Práctica 2

1845056 Aurora Nahomy Martínez Pérez
1854324 Kevin Orlando Huerta Jaramillo
1896681 Jovanny Daniel Alvarado Ramírez
1909876 Fernando Herrera Garza
1991876 Bernardo Gil Villarreal

16 de septiembre de 2022

1. Objetivo

El estudiante deberá presentar una propuesta de análisis de formas y de la programación para la ejecución de la optimización (descripción funcional) de carcaterísticas de trabajo específicas que presneta las(s) ventaja(s) (mencionar ventajas).

2. Introducción

En está práctica se realizará el diseño y optimización de un marco de una bicicleta, utilizando el código que utilizamos en la práctica pasada, haciendo algunas modificaciones a este mismo.

3. Marco Teórico

En los últimos 200 años el diseño de la bicicleta ha evolucionado de diversas maneras a fin de satisfacer las necesidades de una amplia gama de usuarios y mejorar el desempeño de la bicicleta en diferentes disciplinas.

Actualmente hay docenas de tipos de bicicletas especializadas para distintas modalidades de ciclismo y sus diferencias están relacionadas con su desempeño y cualidades. Muchas de estas diferencias están directamente relacionadas con la geometría de su cuadro.

Algunos de los componentes de las bicicletas varían en función del tipo, pero en general todas comparten las siguientes partes:

- Cuadro: es la estructura que sostiene todas las demás partes de la bici
- Horquilla: es la pieza en forma de U a la que se fija la rueda delantera y la une con el cuadro
- Suspensión: es una pieza opcional que equipan algunas bicicletas de montaña para que el eje trasero de la bicicleta no sea rígido y pueda absorber las irregularidades del terreno
- Ruedas: son los elementos que unen la bicicleta con el suelo y le permiten desplazarse
- Manillar: es la pieza que permite controlar la dirección de la rueda delantera y donde se suelen encontrar los frenos y los cambios de marchas
- Frenos: es el sistema que permite detener el desplazamiento de la bicicleta mediante el accionamiento manual de unas palancas

- Transmisión: es el sistema que permite al ciclista convertir la fuerza de sus piernas en capacidad de movimiento para la bicicleta
- Sillín: la parte de la bicicleta donde se sienta el ciclista
- Sistema eléctrico: es el sistema que equipan las bicicletas eléctricas para generar movimiento a partir de electricidad.

[1]

La geometría de la bicicleta hace referencia a las medidas y ángulos de su cuadro.Los dos elementos fundamentales en la geometría del cuadro son lo que en inglés se conoce como: Stack y Reach. El Stack se podría definir como la altura del cuadro y el Reach como su largo o alcance.

- Stack: es la distancia vertical entre el centro del pedalier con respecto al centro del extremo superior del tubo de dirección.
- Reach: es la distancia horizontal desde el eje del pedalier a la parte superior del tubo de dirección.

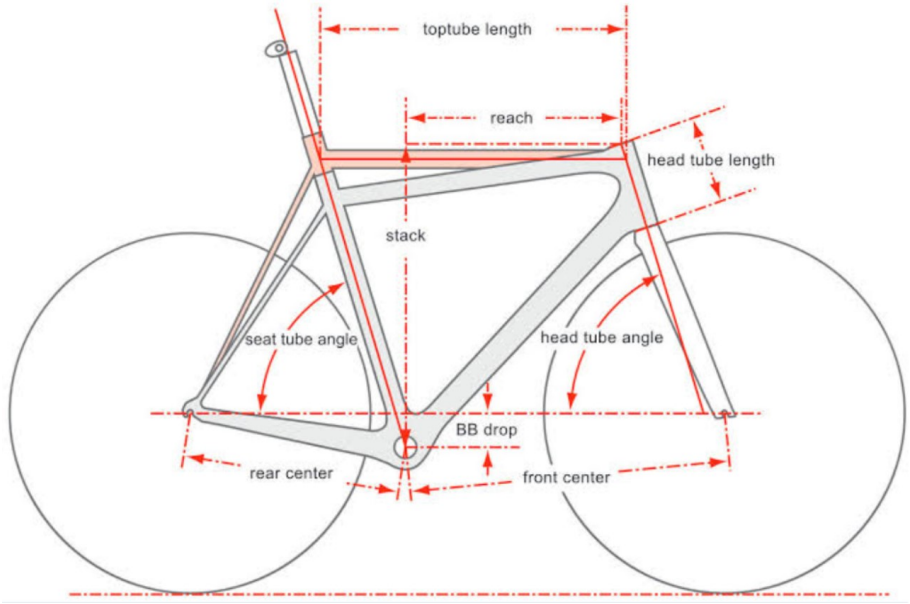

Figura 1: Ubicación del stack y del reach

[2]

4. Desarrollo

4.1. Estado de arte

En la literatura, se pueden encontrar multitud de enfoques para la resolución de problemas de optimización de la topolog´a. El documento original de Bendsoe y Kikuchi (1988) utilizó un enfoque basado en la microestructura u homogeneización basado en estudios de existencia de soluciones. El enfoque basado en la homogeneización se a adaptado en muchos trabajos, pero tiene la desventaja de que la determinación y evaluación de las microestructura y sus orientaciones es engorroso, si no se resuelve, y además, las estructuras resultantes no pueden construirse, ya que no hay una escala de longitud definida, ya que es asociada a las microestructura. Mas sin embargo, el enfoque

de homogeneización de la optimización topológica sigue siendo importante en el sentido de que puede proporcionar l´mites en el rendimiento teórico de las estructuras.

4.2. Pasos del desarrollo de la optimización

Para poder realizar el diseño propuesto del marco de la bicicleta, es necesario agregar algunas líneas de código, al script que usamos en la práctica 1, pero en su mayoría, el código será el mismo y funcionará de una manera similar. Para poder saber que necesitamos modificar, es necesario conocer la sintaxis de la función principal del código de optimización.

La función principal de el código es:

Practica1(nelx,nely,volfrac,penal,rmin)

donde:

- nelx es el número de elementos finitos que hay en la dirección horizontal.
- nely es el número de elementos finitos que hay en la dirección vertical.
- volfrac es la fracción de volumen en el dominio de diseño.
- penal es la penalización de las densidades intermedias. Una penalización alta hará la solución en blanco y negro, es decir los elementos finitos estarán llenos o vacíos.
- rmin es un radio de filtro para un filtro que hace que el diseño de malla-independiente.

Si nosotros utilizamos el código de optimización tal y como lo usamos en la práctica 1, no obtendremos el resultado deseado, debido a que el resultado final presentará inconvenientes en su diseño. Por lo tanto es necesario añadir las siguientes líneas de código:

for ely = 1:nely

for elx = 1:nelx

if $((\text{elx})^2 + (ely - nely)^2) < (0{,}65 * nelx)^2$

passive(ely,elx) = 1;

else

passive(ely,elx) = 0;

end

end

end

x(find(passive))=0.001;

También es necesario cambiar algunas líneas del código de optimización, esto se presenta en el codigo implementado

4.3. Código implemenado

function Practica2(nelx,nely,volfrac,penal,rmin)

x(1:nely,1:nelx) = volfrac;

loop = 0;
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;
change = 1.;
while change ¿0.01
loop = loop + 1;
xold = x;
[U]=FE(nelx,nely,x,penal);
[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)$^p enal * Ue' * KE * Ue$;
dc(ely,elx) = -penal*x(ely,elx)$^($penal $- 1) * Ue' * KE * Ue$;
end
end
[x] = OC(nelx,nely,x,volfrac,dc,passive);
change = max(max(abs(x-xold)));
disp(['It.:' sprintf('
sprintf('
' Vol.: ' sprintf('
' ch.: ' sprintf('
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)/l2 ¿1e-4)
lmid = 0.5*(l2+l1);
xnew(find(passive))=0.001
xnew = max(0.001,max(x-move,min(1.,min(x+move,x.*sqrt(-d
c./lmid)))));
if sum(sum(xnew)) - volfrac*nelx*nely ¿0;
l1 = lmid;
else
l2 = lmid;
end
end
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
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)$^{p}enal * KE$;

end

end

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);

U(freedofs,:) = K(freedofs,freedofs) (freedofs,:);

U(fixeddofs,:)= 0;

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)];

4.4. Resultados de la optimización

Haciendo las primeras pruebas, asignamos los siguientes valores a la función:

Practica2(12,12,0.33,3,0.9)

y obtenemos el siguiente resultado:

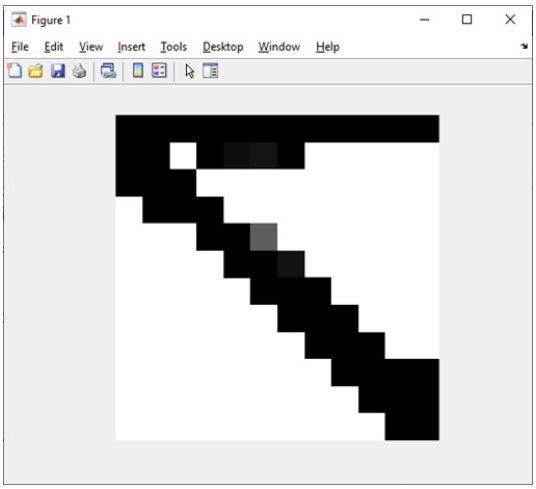

Figura 2: Resultado de la primera prueba

Podemos ir variando el valor de los parámetros de la función hasta obtener un resultado más cercano al esperado, como en las siguientes pruebas:

Practica2(16,16,0.33,3.0,1.2)

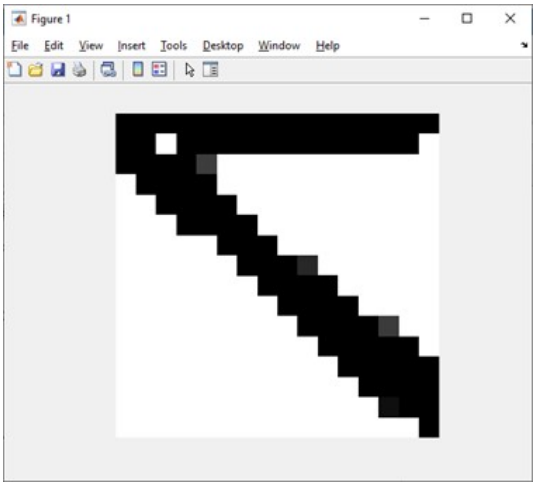

Figura 3: Segunda prueba

Practica2(20,20,0.33,3.0,1.5)

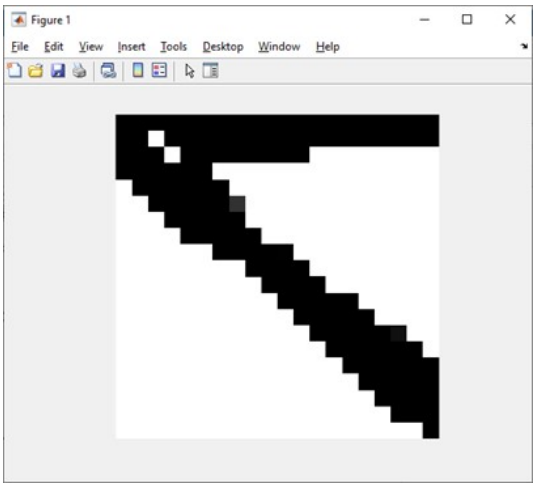

Figura 4: Tercera prueba

Con estos últimos parámetros usados, el resultado ya se asemeja más al marco de una bicicleta. Podemos realizar la comparación acerca de cómo funciona el filtro de mallado. En la última figura, el resultado tiene el filtro desactivado, ahora veremos como se ve con el filtro activado.

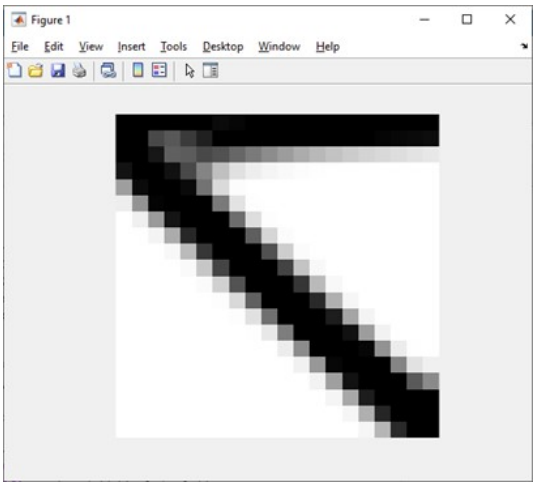

Figura 5: Resultado obtenido con el filtro de mallado activo

.

5. Conclusiones

- Bernardo Gil Villarreal 1991876

  En la elaboración de esta práctica consistió en la realización de un marco de bicicleta, donde se realizó un análisis en base a la programación de la misma práctica anterior y tomando como referencia la práctica pasada sólo tuvimos que modificar ciertos parámetros, en la cual cumplirían con el objeto a realizar. En el código de programación topológica tuvimos que colocar un valor de carga y un valor de fuerzas dentro del mismo diseño para que saliera el marco de bicicleta.

- Jovanny Daniel Alvarado Ramírez 1896681

Básicamente se puede concluir lo mismo que la practica pasada, sólo que en esta ocasión se trabajo con un caso mejor aplicado de la optimización topológica, donde se describio como propuesta un marco de bicicleta. Ya definida dicha propuesta, se presentaron distintos resultados que nos ayudaron a visualizar como el proceso de optimización topológica se aplica de distintas maneras para encontrar distintos resultados y por ende, el mejor.

- Fernando Herrera Garza 1909876

  El tamaño del mallado es muy aspecto muy importante a tomar en cuenta para el diseño final que obtendremos en el resultado. El aplicar un filtro nos sirve para hacer que los bordes del diseño final se vean suavizados, con lo que tenemos una imagen más limpia. Como vimos, el código de optimizacion nos sirve para realizar varios diseños, pero es necesario agregar líneas de código según sea el caso.

- Kevin Orlando Huerta Jaramillo 1854324

  Al igual que en la práctica anterior, se utilizó el software de matlab, el cual se le insertó el código de optimización, el cual nos permite mejorar el diseño del marco. Y al cual se le aplicará un filtro de malla que nos permite ver con más claridad el diseño del marco que deseamos obtener. Por lo cual esto nos permite determinar la precisión de la optimización.

- Aurora Nahomy Martinez Perez 1845056

  Durante esta práctica pudimos crear un cuadro de bicicleta el cual es la parte esencial de dicho mecanismo. Para poder desarrollar de manera correcta un cuadro es necesario tener en cuenta algunas medidas de la persona que usa la bici.En esta práctica se optó por aplicar un filtro el cual nos sirve para hacer que los bordes se suavicen.

Referencias

[1] Álex Esteban Fernández. Estudio y diseño de cuadros de bicicleta. B.S. thesis, Universitat Politècnica de Catalunya, 2021.

[2] Miguel Millet Heras. *Análisis, estudio y diseño de un cuadro de bicicleta elaborado con materiales compuestos.* PhD thesis, Universitat Politècnica de València, 2021.