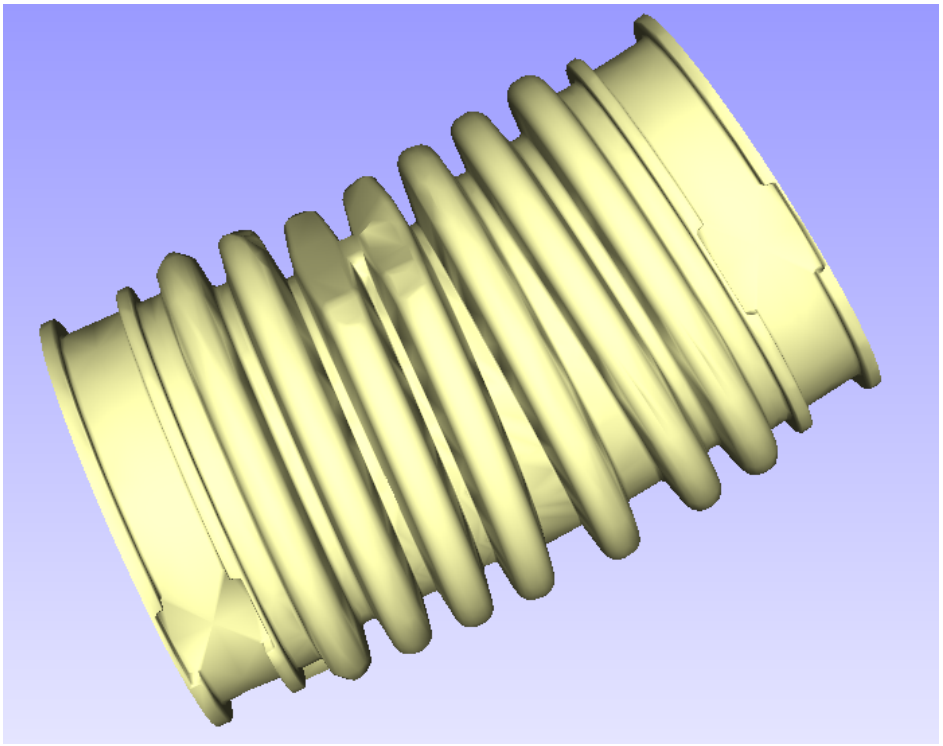
CHALMERS
UNIVERSITY OF TECHNOLOGY



Flexibel yta av luftbälg simulerad i IPS

Flexible surface of air bellow simulated in IPS

Examensarbete inom högskoleingenjörsprogrammet maskiningenjör

Andreas Björklund
Pontus Eljas

Institutionen för Produkt- och produktionsutveckling
CHALMERS TEKNISKA HÖGSKOLA
Göteborg, Sverige 2015

EXAMENSARBETE 2015

Flexibel yta av luftbälg simulerad i IPS

Examensarbete inom högskoleingenjörsprogrammet maskiningenjör

Andreas Björklund
Pontus Eljas

CHALMERS
UNIVERSITY OF TECHNOLOGY

Institutionen för Produkt- och produktionsutveckling
Avdelningen för produktutveckling
CHALMERS TEKNISKA HÖGSKOLA
Göteborg, Sverige 2015

Flexibel yta av luftbälg simulerad i IPS
Examensarbete inom högskoleingenjörsprogrammet maskiningenjör

Andreas Björklund
Pontus Eljas

Examensarbete 2015
Institutionen för Produkt och produktionsutveckling
CHALMERS TEKNISKA HÖGSKOLA
SE-412 96 Göteborg
Sverige
Telephone +46 31 772 1000

Framsida: Luftbälg simulerad i IPS
Tryckeri/Institutionen för Produkt och produktionsutveckling
Göteborg, Sverige 2015

Förord

Denna rapport är resultatet av ett examensarbete vid maskiningenjörsprogrammet på Chalmers Tekniska Högskola i Göteborg. Arbetet har bedrivits i samarbete med Fraunhofer-Chalmers Centre samt Volvo Cars under våren 2015 och omfattar 15 högskolepoäng.

Vi vill tacka vår handledare Johan Nyström på geometri och rörelseplanering avdelningen på Fraunhofer-Chalmers Centre samt även vår kontaktperson ifrån Volvo Cars som försett oss med material och komponenter, Peter Charliemo.

Vi vill även tacka vår handledare Mats Alemyr, tekniklektor vid institutionen för produkt- och produktionsutveckling på Chalmers Lindholmen.

Tills sist vill vi rikta ett stort tack till alla på Fraunhofer-Chalmers Centre för deras hjälp med våra frågor samt även för all hjälp vi har fått via Volvo Cars med vårt examensarbete.

Sammanfattning

Volvo Cars har under åren haft problem med att simulera luftbälgar på ett korrekt sätt. Dessa har tidigare simulerats som vanliga kablar, vilket betyder att luftbälgarna inte beter sig likadant i simuleringsprogrammet som i verkligheten.

Simuleringsprogrammet som Volvo Cars idag använder sig av heter Industrial Path Solutions (IPS) och det tillhandahålls via företaget Fraunhofer-Chalmers Centre.

Fraunhofer-Chalmers Centre har skapat en ny funktion i IPS vid namn Flexible Surface software, som kanske skulle kunna tillgodose Volvo Cars behov.

Syftet med detta examensarbete var att se ifall Flexible Surface software skulle kunna användas för att simulera mer komplexa objekt så som luftbälgar där Volvo Cars idag simulerar luftbälgarna som kablar. Luftbälgar är ofta komplext uppbyggda med fåror och fördjupningar samt är inte alltid cirkulära vilket är något Flexible Surface software kan hantera till skillnad mot dagens Cable Simulation.

Arbetet har jämfört hur simuleringar gjorts tidigare mot simuleringar med det nya programmet och kommit fram till att fortsatt utveckling av Flexible Surface software i samband med Volvo Cars är önskvärt. Mjukvaran hanterar luftbälgarna och resultaten ser lovande ut för en framtida implementering.

De stora sakerna som idag återstår för en full implementering är användarvänligheten, utvecklande av metoder för att simulera luftbälgar med varierande tjocklek, skapandet av bra centrumytor och någon form av verifiering av simuleringarna i verklig miljö.

Arbetet har till största del utförts på Fraunhofer-Chalmers Centre på Johanneberg men även på Volvo Cars i Torslanda.

Summary

Volvo Cars has for years had problems with simulating air bellows properly. These have previously been simulated as usual hoses, which means that the air bellows do not behave the same in the simulation program as in reality.

The simulation program that Volvo Cars use today is named Industrial Path Solutions and is provided via the company Fraunhofer-Chalmers Centre.

Fraunhofer-Chalmers Centre has created a new feature in the IPS called the Flexible Surface software, which could perhaps could meet Volvo Cars needs.

The purpose of this study was to see if the Flexible Surface software could be used to simulate more complex items such as air bellows which Volvo Cars today simulates as cables. The air bellows are often complex built with grooves and craters, which is something the Flexible Surface software can handle unlike the currently used Cable Simulation.

The work has compared how simulations have been done before with simulations from the new software and concluded that continued development of the Flexible Surface software in conjunction with Volvo Cars is desirable.

The software manages the air bellow and the results look promising for future implementation. The major things which today remains for a full implementation is the usability, development of methods to simulate air bellows with varying thickness, the creation of good center surfaces and some real life verification of the simulations.

The work has largely been carried out at Fraunhofer-Chalmers Centre on Johanneberg but also at Volvo Cars Torslanda.

Beteckningar

- **Cosserats teori** - Omfattar en lokal rotation av punkter samt translationen antas i klassisk elasticitet.
- **E-modul** - Elasticitetsmodulen, beskriver förhållandet mellan mekanisk spänning och deformation.
- **FCC** - Fraunhofer-Chalmers Centre
- **IPS** - Industrial Path Solutions
- **Kirchhoffs Modell** - Är helt icke-linjär i sina förflyttningar. Därför kan den användas vid stora förskjutningsberäkningar där materialet utsättas för endast små spänningar.
- **Mesh** -Består av en uppsättning av trianglar i tre dimensioner som är anslutna till varandra genom gemensamma hörn eller kanter.
- **NOD** - En nod innehar en position och dras till en stel kropp och får den stela kroppen att följa ett givet rörelsemönster.
- **Poissons konstant** - Materialkonstant som anger hur ett material reagerar på tryck- och dragkrafter.
- **Punktmoln** - Stort antal punkter i rummet som det är möjligt att skapa en geometri utav .
- **Shell** - Den skalmodell som IPS skapar från en yta
- **Skript** - Ett kodat dokument för att kunna köra saker ännu ej implementerat i program. I det här fallet IPS.
- **Vertex** - Hörnpunkt i triangel

Innehåll

1
Inledning

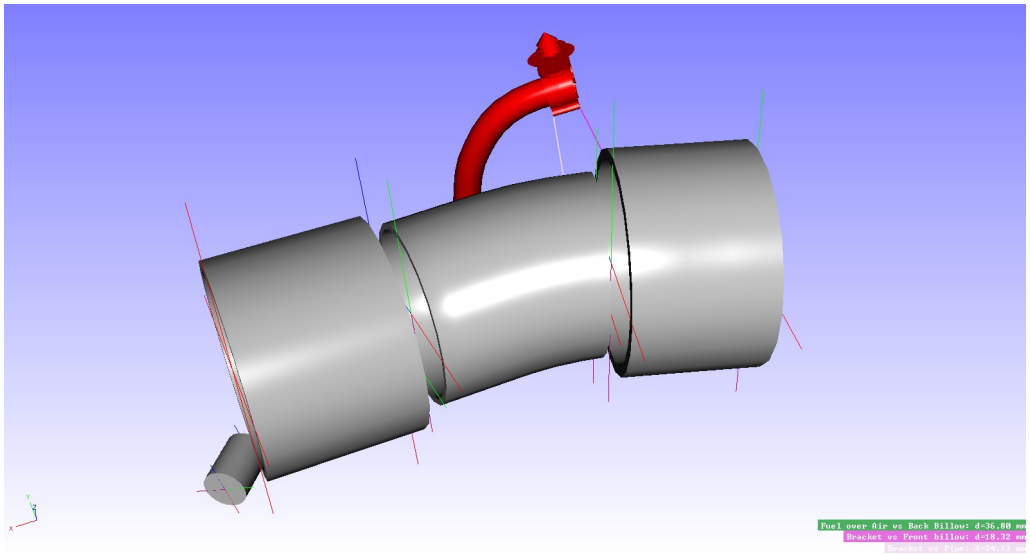
Volvo Cars använder sig av en programvara vid namn Industrial Path Solutions (IPS) och verktyget Cable simulation vid simulering av kablage och luftbälgar. Då Volvo Cars idag stöter på problem vid simulering av luftbälgar finns det ett behov av en bättre lämpad funktion och en bättre metod vid simulering. I arbetet kommer en ny funktion som ännu ej har implementerats i IPS att testas vid namn Flexible surface software. IPS-programvaran tillhandahålls och utvecklas av stiftelsen Franhofer-Chalmers Centre.

1.1 Bakgrund

Bilindustrin jobbar ständigt för att kunna minska tid och utvecklingskostnader samtidigt som höga krav på kvalité och flexibelt modellutbud kvarstår. Dessutom blir rena elbilar och hybridbilar allt vanligare på våra gator. För att kunna få plats med både en elmotor, en förbränningsmotor samt batteripaket krävs det att allt utrymme utnyttjas till maximal nivå. Detta utan att delarna ligger och skaver emot varandra och att komforten samt utrymmet i bilen skall utnyttjas till fullo. För att lättare kunna möta dessa krav sker stora delar av utvecklingen digitalt och många problem kan lösas utan att behöva bygga och prova på en fysisk modell. Då detta lyckas kan kostnaderna för utveckling reduceras markant. Noggrann simulering av luftbälgar har varit ett problem på Volvo Cars. Luftbälg är en detalj som är gjord av antingen gummi eller plast och sitter under huven på bilen mellan luftburken och luftinsuget från motorn. Idag skapas och simuleras luftbälgarna som endimensionella kablar i IPS, se figur 1.1. Detta skapar en inte verklighetstrogen bild vid simulering då luftbälgarna i verkligheten inte rör sig som en endimensionell kabel. Att virtuellt kunna planera hur kablar, slangar samt luftbälgar rör sig i motorrummet är viktigt. Detta för att kunna bestämma utformningen på respektive enskild del så att inga kollisioner, som kan leda till skav eller sönderslitningar, skall uppstå. Då FCC har tagit fram en ny funktion till IPS för att simulera flexibla ytor i två dimensioner finns det en förfrågan om denna kan användas för att simulera luftbälgar.

1.2 Syfte

Syftet med detta arbete är att utvärdera ifall den nya IPS-programvaran är möjlig att använda på Volvo Cars och om den i så fall är fördelaktig mot den gamla vid simulering av luftbälgar. För att uppnå detta syfte behöver ett antal luftbälgar testas, undersökas hur de simuleras och vilka frågor dessa simuleringar ska svara på.



Figur 1.1: Luftbälg simulerad som en en-dimensionell kabel men visualiserad tredimensionell i IPS.

1.3 Avgränsningar

Simulering och skapandet av ytor kommer enbart att ske på de luftbälgar upptagna i rapporten. Utöver detta kommer inte någon verifiering av simuleringarna att ske, utan de resultat som framkommer under arbetets gång kommer att utvärderas och lämnas vidare för framtida projekt.

1.4 Precisering av frågeställningen

1. Hur simulerar Volvo Cars luftbälgar idag? Vilka resultat är de viktiga och vilka är problemen man stöter på vid dagens simuleringar?

2. Hur kan Volvo Cars simulera de problematiska luftbälgarna i Flexible surface software och vilka problem kan uppstå vid förberedelser och simuleringar?

3. Vad kommer efter avslutat arbete återstå för att Volvo Cars ska kunna implementera Flexible surface software i sin dagliga verksamhet?

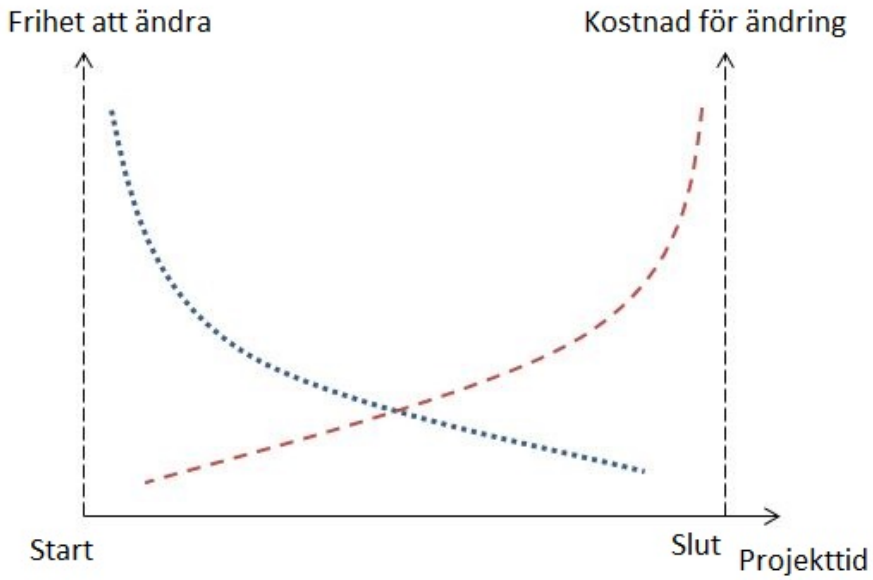
2

Teoretisk referensram

Detta kapitel kommer att behandla de olika teoretiska delarna som krävs för att kunna följa och förstå arbetet samt arbetsgången som används i rapporten.

2.1 Kostnads och effektivitetsökning av simulering i tidigt utvecklingsstadie

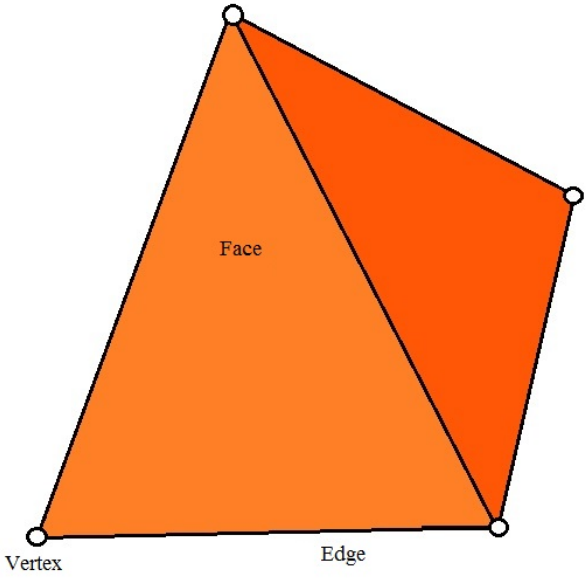
Investering i ett simuleringsprogram kan spara mycket tid och pengar. En simulering är en form av imitation av en verklig händelse för att i ett tidigt skede kunna återskapa möjliga situationer och hänvisa till de problem som kan uppstå. Att ett företag i ett sådant tidigt skede kan få möjlighet att prova om deras konstruktioner fungerar som tänkt skapar en stor kostnadsbesparing. Alla ändringar under ett utvecklingsprojekt kostar pengar, desto senare i processen, desto dyrare och mer omständigt. Se figur 2.1. Simuleringar skapar därför en möjlighet för företaget att tidigt i produktutvecklingen hitta möjliga felaktigheter samt göra något åt dessa. [1]



Figur 2.1: Kostnader och möjligheter för förändringar under ett projekt. [2]

2.2 Mesh

En mesh, se figur 2.2 är uppbyggd utav ett antal trianglar. Denna triangulering, består av vertexes, edges och faces. Den information som fås fram av dessa element är mesh-geometrin samt mesh-anslutning.
Mesh-anslutning även kallat Topologi beskriver relationerna mellan de olika maskelementen. Topologi redogör för hur kanterna av olika faces ser ut och hur angränsande vertex ligger i förhållande till varandra.
Mesh-geometrin anger geometriska egenskaper och positionen hos varje vertex. För att köra olika analyser av meshstrukturer krävs effektiva algoritmer som stöder kompakta datastrukturer eftersom att trianglarna i meshen ofta är komplexa, stora samt väldigt många. [13]



Figur 2.2: Bildbeskrivning av mesh.

2.3 Industrial Path Solutions (IPS)

IPS är en matematikbaserad programvara utvecklad av Fraunhofer-Chalmers Centre för tillverkningsindustrin där användaren har ett flertal olika verktyg att arbeta med i det tidiga utvecklingsstadiet. Dessa verktyg är, Cable simulation, Flexible surface software, Path planner, Robot optimization, Virtual paint, IMMA och Iboflow. [3]

2.3.1 Cable simulation

Cable simulation är ett verktyg i IPS-programvaran där användaren kan simulera kablage med minimalt antal simuleringar som behöver valideras utav fysiska tester. Huvudfunktionen i cable simulation är realtidsberäkning vid rörelse och deformation av kablar, slangar och andra typer av slangliknande föremål med materialparametrar. [4]

2.3.1.1 Simulering i en dimension

Cable simulation bygger på innovativa matematiska metoder i kombination med numeriska procedurer som gör det möjligt att behålla den nödvändiga fysiska noggrannheten. Vid realtidssimulering krävs det att simuleringsverktyget kan hantera stora deformationer samtidigt som det behöver vara beräkningsmässigt effektivt.[8]
För att åstadkomma detta används Cosserat rods, som blivit populära då de har precis dessa egenskaper. I strukturmekanik ses långsmala föremål likt kablar, slangar och liknande som en-dimensionella stänger eller balkar då de har en relativt liten diameter jämfört med dess längd på exempelvis en stång. Detta gör att trots en stor deformation av stången mot dess normala tillstånd, blir de lokala påfrestningarna små och tvärsnittet nästan orört. [10]

2.3.2 Flexible surface

Ett ännu inte lanserat verktyg i IPS som utnyttjar en tvådimensionell triangulerad yta. På denna ytan väljs materialparametrar och tjocklek ställs in varefter den nya modellen blir möjlig att simulera. På ytan väljs randvillkor och sedan kan punkt-, yt- och linje-krafter läggas på objektet för att bland annat kunna se deformationer och påfrestningar. [14]

2.3.2.1 Simulering i två dimensioner

Genom att skapa en skal-modell av Cosserat typ definierad av en triangulering, går det sedan att härleda en rotationsfri Kirchhoff modell med triangelhörnen som frihetsgrader. Denna modell som IPS Flexible surface simulator bygger på beter sig fysiskt rimligt redan vid grova ojämna trianglar. [9]

2.4 Meshlab

Meshlab är ett program med öppen källkod för redigering och bearbetning av ostrukturerade 3D trianguleringar. Systemet är skapat för att hjälpa till med behandlingen av röriga modeller som uppstår vid 3D-scanning. Meshlab tillhandahåller verktyg för att hjälpa till med behandlingen av de ostrukturerade modellerna som uppstår. Verktygen består av inspektion, rengöring, healing, rendering, redigering samt konvertering av de maskor som uppstår.
Meshlab togs fram av ett antal studenter år 2005 som en del av en FGT-kurs, Fondamenti di Grafica Tridimensionale, i datavetenskap på universitetet i Pisa. Under åren har olika studentgrupper fortsatt med arbetet och skapat fler och bättre funktioner samt en stabilare programvara. [5]

2.5 Catia

Catia (Computer aided three-dimensional interactive application) är utvecklat av Dassault system och är ett cad-programpaket. Mycket av det som människor i dag tar i och använder är skapat av tredimensionella-modelleringar i olika cadprogram. Catia erbjuder flera olika användbara verktyg så som mekanisk design där de tillhandahåller verktyg för att kunna skapa, ändra och validera olika typer av objekt. Allt ifrån mekaniska former till vanliga

ytor. Catia ger en fullt skalbar plattform för samarbete vid skapande av produkter och produktdatahantering. Catia ger även en möjlighet att integrera mekaniska-, elektriska- samt vätskesystem i tredimensionell för instruktions eller för olika experiment. Catia används idag i stor utsträckning på Volvo Cars i deras utveckling av nya bilar.[6]

2.5.1 Catia V5

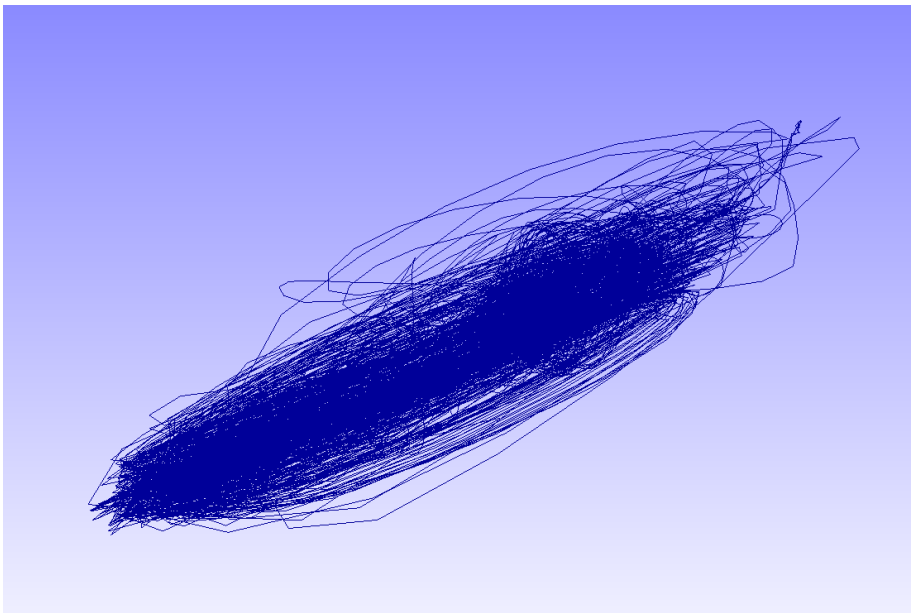
Catia V5 är enligt de själva en av de ledande 3D-mjukvarulösningarna som används för att utveckla olika produkter och används över hela världen. Catia V5 används bland annat för att konstruera bussen som transporterar människor, analysera mobiltelefonen i fickan eller simulera tangentbordet till datorn. Catia V5 är mycket användbart så väl i bilindustrin, flygindustrin, olika industrimaskiner som vid framtagningen av diverse konsumtionsvaror. Catia V5 kan även optimera produktens prestanda snabbare med hjälp av den integrerade designanalysen. [7]

2.6 Lua

Lua är ett programspråk avsett för att användas som ett enkelt men kraftfullt språk för att skripta. Lua är utformat och underhålls av ett team ifrån det katolska universitetet i Rio de Janeiro i Brasilien. Språket är idealiskt för konfiguration, skript och snabb prototypning då Lua besitter en automatisk minneshantering. Lua används ofta vid industriella tillämpningar så som Adobes Photoshop men även vid inbyggda system exempelvis digital-tv samt vid skapande av olika spel till mobiltelefoner och konsoler. Lua är idag det ledande skriptspråket vid konstruerandet av spel. [11]
I det här arbetet har Lua använts för att skripta funktioner till IPS flexible surface simulation och färdiga skript har dessutom använts för att utföra vissa moment. [14]
År 2011 vann Lua Front Line Awards för bästa programeringsverktyg av Game Developers Magazine. [12]

2.7 Rörelsefil

En rörelsefil består av ett rörelsemönster uppbyggt utav ett antal rotationer och translationer i rummet. Tanken med det här rörelsemönstret är att det ska efterlikna den rörelse en motor går igenom under en körning på Volvos Cars testbana i Hällered. Denna rörelsefil är inspelad i en frekvens om 200Hz och tillhandahölls via Volvo Cars. [15] I IPS visualiseras rörelsemönstret enligt figur 2.3.



Figur 2.3: Rörelsemönster vilket ska efterlikna en testkörning på Volvos testbana i Hällered.

3
Metod

I kaptilet beskrivs den arbetsgång som har använts i projektet.

3.1 Val av luftbälgar att simulera

Efter diskussion om vad som behövdes tog Volvo Cars fram tre luftbälgar, dessa tre luftbälgar är lanserade, ej sekretessbelagda och passade därför bra för ändamålet. Under arbetets gång tillkom ytterligare en bälg, då Volvo Cars stött på precis det problem som detta arbete behandlat. Volvo Cars använder idag fler luftbälgar än så, men på grund av sekretess och ständig utveckling får tester och prover ske på de luftbälgar som blivit tilldelade arbetet. Övriga data som tillhandahölls var materialdata, CAD- och rörelsefil samt en kortare genomgång i hur luftbälgarna idag simuleras som kablage.

3.2 Utbildning i IPS-programvaran

För att kunna förstå hur programmet fungerar, vilka indata som behövs och vilka möjligheter som finns krävs utbildning. FCC har för detta ändamål tillhandahållit utbildningsmaterial vilket har bidragit till förståelse för arbetet och programmet. Det verktyg som utbildningsmaterialet främst har gått igenom är Cable simulation. FCC håller på att ta fram en programvara för ytsimulering vid namn Flexible surface software till IPS. Denna programvara är ännu inte släppt på marknaden och därför finns det inte något utbildningsmaterial för detta verktyg mer än några enstaka interna PowerPoints.

3.2.1 Cable simulation

Utbildningsmaterialet i Cable simulation bestod av grunderna i programmet. Bland annat skapandet av kablar, placerandet av noder samt användningen utav rörelsefiler. Utöver detta gick det även igenom saker som möjligheten att kontrollera vridning, spänning och deformation på en kabel.

3.2.2 Flexible surface software

Det utbildningsmaterial som fanns tillhanda och arbetades fram under tiden, bestod av powerpoints som behandlade skript skrivna till verktyget. Avsaknaden av komplett utbildningsmaterial för Flexible surface software gjorde så att mycket tid avsattes för att förstå verktyget. Stor del av denna tiden bestod utav så kallade *trial and error*-försök där kunskap införskaffades om vad som är möjligt och ej.

3.3 Hur simulering utförs idag

Efter diskussion med Volvo Cars framkom det att de idag simulerar luftbälgar som vanligt kablage i IPS. Det finns en uttryckt önskan om att kunna göra detta på ett bättre sätt. Idag skapar de bara en kabel med liknande omkrets som luftbälgen.

3.4 Omvandla solid CAD-modell till en yta

Då programvaran för att simulera luftbälgarna endast har stöd för 2-dimensionella ytor krävs det att den ursprungliga CAD-modellen modifieras. Detta gjordes med de skript som finns att tillgå i IPS och Meshlab eller direkt via Catia.

3.4.1 Inner och ytteryta med hjälp av Catia

Vid användning av Catia plockar man in soliden i arbetsbänken Generative shape design. Nästa steg är att välja verktyget *extract* och under *extract definition* klicka i propagation type: Tangent continuity. Därefter är det möjligt att klicka i de ytor på luftbälgen som är intressanta. Välj ok för att extraherade ytorna. De ytor som misslyckades att plockas med Tangent continuity plockas i nästa omgång med propagation på samma sätt. Därefter väljs verktyget join i samma arbetsbänk och de extraherade elementen markeras i trädet för att bli ihopfogade. För guide med bilder se bilaga E.

3.4.2 Centrumyta från skript och Meshlab

Med hjälp av de lua-skript som finns att tillgå är det möjligt att skapa en centrumyta av en luftbälg. Här nedan följer en enkel stegguide:

1. Ladda fil och kontrollera mesh-kvalitén med *meshStatus*.
2. Generera en färgkodad luftbälg där luftbälgens olika delar visualiseras med olika färger. *colorConnectedSurface*.
3. Skapa en kabel som centrumlinje genom luftbälgen.
4. Kör *shootRaysToLocateSurfaceSide*, skriptet skjuter strålar från centrumlinjen för att lokalisera vilken sida av luftbälgen geometrierna tillhör.
5. Manuellt plocka de geometrier som hamnat fel och förflytta till rätt mapp.
6. Med hjälp av *centerSurface* skapa ett punktmoln som en .xyz-fil.
7. Öppna upp punktmolnet i Meshlab och använd filtret *Surface reconstruction: Poisson*, exportera sedan den skapade meshen som en .wrl och öppna med IPS.
8. Importera två plan och använd skripten *plotTriangleNormals* och *cutMeshAlongPlanes* för att få en luftbälg med hål igen.
9. Reducera antalet trianglar till önskat antal med *Simplify Triangle Mesh*-verktyget.

10. Kontrollera den nya meshen med *meshStatus*.

11. Kör ytterligare en gång *colorConnectedSurface* och flytta den geometri med flest trianglar till Static Geometry.

12. Ta bort oanvända vertices med skriptet *reIndexVertices*.

13. Kontrollera hål i meshen med *showMeshBorders*, markera lines och välj t.ex blå färg och öka storleken för att lättare kunna se.

14. Använd *mergeFromFrames* för att få de hörn som har samma koordinat men inte tillhör varandra att ihop sammanfogas.

15. Kör *showMeshBorders* igen, finns det fortfarande hål markeras de frames som tillhör hålet och läggs i mappen To be triangulated. Sedan körs skriptet *triangulateFromFrames*, detta steg behöver repeteras om hålen är flera.

16. Meshen är nu förhoppningsvis redo för att skapa en shell.

För fullständig och utförligare guide med bilder, se bilaga A samt B.

3.5 Triangelreducering utav mesh

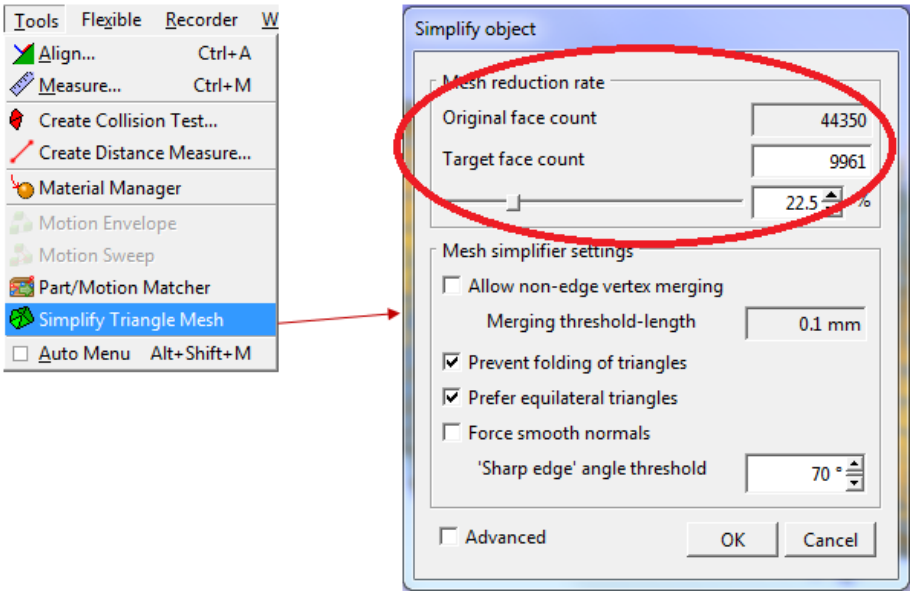
En yta är uppbyggd utav ett stort antal trianglar, ju fler trianglar desto finare yta. Nackdelen med många trianglar är dock att det blir ett väldigt stort antal beräkningar som krävs för att utföra olika simuleringar. Därför krävdes det en reducering av trianglar i modellen. För att göra denna reduceringen går det att använda en funktion som finns inbyggd i IPS kallad Simplify Triangle Mesh, där användaren får välja olika inparametrar för att utföra reduceringen. Max antal trianglar som testades var 10000, sedan kördes luftbälgar i fallande ordning med 5000, 2500 och 1250 trianglar för att se hur lågt det var möjligt att gå utan att tappa för mycket av orginalytans geometri. Se figur 3.1 för utförandet av Simplify Triangle Mesh.

3.6 Kontroll av ytor

Vid överföringen från solidmodell till ytmodell och även vid triangelreducering uppstår det ofta felaktigheter i meshen. För att kontrollera meshens status är det möjligt att använda ett skript kallat *meshStatus*. Skriptet ger viktig information om särskilt två parametrar:

- Hur många trianglar med noll grannar?
- Hur många vertex med noll trianglar?

Dessa bör båda vara noll för att vara en bra mesh. I annat fall behöver meshen modifieras. Skulle en triangel sakna grannar i meshen betyder det att det finns en lös triangel som inte hör hemma nånstans. Detsamma gäller vid de tillfällen då en vertex inte tillhör en triangel. Dessa behöver därför tas bort.



Figur 3.1: Simplify triangle mesh i IPS.

3.7 Modifiering utav dålig mesh

Då en mesh har flera trianglar med noll grannar eller ett flertal vertex utan trianglar behövs en modifiering utav meshen. Inget av dessa är något som eftersträvas och behöver därför tas bort. Vertexen som inte har några trianglar kan tas bort med skriptet reIndexVertices.lua och trianglarna utan grannar kan plockas bort i samband med att man använder sig av colorConnectedSurface.lua. Det finns också möjlighet att laga en mesh som inte håller ihop med skriptet mergeFromFrames-funkar.lua samt att skapa trianglar vid hål med triangulateFromFrames.lua.

3.8 Rörelsesimulering utav luftbälgar i IPS flexible surface software

Under arbetets gång tillhandahölls av FCC ett skript med en exempellösning på hur en simulering skulle kunna se ut. Detta skript har sedan modifierats och anpassats för att kunna köra andra luftbälgar. Det har även körts tester på både inner-, ytter-, samt centrumytor för att se vilken som lämpar sig bäst för simulering.
Modifieringen har skett genom att byta den scen som ska simuleras, längd av rörelsefilen, vilket objekt avstånd ska mätas mot samt vilken luftbälg som ska simuleras.

3.8.1 Nödvändigt antal trianglar

För att se vilket antal trianglar som är lämpligt både tidsmässigt och för att bibehålla hög noggrannhet i simuleringen kördes ett antal tester. Det antal trianglar som simulering kördes i, var i stigande ordning 1250, 2500, 5000 och 10000. Att 10000 trianglar valdes som max antal beror på att tiden vid simulering över 10000 trianglar hade blivit för lång.

Ett antal tester kördes dessutom med 5000 eller färre trianglar, detta då det blev enormt tidskrävande vid fler.

3.8.2 Val av centrum-, inner- eller ytteryta

De olika metoderna för att plocka ut en centrum-, inner- eller ytteryta provades för att se vilken metod som snabbast och enklast gav bäst resultat. Hänsyn togs också till fördelen centrumytan har i ytlösaren då geometrin blir mest lik originalet. Tester gjordes därefter där alla ytor kördes i samma simulering för att se hur graferna korrelerade med varandra.

3.9 Rapportskrivning och resultat

Under arbetets gång skrevs det en veckodagbok för att nu i slutet av arbetet enkelt kunna föra in uppgifter om hur arbetet gått till i rapporten. Arbetes avslutas med att presentera de resultat som har tagits fram, dels på Volvo Cars, Fraunhofer-Chalmers Centre samt på Chalmers.

3.9.1 Presentera resultat

Använda de resultat som fås från IPS för att sedan presentera de som grafer med hjälp av Matlab.

4
Resultat

I detta kapitel beskrivs de resultat som framkommit under arbetet.

4.1 Luftbälgar

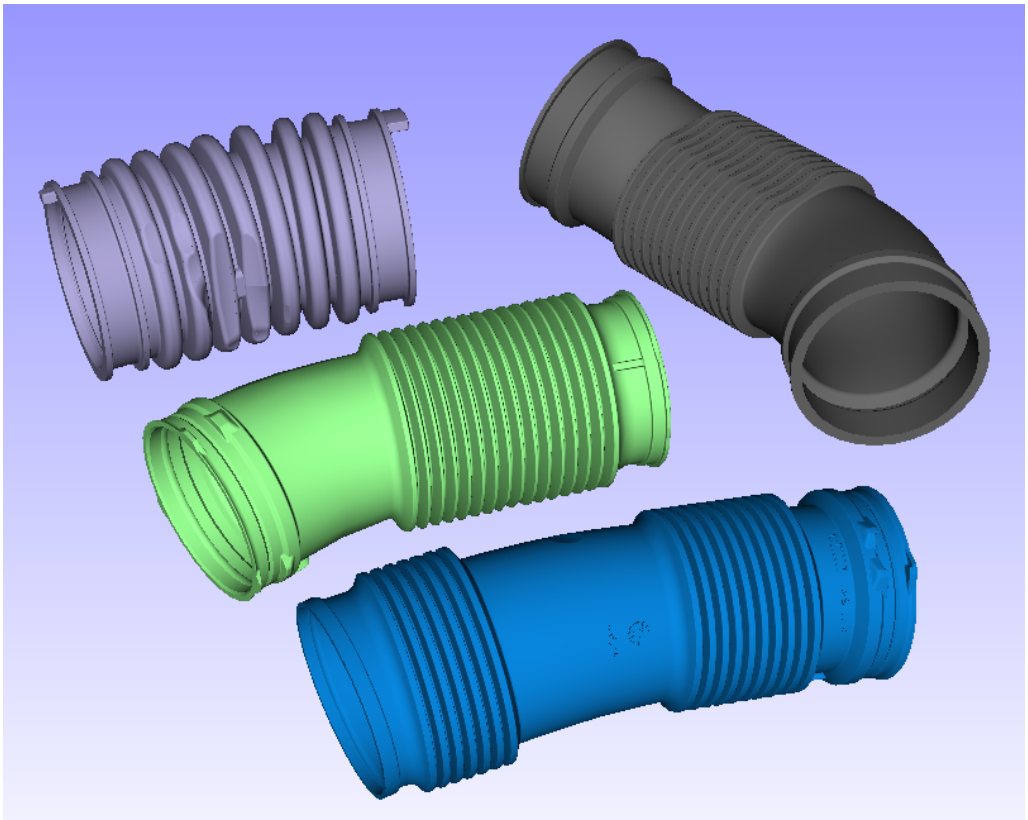
Idag används flera olika typer av luftbälgar, beroende på vad det är för motor som är monterad i bilen. Luftbälgarna har olika utformningar och består av olika material. Några luftbälgar består av plast och andra består av gummilegering. Ingen är den andra lik, dock är de ofta snarlika i designen då de flesta luftbälgar är uppbyggda med olika ribbor för att ge stabilitet och möjlighet till att böja sig. Trots att bälgarnas struktur ser väldigt olika ut varandra, simuleras de på samma sätt. De simuleras nämligen som helt enkla vanliga släta kablar. Detta betyder att simuleringen inte blir helt korrekt då luftbälgar i stort sätt alltid har en mer komplex yta än en kabel.
De luftbälgarna arbetet behandlat syns i figur 4.1 och är följande:

- **31370449** - Grå
- **31370111** - Grön
- **31338745** - Blå
- **31474241** - Svart

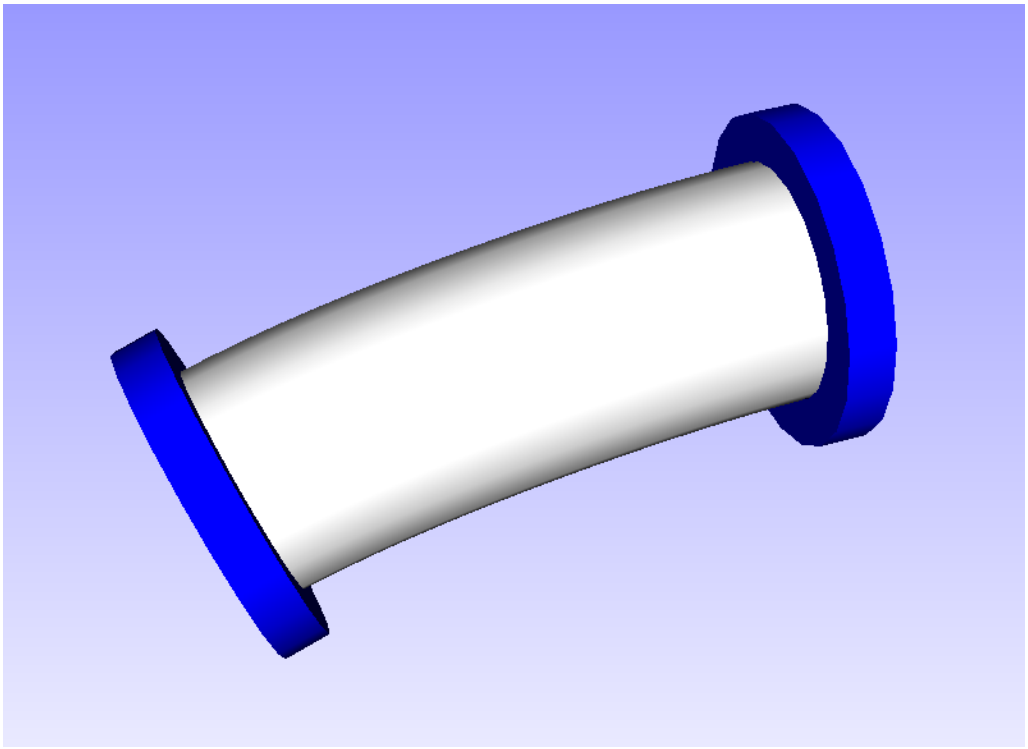
I figur 4.2 syns ett exempel på hur en kabel ser ut, för att kunna jämföra med bilden ovan.

4.2 Simulering av luftbälgar på Volvo Cars idag

Idag när Volvo Cars använder sig av IPS förlitar de sig helt på att simuleringen som är gjord stämmer. Detta ställer stort krav på att programvaran är pålitlig och att personen som gör simuleringen gör simuleringen rätt. Ytterligare ett problem som dyker upp är att den bakomliggande matematiken i Cable simulation bygger på att en kabel är lång och har ett i förhållande litet tvärsnitt jämfört med dess längd.[8] I fall där det har uppstått problem har man märkt det först efter att detaljen är producerad och monterad i en testbil. I dessa fall använder sig Volvo Cars av lermätningar. De sätter fast lera på detaljen som kommer i kläm under färd och kör en ny testrunda. Efter testrundan utvärderas resultatet för att se hur mycket av leran som är klämd för att på så sätt kunna göra en ny simulering där exempelvis kabeldragningen är flyttad så att detaljen inte längre hamnar i kläm. Skulle det vara handhavarfel vid simuleringstillfället gör de om detta för att se vad som gick snett.



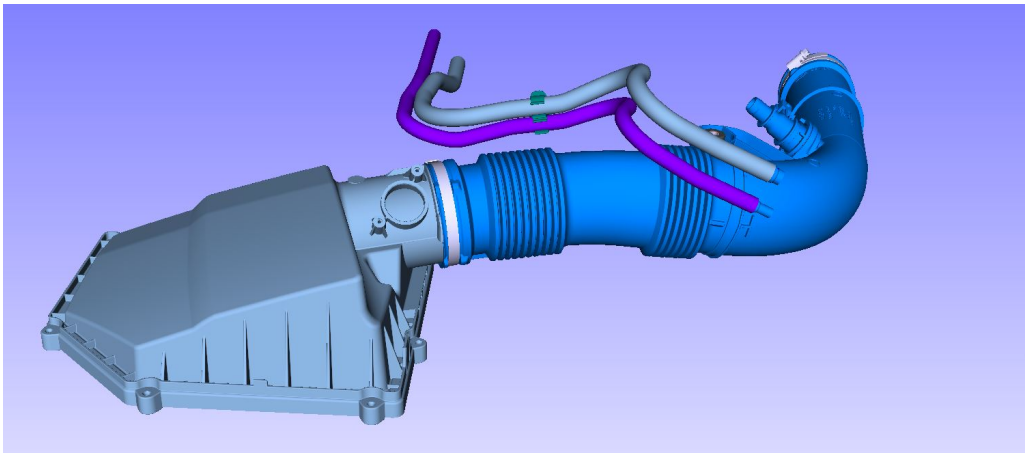
Figur 4.1: Alla luftbälgar öppnade i IPS.



Figur 4.2: Kablage skapat i IPS.

Att få ett fel är både tidskrävande och kostar företaget pengar, eftersom tidigare nämnt, att felen upptäckts i slutet av utvecklingen. Simulering är ett väldigt bra hjälpmedel vid just framställandet av dragning vid olika detaljer och kan ge en första anblick på var problem kan uppstå. Detta kan jämföras med hur det byggdes bilar innan datorns värld, där allt var tidskrävande, krävdes att mycket handbyggdes och testning skedde på detaljer som byggts samt att de oftast fick göras om innan det blev helt rätt. Detta leder till höga kostnader och förseningar som ett vinstdrivande företag givetvis vill undvika. Ytterligare en stor skillnad mellan gårdagens utveckling och dagens, är att i dagens motorer är utrymmet extremt begränsat. I dagens motorer ska fler detaljer samsas och med hybridernas intåg på marknaden ställer detta extra höga krav på att kunna packa sakerna tätt i motorutrymmet.

Fördelen med att använda simuleringen i IPS är att det skapas en snabb översikt på ifall kablarna, slangarna, luftbälgen eller vad det nu kan tänkas vara, som skall simuleras ifall de får plats eller inte. Se figur 4.3.



Figur 4.3: Kablage draget kring luftbälg.

4.3 Skapa ytor

Redan tidigt i arbetet framkom det att det skulle bli svårt att få fram ytorna från de komplexa tredimensionella luftbälgarna. Som tidigare nämnt klarar inte IPS-programmet av att räkna på 3D-objekt utan behöver en 2D-yta. Tre olika lösningsgångar arbetades fram och användes för att göra denna omvandling:

4.3.1 Yta framtagen i Catia V5

I arbetsbänken Generative shape design i Catia fanns det möjlighet att med hjälp av verktygen Extract och Heal enkelt plocka ut ytterytan på en luftbälg. Viktigt vid cylindriska luftbälgar är att använda Tangent continuity för att slippa klicka i varje sektion för sig, vilket är både tidsödande och gör det lätt att missa små sektioner.

4.3.2 Yta från skriptning i IPS och Meshlab

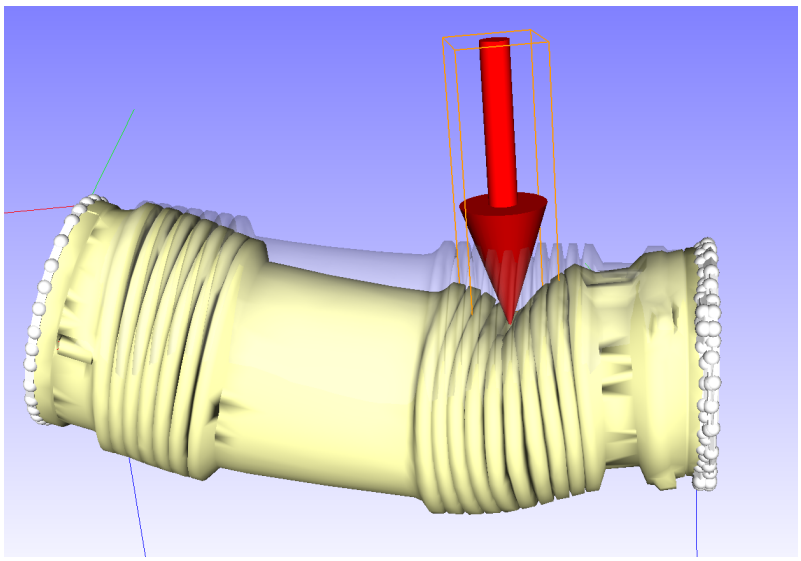
FCC tillhandahöll ett antal olika skript samt tog fram en powerpointguide för hur det var möjligt att med hjälp av dessa skript och Meshlab skapa centrumytor av luftbälgarna. Det är också möjligt att direkt via skripten och utan Meshlab att plocka någon av ytter- eller innerytorna med denna metoden. Fördelen med denna metod blir således att det är möjligt att få ut alla ytorna, nackdelen är att ett tredjeparts program, Meshlab behöver användas för centrumytan.

4.3.3 Yta framtagen i IPS från catPart

Via nya funktioner som ännu ej blivit lanserade, kunde IPS hantera riktiga Catia-filer vilket gjorde att man slipper exportera filer i speciella filformat från Catia.

4.4 Provsimulering

De första simuleringarna som gjordes provade bara funktionerna i programmet samt att ytan var tillräckligt bra för att kunna simulera. Randvillkor lades in och en ytkraft placerades på luftbälgen för att se så att ytan var hanterbar för ytlösaren. Detta gav svar på vad som krävdes för att skapa en yta lämplig för densamma. Denna yta syns i figur 4.4 där en ytkraft har placerats på luftbälgen. Detta för att se så att ytan höll ihop och inte hade några icke synliga sprickor i sig. Den skuggade detaljen som syns är samma luftbälg innan kraften är pålagd. Detta blev ett bevis för att skripten fungerade och gjorde det möjligt att börja fundera över hur rörelsesimulering skulle kunna ske.



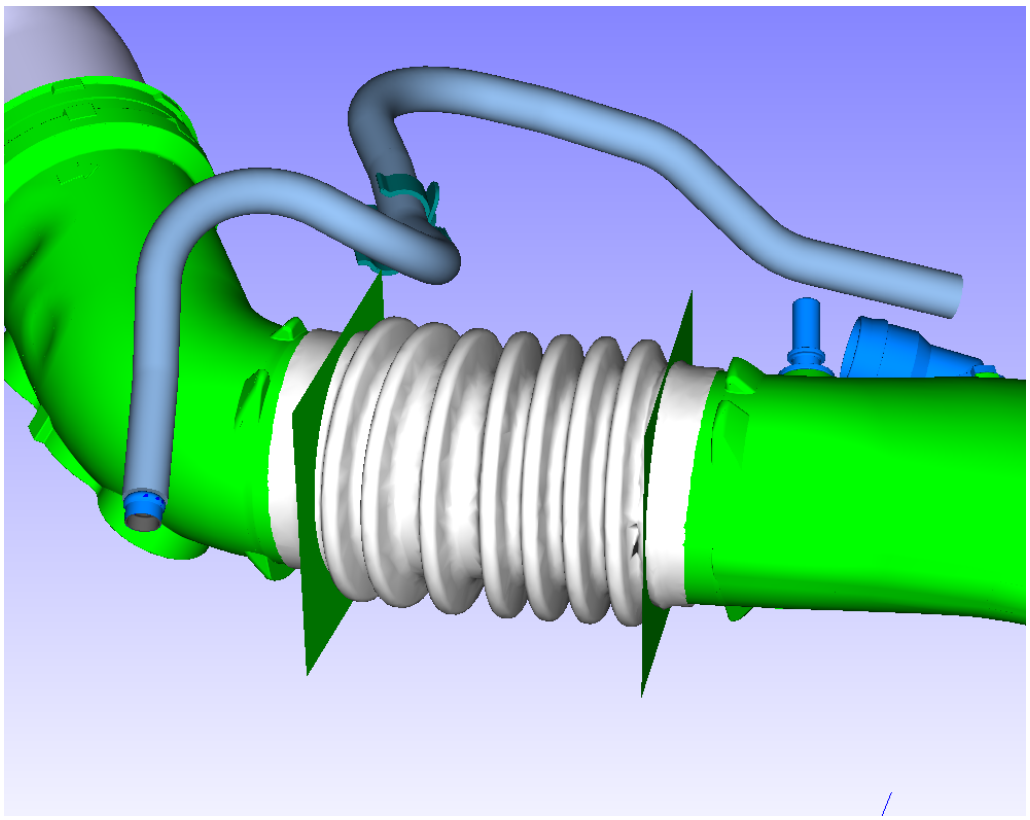
Figur 4.4: Provsimulering av luftbälg skapad som en yta.

4.5 Simulering med rörelse

Ett stort steg i arbetet gjordes efter att FCC skapat ett skript som gjorde det möjligt att translatera och rotera ett av randvillkoren efter ett på förhand bestämt rörelsemönster. Skriptet gjorde det även möjligt att vid varje steg i sekvensen mäta det kortaste avståndet från luftbälgen till en på förhand bestämd annan geometri i rummet. I dessa fall en av de kablar, eller plasthölje som låg i närheten.

4.6 Luftbälg 31370449

I figur 4.5 visualiseras en centrumyta av luftbälg 31370449 och även den kabel mot vilken avstånd mäts under simuleringarna. I figuren syns dessutom två gröna plan som bestämmer randvillkoren för luftbälgen. Luftbälgens högra randvillkor är det som sedan translaterats och roterats i enlighet med rörelsefilen.



Figur 4.5: Luftbälg 31370449 och kabeln vilken avstånd mäts mot.

4.6.1 Material

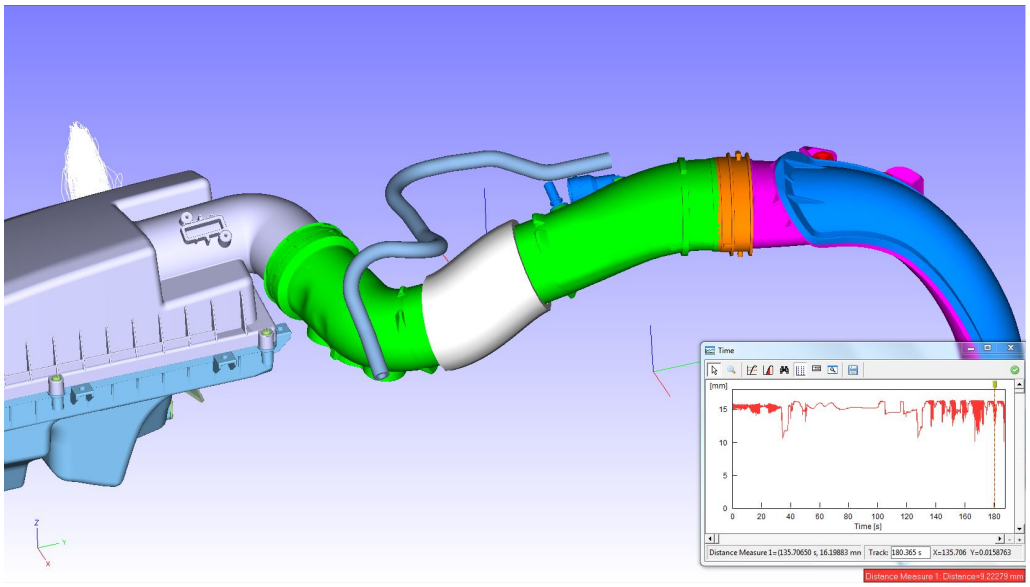
Luftbälg 31370449 består av en gummi kallad EPDM som har följande parametrar:

- **E-modul** - c:a 4,5 MPa
- **Poissons konstant** - 0,5

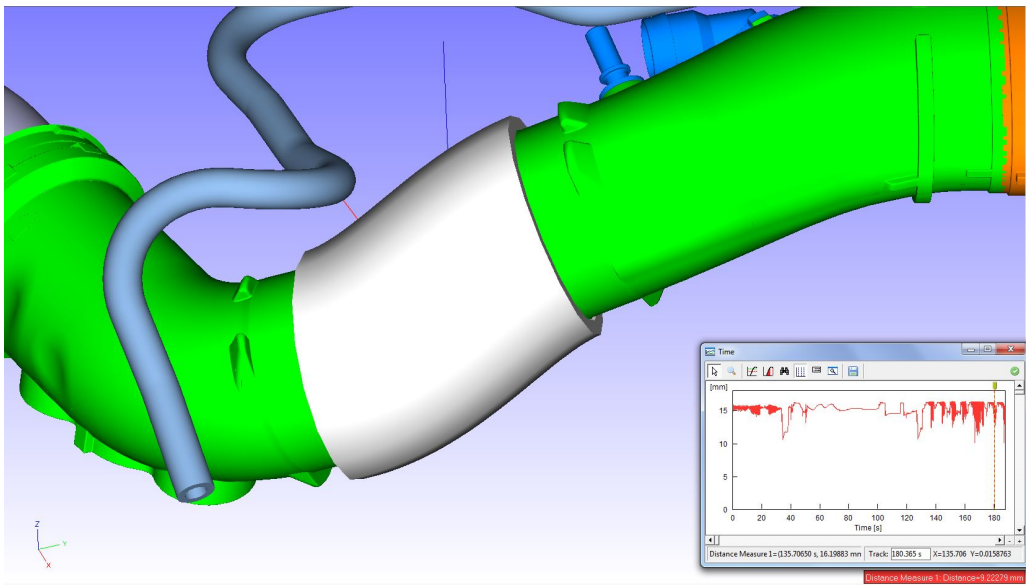
- **Densitet** - c:a 1180 kg/m^3

4.6.2 Luftbälg simulerad som kabel

I figuren 4.6 ser vi hur luftbälg 31370449 är simulerad som en kabel i IPS. Ett måttavstånd är satt mellan luftbälgen och kabeln ovanför. Avståndet är satt så att det kortaste avståndet alltid skall mätas mellan kabeln ovanför och luftbälgen under hela rörelseanalysen.



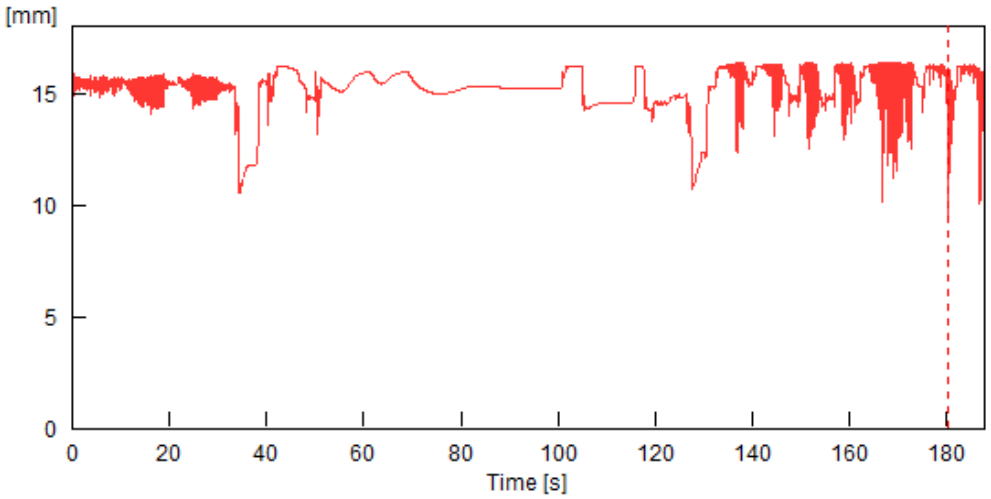
Figur 4.6: Luftbälgen 31370449 simulerad som kabel.



Figur 4.7: Avståndsmätning mellan luftbälgen 31370449 och ovanliggande kablage.

4.6.3 Resultat av luftbälg simulerad som en kabel

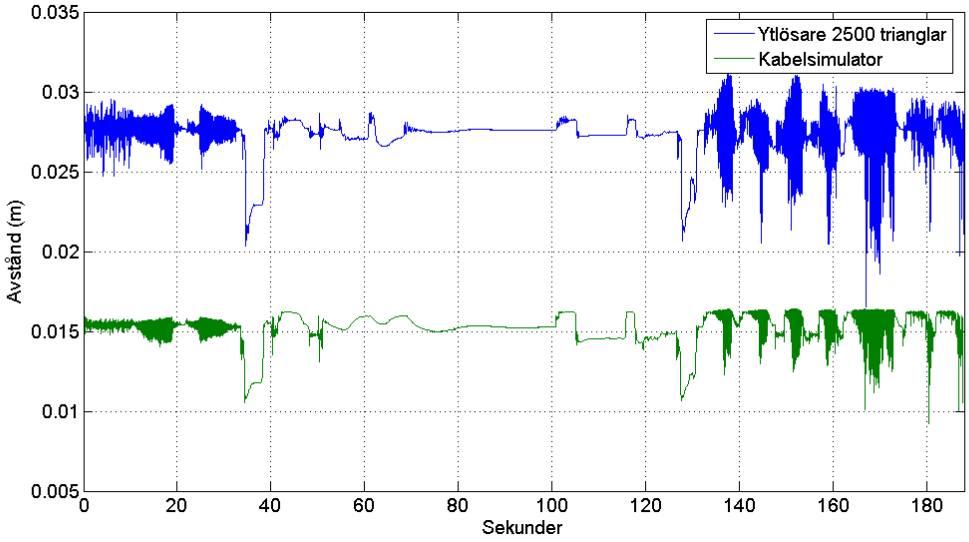
Om vi ser i figuren 4.7 är luftbälgen simulerad som en kabel, alla ribbor är borta och den har samma tjocklek över hela ytan. Här är det minsta avståndet till kabeln som mäts på 9,22mm (se figur 4.8) vilket kan jämföras med figur 4.9, där vi får ett större avstånd. Intressant här är att det är vid samma tid i sekvensen de båda simuleringarna fått sitt minsta avstånd till kabeln. Detta visar på att simuleringar där man skapar en luftbälg som en kabel får en stor förskjutning i avstånd, beroende på hur den är skapad. Vilket betyder att om det är trångt mellan olika delar i motorn kan kabelsimuleringen ta i, eller inte ta i, även om så inte vore fallet vid en luftbälg. Detta är väldigt beroende på hur kabeln skapas då ytgeometrin saknas och därför blir det en approximerad simulering.



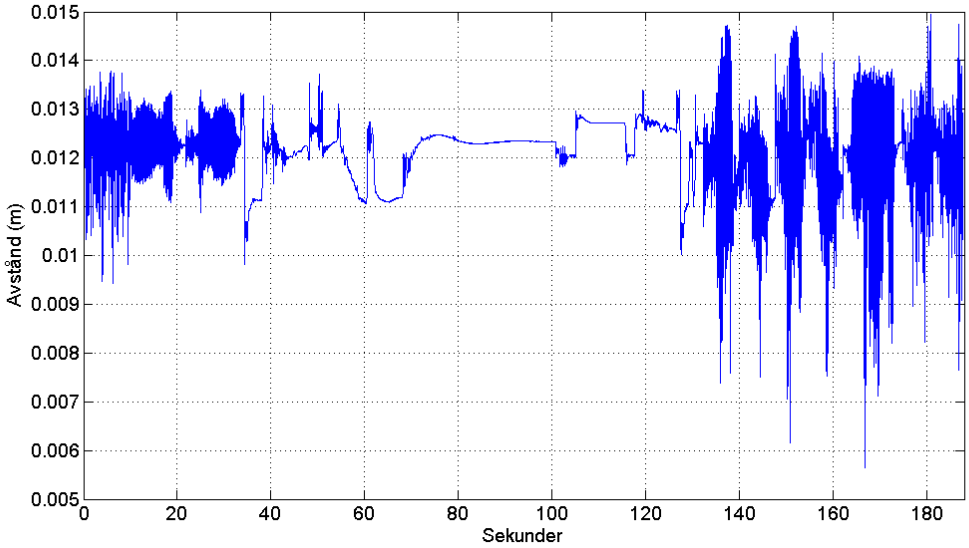
Figur 4.8: Resultat av luftbälgen 31370449 simulerad som kabel.

4.6.4 Jämförelse kabelsimulering och ytsimulering av centrumyta

I figur 4.9 och figur 4.10 visas skillnaden mellan de två simuleringarna. I den senare figuren skulle grafen varit helt rak längs med x-axeln om avstånden korrelerat.



Figur 4.9: Grafer för jämförelse mellan kabelsimulering och ytlösare.

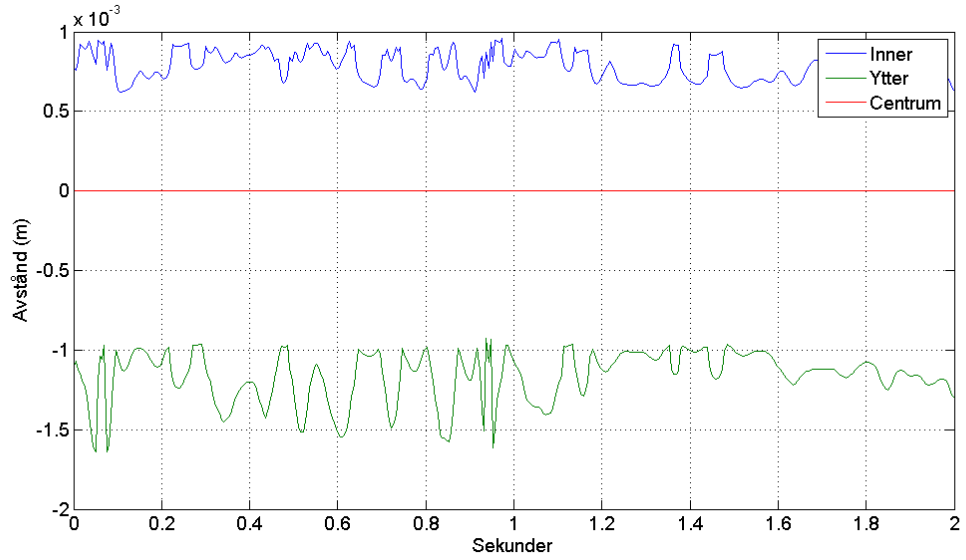


Figur 4.10: Skillnad i avstånd mellan grafer från figur 4.9.

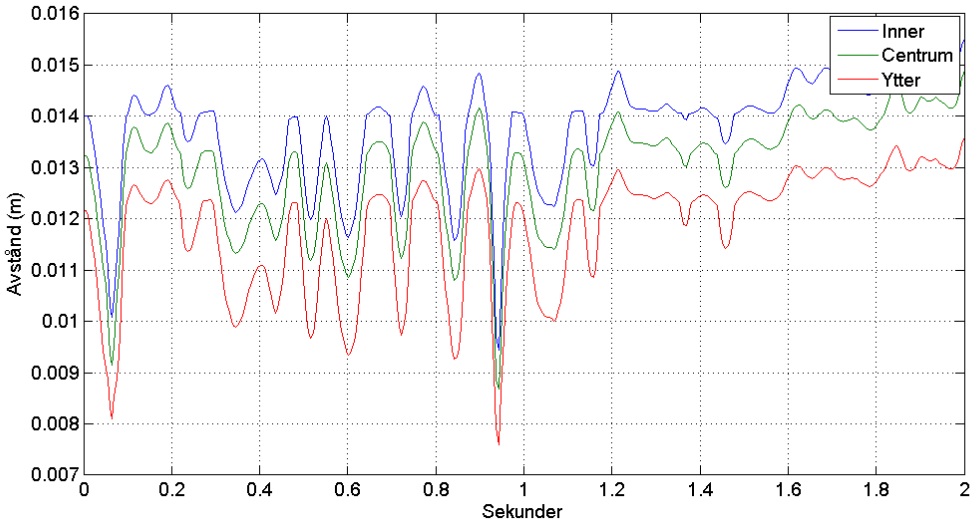
4.6.5 Avståndsskillnad beroende på yta

I den här simuleringen kördes luftbälg 31370449 med en skapad centrumyta, inner samt ytteryta. Skillnaden i avstånd från centrumytan finns plottad i figur 4.11. Då luftbälgen dels har en varierande tjocklek, samt att centrumytan är skapad från ett punktmoln mellan de båda andra ytorna är det i figuren möjligt att se hur stor skillnaden i avstånd blir under

en del av sekvensen. I figur 4.12 är varje ytas avstånd till närliggande kabel plottat. I detta fall ser vi att centrumytan ligger ganska bra till mellan de två andra ytorna, om än något förskjuten mot innerytan. Detta skulle exempelvis kunna bero på hur centrumytan har återskapats från ett punktmoln, bälgens geometri eller vilka inställningar som använts.



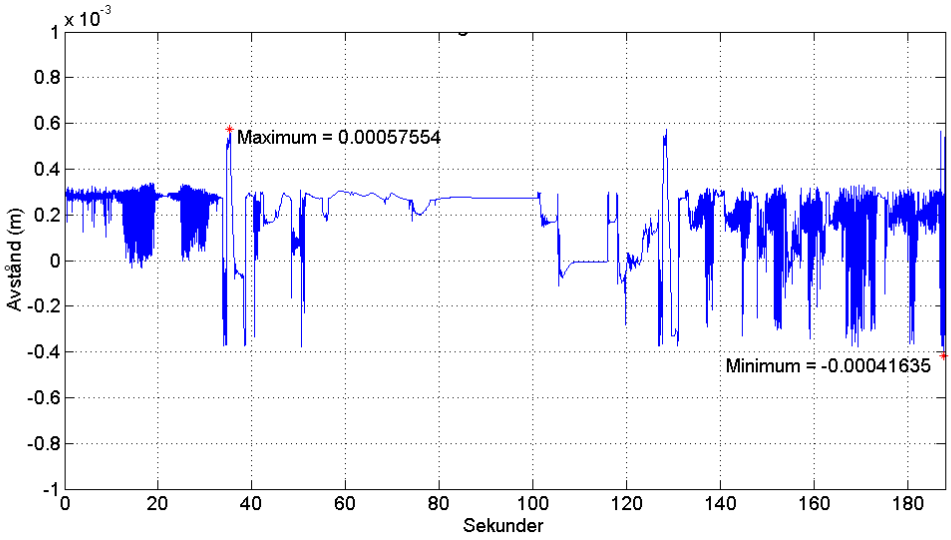
Figur 4.11: Avståndsskillnad från centrumytan till ytter och inner.



Figur 4.12: Avstånd till närliggande kabel för inner, centrum samt ytteryta.

4.6.6 Avståndsskillnad beroende på antal trianglar

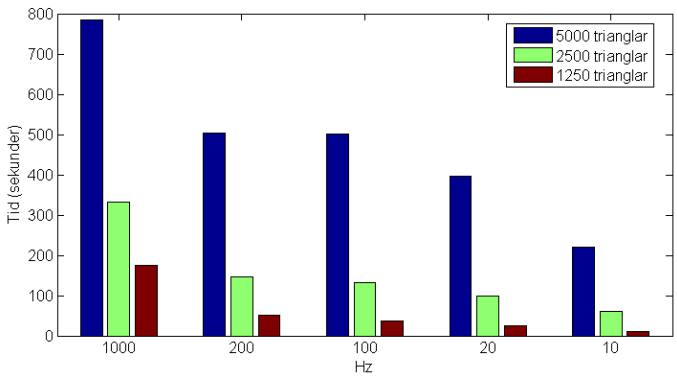
I figur 4.13 är skillnaden i avstånd på luftbälgen uppbyggd med 1250 trianglar och 2500 trianglar över hela sekvensen på 188 sekunder plottad.



Figur 4.13: Skillnad i avstånd beroende på antal trianglar.

4.6.7 Simuleringstid beroende på antal trianglar

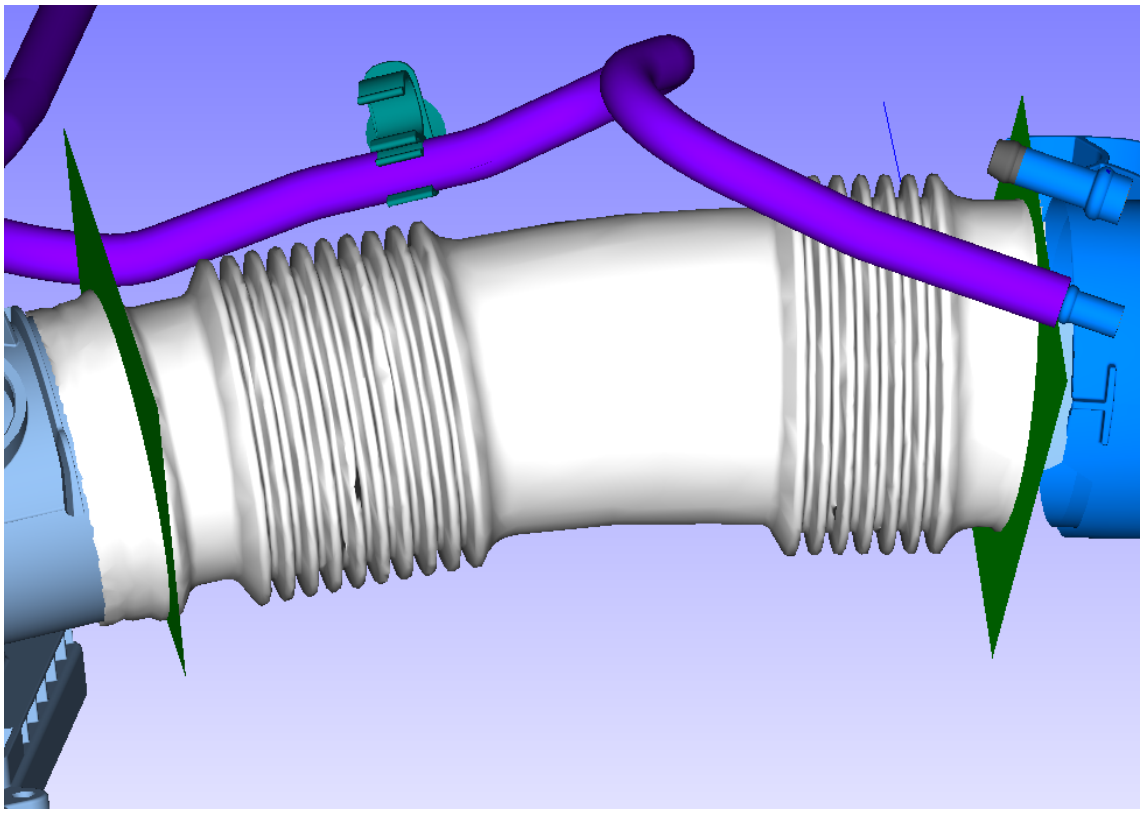
För att kunna se hur lång tid det tog att köra en simulering vid olika antal frekvenser samt olika antal trianglar, kördes samtliga två sekunder av hela sekvensen. Med frekvens menas antalet svängningar per sekund. Därav är det möjligt att använda olika antal steg för att uppnå olika frekvenser, men högre frekvens än 200Hz är onödigt då rörelsefilen inte är inspelad i högre än så. De data som går att avläsa i figur 4.14 är att det snabbt går åt betydligt mer tid när antalet trianglar ökar. Det går nästan att säga att tiden ökar exponentiellt, därmed tar en längre simulering enormt mycket mer tid vid högre antal trianglar.



Figur 4.14: Simuleringstid för olika antal trianglar och steglängd.

4.7 Luftbälg 31338745

I figur 4.15 är ytan av luftbälg 31338745 visualiserad i IPS i den scen den blivit simulerad. Den lila kabeln som går ovan luftbälgen är den kabel som sedan avståndet under simuleringen mäts mot. De gröna plan som syns i figuren är de plan som bestämmer var randvillkoren på luftbälgen är placerade.



Figur 4.15: Luftbälg 31338745 och kabeln som avståndet mäts emellan.

4.7.1 Material

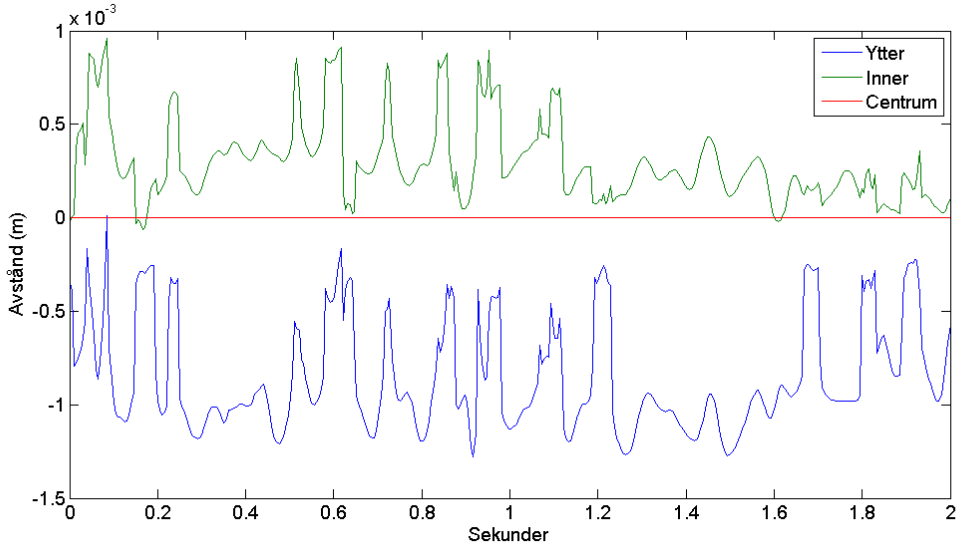
Luftbälg 31338745 består av en plast kallad PA6 (PA SOFT) Grilon ELX H50 HNZ. Denna plast har följande parametrar:

- **E-modul** - 150-220 MPa (Simulering har skett vid 185 Mpa.)
- **Poissons konstant** - 0,35
- **Densitet** - 1010 kg/m^3

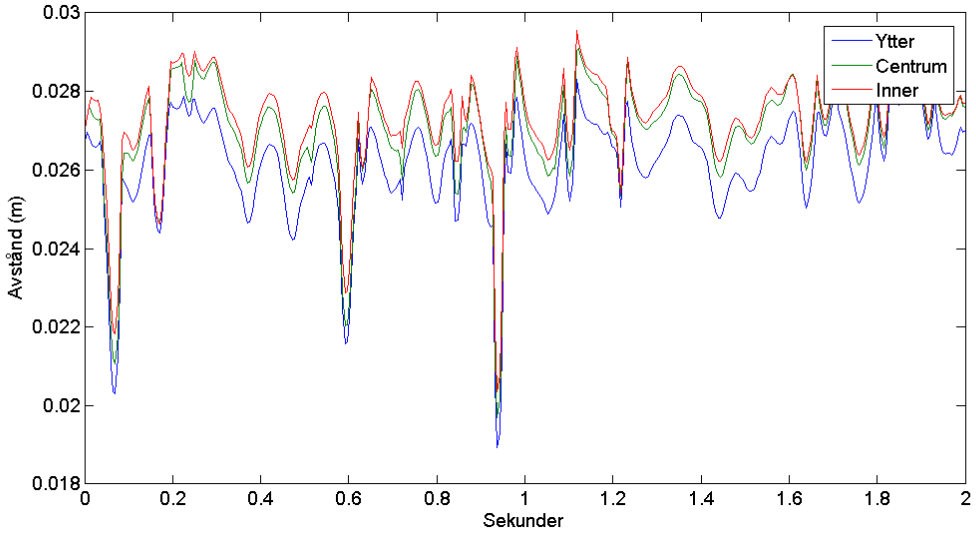
4.7.2 Avståndsskillnad beroende på yta

I den här simuleringen kördes luftbälg 31338745 med en skapad centrum-, inner- samt ytteryta. Skillnaden i avstånd från centrumytan finns plottad i figur 4.16. Då den här luftbälgen dels har en varierande tjocklek, samt att centrumytan är skapad från ett punktmoln

mellan de båda andra ytorna är det i figuren möjligt att se hur stor skillnaden i avstånd blir under en del av sekvensen. I figur 4.17 är varje ytas avstånd till närliggande kabel plottat. I det här fallet ser man tydligt i de båda figurerna att ytan skapad från punktmolnet har hamnat betydligt närmre luftbälgens inneryta än vad som är önskat. Detta kan ha lite olika förklaringar, dels inställningar vid skapandet av punktmoln, hur Meshlab återskapar en yta från ett punktmoln eller möjligtvis luftbälgens geometri.



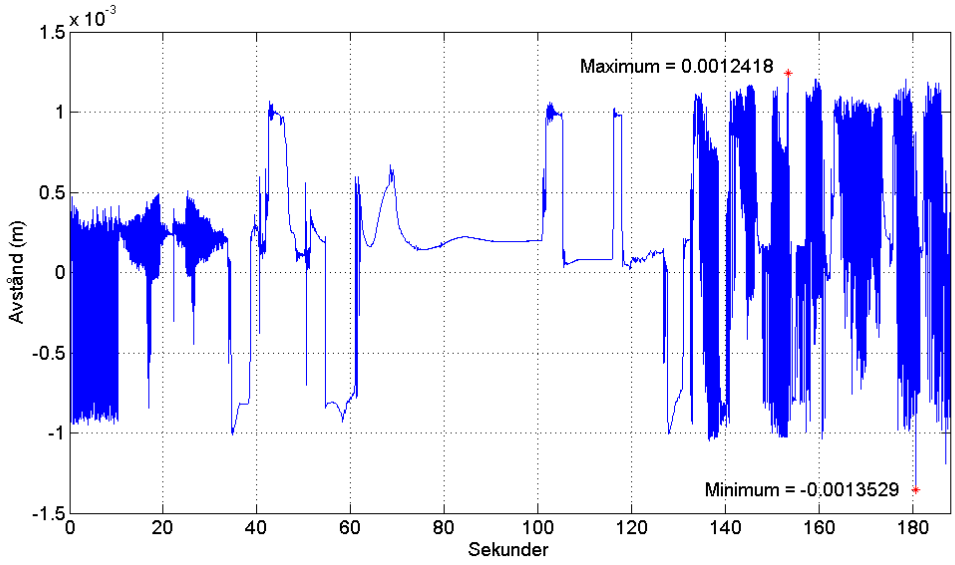
Figur 4.16: Avståndsskillnad från centrumytan till ytter och inner.



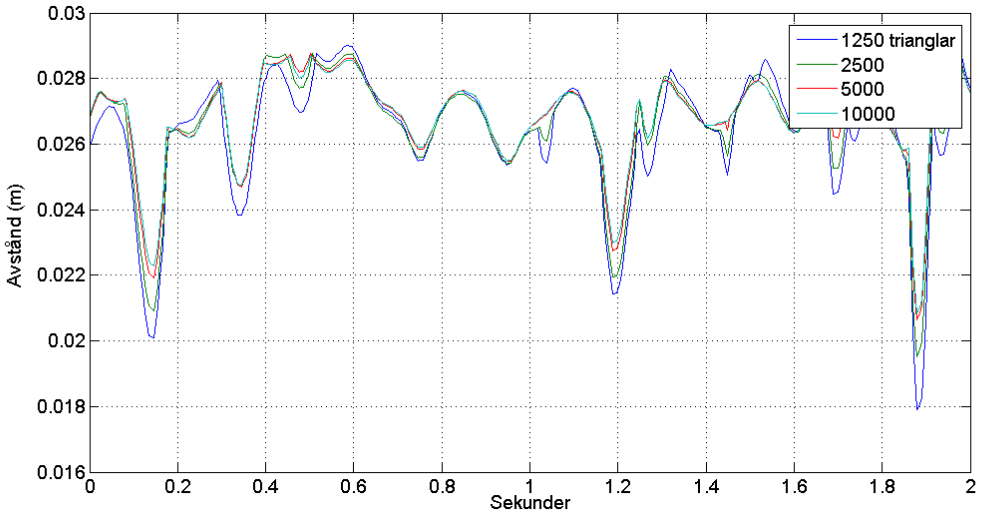
Figur 4.17: Avstånd till närliggande kabel för inner, centrum samt ytteryta.

4.7.3 Avståndsskillnad beroende på antal trianglar

I figur 4.18 är skillnaden i avstånd på luftbälgen uppbyggd med 1250 trianglar och 2500 trianglar över hela sekvensen på 188 sekunder plottad. I figur 4.19 är skillnaden mellan 1250 och 10000 trianglar nästan 3 mm. I övrigt följer särskilt graferna för 10000 och 5000 trianglar varandra åt bra. Båda figurerna visar resultat av centrumytan men är uppbyggda av olika antal trianglar.



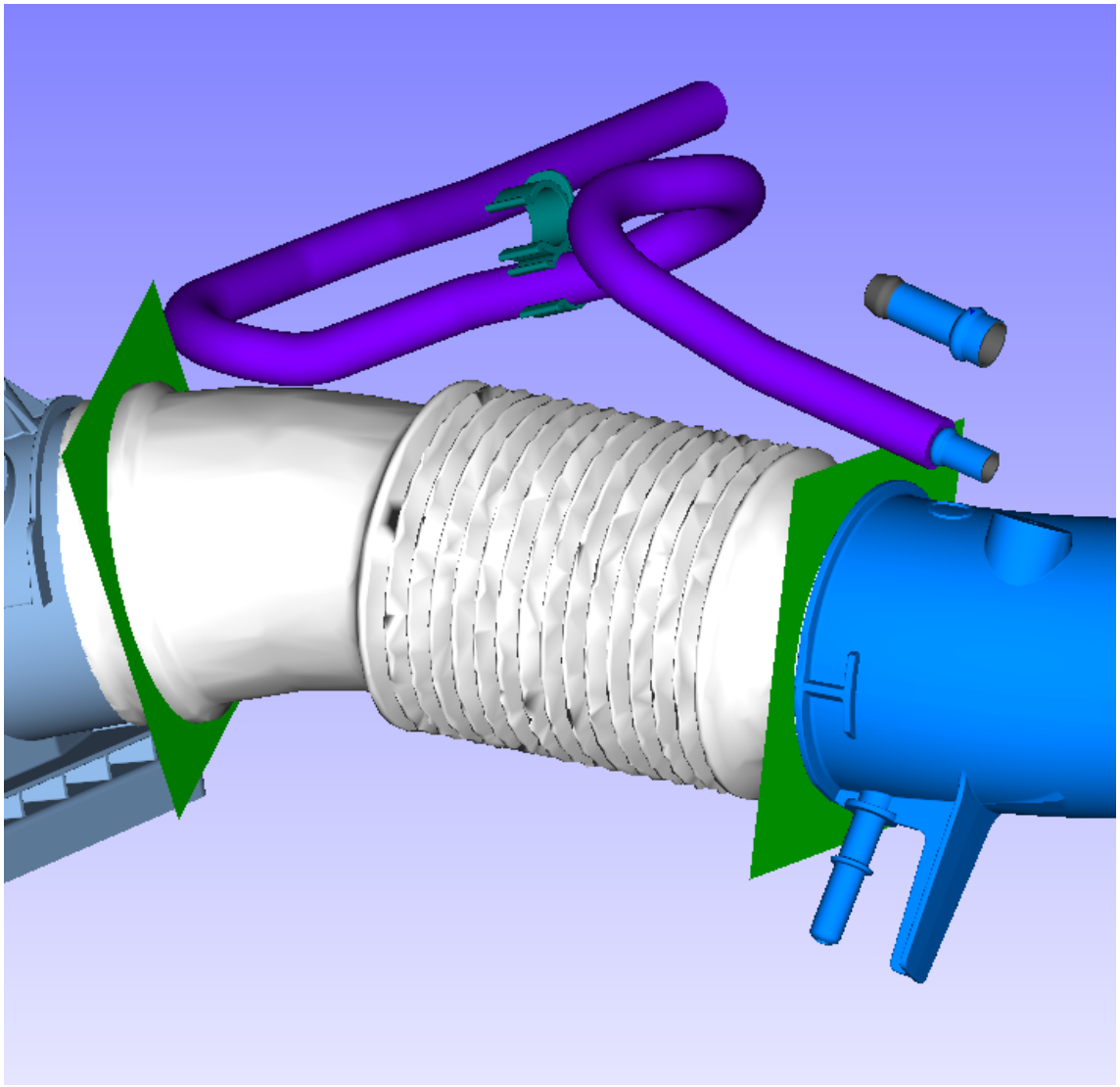
Figur 4.18: Skillnad i avstånd beroende på antal trianglar. (centrumyta)



Figur 4.19: Skillnad i avstånd beroende på antal trianglar (centrumyta).

4.8 Luftbälg 31370111

I figur 4.20 är ytan av luftbälg 31370111 visualiserad och även den kabel som senare avståndet mäts mot under simuleringen. De gröna planen som syns i figuren är de plan som bestämmer var randvillkoren på luftbälgen har placerats och det är det högra randvillkoret som sedan följer motorns rörelser.



Figur 4.20: Luftbälg 31370111 och kabeln, vilken avstånd mäts mot.

4.8.1 Material

Luftbälg 31370111 består av en plast kallad PA6 (PA SOFT) Grilon ELX H50 HNZ. Denna plast har följande parametrar:

- **E-modul** - 150-220 MPa (Simulering har skett vid 185 Mpa.)
- **Poissons konstant** - 0,35

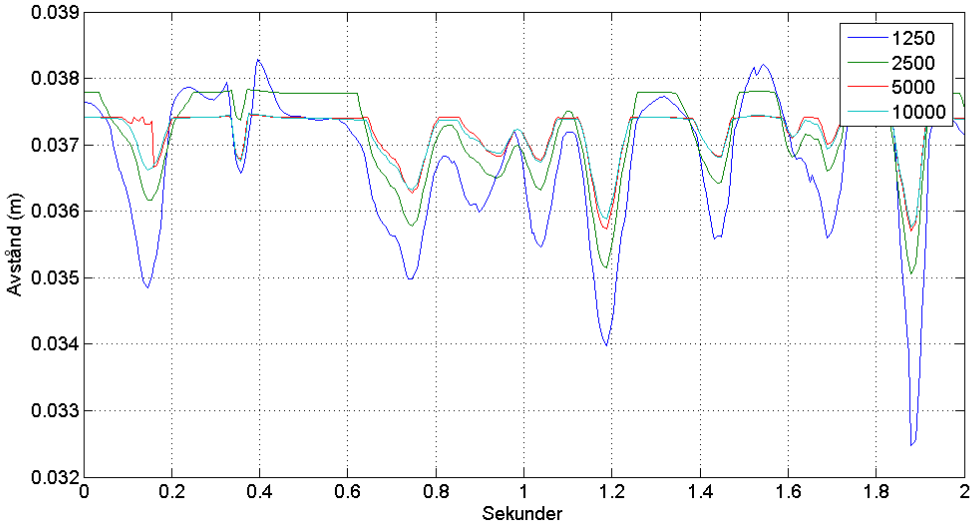
- **Densitet** - 1010 kg/m^3

4.8.2 Avståndsskillnad beroende på yta

Denna luftbälg simulerades aldrig med alla ytor då övergång från solid till yta misslyckades då IPS av någon diffus anledning inte kunder hantera den ytteryta som skapades av luftbälgen.

4.8.3 Avståndsskillnad beroende på antal trianglar

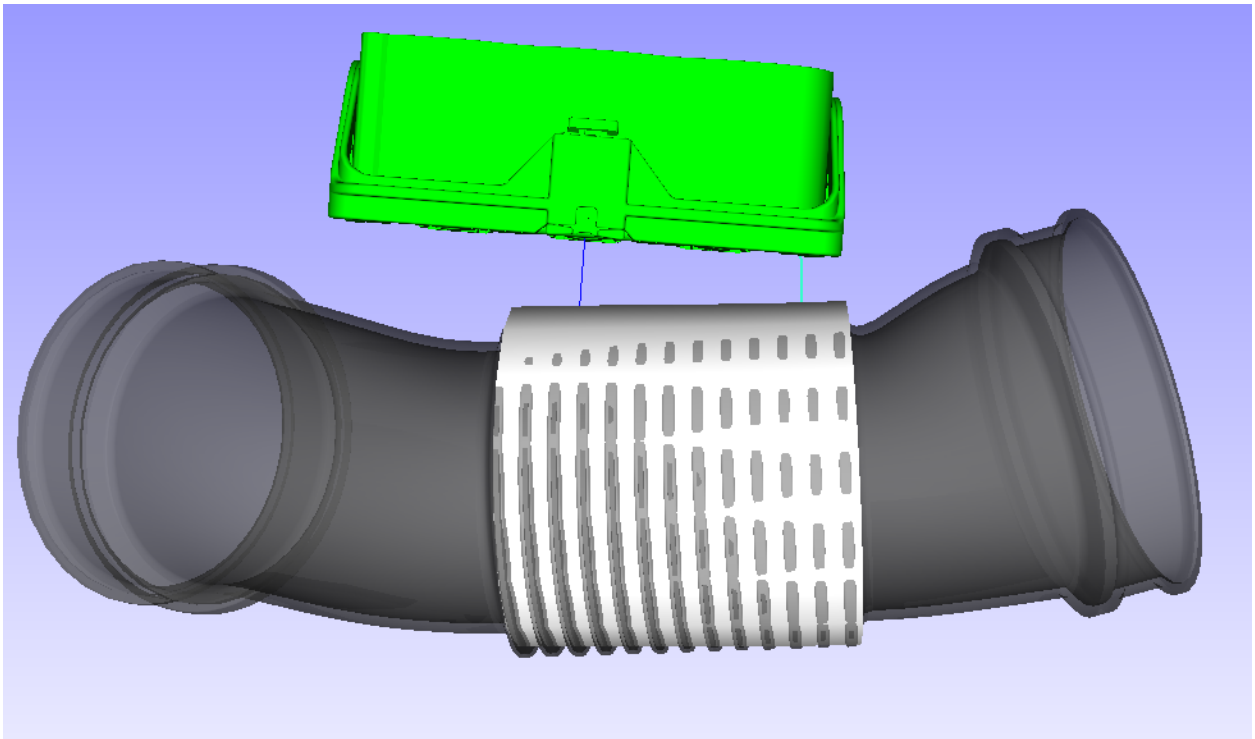
I figur 4.21 är luftbälgen simulerad under 2 sekunder av rörelsefilen. Det som går att utläsa ur figuren är att det i denna skiljer betydligt i avstånd mellan antal trianglar. Största skillnaden mellan 1250 och 10000 trianglar är c:a 3 mm i avstånd.



Figur 4.21: Skillnad i avstånd beroende på antal trianglar (centrumyta).

4.9 Luftbälg 31474241

På luftbälgen har endast en centrumyta skapats och simulerats. Stora skillnaden i denna luftbälgs simulering mot de tidigare är att här har hänsyn tagits till olika materialtjocklek. Luftbälgen tillkom under arbetets slutfas då Volvo Cars hade ett fall där Flexible Surface software hade passat bra att använda. I figur 4.22 ses luftbälgen simulerad som kablage i IPS. Den svarta ytan är hur luftbälgen ser ut och genomlyses. Den gråa ytan är simuleringen gjord som kablage.



Figur 4.22: Hur luftbälg 31474241 simulerats som kabel.

4.9.1 Material

Materialet för denna luftbälg har antagits till samma som de föregående i plast - PA6 (PA SOFT) Grilon ELX H50 HNZ.

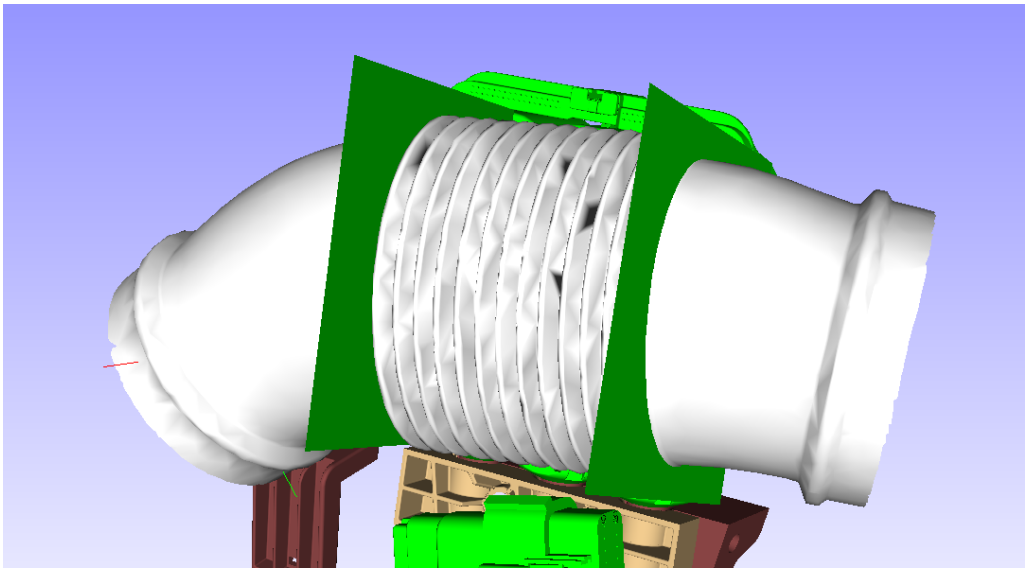
- **E-modul** - 150-220 MPa (Simulering har skett vid 185 Mpa.)
- **Poissons konstant** - 0,35
- **Densitet** - 1010 kg/m^3

4.9.2 Simulering av luftbälg med varierande tjocklek

För att göra det möjligt vid de luftbälgar som inte har samma tjocklek längs med modellen krävs det ett sätt för att komma runt det problemet. Lösningen på detta kom till genom skript och användandet av plan i själva scenen. I figur 4.23 visas ett sådant fall. Här har luftbälgen en tjocklek som varierar mellan 2,2 mm och 0,84 mm. Luftbälgen är tjockare i båda ändarna och i den räfflade delen mellan planen är den smalare för att där kunna bukta. Då luftbälgen är uppbyggd av ett stort antal trianglar användes planen som avdelare för att visa var ytan skulle ha olika tjocklek. Med hjälp av de olika skripten byggdes sedan en luftbälg upp på dessa villkor.

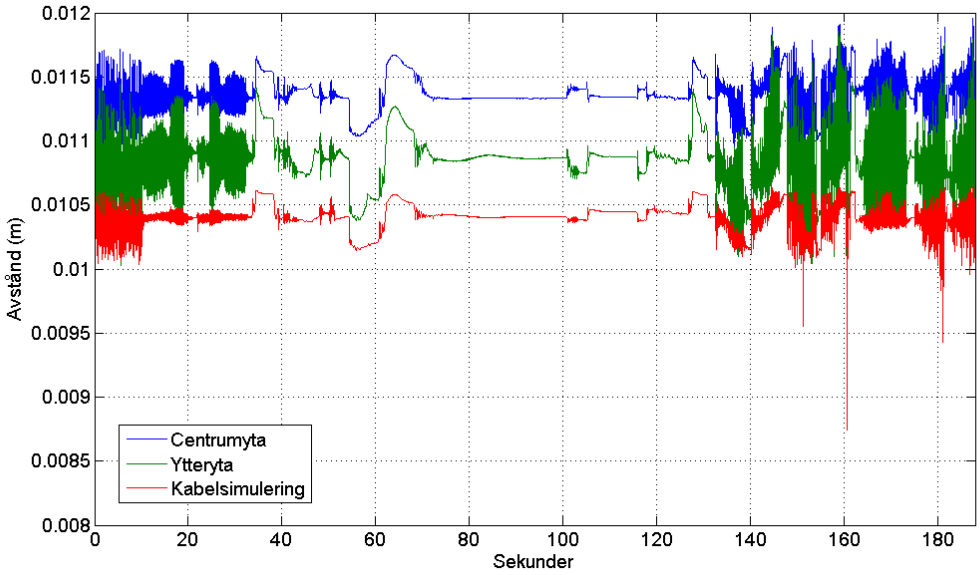
4.9.3 Jämförelse kabelsimulering och ytsimulering

I figur 4.24 visas avståndet till plasthöljet för en skapad centrumyta, ytterytan och en kabelsimulering gjord av Volvo Cars. I graferna för centrum- och ytteryta är inte någon

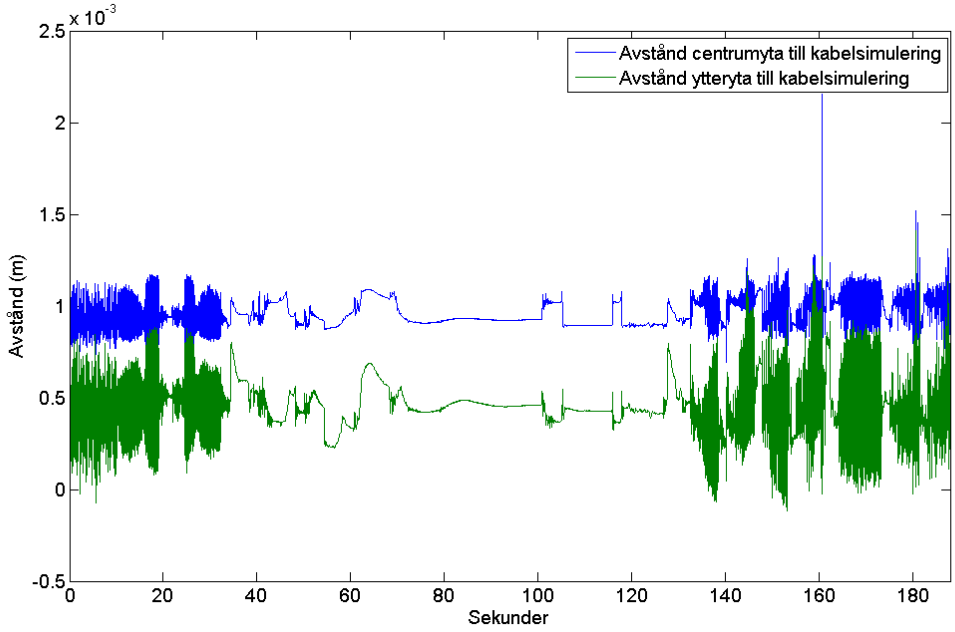


Figur 4.23: Luftbälg uppsatt med plan för att kunna simuleras med varierande tjocklek.

materialtjocklek pålagd. Det som är intressant i figuren är hur de två ytorna simulerade i ytlösaren skiljer sig åt från kabelsimuleringen. Bland annat skiljer det över 1.5mm i minsta värde vid ytterytans simulering. På den ytan bör inte någon materialtjocklek läggas på ytterligare, då det är från den yttre ytan av bälgen som sedan simulerats. Även om materialtjocklek skulle plussats på i det här fallet, är det inte mer än c:a 0,42mm extra som läggs till. I figur 4.25 visas hur avståndet från kabelsimuleringen till de två ytorna som simulerats i ytlösaren ändras under rörelsesekvensen.



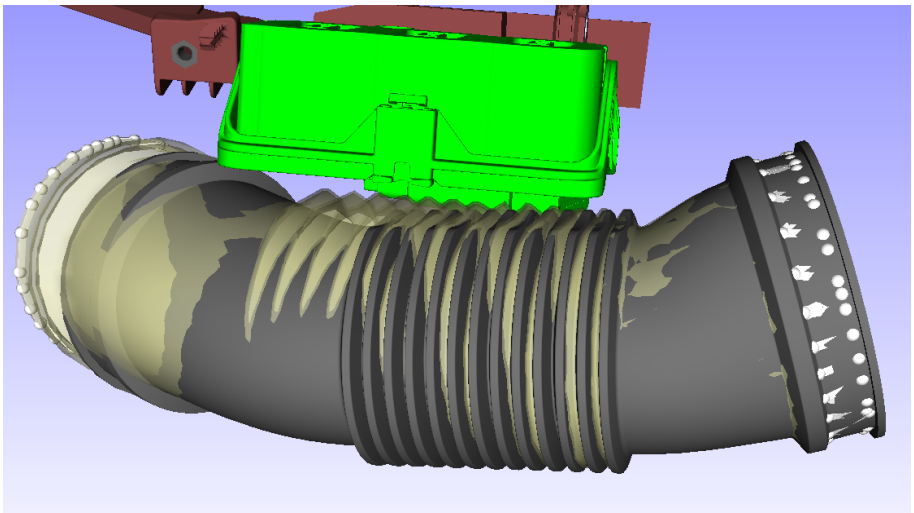
Figur 4.24: Grafer för jämförelse mellan kabelsimulator och ytlösare.



Figur 4.25: Skillnad i avstånd mellan kabelsimulering och grafer från figur 4.25.

4.9.4 Luftbälg vid minsta avstånd till objekt jämfört med original

I figur 4.26 ser vi luftbälg 31474241 som vanlig solid. Över denna solid ligger dessutom den deformerade luftbälgen när den hämtats från den tid i sekvensen där avståndet är som kortast till närliggande objekt. Denna deformerade geometri visar tydligt hur stor skillnad det är mot originalgeometrin. Detta fall visar också tydligt på hur viktigt det är att få med urgröpningar och liknande i geometrin samt hur den varierande tjockleken påverkar. Detta syns då den del med räfflor är betydligt längre på den deformerade, eftersom det också är där den har en mindre tjocklek.



Figur 4.26: Deformerad luftbälg 31474241 något transparant i samma scen som ej deformerad luftbälg 31474241.

5
Diskussion

I kapitlet diskuteras det kring resultaten från föregående kapitel.

5.1 Vilken av ytorna lämpar sig bäst för simulering

Då ytlösaren i IPS tar emot en yta utan tjocklek och sedan bygger den utefter vilken tjocklek användaren ställer in, lämpar sig centrumytan bäst för ändamålet. Både inner och ytterytan skapar en förskjutning i bälgens dimensioner, antingen blir den något större än normalt eller så blir den något mindre. Skapas en centrumyta av luftbälgen så blir den avvikande geometrin som minst vid en luftbälg med jämn tjocklek av geometrin. Den fördel som ytterytan möjligtvis skulle kunna ha är det att där fås en inbyggd tolerans mot fel, då ytan lägger till material där det normalt inte finns något. Detta skapar därför en extra säkerhetszon. Det som är viktigt vid skapandet av centrumytan är att kontrollera hur den förhåller sig till de andra två, skulle det finnas en förskjutning åt något håll gäller det att antingen skapa en ny med andra inställningar eller att kompensera för förskjutningen i simuleringen.

5.2 Lämpligt antal trianglar vid simulering

Antalet nödvändiga trianglar vid simulering kan skilja sig åt något beroende på geometri. De luftbälgar som vi behandlat i detta arbete har alla klarat av en nedskalning till 2500 trianglar utan att få allt för stora svängningar i avstånd mot högre antal trianglar. Vi har också märkt att när man ökar antalet trianglar ökar sannolikheten för att simuleringen kraschar samt att tiden det tar att simulera blir enormt mycket längre. Rekommendationen blir därför att hålla sig runt 2500 trianglar och jämföra ytan med orginalmodellen visuellt för att se om geometrin blir allt för dålig vid nedskalning av antalet trianglar.

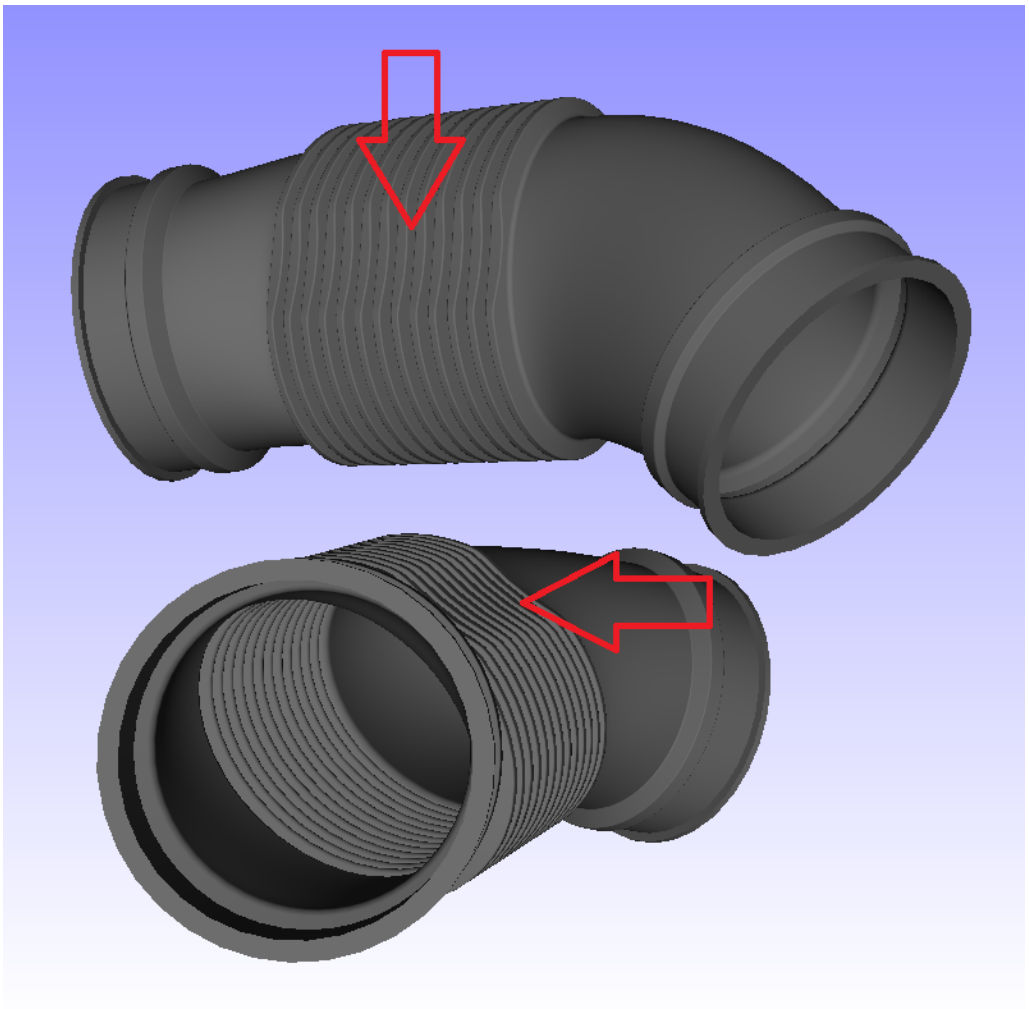
5.3 Vilken metod lämpar sig bäst för att plocka ytor

Den metod som slutligen valdes för att plocka ytorna bestämdes efter att centrumytan sågs som önskvärd att jobba med. Då det i Catia inte finns något smidigt sätt att plocka ut centrumytan så återstår bara en lösning. Nämligen att skapa ett punktmoln av centrumytan med hjälp av de olika skript som finns att tillgå och sedan via punktmolnet skapa en ny yta i Meshlab. Nackdelen för Volvos Cars del med denna metod, är deras önskan att undvika tredjepartsprogram. Det har dock under arbetets gång talats om att det kanske är möjligt att skapa en ny funktion i IPS för att skapa en yta från ett punktmoln. Det är dock

något som ligger längre fram i tiden och kanske är ett möjligt examensarbete för framtida studenter.

5.4 I vilka fall skulle det vara fördelaktigt för Volvo Cars att använda sig utav Flexible Surface Software?

Bland annat de luftbälgar som har både ett ovalt tvärsnitt i kombination med urgröpningar lämpar sig väldigt bra för att simuleras i programmet. Även de fall där luftbälgen har ribbor för att kunna flexa. I kabelsimuleringen saknas särskilt möjligheten att lösa problemet med urgröpningarna och ribbor. Anledningen till att luftbälgarna har urgröpningar är dessutom för att de inte ska komma för nära det objektet som ligger utanför dessa. Så troligtvis är det i första hand där man är intresserad av att veta avståndet. Exempel på sådan urgröpning finns markerad med röda pilar i figur 5.1



Figur 5.1: Luftbälg 31474241 i två olika vinklar. De röda pilarna markerar urgröpningen.

5.5 Vad återstår?

För att Volvo Cars skall kunna använda sig av det nya Flexible Surface Software i sin dagliga verksamhet är det viktigt att utveckla programmet så att det i största möjliga mån går att undvika skriptning. Att lära sig skriptning är troligen för omfattande för de anställda på Volvo som redan har mycket att hantera i den dagliga verksamheten. Därför är ett viktigt steg att programmera in olika knappar och funktioner i IPS som gör samma sak som de skrivna skripten med enkla förklaringar hur hur de används och vilka inställningar som krävs för vad. De skript som finns tillgängliga idag tycker vi löser de allra flesta problem som vi ställts inför. Därför hade det varit en bra idé att jobba vidare för att kunna implementera dessa som funktioner direkt i IPS-programvaran.

Det krävs även en smidigare lösning för att hantera ytor, ifrån cad-filen fås inner-, samt ytter-ytan ifrån Catia. Denna går dock inte att simulera direkt utan behöver lagas och fixas via skriptning i IPS. Därför är det bättre, smidigare och går snabbare att göra detta direkt i IPS från en cad-fil. Då får användaren möjlighet att kunna välja själv vilken av ytter-, inner- och centrumytan som ska användas samt sparar tid. Det är dock viktigt för Volvo Cars att undvika tredjepartsprogram likt Meshlab och därför är det önskvärt att kunna skapa en yta från ett punktmoln direkt i IPS. Volvo Cars vill undvika tredjepartsprogram eftersom att det är en stor process att implementera i verksamheten, lika så att det kostar pengar och tid att införa nya program. Utöver detta finns det också en osäkerhet i tredjepartsprogram med öppen källkod.

Dessutom bör det vara möjligt att under en simulering få ut extremvärdet eller geometrin med det minsta avstånd till närliggande objekt sparad. Denna geometri kan vara intressant att undersöka närmare för att se vilken del av geometrin som har minsta avstånd eller hur den deformerats.

Det bör också vara möjligt att under simulering mäta avståndet från den tjocka ytan till närliggande objekt och inte som idag direkt från ytan. Det blir onödigt omständigt att behöva lägga till tjockleken manuellt och kan lätt bli fel.
Det bör också vara möjligt att i en framtida version hantera varierande tjocklek längs med bälgen på ett bättre sätt. Den ultimata lösningen vore om det var möjligt att skapa ytan direkt från soliden, där tjockleken följer med utan ytterligare inställningar.

Slutligen är vår åsikt att Volvo Cars bör se hur de skulle kunna implementera programvaran redan idag med de skript som finns tillgängliga. Med hjälp från FCC och Volvo Cars egna kunskaper i hur de vill utföra simuleringarna och möjligheter att utföra dessa på verkliga fall. Detta bäddar för en bra grund för programvaran att fortsätta utvecklas och att Volvo Cars i framtiden kan ytterligare öka säkerheten i sina simuleringar och packa saker ännu tätare i sina motorer.

6
Slutsats

Flexible Surface software är att rekommendera vid användning av luftbälgar där det finns en luftbälg som inte är helt cirkulär och då särskild när det finns en fördjupning eller då luftbälges tjocklek varierar.

Idag simulerar Volvo Cars luftbälgarna som kablage och det som är mest intressant är hur luftbälgen rör sig samt vilket som blir minsta avstånd till valt objekt i dess närhet. Här finns därför idag ett problem då de vid skapandet av en kabel tappar luftbälgens struktur och rörelsen samt dess avstånd därför kan variera.

Genom att skapa en yta via IPS är det möjligt för Volvo Cars att simulera de problematiska luftbälgarna. De problem som kan förekomma vid framtagningen av ytan är bland annat håligheter och trasiga geometrier. Men de flesta av dessa problem går idag att lösa med hjälpt utav de skript som finns tillgängliga. Vad simuleringarna visar och ur ett tidsmässigt perspektiv blir rekommendationen därför att hålla sig runt 2500 trianglar och jämföra ytan med orginalmodellen visuellt för att se om geometrin blir allt för dålig vid nedskalning av antalet trianglar.

Flexible Surface software är ett verktyg med framtiden för sig och kommer kunna hjälpa många företag utöver Volvo Cars i sin dagliga verksamhet när det väl kommer ut på marknaden. Det finns många användningsområden för programmet som ännu ej är upptäckta och vi tror att simulering av luftbälgar bara är början. Viktigt att framhäva är dock att det skall vara lätt och enkelt att använda, därför behövs vidare utveckling inom användarvänlighet.

Övrig önskvärd utveckling är funktioner där olika tjocklekar hanteras på ett bättre sätt, halvautomatisk framtagning av ytor, samt att det finns knappar och funktioner inbyggda i IPS för att hantera och göra samma saker som skripten gör.

Det har även under arbetets gång varit på tal från FCC sida att IPS skulle kunna via en ny funktion skapa en yta ifrån ett punktmoln. Men detta är något som ligger längre fram i tiden och kanske skulle kunna vara ett fortsatt examensarbete för framtida studenter.

Litteraturförteckning

[1] Harrold, Dave. (Oct 2000). Control Engineering 47.11. *Virtual reality saves money!*: s.36-43

[2] Johannesson, H., Persson, J-G., Petterson, D. (2013). *Produktutveckling - Effektiva metoder för konstruktion och design.*: s.136 Stockholm: Liber

[3] Fraunhofer-Chalmers Centre, *Industrial Path Solutions*
http://www.industrialpathsolutions.com
(Hämtad 2015-03-25)

[4] Fraunhofer-Chalmers Centre, *Industrial Path Solutions*
http://www.industrialpathsolutions.com/ips-software/ips-cable-simulation (Hämtad 2015-03-25)

[5] Meshlab, *Meshlab*
http://meshlab.sourceforge.net
(Hämtad 2015-03-26)

[6] Dassault system, *Catia*
http://www.3ds.com/products-services/catia/products/v5/portfolio
(Hämtad 2015-04-02)

[7] Dassault system, *Catia V5*
http://academy.3ds.com/software/catia/catia-v5-student-edition
(Hämtad 2015-04-02)

[8] Hermansson, T., Carlson, J., Björkenstam, S. och Söderberg, R. (2013). Institution of Mechanical Engineers. *Geometric variation simulation and robust design for flexible cables and hoses*: s.681-689 http://pib.sagepub.com/content/227/5/681

[9] Weischedel, C., Tuganov, A., Hermansson, T., Linn, J., Wardetzky, M., (2012). Berichte des Fraunhofer ITWM, Nr. 220. *Construction of discrete shell models by geometric finite differences*

[10] Linn, J., Stephan, T., Carlsson, J., Bohlin, R., (2008). Progress in Industrial Mathematics at ECMI 2006. *Fast Simulation of Quasistatic Rod Deformations for VR Applications*: s.247-253

[11] Lua, *Lua*
http://www.lua.org/about.html
(Hämtad 2015-05-05)

[12] Gamasutra, *Announcing Game Developer magazine's 2011 Front Line Award winners* http://www.gamasutra.com/view/news/129084/AnnouncingGame Developermagazines2011FrontLineAwardwinners.php
(Hämtad 2015-05-06)

[13] Michigan Tech (2010), Mesh Basics
[PowerPoint-presentation]. www.mtu.edu/cs
(Hämtad 2015-05-11)

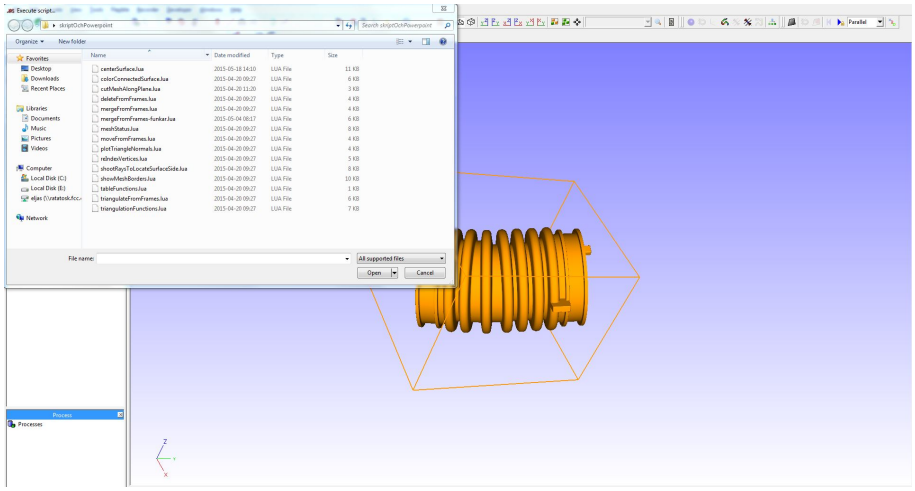
[14] FCC, interna dokument
(hämtad 2015-05-08)

[15] Charliemo, Peter; Volvo Cars Block Leader Engine Bay. 2015. E-mail 18 maj.

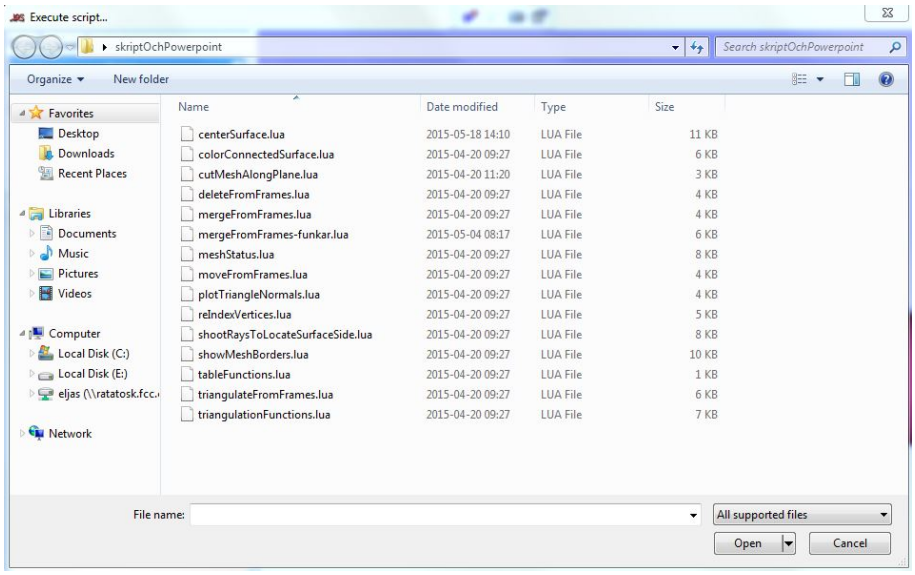
A

Skapa centrumyta från solid luftbälg

I bilaga A förklaras det hur ifrån grunden hur man får ut en bra luftbälg utan några hål eller vertex som ligger fel.



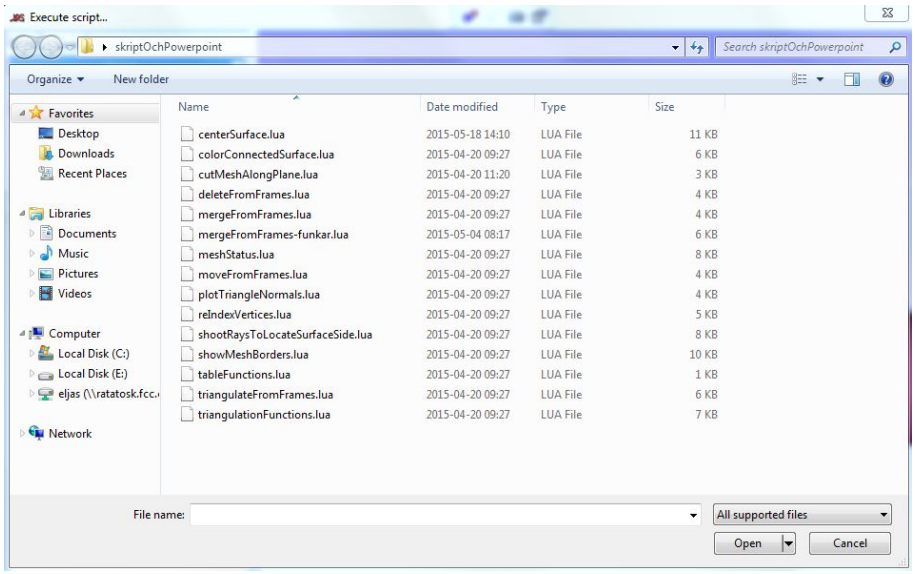
Figur A.1: Klicka på bälgen och välj skriptet meshStatus.lua



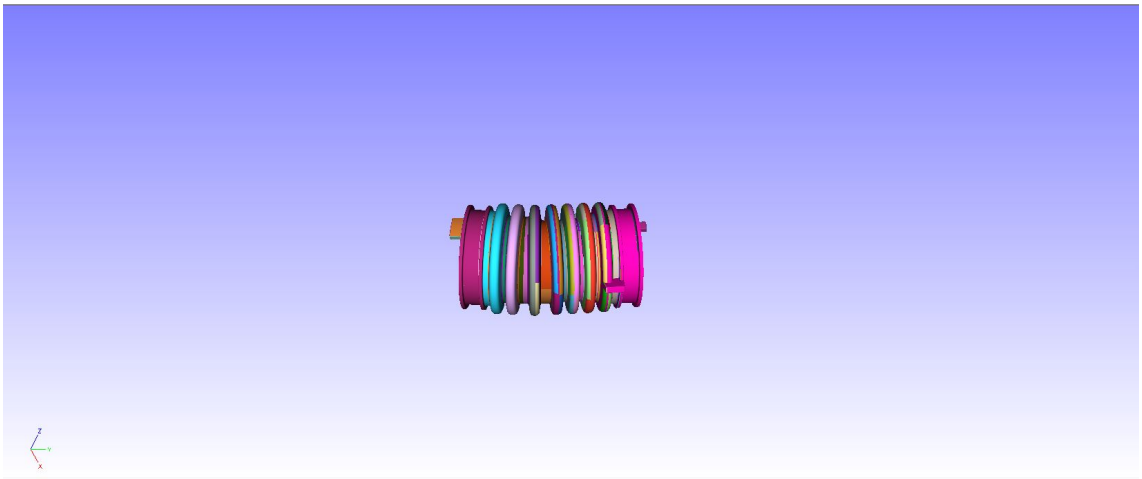
Figur A.2: Alla olika skript synliga

```
[14:57:54] Maximum edge length / Minimum edge length: 13136.219830857
[14:57:54] Number of triangles with 3 neighbors: 15080
[14:57:54] Number of triangles with 2 neighbors: 19356
[14:57:54] Number of triangles with 1 neighbors: 1087
[14:57:54] Number of triangles with 0 neighbors: 6      <----- BAD!
[14:57:54] Number of vertices with 0 triangles: 0
[14:57:54] Number of vertices with 1 triangles: 814
[14:57:54] Number of vertices with 2 triangles: 4574
[14:57:54] Number of vertices with 3 triangles: 10948
[14:57:54] Number of vertices with 4 triangles: 3076
[14:57:54] Number of vertices with 5 triangles: 1750
[14:57:54] Number of vertices with 6 triangles: 4479
[14:57:54] Number of vertices with 7 triangles: 1966
[14:57:54] Number of vertices with 8 triangles: 230
[14:57:54] Number of vertices with 9 triangles: 13
[14:57:54] Number of vertices with 10 triangles: 7
[14:57:54] Number of vertices with 11 triangles: 5
[14:57:54] Number of vertices with 12 triangles: 1
[14:57:54] Number of vertices with 13 triangles: 0
[14:57:54] Number of vertices with 14 triangles: 0
[14:57:54] Number of vertices with 15 triangles: 0
[14:57:54] Number of vertices with 16 triangles: 0
[14:57:54] Number of vertices with 17 triangles: 0
[14:57:54] Number of vertices with 18 triangles: 0
[14:57:54] Number of vertices with 19 triangles: 0
[14:57:54] Number of vertices with 20 triangles: 0
[14:57:55] numSum: 27863 should equal 27863
[14:57:55] All done in 10 s
[14:57:55] Script: Finished in 9.376 s
```

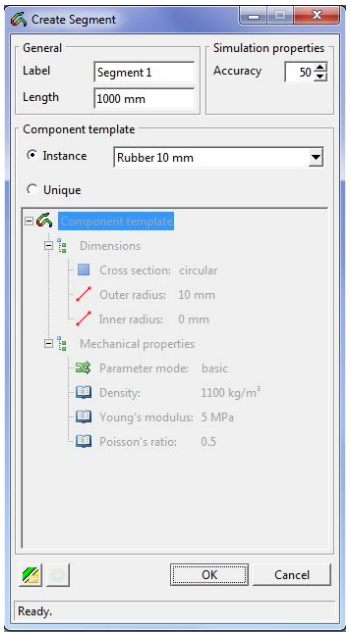
Figur A.3: Resultatet av skriptet, vi har nu trianglar som saknar grannar, därav Bad



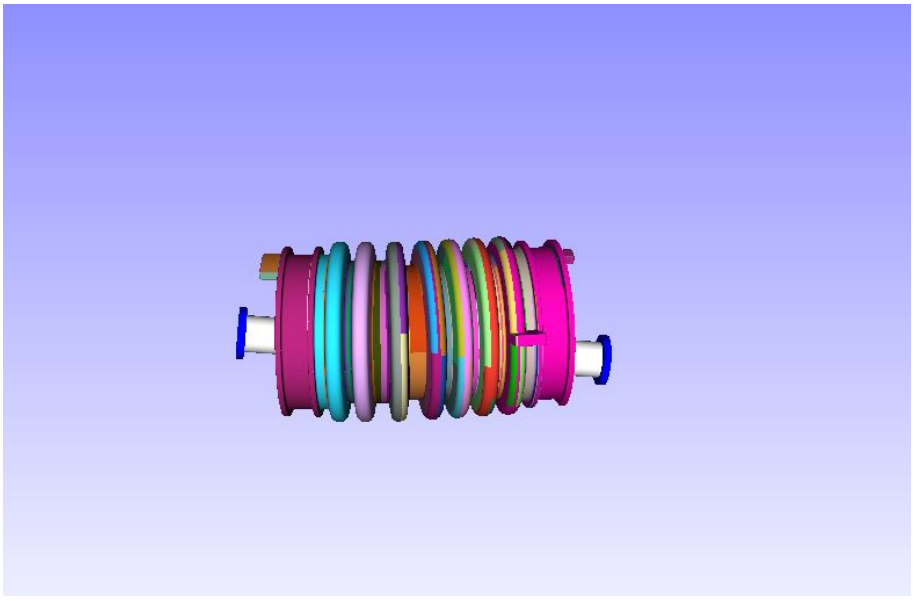
Figur A.4: Klicka på bälgen och välj colorConnectedSurface.lua



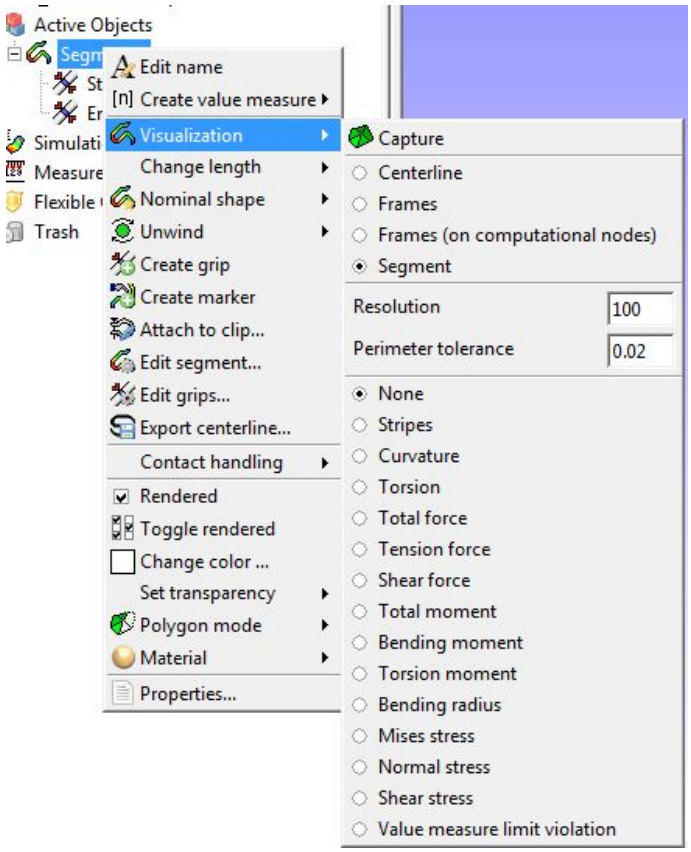
Figur A.5: Resultatet av skriptet colorConnectedSurface.lua



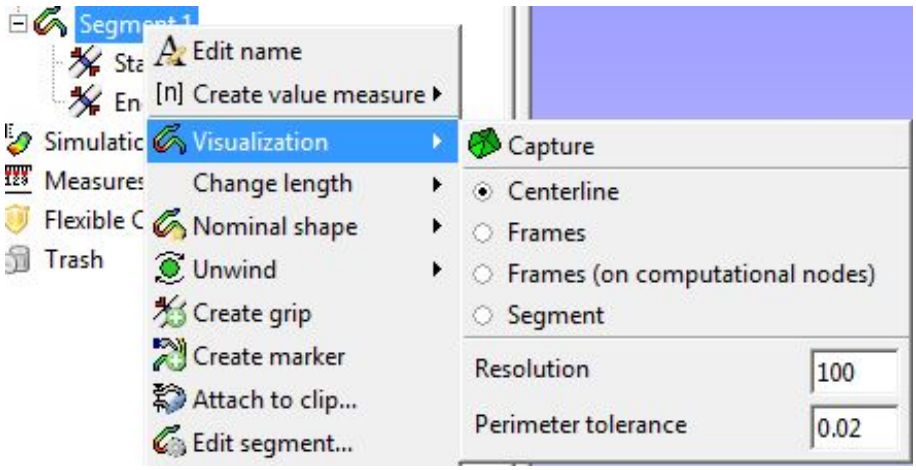
Figur A.6: Skapa en kabel och låt den gå igenom bälgen



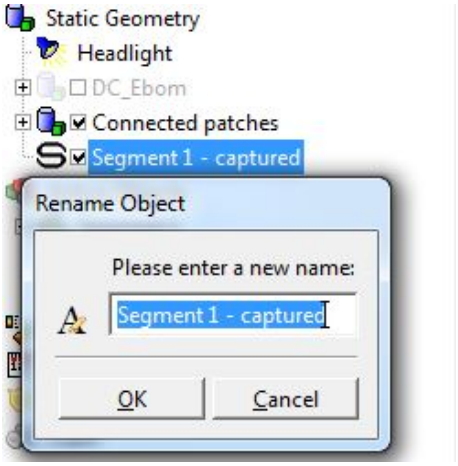
Figur A.7: Placera kabel enligt figur, tanken är att kabeln ska fungera som en centrumlinje genom bälgen



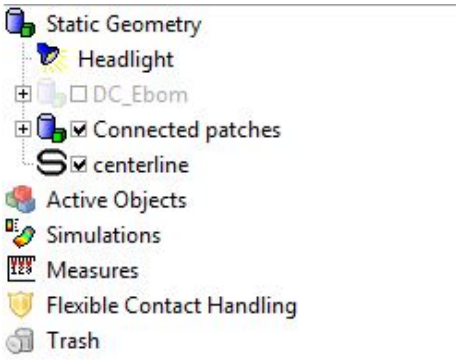
Figur A.8: Visa Centerline på kabeln du nyss skapat



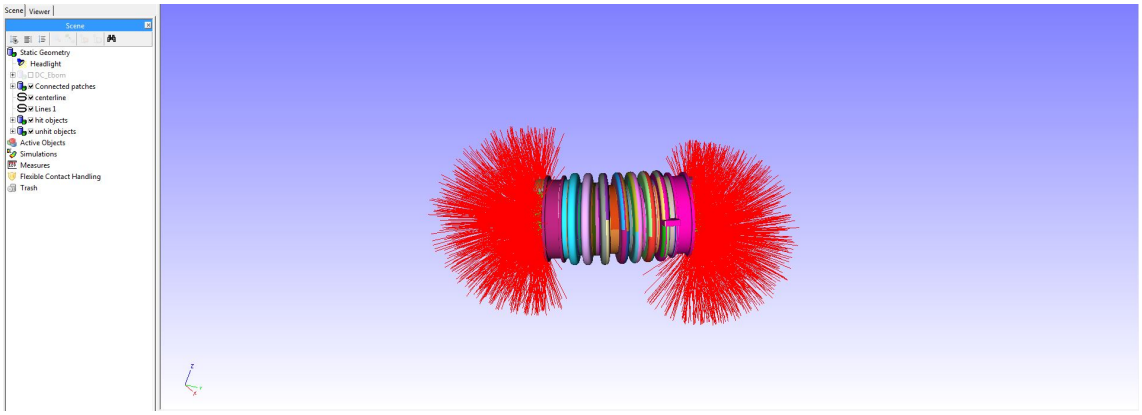
Figur A.9: Därefter väljer du Capture



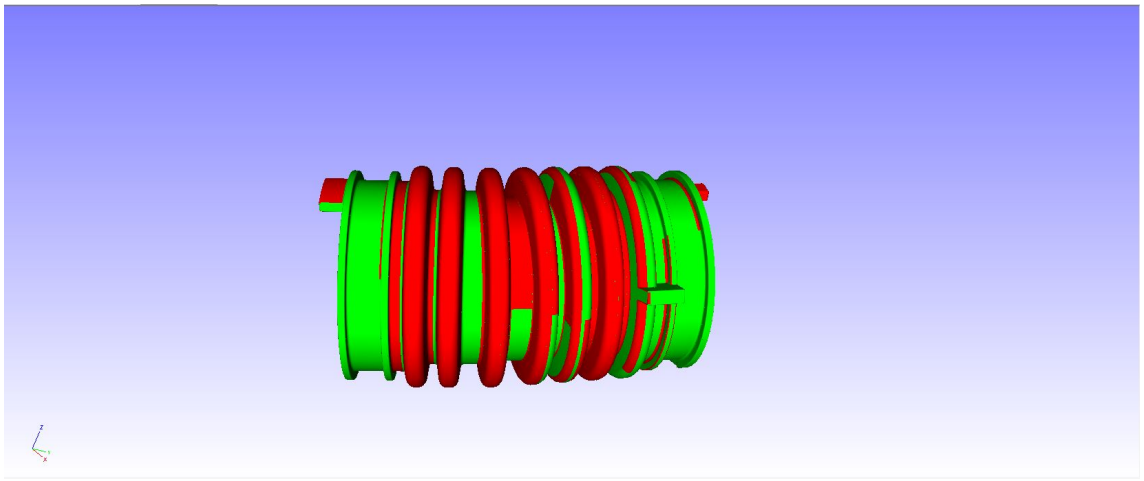
Figur A.10: Döp om det som skapades via Capture till centerline



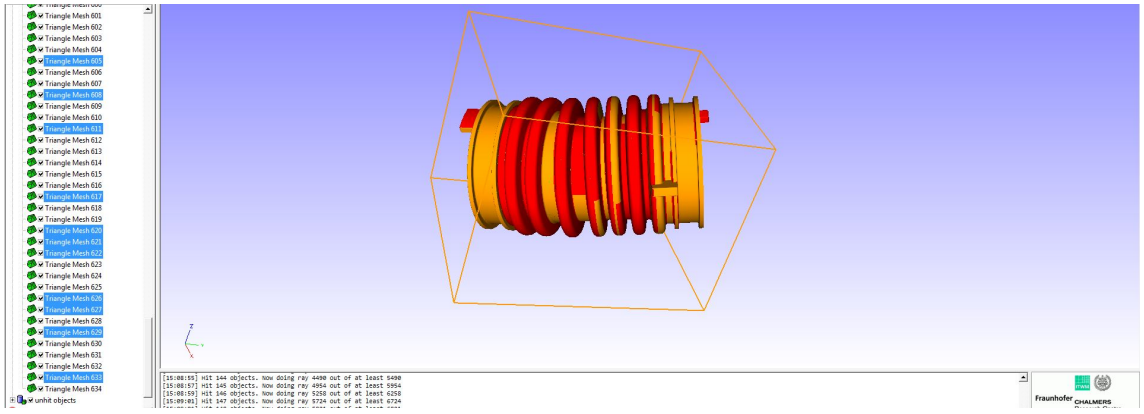
Figur A.11: Ta bort kabeln (Segment 1) du skapade, kvar blir centerline



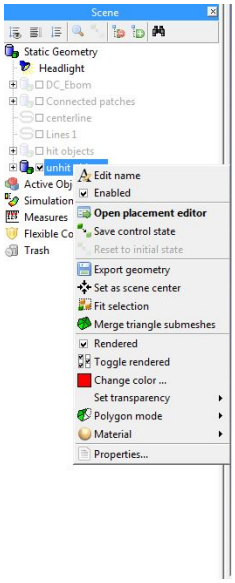
Figur A.12: Klicka på luftbälgen, kör skriptet shootRaysToLocateSurfaceSide.lua. Figuren visar resultatet av skriptet, välj att inte rendera något mer än hit och unhit objects i trädet



Figur A.13: Det rödfärgade området är ytterytan och det grönfärgade är innerytan, tanken här är att alla objekt på bälgens utsida ska vara röda och de på insidan gröna. För att åstadkomma detta markera alla objekt som hamnat fel och flytta till rätt mapp

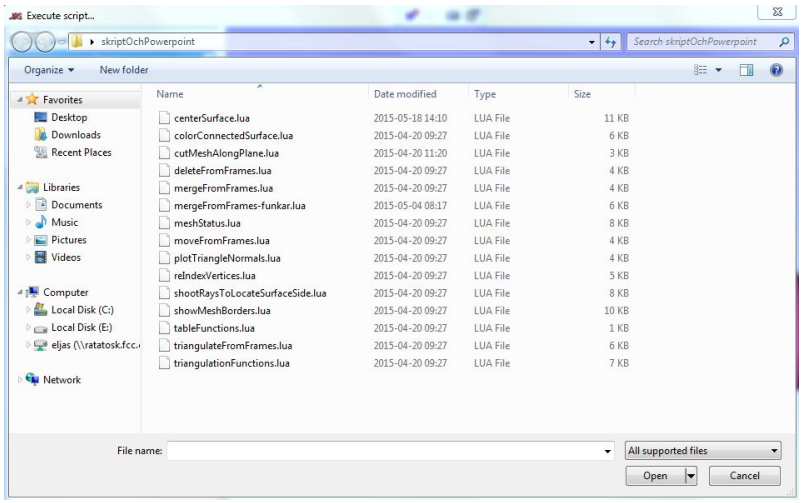


Figur A.14: När ytorna är placerade i rätt mappar, hit och unhit objects, byt färg så att alla under samma grupp har samma färg.

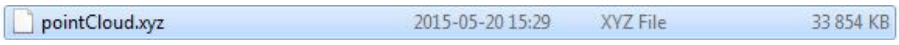


Figur A.15: Markera exempelvis Unhit objects och byt färg

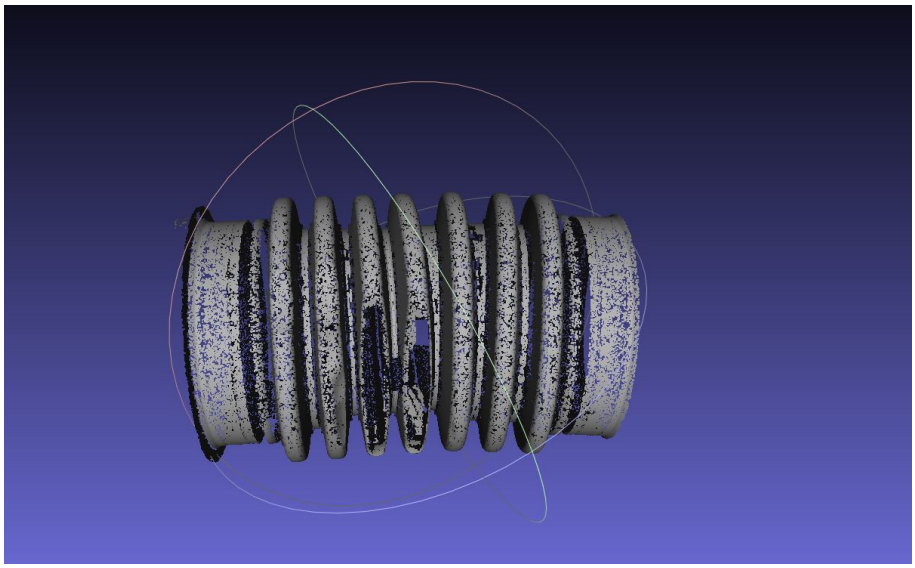
Figur A.16: För att se håligheter på innerytan inne i bälgen, klicka på Viewer och drag sedan i Relative Near Clip Distance. Det gör det möjligt att se igenom bälgen och markera objekt på insdan



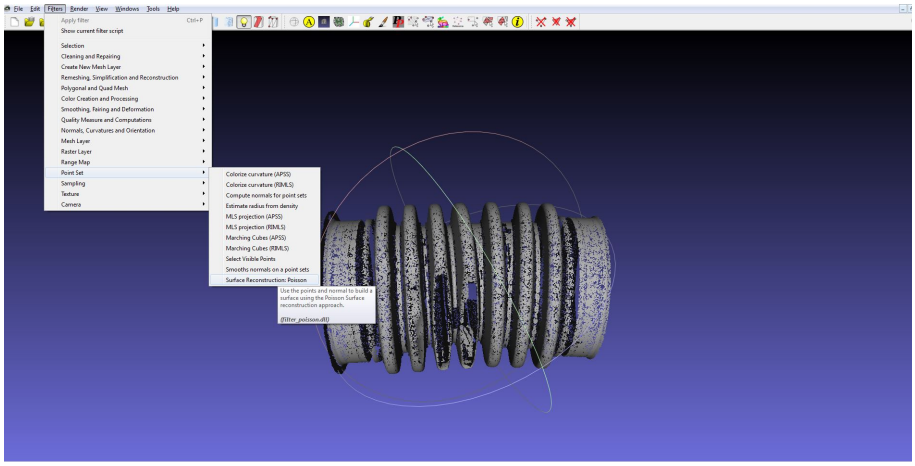
Figur A.17: När alla ytor ligger i rätt mapp kör skriptet centerSurface.lua som skapar ett punktmoln av centrumytan. För att få ett bra punktmoln kan variablerna i skriptet centerSurface.lua behöva ändras



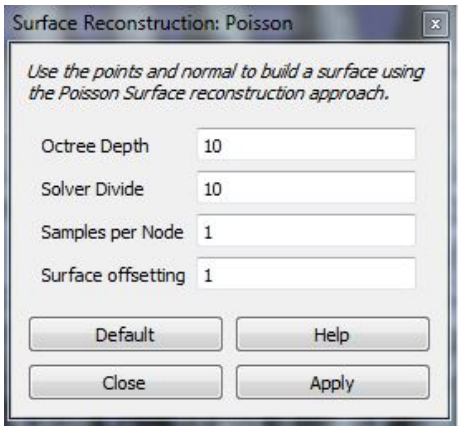
Figur A.18: En fil pointCloud.xyz skapas på Z-disken, importera filen i programmet Meshlab



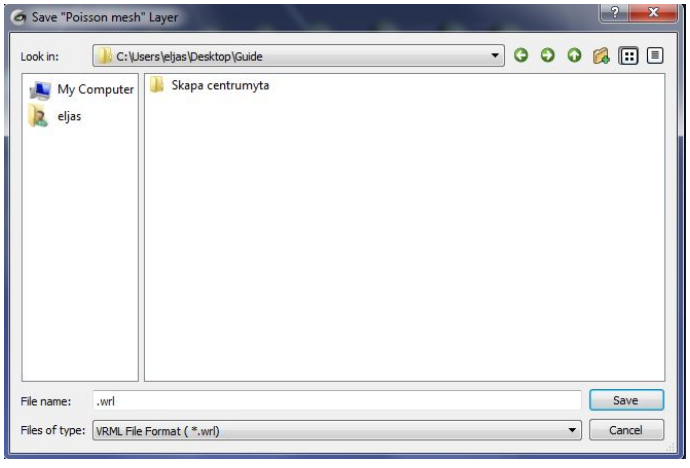
Figur A.19: Hur filen ser ut i programmet Meshlab



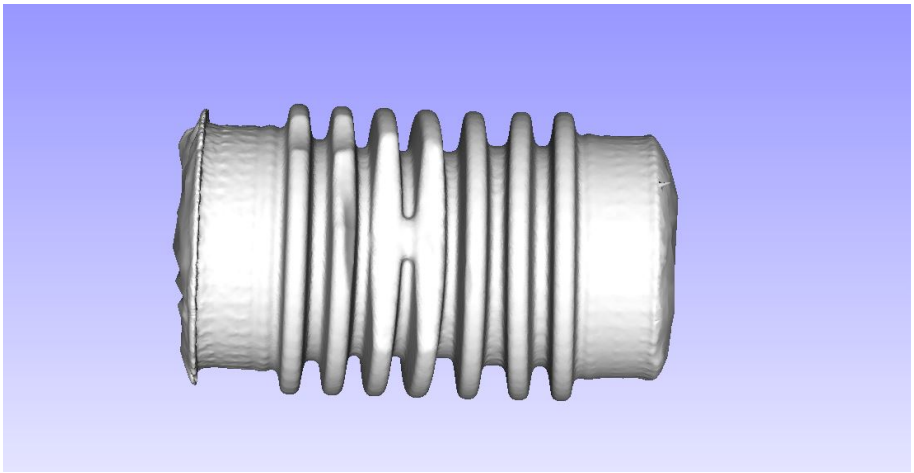
Figur A.20: Använd dig av Filters->Point set-> Surface reconstruction: Poisson



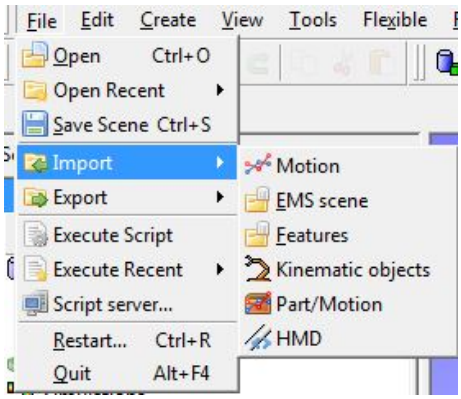
Figur A.21: Olika bälgar kräver olika inställningar dock fungerar inställningarna enligt figur bra för denna bälg



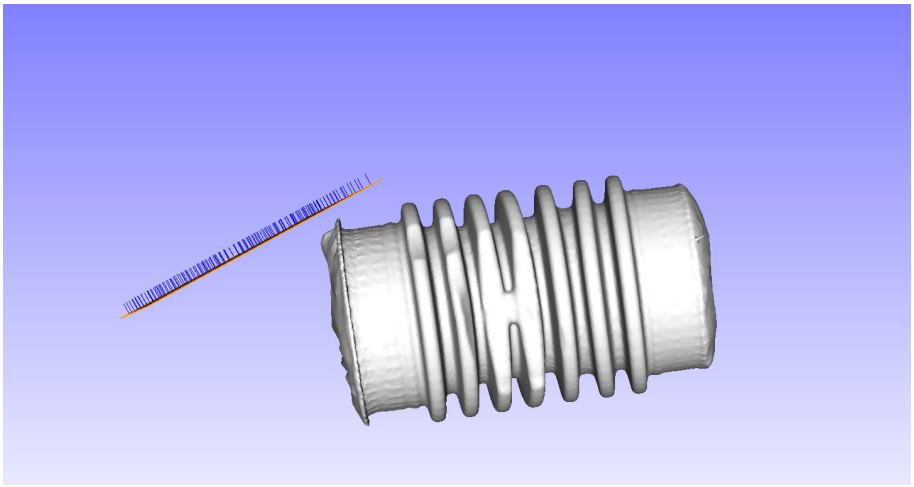
Figur A.22: Exportera filen i .wrl format



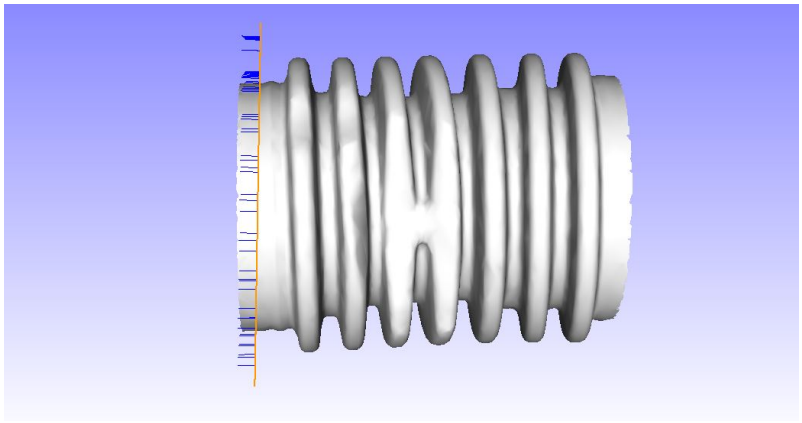
Figur A.23: Importera den nyligen skapade .wrl filen i IPS igen



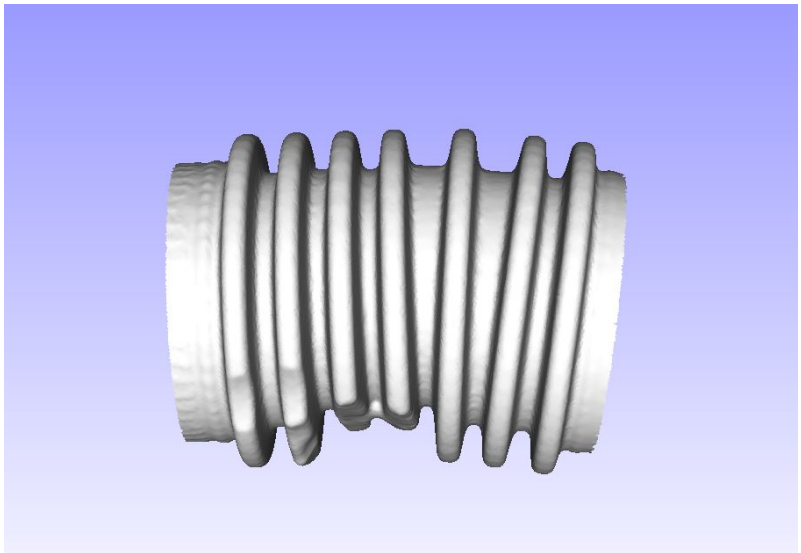
Figur A.24: Importera filen plane.wrl



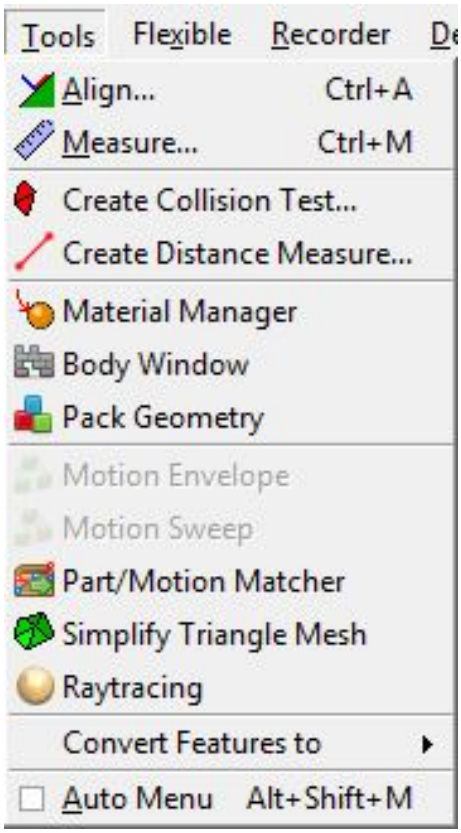
Figur A.25: Markera planet som importerats och kör skriptet plotTriangleNormals.lua för att se var normalerna ligger. De blåa pinnarna i figuren



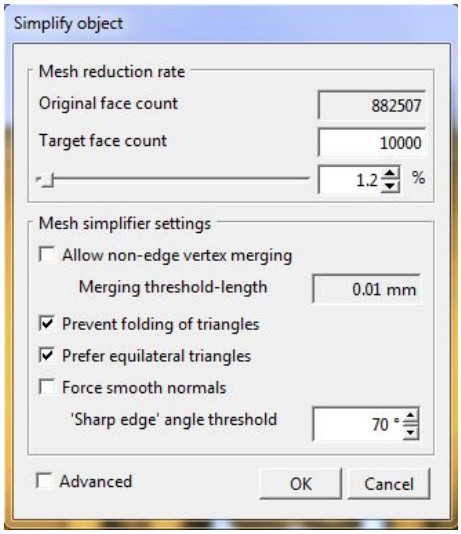
Figur A.26: Lägg planet så att normalerna ligger utåt och placera det så att sidorna kan tas bort. Utför skriptetcutMeshAlongPlane.lua på båda sidorna och bälgen ska nu ha hål på båda sidorna igen



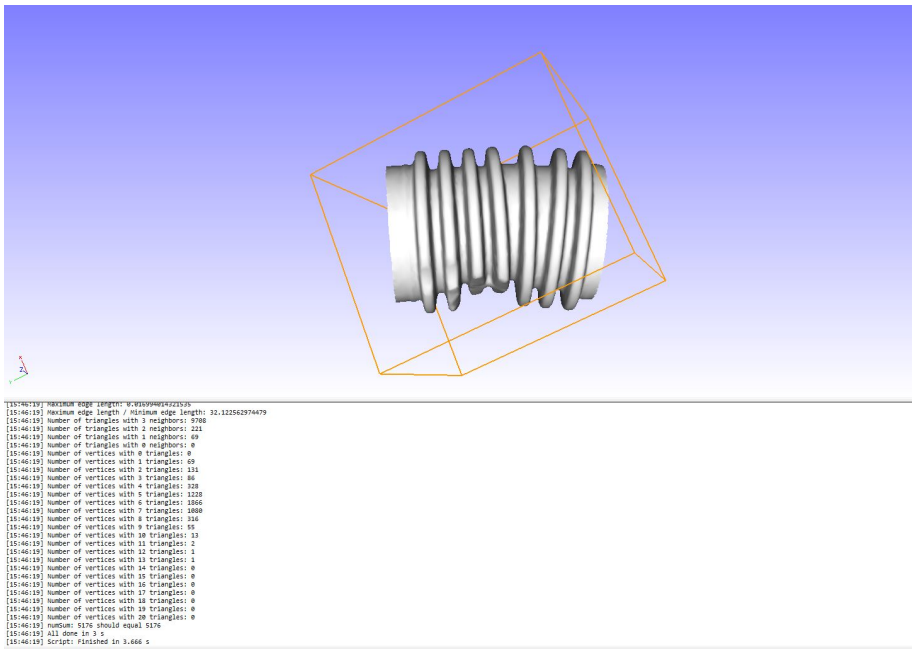
Figur A.27: Resultatet bör likna det i figuren



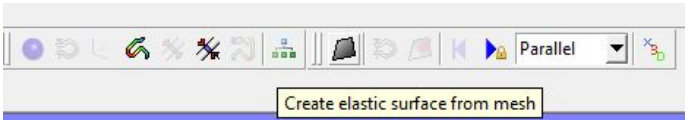
Figur A.28: Minska geometrins antal trianglar i IPS med funktionen Tools-> Simplify Triangle Mesh



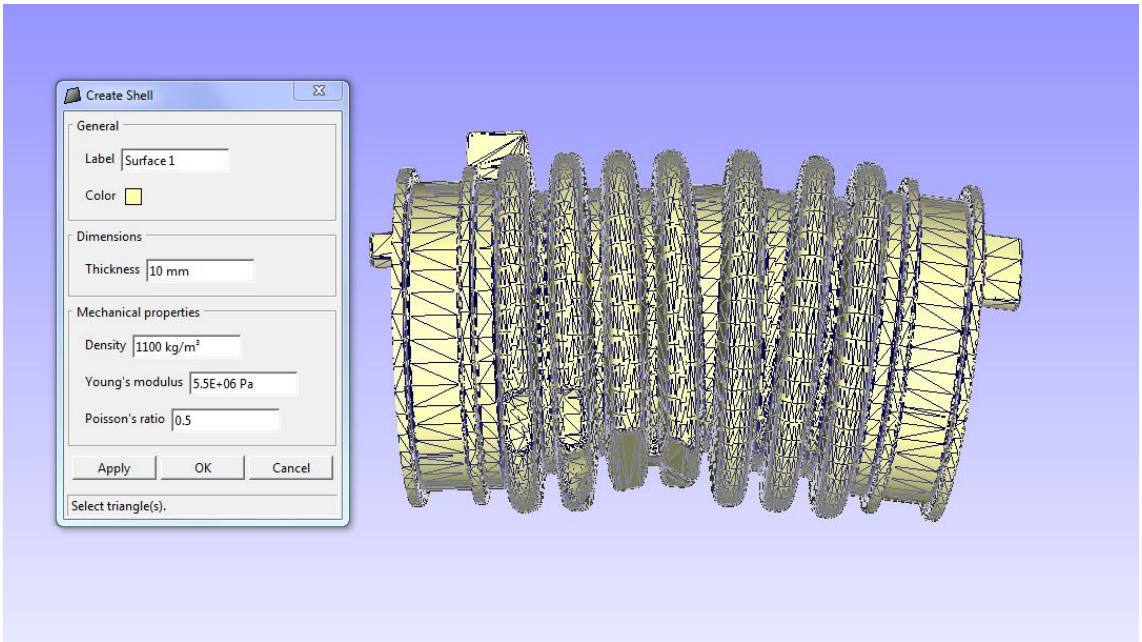
Figur A.29: I det här fallet från 882507 trianglar till 10000 trianglar



Figur A.30: Kör skriptet meshStatus.lua, notera att Bad är borta. Vilket betyder att trianglar ligger rätt



Figur A.31: Tryck nu på knappen Create elastic surface from mesh

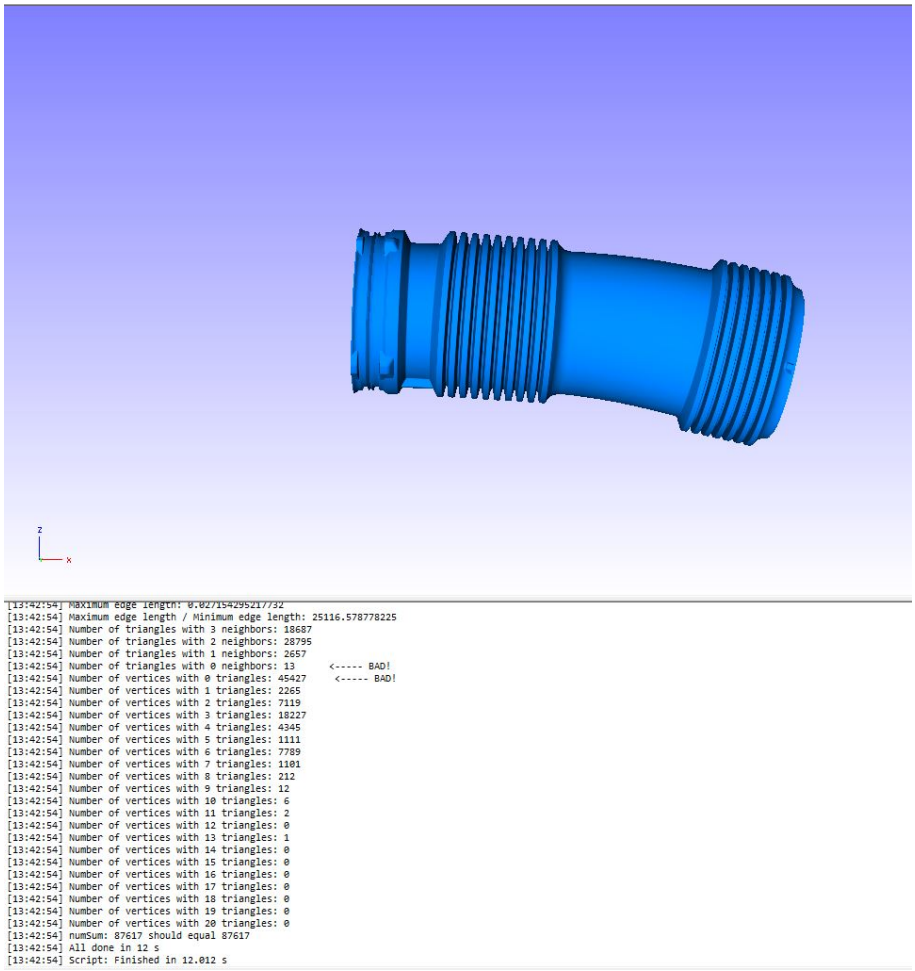


Figur A.32: Om allt är gjort rätt skall du nu kunna skapa en yta av lyftbälgen. Knappa in rätt parametrar och tryck apply.

B

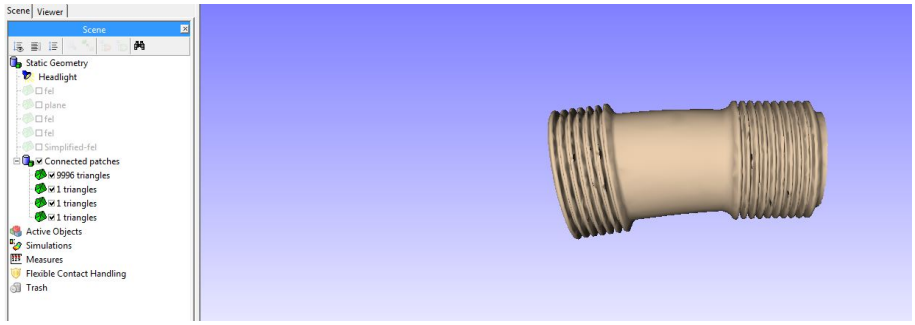
Laga och kontrollera mesh

I bilaga B förklaras det hur man kan lösa problemet med en luftbälg där trianglarna fortfarande ligger fel, i bilaga A kan även problemet med hål lösas samt med trianglar som är dubbletter. Om problem uppstår och du fortfarande får bad på några av trianglarna fortsätt då med dessa steg.

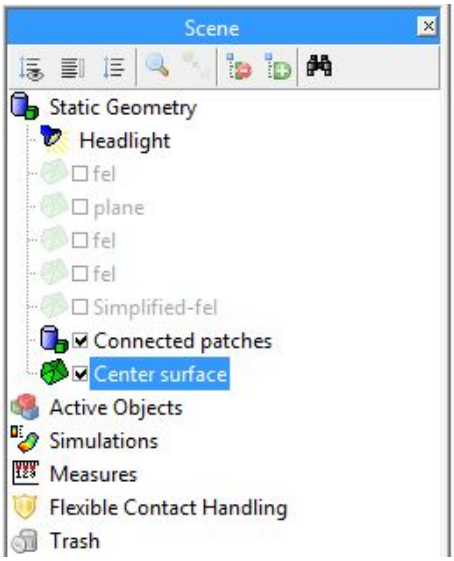


Figur B.1: En annan bälg, gör som tidigare med skriptet meshStatus.lua

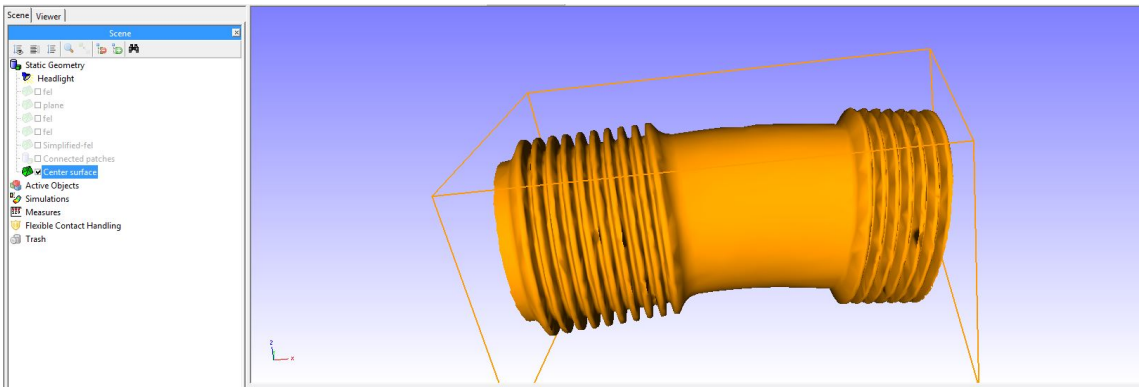
Figur B.2



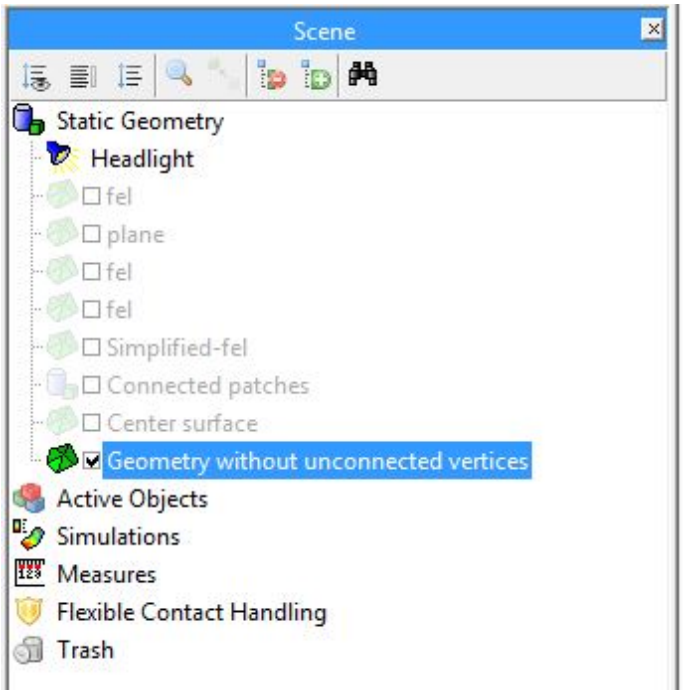
Figur B.3: Kör skriptet colorConnectedSurface.lua. Resultatet ger 9996 trianglar som sitter ihop med varandra, tre som inte gör det. Dessa går att ta bort.



Figur B.4: Byt namn på 9996 triangles till Center surface



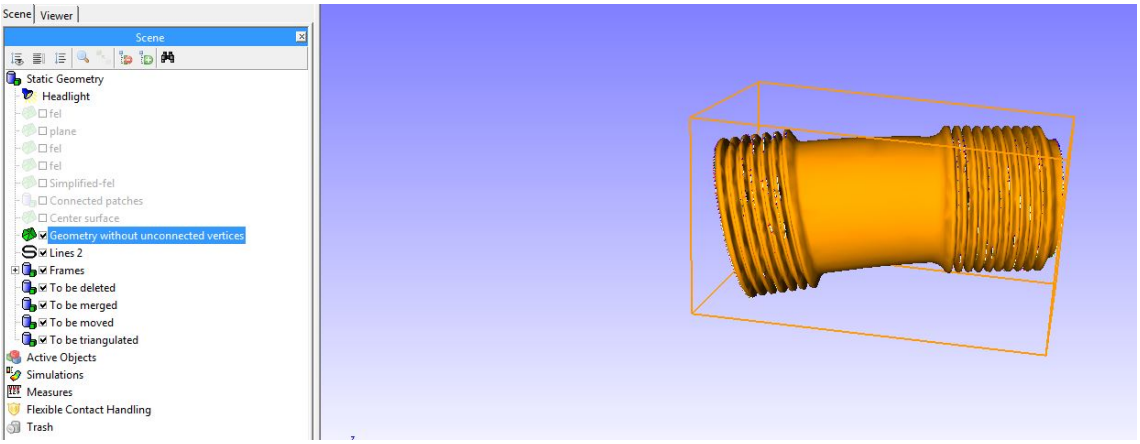
Figur B.5: Kör skriptet reIndexVertices.lua



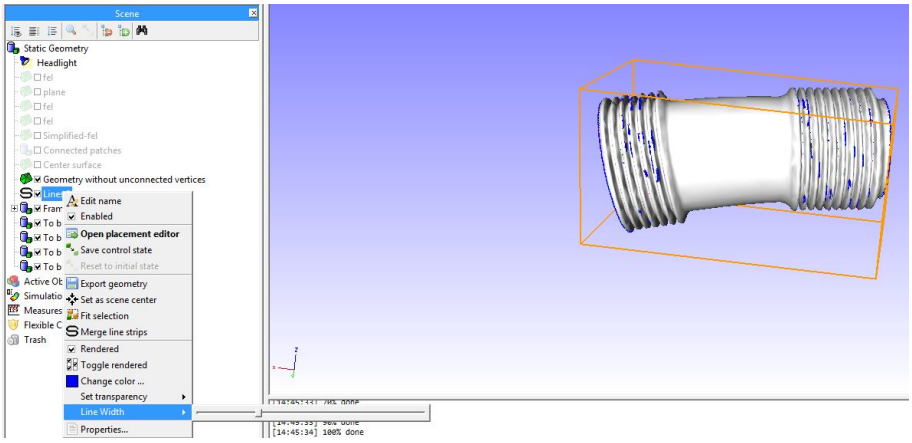
Figur B.6: Utav skriptet reIndexVertices.lua fås en ny geometrigrupp

```
[14:44:50] Number of vertices: 5245
[14:44:50] Number of triangles: 9996
[14:44:50] Minimum angle: 0.74241403871584
[14:44:50] Maximum angle: 176.17321556256
[14:44:50] Minimum angle sum: 180
[14:44:50] Maximum angle sum: 180
[14:44:50] Minimum edge length: 0.00053222501696287
[14:44:50] Maximum edge length: 0.027344521602246
[14:44:50] Maximum edge length / Minimum edge length: 51.377745748005
[14:44:50] Number of triangles with 3 neighbors: 9297
[14:44:50] Number of triangles with 2 neighbors: 668
[14:44:50] Number of triangles with 1 neighbors: 30
[14:44:50] Number of triangles with 0 neighbors: 0
[14:44:50] Number of vertices with 0 triangles: 0
[14:44:50] Number of vertices with 1 triangles: 27
[14:44:50] Number of vertices with 2 triangles: 154
[14:44:50] Number of vertices with 3 triangles: 243
[14:44:50] Number of vertices with 4 triangles: 413
[14:44:50] Number of vertices with 5 triangles: 1307
[14:44:50] Number of vertices with 6 triangles: 1577
[14:44:50] Number of vertices with 7 triangles: 1034
[14:44:50] Number of vertices with 8 triangles: 391
[14:44:50] Number of vertices with 9 triangles: 82
[14:44:50] Number of vertices with 10 triangles: 16
[14:44:50] Number of vertices with 11 triangles: 1
[14:44:50] Number of vertices with 12 triangles: 0
[14:44:50] Number of vertices with 13 triangles: 0
[14:44:50] Number of vertices with 14 triangles: 0
[14:44:50] Number of vertices with 15 triangles: 0
[14:44:50] Number of vertices with 16 triangles: 0
[14:44:50] Number of vertices with 17 triangles: 0
[14:44:50] Number of vertices with 18 triangles: 0
[14:44:50] Number of vertices with 19 triangles: 0
[14:44:50] Number of vertices with 20 triangles: 0
[14:44:50] numSum: 5245 should equal 5245
[14:44:50] All done in 3 s
[14:44:50] Script: Finished in 3.51 s
```

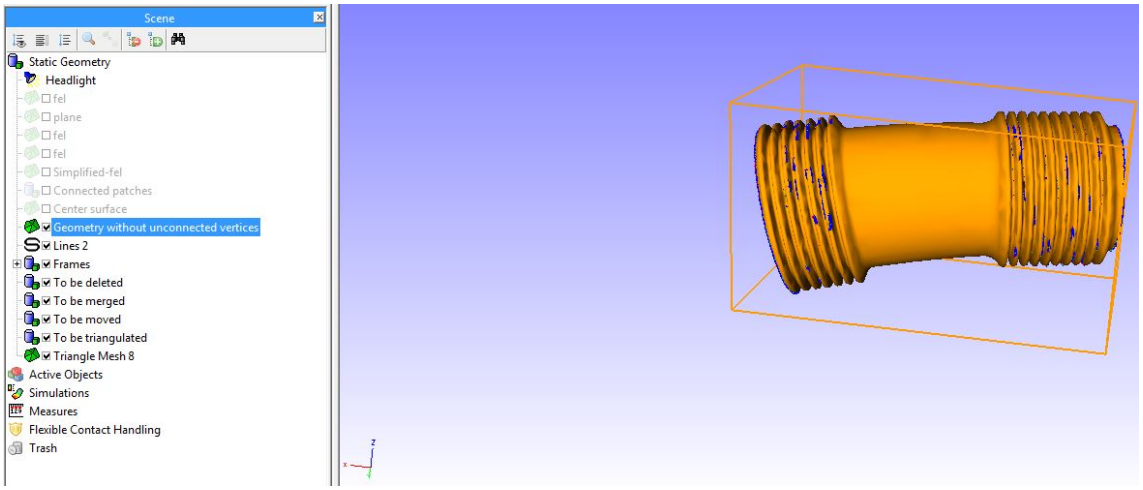
Figur B.7: Kör skriptet showMeshBorders.lua, resultatet visar inga *bad* längre



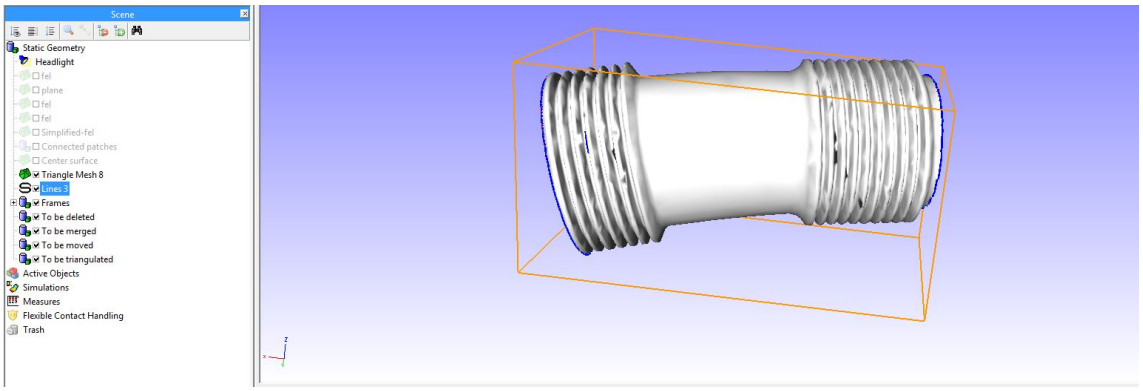
Figur B.8: Kör skriptet showMeshBorders.lua, detta för att meshen kan se bra ut, men ha delar i meshen som inte håller ihop



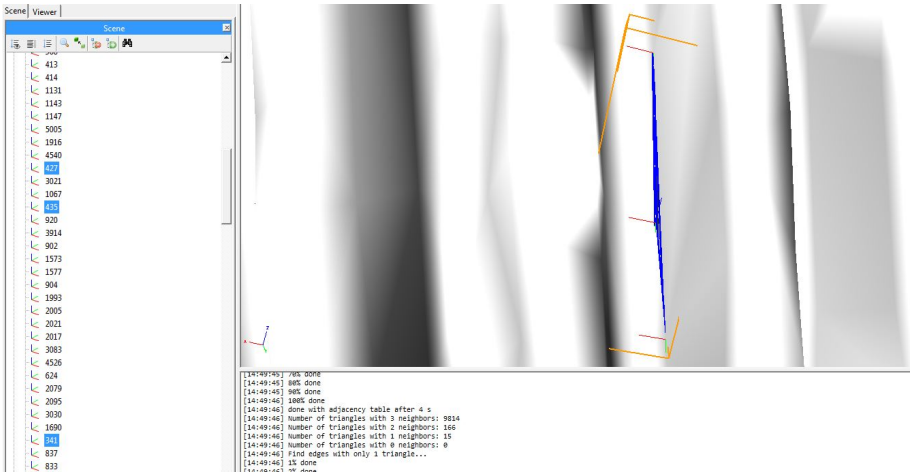
Figur B.9: Klicka på lines, ändra färgen samt tjockleken på linjerna för att lättare kunna se var meshen inte håller ihop



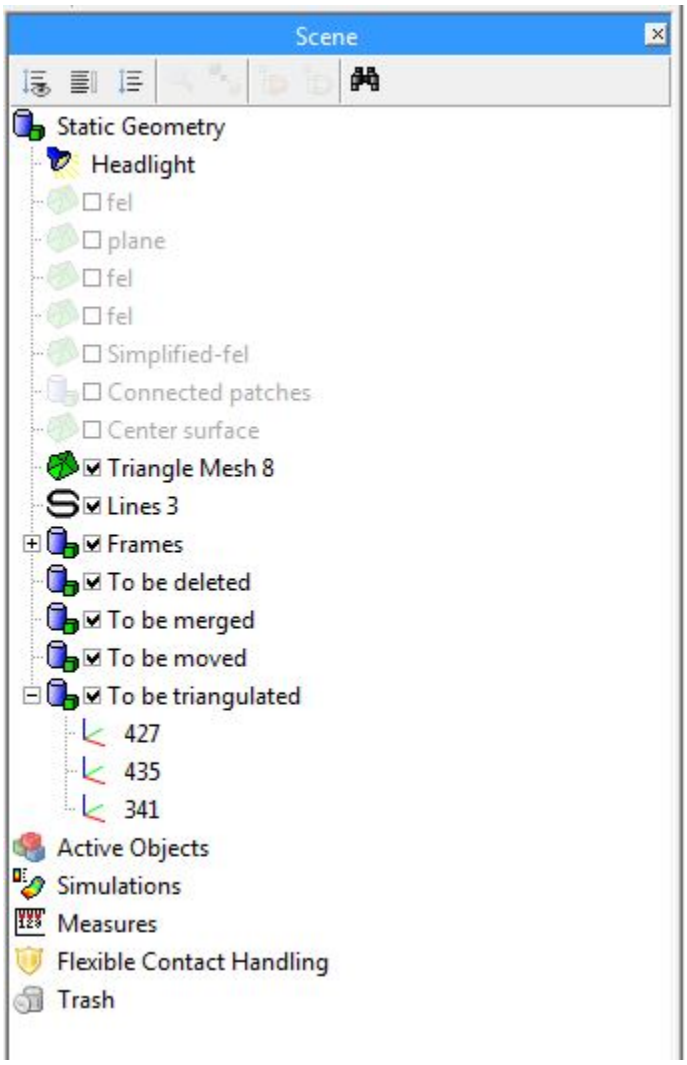
Figur B.10: Laga meshen med skriptet mergeFromFrames-funkar.lua



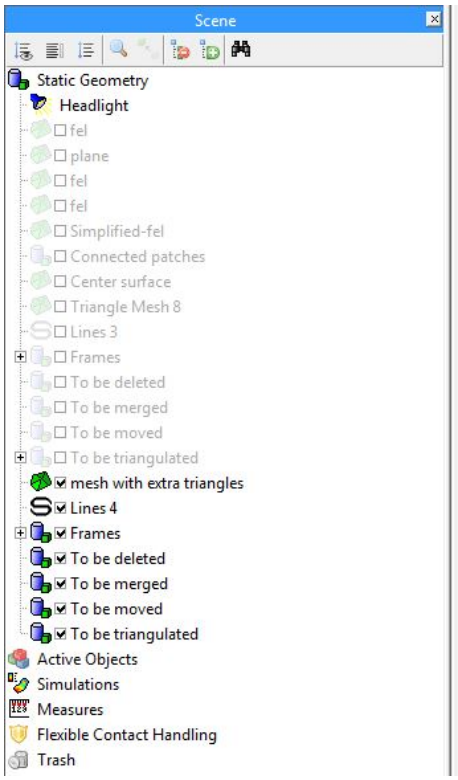
Figur B.11: Alla håligheter försvinner dock inte alltid, vissa trianglar måste manuellt skapas om det finns ett riktigt hål, därför körs skriptet showMeshBorders.lua ytterligare en gång för att lokalisera dessa



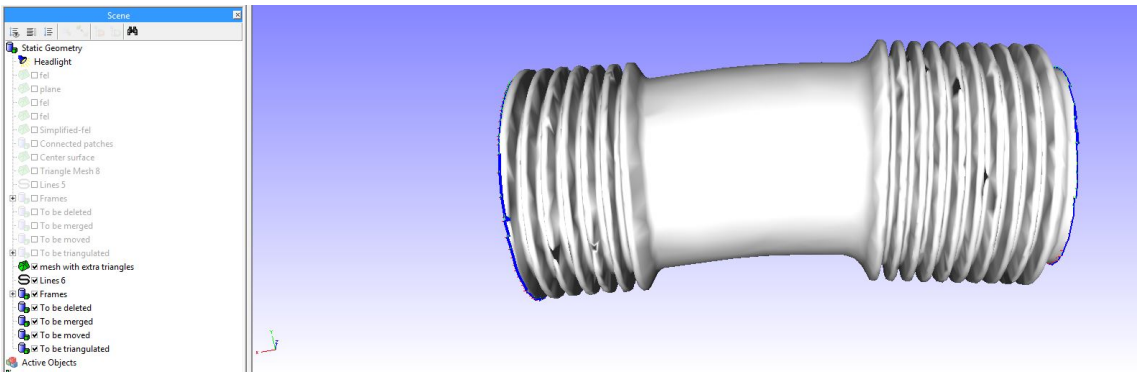
Figur B.12: Hål visualiserat av showMeshBorders.lua, här väljs de frames som tillhör hålet för att kunna skapa en triangel



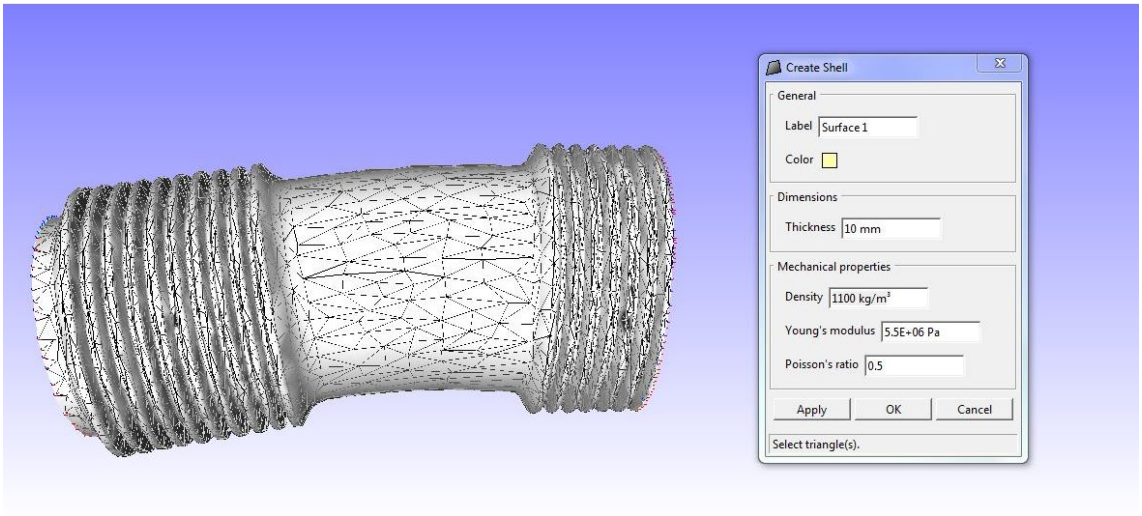
Figur B.13: De markerade frames från föregående bild flyttas därefter till mappen To be triangulated i trädet.



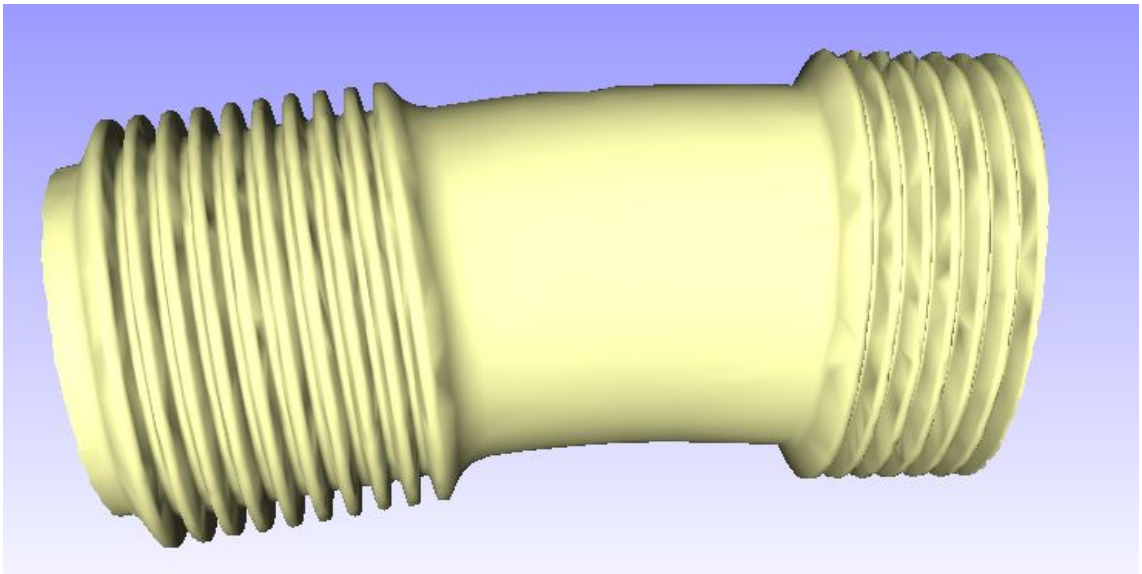
Figur B.14: Kör skriptet triangulateFromFrames.lua och därefter showMeshBorders på den nya geometrin för att se ifall det återstår några hål, om så är fallet repetera föregående två steg.



Figur B.15: I det här fallet är geometrin nu redo att skapa en yta på



Figur B.16: Efter att ha klickat i Create Shell fås ett fönster upp där det är möjligt att välja materialparametrar samt tjocklek

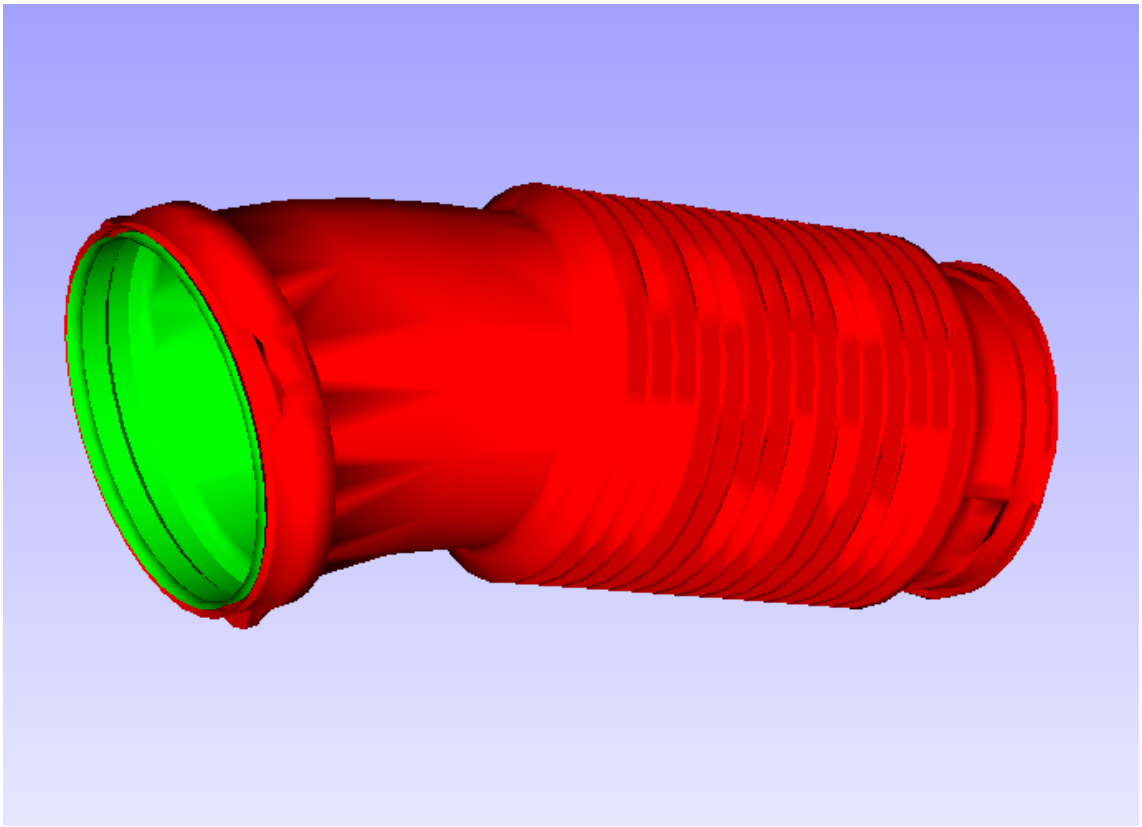


Figur B.17: Färdig yta

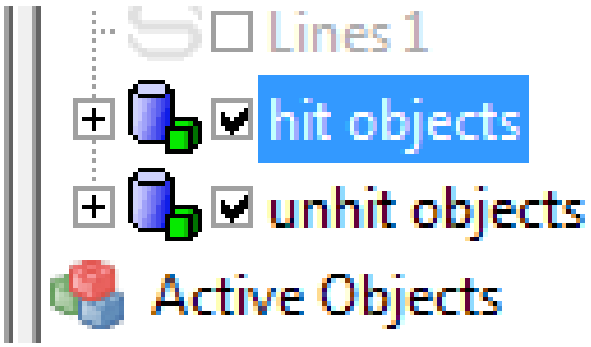
C

Plocka ut ytter och inneryta från solid luftbälg

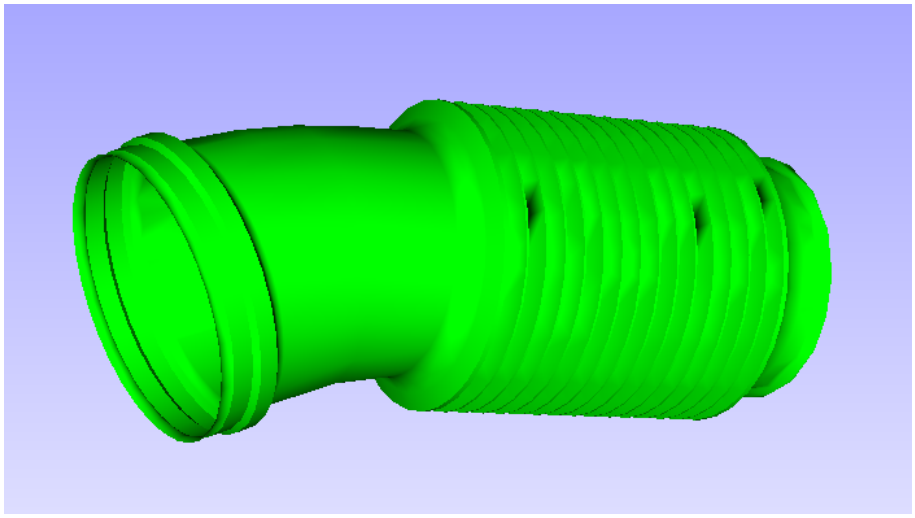
I bilaga C förklaras det hur via IPS går att plocka ut en inner samt ytteryta via olika skript som FCC tillhandahåller.
Efter att ha gjort de första 16 stegen i bilaga A, är det möjligt att plocka ut ytter och innerytan av bälgen istället för centrumytan.



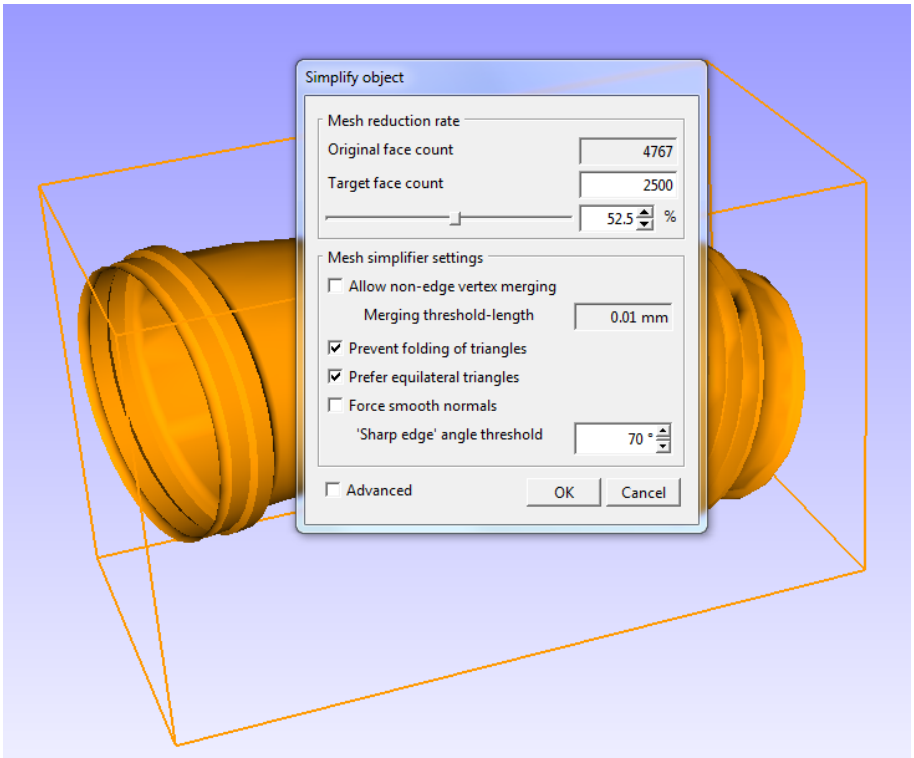
Figur C.1: Här är bälgens alla geometrier placerade i hit och unhit objects



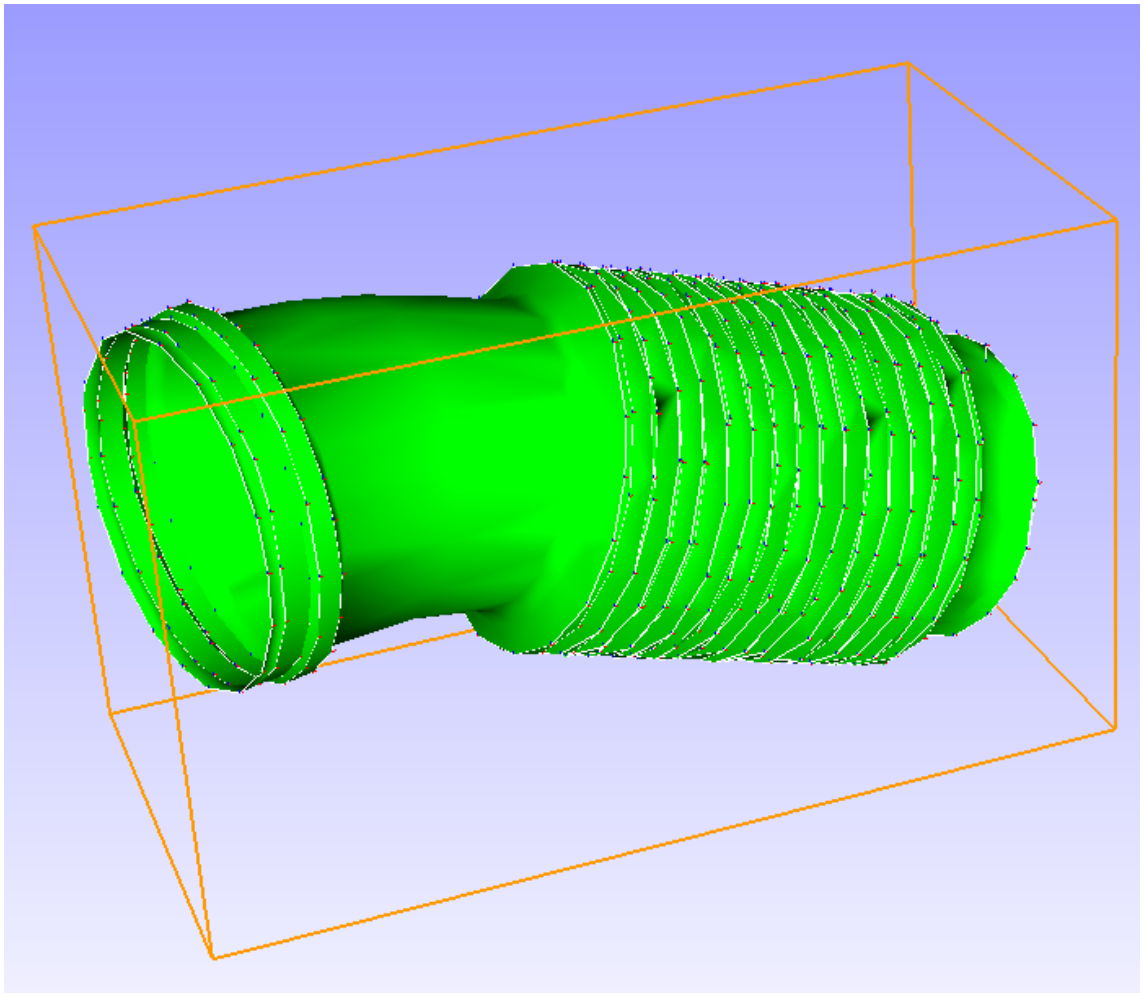
Figur C.2: Markera den av ytorna som är intressant, i det här fallet innerytan. Bocka sedan ur motstående yta så den inte syns.



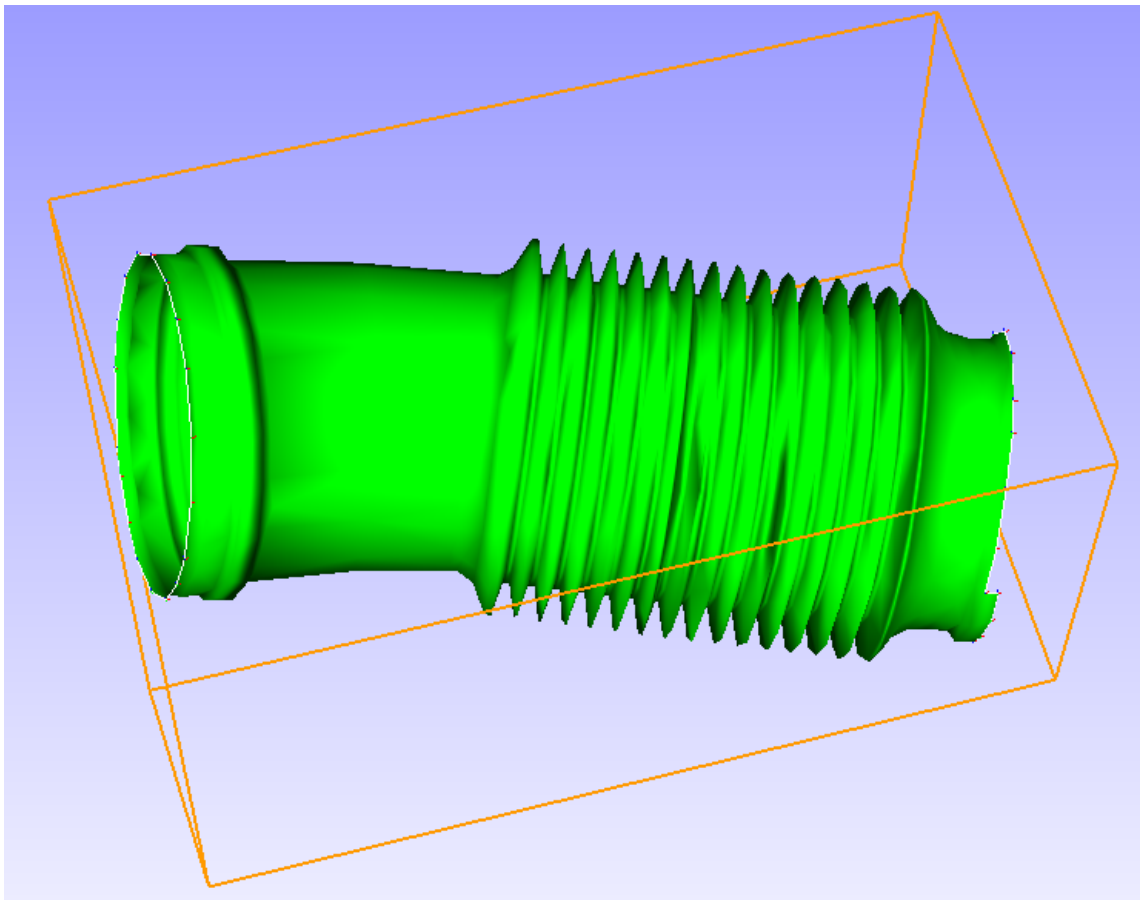
Figur C.3: Nu ska bara vald yta synas enligt bild.



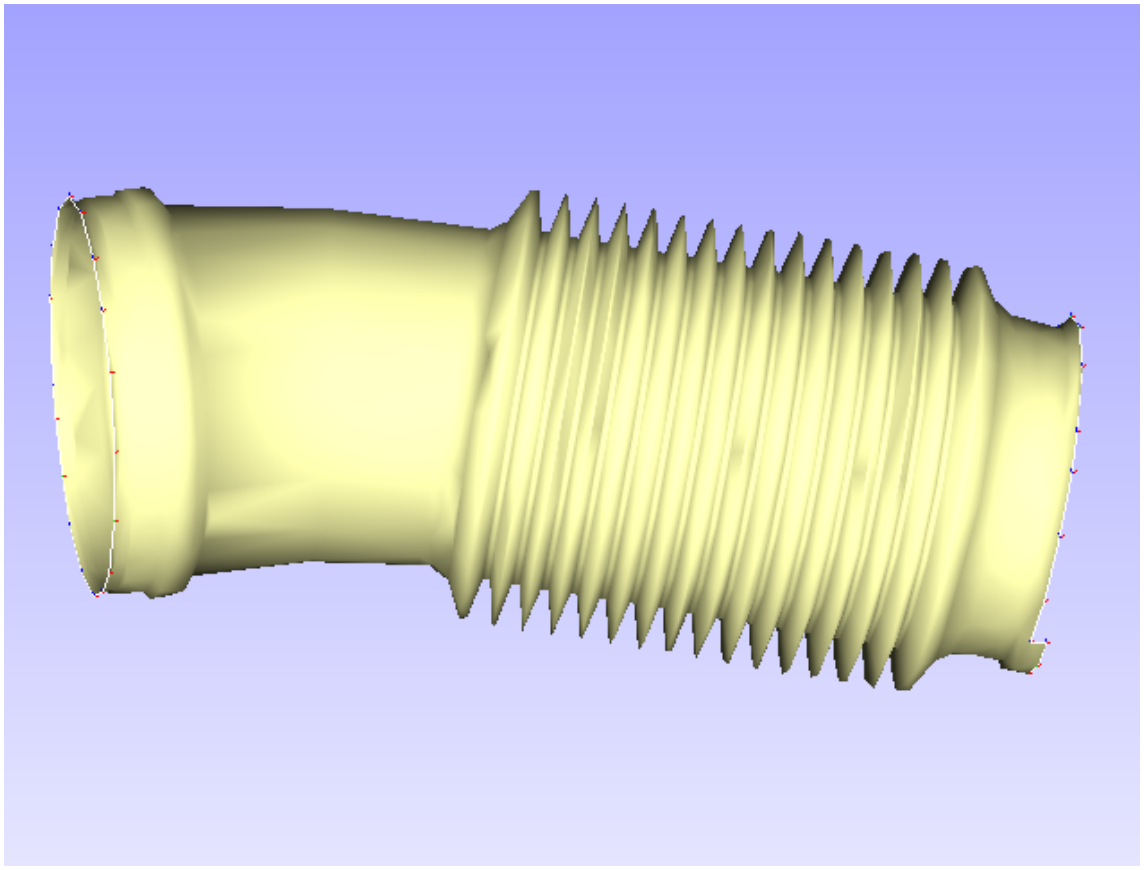
Figur C.4: Den ytan som fås är inte en enkel geometri, för att skapa en sådan utav den använd verktyget Simplify triangle mesh och skala ner till lämpligt antal trianglar



Figur C.5: Använd skriptet showMeshBorders.lua, troligen ser bälgen ut så här. Linjerna markerar var bälgen inte håller ihop



Figur C.6: Använd skriptet mergeFromFrames-funkar.lua, vilket kan vara tidskrävande. Kör sedan showMeshBorders.lua igen och om inte bälgen har några hål i sig bör den se ut som ovan. Skulle det fortfarande synas hål, följ stegen i B.12 och framåt.

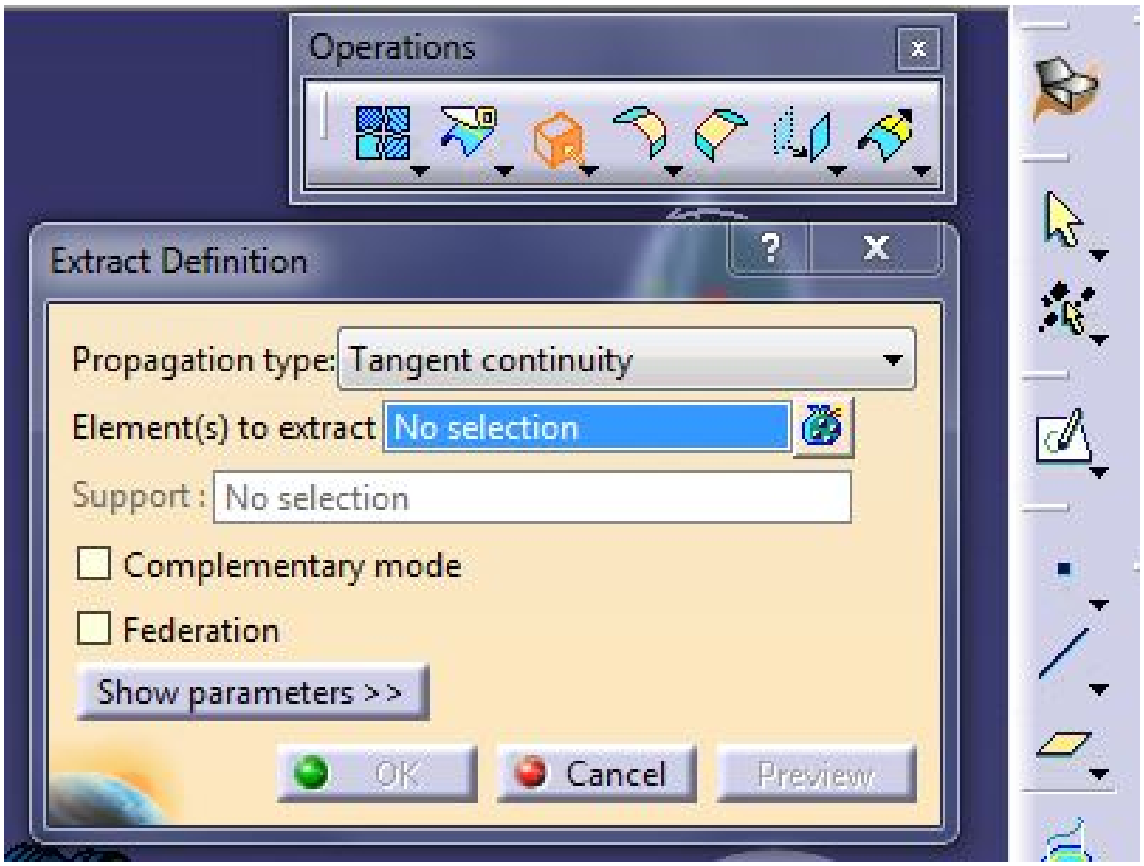


Figur C.7: Annars är bälgen redo att skapas en yta på och du är klar, likt ovan.

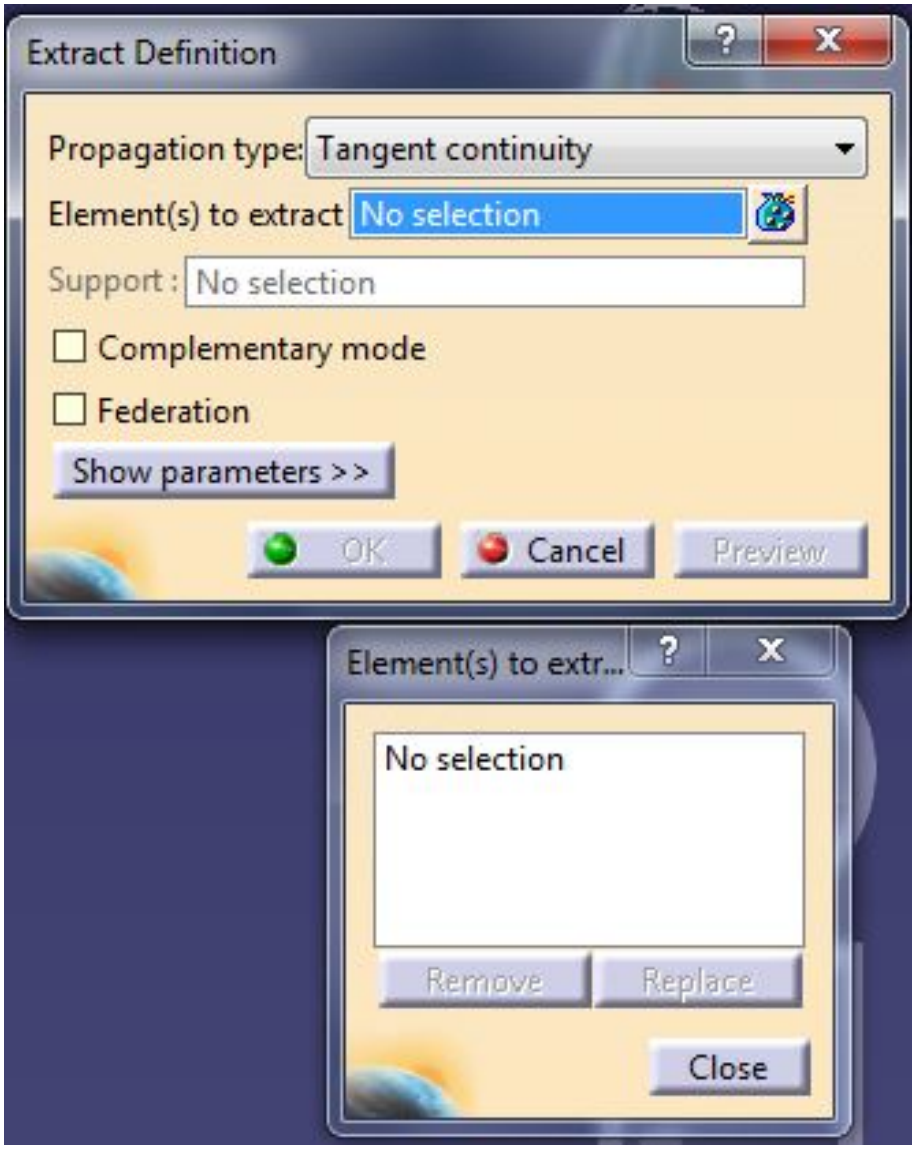
D

Plocka ut ytter och inneryta i Catia v5

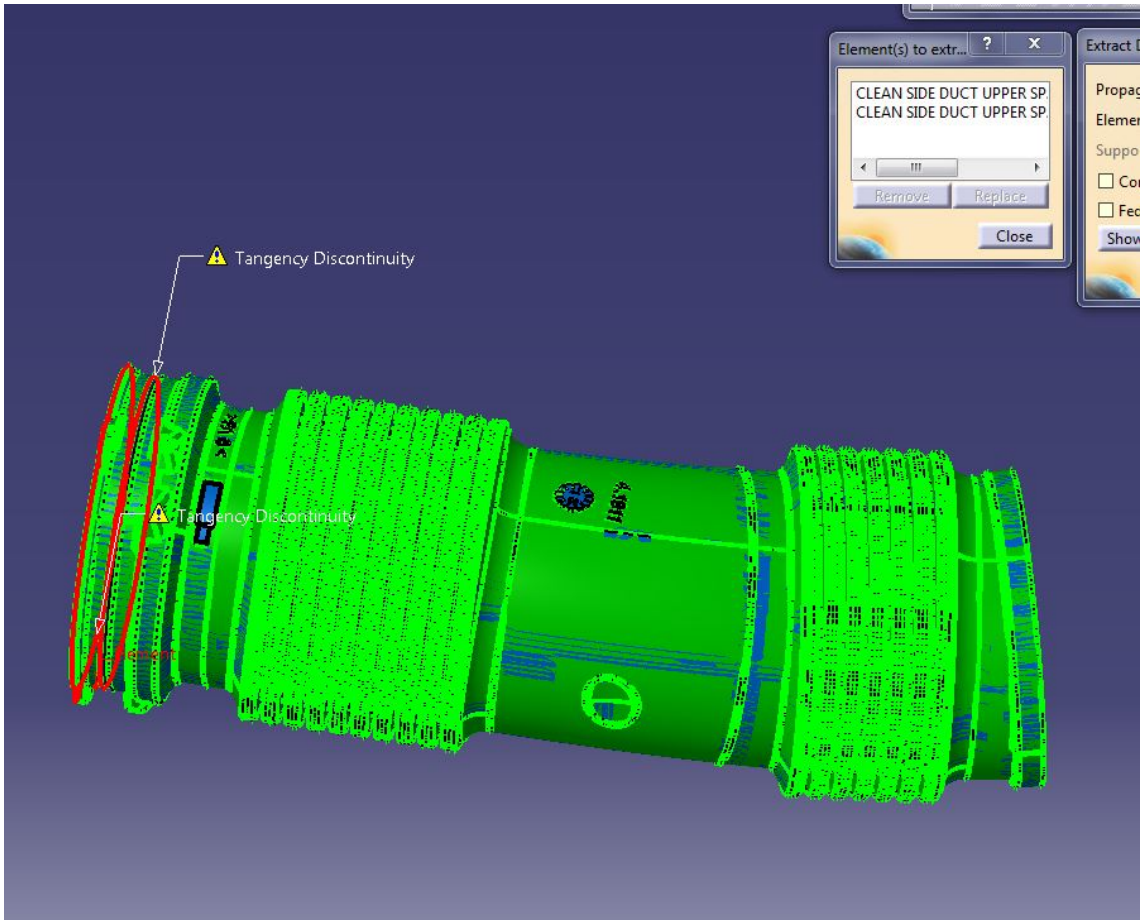
I bilaga D förklaras det hur en inner och/eller ytteryta plockas ut från Catia v5



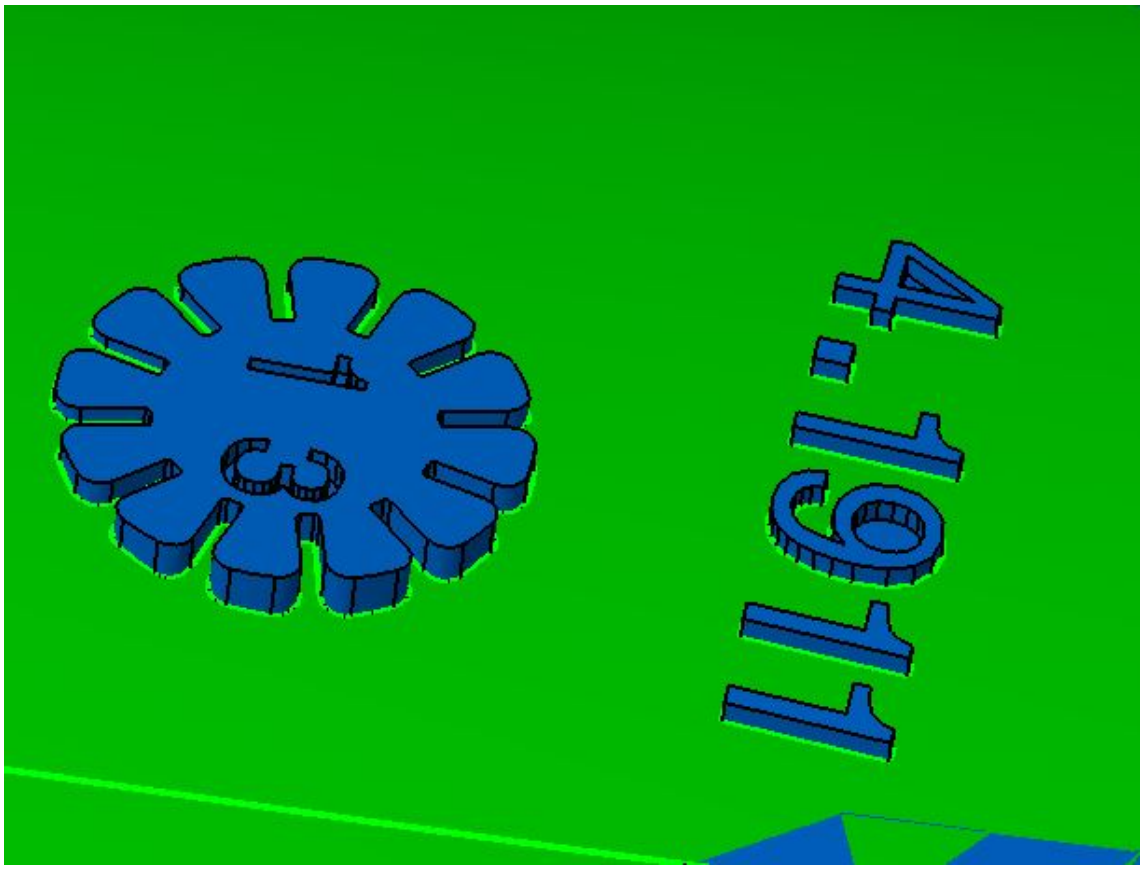
Figur D.1: Öppna upp bälgen i Catia och gå in i arbetsbänken Generative shape design. I arbetsbänken hittas Operations -> Extract.



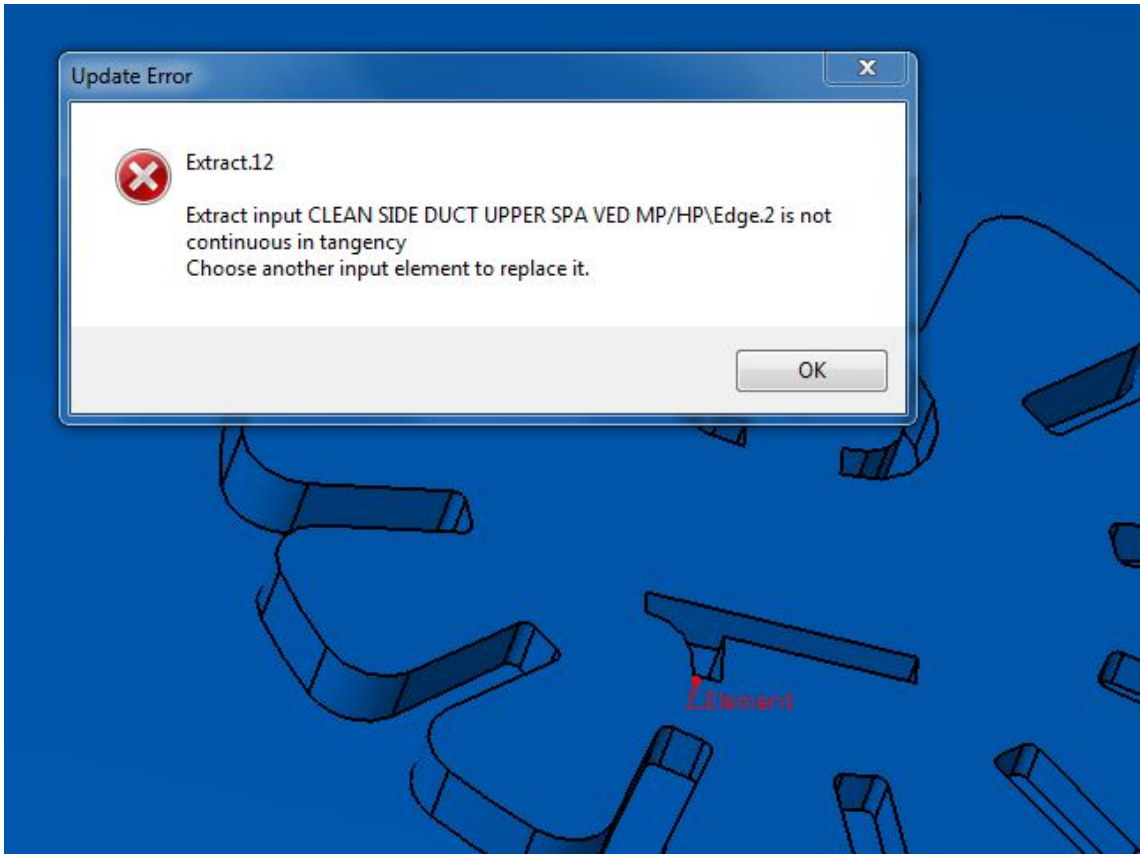
Figur D.2: Klicka i Propagation type: till Tangent continuity. Klicka på den blå figuren jämte No selection brevid Element(s) to extract för att kunna välja flera segment.



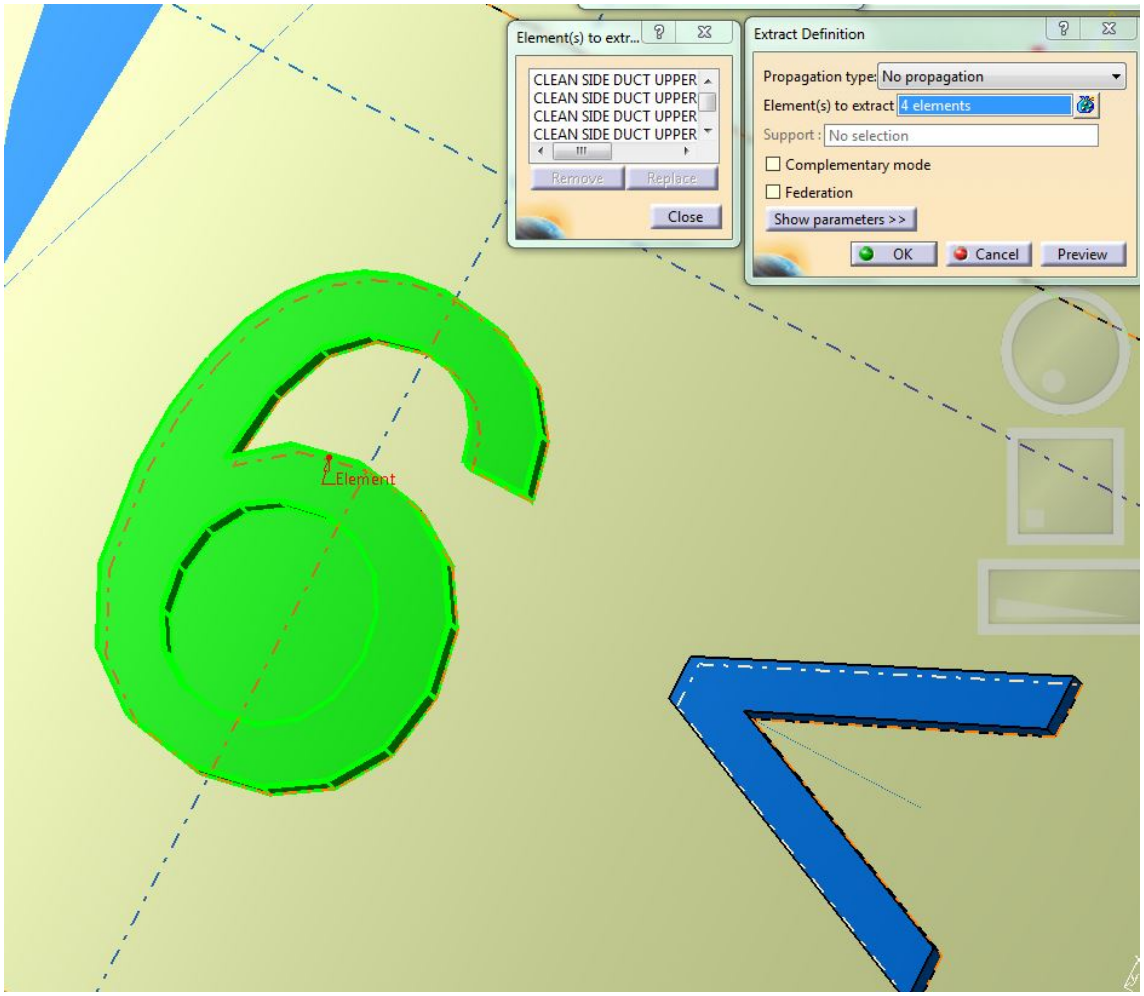
Figur D.3: Klicka på de stora runda ytorna på bälgen och de markeras enligt ovan.



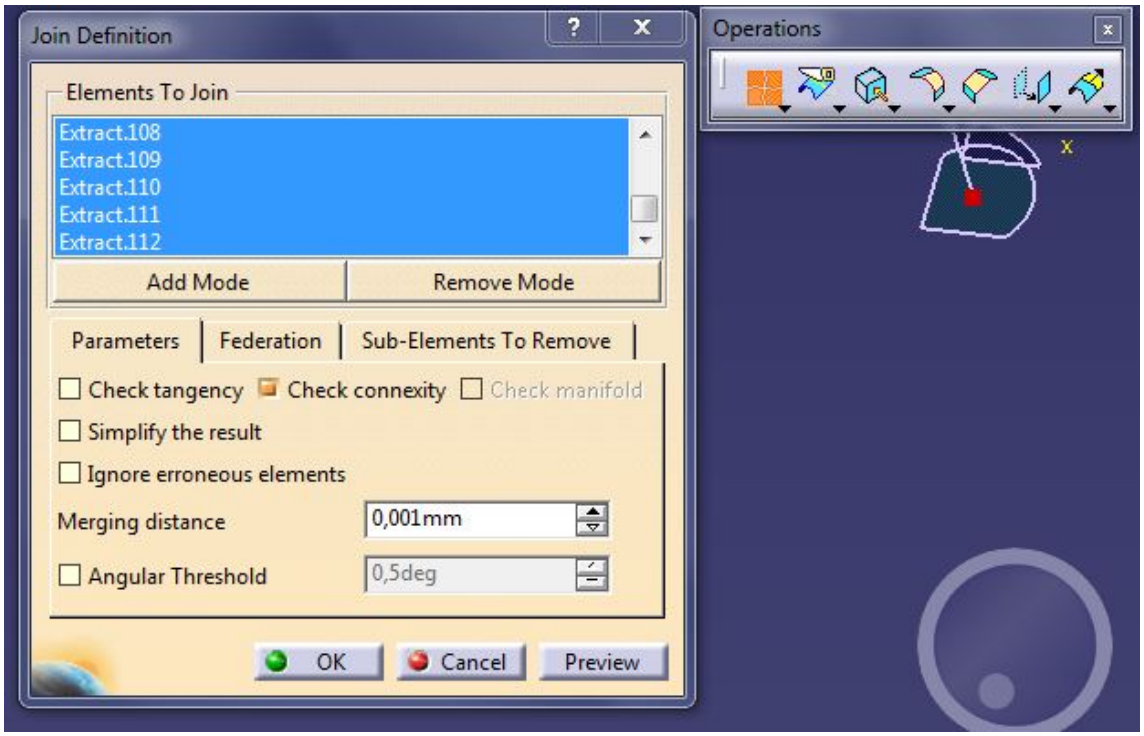
Figur D.4: När de stora är gjorda, zooma in och kolla efter mindre områden likt det ovan.



Figur D.5: Vid markering av sådana områden kan ovan felmeddelande förekomma. Välj att ta bort den senaste som blivit markerad och spara det området till senare. Välj Ok.

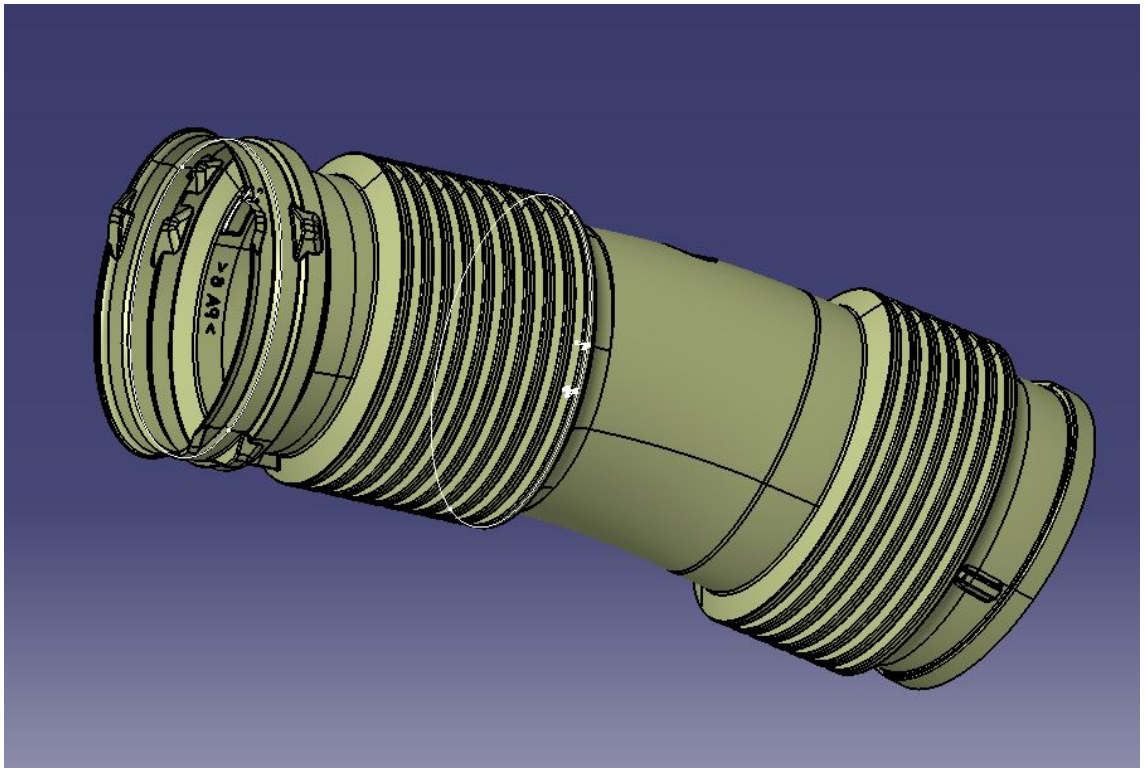


Figur D.6: För att ta de ställen som inte fungerade i de tidigare stegen, gå in i Extract igen. Välj Propagation type: No propagation. Klicka i den blå figuren jämte Element(s) to extract och börja markera de områden som inte tidigare blivit markerade. När klar, välj OK.



Figur D.7: I samma verktyg som Extract finns en funktion vid namn Join. Öppna denna.

Figur D.8: Markera alla Extract som ligger i trädet till vänster. Markera den översta, håll inne Shift-tangenten och markera den nedersta. Välj sedan OK i Join verktyget.



Figur D.9: Nu är ytan klar att exporteras.