0.2;
50 -    while ((l2-l1)/l2 > 1e-4)
51 -    lmid = 0.5*(l2+l1);
52 -    xnew = max(0.001,max(x-move,min(1.,min(x+move,x.*sqrt(-dc./lmid)))));
53 -    xnew(find(passive)) = 0.001;
54 -    if sum(sum(xnew)) - volfrac*nelx*nely > 0
```

Figura 7. *Modificación de código.*

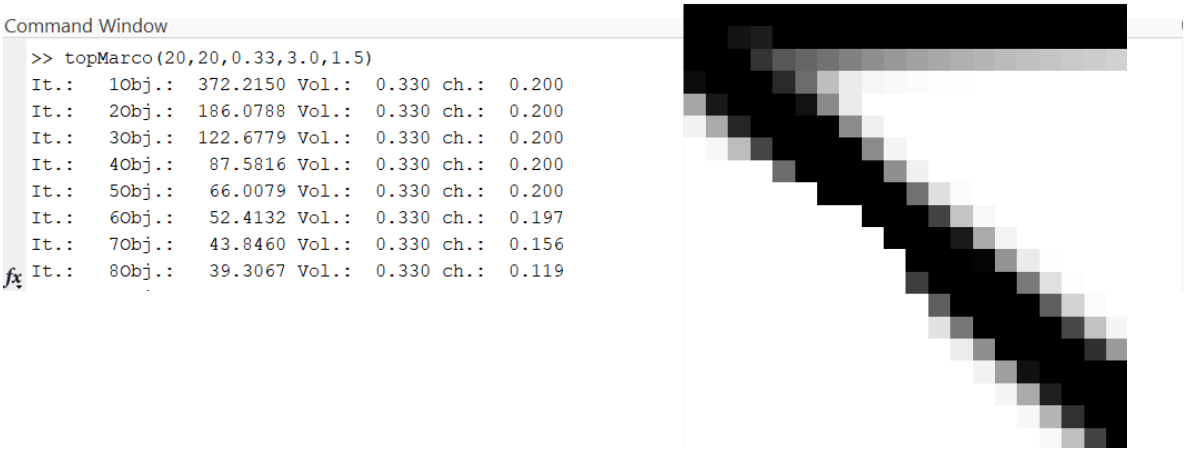

Figura 8. *Resultado de modificación de código y optimización con topMarco(20,20,0.33,3.0,1.5).*

a. Resultados de otras pruebas.

Ya que se ha conseguido el propósito deseado, será de gran utilidad en el aprendizaje "jugar" con los parámetros. Así, se podrán observar los cambios y afectaciones que tiene en el diseño final. De acuerdo con el tamaño del mallado, la exigencia del color en blanco y negro, así como el filtro del mallado mismo.

1. Observar si el diseño final depende del tamaño de mallado comparando con la 2da optimización

En la primera prueba, reducimos el valor del radio de filtro *rmin* que hace el diseño del mallado. Y como se puede observar en el diseño final, se muestra una figura más "clara" en cuanto a la delimitación de esta. Pero es como una pieza simplificada, no tan detallada, como en casos anteriores donde el valor de la malla es mayor.

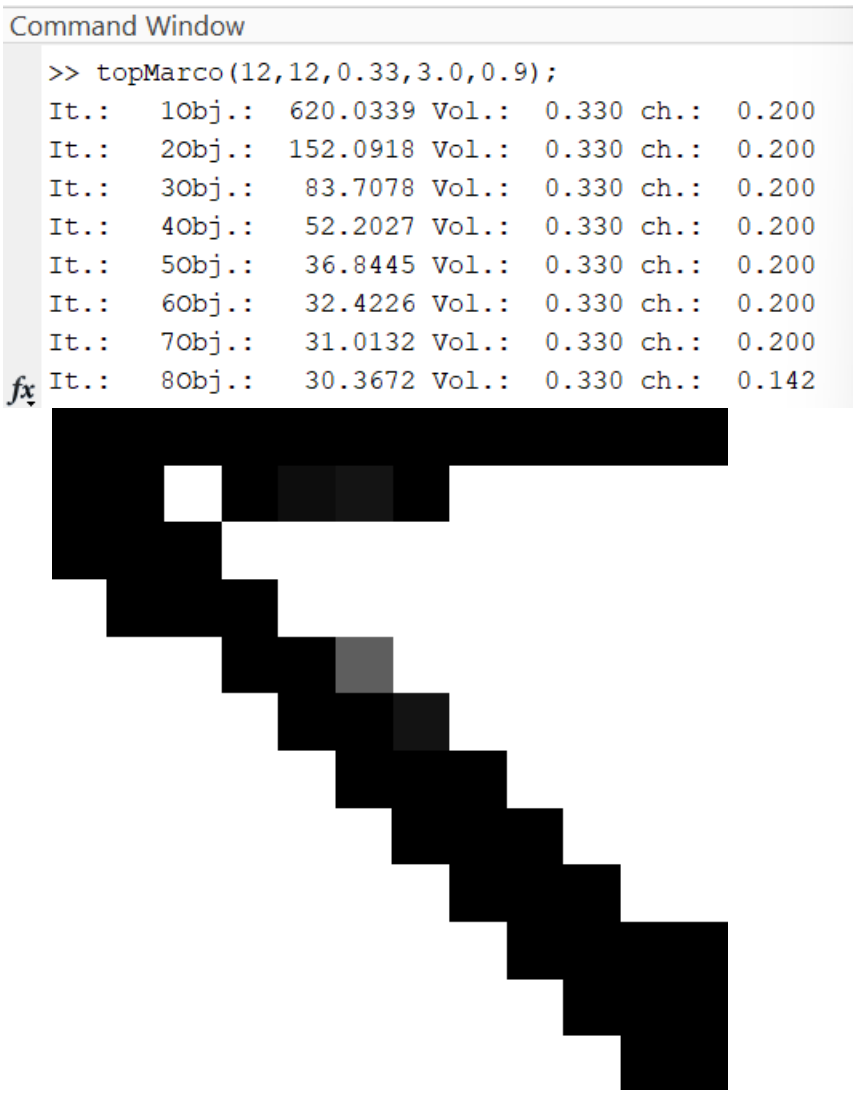

Figura 9. *Resultado de optimización topológica con topMarco(12,12,0.33,3.0,0.9).*

Como segunda prueba, se coloca ahora, un valor de 1.2 en *rmin*, que se encuentra dentro del rango predeterminado 1.5 y el valor anterior de 0.9. Y como es de esperarse, el resultado de la pieza final es una combinación de la 2da optimización obtenida y el ejercicio anterior. Las partes del marco donde se encuentran la carga del manubrio y el "soporte" del bastidor, generan un poco menos de claridad en la pieza.

```
Command Window
>> topMarco(16,16,0.33,3.0,1.2);
It.:   1Obj.:  664.6669 Vol.:  0.330 ch.:  0.200
It.:   2Obj.:  162.1308 Vol.:  0.330 ch.:  0.200
It.:   3Obj.:   91.0349 Vol.:  0.330 ch.:  0.200
It.:   4Obj.:   60.0106 Vol.:  0.330 ch.:  0.200
It.:   5Obj.:   42.8836 Vol.:  0.330 ch.:  0.200
It.:   6Obj.:   37.0143 Vol.:  0.330 ch.:  0.184
It.:   7Obj.:   34.9988 Vol.:  0.330 ch.:  0.154
It.:   8Obj.:   34.1702 Vol.:  0.330 ch.:  0.096
```

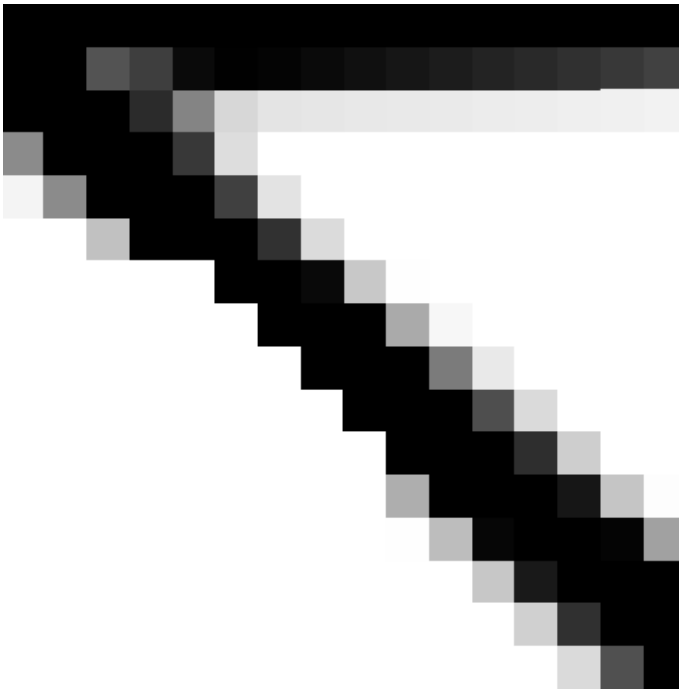

Figura 10. *Resultado de optimización topológica con topMarco(16,16,0.33,3.0,1.2).*

2. Comparación diseño B/Negro

Como segundo cambio en los parámetros del diseño, se realiza una modificación en la exigencia de color. Para esto, se cambia el valor de *penal*, Este valor se refiere a la penalización de densidades intermedias. Una penalización alta como 3.0, que se había asignado anteriormente, se coloca ahora un valor de 1, que significa que no hay penalización de densidades intermedias.

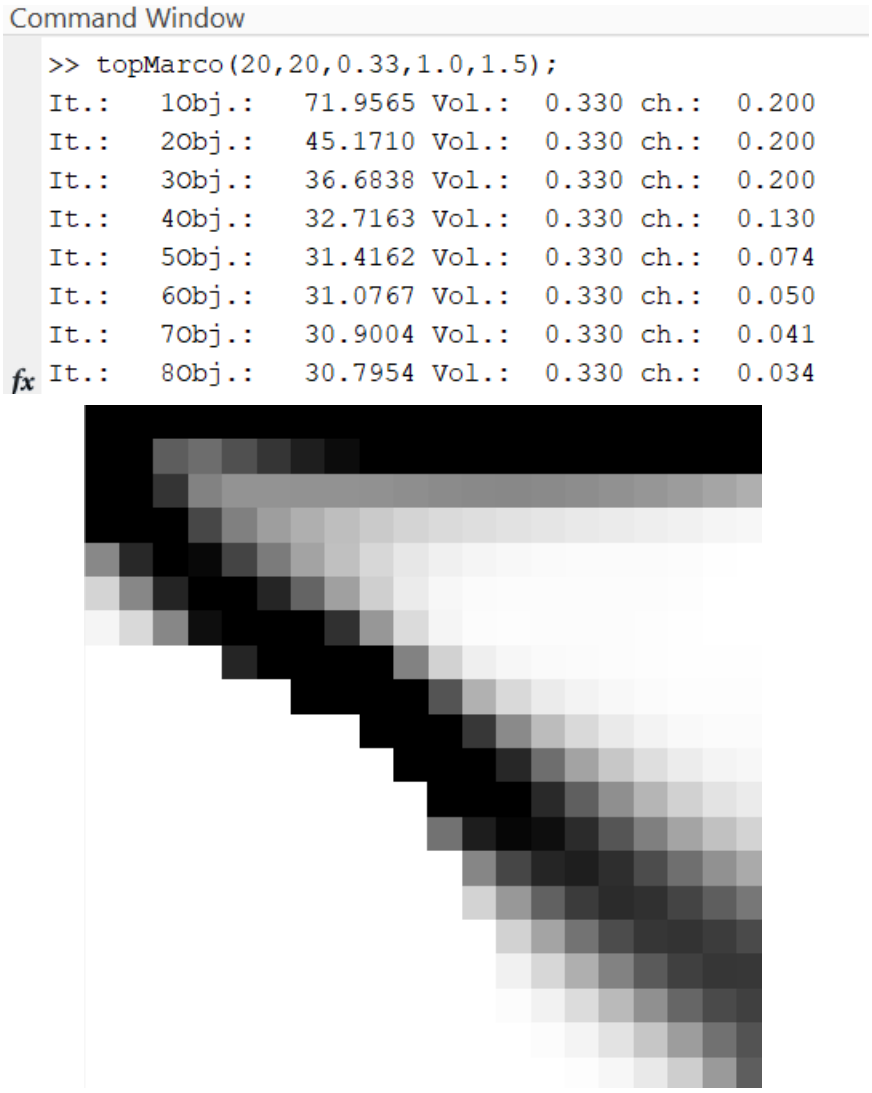

Figura 11. *Resultado de optimización topológica con topMarco(20,20,,0.33,1.0,1.5).*

3. Estudio del filtro de mallado

Para el último análisis, hacemos una modificación en el filtro del mallado. El filtro se desactiva colocando un valor rmin, menor a 1 o también asignando como comentario la instrucción:

$$[\mathbf{dc}] = \mathbf{check}[\mathbf{nelx,nely,x,volfrac,dc,passive}]$$

Así, el programa ignora la función y desactiva el filtro, de modo, que nos entrega una pieza de resultado final diseñada sin el filtro de mallado. Como se muestra a continuación:

```
33 -    end
34 -    end
35      %25 FILTERING OF SENSITIVITIES
36      %[dc] = check(nelx,nely,rmin,x,dc);
37      %27 DESIGN UPDATE BY THE OPTIMALITY CRITERIA METHOD
38 -    [x] = OC(nelx,nely,x,volfrac,dc,passive);
39      %29 PRINT RESULTS
40 -    change = max(max(abs(x-xold)));
```

Figura 12. *Código modificado para la desactivación del filtro.*

Y se describen los resultados sin filtro, así como como se realizó anteriormente con filtro mediante:

```
Command Window
>> topMarco(20,20,0.33,3.0,1.5);
It.:    1Obj.:  666.5350 Vol.:  0.330 ch.:  0.200
It.:    2Obj.:  162.7684 Vol.:  0.330 ch.:  0.200
It.:    3Obj.:   90.4112 Vol.:  0.330 ch.:  0.200
It.:    4Obj.:   57.7144 Vol.:  0.330 ch.:  0.200
It.:    5Obj.:   41.3544 Vol.:  0.330 ch.:  0.200
It.:    6Obj.:   35.4010 Vol.:  0.330 ch.:  0.200
It.:    7Obj.:   33.3361 Vol.:  0.330 ch.:  0.200
It.:    8Obj.:   32.4520 Vol.:  0.330 ch.:  0.195
```

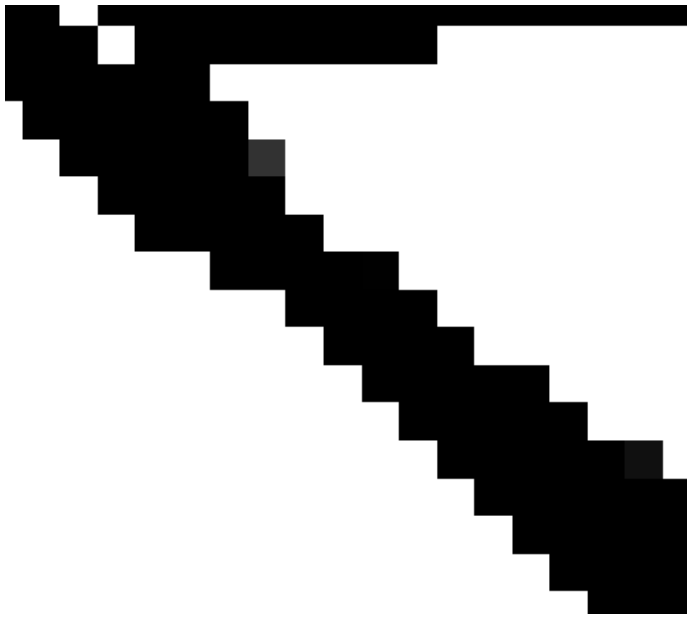

Figura 13*. Resultado de optimización topológica con topMarco(20,20,0.33,3.0,1.5).*

5) Resultados de la optimización

En esta optimización logramos observar que con el código utilizado en la práctica anterior y solamente haber editado la línea 80 y 81, tenemos la misma pieza, pero del lado inverso, así como se muestra vemos que la magnitud de fuerza no es real por lo tanto el módulo de Young debe ser corregido, de tal forma para que también se encuentre el valor correcto para la optimización de la cual se busca encontrar para la bicicleta (Fig. 16.).

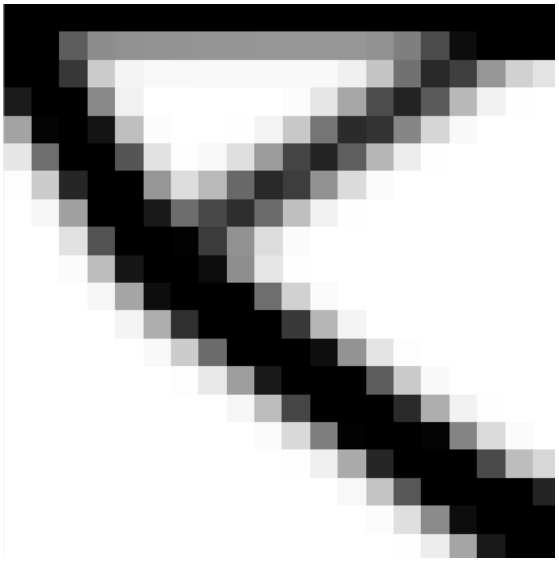

Figura 14. *Resultado de 1era optimización.*

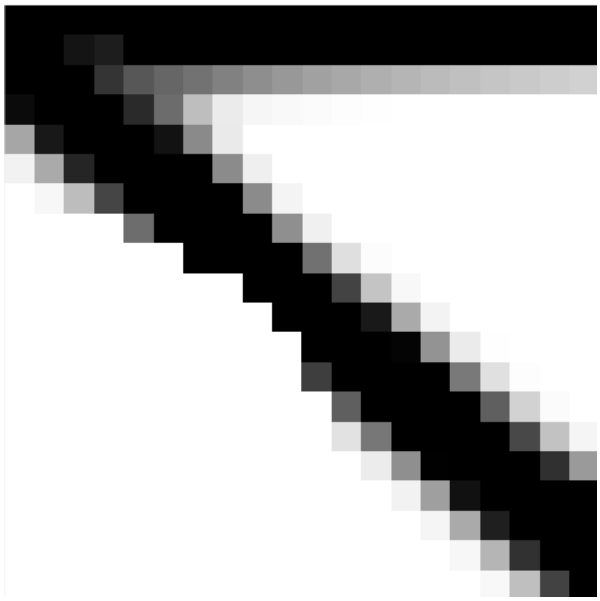

Figura 15. *Resultado de 2da optimización.*

Después de realizar los cambios sugeridos, explicamos lo que es el ámbito de las regiones vacías ya que en nuestra optimización anterior no dejamos ninguna zona hueca para aplicar la rueda, por ende, se utilizan elementos finitos para este vacío, donde que define una matriz con ceros en elementos libres y seres en pasiva. Por ende, se editan las líneas 43 44 con el comando que se indica y ese sí cómo se genera el hueco para nuestra rueda (Fig. 18).

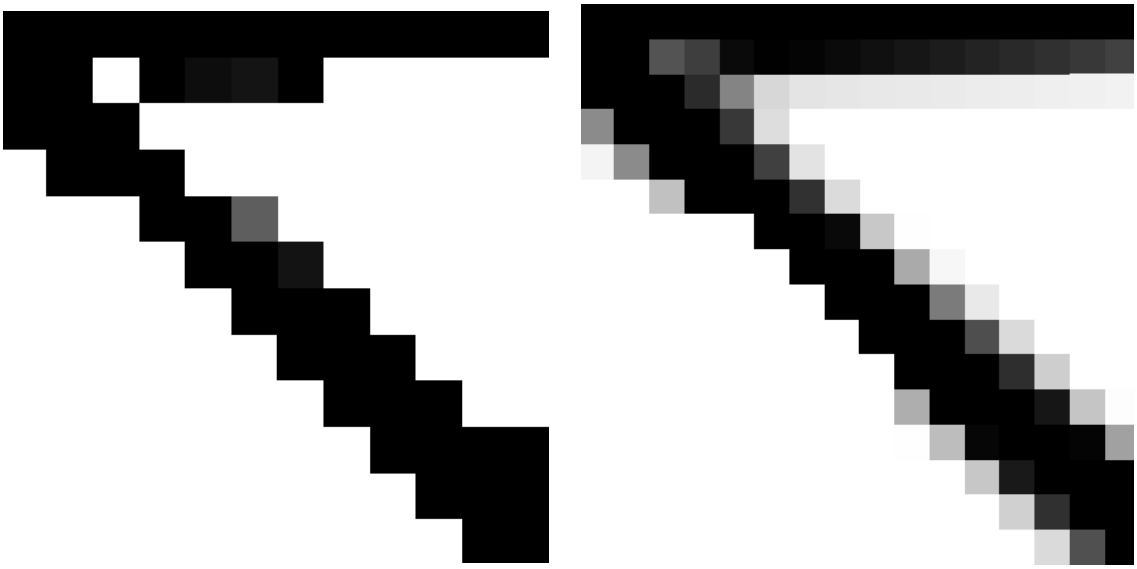

Figura 16. *Comparación de resultados entreptimizaciones con topMarco(12,12,0.33,3.0,0.9) y topMarco(16,16,0.33,3.0,1.2).*

En esta parte de los resultados vemos acerca de la comprobación si realmente el diseño final depende del tamaño del mallado, haciendo comprobación de los resultados con los comandos que realizamos en la programación y editando las diferentes especificaciones, puede comprobar que en la primera prueba se reduce el radio del Filtro y mostramos una figura más clara pero como es una pieza simplificada no es tan detallada en el ámbito del mallado por ende este caso no tiene un valor tan alto de mallado (Fig. 18).

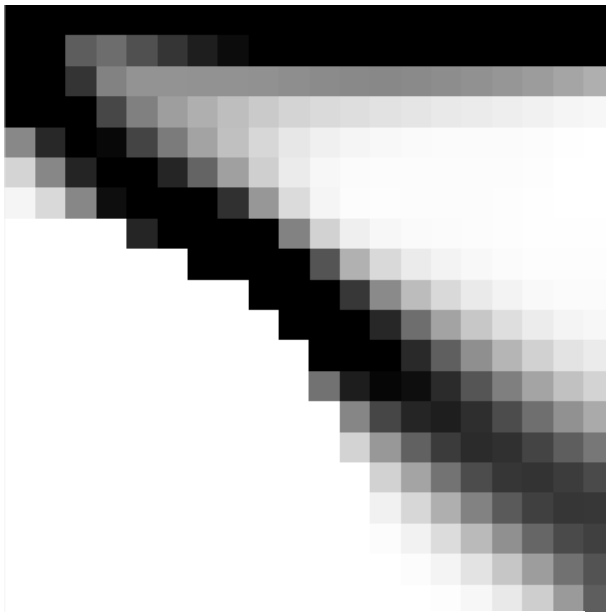

Figura 17. *Resultado de optimización topMarco(20,20,,0.33,1.0,1.5).*

Pero por la segunda prueba se coloca otro valor que está dentro del rango predeterminado y como se observa en la imagen de la derecha, se obtiene que la pieza final es la combinación de la segunda optimización obtenida anteriormente antes de realizar estos comandos, donde vemos que el manubrio y el soporte del bastidor generan menos claridad a través de la pieza (Fig. 18).

Para la comparación de diseño blanco/negro, se van a realizar 100% una modificación a través de la exigencia

de color. El cual, el valor que se modificó fue el valor penal, donde este valor a través del diseño podemos referirnos a la penalización de densidades intermedias, pero cómo proponemos que es el mejor resultado del diseño a través de blanco y negro, esto se debe a las incertidumbres que se tienen a través del diseño ya que se verá a la imagen del diseño al aplicarle un filtro se podrá suavizar los bordes del resultado final del diseño para obtener la imagen más limpia. Para ello se coloca un valor de uno, haciendo referencia a que no habrá penalización de entidades intermedias (Fig. 19).

Por último, el estudio del filtro del mallado, para la superficie hacemos el análisis modificando el filtro que desactiva un valor mínimo, el cual este valor mínimo de uno, modificándolo dentro de un comando, este hará que se desactive el filtro lo cual nos dará la pieza que se ve como resultado final la cual fue diseñada sin el filtro de mallado, lo cual la mejor opción para nosotros fue mostrar la opción sin filtro, ya que se contempló el realizar una mejor optimización de diseño en esta pieza (Figura 20).

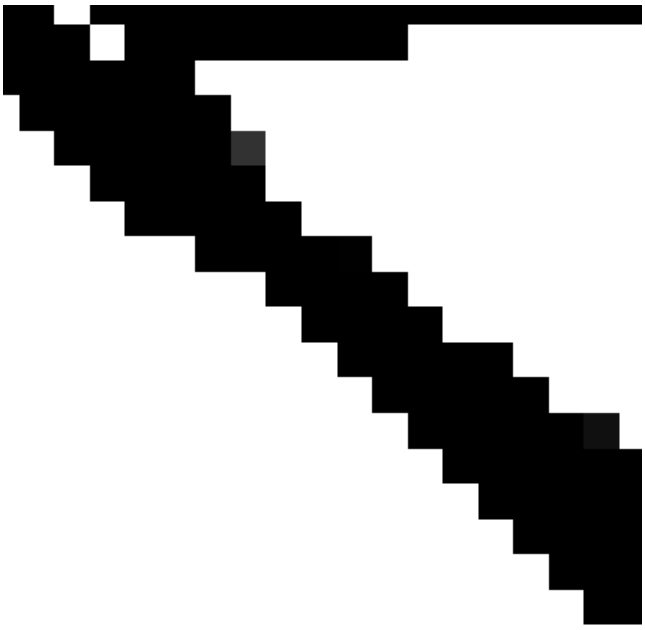

***Figura 18**. Resultado de optimización topológica con topMarco(20,20,0.33,3.0,1.5).*

6) Conclusiones

Diana Elisa Acosta Rodríguez:

En esta práctica, se ha realizado un análisis más detallado y en donde se puede comprender mejor el funcionamiento del código de 99 líneas para la optimización topológica en Matlab. En este caso, fue un diseño diferente en el que se produjo el marco de una bicicleta, tomando en cuenta las consideraciones especificadas, tanto como las fuerzas que se le aplican. Para su análisis de diseño, el manubrio se consideró como una carga vertical, mientras que el bastidor trasero como soporte. Además, de observar la importancia que los elementos finitos tienen en la optimización

topológica. Y, por último, para entender mejor el código, se realizaron cambios en los parámetros, tales como en el tamaño de la malla, que se hace modificando el valor de *rmin*, la exigencia del color blanco y negro, modificando en la sintaxis, la variable penal, al igual que probando el resultado con y sin filtro de mallado. Todo esto, nos ha permitido observar la influencia que estos cambios, que, por ser tan solo diferencias de un número, genera notables alteraciones en el diseño de la pieza final.

Joaquín Abdiel Coronado Vázquez:

Para la realización de la práctica, vimos un poco más a detalle sobre lo que es el miedo a través de la realización de Matlab, el cual implica un poco más cerca sobre la optimización topológica, que se suma importancia aprender que esta parte debido a que el diseño se basa en esto y también la materia va por ese rumbo. el caso de la bicicleta vemos que utilizamos el mismo código de la práctica pasada, pero con unos ciertos detalles que se realizaron Y se fueron cambiando, esto debido a la necesidad de saber distintos factores de optimización lo cual ayuda también acerca sobre los diferentes factores que pueden tener una persona a través de la bicicleta como pueden ser peso, altura, etcétera. ya que nosotros no sólo nos basamos en implementar el código, pero también calculamos las fuerzas y vemos distintos factores que son importantes para el diseño e implementación de cada optimización de esta práctica realizada.

Brenda Giselle Hinojosa:

Poco a poco con la realización de las prácticas vamos comprendiendo lo que es la optimización topológica; como esta es de suma importancia en cada diseño que se haga. En el caso del cuadro de la bicicleta al momento de investigar, nos informamos y pudimos comprender que para el diseño de este es muy importante diferentes factores como el peso, la altura de la persona, el uso que se le vaya a dar a la bicicleta, etc.; por ello al momento de hacer el análisis de optimización topológica se deben tomar en cuenta factores como este para así en la simulación poder obtener el marco más optimo y posteriormente poder pasarlo en físico para igualmente someterlo a pruebas físicas y así ir obteniendo cada vez un mejor marco para la bicicleta.

Gloria Rosalía Domínguez Azueta:

Al optimizar el diseño geométrico del marco de una pieza comprendí las diferentes modificaciones del código de 99 líneas utilizada en la práctica anterior, para acoplarlo

a un nuevo diseño, observé como al cambiar los diferentes parámetros de la optimización se reducía o aumentaba el peso hasta llegar a un punto en donde se obtuvo la forma mucho más conservadora y no tan agresiva, es decir, con mayor rigidez, que es lo que se buscaba, por su parte, los diseños deben cumplir ciertos requisitos para su optimización en este caso se tomó en cuenta la fuerza producida por el manubrio, así como el soporte y la consideración de la llanta delantera, de esta manera el diseño se adecua a lo que se necesita. En general, con las prácticas ya realizadas, he podido comprender mejor la lógica del código en Matlab y la relación con la optimización topológica, al momento de hacer la colocación de cargas, apoyos y fuerzas dentro del espacio del diseño, aún faltan conceptos por comprender, sin embargo, considero que con la práctica quedarán cada vez más claros.

Andrea Anette Celestino Castillo:

Para concluir con la elaboración de esta práctica puedo mencionar que ahora con la elaboración de esta, logramos darle solución al problema planteado. Con la realización del código pudimos notar lo que es la optimización topológica y como es que funciona debido a que en las dos practicas hasta ahora realizadas las hemos abordado, también gracias a la elaboración de esta práctica pudimos notar lo que es la optimización topológica abordando diferentes parámetros, ya que en la elaboración de esta práctica se tuvieron que realizar distintos cambios al código para ver su optimización. Asimismo, pudimos utilizar lo que es la lógica del código en el software de Matlab para la colocación tanto de cargas como apoyos dentro de un espacio de diseño ya propuesto. Creo que con la utilización del software durante las siguientes prácticas todos sus usos quedaran aún más claro para la modelación de distintos dispositivos.

Ana Paulina Pérez Salazar:

Conforme estamos elaborando estas prácticas pudimos notar que analizamos y definimos de la mejor manera posible el problema implicado en esta práctica, de igual forma resolvimos de una manera óptima este. Es importante mencionar que esta práctica se observó cómo se puede ejecutar y realizar simulaciones mecánicas y de diseño mecanico en Matlab. También se implementaron las ideas o conocimientos obtenidos dentro de la materia de biomecanica desde los primeros pasos hasta el final en donde se argumenta el cómo la pieza que se genera bajo la metodologia de

prototipado y construccion virtual por medio de software ahora tiene un mejor desempeño mecanico debido al mallado que se realizo. Volver a aplicar el ambiente o desarrollo de un diseño mecanico fue muy importante, ya que nos indica la metodologia o proceso que se debe seguir para una elaboracion correcta de una pieza.

Javier Cisneros Saldivar:

Como pudimos observar en esta práctica, tuvimos que optimizar lo mejor posible el marco de una bicicleta, tuvimos que cambiar los parámetros del programa varias veces para poder obtener una imagen definida y realmente optimizada del marco de la bicicleta, en total se hicieron 4 pruebas para poder llegar a nuestro resultado final, cabe destacar que en una prueba el marco de la bicicleta casi estuvo bien definido pero un pixel se veía muy borroso a comparación de los demás lo que provocó que tuviéramos que seguir cambiando los parámetros, hacer construcciones de objetos en MATLAB es algo tedioso pero nos ayuda a saber que tan optimo esta nuestro objeto.

Bibliografía

- Carlos Alberto Meza Valencia. (3 octubre 2012). Optimización topológica en el diseño de elementos estructurales mecánicos. 11 septiembre 2022, De Universidad Autónoma de occidente Base de datos.
- Carlos Julio Camacho Flores. (febrero 2011). Optimización topológica estructural de ensambles. 11 septiembre 2022, de Universidad Nacional de Colombia Base de datos.
- Alex Esteban Fernández. (11 enero 2021). Estudio y diseño de cuadros de bicicleta. 11 septiembre 2022, de Universitat politécnica de catalunya

Anexo

Código de 99 líneas en Matlab para marco de una bicicleta:

```
%%%%1 A 99 LINE TOPOLOGY OPTIMIZATION CODE BY
OLESIGMUND, OCTOBER 1999, MODIFICACIÓN PARA MARCO DE
BICICLETA %%%
function topMarco(nelx,nely,volfrac,penal,rmin)
%3 INITIALIZE
x(1:nely,1:nelx) = volfrac;
for ely = 1:nely
for elx = 1:nelx
```

```
if ((elx)^2+(ely-nely)^2) < (0.65*nelx)^2
passive(ely,elx) = 1;
else
passive(ely,elx) = 0;
end
end
end
x(find(passive))=0.001;
loop = 0;
change = 1.;
%7 START ITERATION
while change > 0.01
loop = loop + 1;
xold = x;
%11 FE-ANALYSIS
[U]=FE(nelx,nely,x,penal);
%13 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;
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);
c = c + x(ely,elx)^penal*Ue'*KE*Ue;
dc(ely,elx) = -penal*x(ely,elx)^(penal-1)*Ue'*KE*Ue;
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 )])
%34 PLOT DENSITIES
colormap(gray); imagesc(-x); axis equal; axis tight;
axis off;pause(1e-6);
end
%%%%%%%%%%37 OPTIMALITY CRITERIA UPDATE %%%%%%%%%
```

```
function [xnew]=OC(nelx,nely,x,volfrac,dc,passive)
l1 = 0; l2 = 100000; move = 0.2;
while ((l2-l1)/l2 > 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
%%%%%%%%%%49 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),1); U
=sparse(2*(nely+1)*(nelx+1),1);
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
```

```
% 78 DEFINE LOADSAND SUPPORTS(HALF MBB-BEAM)
F(2,1) = 1;
fixeddofs =2*nelx*(nely+1)+1:2*(nelx+1)*(nely+1);
alldofs = [1:2*(nely+1)*(nelx+1)];
freedofs = setdiff(alldofs,fixeddofs);
% 83 SOLVING 127
U(freedofs,:) = K(freedofs,freedofs) \F(freedofs,:);
U(fixeddofs,:)= 0;
%%%%%%%%%%86 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)];
```

.