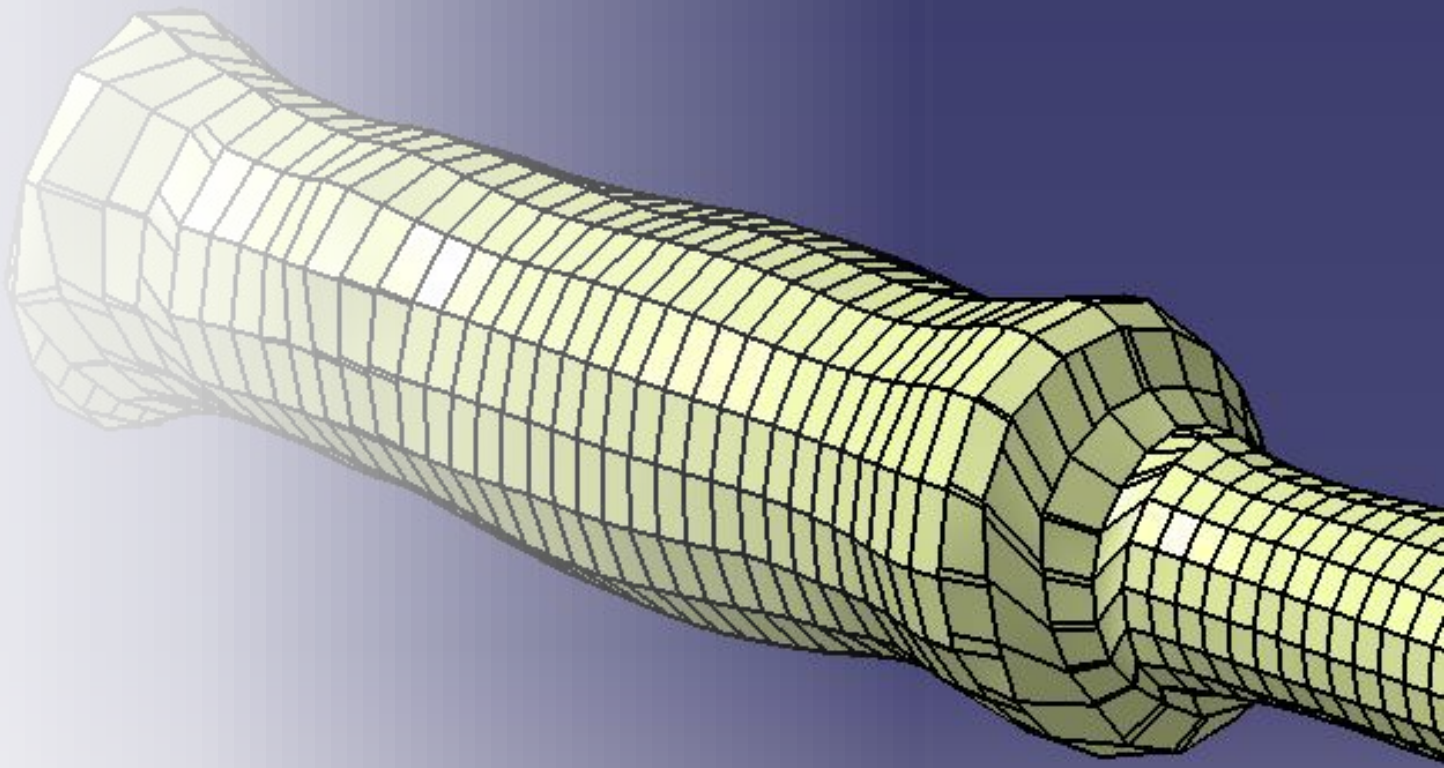
Automatic Geometry from STL



Design Automation Lab
mehdi.tarkian@liu.se

LINKÖPING UNIVERSITY

Objective

In this tutorial you will use clash analysis when stepping through a STL model (which you may have scanned) and then create a parametric* reference model

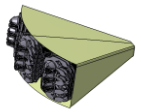
In order to really understand what code you have assembled, you need to step through the code (F8) and see the effects directly in CATIA

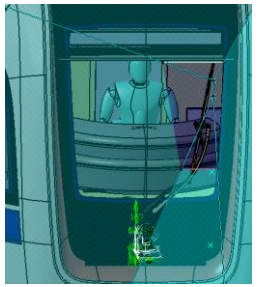
*Parametric in the sense that the objects can be utilized through API programming

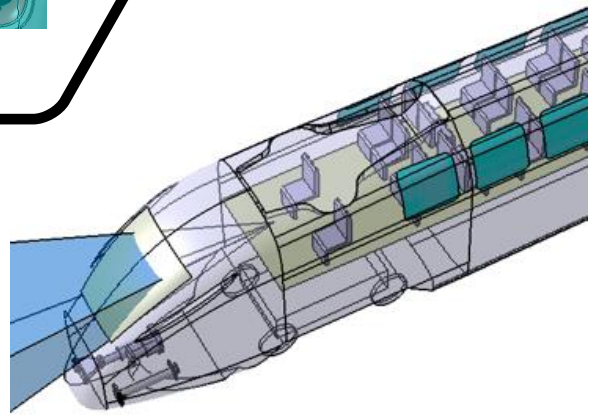
Design Automation Lab
mehdi.tarkian@liu.se

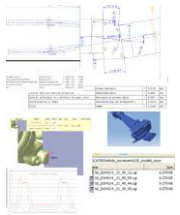
Why automated clash analysis is valuabe

- There are many situation where optimized placement is a time consuming and iterative process
- By utilizing an automated clash analysis it is possible to reduce the amount of tidious and boring work
- We have utilized such techniques for many industrial partners. Automated coupling analysis and automated driver view analysis at Bombardier Transportation is one example.





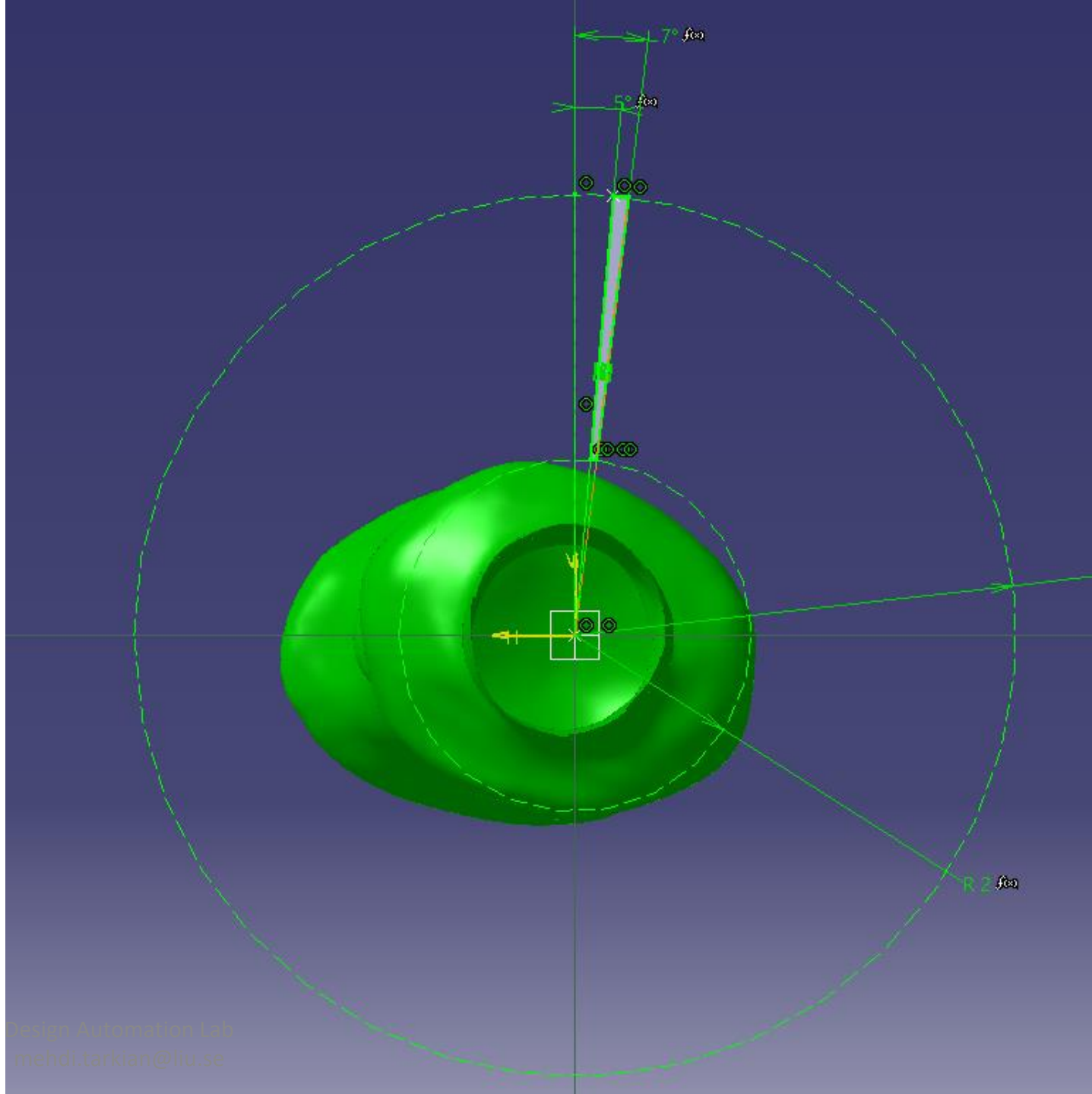




Design Automation Lab
mehdi.tarkian@liu.se

Building a slice (for the clash detection)

- The clash detection geometry is here built as circle arc as it is (subjectively speaking) an optimal shape because of ease of morphological modifications.
- You need to create a slice by utilizing foremost two construction circles
- See also next slice in regards to parameters used
- There is also a video included where it is possible to see how the automated procedure works

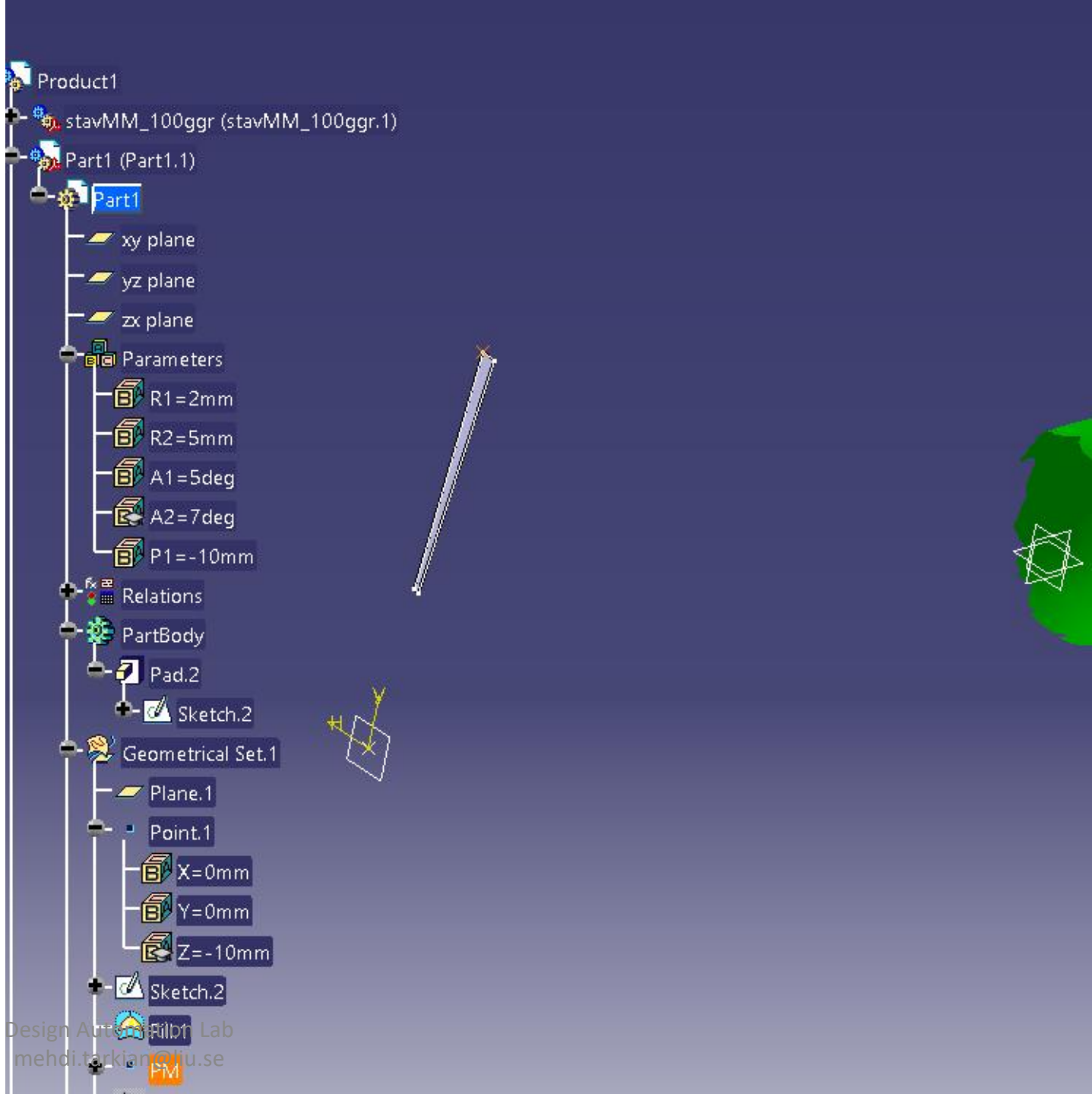


Parametric slice

Build the slice in a parametric way so that is can vary in size and also move along a specific direction.

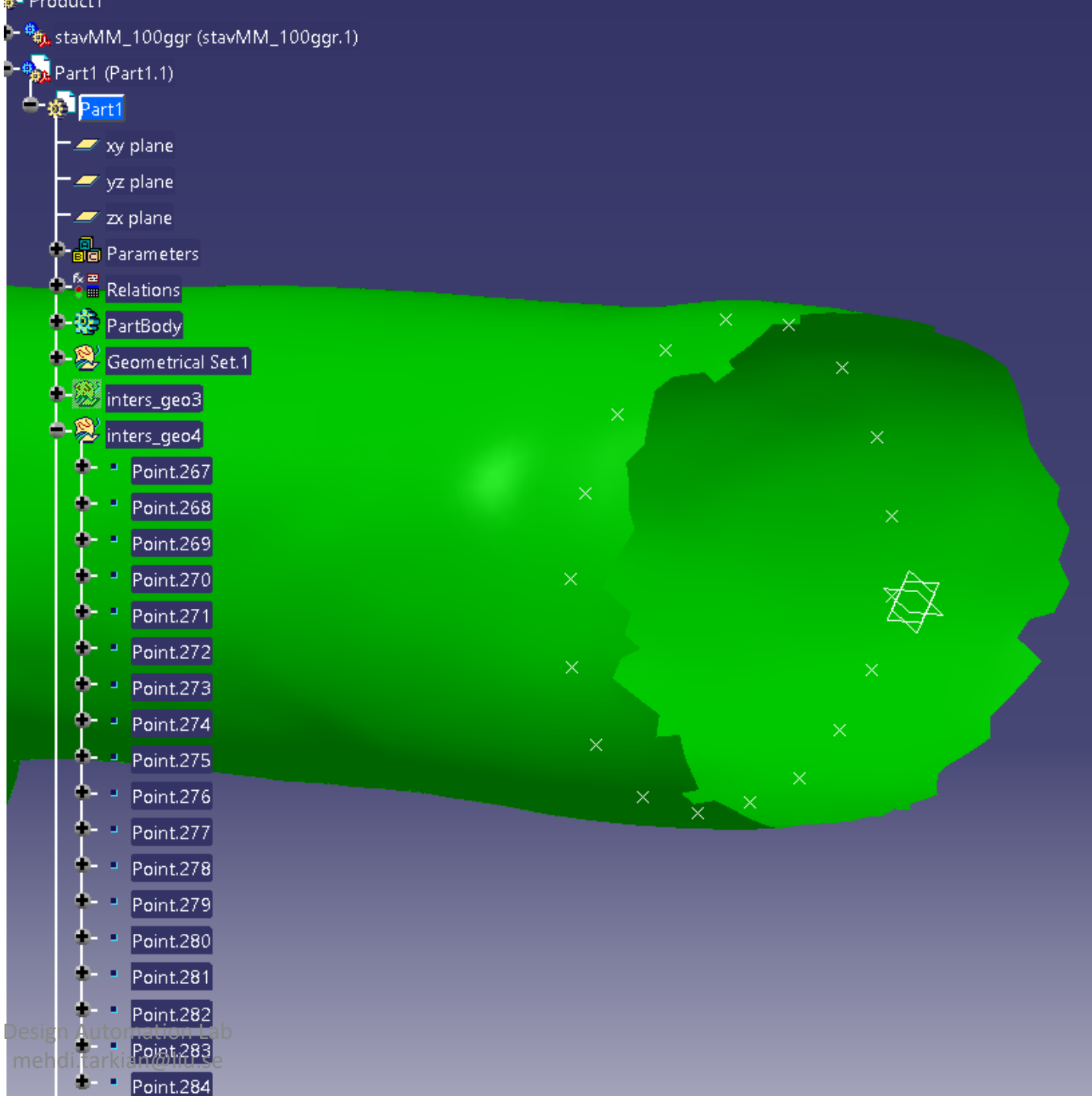
We want to use a script in order to start far away from the object and then close in and thus finding the first occurrence of collision.

Also note the point PM which is placed in a position where future clashes will help the code identify the position of the clash.



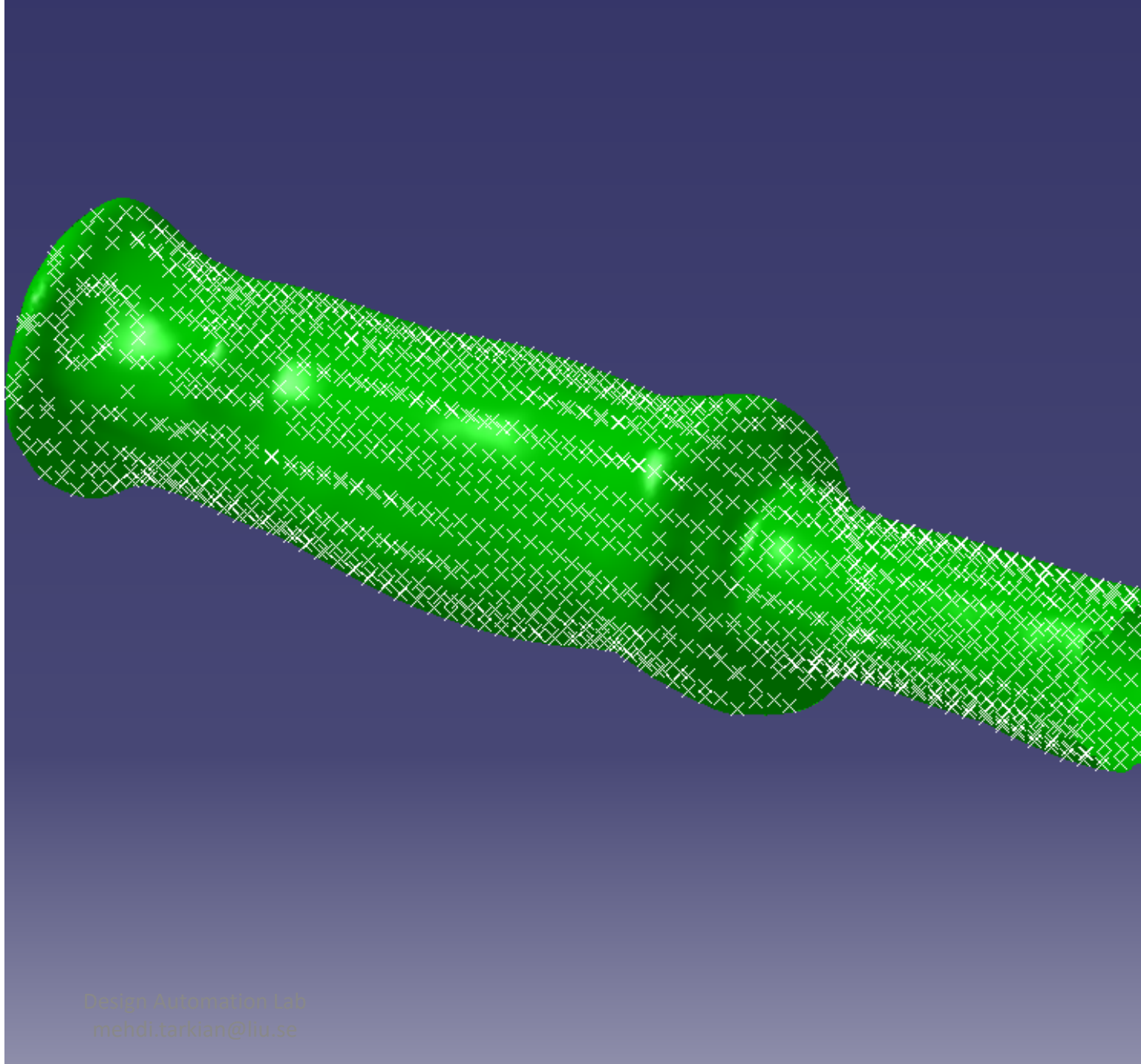
Finding intersections and creating points

By utilizing the codes in the upcoming slides it is possible to locate intersections and thus building points upon the stl geometry



Intersection-points on the entire object

The same code can then be used to create points on the whole object



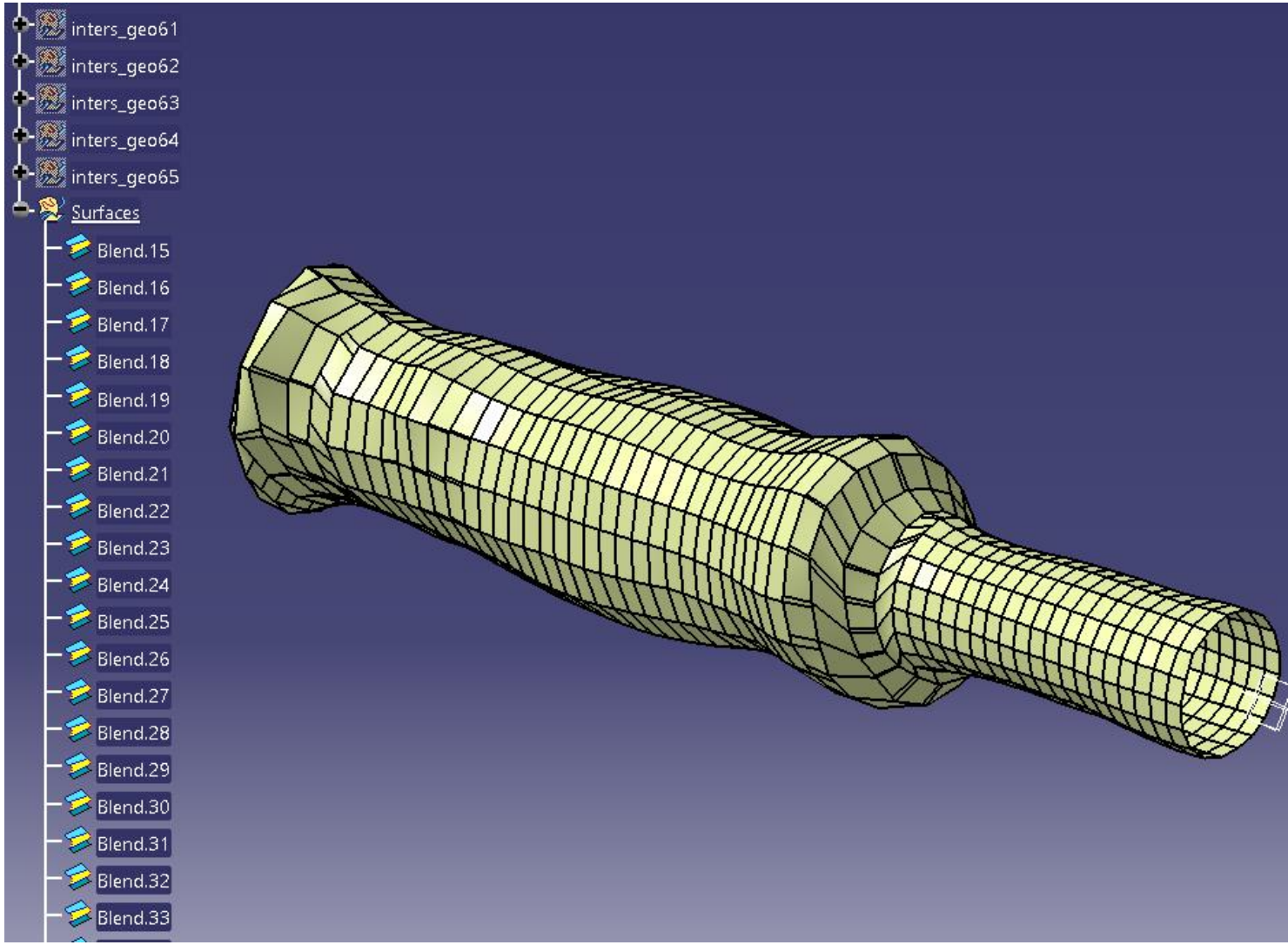
Design Automation Lab
mehdi.tarkian@liu.se

Creatin Blend Surfaces

By creating native CATIA surfaces we have now transformed the dead geometry to something which can be used in order to further build upon and automate in later stages.

In the following pages you will see code templates that you can use in order to assemble a working framework.

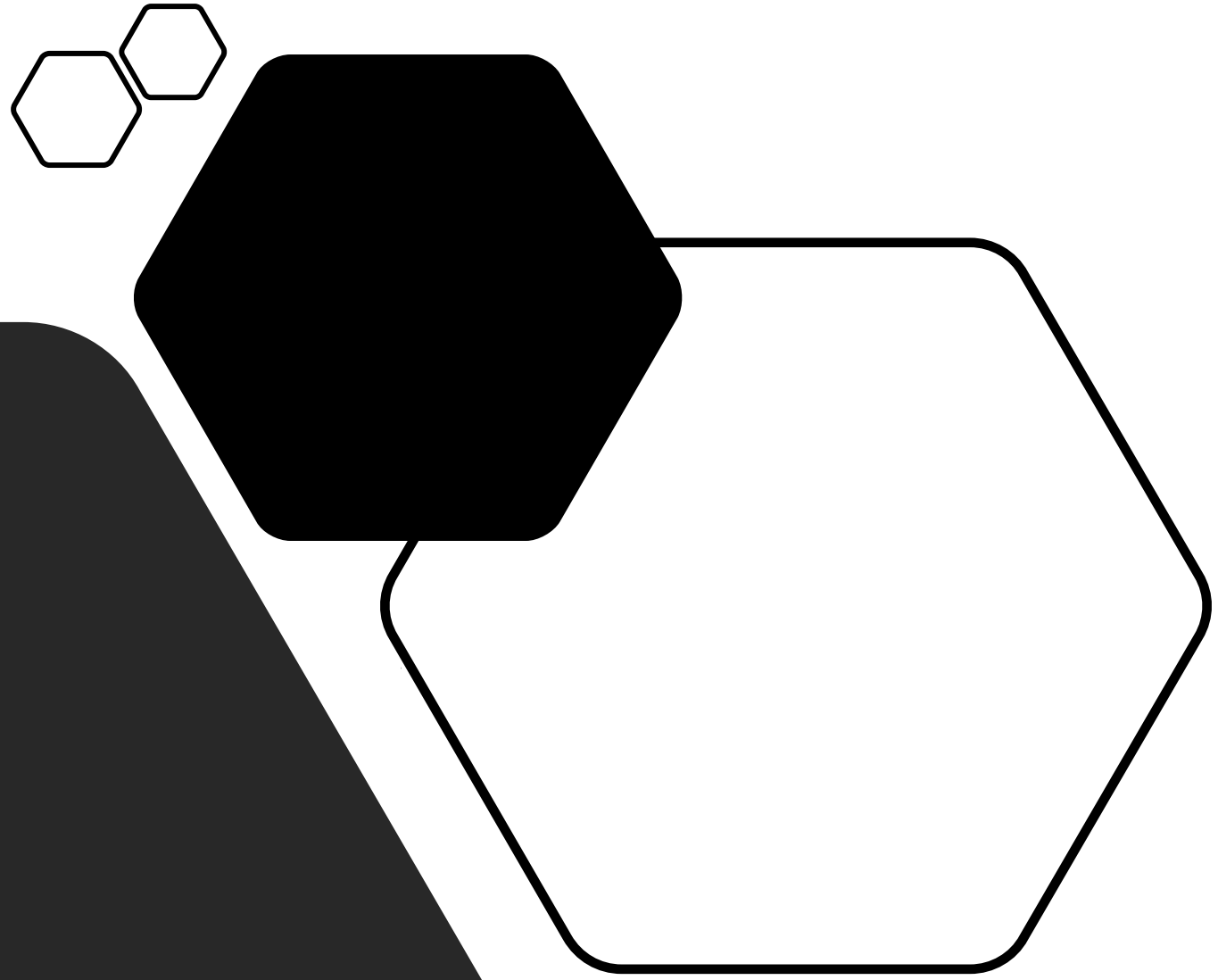
You still need to figure out how everything should be structured and there are a few templates that are not given and is up to you to figure out.



LINKÖPING UNIVERSITY

Design Automation Lab
mehdi.tarkian@liu.se

Main Sub-Routine



LINKÖPING UNIVERSITY

Design Automation Lab
mehdi.tarkian@liu.se

Main Subroutine

```
Set CATIA = GetObject(, "CATIA.Application")
Set documents1 = CATIA.Documents
Set oSel = CATIA.ActiveDocument.Selection

Set partDocument1 = documents1.Item("Part1.CATPart")
Set part1 = partDocument1.Part
Set Parameters1 = part1.Parameters
Set hybridBody12 = part1.HybridBodies.Add()
hybridBody12.Name = "poly"

stepdir1 = 0.5
Parameters1.Item("P1").Value = -10
Parameters1.Item("R2").Value = 10
part1.Update
k = 0
'thick_y = 10
inters = 0
```

Finding the first collision

Dir_v1 = 1

Start_v1 = -10

End_v1 = 50

Step_v1 = 5

P1_val = check_rep(k, part1, Parameters1, "P1", Dir_v1, Start_v1, End_v1, Step_v1)

Finding intersections all around the circle

```
For j = 1 To 100
   inters = inters + 1
   Set hybridBody11 = part1.HybridBodies.Add()
   hybridBody11.Name = "inters_geo" & inters

   For i = 1 To 359 Step 20

      Parameters1.Item("A1").Value = i
      Parameters1.Item("R2").Value = 10
      part1.Update

      Dir_v1 = 1
      Start_v1 = 0.5
      End_v1 = 10
      Step_v1 = 0.5
      k = 0
      V1 = check_rep(k, part1, Parameters1, "R2", Dir_v1, Start_v1, End_v1, Step_v1)
```

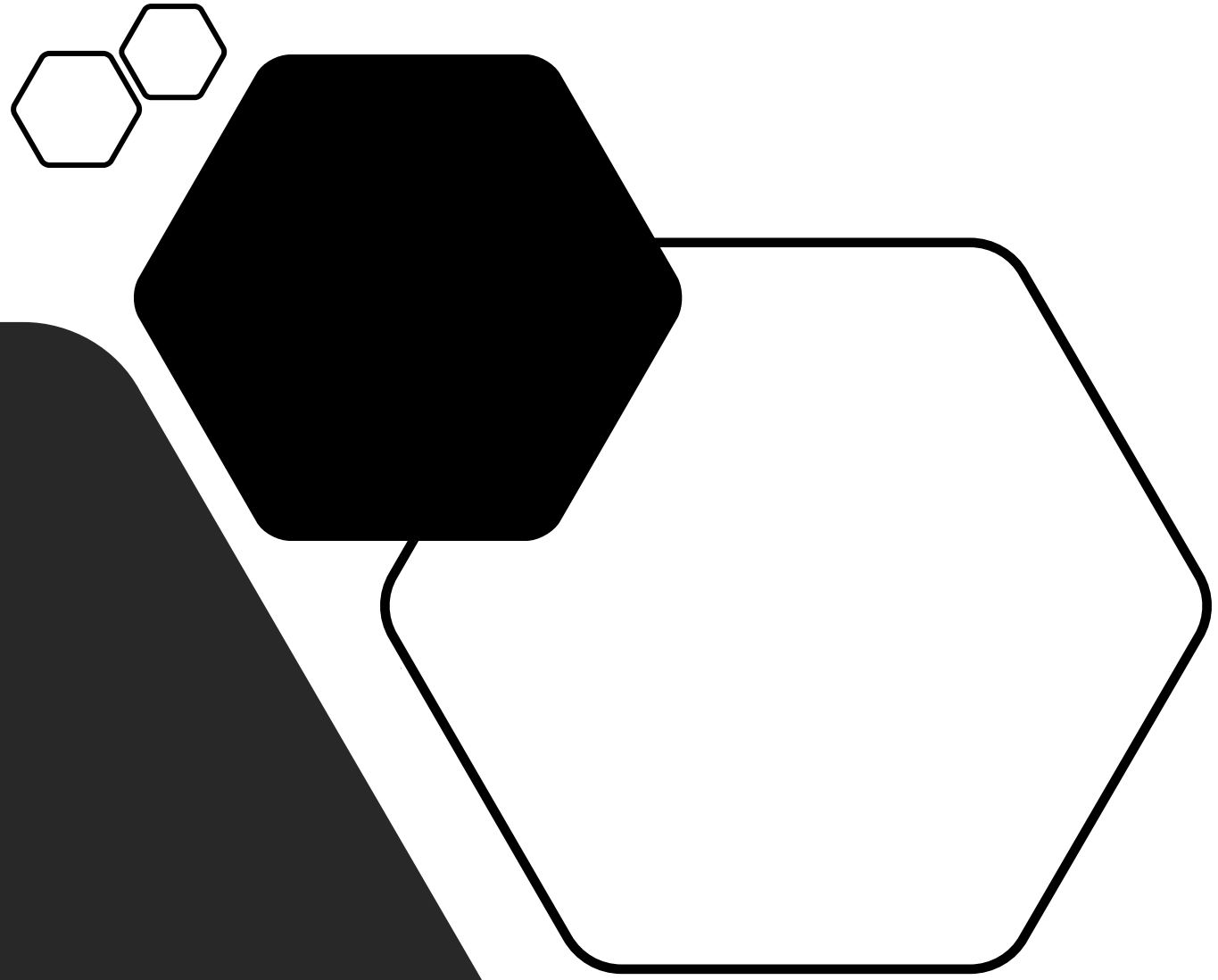
If there is no interection then remove the geometric set

```
If V1 >= End_v1 - 0.5 Then
        On Error Resume Next
            Set RM_H = part1.HybridBodies.Item("inters_geo" & inters)
            oSel.Clear
            oSel.Add RM_H
            oSel.Delete
            oSel.Clear
        On Error GoTo 0

        Exit For ' Exit loop since no intersection existed

    End If
```

Creating a point on the found intersection

```
Parameters1.Item("A1").Value = i

    Dim point2D1 As Object
    Set point2D1 = part1.HybridBodies.Item(1).Hybridshapes.Item("PM")
    Dim Point3D1 As Object
    Set Point3D1 = point2D1

    Dim Coords(2)
    Point3D1.GetCoordinates Coords

    Set hybridShapePointCoord1 = part1.HybridShapeFactory.AddNewPointCoord(Coords(0), Coords(1), Coords(2))


    hybridBody11.AppendHybridShape hybridShapePointCoord1
    part1.Update
```

Functions



LINKÖPING UNIVERSITY

Design Automation Lab
mehdi.tarkian@liu.se

Check collision

The code in the following page iterates through a given parameter in a certain range in order to find the first occurrence of clearance

Design Automation Lab
mehdi.tarkian@liu.se

Check collision

```
Function checkcol(k, steg1, part1, Parameters1, paramname, dir, start_val, end_v)
    k = start_val + k
    Parameters1.Item(paramname).Value = k
    part1.Update
    loop2 = (end_v - k) / steg1
    For i = 1 To loop2
    k = dir * steg1 + k
    Parameters1.Item(paramname).Value = k
    part1.Update
    kl = ClashAnalysis1
    If kl > 0 Then
    Exit For
    End If
    Next
    checkcol = k
End Function
```

Finer search of collision

In order to have more detailed search of where exactly the collision occurs, the function in following page is utilized

Finer search of collision

```
Function check_rep(k, part1, Parameters1, paramname, dir, start_val, end_val, step_size)

    k = checkcol(k, step_size, part1, Parameters1, paramname, dir, start_val, end_val)
    k = k - dir * step_size
    step_size = step_size / 10
    k = checkcol(k, step_size, part1, Parameters1, paramname, dir, 0, end_val)
    k = k - dir * step_size
    step_size = step_size / 10
    k = checkcol(k, step_size, part1, Parameters1, paramname, dir, 0, end_val)

    check_rep = k

End Function
```


Design Automation Lab
mehdi.tarkian@liu.se

Clash Analysis

```
Function ClashAnalysis1()

    '--- Initial Set-up ---
    Set CATIA = GetObject(, "CATIA.Application")
    Set oDocs = CATIA.Documents

    Set oProductDoc = CATIA.ActiveDocument
    Set oProduct = oProductDoc.Product
    Set oProducts = oProduct.Products
    Set oSel = oProductDoc.Selection

    oSel.Clear
    oSel.Add oProduct

    Set cClashes = oProduct.GetTechnologicalObject("Clashes")
    Set oClash = cClashes.AddFromSel
```

```
oClash.ComputationType = 0
    oClash.InterferenceType = catClashInterferenceTypeContact       'Clash + Contact
    oClash.Compute
    Set cConflicts = oClash.Conflicts

    k = cConflicts.Count
    ClashAnalysis1 = k



    oSel.Clear
    oSel.Add oClash
    oSel.Delete
    oSel.Clear

End Function
```

Function for creating polylines

```
Function create_poly_line(part1, hybridBody12, hybridBody11)

Set hybridShapeFactory1 = part1.HybridShapeFactory

Set hybridShapePolyline1 = hybridShapeFactory1.AddNewPolyline()

Set hybridShapes1 = hybridBody11.Hybridshapes
point_nr = 1
For i = 1 To hybridShapes1.Count
   Set hybridShapePointCoord1 = hybridShapes1.Item(i)

   Set reference1 = part1.CreateReferenceFromObject(hybridShapePointCoord1)

   hybridShapePolyline1.InsertElement reference1, point_nr
   point_nr = point_nr + 1
Next
```

```
   Set hybridShapePointCoord1 = hybridShapes1.Item(1)
   Set reference1 = part1.CreateReferenceFromObject(hybridShapePointCoord1)
   hybridShapePolyline1.InsertElement reference1, point_nr

hybridShapePolyline1.Closure = False

hybridBody12.AppendHybridShape hybridShapePolyline1
hybridShapePolyline1.Name = "Poly" & inters
part1.InWorkObject = hybridShapePolyline1

part1.Update

End Function
```

Function for creating Blend Surface

```
Function blend_surface(part1, hybridShapes1, poly1, poly2, pointgeoN1, pointgeoN2, hybridBody2)
Set hybridShapeFactory1 = part1.HybridShapeFactory
Set hybridShapeBlend1 = hybridShapeFactory1.AddNewBlend()
hybridShapeBlend1.Coupling = 1
Set hybridShapePolyline1 = hybridShapes1.Item("Polyline." & poly1)
Set reference1 = part1.CreateReferenceFromObject(hybridShapePolyline1)
hybridShapeBlend1.SetCurve 1, reference1
hybridShapeBlend1.SetOrientation 1, 1
Set hybridpoint1 = part1.HybridBodies.Item("inters_geo" & pointgeoN1).Hybridshapes.Item(1)
Set reference2 = part1.CreateReferenceFromObject(hybridpoint1)
hybridShapeBlend1.SetClosingPoint 1, reference2
Set hybridShapePolyline2 = hybridShapes1.Item("Polyline." & poly2)
Set reference3 = part1.CreateReferenceFromObject(hybridShapePolyline2)
hybridShapeBlend1.SetCurve 2, reference3
hybridShapeBlend1.SetOrientation 2, 1
Set hybridpoint2 = part1.HybridBodies.Item("inters_geo" & pointgeoN2).Hybridshapes.Item(1)
Set reference4 = part1.CreateReferenceFromObject(hybridpoint2)
hybridShapeBlend1.SetClosingPoint 2, reference4
hybridShapeBlend1.SmoothAngleThresholdActivity = False
hybridShapeBlend1.SmoothDeviationActivity = False
hybridShapeBlend1.RuledDevelopableSurface = False
hybridBody2.AppendHybridShape hybridShapeBlend1
part1.InWorkObject = hybridShapeBlend1
part1.Update
End Function
```

Results

- Depending on how detailed you search is the code will need around 10-60 minutes to complete
- If in doubt, then please discuss with the teachers
- When complete then share your result with the teachers

Design Automation Lab
mehdi.tarkian@liu.se