

# Advanced geometry tools for CEM

---

## Introduction

Modern aircraft designs are extremely complex CAD models. For example, a BAE Systems aircraft assembly consists of over 30,000 individual components. Since CAD systems generally do not have physical simulation in mind, these components have correct shapes and locations, but have no record of whether they are physically connected. An old aerospace joke states “an aircraft is just a bunch of parts flying in close formation,” but this is precisely how an aircraft is represented in a CAD system.

For geometry to be useful for computational electromagnetic (CEM) simulation, it must be:

- Correctly connected – components with an electrical connection in real life must be mathematically connected in the CAD model
- Appropriately detailed – CAD models typically contain full manufacturing detail, which is excessive for CEM simulation, and would cause the simulation to be prohibitively expensive
- Meshable – before geometry can be used for CEM, it must be converted to a tetrahedral mesh, which requires that the geometry be free from certain kinds of defect – for example, short edges, small faces, and sharp corners.

Taking a CAD assembly, and generating a useful CEM geometry which satisfies these criteria, can be a labour intensive process. Within the ICE NITe project, we have developed new tools to improve the connection between CAD and CEM by automating some of the common steps required to prepare a CEM model:

- Robust imprinting
- Complex welding
- Automatic fillet removal
- Feature suppression
- Advanced face joining

The remainder of this report will provide details on the algorithms, methods, and tools we have researched, developed, and implemented within CADfix to perform these steps.

## Robust Imprinting

Imprinting is the process of finding faces in the CAD model which are in contact (to within a given tolerance), and splitting them to ensure each edge on one side of the contact has a corresponding edge on the other side (figure 1). This is the first (and most difficult) step towards joining touching bodies together, with a shared face between them, which is the preferred representation of contact for CEM simulation.

Imprinting in CADfix is quite an involved process involving several topology/geometry inquiries followed by topological manipulation. The first step is to determine where imprinting needs to be

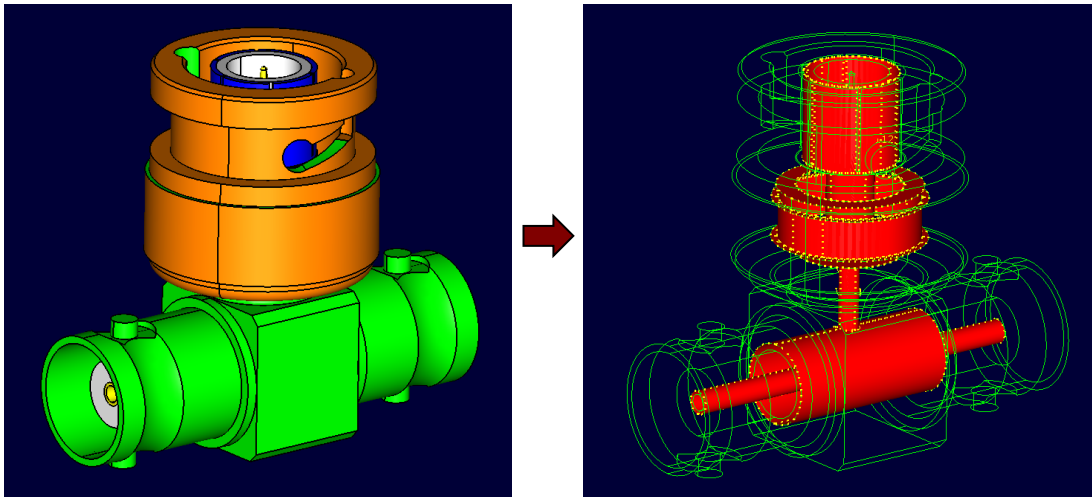


Figure 1. Imprinting on a simple BNC connector

applied, as determining this manually would be prohibitively expensive. Given a set of faces (or bodies), we find out which pairs of faces overlap each other:

- First, we make sure that the underlying surfaces of a given pair of faces are within tolerance, within the boundaries of the faces.
- Then we only need to make sure that there exists a point inside face1 which is also inside face2, to classify them as overlapping.

Once we've collected the overlapping faces, we then split the boundaries:

- We collect the boundary edges of all faces and split each pair of intersecting edges, while updating the parent face edge references.
- Splitting intersecting edges must be performed very carefully. Often, edges will weave across each other, due to unintentional differences in their geometric definition, and the edge splitting needs to handle these cases. Cusps also require special attention, to ensure the split point is placed correctly.

We then look at the vertices of all edges and try to merge them if they are within tolerance. The simplest method is to simply merge vertices which are within tolerance, but this gets more complicated when there are short edges (edge length < merge tolerance). These are common when there are errors in the CAD model, and bodies are slightly misaligned. Although such edges are undesirable in the final CEM model, collapsing them during the imprinting process produces unexpected results such as incorrect topology (high connectivity) or leaks (low connectivity) – it is better to remove them afterwards using collapse or join operations. Instead, we break down the vertex connections at the short edge, and reconnect them to give a "snap-into-place" effect (figure 2).

Once vertex merging is complete, we merge any edges which are within tolerance. We then reconstruct the face loops using the remaining edges:

- If the loop of an existing face has been split by several other edges, they can trace out a new closed boundary.

- We work out this boundary, and determine if the closed boundary has material inside it or not. If yes, we make a new face (the imprinted face), and if no we discard it.

The model is now properly imprinted, but faces can have several shared edges. The edge topology is unshared so that each edge is connected to only two faces, and final clean-ups are performed – healing of lines to the intersection of their neighbouring faces, collapsing of sliver faces, and incorporating orphan faces into body definitions.

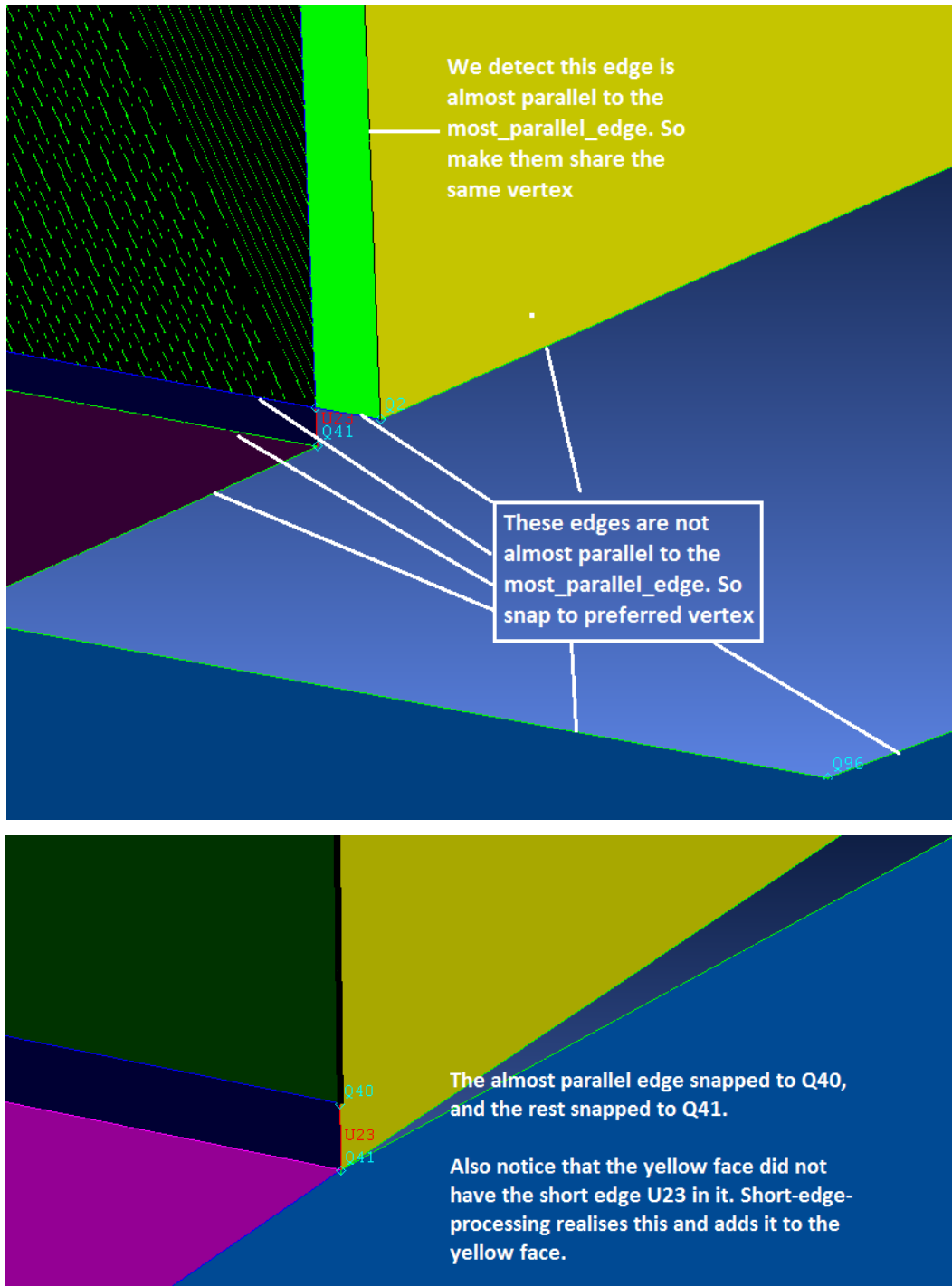


Figure 2. Example of merging in the presence of a short edge.

## Complex welding

CAD models are often developed to support manufacturing, and as such do not contain every component in the physical product. Small items like washers and shims, or filling and sealing materials, may be specified by metadata, rather than explicitly modelled with CAD geometry. As a result, the CAD model may contain gaps between bodies which are connected in the manufactured object.

The CEM model must be properly connected, and so these gaps must be filled to bridge between bodies. We call this process *welding*, as it is conceptually similar to connecting metal parts using a welded bead – and indeed, CAD models of welded objects often omit the welds themselves, requiring precisely this operation before they can be analysed.

Earlier versions of CADfix contained a weld generation tool, but this was a manual process, requiring the user to select the faces to weld between, working body-by-body. This cannot scale to a complex aircraft assembly, as the user cannot be expected to even identify every potential gap. Therefore, research in ICE NITe has focussed on the automatic detection of weld gaps, and the subsequent construction of welds to fill them (figure 3).

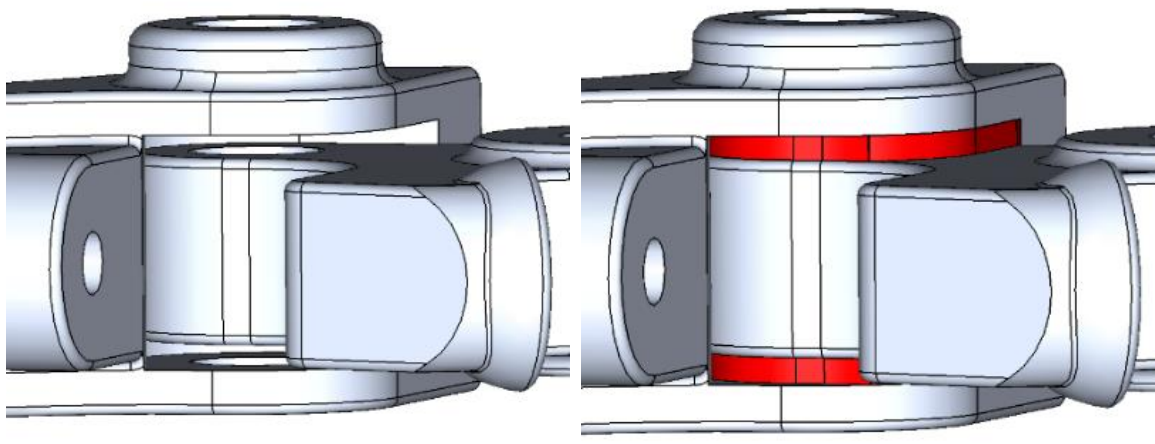


Figure 3. An example weld (shown in red) between two components in a CAD model. The gap is detected, and the weld is constructed, automatically.

The weld gap detection starts by constructing a faceted representation of the model. The facets are placed in a spatial index, which allows efficient discovery of facets near a given point, and facets which intersect a particular ray. For each facet, we can then determine other facets which are within the maximum weld gap distance, and compare normals to establish that the facets face each other, and can be paired. Finally, a ray is fired from each facet to its pair to check that the candidate pairs are not occluded by other pieces of geometry.

There is a risk that naïve construction of a weld body could introduce hard to mesh geometric features into the model, such as short edges, and narrow faces. To alleviate this, snapping has been added to the weld construction (figures 4, 5), to ensure the weld bodies re-use existing edges and vertices in the CAD model where possible, and do not create CAD topology below a target size.

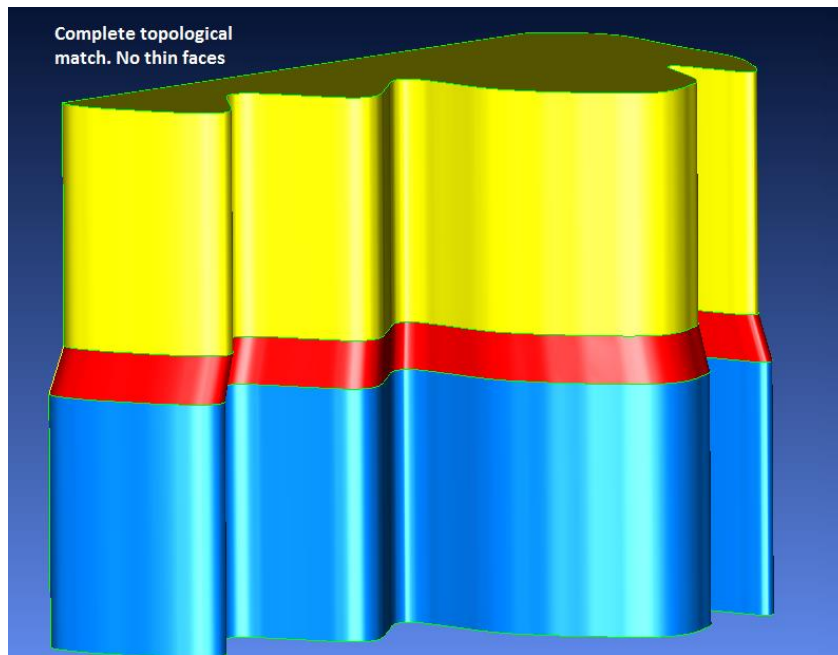


Figure 4. A weld between two matching bodies. Note that the weld is not a simple extrusion of one body towards the other, but instead has snapped to the boundary curves of the two bodies, to avoid introducing small CAD faces or edges.

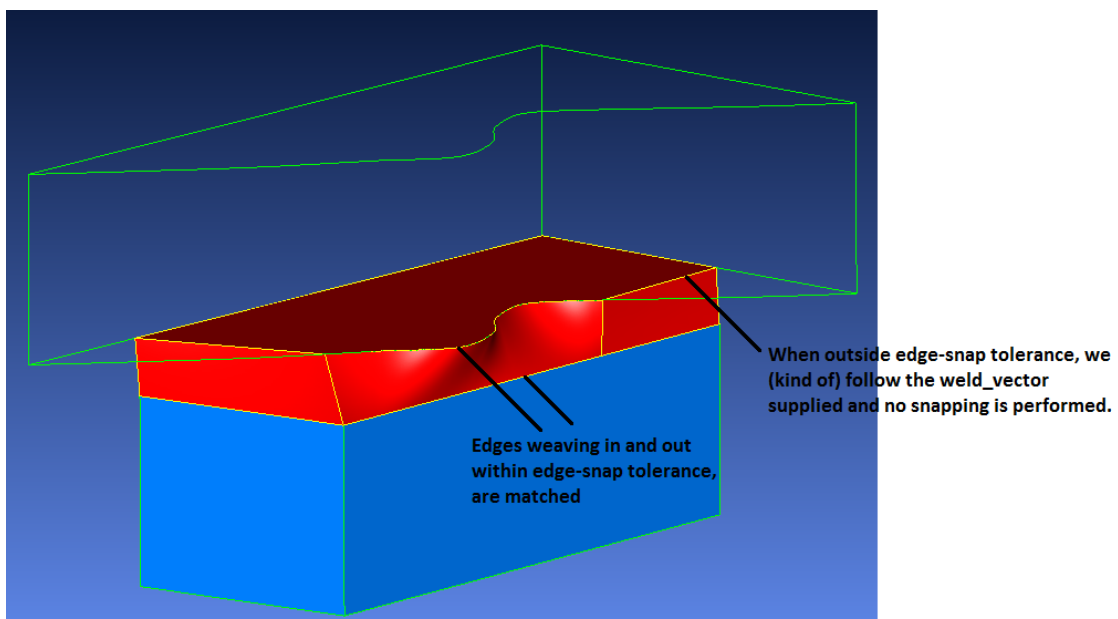


Figure 5. Welding construction when the bodies only partially overlap.

## Automatic fillet removal

Fillets are a common issue during mesh generation. They are often comprised of small faces, forcing additional elements into the mesh, and have high curvature, requiring a curvature-sensitive mesh generator to use many small elements to represent them correctly. Depending on the component

and analysis being performed, they can also be superfluous; the de-filleted model is a good enough approximation.

Fillet removal is therefore a common step to simplify CAD models prior to analysis. CADfix already contains fillet detection code, and tools for fillet removal, but the removal tools are based on the general feature removal tool. As such, the removal process can be slow, and can perform poorly on large, complex, fillet networks.

A new fillet-specific removal tool has been developed to address these deficiencies. It analyses the topology of a fillet network – how the CAD faces are connected to each other – to establish how to remove the fillet, before constructing the intersection lines, and extending the geometry, to remove the fillet.

A key improvement in this fillet removal tool is its ability to recover from failure. When a fillet face is identified which cannot be removed – either the geometry is too complex, or the removal process would damage the model – the fillet removal tool can wall this face off, and still remove the connected fillet faces (figure 6).

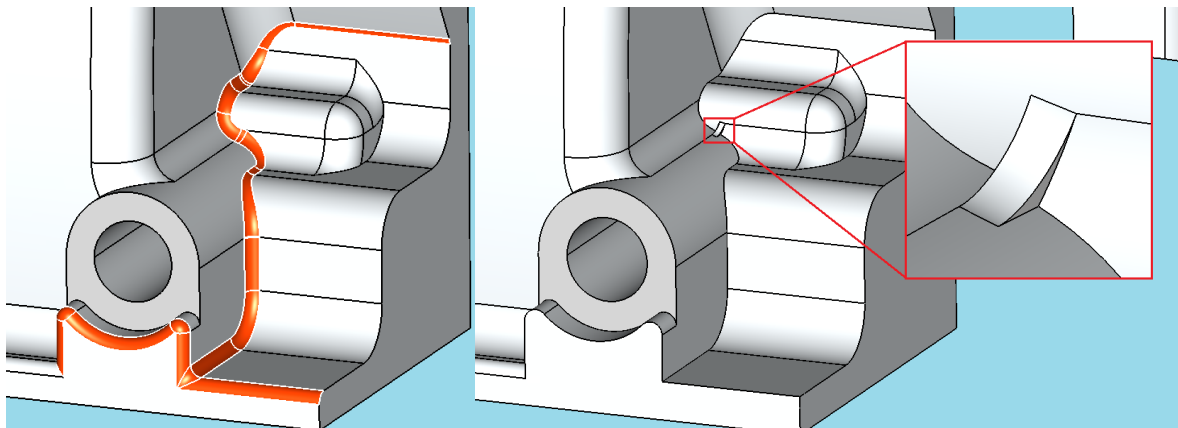


Figure 6. Example removal of a complex fillet. The complete chain cannot be removed, but the tool still removes as much as possible.

## Feature suppression

Different areas of the CAD model require different levels of simplification for CEM. Whilst key areas may need careful treatment to preserve the accuracy of the simulation, other areas simply need a gross representation of an electrical conductor, and can be aggressively simplified.

One such method of aggressive simplification, researched within ICE NITe, is replacement of a collection of solids with the union of their minimum volume bounding volumes. For example, figure 7 shows a scaffolding structure, used to support the ICE NITe test object. The precise shape of this object is not believed to affect the simulation – just a rough approximation is needed – and so figure 7 shows the structure converted into a single, connected, solid by computing a bounding volume for each body, and calculating the union of all bounding volumes. Whilst the original assembly held 100 bodies, with 2689 faces, the defeatured version contains a single connected body, with only 783 faces. Only two user operations were required to achieve this result: one operation to replace the bodies by their bounding volume, and a second to compute the union.

The minimum volume bounding boxes are calculated using a published method (An Exact Algorithm for Finding Minimum Oriented Bounding Boxes, 2015), starting from a faceted convex hull of the object, which in turn is calculated using a Delaunay tetrahedral mesh. The Boolean union functionality was added to CADfix by integrating it from SMLib (Solid Modeling Solutions, 2016), a geometry toolkit which ITI licenses.

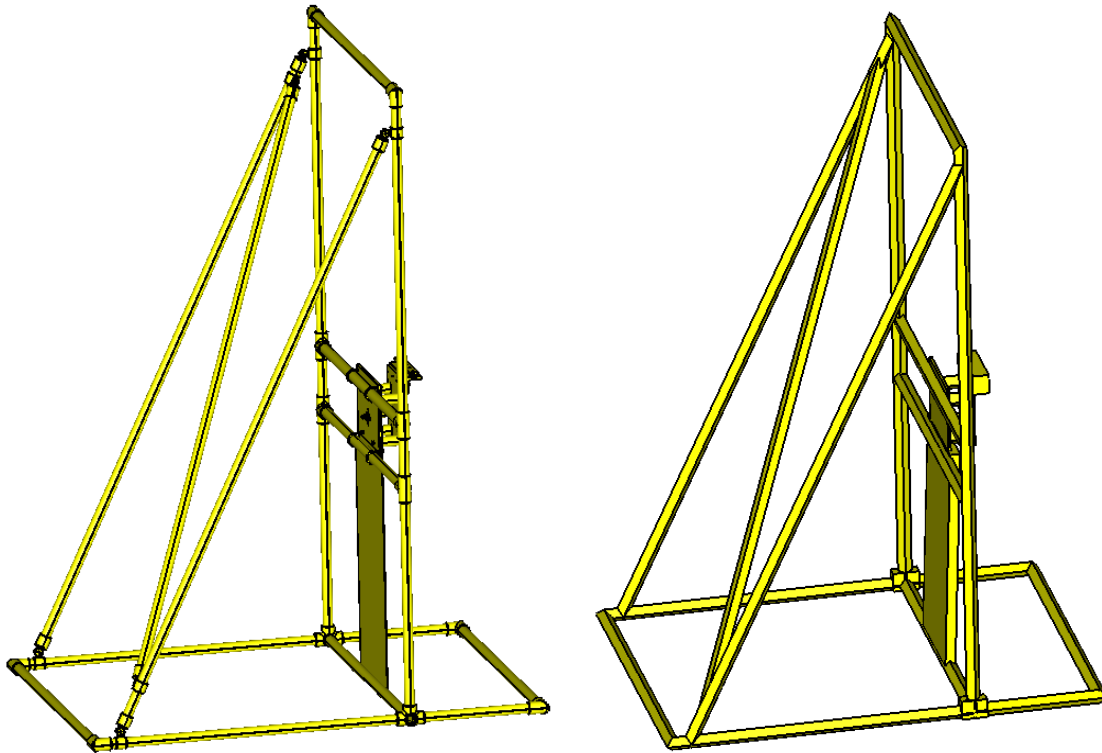


Figure 7. Replacement of bodies by their minimum volume bounding boxes, and Boolean union. The original model is on the left, the defeatured one on the right.

## Advanced face joining

CAD systems represent bodies as collection of connected faces, forming one or more closed shells. The underlying CAD engine has complete freedom in constructing these faces, so long as they represent the chosen geometry, and the face structure is not shown to the CAD operator. However, when constructing a mesh from a CAD model, the boundaries of these faces become constraints on the mesh. Excessive fragmentation of bodies into faces can then cause significant issues when meshing for CEM simulation.

The solution to this problem is to join faces into larger groups (figure 8). This can either be done virtually – constructing a new virtual face (zone), which is defined as a collection of original faces, or done geometrically – constructing a new face definition, complete with a new underlying NURBS geometry. In both cases, the challenges are:

- To construct a parameterisation for the collection of original faces: a mapping from each point on the original faces to a 2D plane.
- To construct a geometric definition for the interior of the new face.



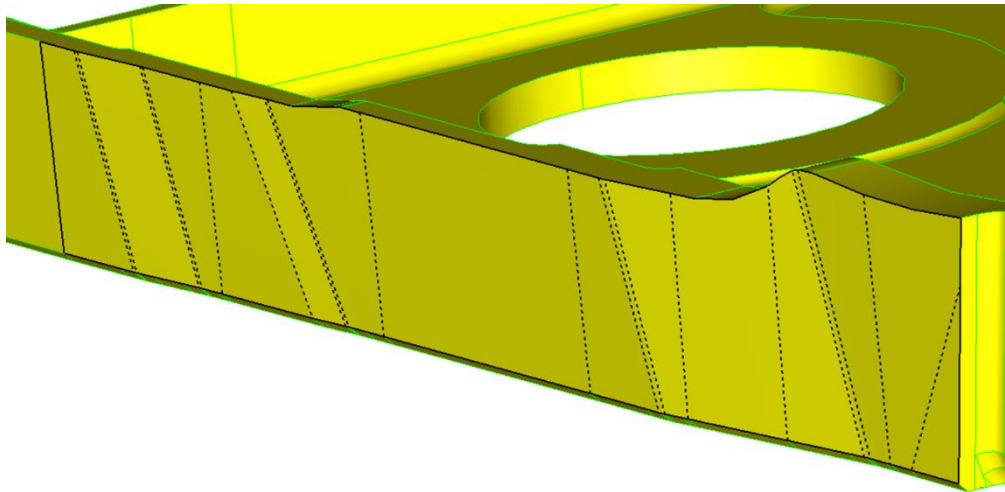


Figure 8. An example joined face on a BAE Systems geometry. The dotted lines indicate the original faces.

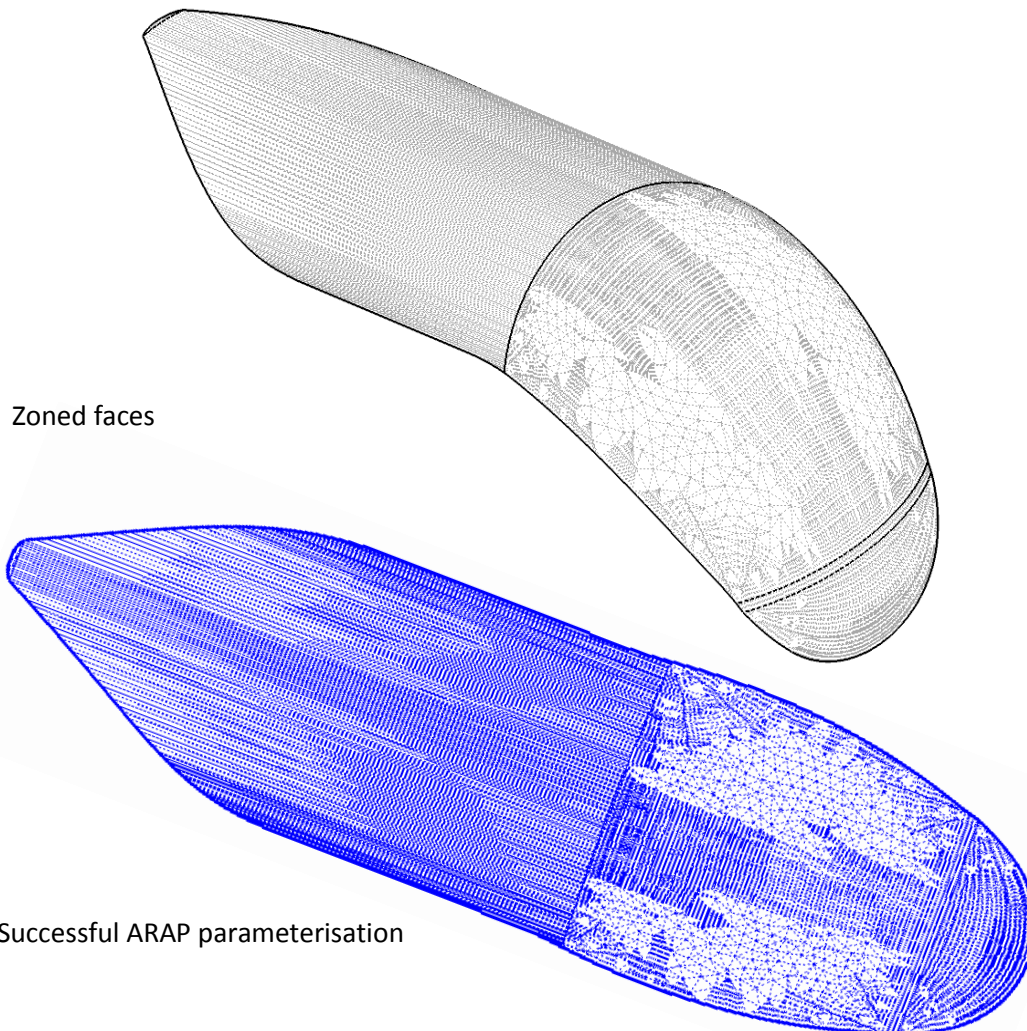


Figure 9. An example ARAP parameterisation

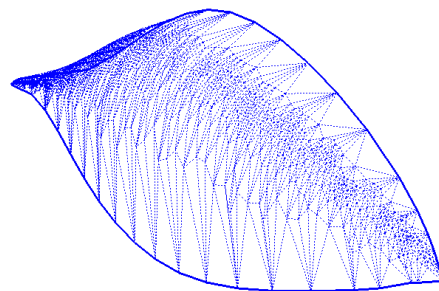
The As Rigid As Possible (ARAP) (A local/global approach to mesh parameterization, 2008) technique has been implemented to provide reliable parameterisations over general collections of faces. This works by computing a compatible triangulation over each of the original faces, joining these



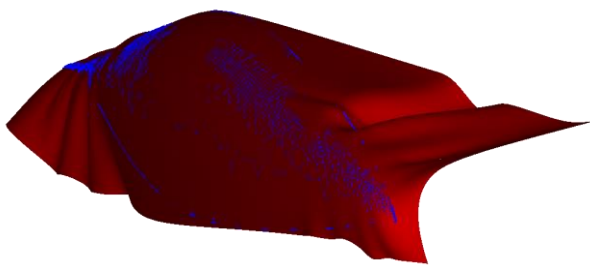
together to create a consistent triangulation over the joined face, and then assigning parameter space coordinates to each triangle vertex by keeping the 2D triangle shapes as similar to the 3D ones as possible (figure 9).

Once the collection of faces has been parameterised, we need to generate geometry to represent the interior of the joined face. This new geometry must be within some tolerance of the original faces, for the joined face to be acceptable. We find such a geometry by first computing compatible Bezier cubic triangulations to represent each of the original faces. These triangulations are then joined to produce a cubic triangulation covering the joined face. Special care must be taken to translate between the individual parameter spaces of the original faces, and the global parameter space of the joined face, whilst preserving the shape of the triangulation. This method gives good control over the accuracy of the joined face, and the use of curved (rather than linear) triangles means that far fewer need to be used, resulting in reduced memory footprint and faster evaluation.

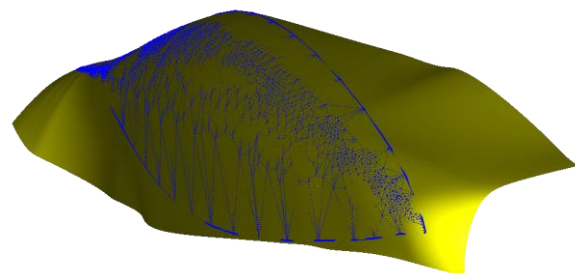
If a joined face needs to be exported, its geometry must be converted to a NURBS surface. This is done by a fitting procedure, which first determines locations for the four corners of the NURBS, and then inserts control points to fit the NURBS to the cubic Bezier triangulation. As the triangulation can be highly non-uniform, with large changes in triangle size and shape, it's critical to take account of these changes in sampling density when performing the fit, and the improved fitting algorithm researched in ICE NITe does this (figure 11).



Example mesh on highly curved wing tip faces



Poorly fitted surface caused by non-uniform mesh density



Improved fit by taking sampling density into account

Figure 11. Improved NURBS fitting by accounting for sampling density.

## Conclusion

Before CAD data can be meshed for CEM simulation, it must be well connected, with an appropriate level of detail. Converting models into this form can be difficult, requiring extensive manual

intervention. In the ICE NITe project, ITI have researched new geometric algorithms, and developed them into new tools, to improve the automation of the CAD to CEM process. These imprinting, welding, fillet removal, feature suppression, and face joining tools have been integrated into the ITI CADfix product, made available to the ICE NITe partners, and will form the basis for enhancements and additional modules in future versions of CADfix.

## References

*A local/global approach to mesh parameterization.* **Liu, Ligang, et al. 2008.** 2008. Proceedings of the Symposium on Geometry Processing. pp. 1495-1504.

*An Exact Algorithm for Finding Minimum Oriented Bounding Boxes.* **Jylanki, Jukka. 2015.** 2015.

**Solid Modeling Solutions. 2016.** Solid Modeling Solutions - NURBS-Based Geometry Kernel Libraries. [Online] 2016. <http://www.smlib.com/>.