Summer Research Report: An investigation into CAD/CFD meshing strategy

Duncan Weatherhead

October 6, 2012

Abstract

This report presents the work done within a summer research project, the goal of which was to identify contemporary problems within the industrial CAD/CFD interface and attempt to provide solutions to these problems. The report begins with a summary of the background research which helped to identify problematic areas. These issues were then subdivided into two main areas; issues with correct drawing practice in the case of direct import CAD/CFD interfacing and issues with defeaturing in the case of non-native file import. Following this the results of investigations into both of these cases are presented, concluding with an assessment of the further work needed to complete the proposed solutions in the case of non-native file defeaturing.

Introduction

For many years those operating within the CFD industry have been plagued by the difficulties that the CAD/CFD interface presents. In 1999, the Society for Industrial and Applied Mathematics (SIAMM) estimated that on average 80 percent of the man hours used up in the entire process of drawing a CAD model and completing a downstream analysis upon it were involved in 'cleaning up' CAD models to allow them to be imported into and meshed successfully within CFD programs [1].

With the use of CAD constantly evolving and spreading within all branches of engineering for the purposes of both design and analysis, it was felt that an assessment of the progress made since SIAMM's last study was important to discover not only areas of improvement but also areas where engineers were still finding problematic within this field. This was done by means of a series of interviews with several profesionals working in a variety of fields but all using CFD in some capacity. It was decided that the information obtained within these interviews were to form the basis of an overall picture of the current state of CAD and CFD within modern industry, and from this identify areas in which time and money saving progress could be made.

Background and interview findings

In the course of the acquisition of background research for this project, representatives of the following firms were interviewed:

- *Jaguar LandRover* - A high end automotive manufacturer
- *Hydro International* - A company specialising in water drainage products
- *Icon CFD* - CFD consultency using open source *OpenFOAM* CFD package
- *Totalsim* - CFD consultancy specialising in performance vehicle analysis

These firms are a suitable sample set of current industrial CFD users as they span many applications from large civil engineering problems in the case of *Hydro* to complex automotive jobs such as engine bay airflow simulation in the case of *JLR*. They also boast between them a good range of CAD/CFD combinations which should give a reliable overview of CAD/CFD use in wider industry to date. The varying capabilities and limitations of these systems are summarised and reviewed below.

CAD

It was found that CAD files were generated or received by the above firms in 2 main ways. The most popular of these means was the non-native file type. These tended to be `.msh`,`.stl` or `STEP` filetypes. These files describe geometry as simple surfaces defined by triangles or point data, and were found to be generated either by laser scanning existing solid parts to obtain discretized point-maps or by using conversion algorithms within CAD drawing software to convert drawn geometry from the native format of the program into one of these neutral types.

The second means was by way of direct import of CAD data from the package in which it was drawn into a CFD package, usually by means of a plug-in which linked the two software packages.

CFD/Meshing

Once again, a variety of packages were used amongst the interviewed firms for this, but their meshing strategies fell into one of two categories. Firstly structured extrusion meshing. This method relies on meshing a domain outwards from surface geometry provided within it. This often results in very high cell economy in terms of acurately describing surface geometry without the need for a very high global cell count but was also very dependent upon the quality of the surface provided to it. As a result users often found packages employing this technique to be quite temperamental.

In contrast to this the remainder of the meshers employed were of the cartesian cut-cell variety. These relied on a strategy of splitting a domain into a uniform set of cells prior to inserting the geometry, then deleting and snapping the existing cells to produce a mesh representation of the original part. This method was generally prefered as its robust nature relative to the surface geometry it is provided with greatly reduces preprocessing time. It is generally considered computationally expensive however, due to the high local cell density often required to capture small details in large parts.

Cleanup

Cleanup is a reference to the various processes that are currently implemented between the CAD and CFD stages in order to render the CAD geometry suitable for meshing. The processes here appeared to vary slightly depending upon the firm's software capabilities and the file types in use. It was generally found however, that native CAD files did not require cleanup and their manner of import was a very black and white affair; they would either work or have to be re-draw

In contrast, a large amount of CAD cleanup was often required for non-native file formats. This appeared to often be a result of poor geometry description on the part of the triangulation algorithms which often created duplicated surfaces or other such errors. It was also often the case, especially with files recieved from scans, that the provided geometry was far too literal for CFD. The high levels of detail within these models such as bolts,small holes or sharp edges in large parts were often found to have little bearing on the value of the simulated results, but resulted in incredibly high cell density being used to describe the minutae of these features. As such much CAD cleanup software was used to simply close, fillet or simply remove features such as these to provided a more 'CFD-friendly' geometry.

As such, firms had two major complaints about the state of the current CAD/CFD interface. Those that used direct import of native files prefered it but were very keen to promote good drawing practice in order to ensure these files were of a sufficiently good mathematical quality to be passed directly to the CFD package. Two particularly problematical areas identified in relation to this were the use of surface modelling tools and the use of CAD assemblies.

In terms of non-native filetype users, defeaturing arose as the primary costly area that all firms wished to see removed from downstream analyses. *JLR* estimated that for, a scan of one of their vehicles, it took a trained professional close to 5 days to tidy up CAD geometry to a state at which it could be successfully passed to their CFD software.

As such two goals for the project were defined:

- Use a combination of the *Solidworks* CAD package and *ANSYS* CFD package to investigate import problems caused when using surface modelling and assemblies within CAD in a direct import environment.
- Use the open source CFD package *OpenFOAM* to investigate the possibility of generating a defeaturing algorithm that could be implemeneted within the package itself, thereby negating the requirement for manual defeaturing prior to meshing.

Native file types

Surface modelling

In recent years many advances have been made in the field of CAD programing. This has seen it surpass its beginnings as a simple drawing tool and allowed for the generation of models with great analytical potential in a number of engineering applications including CFD, FEA and Rapid Manufacture. One of the foremost advances here has been the development of a number of surface modelling tools. Much less restrictive than their solid modelling counterparts, these tools allow for the freeform modelling of many complex and irregular geometries, making them a powerful tool in the arsenal of the designer.

In order to understand why one would use surface models over solid models it is important to understand the differences between the two. With this in mind the following definitions have been provided:

"freeform surfacing is used in CAD and other computer graphics software to describe the skin of a 3D geometric element."

"solid modelling is a consistent set of principles for mathematical and computer modelling of three-dimensional solids. Solid modelling is distinguished by its emphasis on physical fidelity."

Thus, solid models are generally used to generate models for mechanical analysis. The rigid parameterization of their mathematical structure gives them a very good basis for downstream analysis, but renders them rather stiff when it comes to aesthetic design. As a result, surface modelling is still widely used in the fields of automotive, aeronautical and naval design amongst others.

Surface models are essentially connections of NURBS (Non Uniform Rational B-Spline) curves which are defined by a series of control points along their length. These control points can be added or deleted as required and manipulated in all 3 dimensions. It is this mathematical freedom of definition that makes surface modelling so attractive to those designing non-conventional geometry. They can have mathematical definitions added to them to make them more sound but do not require them to exist.

Whilst conducting the research that has lead to this report a number of deficiencies were found when it came to importing these models into CFD packages. For the purposes of this study two different CFD packages were used; *ANSYS Workbench*, a commercial closed source CFD code with a direct plug-in to *Solidworks*, and *OpenFOAM*, an open source code accepting non-native file types such as `.stl` and using a cartesian cut-cell mesher: `snappyHexMesh`.

Firstly, discussing the direct import into *ANSYS* it was hoped that this would be very accepting of surface models given its direct connection to the geometry. Many CAD/CFD interfacing problems are caused by converting the files into multiple formats along the way and so removing this step was expected to eliminate many of these problems.

This was sadly not found to be the case. It was found that pure surface models, as simple as a cube, could not be analysed correctly. Whilst a mesh of the geometry could be generated for most cases, this mesh would not correctly transfer to the solver, *Fluent*, and as such resulted in the flow simply being calculated for empty domains.

Given the failure at this stage to mesh any of the geometry that was supplied to a transferrable standard, attempts were then made to convert the surface files into `.stl` files in order to transfer them to *OpenFOAM* for analysis. It was hoped in this case that, given that the `.stl` conversion was to be done in *Solidworks* itself, this process would be less dependent upon the CFD program itself and therefore possibly more reliable. Sadly, once again, it was impossible to create `.stl` files of pure surface models, with *Solidworks* claiming that they were of the wrong filetype for that opperation.

On this evidence, it would appear that surface modelling has no place in the toolbox of those creating models for CFD. Actually this is not the case, but it is obvious that surface models cannot simply be generated on their own as solid models can to be simply passed on from program to program. There are a number of methods which need to be used, and indeed practices to be observed to make surface models suitable for downstream applications.

Surface knitting

Surface knitting is the action of unifying two or more surfaces with adjacent edges. This goes some way to eliminate the gaps that occur between ajacent surfaces as shown in the following example:

Case 1: Quarter-disc case In this example, a quarter of a disc has been drawn by drawing half a right angled arch and then a symmetrical, leaf-shaped surface. Both are planar surfaces with the resulting profile then being extruded, and finishing with both original planar surfaces being mirrored about the centre of the part in order to create the far side (figure 1). Even in *Solidworks* if we zoom in we can see that there are evident gaps between the two surfaces, despite the fact that they share the same sketch spline (figure 2).

Figure 1: Quarter disc

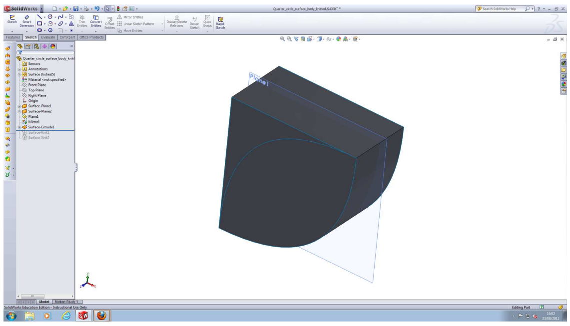
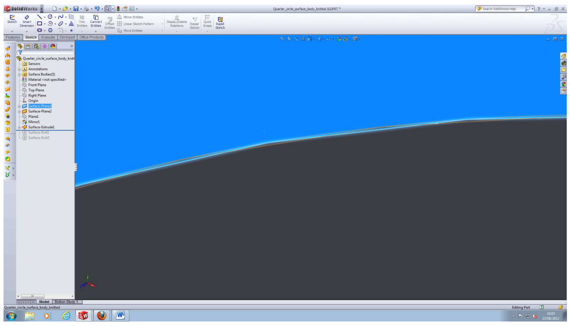


Figure 2: Gaps between surfaces



These gaps will be enough to prevent the tesselation algorithm within the software generating an `.stl` model of the part as it requires the continuous geometry usually associated with a solid model. There is a way of dealing with this however. By knitting the surfaces together it is possible to minimise these gaps to a given tolerance as can be seen in the left of the image below. Whilst purely knitting surfaces is not usually enough to allow `.stl` generation, there is another knitting option available that allows the user to convert the geometry to a solid body (figure 3).

Once this has been done, the surface body now posseses all the qualities of a solid body and can be successfully exported as a native *Solidworks* file or converted to an `.stl` as seen here (figure 4). In this case it has been possible to combine the best features of surface and solid modelling using the knitting feature.

Surface thickening

Whilst knitting can be a very effective solution, it is not foolproof. Often the surface geometry, if sufficiently complex or containing too many large gaps, cannot be converted into a solid body simply by knitting all the surfaces and then converting. Occasionally it is possible to knit groups of surfaces

Figure 3: Knitting options

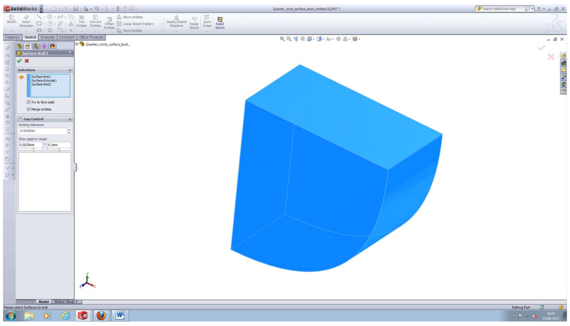
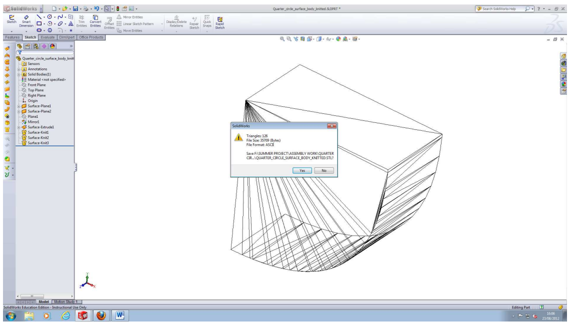


Figure 4: Final .stl model



and then knit these knits together to achieve solidity, but again this approach is largely hit and miss. In this case another tactic must be adopted, one refered to as surface thickening.

As its name suggests, this tool literally allows the user to add thickness to one or both sides of a surface. This effectively turns the surface into a very thin solid body and so inherits all the relavant structural integrity that it needs to undergo downstream analysis. At the same time, it is often considerably quicker when modelling complex geometry to create a surface and then thicken it than it is to trying to sketch profiles for the equivalent solid loft or by trimming a solid extrusion. It should be noted at this point, however, that thickening is not always an option with significantly complicated or ill-defined surfaces.

Case 2: Torus case In the last case, the part could quite easily have been modelled as a solid extrusion, thus circumventing all of the associated surface modelling issues. In this case, the freeform nature of surface modelling design begins to come clear as a better solution for drawing complex geometry. This case involves a torus which is designed to transport two fluid seperately in a helical fashion.

It has been created by drawing the profile of a semicircle which has then been swept along a circular path for 180 degrees whilst also undergoing a 180 degree twist. The resulting sweep has then been mirrored about the axis that runs diametrically through the circle, with inlet and outlet tubes created as surface extrusions of a circular profile.

Examining the image below we can see that a hexahedral mesh was created in *ANSYS* of the geometry (figure 5) but would not transmit successfully to *Fluent* in order for an analysis to be completed. The second image shows the same torus with the exception that all of the surfaces have been thickened by 10mm (figure 6). It was then possible to mesh the geometry and the data was passed successfully to *Fluent* for analysis.

Figure 5: Surface torus

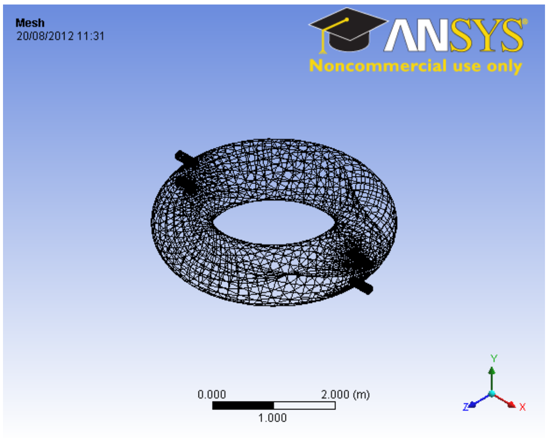
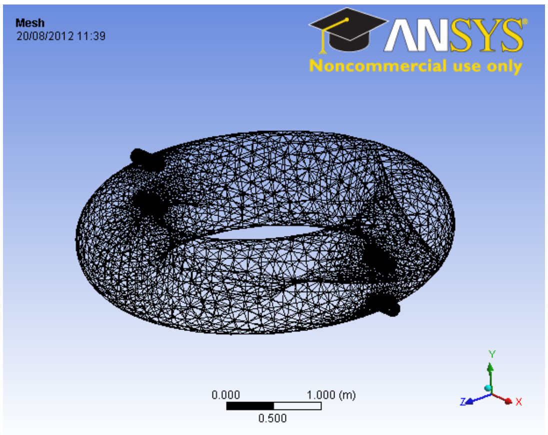


Figure 6: Thickened torus



Lofted surfaces

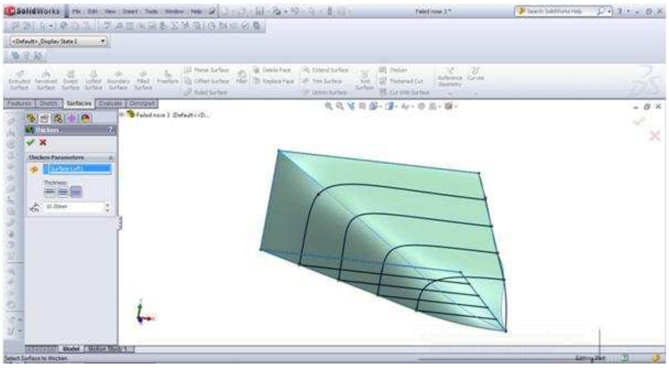
The discussion of these issues comes to a head within the issue of lofted surfaces. Lofted surfaces are the most freeform of surface design tools with the minimum ammount of constraints applied to them. As such this makes them the ultimate design tool, especially for aesthetically pleasing geometry, but also makes them the most mathematically unstable and therefore the most likely to cause meshing algorithms to fail.

This instability can also render them incapable of cooperating with either knitting or thickening tools, due to their largely undefined relationships to adjoining surface geometry and poorly defined normal directions. As such, drawing lofts that work is often down partially to luck but developing correct and mathematically efficient drawing practices can lead to a much better chance of thickening or knitting a lofted surface.

Case 3: Car nose case As an example of this, an industry-relevant case is presented here. This is of half a nose cone for a racing car, similar to those use in Lemans-esque endurance races. The reason that only half of the nose has been drawn is that cars are often only drawn in half down their central line of symetry and then the features mirrored to minimise drawing time (a generally useful tactic for drawing any vehicle or similarly symmetrical item).

In the first example here (figure 7), the designer has attempted to utilise the freeform nature of surface design by drawing the entire cone as a loft consisting of a series of splines. Whilst this has allowed the designer to create quite a fluid form, some issues can immediately be identified with this method.

Figure 7: Unsuccessful nose loft

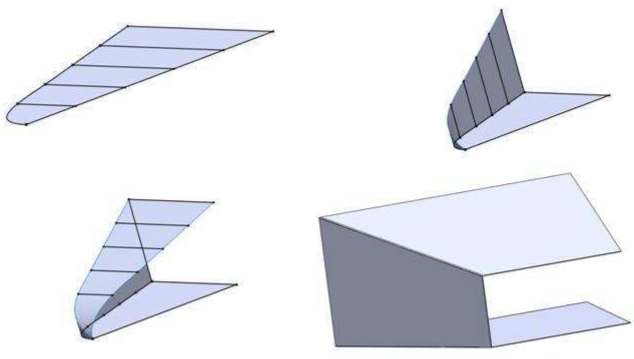


Firstly observation of the area at the end of the cone shows a demonstrable deviation from the guide curve in the actual surface geometry created. By using single splines with many variations and control points or sharp angles, the already vaguely mathematial definition of the surface bounds have been further complicated and so the program is having difficulty fulfilling all the boundary conditions it is being given. This is a common feature of lofts that involve too many, or overly complicated splines.

The other problem is that the mathematical vagueries which define these surfaces have rendered them incapable of being thickened and so will prevent this part from being transferred downstream as a solid model.

In the second example, the designer has gone about the task in a more methodical manner. Here the geometry has been created in a piecewise manner, first the base, then the side and finally the top. This means that the lofts have been created as a series of vertical and horizontal lines rather than splines. As a result the model is much more mathematically sound and the surface geometry has been created successfully. The final image presents a section view of the finished part to demonstrate that it has been thickened and also filleted at the edge to include the curvature present in the first example 8.

Figure 8: Successful nose loft



Thus in this example it has been shown that lofts can be problematic, but with some careful thought into the geometry of the part, do not need to cause potential meshing failure. It is often better to take the time to draw something in a more mathematically sound manner as, whilst it may take longer at the drawing stage initially, it can save much head-scratching and re-drawing down the line.

Assemblies

Due to advances in CAD technology, research has revealed that many mated assemblies can be successfully analysed downstream, regardless of how poorly or vaguely defined they may be. Regardless of this however, there are still some issues that prevent some assemblies being transferred downstream. The following examples attempt to demonstrate some of these remaining issues and show how to overcome them using good modelling practice.

Case 4: Tangent mates Tangent mates are often an acceptable way of locating tubes and other cylindrical objects relative to flat surfaces, a common case in engineering systems in everything from pipework jet aircraft and spacecraft. Thus as a simple locating tool then tangent mates are very useful but they do present problems when attempting to transfer these models downstream.

In the following case an attempt was made to analyse a simple model of a cylinder running adjacent to a plate. As can be seen in the figure below, the assembly is joined by a tangent relationship (figure 9). However, the figure below it shows that, in an attempt to import this into the CFD program *ANSYS Workbench*, an error occured citing the reason as 'zero thickness geometry' (figure 10). Zero thickness geometry occurs when two entities are not sharing an edge. In this case it represents the infinitely small contact point between the cylinder and the plate.

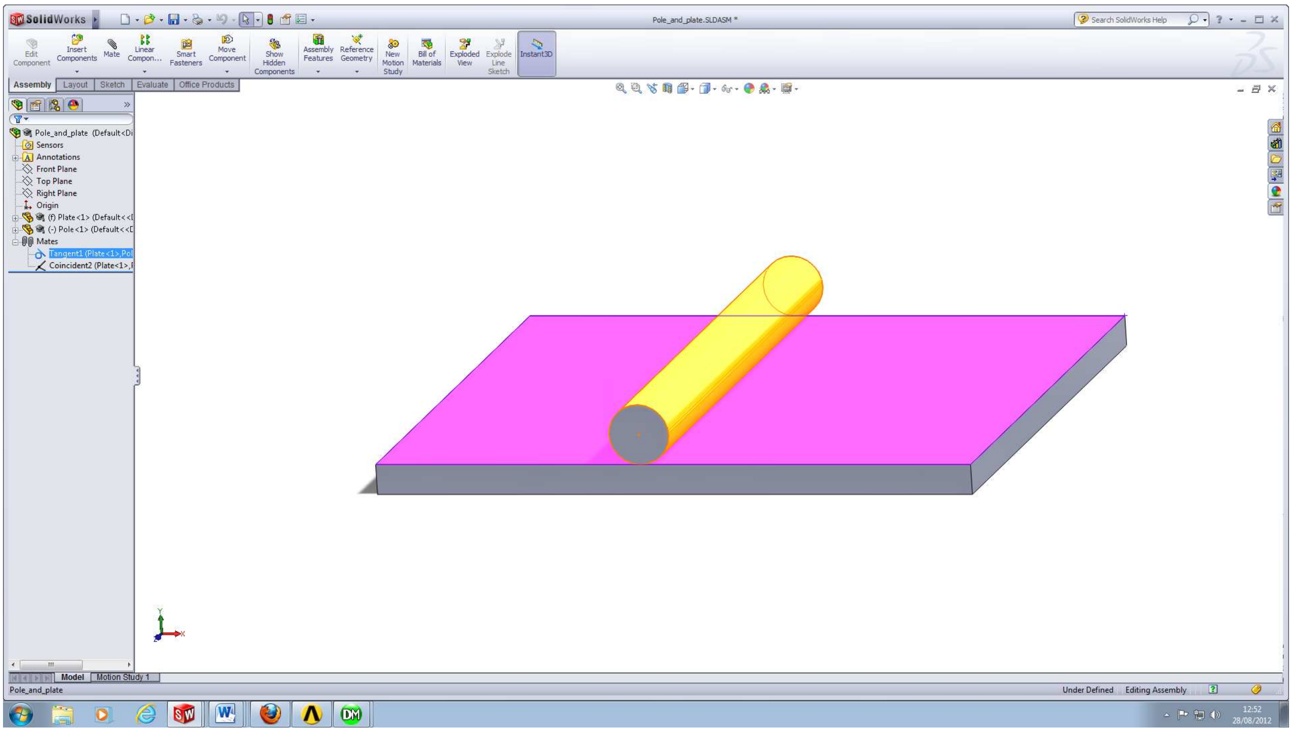


Figure 9: Tangential relationship

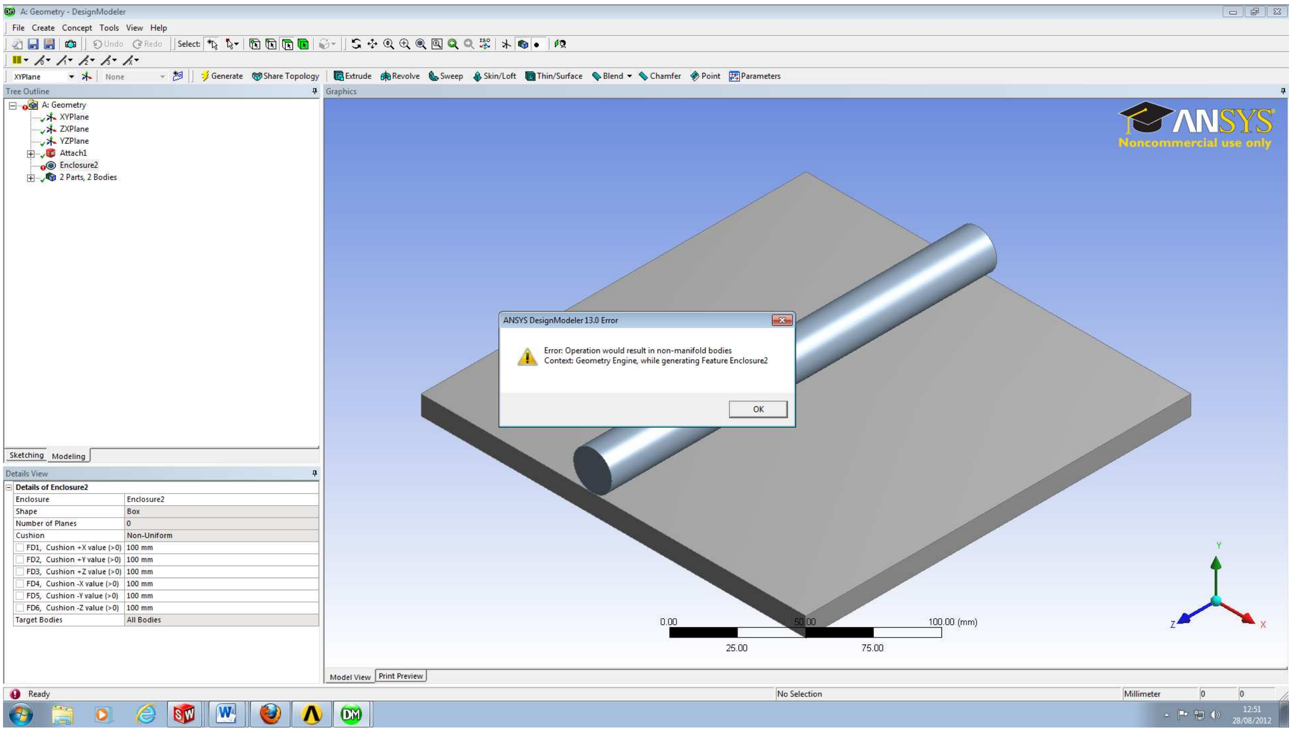


Figure 10: Import error

The reason why this does not work for CFD applications is simple. As the program attempts to mesh the geometry the program will try and fit increasingly small cells into this gap until the cells effectively get infinitely small and will result in a failure of the meshing algorithm.

Thus, whilst structurally very useful, it is often impossible to use them in models intended for downstream applications. As such ways around this must be found that retain as truly as possible the geometry of the original design but at the same time allow for the meshing requirements of the CFD program.

In this case, as the figure below (figure 11) shows, this has been achieved by creating a small ammount of virtual material at the point where the tube contacts the plate. This allows a coincident mate to be used instead of a tangential mate and provides a definite boundary which the meshing algorithm can adhere to.

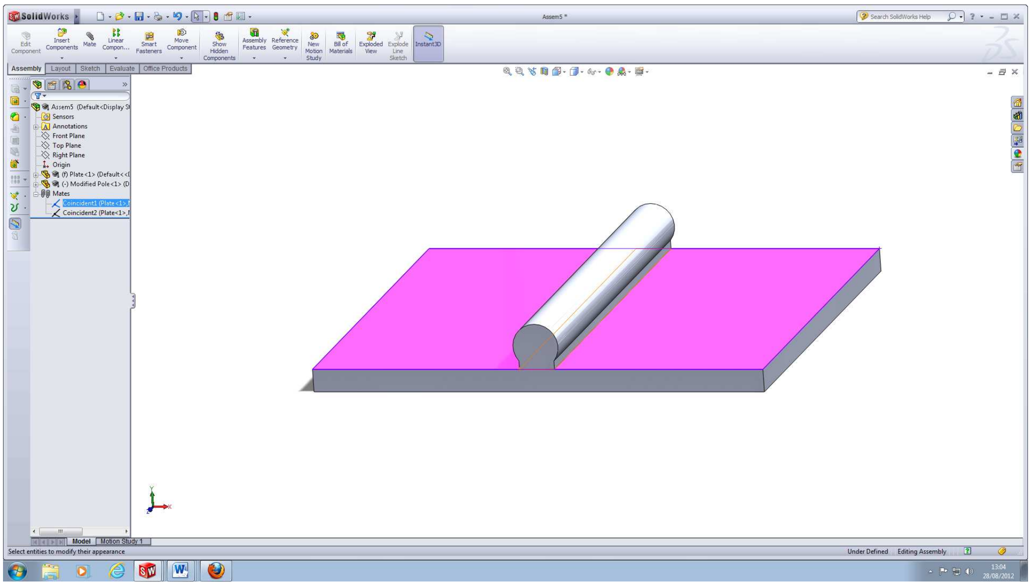


Figure 11: Model including extra material

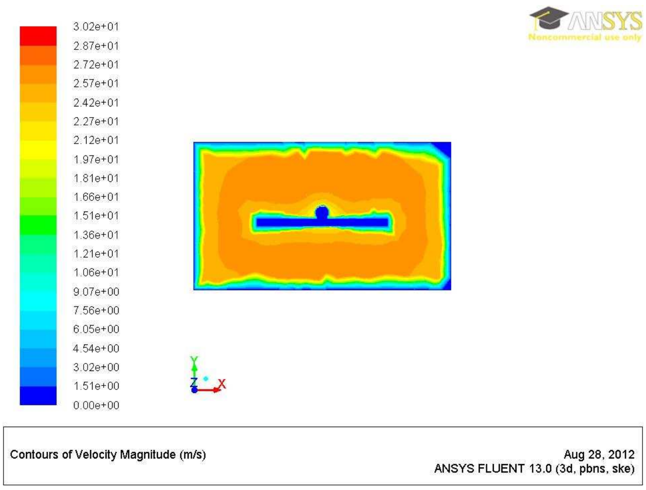


Figure 12: Velocity profiles around tangent example

Case 5: Concentric mates and tolerances In many engineering systems, especially those drawn with manufacture in mind, the concept of tolerances is one that is very important. In terms of machining interacting parts it is very difficult, even with modern manufacturing techniques to manufacture two parts exactly the same size. As such, tolerances need to be put in place when drawing parts that take into account these manufacturing deficiencies.

In this case, an internal and external tube are mated by means of a concentric mate which involves one slotting inside of the other. The diameter of the internal tube is 149mm and the internal diameter of the outer tube is 150mm. This would be a common tollerance for a relaxed sliding fit in many engineering systems such as shock absorbers. When transferred to *ANSYS* this resuted in a catastrophic divergence in the solver.

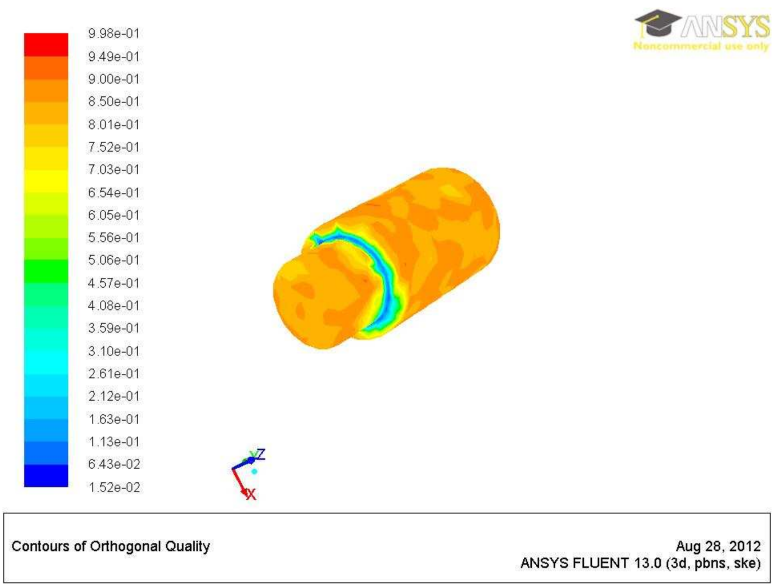


Figure 13: Contours indicating orthogonal cell quality

An answer to why this happens is can be found in an *ANSYS* scan of the orthogonal cell quality of the overall mesh (figure 13). Here we can see that the exeternal area of the tube surface mesh contains some very poor quality cells which, it was suspected, contributed to the failure of the solver in this case. In order to improve this the diameter of the inner tube was increased to 150mm to be flush with the outer tube and the analysis repeated.

An immediate difference can be seen in the quality of mesh if we compare the inital mesh ’a’ (figure 14 with the new mesh ’b’ (figure 15) we can see a comparative lack of complexity in the second mesh as the meshing algorith has not tried to cram cells into the tiny gap between the two tubes.

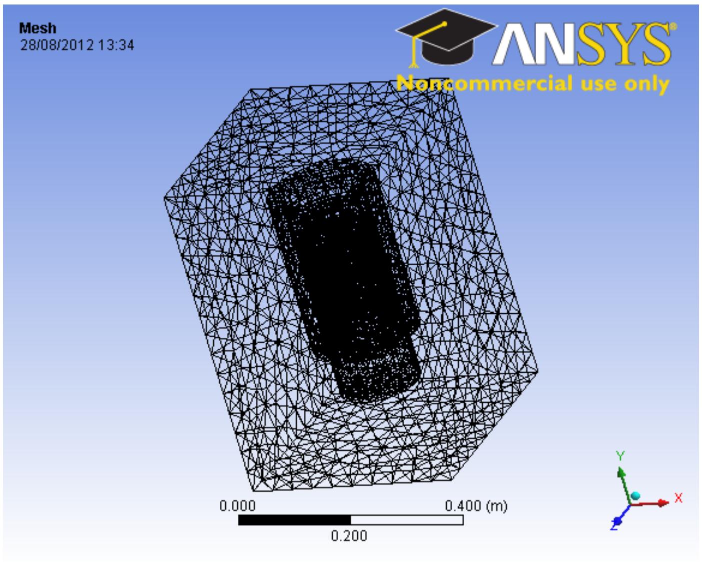


Figure 14: a)Mesh of initial geometry

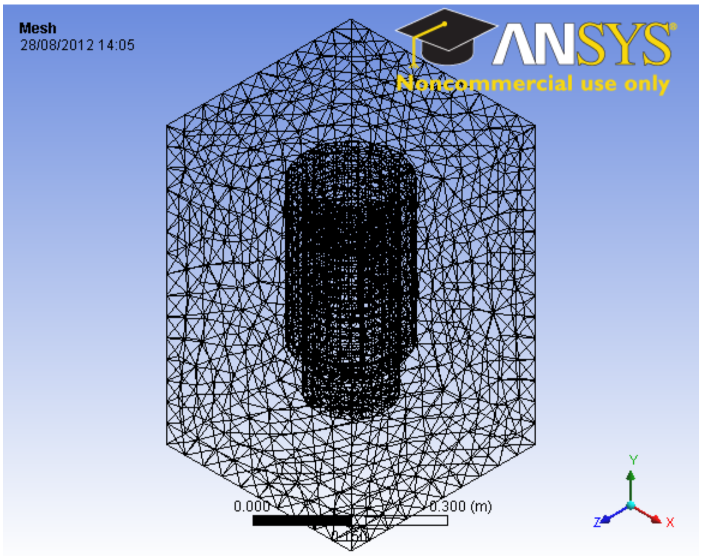


Figure 15: b)Mesh of improved geometry

As we can see below (figure 16) this allows for a complete and successful analysis of the part, and a further demonstration of how using assemblies designed for manufacture or other purposes is not always suitable for CFD analysis due to the excessive level of detail that they contain.

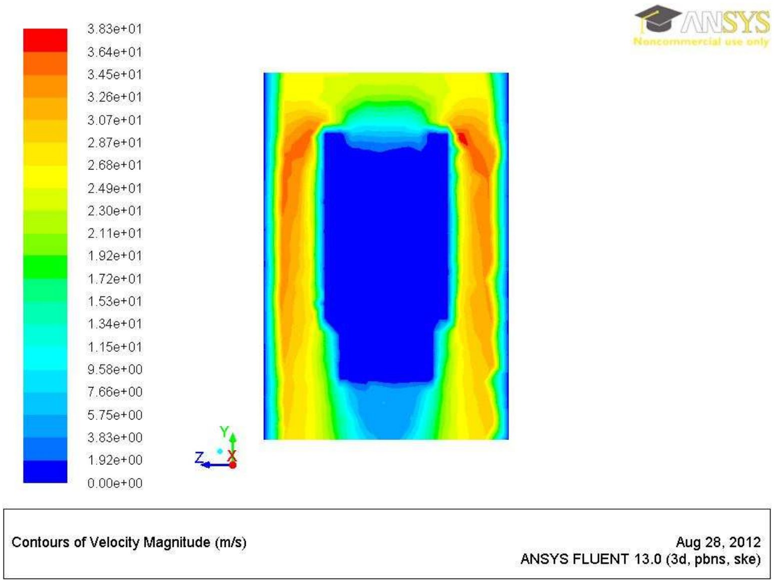


Figure 16: Velocity contours around sliding cylinders

Case 6: Angled mates Finally, this case attempts to demonstrate the problems experienced when assemblies are insufficiently or poorly defined assemblies, especially cases when incorrect mates or too few mates are used.

This case presents a simple assembly in the form of what is essentially a shelf bracket. The bracket consists of an 'L' shaped main piece with a cross piece set about halfway up. The constraint holding this together is simply an angular mate between the main body and the cross piece set at 45 degrees. Other than this the cross piece has been placed 'ad hoc' in the centre of the 'L' which, visually at least, has it resembling a complete part (figure 17.

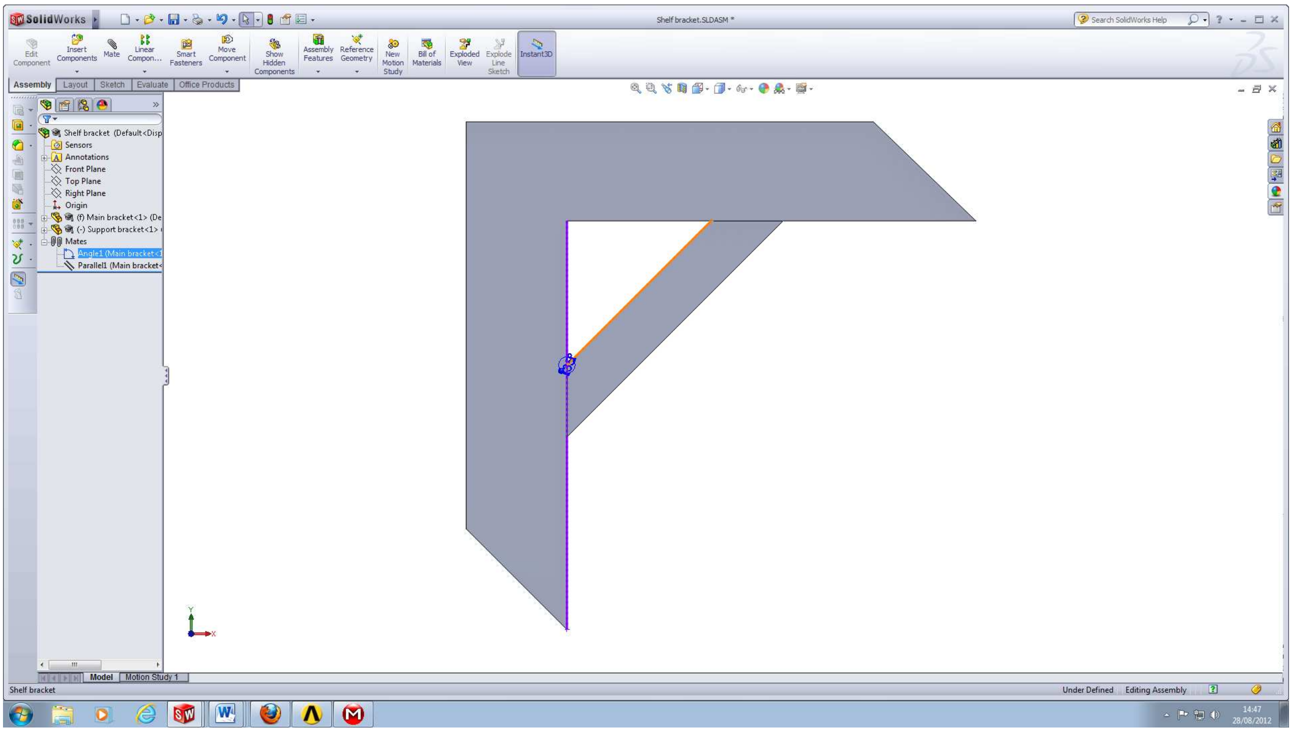


Figure 17: Shelf bracket model

When an external analysis was undertaken, however, this experienced a divergent solver failure similar to that of the previous case. Again a simple analysis of the mesh quality highlights several areas populated with low quality cells at the intersection between the two bodies (figure 18. This is a classic case of the mathematics of the model have not been considered during the drawing phase. As previously stated, the model appears visually intact but is obviously not mathematically so.

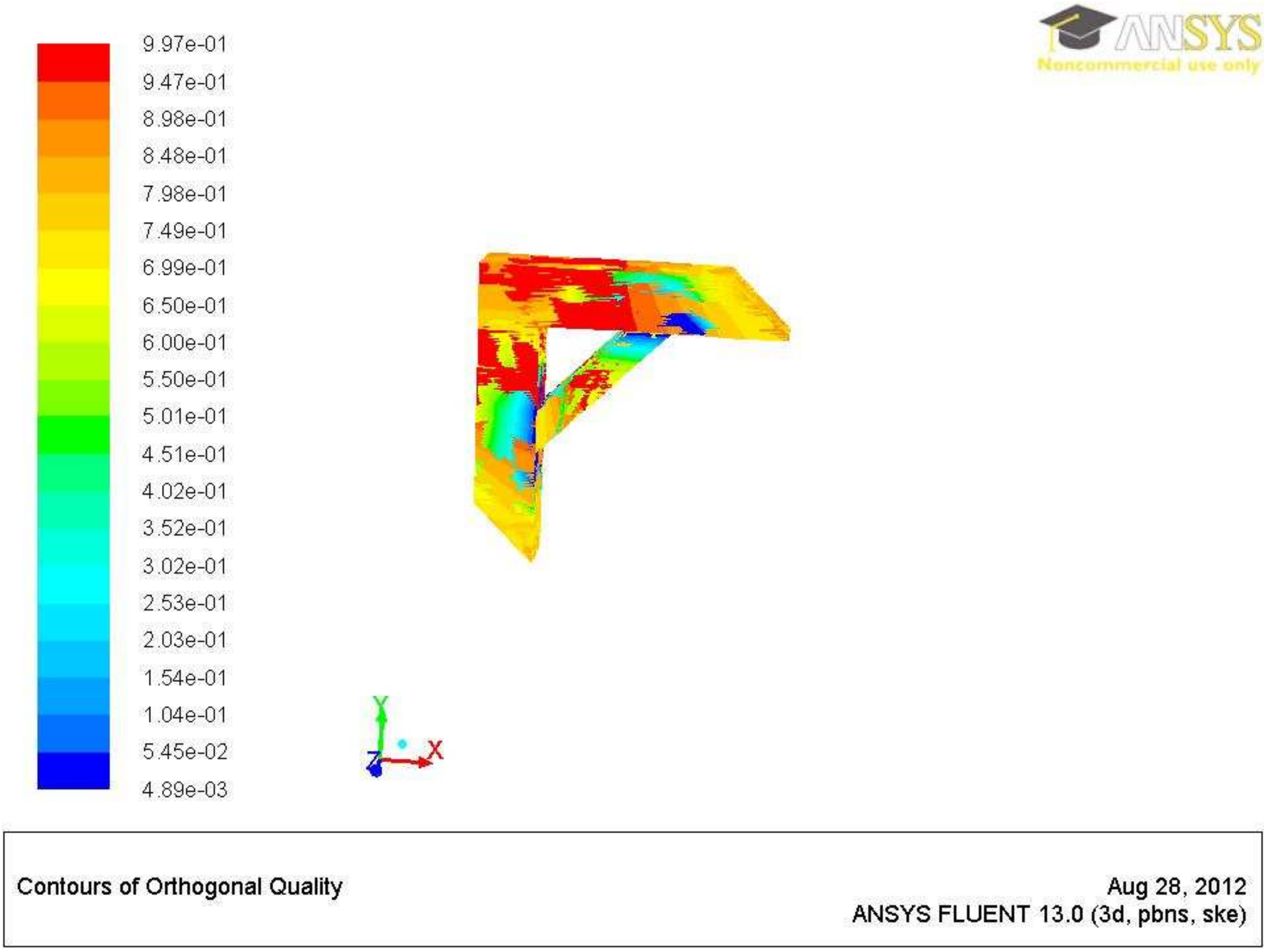


Figure 18: Poor surface mesh quality

This is demonstrated by the introduction of some other mathematical detail in the form of coincident mates between the ends of the crosspiece and the internal faces of the bracket. The results of this are a complete analysis with consistent mesh quality, and thus show the importance of taking some extra time and care during the drawing stage to ensure models are mathematically sound for downstream analysis.

Non-native file types

In this portion of the research, efforts were made to construct an algorithmic de-featuring tool to be implemented within *OpenFOAM* to allow defeaturing of geometry to be a part of the meshing procedure. It was hoped that by doing this it would reduce the ammount of time spent by engineers manually defeaturing parts prior to CFD analysis.

Theoretical premise

After studying available data sets within *OpenFOAM's* meshing tool; `snappyHexMesh`, the following concept was developed for a defeaturing tool.

Consider a flat plate with a small hexagonal deviation at it's center as shown below 19.

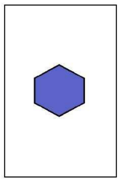


Figure 19: Flate plate with hexagonal deviation

As with any meshing tool, the deviation will result in a much denser mesh being generated in that particular area. Given that this is a cartesian cut-cell mesher, it is most likely that the algorithm will generate a regular hexahedral mesh as much as it can over the remainder of the plate. Thus, a cartesian plot of the cell center locations would result in something approaching the figure below 20.

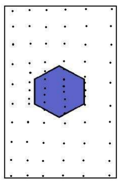


Figure 20: Cell center plot of flat plate

This best illustrates the increased cell density around the deviation, and it is this increase in cell density that is symptomatic of the features that are unimportant to large scale analyses and thus should be removed by the de-featuring tool.

From a mathematical standpoint, given that the mesh is created within a cartesian reference frame, all of these points will have an x, y and z coordinate. Thus, by averaging the distance between the relevant coordinates of all of these points, it is possible to gain a mean spacing value, δx,δy etc, for the coordinates of all the cell centers in all 3 dimensions. This gives a metric for the current overall cell density of the mesh for that particular surface.

Given this information, the proposed tool would allow the user to set acceptable upper and lower limits for these δ values, both overall and between individual cells. Once the mesh has been created it would then be possible to scan through the list of cell center points and identify pairs or collections of cells with values of δ outside of the limits. An instruction would then be written to the program to re-mesh these areas to a slightly lower set of tolerances, effectively smoothing out and locally erasing the small geometrical deviation and thereby reducing unnecessary cell numbers.

Practical Implementation

In order to complete this a function was implemented that was able to write out a list of the cell centers for those cells making up any patch or suface within the mesh. This was important is it meant that only the relevant cells were included in any scanning and re-ordering that followed. Furthermore, the intention was to run the tool at the castelated mesh stage, i.e. before the effects of the surface truncation had had a chance to affect the structure of the remainder of the mesh.

On inspection however, it was discovered that the function was writing out the point data in a geometrically random fasion. This meant that while it was possible to gain mean values of δ for the entire plate in all 3 dimensions, it made scanning through the list to identify geometrical imperfections impossible as the chance of enough neighbouring points apprearing next to each other in the list is virtually zero.

As a result, in order for the idea two potential solutions have been proposed. The first being to produce an algorithm to re-order the list written out by the program so that all the cell coordinates are in neighbour order. This is likely to be largely time-consuming however as many surfaces in even a coarse list consist of thousands of cells. The alternative is to examine the original OpenFOAM code and attempt to amend it in such a way that the program writes out the vertex list in neighbour order.

Conclusions and Further Work

In conclusion this project has demonstrated that the state of the CAD/CFD interface has improved significantly in the past 10 years. Nonetheless it still presents problems to engineers today that serve to slow down, significantly in some cases, industrial CFD analyses. It has been shown that whilst direct import using native CAD formats is the most efficient method of interfacing, but it relies very much on the quality of model creation and is also not always possible if files are being passed between firms. Non-native filetypes do allow for this and are still very much the standard currency for inter-firm and even inter-departmental CAD file transfer. These do often present many cleanup issues, often in terms of defeaturing or other forms of digital 'tidying'. This research proposes the definite possibility of modifications to current meshing algorithms which could lessen the need for these time-consuming task. Further work could expand on this projects investigation into surface based defeaturing by further investigating the way in which *OpenFOAM* writes out lists of its mesh points and to try and produce this in a more ordered fasion. This would potentially allow for the generation of algorithms to scan the surface mesh and re-mesh deviations by means of cell center density identification.

Bibliography

[1] Rida T. Farouki. Closing the gap between cad model and downstream application, 1999.