Local remeshing of CAD models for finite element analysis: a feature-based approach

M. Sypkens Smit and W. F. Bronsvoort

Delft University of Technology, Mekelweg 4, 2628 CD Delft, The Netherlands
m.sypkensmit@tudelft.nl, w.f.bronsvoort@tudelft.nl

Summary. An original approach to locally remeshing CAD models for finite elements analysis is proposed. The method analyses the geometry of two feature model variants by individually comparing features. The ideas are validated by experimental results.

1 Introduction

Finite element analysis is nowadays an integral part of the product development cycle. It stands apart from the principal design step, as the two fields evolved largely separately, each bearing their particular history of developed solutions and associated practices.

This difference is reflected in practice by design and analysis each having their distinct geometries. One of the areas addressed by research has been the link between the design and the analysis model. The former is nowadays commonly a feature model and the latter a mesh. There does not really exist a link between the two yet, as the mesh is often completely regenerated at each step of the design cycle. With models getting more complex, the approach of reusing meshes by locally adapting them, instead of completely reconstructing them at every step, is getting more attractive.

2 Basic idea of remeshing

The concept of remeshing for finite element analysis is that, instead of completely regenerating the mesh for each analysis, the mesh is adapted to reflect the modifications in the geometry it is linked to. The concept is not new, but its broader application to the development cycle has yet to catch on. A likely reason is that the effort is perceived to be too large for the gain: meshes can be generated fairly quickly and dealing with modifications between geometries is far from easy. Quality meshes, however, are often more costly to generate.

What constitutes 'a quality mesh' depends on the particular problem context, but obtaining it usually involves some kind of optimisation procedure or comprehensive exploration of a set of possible solutions, e.g. [1] and [2]. Especially in the case of small modifications to large models that already have a quality mesh, remeshing is likely to be beneficial.

Many research efforts have been, and still are, directed at the mesh deformation / morphing approach. Here the existing mesh is reshaped, as if it were rubber, by stretching, compressing and bending. See [3] for a recent example. The quality of the mesh elements can be severely reduced by such mesh deformation and changes in topology cannot be handled generically. Amongst the few contributions that did consider modifications to the topology in the context of remeshing are [4] and [5]. These approaches are lacking, however, on genericness, flexibility, or implementation. Efficiently handling generic topological changes is in particular relevant when looking at the remeshing problem in the context of the product development cycle, where changes in model topology are common.

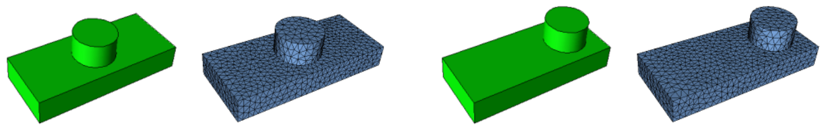
In our research, we approach the remeshing problem from the angle of feature modelling and the desire to maintain the analysis model as an integral part of the design model. To this end, we have come up with a new way to associate geometry between two different feature models.

3 Remeshing feature models

Feature modelling today is the standard in CAD. There is no uniform definition of what feature modelling is. Generally it enables a designer to construct a model by combing higher level objects (the features), instead of manipulating lower level geometry. Examples of common features are holes, ribs, bumps and fillets. Features are manipulated through their parameters, which affect their shape and position in the model. In advanced modelling systems, they also carry semantics that actively support the designer in his decisions and can warn him of unintended consequences.

The design space is thus essentially parameterised by means of the features. This makes it a natural choice to reason by means of the features when looking at the difference between two models, such as needed when remeshing. The idea is that each additive feature corresponds to some part(s) of the model's mesh. We could say that these features 'carry' the mesh. After a modification, we want to create the new mesh by copying, as much as possible, the mesh that the features carried in the previous mesh. Figure 1 shows two variants of a simple model and the corresponding meshes. It is easy to imagine that the mesh of Figure 1(b) could *almost* be acquired by just recombining the mesh covering the base block and the mesh covering the cylinder.

In reality, however, we cannot recombine the meshes corresponding to features without some extra work. The principal reason is that features interact, i.e. they overlap at faces or even with their volumes. Volumetric meshes do



(a) Original model: base block with a cylindrical feature on top, and its mesh

(b) Modified model: base block with cylindrical feature translated, and its mesh

Fig. 1. Original and modified model and their meshes

not, and should not, respect the internal feature boundaries, and thus mesh elements will not align on the whole boundary of features. When the relative positioning between features changes, such as in Figure 1, then the mesh needs to be adapted in the areas where the interaction changes.

For our feature-based remeshing approach, we thus need to find all areas where feature interactions change. Generally, feature modelling systems are history-based, i.e. they do not explicitly store the interactions of all features, but compute a new boundary representation (BRep), and continue the rest of the operations with the new geometry. This approach is thus not suitable for finding and describing the changes in interaction.

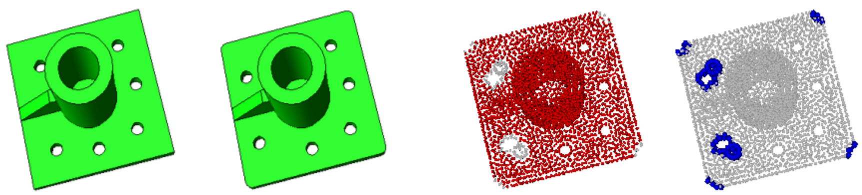
A data structure that does store the complete geometry of all the features is the *cellular model* [6]. It does this by keeping the complete geometry of all features, and where features intersect they are divided into cells. A feature that covers a cell, is said to *own* that cell. Each cell is owned by at least one feature; where features overlap, the cell has multiple owners.

In the cellular model, we have a complete geometric description of each feature, which includes those parts of its boundary that are not part of the BRep. It also contains an explicit partition of the volume of the model. After a model has been modified, we can thus analyse for each feature how its geometric interaction with the other features has changed. If nothing has changed, then all mesh elements covering the feature can be copied to the mesh of the new model.

For the case that something has changed, we need a consistent classification of what has changed and in what respect. For example, feature geometry that previously was internal to the model, can have become part of the BRep; cells that used to be empty, due to overlapping subtractive features, can now contain material. For this classification, we use the *feature difference* [7]. The feature difference compares the geometry of a feature, as it was in a previous variant of the model, with the geometry of the feature in a new variant of the model. This geometry includes all cells, faces, edges and vertices that emerged through interaction with other features in the model.

Based on the feature difference, we can thus decide for each feature which parts of it could contribute to the new mesh model and where/how the mesh needs to be adapted. The final challenge is to combine this information, on each individual feature, for the construction of a single mesh for the complete model. Since features intersect, we must be careful not to copy points multiple times to the same region. Also, we need to identify the regions for which no mesh can be copied, which can be due to new features or the relocation of subtractive features. Lastly, we must pinpoint the regions where meshes need some further work, e.g. because mesh with different origins ended up copied next to each other, or because of interaction changes (such as in Figure 1.

The outline of the algorithm is: 1) analyse the difference between the two feature models, 2) identify the regions in the new model for which mesh can be copied and actually copy it, 3) create new mesh in the remaining areas of the new model, 4) connect and optimise the mesh in all regions where a (good) connection is lacking. Figure 2 illustrates the regions of mesh that are fixed (red) or should be optimised (blue) at the start of step 4 of the algorithm.



(a) Original and modified model

(b) Copied nodes (red) and new/active nodes (blue)

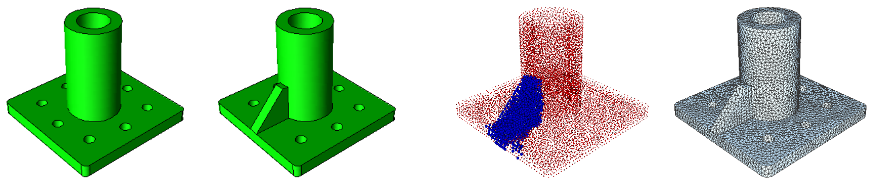
Fig. 2. Example of model modification with corresponding copied nodes (red) and new/active nodes (blue)

4 Experimental results

To test our feature-based approach to remeshing, we have implemented the variational tetrahedral meshing algorithm [8], with some minor modifications that improve its applicability to mechanical models. This algorithm results in high quality mesh elements, by alternating optimisation steps of the boundary and interior mesh. This algorithm produces a Delaunay mesh, which is a major advantage here as it avoids the need to explicitly copy the connectivity of the elements. Instead, we can just copy the nodes.

Figure 3 illustrates a result of this approach. We compared the result with that of the full meshing procedure. The remeshing took *30*% of the normal meshing time. The element quality distrubition is nearly identical.

In general, our experiments indicate a efficiency gain between 50% and 80%. The gain depends on the number of mesh nodes, the amount of surface area relative to the volume, and the extent of change in feature intersections.



(a) Original and modified model **(b)** Region of optimisation and result of remeshing

Fig. 3. The result of remeshing a tool with a new stiffener feature

5 Conclusions and looking ahead

We have demonstrated that our approach to remeshing, based on the feature difference, is feasible and that it can improve efficiency. The gains depend on the number of mesh elements, the extent of the model modifications, and the complexity of the meshing algorithm.

Our current approach is not universal. It relies on the Delaunay criterion for the connectivity. In order to support other kinds of meshes and meshing algorithms, we need to be able to reliably copy the mesh connectivity between the nodes. The handling of graded meshes is another challenge. Can a sizing field efficiently be maintained under model modification? What are good strategies to maintain the conformity of the mesh to the sizing field when remeshing? What is the impact on efficiency? Opportunities for further research are abundant.

References

1. Lagüe, J.F., Hecht, F.: Optimal mesh for P_1 interpolation in H^1 seminorm. In: Proc. of the 15th International Meshing Roundtable, pp. 259–270. Birmingham, AL (2006)
2. Klingner, B.M., Shewchuk, J.R.: Aggressive tetrahedral mesh improvement. In: Proc. of the 16th International Meshing Roundtable, pp. 3–23. Seattle, WA (2007)
3. Date, H., Onosato, M.: Triangular mesh deformation based on dimensions. Computer-Aided Design and Applications **5**(1–4), 287–295 (2008)
4. François, V., Cuillière, J.C.: 3D automatic remeshing applied to model modification. Computer-Aided Design **32**(7), 433–444 (2000)
5. Sheffer, A., Ungor, A.: Efficient adaptive meshing of parametric models. Journal of Computing and Information Science in Engineering **1**(4), 366–375 (2001)
6. Bidarra, R., de Kraker, K.J., Bronsvoort, W.F.: Representation and management of feature information in a cellular model. Computer-Aided Design **30**(4), 301–313 (1998)
7. Sypkens Smit, M., Bronsvoort, W.: The difference between two feature models. Computer-Aided Design and Applications **4**(6), 843–851 (2007)
8. Alliez, P., Cohen-Steiner, D., Yvinec, M., Desbrun, M.: Variational tetrahedral meshing. ACM Transactions on Graphics **24**(3), 617–625 (2005)