

Práctica 2

Diseño del marco de una bicicleta

Equipo 4

Jorge Fuentes, Tania Hernandez, Anahi Herrera, Gustavo Díaz, Miriam Mata, Alejandro Ramos

23 de septiembre de 2022

Resumen

Objetivo: El estudiante deberá 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).

La geometría de la bici es algo así como el ADN de la misma. Un error de medidas y el mejor de los modelos puede fallar estrepitosamente. Te explicamos qué son las medidas de una bici y como influyen en su rendimiento. La geometría de una bici mide las longitudes de los tubos que la conforman, así como los ángulos que forman dichos tubos en la dirección y en el tubo de sillín principalmente. Los tubos se miden desde centro a centro y evidentemente no es necesario que la forma de los tubos sea convencional para medirlos. Lo que se mide es la longitud; no importa si el cuadro está realizado en algún tipo de monocasco o con tubería convencional o hidroformada. De este modo, además de la talla, que es el primer parámetro por el que elegimos una bici a nivel de medidas, la geometría es básica para que la bici se comporte de una manera u otra dependiendo del conjunto de medidas y ángulos.

Índice

1. Introducción	3
2. Desarrollo	3
2.1. Nombre y definición de la forma geometría	3
2.1.1. Marco de una bicicleta	3
2.1.2. Longitud del tubo de sillín (Seat Tube Lenght)	4
2.1.3. Ángulo de sillín (Seat Tube Angle)	4
2.1.4. Longitud del tubo de dirección (Head Tube Lenght)	5
2.1.5. Ángulo del tubo de dirección (Head Tube Angle)	5
2.1.6. Tubo superior (Top Tube)	6
2.1.7. Altura de pedalier (Bottom Bracket Height)	6
2.1.8. Caída del pedalier (Bottom Bracket Drop)	6
2.1.9. Longitud de vainas (Chain Stay Length)	7
2.1.10. Reach	7
2.1.11. Stack	7
2.1.12. Offset	8
2.1.13. Trail	8
2.1.14. Distancia entre ejes (Wheel Base)	8
2.1.15. Rake	8
2.2. Estado del arte	8
2.2.1. Longitudes y ángulos de la MTB	9
2.2.2. Algoritmos evolutivos	9
2.3. Pasos del desarrollo de la programación	10
2.4. Resultados de la optimización	12
3. Conclusiones	14
3.1. Jorge Fuentes	14
3.2. Tania Hernandez	14
3.3. Anahi Herrera	14
3.4. Gustavo Díaz	14
3.5. Miriam Mata	14
3.6. Alejandro Ramos	15

1. Introducción

En esta práctica se estará realizando el código de optimización topológica de un marco de una bicicleta. Esto para seguir comprendiendo el proceso de programación necesario en Matlab, y la aplicación que se tendría para futuras referencias. Para poder llevar a cabo la práctica se tuvo que realizar una investigación grupal para poder entender la metodología y para poder llegar a una implementación exitosa en el programa de MATLAB. Las evidencias de lo mismo se anexarán más adelante.

2. Desarrollo

2.1. Nombre y definición de la forma geometría

2.1.1. Marco de una bicicleta

Desde que se inventaron las bicicletas, su diseño se ha desarrollado con un fin, satisfacer las necesidades del usuario y mejorar su rendimiento independientemente de la disciplina en la que se emplee[3]. Actualmente, hay muchos tipos de bicicletas para diferentes estilos de conducción, sus diferencias están relacionadas con su rendimiento y calidad. Muchas de las diferencias están directamente relacionadas con las medidas del cuadro, medidas que conforman la geometría de una bicicleta.



Figura 1: Partes de una bicicleta.

2.1.2. Longitud del tubo de sillín (Seat Tube Length)

Es la distancia desde el centro del eje de pedalier (Bottom Bracket, BB en adelante) hasta el punto más alto del tubo. Esta medida suele definir la talla de la bicicleta, dependiendo del fabricante se suelen expresar en centímetros o en pulgadas.

En bicicletas de doble suspensión, también se representa una medida virtual de la longitud del tubo de sillín (Seat Tube, ST en adelante), porque la longitud real no representa la talla debido a las formas que suelen utilizar en los tubos para añadir los puntos de unión de la bieleta.



Figura 2: Longitud del tubo de sillín.

2.1.3. Ángulo de sillín (Seat Tube Angle)

Es la medida que representa el ángulo formado entre el ST (Seat Tube) y el plano horizontal (Suelo). El rango que los fabricantes solían trabajar estaba entre 69° y 74° dependiendo de la modalidad.

Pero la evolución de la geometría de la bicicleta, la aparición de nuevas modalidades y la implementación de los nuevos tamaños de rueda como de la biomecánica las medidas dieron un gran giro y en concreto el ángulo de sillín (Seat Tube Angle – STA).

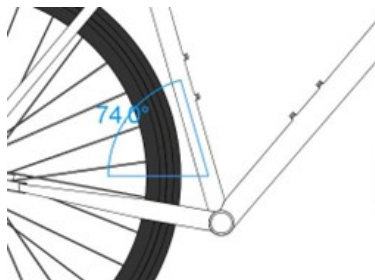


Figura 3: Ángulo de sillín.

2.1.4. Longitud del tubo de dirección (Head Tube Length)

Esta es una de las medidas que indica las instrucciones ergonómicas del cuadro.

La longitud del tubo de dirección (Head Tube Length – HTL) nos va a proporcionar la altura de la parte delantera de la bicicleta haciendo que varíe la posición del ciclista.



Figura 4: Longitud del tubo de dirección.

2.1.5. Ángulo del tubo de dirección (Head Tube Angle)

Es el ángulo que forma el tubo de dirección (Head Tube) con respecto al plano horizontal (Suelo). Influyen directamente en el comportamiento de la bicicleta. Un ángulo de dirección muy abierto ayudará a mejorar las subidas y estable en los tramos llanos, con una posición y manejabilidad más cómoda. También provoca que la bicicleta tenga un comportamiento direccional más directo, nervioso e inestable en las bajadas. Un ángulo de dirección cerrado se emplea en bicicletas de enduro y descenso donde necesitamos que la bicicleta sea estable y ataque muy bien los obstáculos.

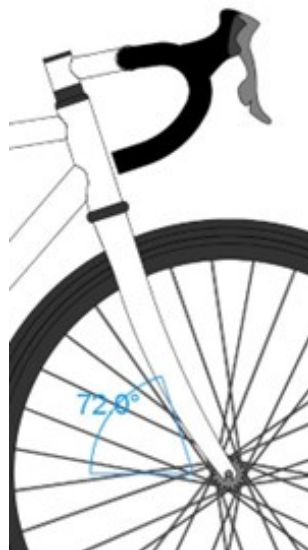


Figura 5: Ángulo del tubo de dirección.

2.1.6. Tubo superior (Top Tube)

Es el tubo que une el Head Tube (Tubo de dirección) y el Seat Tube (Tubo de sillín). Su medida determina la distancia que hay desde el centro de HT a centro de ST. Su valor varía con respecto a la talla de la bicicleta, siendo mas largo cuando mas grande sea esta.

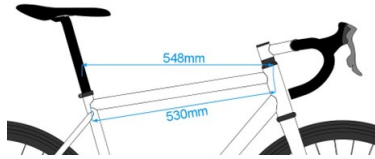


Figura 6: Tubo superior.

2.1.7. Altura de pedalier (Bottom Bracket Height)

Es la distancia vertical desde el centro del eje pedalier al suelo. Determinar el espacio que tendrá la bicicleta para superar obstáculos sin golpear con el cuadro o los platos. Espacio determinante para la estabilidad y maniobrabilidad de la bicicleta.

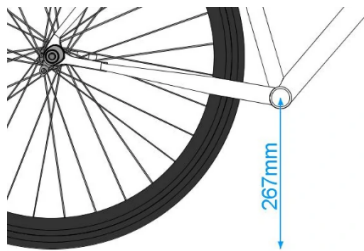


Figura 7: Altura de pedalier.

2.1.8. Caída del pedalier (Bottom Bracket Drop)

Puede parecer que es una medida que va relacionada con la altura de la caja de pedalier, pero no, es totalmente independiente. Distancia que se toma entre la línea horizontal imaginaria que une los ejes de las ruedas y el punto central del eje de pedalier. El valor vertical que hay entre estos dos puntos es el BB Drop (Caída del pedalier).



Figura 8: Caída del pedalier.

2.1.9. Longitud de vainas (Chain Stay Length)

Es la longitud que existe entre el centro del eje de pedalier y el centro del eje de la rueda trasera. Longitud que hará que la bicicleta sea más o menos estable al mismo tiempo que rápida o lenta en subidas y bajadas.

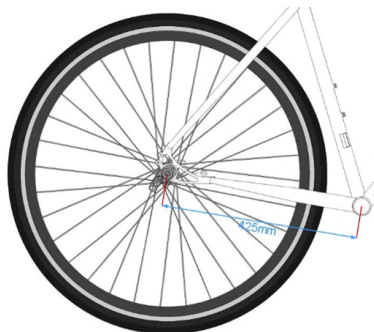


Figura 9: Longitud de vainas.

2.1.10. Reach

Imaginemos una línea vertical que cruza el centro del eje de pedalier. El Reach es la distancia horizontal desde esta línea hasta la parte superior del centro del tubo de dirección.

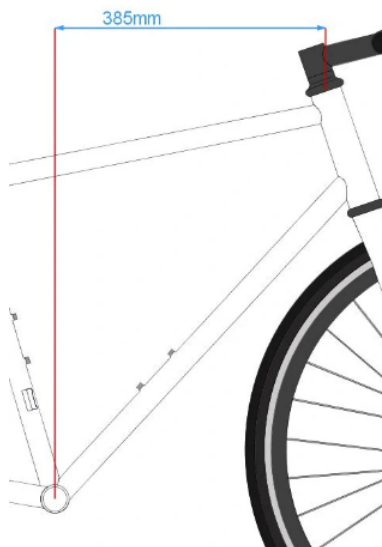


Figura 10: Reach.

2.1.11. Stack

El Stack, o altura del cuadro, es la distancia vertical que existe entre el centro del eje de pedalier y la parte superior del centro del tubo de dirección.

2.1.12. Offset

Esta es una de las medidas que más confusión produce en la geometría de la bicicleta junto con el trail. El offset se mide trazando una línea vertical en el eje de la rueda delantera y otra prolongando la línea imaginaria de tubo de dirección. La diferencia de distancia entre una y otra es el Offset.

2.1.13. Trail

A simple vista podría parecer la misma medida, pero no se parecen en nada. El trail de una horquilla es la distancia entre el punto de contacto de la cubierta con el suelo y donde la línea imaginaria del Head Tube hace contacto con el suelo.

2.1.14. Distancia entre ejes (Wheel Base)

En el mundo de la automoción esta medida se conoce como batalla. Cuando hablamos de la geometría de la bicicleta es la distancia que existe entre el centro de los ejes de las ruedas. El Wheel Base varía mucho el comportamiento de la bicicleta tanto en estabilidad como en capacidad de reacción. A menor distancia entre ejes, más rápida de reacción y ágil es la bicicleta, pero a la vez reduce su estabilidad y, sobre todo, la seguridad a altas velocidades. Cuando aumentamos la distancia, la bicicleta es más lenta o torpe en los giros rápidos, pero será mucho más estable y segura a velocidades elevadas.

2.1.15. Rake

El rake de una horquilla es el avance del eje de la rueda sobre la línea del tubo de dirección. Dirás que es lo mismo que el Offset, pero no, no es lo mismo. Similar y relacionado.

2.2. Estado del arte

La geometría de la bicicleta es algo así como el ADN de la misma. Un error de medidas y el mejor de los modelos puede fallar estrepitosamente. Te explicamos qué son las medidas de una bici y como influyen en su rendimiento. La geometría de una bici mide las longitudes de los tubos que la conforman, así como los ángulos que forman dichos tubos en la dirección y en el tubo de sillín principalmente. Los tubos se miden desde centro a centro y evidentemente no es necesario que la forma de los tubos sea convencional para medirlos. Lo que se mide es la longitud; no importa si el cuadro está realizado en algún tipo de monocasco o con tubería convencional o hidroformada. De este modo, además de la talla, que es el primer parámetro por el que elegimos una bici a nivel de medidas, la geometría es básica para que la bici se comporte de una manera u otra dependiendo del conjunto de medidas y ángulos.

2.2.1. Longitudes y ángulos de la MTB

La geometría de una bici mide las longitudes de los tubos que la conforman, así como los ángulos que forman dichos tubos en la dirección y en el tubo de sillín principalmente. Los tubos se miden desde centro a centro y evidentemente no es necesario que la forma de los tubos sea convencional para medirlos. Lo que se mide es la longitud; no importa si el cuadro está realizado en algún tipo de monocasco o con tubería convencional o hidroformada. De este modo, además de la talla, que es el primer parámetro por el que elegimos una bici a nivel de medidas, la geometría es básica para que la bici se comporte de una manera u otra dependiendo del conjunto de medidas y ángulos.

Te explicamos todo con detalle más adelante, pero te queremos contar un poco de antemano por qué la geometría afecta tanto al comportamiento de una bici. Los dos parámetros más importantes de una bici son los ángulos de dirección y de sillín. El de dirección va a hacer que la bici sea más estable, o que gire más rápido y que tenga una mayor viveza de reacciones. Esto sumado a una mayor seguridad a la hora de bajar y una absorción de impactos más efectiva, por el propio ángulo que forma la horquilla sobre el terreno por el que pasamos. Ángulos de dirección más verticales (entre 67 y 70 grados) son más propios de modelos de cross country. Ángulos de dirección más cerrados (64-65 grados), son utilizados en los modelos de enduro. Menos de 64 grados son para modelos de descenso y también para algunos modelos de enduro.

En cuanto al ángulo de sillín, determina la posición donde nos sentamos a la hora de pedalear, dependiendo de si estamos muy lejos o demasiado encima del eje de pedalier. Esto influye también en el reparto de pesos de la bici así como en la manejabilidad de la misma. Normalmente oscilan entre los 73 y 77 grados. En los últimos 3 años ha habido un avance muy rápido y radical sobre el ángulo de sillín, ya que se ha ido verticalizando más, hasta llegar a cifras de hasta 76 y 78 grados. Esta tendencia es básica para que el pedaleo sea más efectivo al situarnos más encima en la vertical del eje de pedalier. Y esto se aplica tanto a cross country, como trail, enduro, descenso y también las ebikes.

2.2.2. Algoritmos evolutivos

Los algoritmos evolutivos son estrategias de optimización y búsqueda de soluciones que toman como inspiración la evolución en distintos sistemas biológicos. La idea fundamental de estos algoritmos es mantener un conjunto de individuos que representan una posible solución del problema. Estos individuos se mezclan y compiten entre sí, siguiendo el principio de selección natural por el cual solo los mejor adaptados sobreviven al paso del tiempo. Esto redundará en una evolución hacia soluciones cada vez más aptas[1].

Los algoritmos evolutivos son una familia de métodos de optimización, y como tal, tratan de hallar una tupla de valores (x_1, \dots, x_n) tales que se minimice una determinada función $F(x_1, \dots, x_n)$. En un algoritmo evolutivo, tras parametrizar el problema en una serie de variables, (x_1, \dots, x_n) se codifican en una población de cromosomas. Sobre esta población se aplican uno o varios operadores genéticos y se fuerza una presión selectiva (los operadores utilizados se aplicarán sobre estos cromosomas, o sobre poblaciones de ellos). Esta forma de funcionamiento les confiere su característica más destacable: un algoritmo evolutivo puede ser implementado de forma independiente del problema, o a lo sumo, con un conocimiento básico de este, lo cual los hace algoritmos robustos, por ser útil para cualquier problema, pero a la vez débiles, pues no están especializados en ningún problema concreto siendo los operadores genéticos empleados los que en gran parte confieren la especificabilidad al método empleado[2].

2.3. Pasos del desarrollo de la programación

A continuación, veremos la codificación de la programación en MATLAB.

```
1 %%%% A 99 LINE TOPOLOGY OPTIMIZATION CODE BY OLESIGMUND, OCTOBER 1999 %%%
2 function topp1(nelx,nely,volfrac,penal,rmin)
3 % INITIALIZE
4 x(1:nely,1:nelx) = volfrac;
5 loop = 0;
6 for ely = 1:nely
7 for elx = 1:nelx
8 if ((elx)^2+(ely-nely)^2) < (0.65*nelx)^2
9 passive(ely,elx) = 1;
10 else
11 passive(ely,elx) = 0;
12 end
13 end
14 end
15 x(find(passive))=0.001;
16 change = 1.;
17 % START ITERATION
18 while change > 0.01
19 loop = loop + 1;
20 xold = x;
21 % FE-ANALYSIS
22 [U]=FE(nelx,nely,x,penal);
23 % OBJECTIVE FUNCTION AND SENSITIVITY ANALYSIS
24 [KE] = lk;
25 c = 0.;
26 for ely = 1:nely
27 for elx = 1:nelx
28 n1 = (nely+1)*(elx-1)+ely;
29 n2 = (nely+1)* elx +ely;
30 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);
31 c = c + x(ely,elx)^penal*Ue'*KE*Ue;
32 dc(ely,elx) = -penal*x(ely,elx)^(penal-1)*Ue'*KE*Ue;
33 end
34 end
35 % FILTERING OF SENSITIVITIES
36 %[dc] = check(nelx,nely,rmin,x,dc);
37 % DESIGN UPDATE BY THE OPTIMALITY CRITERIA METHOD
38 [x] = OC(nelx,nely,x,volfrac,dc,passive);
39 % 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 % PLOT DENSITIES
45 colormap(gray); imagesc(-x); axis equal; axis tight; axis off; pause(1e-6);
46 end
47 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
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(find(passive))=0.001
53 xnew = max(0.001,max(x-move,min(1.,min(x+move,x.*sqrt(-dc./lmid)))));
54 if sum(sum(xnew)) - volfrac*nelx*nely > 0;
55 l1 = lmid;
56 else
57 l2 = lmid;
58 end
59 end
60 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
61 function [dcn]=check(nelx,nely,rmin,x,dc)
62 dcn=zeros(nely,nelx);
63 for i = 1:nelx
64 for j = 1:nely
```

```

65 sum=0.0;
66 for k = max(i-round(rmin),1):min(i+round(rmin),nelx)
67 for l = max(j-round(rmin),1):min(j+round(rmin),nely)
68 fac = rmin-sqrt((i-k)^2+(j-l)^2);
69 sum = sum+max(0,fac);
70 dcn(j,i) = dcn(j,i) + max(0,fac)*x(l,k)*dc(l,k);
71 end
72 end
73 dcn(j,i) = dcn(j,i)/(x(j,i)*sum);
74 end
75 end
76 %%%%%%%%%%FE-ANALYSIS %%%%%%%%%%
77 function [U]=FE(nelx,nely,x,penal)
78 [KE] = lk;
79 K = sparse(2*(nelx+1)*(nely+1), 2*(nelx+1)*(nely+1));
80 F = sparse(2*(nely+1)*(nelx+1),1); U =sparse(2*(nely+1)*(nelx+1),1);
81 for ely = 1:nely
82 for elx = 1:nelx
83 n1 = (nely+1)*(elx-1)+ely;
84 n2 = (nely+1)* elx +ely;
85 edof = [2*n1-1; 2*n1; 2*n2-1; 2*n2; 2*n2+1;2*n2+2;2*n1+1; 2*n1+2];
86 K(edof,edof) = K(edof,edof) + x(ely,elx)^penal*KE;
87 end
88 end
89 % DEFINE LOADSAND SUPPORTS(HALF MBB-BEAM)
90 F(2,1) = 1;
91 fixeddofs =2*nelx*(nely+1)+1:2*(nelx+1)*(nely + 1);
92 alldofs = [1:2*(nely+1)*(nelx+1)];
93 freedofs = setdiff(alldofs,fixeddofs);
94 % SOLVING 127
95 U(freedofs,:) = K(freedofs,freedofs) \F(freedofs,:);
96 U(fixeddofs,:)= 0;
97 %%%%%%%%%%ELEMENT STIFFNESS MATRIX %%%%%%%%%%
98 function [KE]=lk
99 E = 1.;
100 nu = 0.3;
101 k=[ 1/2-nu/6 1/8+nu/8 -1/4-nu/12 -1/8+3*nu/8 ...
102 -1/4+nu/12 -1/8-nu/8 nu/6 1/8-3*nu/8];
103 KE = E/(1-nu^2)*[ k(1) k(2) k(3) k(4) k(5) k(6) k(7) k(8)
104 k(2) k(1) k(8) k(7) k(6) k(5) k(4) k(3)
105 k(3) k(8) k(1) k(6) k(7) k(4) k(5) k(2)
106 k(4) k(7) k(6) k(1) k(8) k(3) k(2) k(5)
107 k(5) k(6) k(7) k(8) k(1) k(2) k(3) k(4)
108 k(6) k(5) k(4) k(3) k(2) k(1) k(8) k(7)
109 k(7) k(4) k(5) k(2) k(3) k(8) k(1) k(6)
110 k(8) k(3) k(2) k(5) k(4) k(7) k(6) k(1)];

```

Run: topp1(20,20,0.33,3.0,1.5)

Run: topp1(20,20,0.33,3,1.5)

Run: topp1(12,12,0.33,3.0,0.9)

Run: topp1(16,16,0.33,3.0,1.2)

Figura 11: Configuraciones iniciales y análisis de sensibilidad.

2.4. Resultados de la optimización

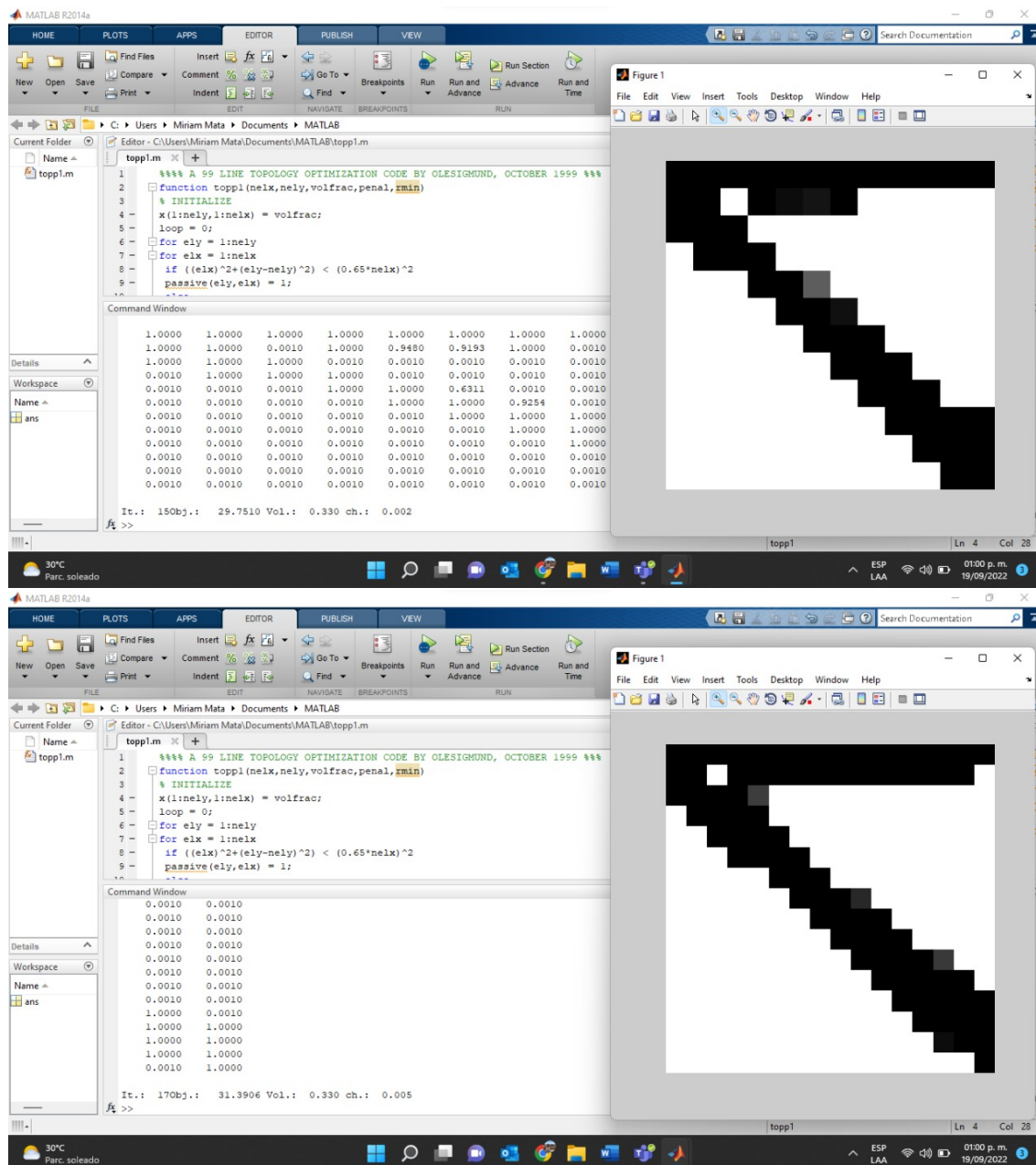


Figura 12: Resultados de la Optimización topológica 1.

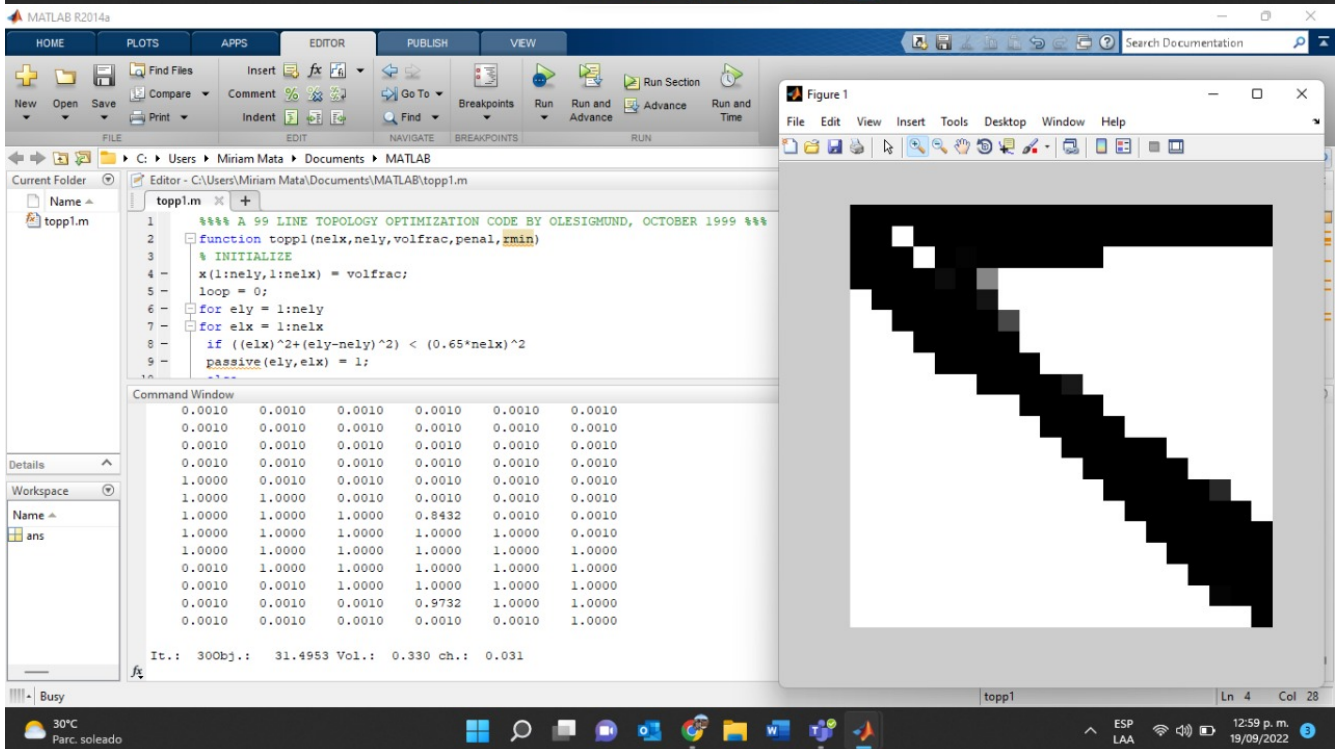
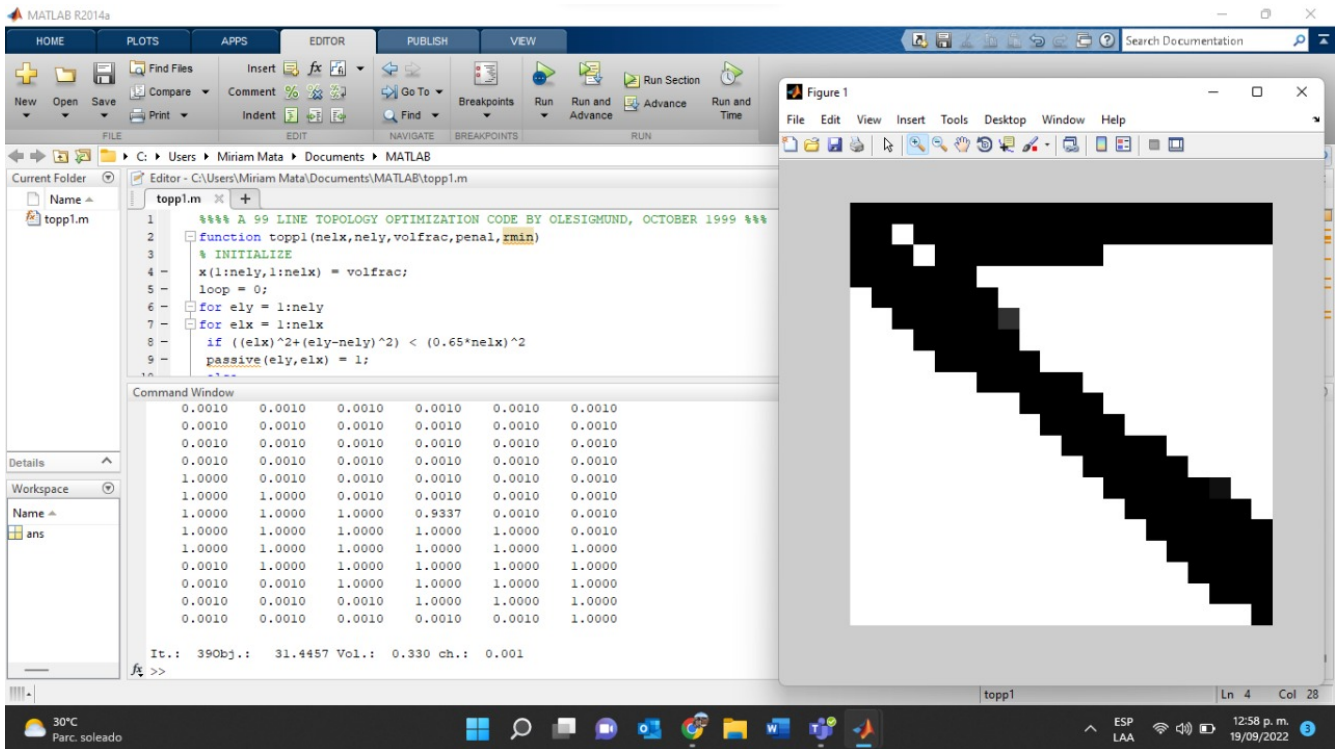


Figura 13: Resultados de la Optimización topológica 2.

3. Conclusiones

3.1. Jorge Fuentes

Trabajar nuevamente con MATLAB fue gratificante para mí, aplicando lo de la práctica anterior sobre la simulación de matrices llegamos a reforzar los conocimientos comprendidos y volverlos aplicar en esta práctica as entendiendo sobre que estaba haciendo exactamente el código utilizado en nuestra práctica. Una vez entendido fue fácil modificar las variables para ver los resultados que mostraba la optimización de la topología de los elementos que estábamos usando. La ejecución del análisis así como se hace el código en MATLAB fueron de gran ayuda para repasar la programación, ya que, como sabemos la optimización topológica va de la mano con la biomecánica al momento de optimizar los elementos a un grado óptimo y funcional. Me gusto saber mucho sobre la optimización del marco de una bicicleta no sabía mucho sobre sus partes y fue interesante ver estos detalles que desconocía durante la investigación.

3.2. Tania Hernandez

La práctica realizada consistió en la elaboración de un marco de bicicleta o también conocido como cuadro, con el proceso de realización, me pude dar cuenta de lo complejo que es hacerla, desde la parte del funcionamiento que sea practico y aplicable, se puede decir que el marco de la bicicleta, su geometría es realizada con el fin de que la posición y la medida de la persona sea la deseable indicada para el tamaño del cuadro, ya que cada largo, ancho, grados de inclinación, etc., Está pensado para las medidas estándares de cada individuo.

3.3. Anahi Herrera

Gracias a la elaboración de esta práctica se entendió de una mejor manera el funcionamiento de MATLAB aplicado hacia la materia de biomecánica, se realizaron investigaciones grupales llevando esta teoría recopilada a la práctica de manera exitosa.

3.4. Gustavo Díaz

Dentro de esta practica se vio un poco de los temas vistos dentro de la practica pasada por ende se pudieron reforzar dichos conocimientos de la practica pasada al ambas buscar la optimizacion de una pieza esto hace un poco mas sencilla esta practica, en este caso vimos la manera de optimizar la manera de hacer un marco de bicicleta dentro de nuestra investigacion nos dimos cuenta que en como algo tan simple comolo es un marco de bicicleta un cambio de unos milímetros puede hacer que su estructura se vuelva demasiado fragil para poder usarse por ende al hacer la optimizacion fue de ayuda ya que pudimos observar diferenes cuestiones que se ven afectadas al momento de modificarse. Al ser similar a la practica pasada y solo cambiar algunos valores el codigo de la practica pasada nos sirvio de apoyo para esta misma.

3.5. Miriam Mata

Esta práctica estuvo muy interesante porque a pesar de ya haber de usar Matlab en semestres anteriores en lo personal nunca había usado un código como este, fue un poco complicado, pero se logró, siento que a nosotros como futuros ingenieros en mecatrónica se nos debió de haber inculcado más la programación desde los primeros semestres, ya que si bien la mayoría se va a el área de diseño en cierta parte seguimos viendo programación, pero esta actividad me ayudo para poder refrescar un poco mis conocimientos en Matlab porque en sí es un compilador muy sencillo de usar también depende mucho de la librería que te sepas por qué va cambiando cada año.

3.6. Alejandro Ramos

En esta práctica me interesó más que ya nos comenzamos a introducir un poco más en el área de ponernos a diseñar algo un poco más avanzado con respecto a lo anterior y que gracias a esto seguimos avanzando en aprender nuevas cosas y el funcionamiento de las herramientas utilizadas me gusta como va desarrollándose las prácticas y espero seguir aprendiendo hasta lograr hacer las cosas lo mejor posible.

Referencias

- [1] V. J. Challis. A discrete level-set topology optimization code written in matlab., December 2010.
- [2] Yamada T. Izui K. Nishiwaki S. Otomori, M. Matlab code for a level set-based topology optimization method using a reaction diffusion equation., March 2015.
- [3] K. Suresh. How is the geometry of a bicycle., January 2021.