COMPUTER-AIDED DESIGN COURSE WITH MATLAB

Assoc. Prof. Recep Muhammet Görgülüarslan
TOBB University of Economics and Technology
rgorguluarslan@etu.edu.tr

Computer-Aided Design Course
Course description and learning objectives

- Offered every spring semester
- Undergraduate level senior (4th year) students
- Mechanical Engineering
- Physical

Learning objectives:

- To introduce fundamental technologies underlying Computer-Aided Design (CAD) and implementing in MATLAB.
  - To show how they can be used to aid engineering design, not just the geometry documentation.
- To provide an open classroom where students can learn by doing,
  - To relate the course material to their research and experience.

Computer-Aided Design Course
Reason for adoption MATLAB

- To teach the computer-aided design (CAD) fundamentals by implenting them in MATLAB.
- The topics are as follows:

  1) Primitive Instance Modeling

  2) Rigid Body Transformations (Scaling, Translation, and Rotation of Models)

  3) Curve and Surface Modeling

  4) Topology Optimization

  5) CAD to Additive Manufacturing (3D Printing)

How MATLAB and Simulink were used in the course

MATLAB is used in class practices and additional activities:

- 5 Homework Assignments (35% of grade)
  - Significant Matlab programming is required: part modeling, curve and surface modeling, topology optimization.
- A Final Project based on MATLAB programming (35% grade)
  - Select projects and groups based on your interests and/or research.
  - Topology Optimization + Surface modeling + STL model conversion + 3D printing in MATLAB

1) PRIMITIVE INSTANCE MODELING IN MATLAB

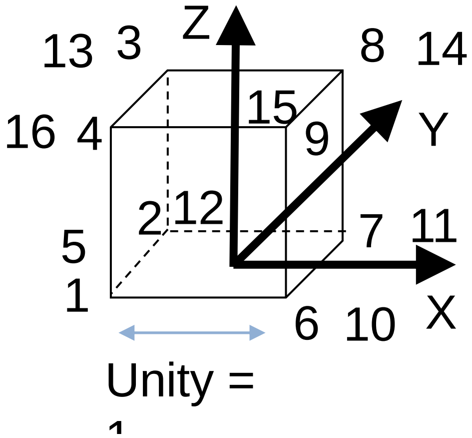
- Learning primitive instancing (creating simple shapes) and MATLAB

•Primitives = simple shapes

•Instancing = creating a primitive

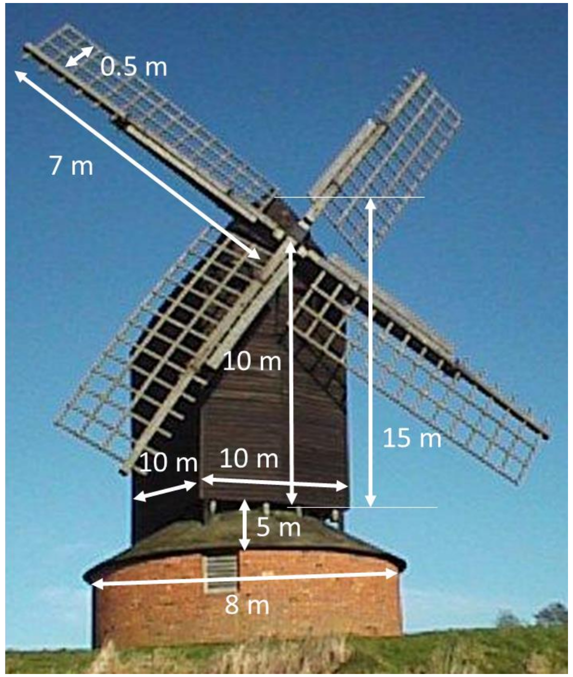
• Learn how to plot graphs in MATLAB starting by plotting a simple cube.

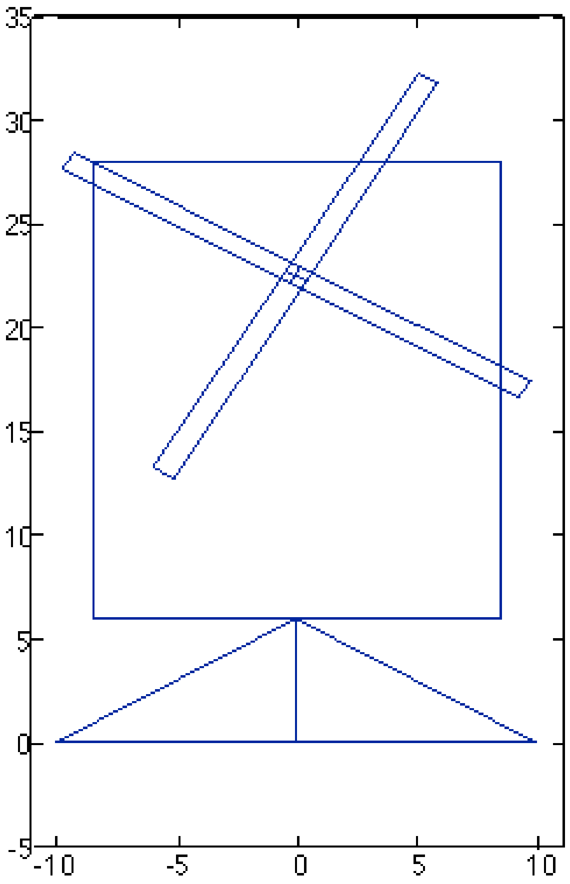
Unit Cube



Cube = [-0.5, -0.5, -0.5, -0.5, -0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, -0.5, -0.5, 0.5, 0.5, -0.5;
0, 0, 1, 1, 0, 0, 0, 1, 1, 0, 0, 0, 1, 1, 1, 1;
-0.5, 0.5, 0.5, -0.5, -0.5, -0.5, 0.5, 0.5, -0.5, -0.5, 0.5, 0.5, 0.5, 0.5, -0.5, -0.5;
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1];

Plotting a wind mill in MATLAB using primitive instances.



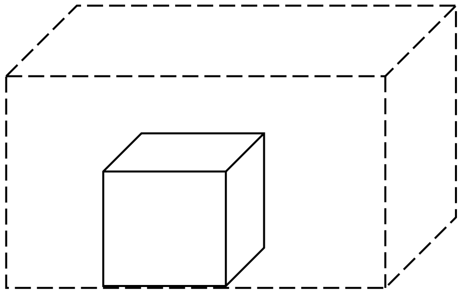


2) RIGID BODY TRANSFORMATIONS

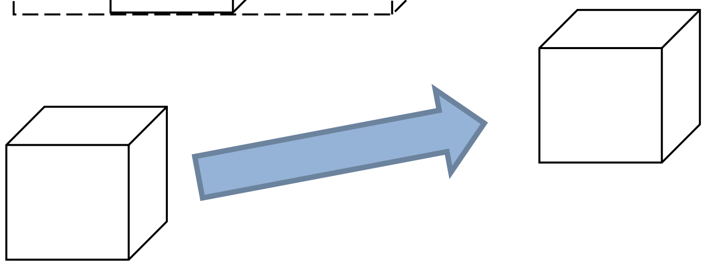
- Describe relative motion among parts in an assembly.
- Size, position and orient features in a part (rigid body transformations).
- Describe CAD model views.
- Apply rigid body transformations in MATLAB in a wind mill example and in a homework assignment

3 OPERATIONS:

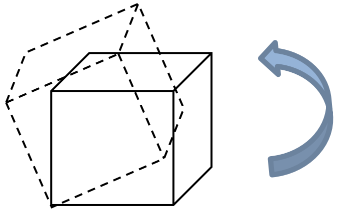
1) SCALING:

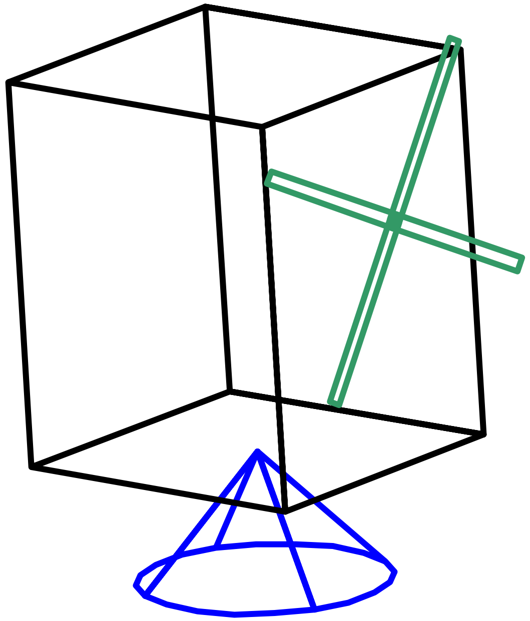


2) TRANSLATION:



3) ROTATION:



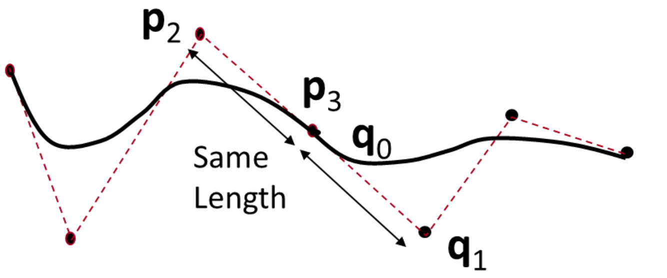


3) CURVE AND SURFACE MODELING

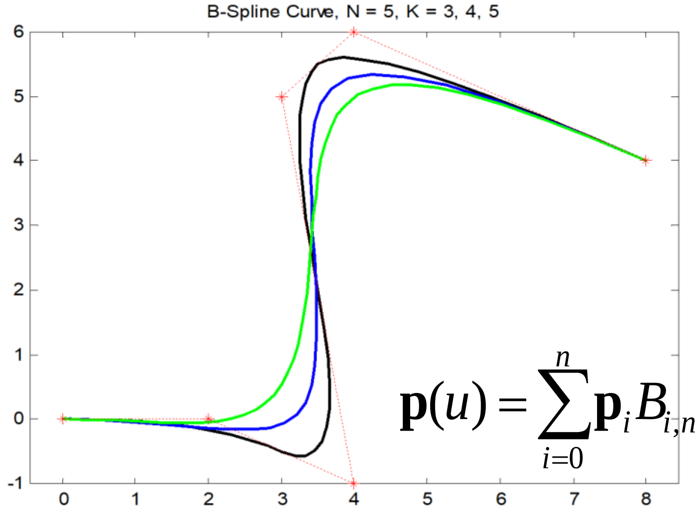
- Parametric Curves
  - Hermite Curves
  - Bezier Curves
  - B-spline Curves
  - NURBS
- A homework assignment to apply them in MATLAB

- Parametric Surfaces
  - Hermite Curves
  - Bezier Curves
- A homework assignment to apply them in MATLAB

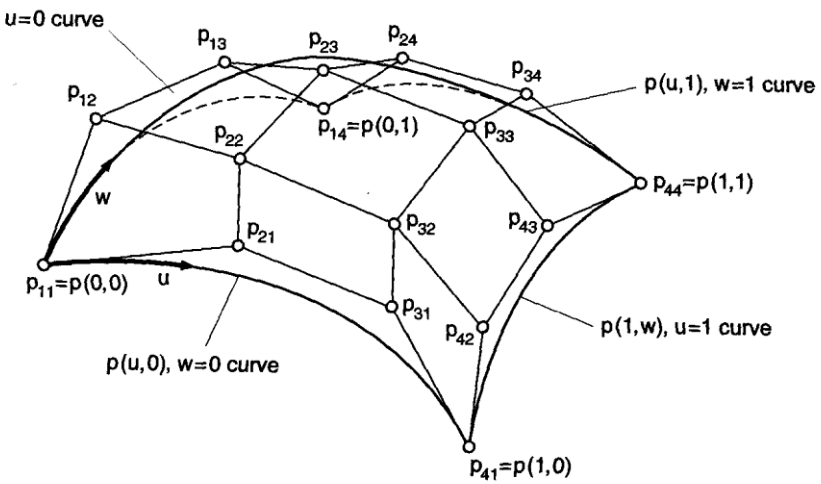
- Continuity in Curves and Surfaces

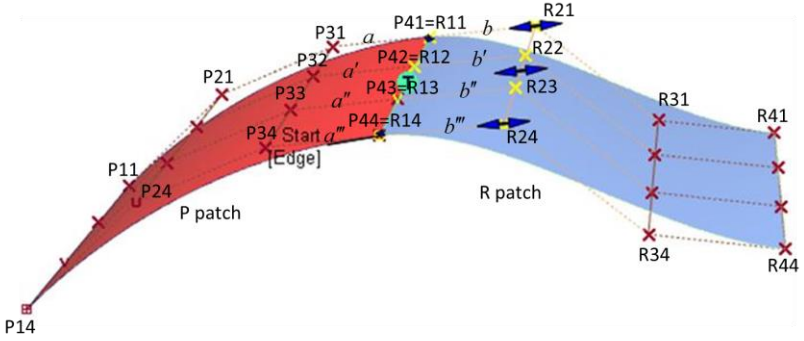


$$\mathbf{p}(u,w)=\sum_{i=0}^{m}\sum_{j=0}^{n}\mathbf{p}_{ij}B_{i,m}(u)B_{j,n}(w)$$



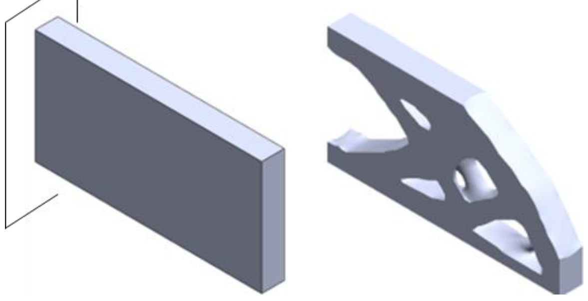
$$\mathbf{p}(u)=\sum_{i=0}^{n}\mathbf{p}_iB_{i,n}(u)$$





4) TOPOLOGY OPTIMIZATION

Removing material from an initial layout



Compliance minimization

Compliance — Displacement Vector — Stiffness matrix

$$\min : \quad C = \sum_{i=1}^{n} u_i{}^T k_i u_i$$

Subject to: $V(\rho) - f\,V_0 \leq 0$

$; KU = F$ →

$; 0 < \rho_{min} \leq \rho \leq 1$

$V(\rho)$: Optimized Volume
f : Volume fraction
V_0 : Initial total volüme
F : Force vector

$$\mathbf{k}_i(\rho_i) = \int_{-1}^{+1}\int_{-1}^{+1}\int_{-1}^{+1} \mathbf{B}^{\mathrm{T}}\mathbf{C}_i(\rho_i)\mathbf{B}\,d\xi_1 d\xi_2 d\xi_3$$

$$\mathbf{C}_i(\rho_i) = \mathrm{E}_i(\rho_i)\mathbf{C}_i^0$$

Solid Isotropic Material Penalization (SIMP)

$$E_i = E_0 \rho_i^p$$

For isotropic material

$$\mathbf{C}_i^0 = \frac{1}{(1+\nu)(1-2\nu)} \times \begin{bmatrix} 1-\nu & \nu & \nu & 0 & 0 & 0 \\ \nu & 1-\nu & \nu & 0 & 0 & 0 \\ \nu & \nu & 1-\nu & 0 & 0 & 0 \\ 0 & 0 & 0 & (1-2\nu)/2 & 0 & 0 \\ 0 & 0 & 0 & 0 & (1-2\nu)/2 & 0 \\ 0 & 0 & 0 & 0 & 0 & (1-2\nu)/2 \end{bmatrix}$$

OPTIMIZATION ALGORITHM

Optimality Criteria:

A gradient-based algorithm. Recall update scheme: $\mathbf{X}^{k+1} = \mathbf{X}^{k} - \alpha \nabla f^{k}(\mathbf{X})$

A heuristic update scheme is used in Optimality Criteria Method:

$$x_e^{\text{new}} = \begin{cases} \max(0, x_e - m) & \text{if } x_e B_e^{\eta} \le \max(0, x_e - m) \\ \min(1, x_e + m) & \text{if } x_e B_e^{\eta} \ge \min(1, x_e - m) \\ x_e B_e^{\eta} & \text{otherwise} \end{cases}$$

m=0.2, η=0.5 is recommended.

B_e are obtained from the optimality condition defined by the derivatives:

$$\frac{\partial c(\tilde{\mathbf{x}})}{\partial x_e} + \lambda \frac{\partial v(\tilde{\mathbf{x}})}{\partial x_e} = 0 \Rightarrow B_e = \frac{-\frac{\partial c}{\partial x_e}}{\lambda \frac{\partial V}{\partial x_e}}$$

$$\frac{\partial c}{\partial x_e} = -p x_e^{p-1}(E_0 - E_{\min})\mathbf{u}_e^{\mathrm{T}}\mathbf{k}_0\mathbf{u}$$

$$\frac{\partial V}{\partial x_e} = v_e$$ = 1 for unit volume elements

λ is the lagrange multiplier.

It is determined by the bisection algorithm such that it satisfies the volume constraint:

$\lambda = 0.5 * (l_2+l_1)$ Initially l1 = 0; l2 = 1e9

Stops when (l2-l1)/(l1+l2) > 1e-3

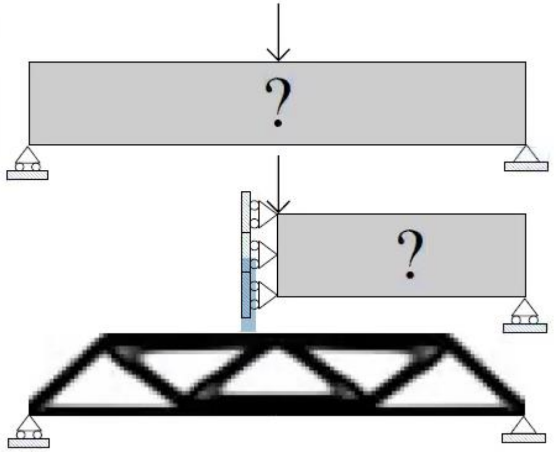
$l_1=\lambda$ if Vol constraint is not satisfied.

$l_2=\lambda$ if Vol constraint is satisfied.

MATLAB CODES FOR TOPOLOGY OPTIMIZATION

http://www.topopt.dtu.dk/

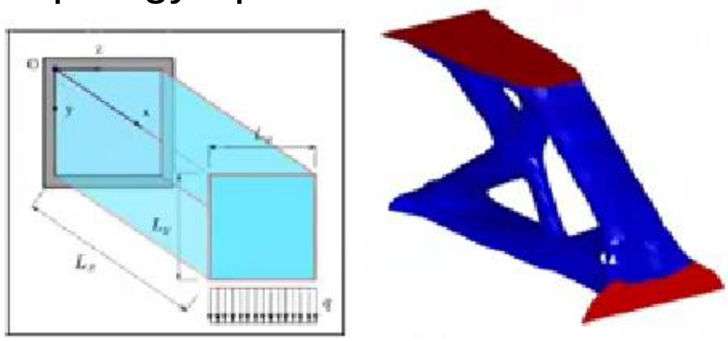
2D topology optimization code



Software produced by the Technical University of Denmark

- They offer software for free download.
- Good summary and many examples
- Provides also 2D/3D computer code that we will analyze.
- Can get also a mobile phone ap that would do for you topology optimization.
- Exercise: Use the top ap or the web site to design a bar in tension with aspect ratio of 3, with the tensile loads applied at two corners of the rectangle.

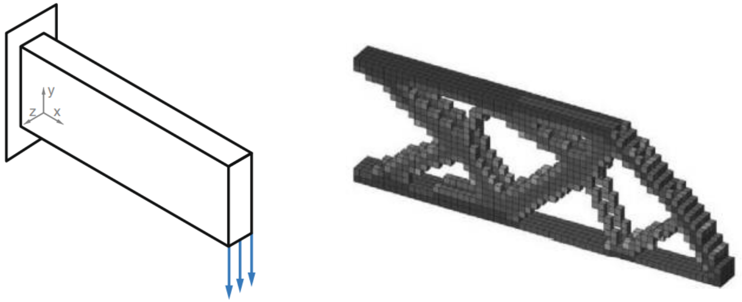
3D topology optimization code



top3d

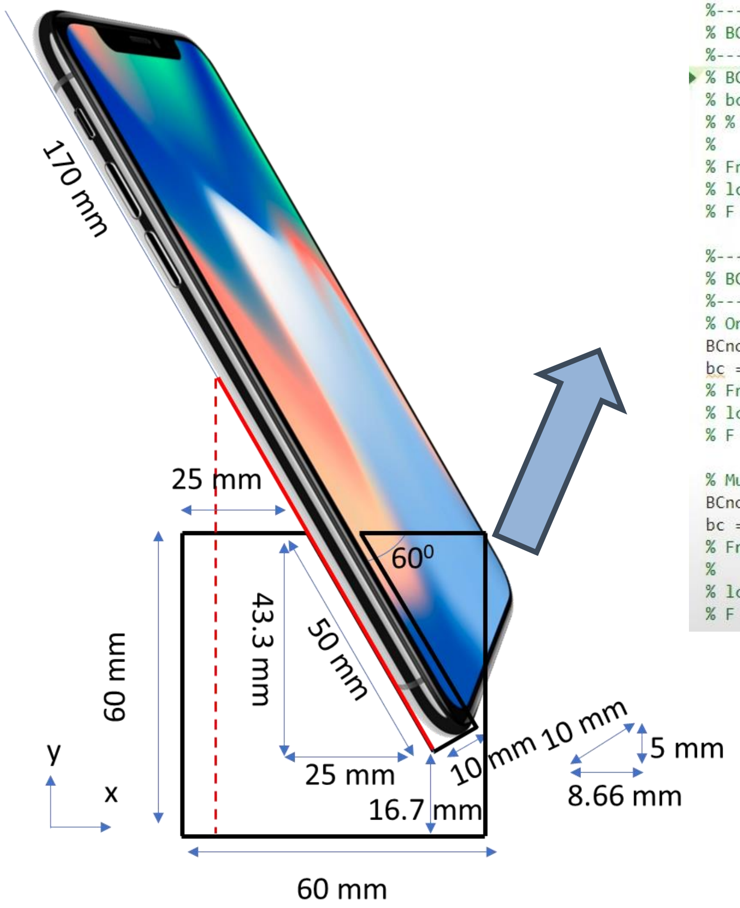
Top3d is a **free** MATLAB® program that solves topology optimization problem in 3D. The program is:

- compact without sacrificing readability,
- easy-to-use without losing complexity, and
- multi-functional without programming immensity.

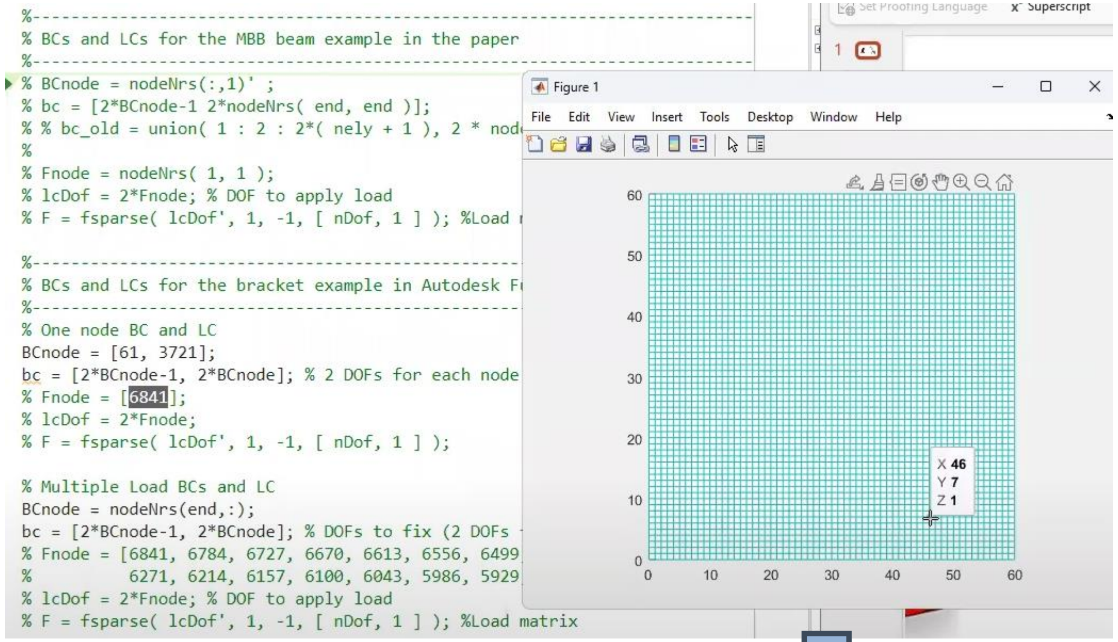


HOMEWORK ASSIGNMENT FOR TOPOLOGY OPTIMIZATION

- Phone stand design

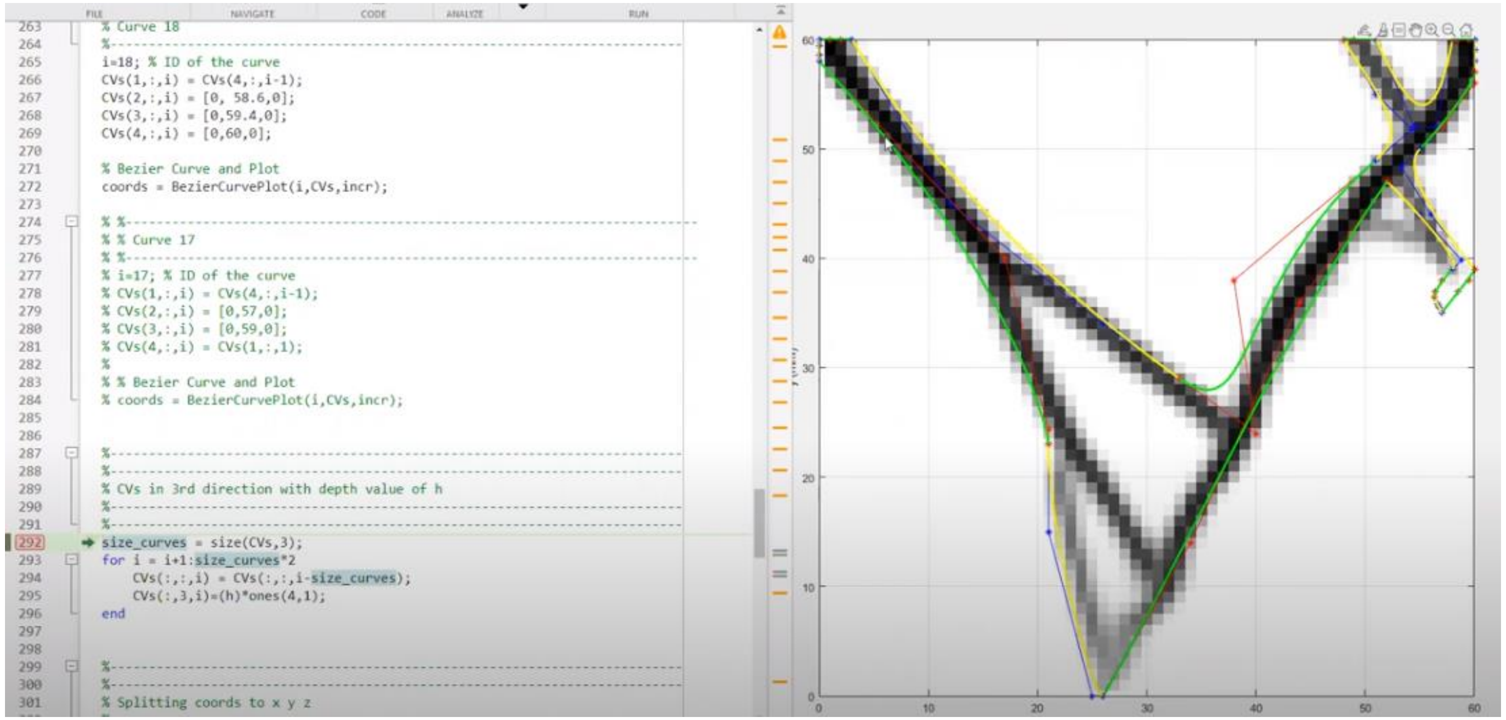


Initial domain discretization and loadings in MATLAB

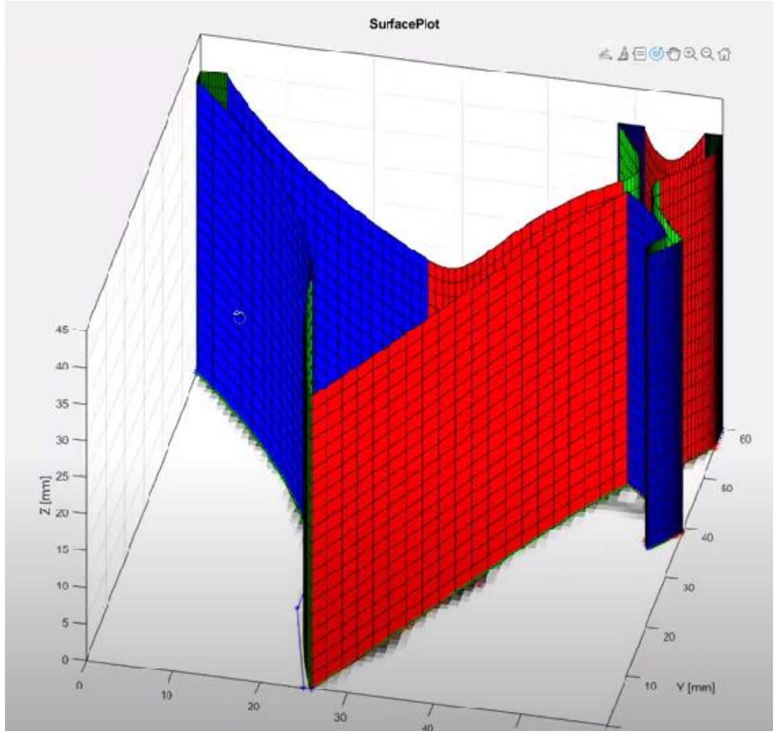


BEZIER CURVE AND SURFACE MODELING OF OPTIMIZED DESIGN

Bezier curve MATLAB function is used with continuities to define the geometry boundaries using the topology optimization image

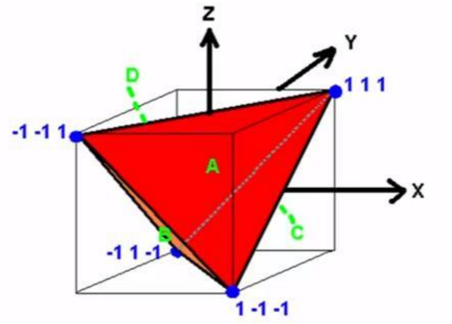


The curves are converted to loftes Bezier surfaces in MATLAB to have a 3D model:



CAD to Additive Manufacturing

STL format and generation in MATLAB



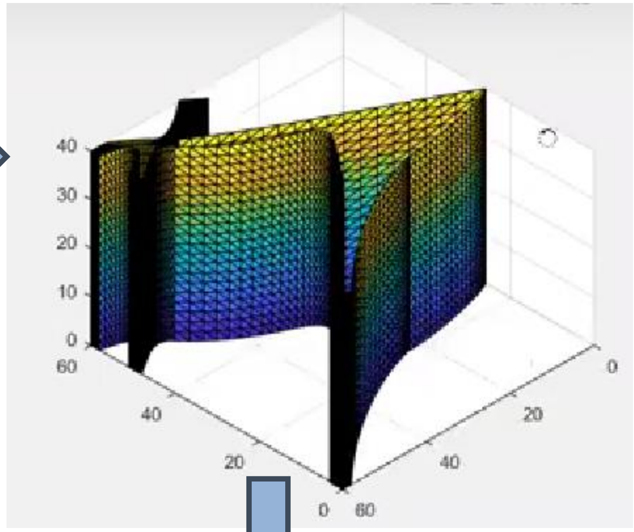

```
solid tetrahedron
 facet normal 0 0 0
  outer loop
   vertex 1 1 1
   vertex 1 -1 -1
   vertex -1 1 -1
  endloop
 endfacet
 facet normal 0 0 0
  outer loop
   vertex 1 1 1
   vertex -1 1 -1
   vertex -1 -1 1
  endloop
 endfacet
 facet normal 0 0 0
  outer loop
   vertex 1 1 1
   vertex -1 -1 1
   vertex 1 -1 -1
  endloop
 endfacet
 facet normal 0 0 0
  outer loop
   vertex -1 1 -1
   vertex 1 -1 -1
   vertex -1 -1 1
  endloop
 endfacet
endsolid tetrahedron
```

MathWorks

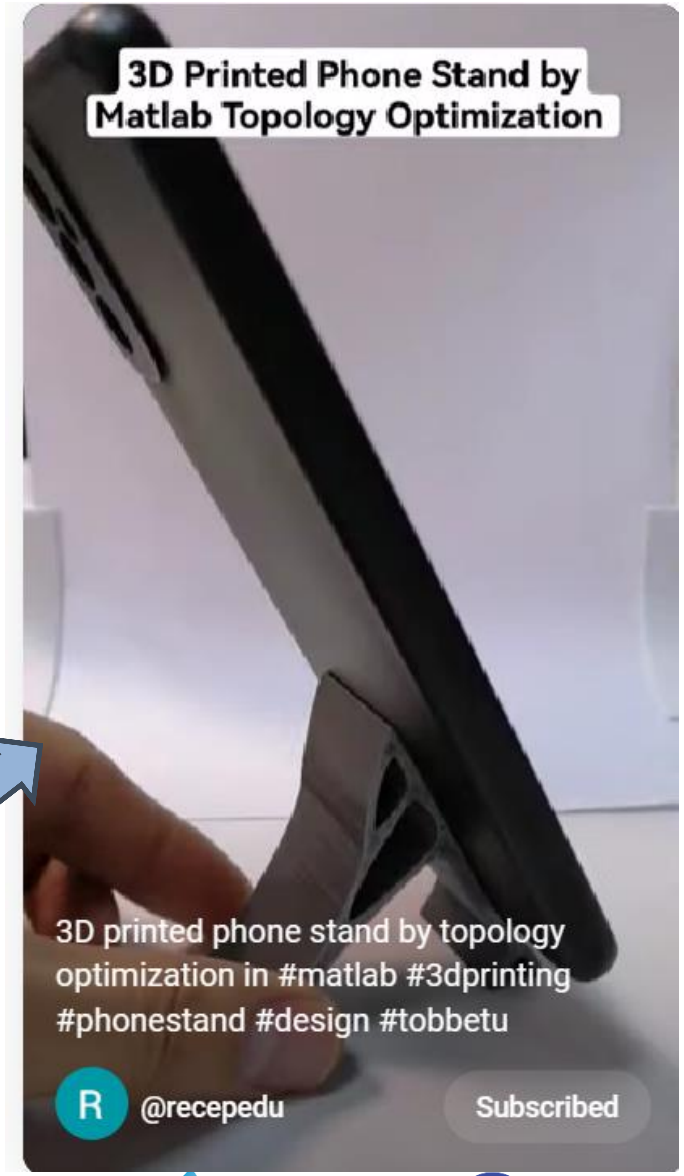
PHONE STAND EXAMPLE FOR STL GENERATION AND 3D PRINTING

MATLAB code for STL Generation from Bezier Surface

```
%Write each surface to STL
str1 = 'Surface';
str2 = '.stl';
for i = 1:size(CVx,3)
    str3 = num2str(i);
    surf2stl(strcat(str1,str3,str2),Px(:,:,i) ,Py(:,:,i), Pz(:,:,i));
end

%Read All STLs
for i = 1:size(CVx,3)
    str3 = num2str(i);
    fv(:,:,i) = stlread(strcat(str1,str3,str2));
end

%Merge All STLs
fvmerged = stlmerge(fv(:,:,1),fv(:,:,2));
for i = 3:size(CVx,3)
    str3 = num2str(i);
    fvmerged = stlmerge(fvmerged,fv(:,:,i));
end

%Write the final STL file
stlwrite('Model_Main.stl',fvmerged)

%Plot the STL model
data = stlread('Model_Main.stl');
stlPlot(data.vertices, data.faces, 'sonuc')
h=plotmesh(data.vertices,data.faces);

%Delete surface patch STL files
for i = 1:size(CVx,3)
    str3 = num2str(i);
    namestl = strcat(str1,str3,str2);
    delete (namestl)
end
```



Using a free 3D printing software (Prusa Slicer) to create g-code for 3D printing from the STL file.



FIGES

FINAL PROJECT FLOW CHART

3D Topology Optimization in MATLAB

+

3D Parametric Surface Modeling in MATLAB

+

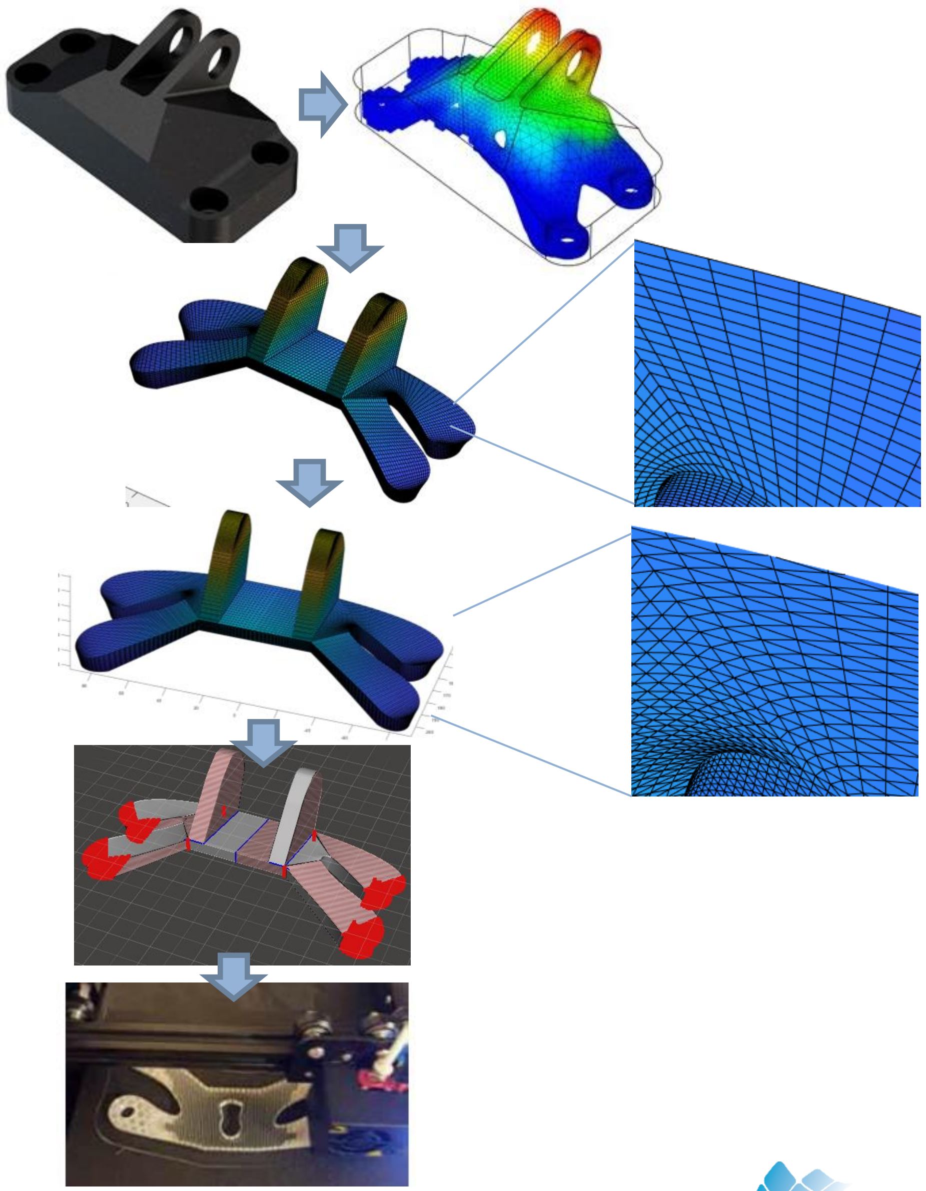
STL Model Conversion in MATLAB

+

STL Repair in 3rd party software (if necessary)

+

Slicing and 3D printing

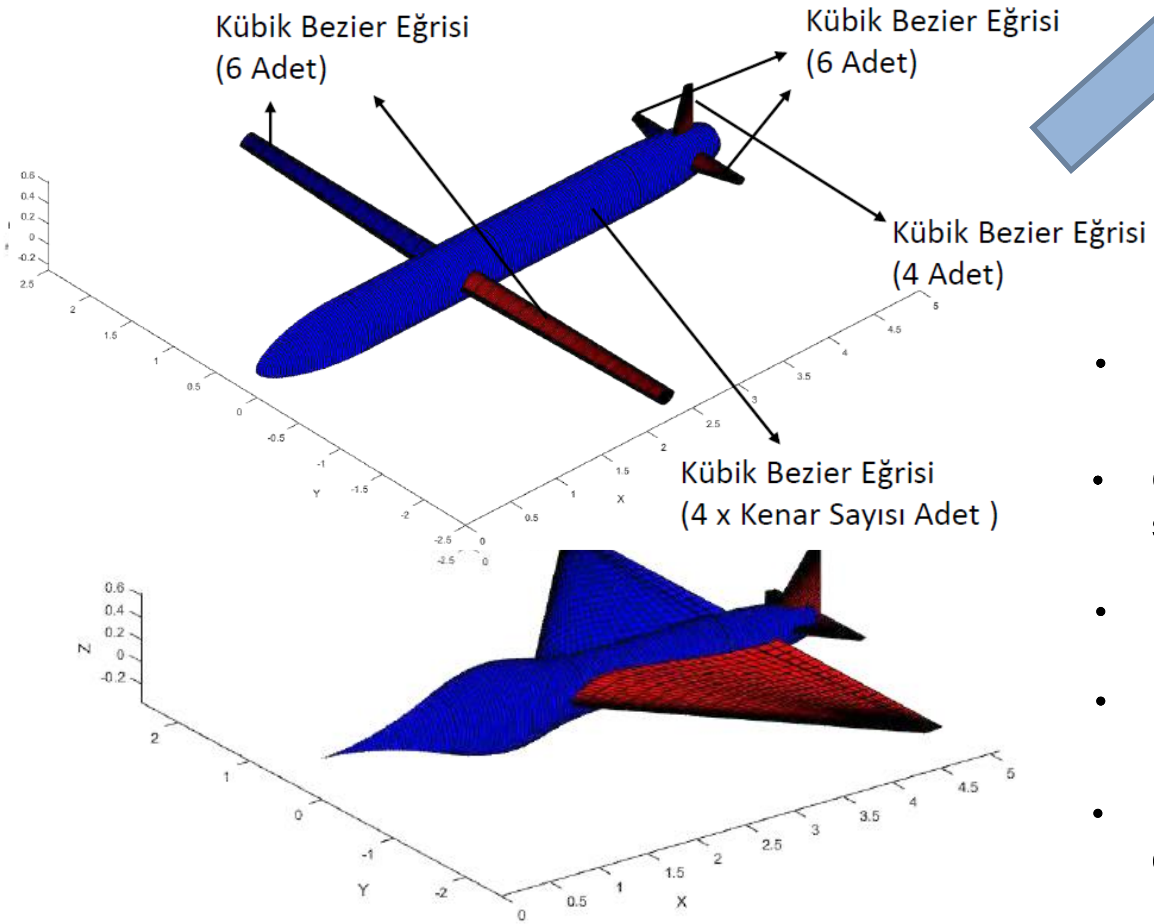


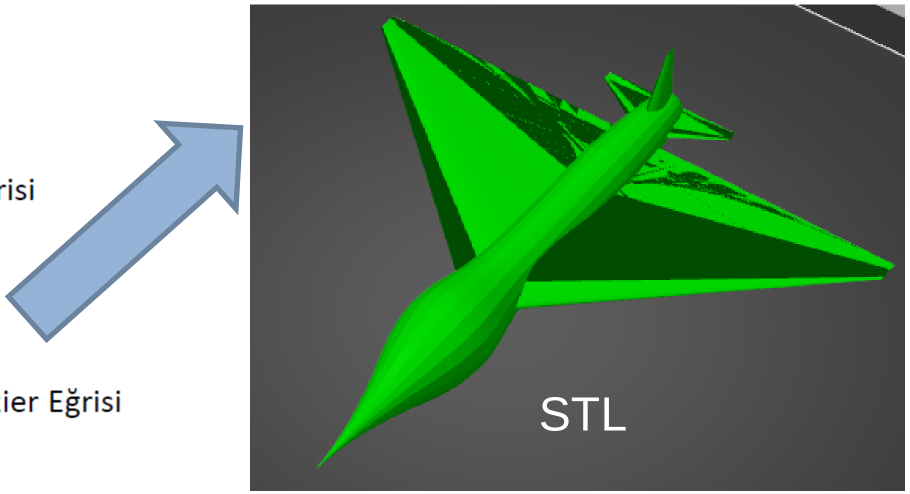
Benefits / added value of using MATLAB and Simulink

- Students learn CAD Fundamentals by applying.
  - Primitive Instancing
  - Curve and Surface Modeling
  - Topology Optimization
  - CAD to Additive Manufacturing

- New project / design / coding ideas are developed in different disciplines (e.g. Aerospace, biomedical, consumables, etc).
- Customized CAD programs are developed in MATLAB.
- Some examples are given in the following slides.

Example 1: Unmanned Aerial Vehicle Design Tool in MATLAB

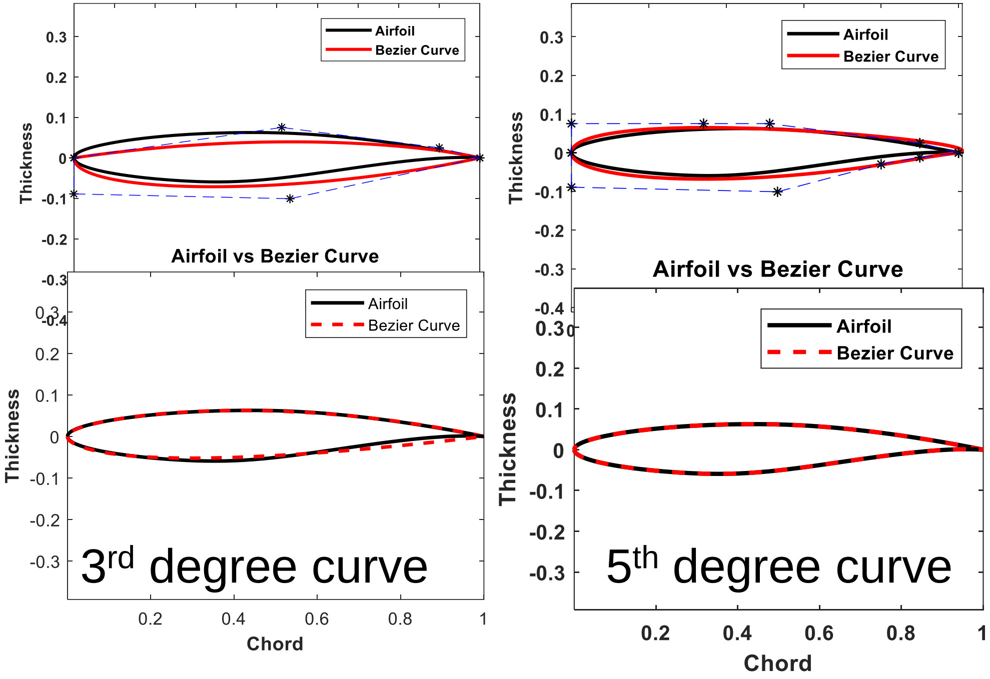


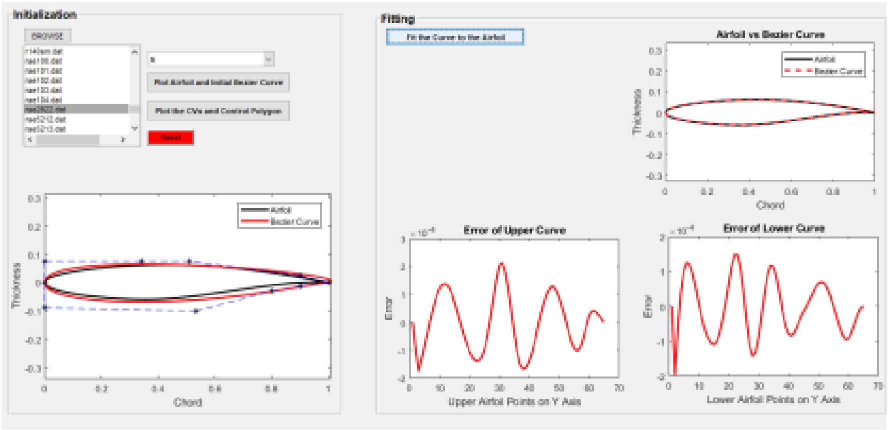


- Converting each surface by “surf2stl.m” to an STL file.
- Open each STL in .txt format and delete the final rows starting with “endsolid”.
- Add all texts in all these STL files together.
- Add a row of “endsolid” to the end of the text and save as STL extension.
- Repair the STL file in a free software such as Netfabb or meshmixer if necessary.

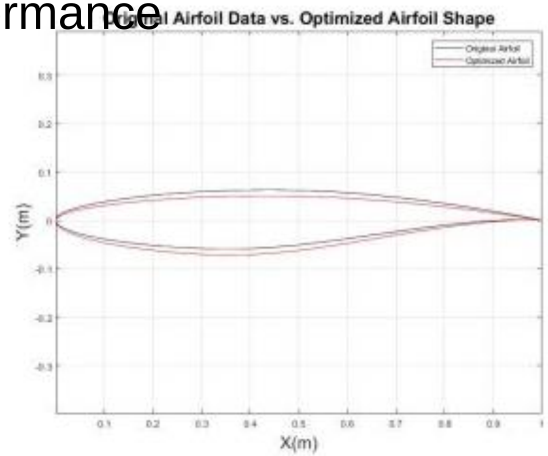
Example 2: Airfoil curve/surface fit

Parametric Curve Fitting to Airfoil Geometry in MATLAB

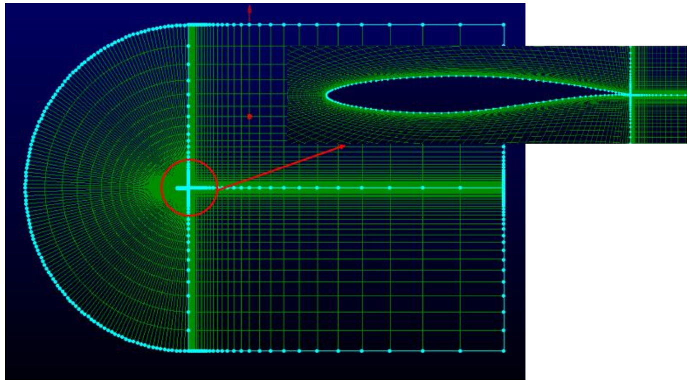




Optimal design for best performance



Mesh generation for flow analysis.



Example 3: Wing Design with Lattice Core

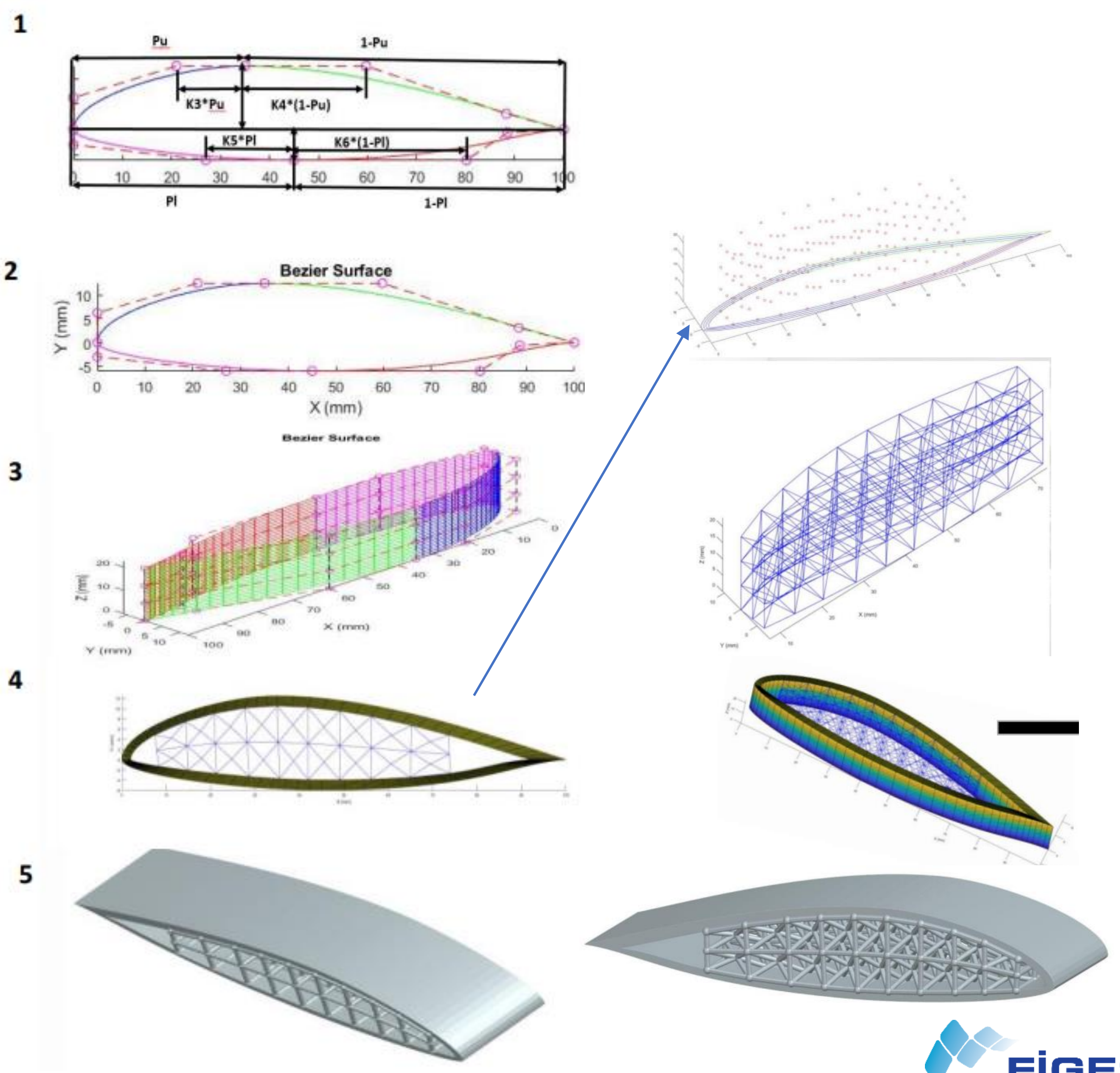
1. Identify wing profile parameters

2. Parametric modeling of airfoil using Bezier curves

3. Parametric modeling of wing using Bezier surfaces

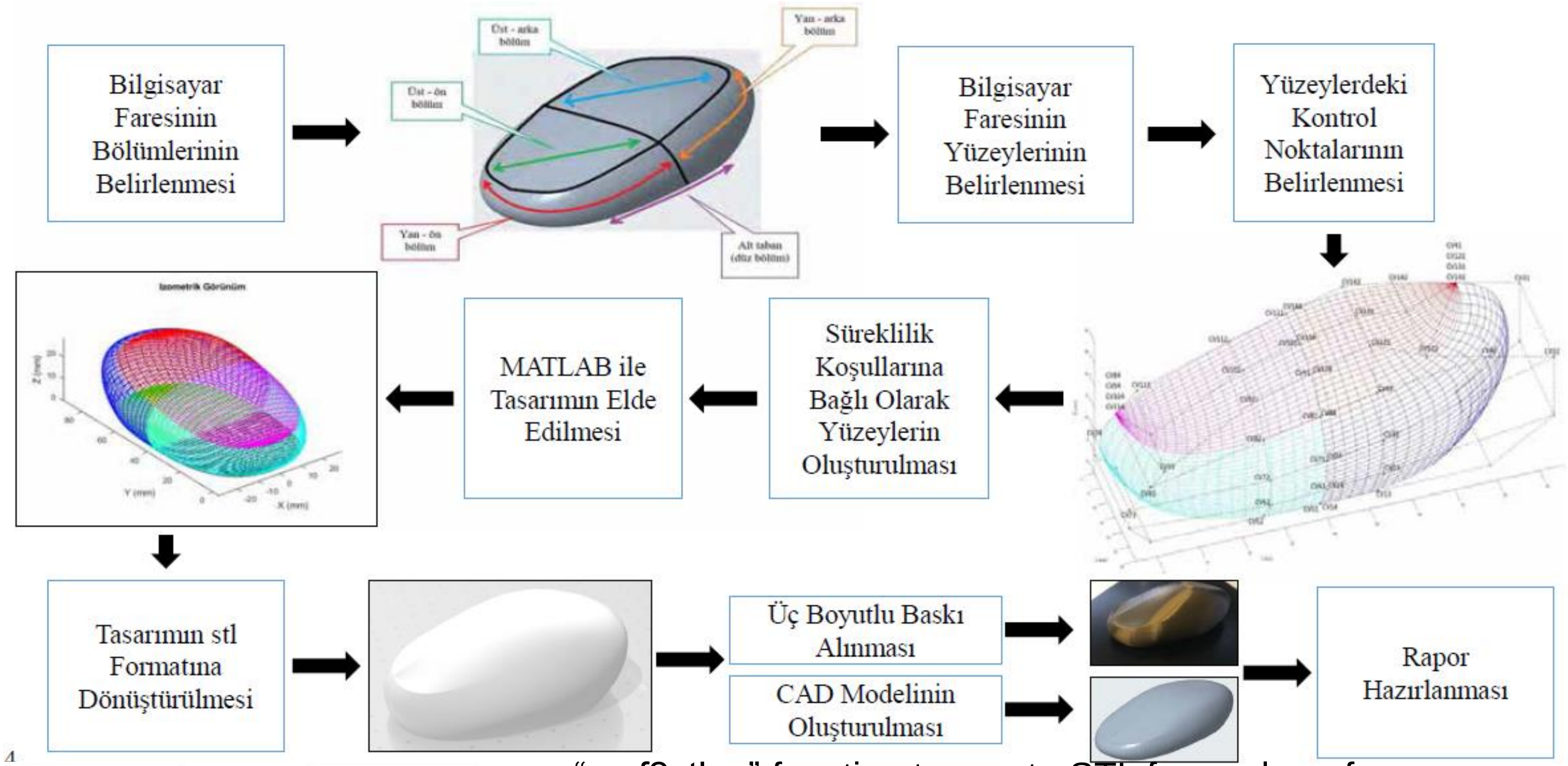
4. Placing lattice structure conformal to wing surface

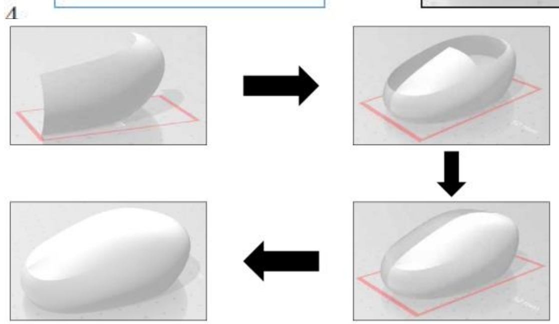
5. STL model generation of the wing model.



FIGES

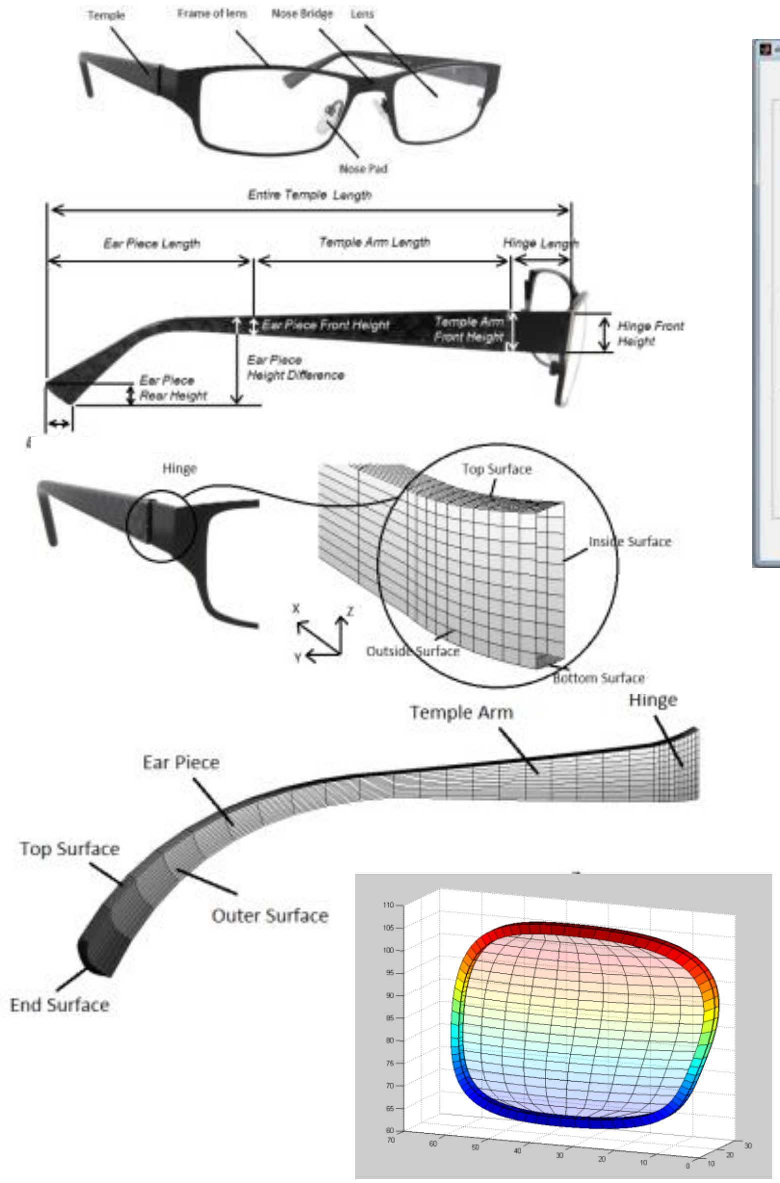
Example 4: Parametric Computer Mouse Design



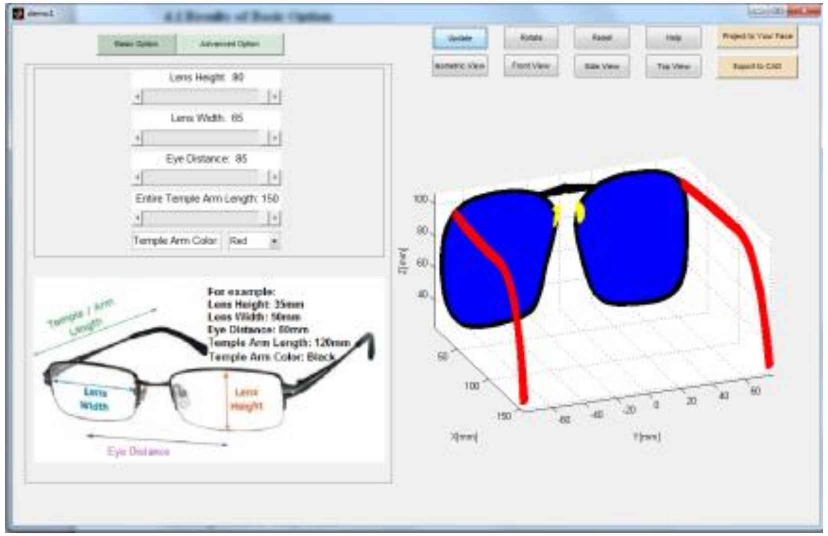


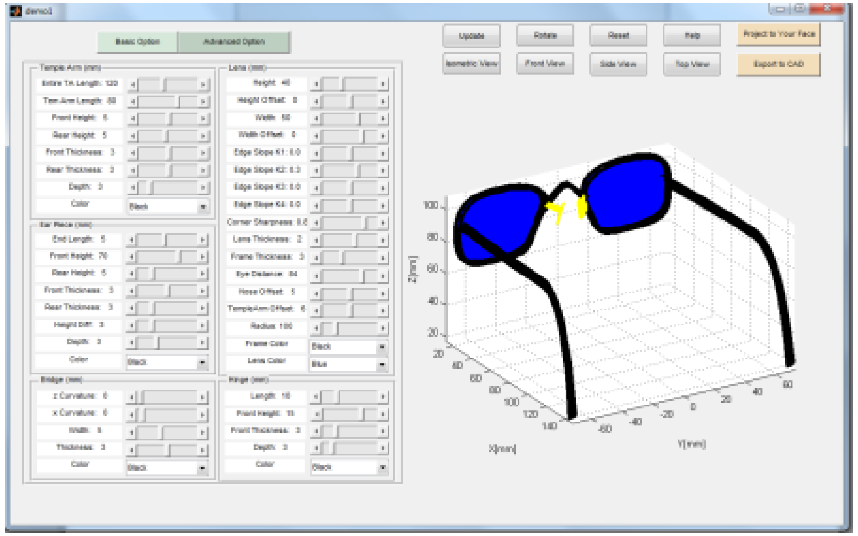
- “surf2stl.m” function to create STL for each surface.
- “stlread.m” to read each STL file.
- “stlcombine.m” to combine each STL file read by MATLAB.
- “stlwrite.m” to write the combined STL file.
- Repair the STL file in Netfabb if necessary. Save as an STL file
- Slice the STL model in PrusaSlicer and generate the g-code for 3D printing.
- 3D print the model using the g-code in a 3D printer.

Example 5: Customized Eye Glass Design Tool

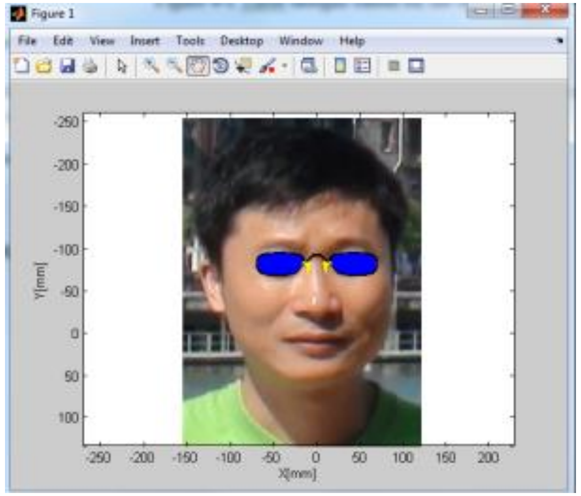


GUI in MATLAB for Parametric Eye Glass Modeling

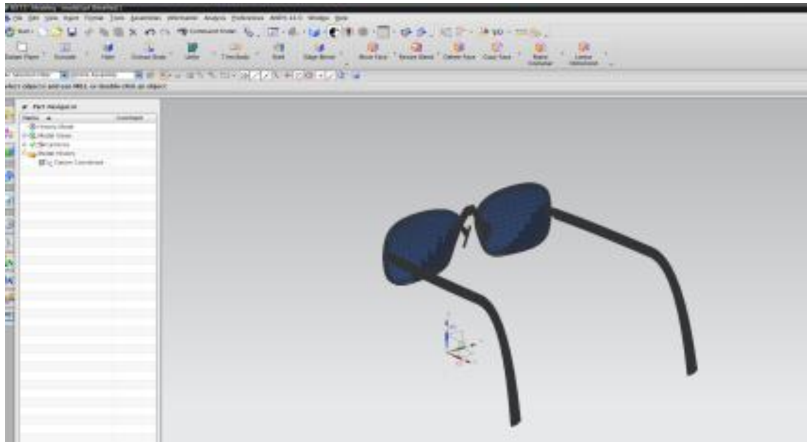




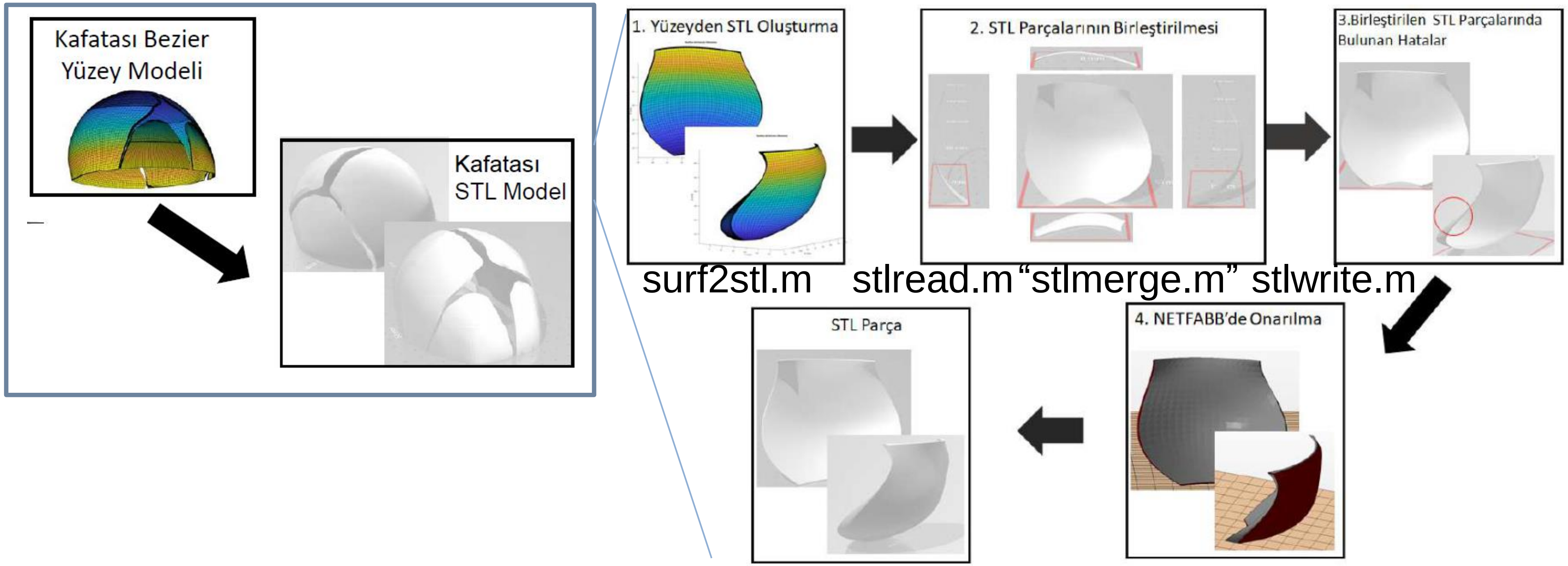
Projection to your face



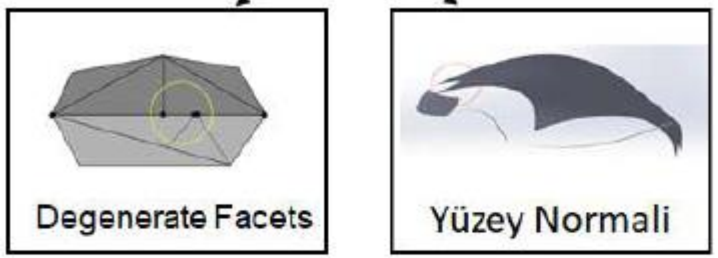
Export to CAD system



Example 6: Skull Modeling with Parametric Surfaces



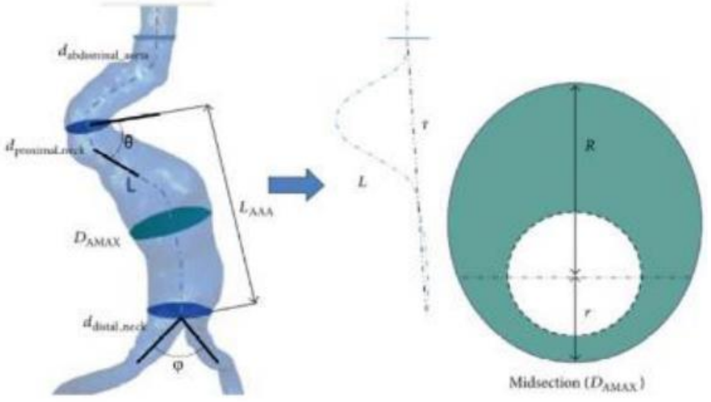
STL Issues to be fixed in



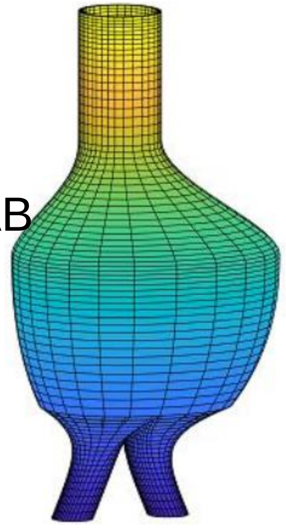
- “surf2stl.m” function to create STL for each surface.
- “stlread.m” to read each STL file.
- “stlcombine.m” to combine each STL file read by MATLAB.
- “stlwrite.m” to write the combined STL file.
- Repair the STL file in Netfabb if necessary. Save as an STL file
- Slice the STL model in PrusaSlicer and generate the g-code for 3D printing.
- 3D print the model using the g-code in a 3D printer.

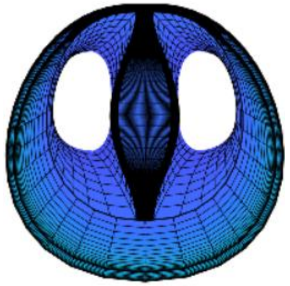
Example 7: Abdominal Aort Model

Parameterization of the geoemtry

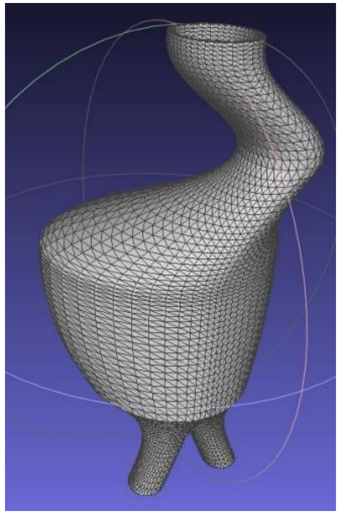


Bezier
Surface
Modeling
in MATLAB





STL
Model
Generatio
n
In
MATLAB

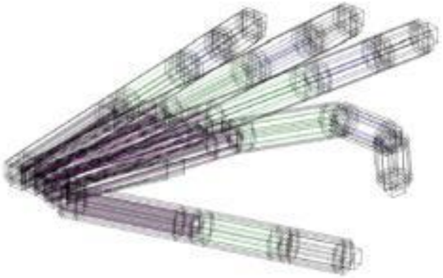


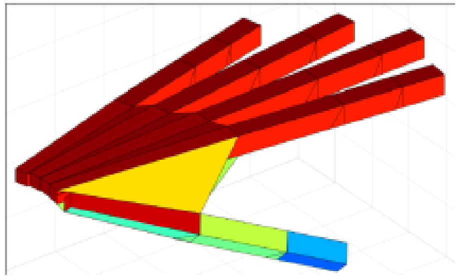
Actual Model

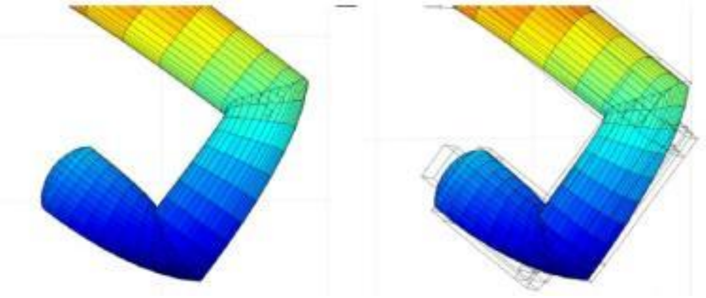
Surface Model

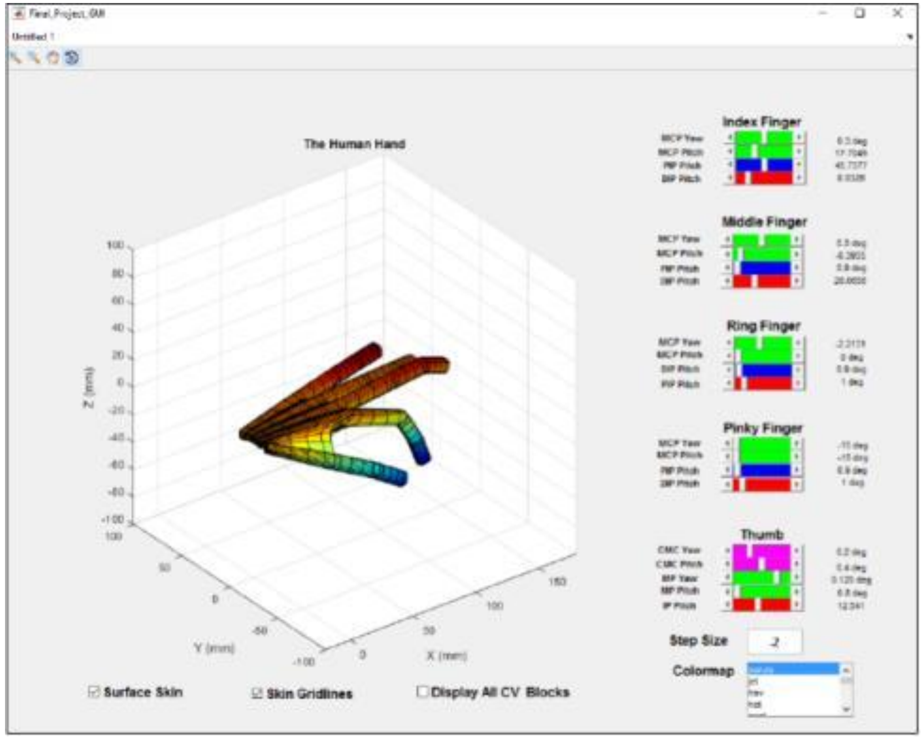
Example 8: Hand Design for Robotics

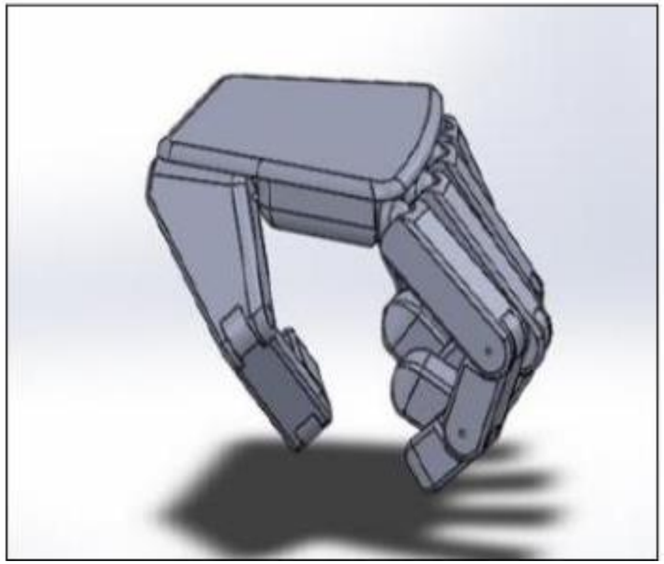
Parametric Robotic Hand Modeling in MATLAB





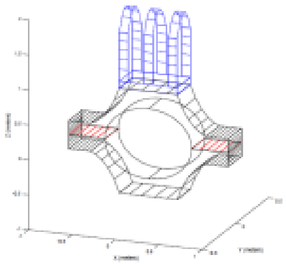




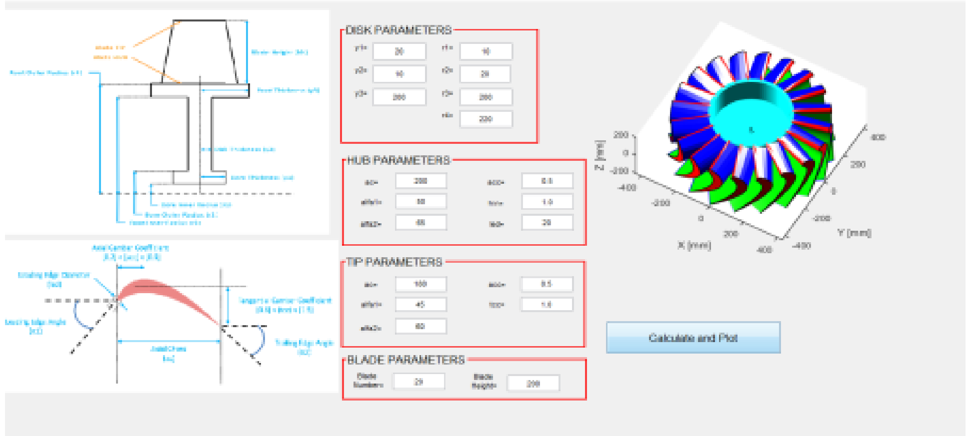


Other Examples

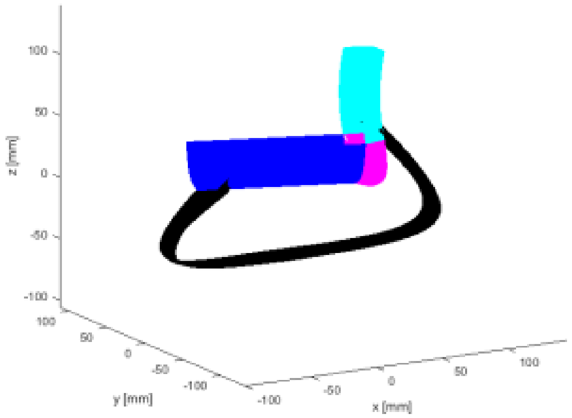
Internal Gear Design Optimization In MATLAB



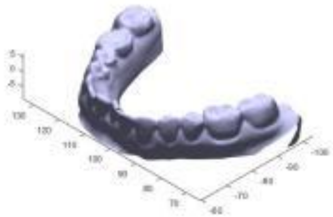
Planetary Gear Design in MATLAB

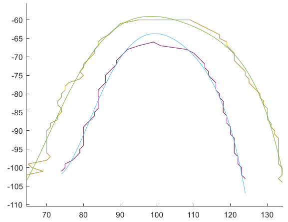


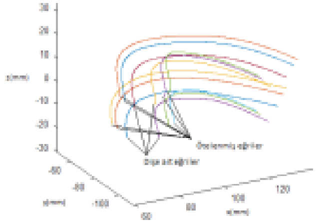
Leg Prosthesis Modeling for Animals

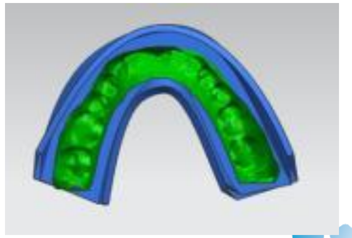


Parametric Mouthguard Modeling









Future plans

- CAD is an always developing area.
- Hence, new modeling and design optimization strategies can be added in the future.
- The course can also be opened in the graduate level.
- MATLAB Grader will also be integrated for grading the MATLAB assignments.

Thank you

Q&A – 5min

Assoc. Prof. Recep Muhammet Görgülüarslan
TOBB University of Economics and Technology
rgorguluarslan@etu.edu.tr

Youtube Channel for the Course Videos: **RecepEdu**
(Please subscribe and follow ☺)

https://www.youtube.com/@recepedu