

ENG 100 Introduction to Engineering

CAD-LAB-2: Surface Design – Rhino Automation –
Solid Modeling (CSG & B-Rep)

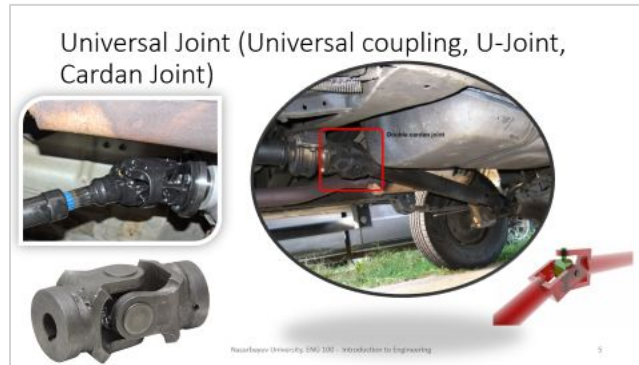
Fall semester, Academic Year 2018-19

Course Instructor:

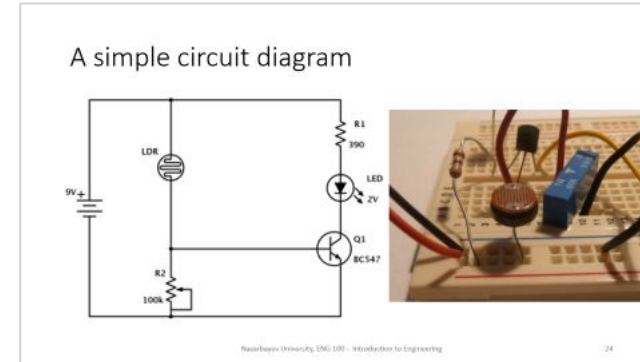
Konstantinos Kostas (konstantinos.Kostas@nu.edu.kz)

Contents

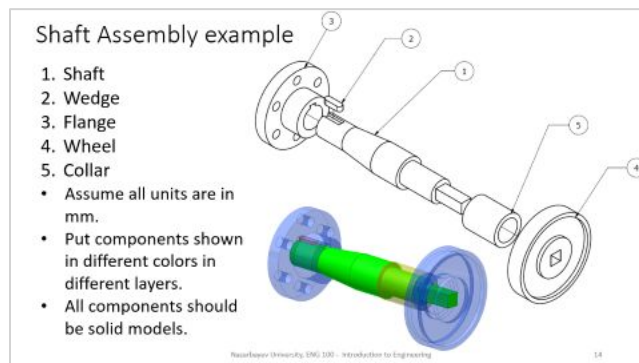
1. CSG and B-Rep Solid Modeling



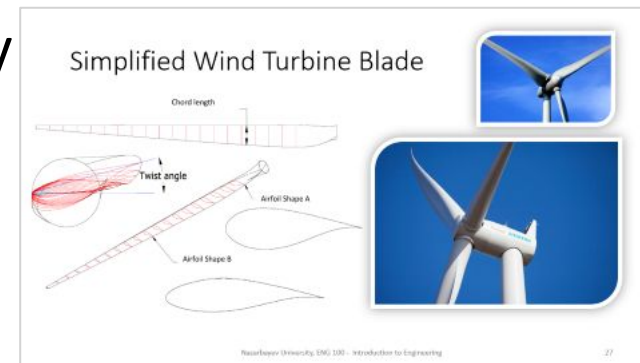
2. Universal Joint



4. Circuit Diagram



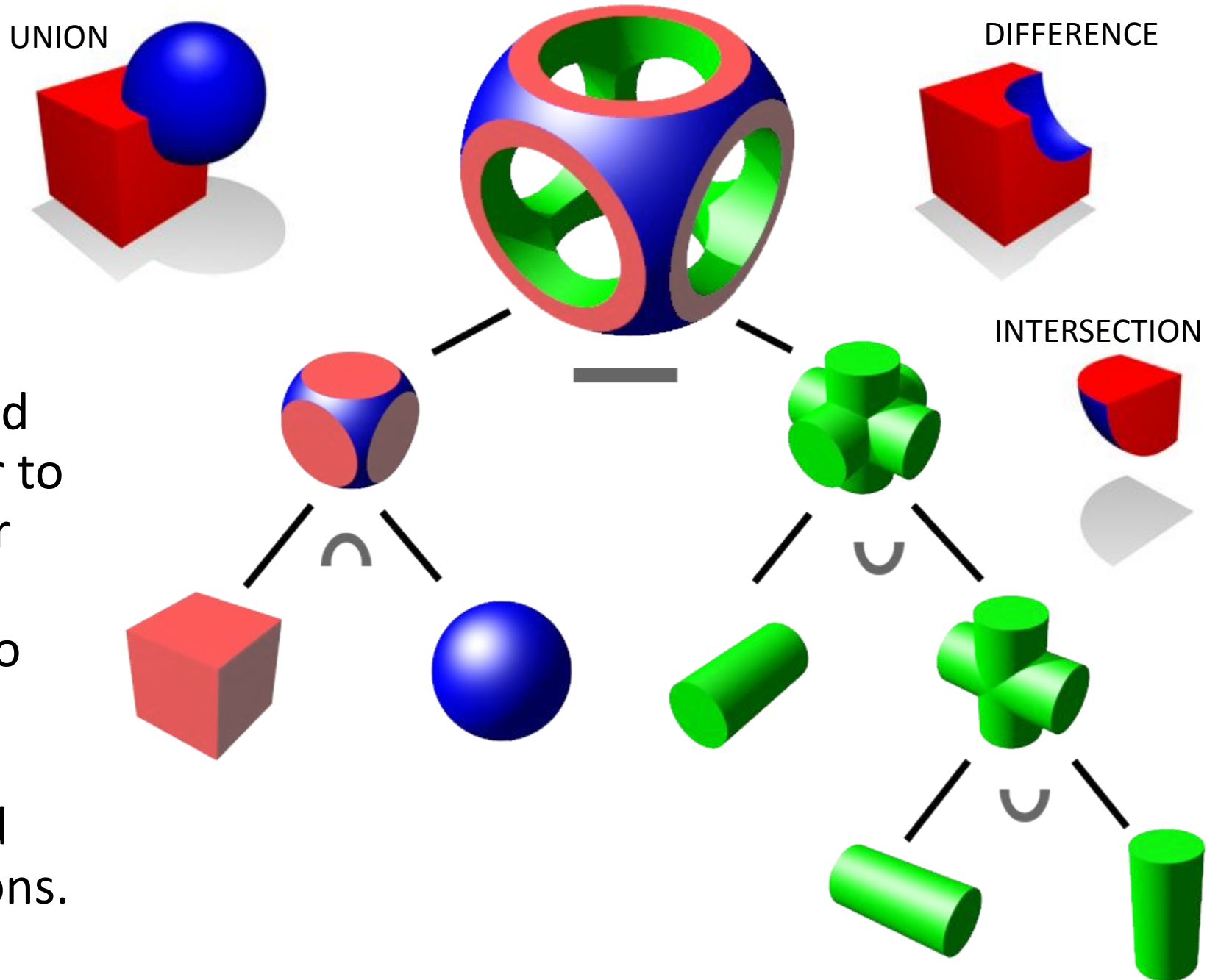
3. Shaft Assembly



5. Wind turbine Blade

CSG

Constructive solid geometry (CSG) is an approach followed in solid modeling. Constructive solid geometry allows a modeler to create a complex surface or object by using Boolean operations to combine simpler objects (solid primitives) and generate complex solid models by such combinations.

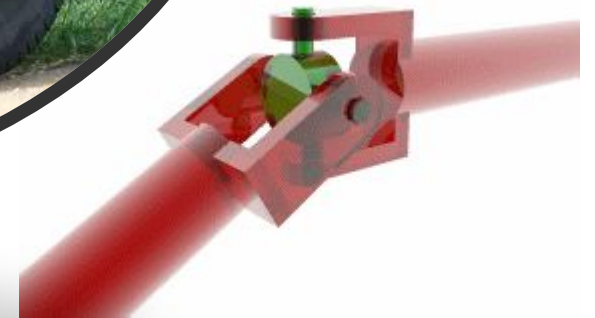
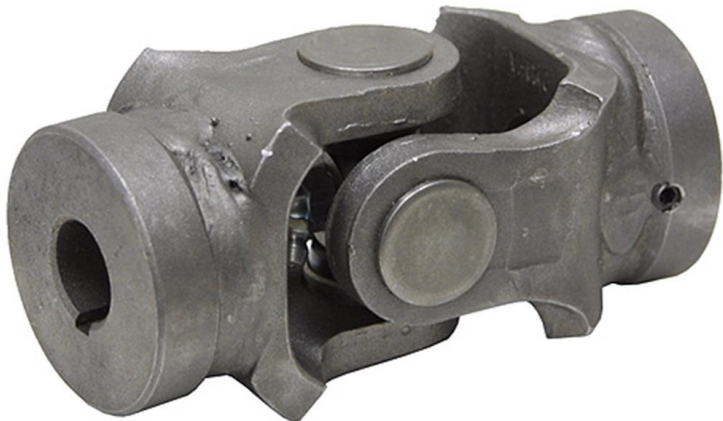
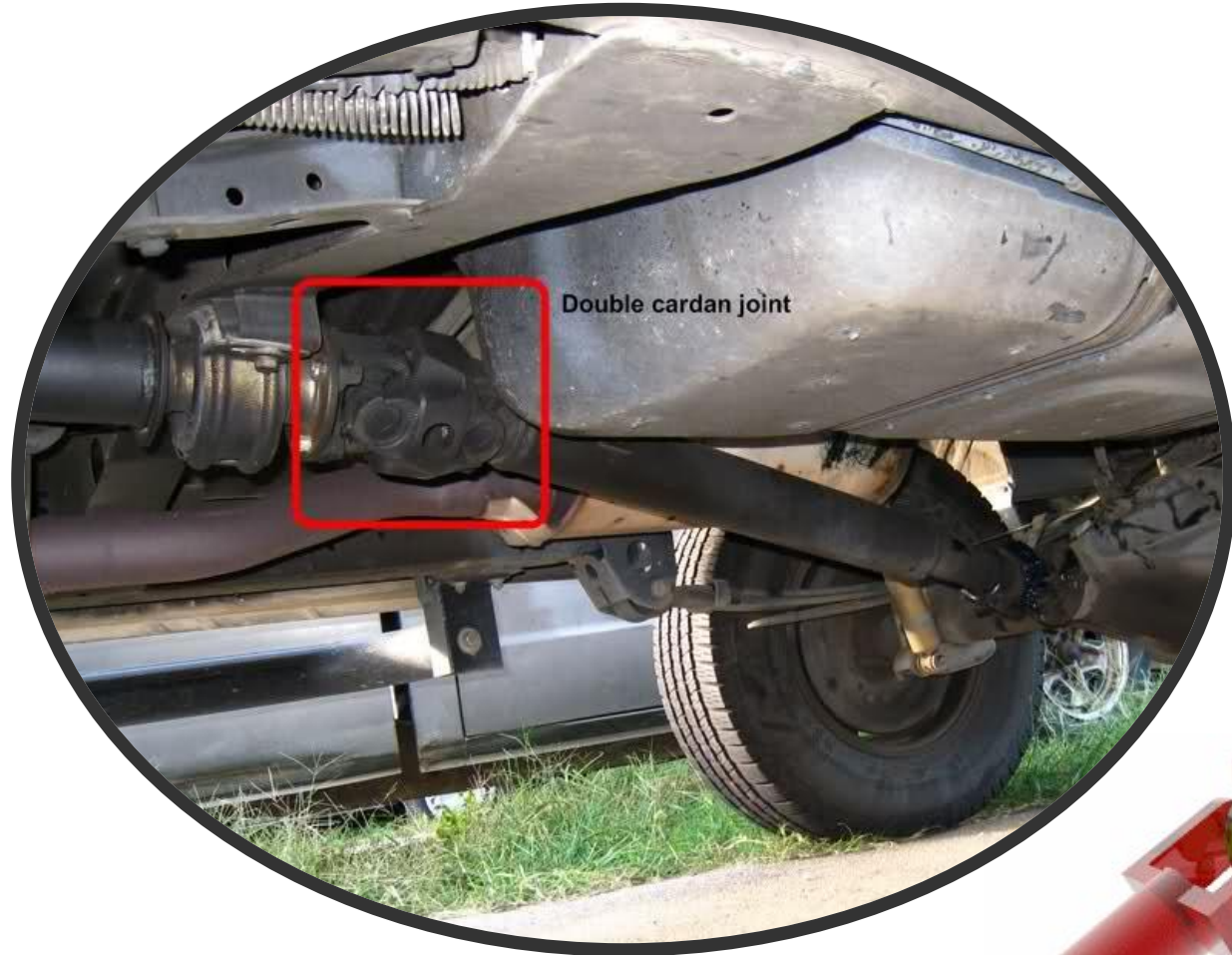


B-rep

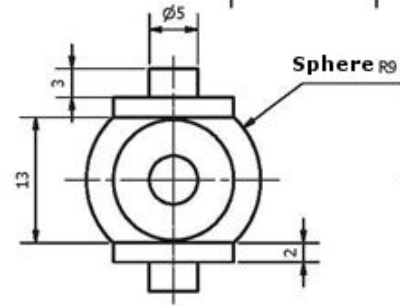
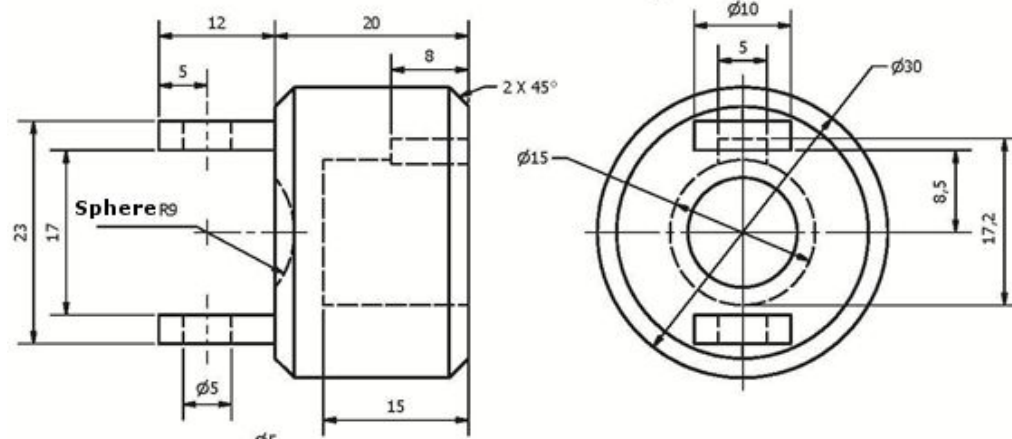
The second and most popular approach in solid modeling and computer-aided design, is the **boundary representation (B-rep)** approach for modeling solid objects. A solid is represented as a collection of connected surface elements, the boundary between solid and non-solid.

Boundary representation is more flexible and has a much richer operation set than CSG. In addition to the Boolean operations, B-rep has extrusion (or sweeping), chamfer, blending, drafting, shelling, tweaking and many other operations as we will explore later.

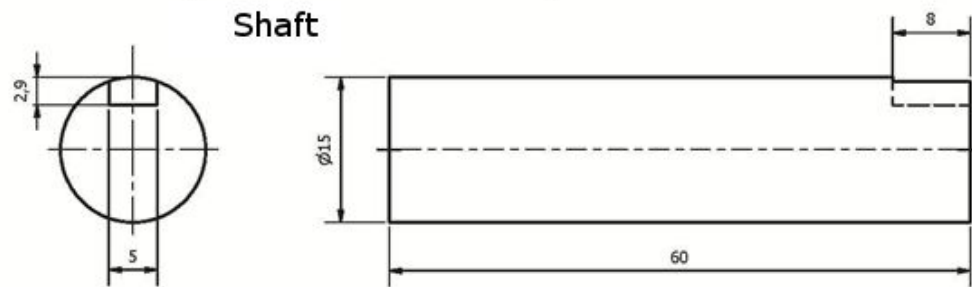
Universal Joint (Universal coupling, U-Joint, Cardan Joint)



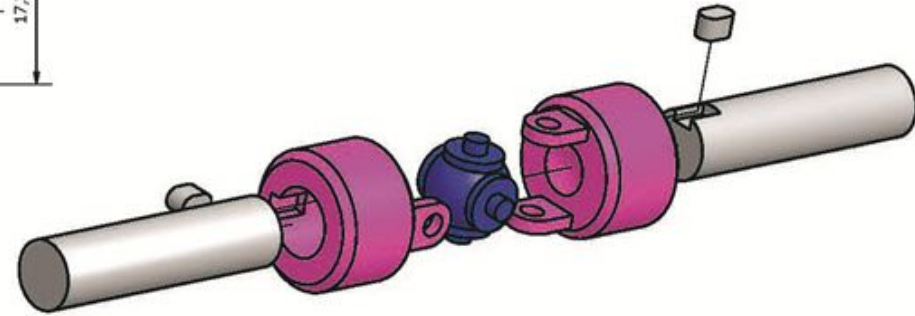
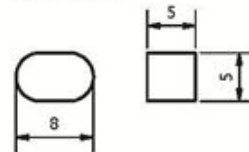
CARDAN joint



Shaft



Wedge

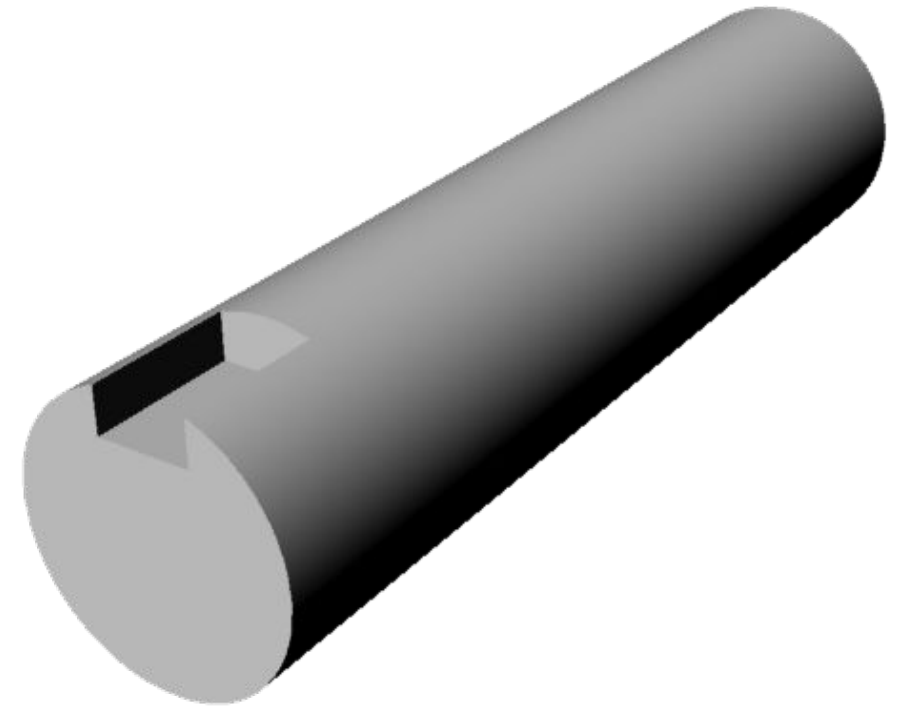


3d modeling of simple U-joint

- Description: Create the 3d model of the U-joint shown in previous slide.
 - Assume all units are in mm.
 - Put components shown in different colors in different layers.
 - Create two copies of the model: one assembled and one disassembled (as shown in previous slide).
 - All components should be solid models.
 - Assuming that all components will be made out of steel with a relative density of 7.7, what will be the weight of the complete U-joint when manufactured?

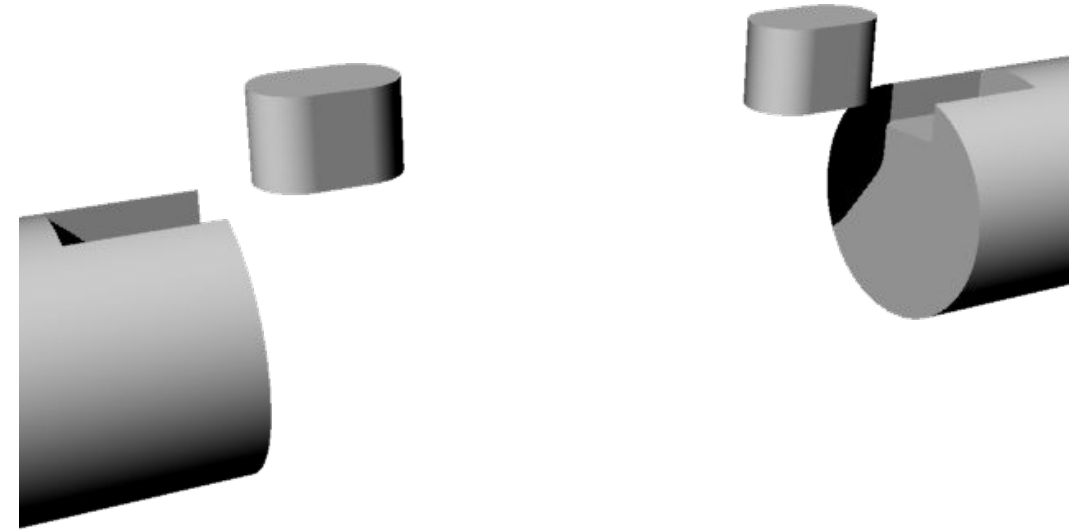
Modeling steps: Step 1

- Launch Rhino using mm as units and a tolerance of 1 micron (=0.001mm)
- Create three layers, named “shaft”, “sphere” and “joint” and use gray, blue and red colors, respectively
- Make “shaft” layer the current layer and create the shaft in the front viewport
 - For example:
Create a cylinder with the appropriate dimensions and generate the wedge slot by Boolean difference. For the wedge slot you may use an appropriately size box solid



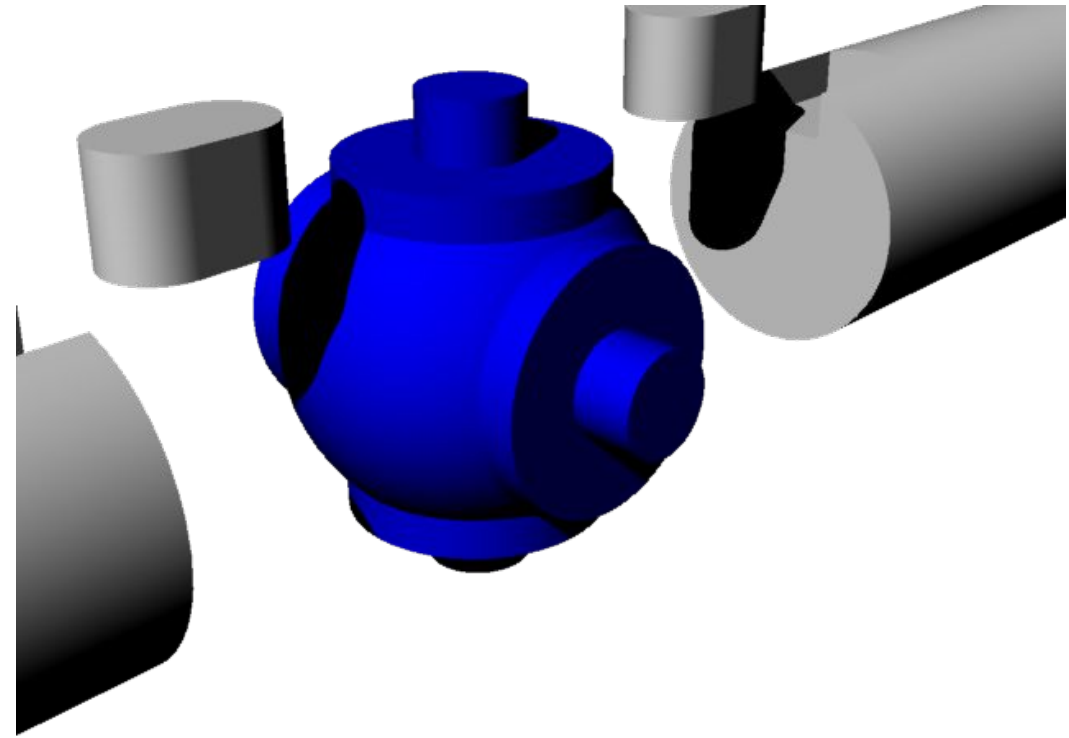
Modeling steps: Step 2

- Create the second shaft by mirroring the first one (remember to set Copy to Yes in mirror).
- Create the wedge by generating a box of appropriate dimensions and the use filletEdge to create the fillets (assume that fillet arcs meet at the corresponding side's mid-point).
- Create the second wedge by copying the first one.
- Position the components similar to the positions in the image shown above.



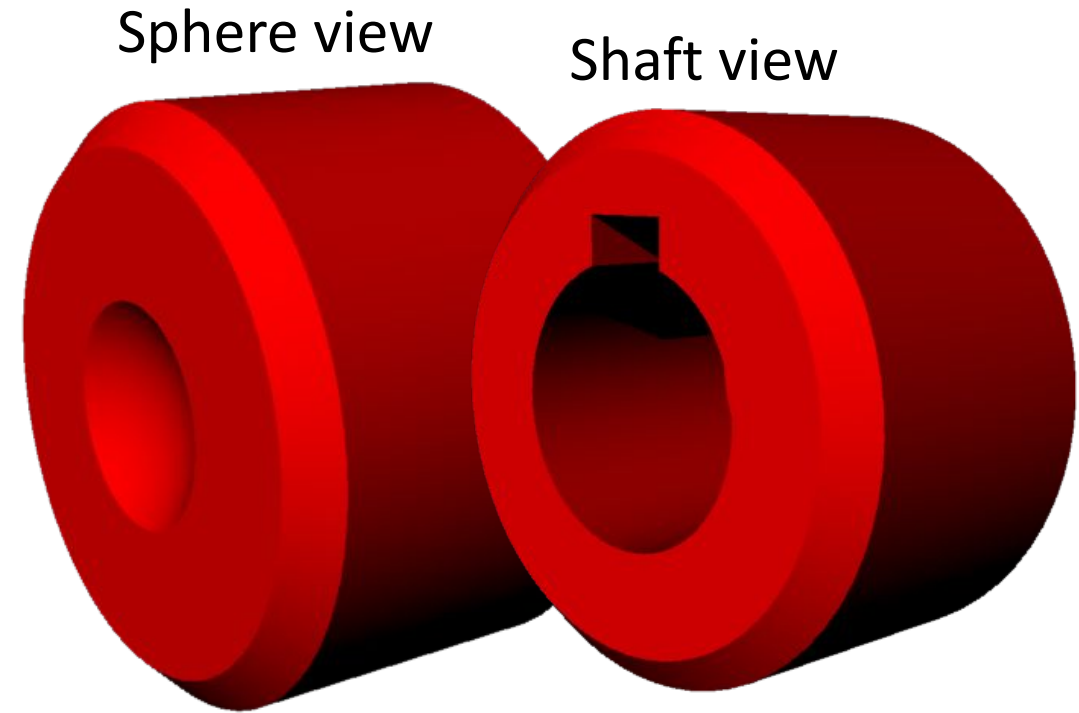
Modeling steps: Step 3

- Make “sphere” layer the current layer
- Create a sphere with radius 9 and cut the pieces that are not needed (you may use Boolean differences with boxes or trim the sphere with appropriately placed lines and then cap the resulting surface to get a solid)
- Create one of sphere’s joint pins using two stacked cylinder and
- Generate the remaining pins by appropriately mirroring/rotating the one you’ve create above
- Create a union of all sphere components to get a single solid



Modeling steps: Step 4

- Make “joint” layer the current layer
- Create a cylinder with diameter 30 and height 20 and use chamferEdge to create the chamfering for both bases of the cylinder.
- Use a cylinder (diameter 15) and BooleanDifference to create the cavity in the joint.
- Use a box and BooleanDifference to create the wedge’s slot in the interior part of the cavity.
- Use a sphere with radius 9, positioned appropriately to create the required cavity at the opposite face of the joint.
- The resulting joint should be a single solid object.



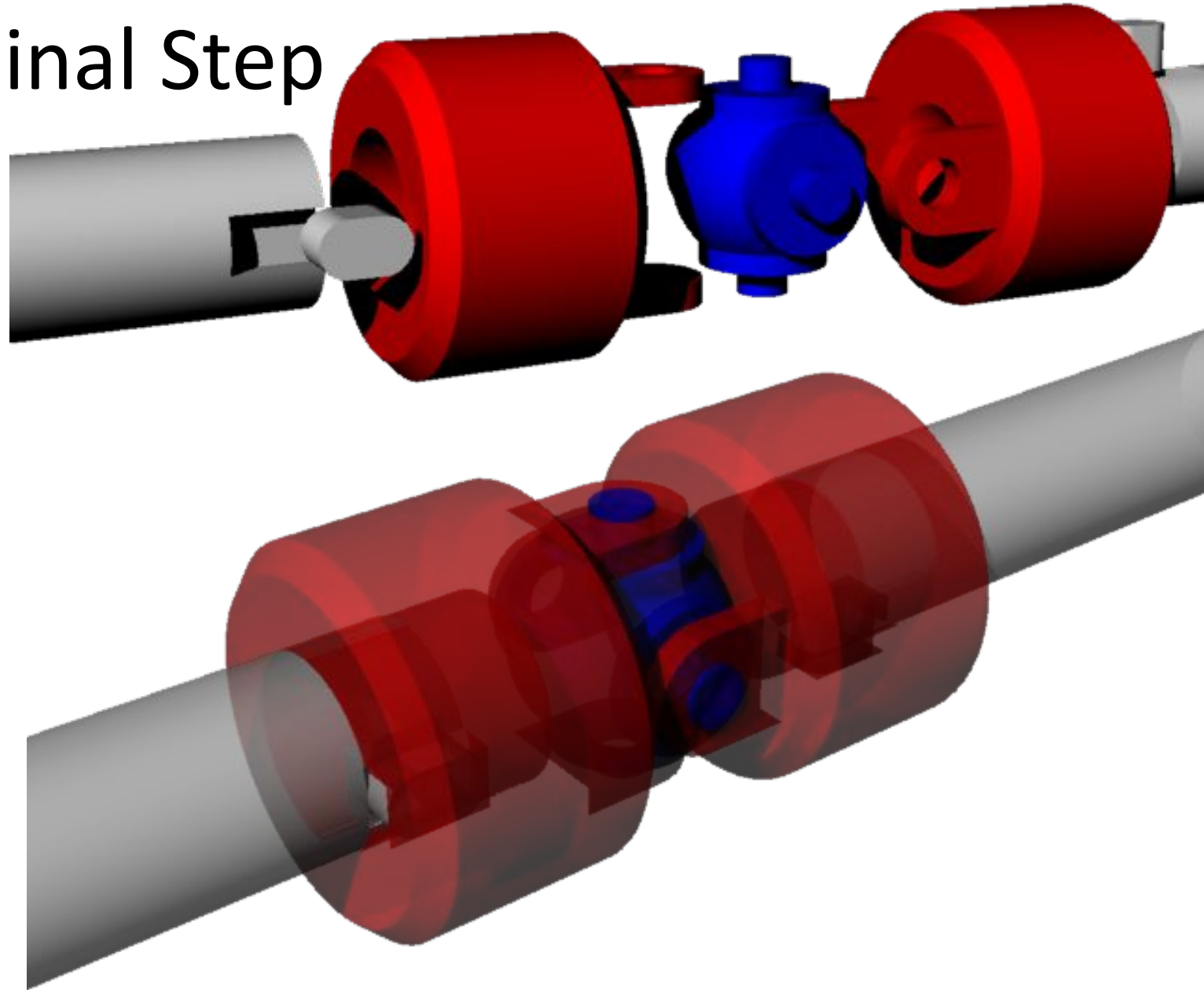
Modeling steps: Step 5

- On the sphere side of the joint, create the two holding brackets:
- Create the first one by sketching its border and a hole and then make it a solid by extruding at a distance of 5mm
- Create the second bracket by mirroring with respect to cylinder base's center
- Use Boolean union to create a single joint and mirror the result to get the second joint
- Rotate one set of shaft, wedge and joint by 90 degrees as shown in the initial sketch



Modeling steps: Final Step

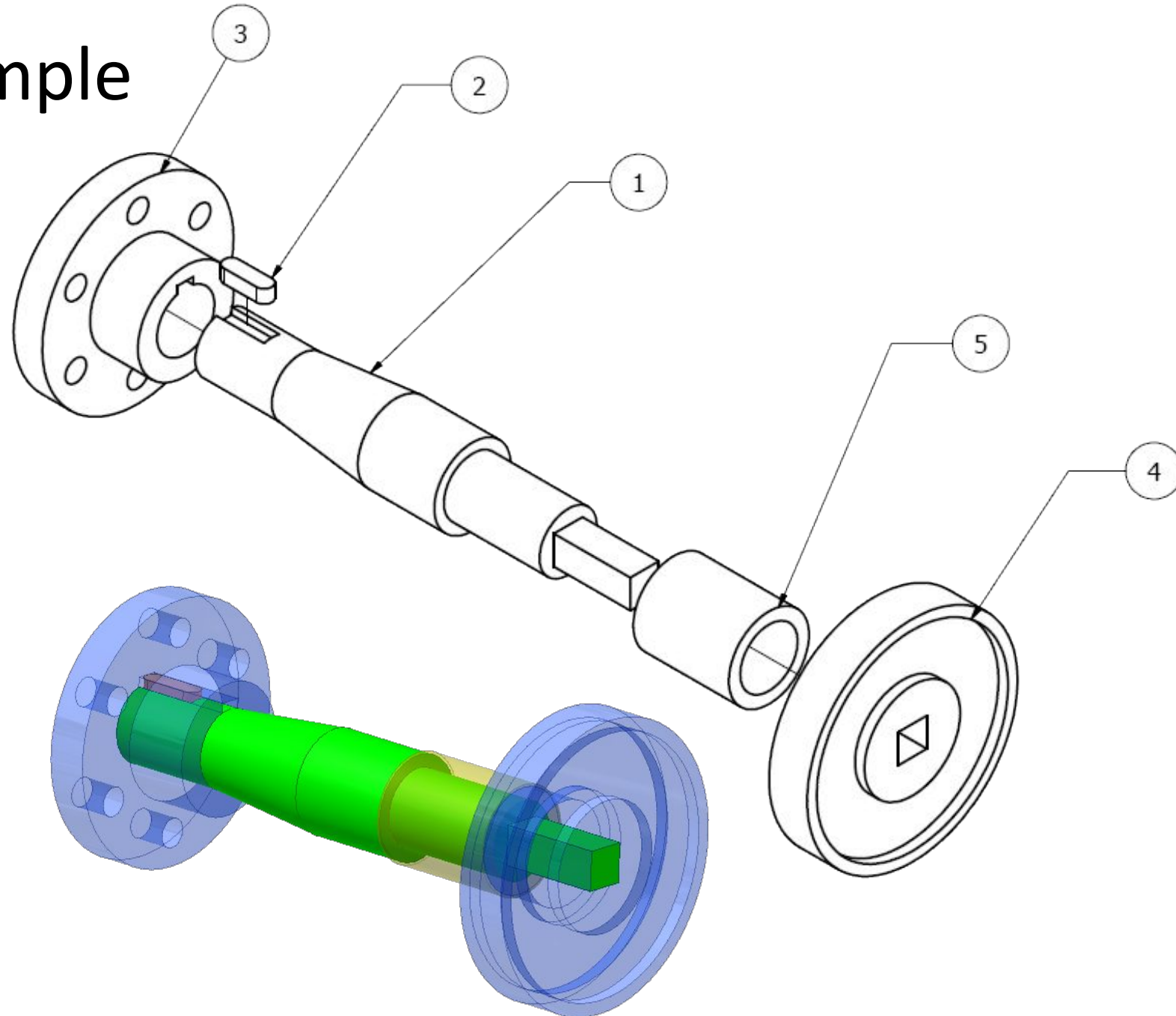
- The resulting model should look like the first picture besides
- Create a copy of all parts and move the copied parts to their final positions so that you can get the assembled model.
- Select all solids and type volume to get the overall volume. You should get a value of around 47705.23 mm^3 . Hence, the overall weight is approximately 367g

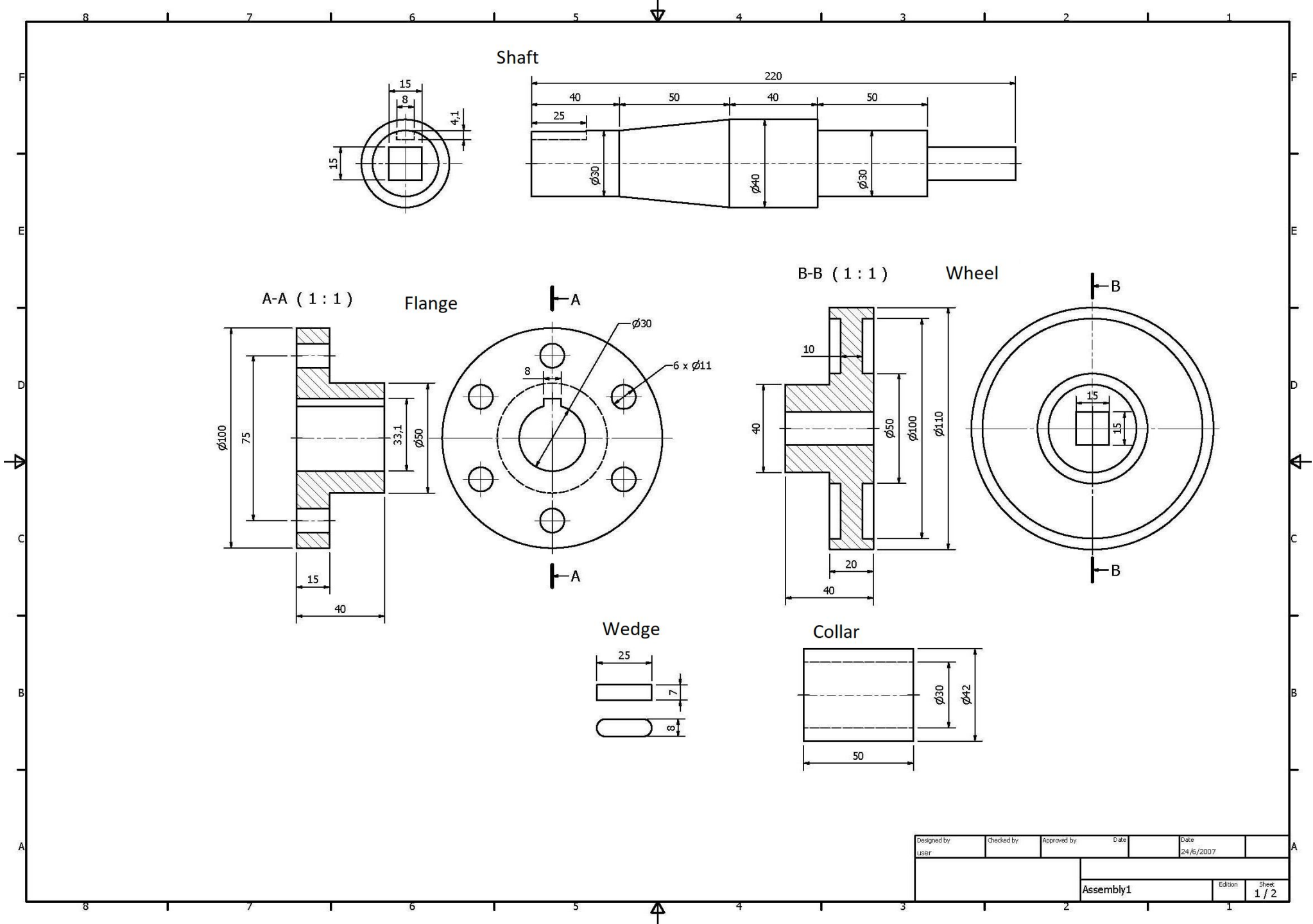


Shaft Assembly example

1. Shaft
2. Wedge
3. Flange
4. Wheel
5. Collar

- Assume all units are in mm.
- Put components shown in different colors in different layers.
- All components should be solid models.

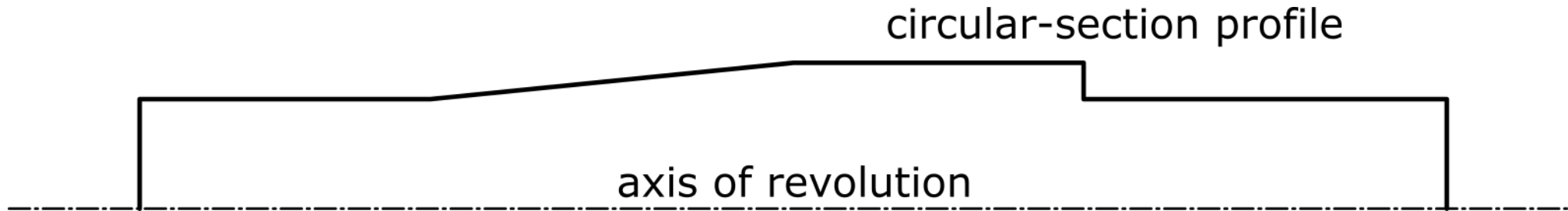




Modeling Steps: Step 1

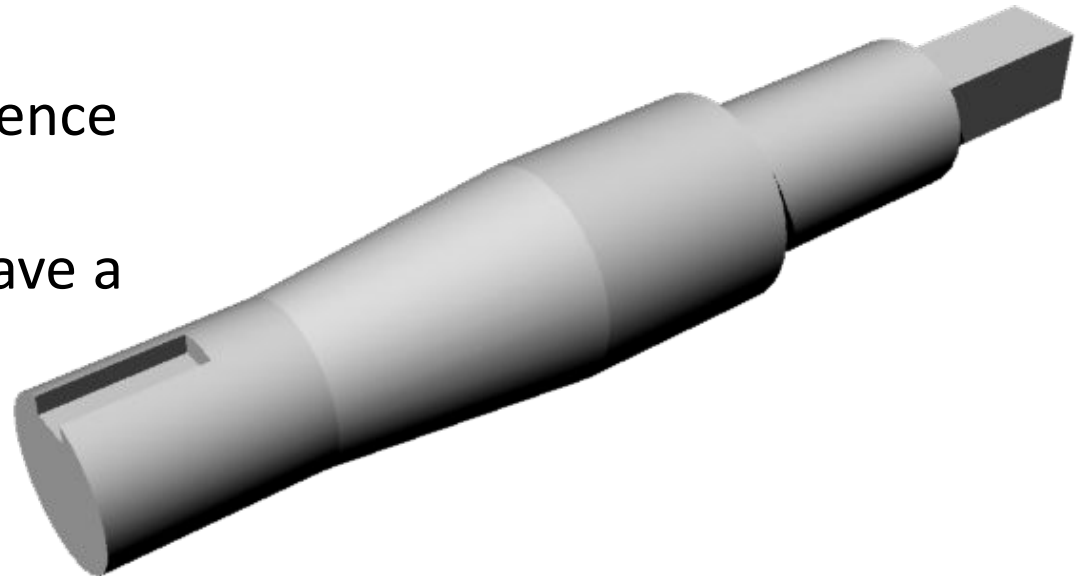
- Launch Rhino using mm as units and a tolerance of 1 micron (=0.001mm)
- Create five layers, named “shaft”, “flange”, “wheel”, “wedge” and “collar”; pick a different color for each layer
- Make “shaft” layer the current layer and create the shaft in the front viewport:
 - Create the profile of the circular cross-section part and generate the solid by revolving the profile around the shaft’s axis (Use the **Revolve** command).
 - Create a box of an appropriate size and generate the wedge slot with the aid of the **BooleanDifference** operation.
 - Create a box to represent the square cross-section part and finally use **BooleanUnion** to generate one solid representing the shaft

Step 1 Result



Hints:

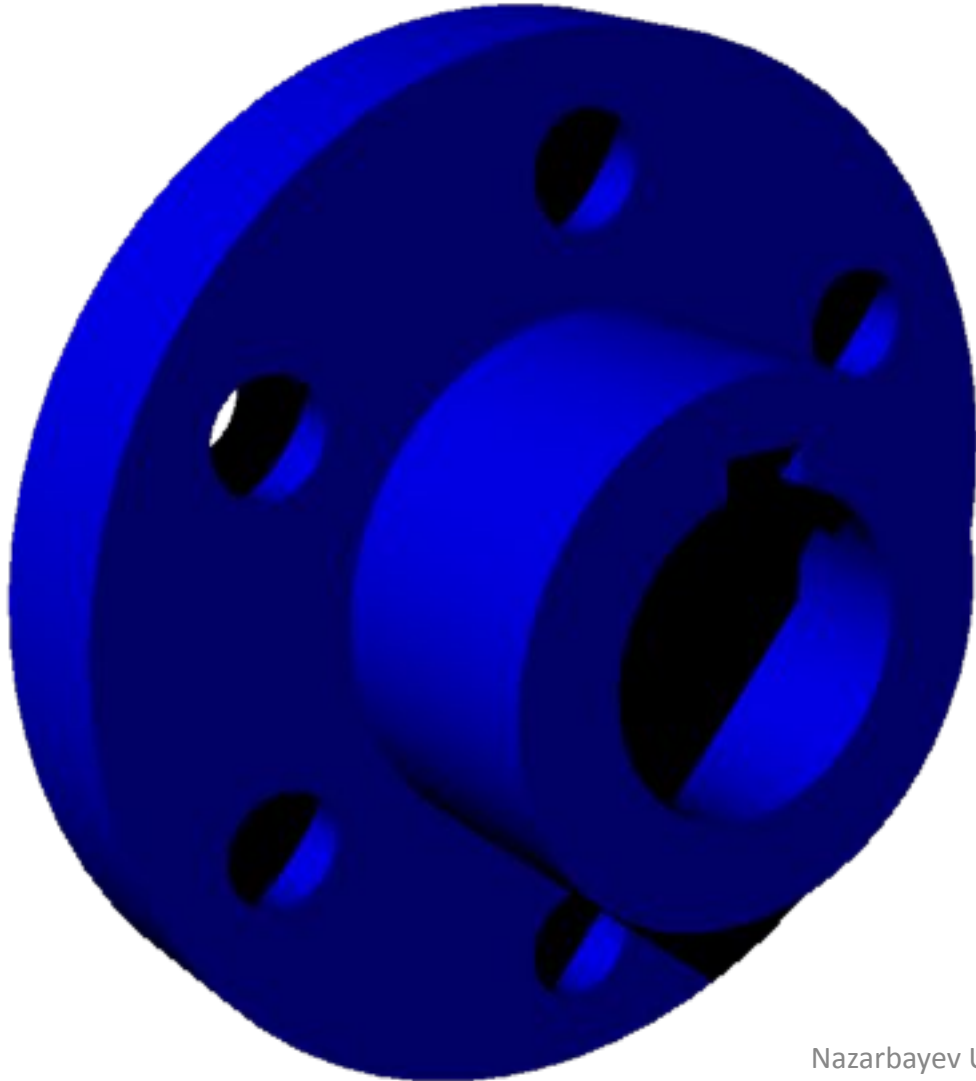
1. Use the subcommand "from" when moving/drawing a geometric entity with reference to an existing point
2. The box used for cutting the wedge slot can have a height larger than the slot's height
3. There's an option to create rectangles starting from their center point



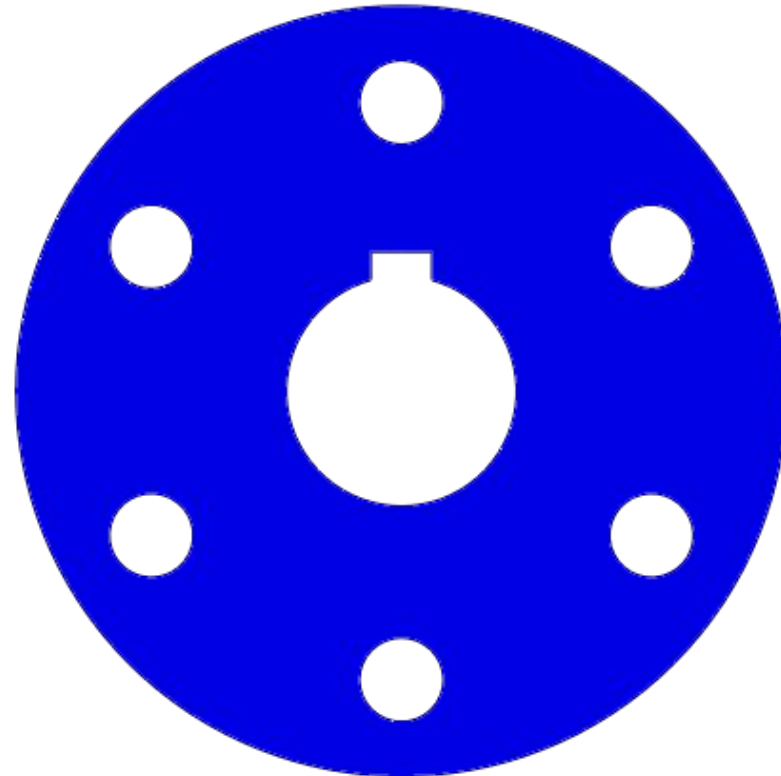
Modeling Steps: Step 2

- Make “flange” layer the current layer and create the flange in the ‘right’ or ‘left’ viewport:
 1. Create two circles with the same center and diameters 100 and 30, respectively.
 2. Using the upper **quad** of the small circle as a base point, create a **polyline** to represent the slot of the flange: the first point should be 3.1 mm above the quad; use **trim** to remove the not-required part of the inner circle.
 3. Create a **circle** of diameter 11 with a center that is 12.5 mm above the lower **quad** of the outer circle; use **arraypolar** command to create the remaining circles around the flange (use the common center of the two large circles as the center for the command and generate 6 circles filling an angle of 360 degrees); see right sketch in next slide.
 4. Create the first part of the flange solid by extruding (**ExtrudeCrv** with solid option set to Yes) all curves by 15 mm vertically to the construction plane.

Step 2 Result

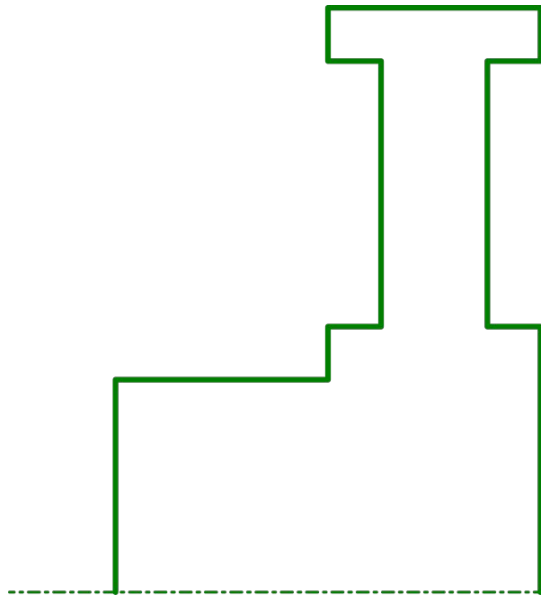


5. Create a circle with the same center as the outer circle of the flange and a diameter equal to 50mm
6. Execute the ExtrudeCrv command (make sure the solid option in set to yes), pick the newly created circle and the inner circular section of solid lying on the same plane as the selected circle and extrude both curves by 25mm; use union to unify the two extruded solids.



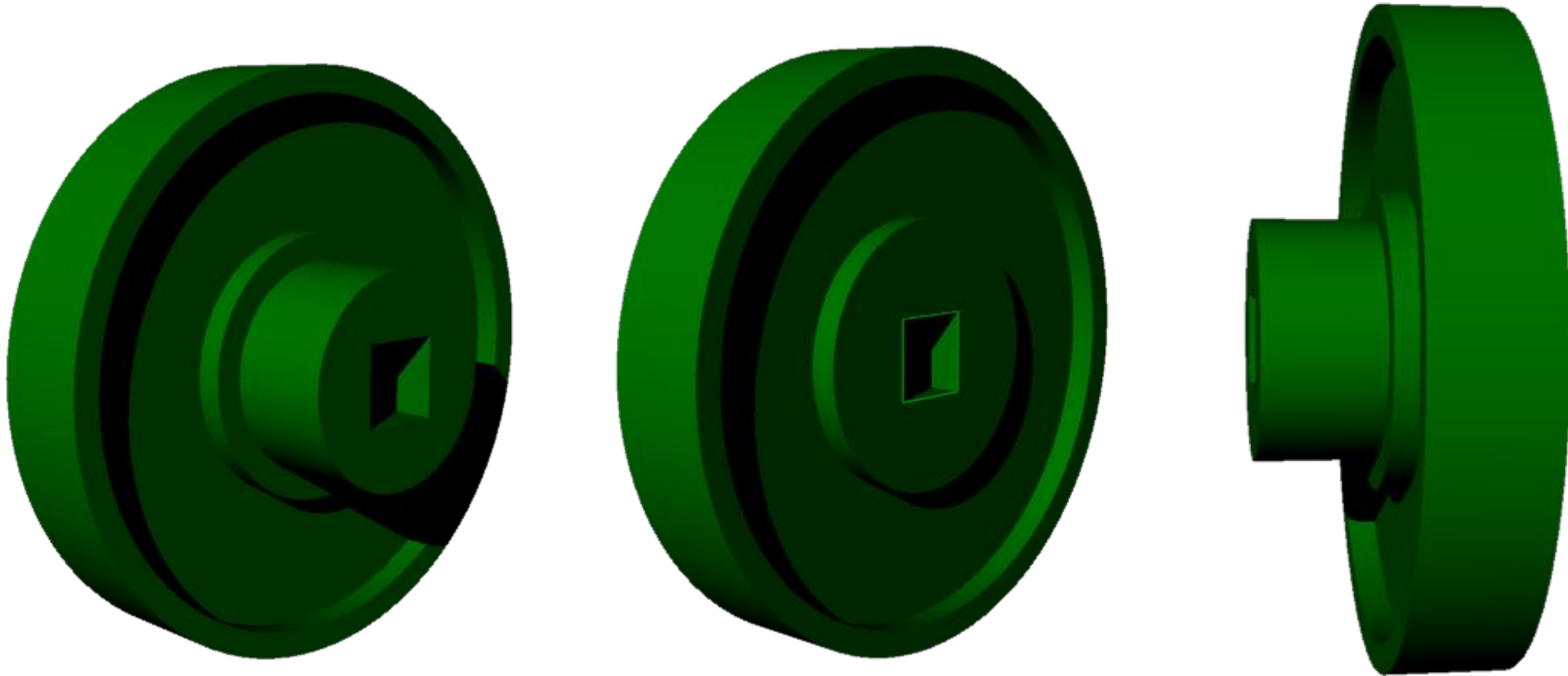
Modeling Steps: Step 3

- Make “wheel” layer the current layer and create the half of wheel’s sectional profile (ignore the square hole at the center; make sure the profile is a single curve)



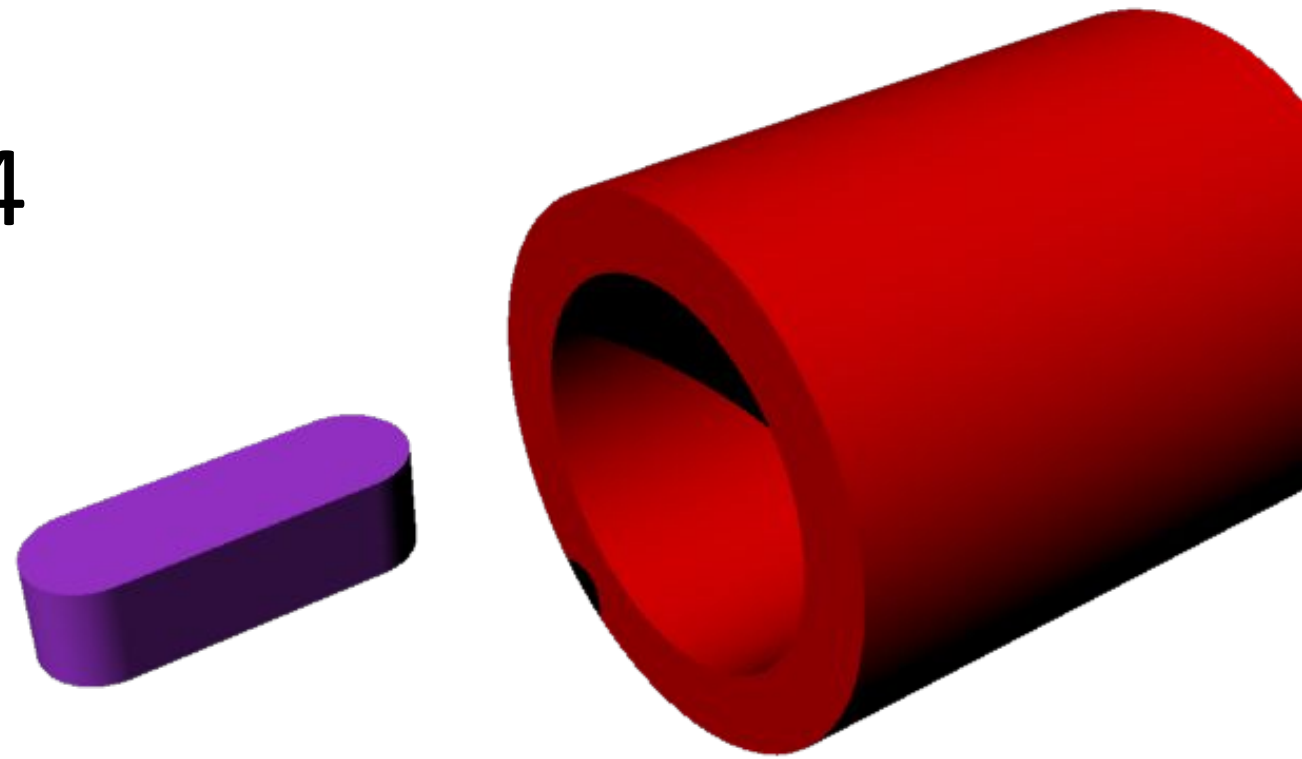
1. Use **Revolve** to create the wheel solid.
2. Create a **rectangle** (using the center-point option) on one of the two bases (on the ‘left’ or ‘right’ view); use **ExtrudeCrv** to create the solid representing the square-shaped hole at the center of the wheel (Alternatively, a **box** with appropriate dimensions could be used)
3. Use **booleanDifference** command to subtract the box from the wheel solid.

Step 3 Result



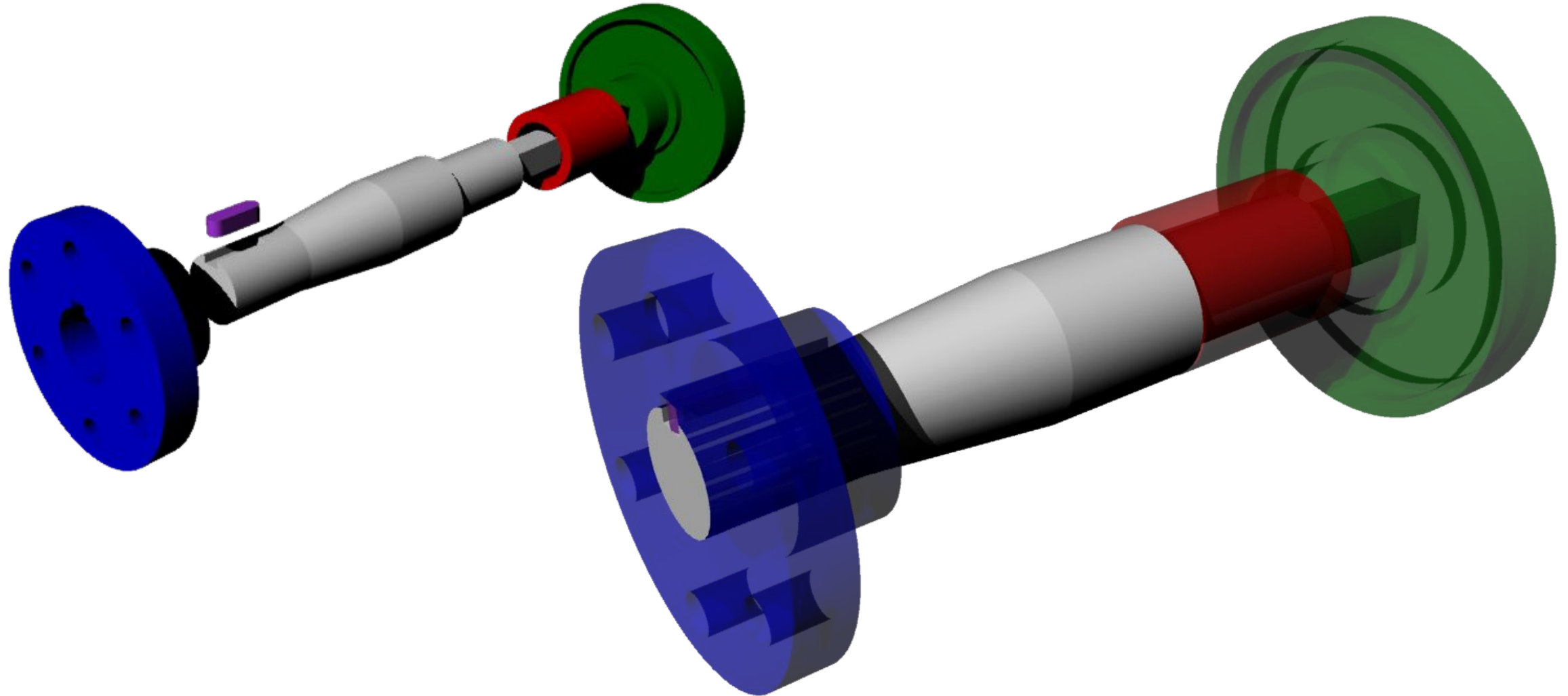
Modeling Steps: Step 4

Wedge: Make “wedge” layer the current layer; create the wedge by generating a box of appropriate dimensions and the use filletEdge to create the fillets (assume that fillet arcs meet at the corresponding side’s mid-point).

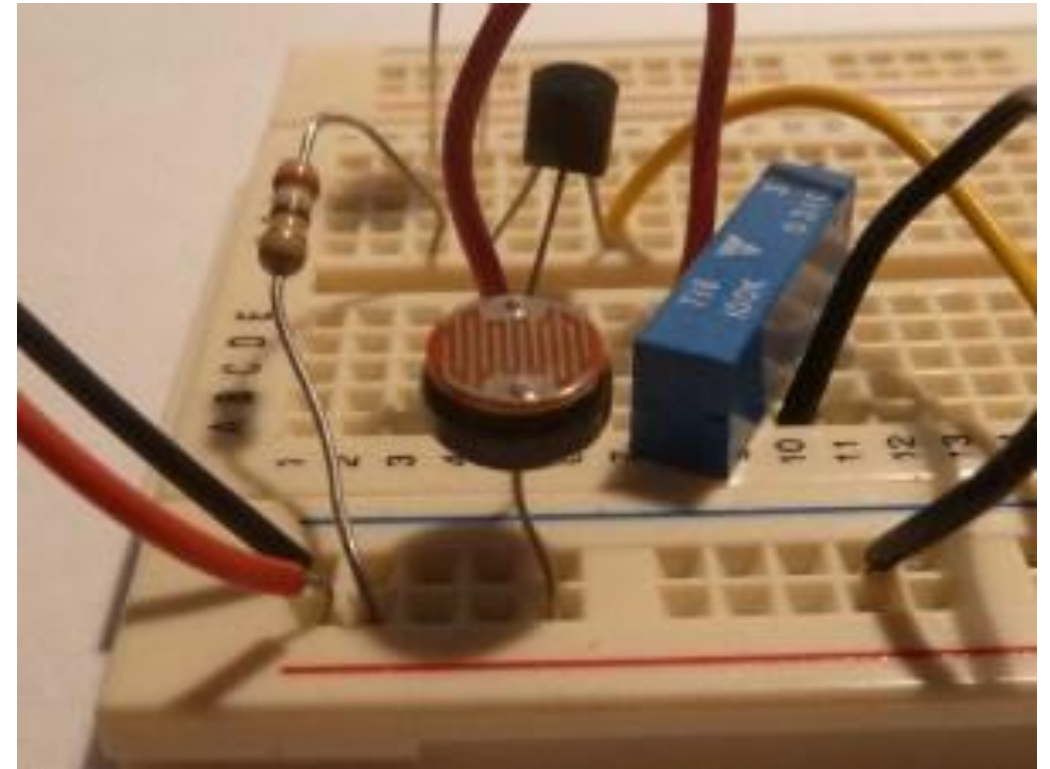
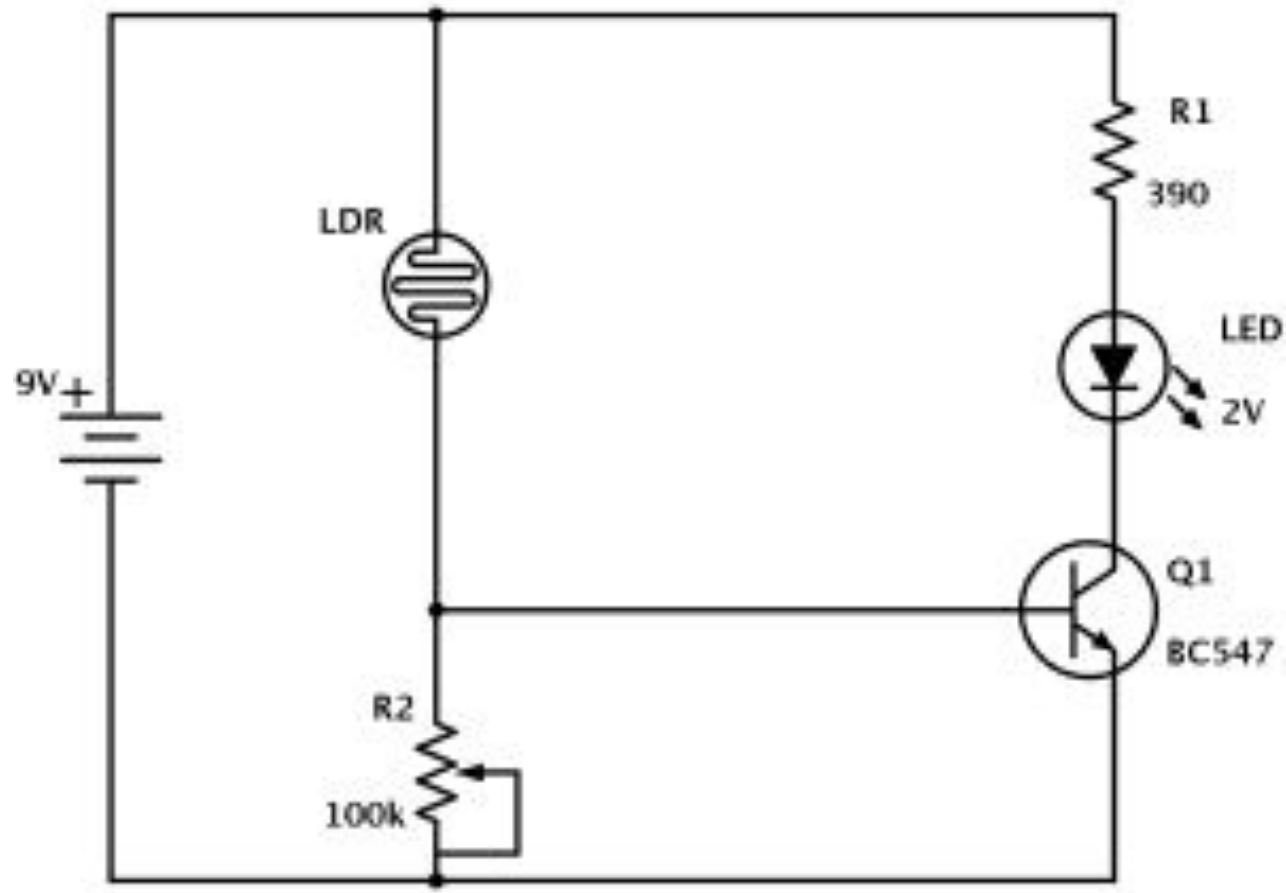


Collar: Make “collar” layer the current layer; create the collar by drawing two same-centered circles with diameters 30 and 42 mm and extruding both (ExtrudeCrv with solid option set to yes) straight for 50mm [use left or right view for the circles’ creation]

Modeling Steps: Final step (Assembly)



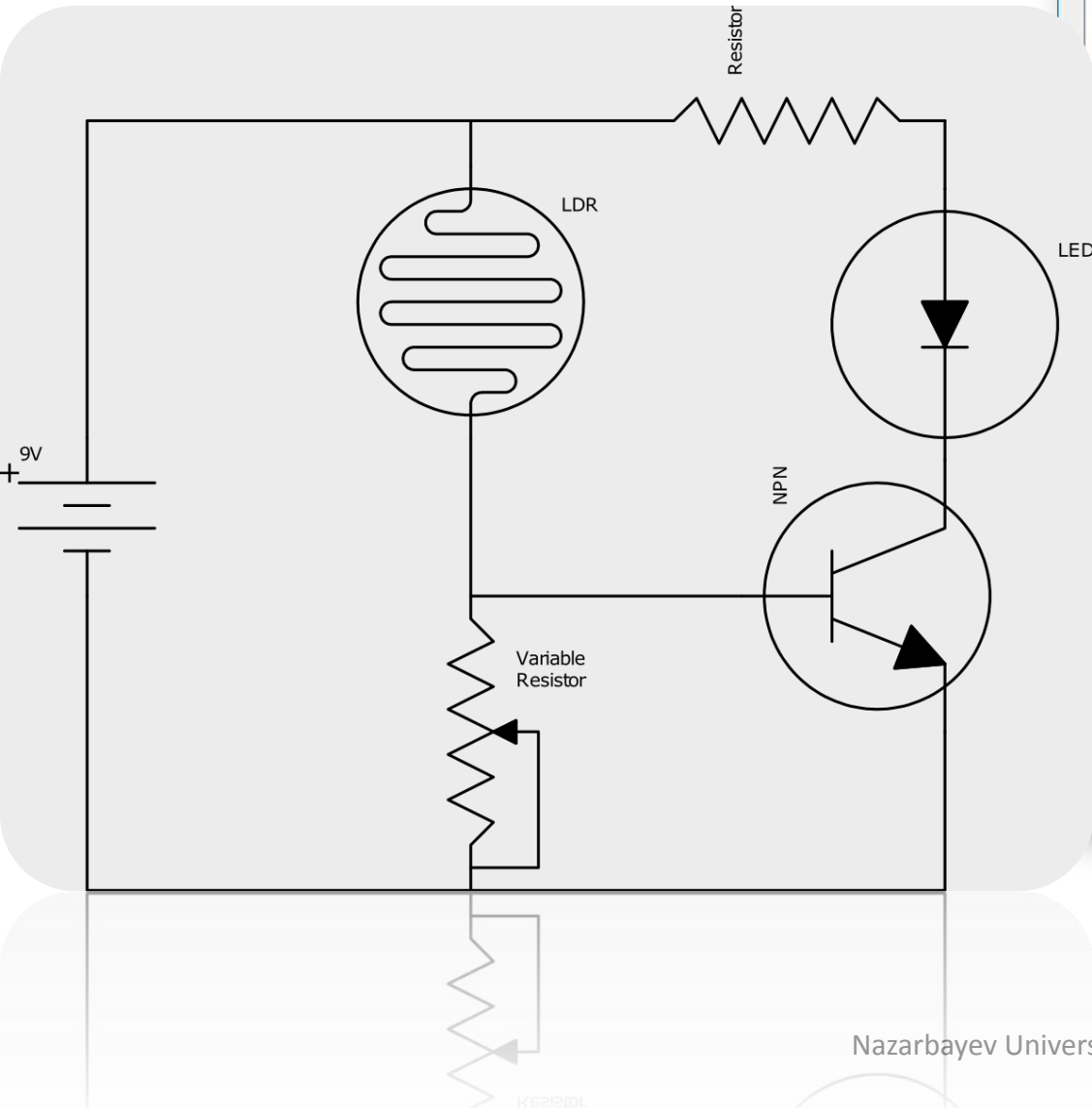
A simple circuit diagram



Modeling Steps: Step 1

- Create all components symbols as blocks using Rhino's block command
 - light-dependent resistor (LDR)
 - Potentiometer (variable resistor)
 - Resistor
 - LED
 - Transistor
 - 9V Power source
- Create the wiring (use approximate dimensions) and add all components as required
- Use **Text** command to add additional text on the diagram
- Use **Insert** command to insert already defined blocks and **BlockManager** to manage block library

Step 1 Result



Block Manager

Block Definitions:

| Name | Link Status | Link File Name |
|---------------|-------------|----------------|
| 9V power | Not linked | |
| LDR | Not linked | |
| LED | Not linked | |
| Potentiometer | Not linked | |
| Resistor | Not linked | |
| Transistor | Not linked | |

Model has no hidden or reference block definitions

☐ Show hidden block definitions

☐ Show reference block definitions

Description

Embedded and linked blocks should:

☐ Always update

☒ Prompt when update required

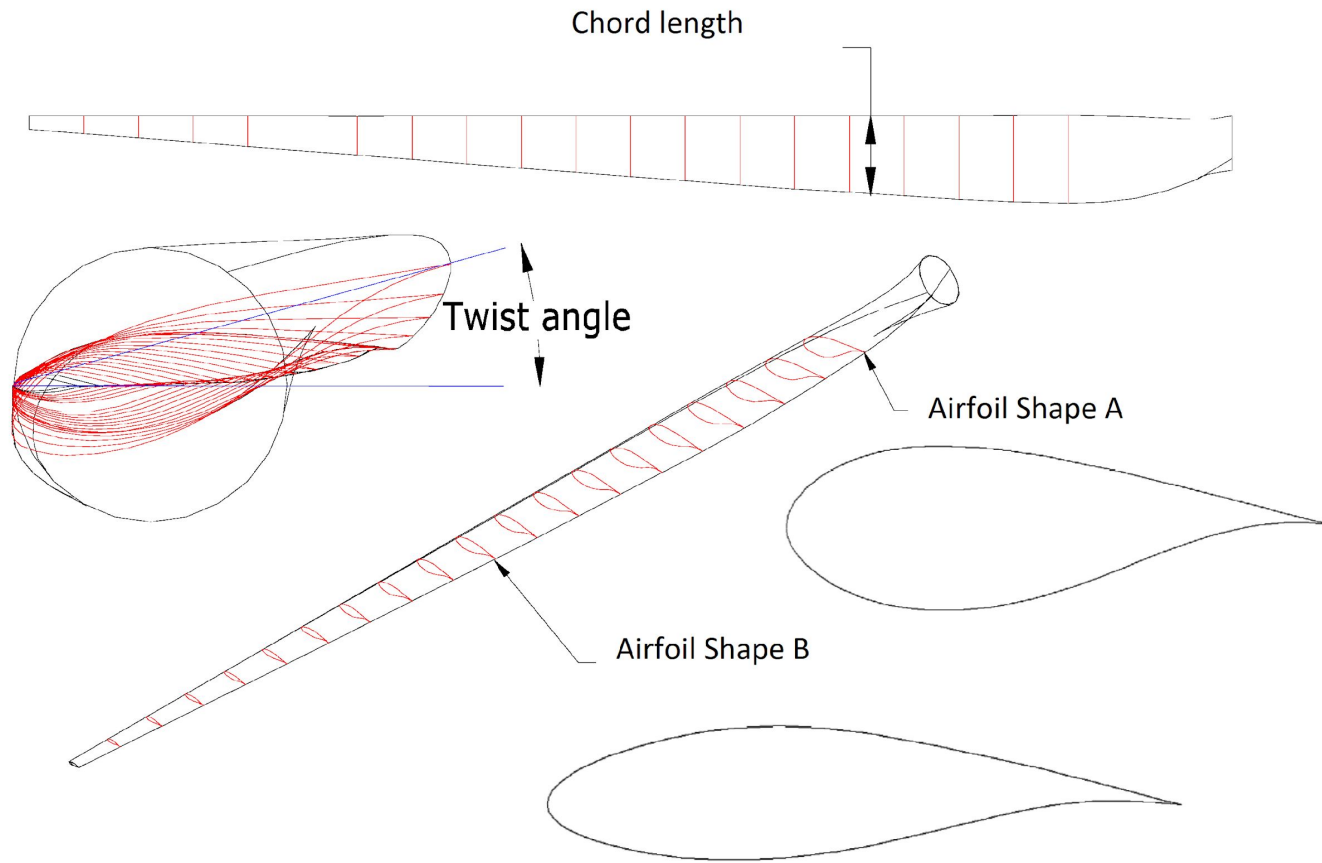
☐ Never update

☒ Block preview

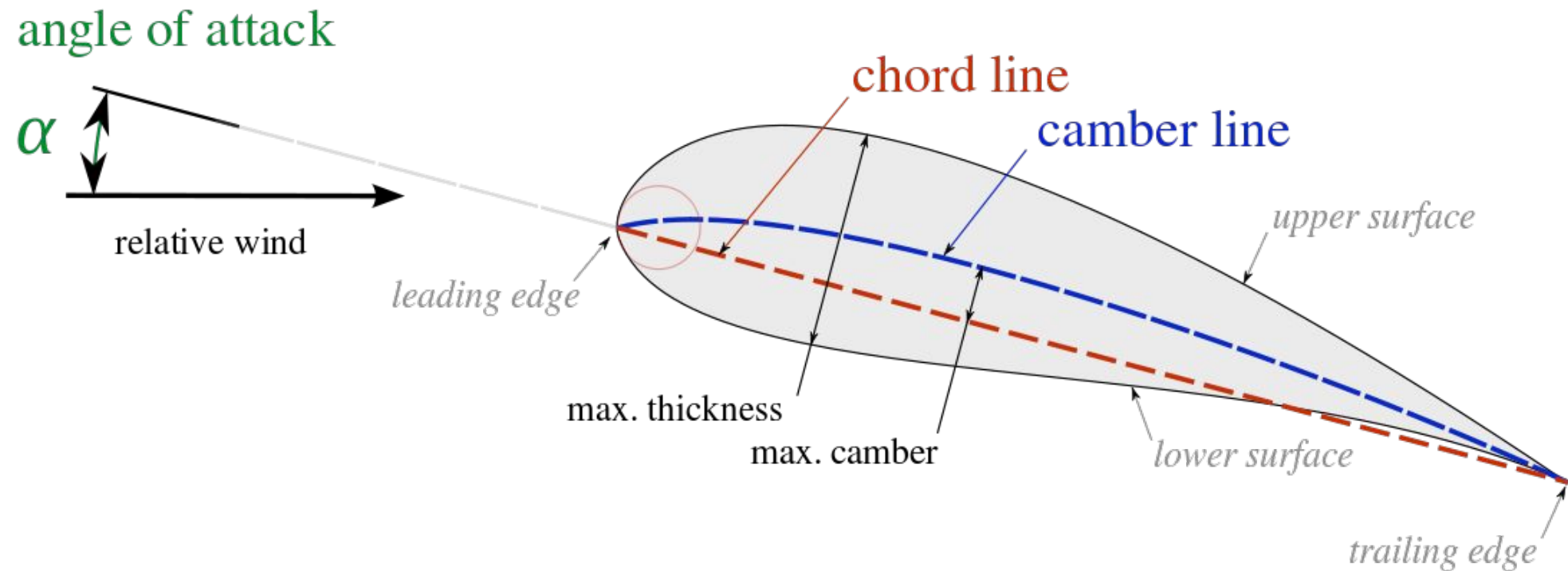
Help

Close

Simplified Wind Turbine Blade

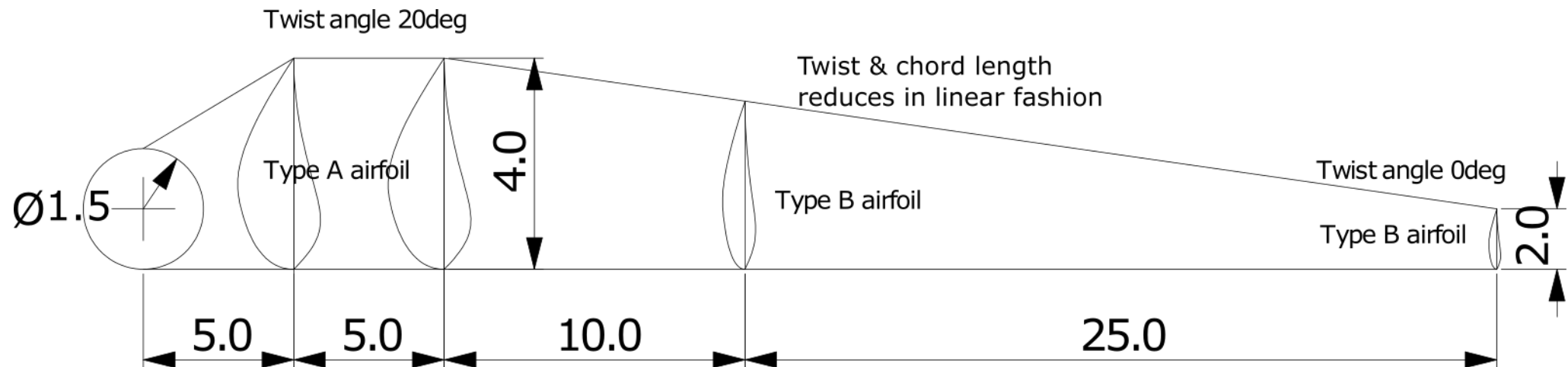


Basic airfoil terminology



Modeling Assumptions & input

- Airfoil type A shape: points in file airfoilA.txt
- Airfoil type B shape: points in file airfoilB.txt
- Assume an overall blade length $L = 45\text{m}$
- Twist rotation is performed with respect to airfoil's center (assume center to lie on the airfoil's chord, $1/3$ the distance from leading to trailing edge)
- Remaining dimensions and measurements as shown below:



Modeling Steps: Step 1

- Modify airfoilA.txt and airfoilB.txt to include the appropriate Rhino command (**InterpCrv**) to facilitate the automatic generation of the corresponding curves
- Using Tools □ Command □ Read from File... construct the two airfoils
- Create the hub-connection section (circular section) and place it at L=0m
- Place a scaled copy (enlarged by a factor of 4) of airfoilA at L=5m and L=10m
- Place a scaled copy (enlarged by a factor of 2) of airfoilB at L=45m
- Place a scaled copy (compute the required factor) of airfoilB at L=20m
- Rotate Type A airfoil copies 20° around blade's longitudinal axis (with respect to airfoil's center)
- Rotate the airfoil placed at L=20m using the required angle
- Create at least two intermediate sections between L=20m and L=45m

Step 1 Result

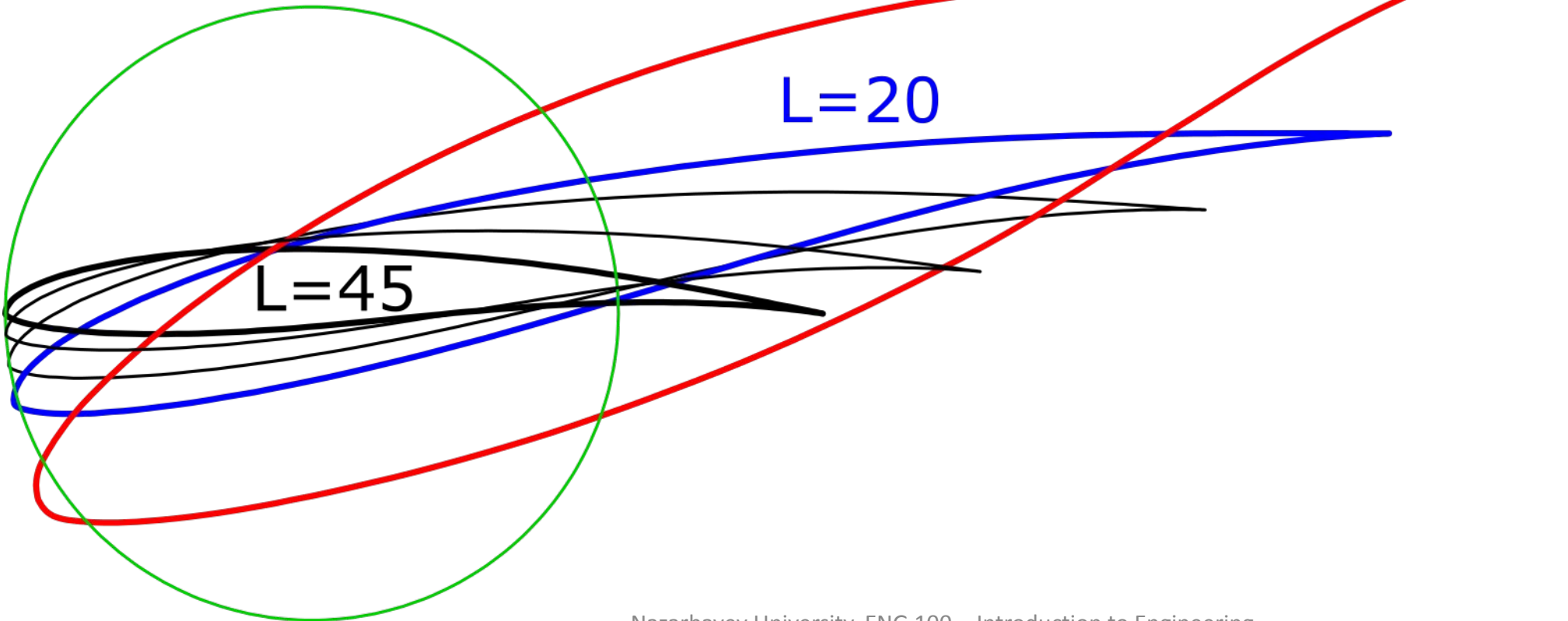
○



$L=5$ & $L=10$

$L=20$

$L=45$

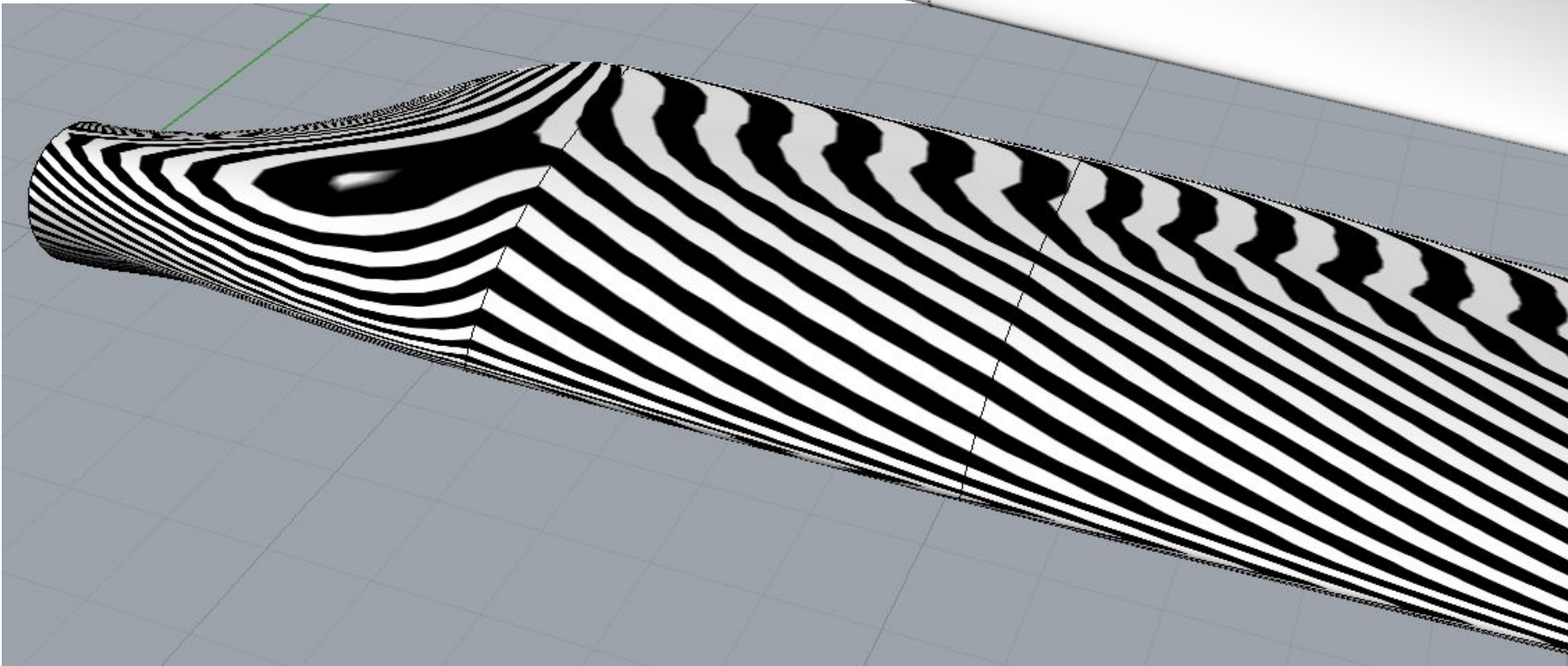
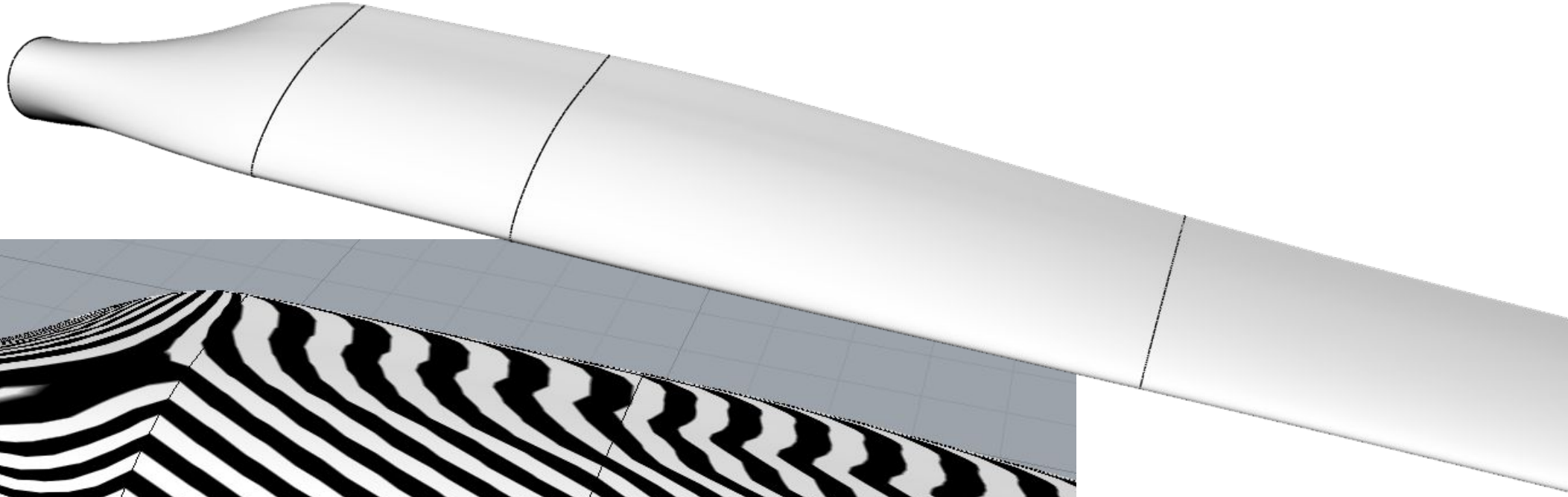


Modeling Steps: Step 2

- Use **ExtrudeCrv** to model the part between L=5 and L=10 (solid option should be set to No)
- Use **Sweep1** or **Sweep2** or **Loft** to generate the part between L=10 and L=45. Make sure that you interpolate all sections in that part and that you create the required rails, if you use Sweep1 or Sweep2. In all cases, pick the edge of the first surface so that you can create a tangent continuous surface.
- Use **BlendSrf** to create the surface patch between the hub and the first airfoil section. You may create an auxiliary cylindrical surface next to the circular section to get the required tangent plane information.



Step 2 Result

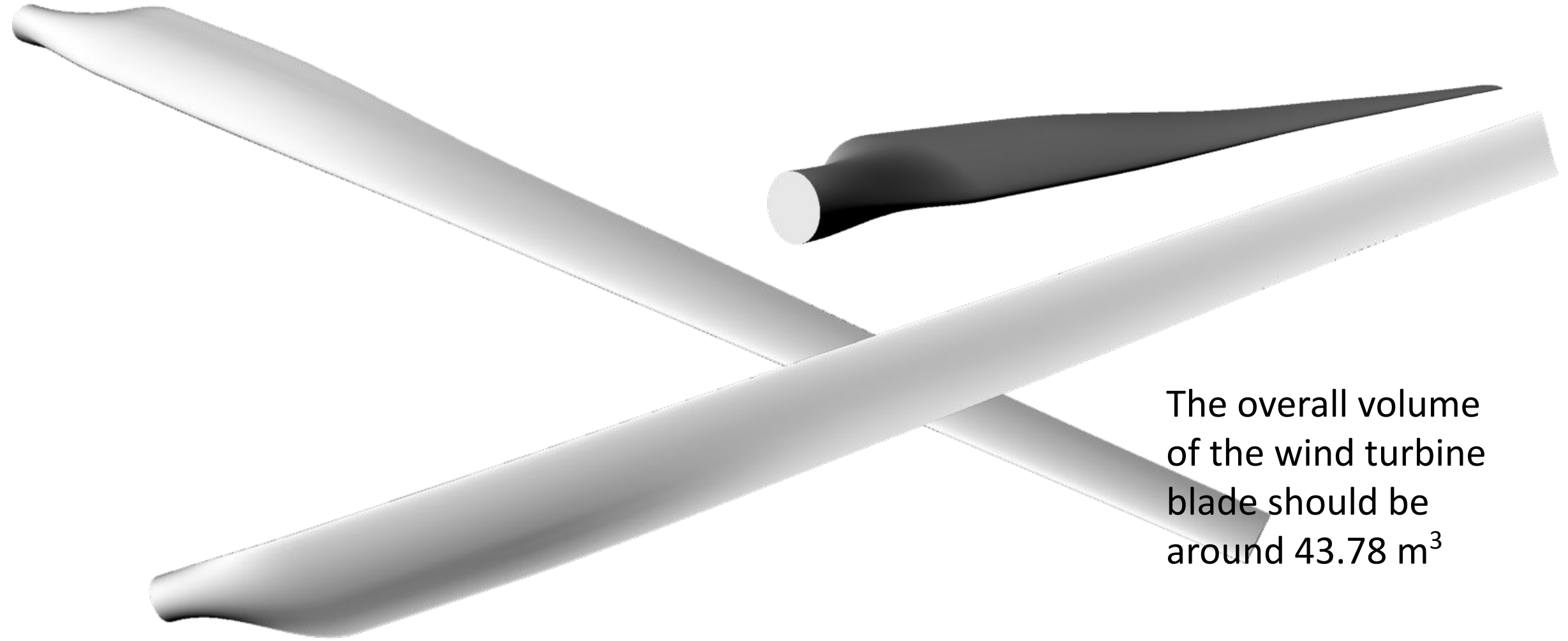


Using the **Zebra** command you may check the tangential continuity between surface patches

Modeling Steps: Step 3

- Join all surfaces together using the **join** command
- Use the **cap** command to create the missing planar surfaces and complete the blade solid model.
- Use the volume command to get the volume of the blade. If Rhino reports no volume then you probably have some gap(s) in your model:
 - Use **ShowEdges** and look for any Naked or Non-manifold edges. If such edges exist you need to remodel the parts affected.
 - If the gap is small, using **JoinEdge** might fix the problem for naked edges.

Modeling Steps: Final step



The overall volume of the wind turbine blade should be around 43.78 m^3