Efficient Adaptive Meshing of Parametric Models

Alla Sheffer

Computer Science Department
Technion
Haifa, Israel, 32000
e-mail: sheffa@cs.technion.ac.il

Alper Üngör

Department of Computer Science
University of Illinois at
Urbana-Champaign
Urbana, IL, 61801
e-mail: ungor@cs.uiuc.edu

October 18, 2001

Abstract

Parametric modeling is becoming the representation of choice for most modern solid modelers. However, when generating the finite-element mesh of the model for simulation and analysis, most meshing tools ignore the parametric information and use only the boundary representation of the model for meshing. This results in re-meshing the model basically from scratch each time a parametric change is instantiated, which happens numerous times throughout the design process.

In this paper we look at ways to use the parametric information during the meshing procedure to prevent unnecessary re-meshing. The paper examines existing meshing techniques developed for other purposes, which can be applied to this problem. It also suggests several new mesh modification techniques specifically designed for efficient mesh adjustment after parametric model changes.

Keywords: Mesh generation, Parametric models, Adaptivity

1 Introduction

Engineering design is an iterative process encompassing requirement definition, concept design, design validation and analysis. Parametric and variational models [1, 2] provide the tools for easy modification of the model parameters, based on analysis results or design decisions. Recent extensive research makes those models significantly more robust and reliable [3, 4]. A testament to the popularity of this representation is the growing number of commercial solid modeling and CAD systems using this representation. These include Pro-E [5], IDEAS [6], UG/Solid-Modeling [7], to name just a few.

With further integration of the product data management applications, solid models generated by the design are used directly in the finite element analysis (FEA) and simulation applications. The use of CAD geometry for FEA had increased from 11% in 1991 to nearly 80% in 1996 [8]. This encourages parametric studies, which address the impact of design variables on product behavior. The integration of CAD and FEA depends on the ability to efficiently generate high quality finite-element meshes of the CAD models. There have been significant advances in the development of mesh generation algorithms [9], however, the process is still far from full automation [8, 10, 11]. The two main areas requiring manual intervention are geometry adjustment prior to meshing and hexahedral mesh generation [12]. Adjustment of model geometry prior to meshing is often required to improve the resulting mesh quality. Additionally, geometry adjustment is a prerequisite for many of the available meshing algorithms. This makes the mesh generation a major bottleneck in the engineering design process.

When generating the finite-element mesh of the model for analysis and simulation purposes, most mesh generation algorithms use only the boundary representation of the model. Hence, parametric changes are not directly considered. As a result, most parameter changes requires re-meshing the model practically from scratch. This happens even if most of the boundary model remains the same, since changing a boundary entity causes most systems to remove the mesh on that entity and all higher dimensional entities that rely on it. This leads to a significant slow down in the design process as multiple repetitive parametric changes are often performed until the desired model is generated.

In addition to efficiency concerns, preserving the links between original and new meshes when parametric changes occur is required when the changes are part of an automatic shape optimization procedure. Shape optimization applications repeatedly modify model parameters based on simulation results. Since the simulation results are often affected by mesh-based noise, minimizing the changes of the mesh in between the optimization iterations improves the optimizer's estimates.

In this paper we examine the tools that can be used to avoid full re-meshing due to parameter changes. The suggested tools use the information provided by the parametric representation both when generating the model mesh and when adjusting it following the change in parameters. Both hexahedral and tetrahedral finite-element meshes are considered.

This work examines the possibility of using existing meshing tools developed for other purposes to be used to facilitate efficient mesh adjustment. The research introduces several new tools for mesh adjustment, which are particularly well suited for efficient mesh modification in the context of parametric model changes.

The approach presented here is based on using a dual representation of the model, containing both a boundary representation and a parametric one, such as the one suggested by Raghothama and Shapiro [3]. This way the mesh generation and modification algorithms can continue to use the boundary representation, while the parametric model will provide the information required during the mesh adjustment procedure. The parametric representation will provide the mapping between the original and new model entities as well as the size/shape changes for each of the entities. Note that in the context of this paper parametric changes are viewed as changes which do not affect the model connectivity. Changes in connectivity tend to require major changes in the model mesh for the analysis purposes, and often require human interaction to decide on those changes.

The rest of the paper is organized as follows. Section 2 describes the various issues which need to be addressed when adjusting a model mesh after parameter changes. Section 3 addresses model geometry adjustment for meshing. In Section 4, the mesh copying from the original to the modified model is described. In Section 5, we describe tools for mesh improvement for both tetrahedral and hexahedral meshes. Section 6 provides two examples of generating a mesh for a complex model after a parameter change. Finally, Section 7 summarizes the results and discusses the effectiveness of using the parametric information.

2 Parameter Modification - Meshing Issues

In this section we consider the issues involved in generating a mesh for a new model, based on an existing mesh of an original model, where the new model was created by changing some of the parameter sizes in the original one. The problems and the solutions are addressed in details in the following sections. The process of generating a mesh for a new model after parametric changes to the original one, involves the following steps.

The first step is adjusting the new CAD model for meshing. As mentioned above to make a model suitable for meshing it often has to be modified. To distinguish between the model generated by the CAD system and the modified model used for meshing we refer to the former as the *design model* and to the later as the *mesh model*. The mesh model may differ significantly from the original design one. The required modifications include suppression of minor details, redefinition of ill conditioned CAD geometry, such as surface slivers, and geometry decomposition. If the modifications are not preserved when the model parameters change, the editing process will need to be repeated, requiring a lengthy and only partly automated procedure. The preservation of the mesh model is addressed in Section 3.

After a mesh model is defined the next step prior to meshing it is the assignment of meshing parameters. Most meshing softwares require setting of such parameters as mesh sizing and type of mesh generation algorithm to be used for each model boundary entity, including the model as a whole. Setting those parameters is necessary to generate the mesh of the new model. If the new mesh model is identical to the original one, those parameters can be copied from each original model entity to its counterpart in the new model.

When the mesh models' connectivity is identical and a one-to-one mapping of the model entities exists, the new mesh can be generated, by finding the locations for the new nodes and copying the original connectivity. This step in the generation of a mesh for the new model is explained in Section 4.

The final step in the meshing process is mesh connectivity adjustment. After the mesh is copied to the new model it might no longer have an acceptable quality or observe the desired mesh sizing. In this case the mesh connectivity needs to be modified. The required amount of work depends on the type and size of the parameter changes. The process is described in Section 5. This step might not be necessary if the model changes are minor and the mesh quality if sufficiently high after the mesh reassignment step. On the other side, if the parameter changes are too big, a complete re-meshing might be necessary.

Figure 2 shows the steps for generating the mesh given a new model, using the mesh of the original model (Figure 1). The tools, which can be used in each step, are discussed below.

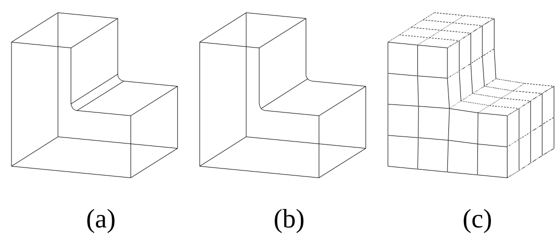


Figure 1: Meshing an "L" shaped model. (a) Original design model. (b) Mesh model (after simplification). (c) Hexahedral mesh.

3 Preserving the Mesh Model

As explained above, when preparing CAD models for meshing, the models often need to be modified to suit the meshing needs. Such an adjustment usually includes [8, 13]:

- Suppression of design details that are too small to affect the analysis results but that impose severe constraints on the meshing procedure. Since mesh generation algorithms are guided by the B-rep, the mesh connectivity has to follow the face and edge boundaries in the model. Hence if the model local feature size is small compared to the desired mesh element size, it adversely effects the mesh, resulting in reduced mesh quality and larger number of mesh elements than necessary. An example of detail suppression is shown in Figure 1, where a fillet narrower than the desired mesh size is removed.

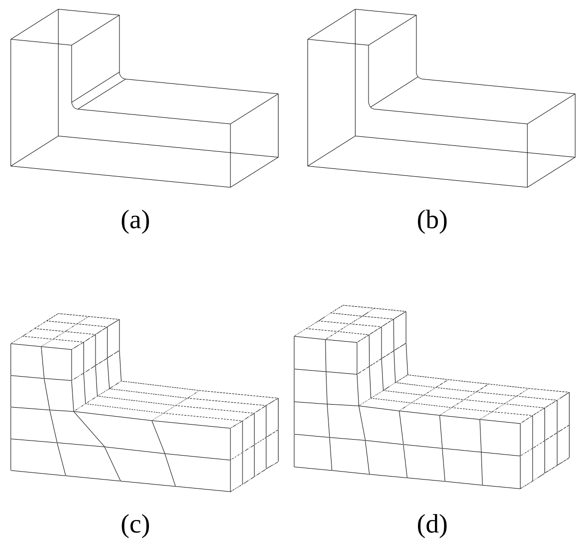


Figure 2: Adjusting the mesh of “L” shape in Figure 1 to different parameter sizes. (a) Design model. (b) Mesh Model (same topology as original mesh model). (c) Reassigned Mesh. (d) Mesh after connectivity adjustment (whisker-sheet splitting, Section 5.2.2).

- Redefinition of ill conditioned CAD geometry like sliver faces, narrow corners and minute edges. Such geometries have and adverse effect on meshes similar to the minor details above, and have no design or analysis significance.
- Decomposition of complex models into meshable parts. Decomposition is often required to enable the use of available meshing algorithms and to improve the mesh quality. It is especially common when hexahedral meshes are generated, as hexahedral meshing algorithms are less generic.

Numerous tools [14, 13, 15] were developed to perform those adjustments. The tools offer partial automation of the model editing process but some user interaction is still required. Examples of typical necessary editing are shown in Figure 3.

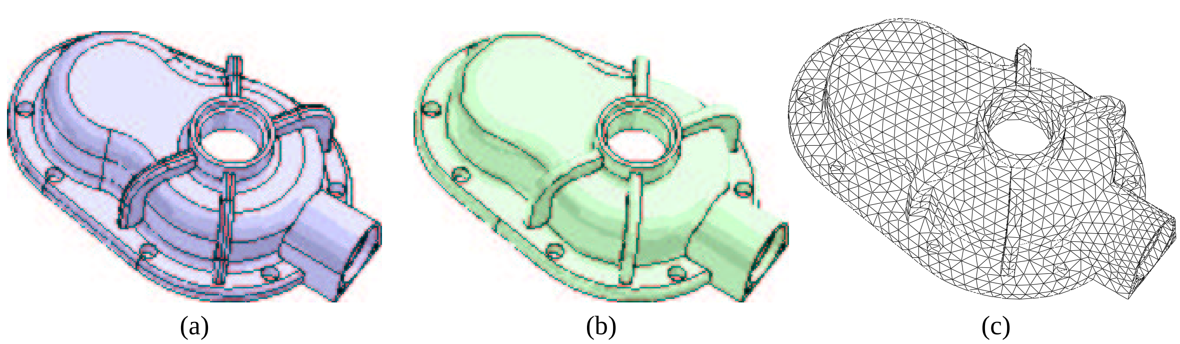


Figure 3: A pump housing model. (a) Design model. (b) Mesh model (after simplification). (c) A tetrahedral mesh of the model.

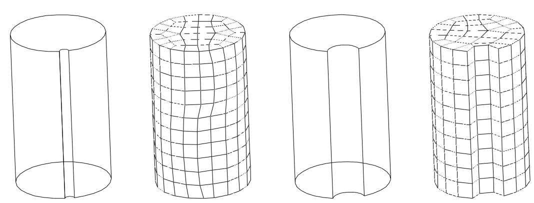
Since those modifications are performed for meshing purposes only, the design model is unchanged. In the current solid-modeler systems the mesh and design models are not connected and the parametric changes are applied only to the design model. Hence, the mesh model needs to be explicitly rebuilt after the parameters change. There are tree alternatives to avoid editing yet again the design model to fit the meshing needs.

The first is to maintain the connection between the mesh and design models within the system, in order for parametric changes to be applied to both. This is probably the optimal solution, but to the authors knowledge, no existing commercial software embraces it.

Another approach is to maintain a history log of the model editing process, and when the design model is modified, to rerun the sequence of editing commands based on the log. This approach can work for many models, but it is likely to encounter entity naming problems [16], since a log re-run relies on the entity names to perform the desired operation sequence. Names might often change, since parameter changes can cause different ordering of operations, especially for size based procedures, like detail suppression.

An alternative approach we suggest is to perform the editing using a virtual topology model [13]. The virtual topology model representation separates the model geometry and connectivity information and allows topological (connectivity) entities to correspond to a part or a set of geometric entities (e.g. a face can correspond to a part of a model surface or to a set of adjacent surfaces). The virtual topology editing operators construct new topological entities on top of the existing ones, thus keeping the original model underneath intact. The virtual topology operators were used for editing the models in all the examples shown in this paper. Two examples of editing complex models are shown in Figure 3 and Figure 17. Note that in all cases, the model geometry is unchanged and only the connectivity is modified.

Using the virtual topology editing tools for generating a mesh model from a design model [10, 15], constructs the mesh model on-top of the geometric, design model without changing it. Hence, given a new design model, differing from the original only in parameter sizes, the virtual topology model can be simply copied on top of the new design model. This would only require copying the topological entity hierarchy from the original model to the new. The copy operation together with the virtual topology model representation and other virtual topology operators were implemented as part of the Gambit analysis pre-processor [17]. An example of topology copy is shown in Figure 2(b), where the topological hierarchy as built for the original design model (Figure 1(b)) was copied onto the new model (Figure 2(a)).



(a) (b) (c) (d)

Figure 4: Feature size impact on mesh generation. A notch narrower than prescribed element size (a), has little impact on analysis, and hence can be suppressed in the mesh model (b). When the notch width increases above the mesh size (c), it starts to have an impact on analysis results. Hence, the model needs a mesh reflecting it (d).

Prior to generating the mesh model for the new design model, a parameter check is required to see if the criteria used for generating the mesh model still apply. For example a minor feature suppressed in the original model, might in the new model grow to be of impact for the analysis (Figure 4). This check can be done using the model parameter sizes and the mesh sizing. If some of the criteria no longer apply, mesh adjustment, as proposed here,

will not generate a mesh suitable for analysis, as it will not follow important model features. In such a case to generate a good quality mesh, the model needs to be re-edited and re-meshed.
Once the new mesh model is constructed, since the original and new mesh models have identical connectivity, the mesh can be copied to the new model.

4 Mesh Reassignment and Smoothing

4.1 Mesh Reassignment

Given two models with the same connectivity, the next step is to assign a copy of the original mesh to the new model. This requires to position the new nodes and then generate a mesh connectivity between them identical to the original one. To position the nodes the model boundary representation is used. Each node location is generated based on the model entity the original node was on. The positioning process is done based on the entity hierarchy, first vertex nodes are placed, than edge ones and so on.

- The new nodes corresponding to original mesh nodes located at the model vertices, are placed at the corresponding new model vertices.
- New edge nodes are placed on the corresponding new edge, at the same lengthwise parameter as were the original nodes. (Lengthwise parameterization is independent of the actual geometric curve definition.)
- Face nodes are placed on the appropriate new face. When a surface parameterization for the face exists, the u, v coordinates of the original node can be used to place the node on the new face. However, in some cases such parameterization is not available, or is highly non-uniform. In this case, the placement strategy in Section 4.1.1 can be used.
- Nodes on the interior of the solid are placed last. Since there is rarely a volume parameterization (u, v, w) present, the placement strategy described below has to be used.

4.1.1 Placing interior mesh nodes for faces and solids

Below, we describe two approaches for placing interior nodes on face or inside a solid, based on the boundary nodes. The first approach is to use a background mesh [18] (Figure 5). First, a background mesh of the original boundary is generated, by constructing a Delaunay triangulation of the boundary nodes. Next, a similar background mesh is assigned to the new boundary, by connecting the new boundary nodes using the same connectivity as generated for the original ones. To place an interior node, its containing triangle in the background mesh of the original face is found. Then, the barycentric coordinates of the node in the triangle are used to place it on the corresponding triangle in the new background mesh.

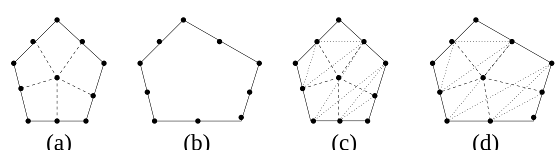


Figure 5: Copy of interior face nodes using a background mesh. (a) Mesh of the original face. (b) New, modified face with copied boundary nodes. (c) Background mesh of (a). (d) Background mesh and copied interior node on (b).

For a solid, the same procedure is used, only a tetrahedral background mesh is generated instead. The generation of the background mesh while relatively expensive can be done once for the original model and then, it can be used repeatedly for different parameter changes. The use of the background mesh, provides a good estimate for the new node locations, but does not guarantee optimal placement. Often, mesh smoothing as described below should be performed to improve the node locations.
The second alternative is to keep the node locations as they were in the original model and let the mesh smoothing procedure, which follows, to find the optimal node locations based on the location of the boundary nodes and the connectivity [19]. This approach avoids the complexity of generating the background mesh, but might require significantly more effort at the smoothing stage. It is likely to be more efficient when only minor parameter changes are done, but will require strong smoothing tools when changes are large.

4.2 Smoothing

The mesh copy does not ensure optimal node locations, with respect to the mesh quality and further processing is often required to improve it. Node locations define the shape of the mesh elements, which in turn determine the mesh quality, i.e. its suitability for analysis. Mesh quality measures evaluate the effect of the shape of the elements in the mesh on the correctness of the analysis results. Various simplicial mesh quality measures, such as minimum angle, maximum angle, aspect ratio, radius-edge ratio are extensively studied in the literature [20, 21]. Also a large number of criteria are suggested for measuring the quality of hexahedral elements [22, 17]. Some recent results by Knupp [23], suggest a systematic approach for measuring quality by use of algebraic mesh quality metrics. The work proposes several metrics like shape, skew, volume, and combinations of those. Usually, to answer the question if the mesh is suitable for analysis several measures have to be checked, as each measure captures some aspects of the element shape, but none captures all. The suitability for analysis depends on both the average (median, etc.) and the worst values of the quality measures. For example, if a single element with a negative or nearly zero Jacobian exists, the entire mesh is invalid. Additionally mesh sizing has to be considered, so that the mesh conforms to a prescribed sizing function, defined by the accuracy required by the analysis.
If after the mesh copy is performed, the mesh quality is not acceptable, one option to improve it is to perform a smoothing procedure. Smoothing is a general term for mesh improvement process in which mesh nodes are relocated but the mesh connectivity is unchanged. There is a large volume of work on different smoothing algorithms, as reviewed in [9]. The methods can be roughly divided into two groups, global and local. Global algorithms are those used to move every node in the mesh [24]. Such methods usually perform relatively simple computations for each node and perform several iterations of those. They are usually relatively cheap to perform but do not handle well extremely distorted meshes and often achieve sub-optimal results. Local methods, like in [22, 25, 23] perform more complex computations per node, including application of optimization techniques. They achieve much better results and can handle very distorted meshes, even those that include elements with negative Jacobian. However they are significantly more expensive and hence, are usually used only in small regions of the mesh, where the quality is low.
When the parameter changes are small (e.g. Figure 6) the mesh copy together with smoothing is usually sufficient to generate a high quality mesh of the new model. However, because smoothing doesn't change the mesh topology, its ability to improve mesh quality is limited (e.g. Figure 2). Moreover, when model parameters change, the element sizing might no longer satisfy the analysis requirements. In such cases a modification of the mesh connectivity is required, as described in the next section.

5 Mesh Connectivity Adjustment

Changing the mesh connectivity is the last option available, when after mesh copy and smoothing the mesh quality remains unacceptable. When changing mesh connectivity there are significant differences between the options

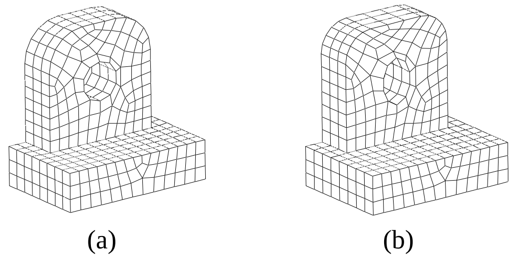


Figure 6: Adjusting the mesh for a simple bracket. The changed parameters include: hole radii, blends radius, and base width. The changes are relatively small, hence mesh copy with smoothing is sufficient to get a good quality mesh. (a) Original model. (b) New model.

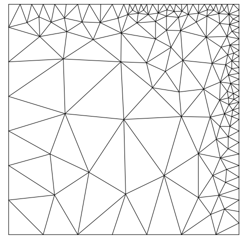
available when dealing with hexahedral or tetrahedral meshes. The possible modification techniques for each mesh type are described below.

5.1 Simplicial Meshes

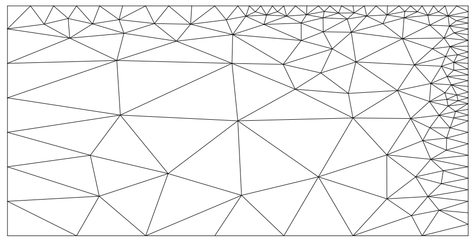
This section surveys existing triangular/tetrahedral meshing methods to provide an efficient solution for adaptive meshing of parametric models. These techniques were introduced to handle mesh adaptation when the mesh sizing function changes, here we apply them to the problem of parametric changes in the model. More specifically we address the issue of handling the copied mesh on the new model, which as result of the parameter changes no longer satisfies the mesh quality requirements or the mesh sizing.
Since most parametric changes apply to only a part of the model, a large part of the model mesh maintains the required sizing and quality. The simplicial adaptive techniques, described below, can be applied solely in the areas where the mesh changed, since all of them have only local effect.

5.1.1 Refinement Techniques

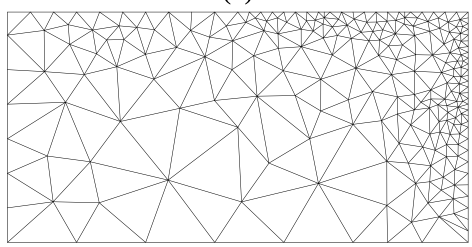
Recent unstructured mesh generation literature witnessed many refinement techniques. Most of them are designed to generate quality guaranteed triangular/tetrahedral meshes. Among them, Ruppert's Delaunay refinement method [26] was the first to provide a quality guarantee for optimal size meshes in two dimensions. This method starts by generating the constrained Delaunay triangulation of the input domain. Then, bad quality triangles are eliminated by inserting their circumcenters while maintaining the Delaunay triangulation. The work proves a $20.7°$ minimum angle guarantee. Shewchuk extended this approach to 3D, but with no guarantee on avoiding slivers [27]. Several new algorithms were recently suggested to remove slivers [28, 29].
The sink insertion method [30] also uses point insertion to improve the triangulation quality, however it behaves selectively in deciding which circumcenters to insert. Alternative to circumcenter insertion, Rivara [31] suggested to split the longest edges to generate quality triangulations.
We suggest using the refinement techniques for adaptive meshing of the parametric models. After the original mesh is copied to the new model, the quality of the new mesh can be unsatisfactory, i.e. if the original mesh is a Delaunay triangulation the resulting mesh is no longer guaranteed to be Delaunay. The mesh refinement methods discussed can be used to improve it. This process avoids complete re-meshing. Hence it tends to require less work when the parameter changes are small. If the parametric scaling of the model parts is more than one, this immediately implies a coarse transformed mesh. Hence, there is no unnecessary refinement in the final mesh. If the scaling is smaller than one however, the final mesh could have a larger than necessary number of elements.



(a)



(b)



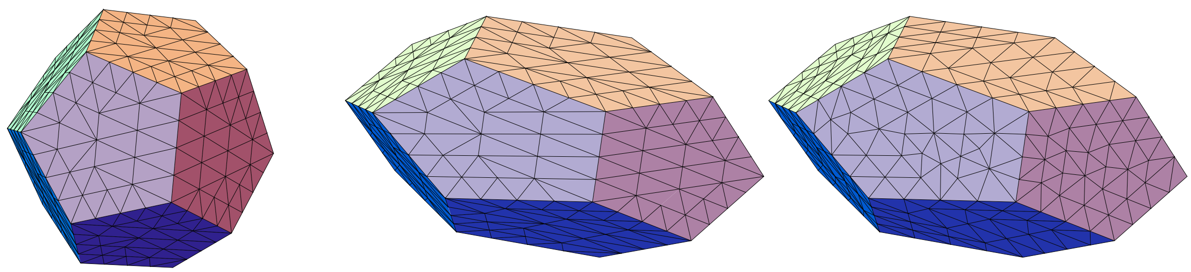
(c)

Figure 7: Adaptive re-meshing of a 2D graded mesh using the refinement technique. (a) Original mesh with 169 nodes. (b) The model is stretched by a factor of 2 along X direction. This creates 54 bad elements in the mesh. (c) Final mesh is a result of 75 circumcenter points insertions.

This issue is discussed in Section 5.1.3 and a solution is suggested. We suggest the following four step algorithm as a formalization of our refinement based adaptive parametric model meshing method.

1. Copy the mesh of the old model to the new model as explained in Section 4.
2. Perform face swaps to improve the quality, e.g. to get a Delaunay triangulation.
3. Recompute the quality of each simplex element
4. While there exists a simplex T that has a bad radius to shortest edge ratio, insert the circumcenter of T and perform face swaps to maintain the optimality.

The face swap operation suggested in the second step of the algorithm is discussed in computational geometry and mesh generation literature [20, 21]. It is used both for computing the Delaunay triangulation and for improving mesh quality. The last two steps of this method suggest the use of a classical Delaunay refinement technique. It could be replaced with other refinement techniques discussed above, e.g. longest edge refinement method. The last two steps could be omitted if the There are several criteria to measure the quality of a simplex, including aspect ratio



(a) (b) (c)

Figure 8: Surface mesh of a dodecahedron. (a) Original mesh with 290 nodes. (b) The model is stretched by a factor of 2 along X direction (for illustration purposes the top view is chosen). The stretching creates 98 bad simplices. (c) Inserting 43 circumcenter points removes all the bad simplices.

(inradius over circumradius), and minimum angle. The above algorithm employs the circumradius over shortest edge and the illustrations in Figure 7 and 8 are generated using this criteria. These three quality criteria are known to be equivalent in 2D. However, circumradius shortest edge criteria does not prevent the existence of slivers in 3D. This requires the post-processing of the mesh to remove slivers using one of the techniques suggested in [28, 29] .

5.1.2 Maintaining a Sphere Packing

Sphere packing became a popular tool in mesh generation research [32, 33, 34, 35]. It enables generating well-spaced points sets which in turn guarantees quality meshes [34]. The sphere packing methods pack the domain with spheres where the size of the sphere centered at a point x is related to the desired mesh size at x. The desired mesh size is determined as a combination of the geometric properties of the domain and the numerical properties of the simulation. Geometric component of the spacing is usually represented by the *local feature size* function introduced by Ruppert [26]. The *local feature size* at a point x, $lfs(x)$ is the radius of the smallest disk centered at x that intersects two non-incident vertices or segments of the domain. The Lipschitz property of this function, which implies a slow change in the function, lies behind the technical proofs of Ruppert's quality guaranteed triangulation results [26]. Later, Li *et al.* [33] defined an analogous function which also has Lipschitz property to denote the numerical spacing:

$$nsf(x) = min\{nsf_e(x), min_y\{nsf_e(y) + ||x - y||\}\} \tag{1}$$

where $nsf_e()$ is the numerical spacing driven by the error analysis. Global spacing function can be defined simply the minimum of these two spacing functions.

$$gns(x) = min\{lfs(x), nsf_(x)\} \tag{2}$$

This section reviews the sphere packing based quality guaranteed mesh generation methods and describes their use for adaptive meshing of parametric models.
Suppose $gns()$ is the desired size function of a well-shaped mesh for a domain Ω. Let $B(x, r)$ denote the sphere centered at point x with radius r. Let β be a positive real constant. A set S of spheres, $B(p, gns(p)/2)$, $p \in \Omega$ is a β-packing [33] if

- No two spheres in S overlap; and

- There are no big gaps, i.e. $\forall q \in \Omega$, $\exists$ a sphere in S that overlaps with $B(q, \beta gns(q)/2)$.

The structure theorem of Miller *et al.* [34] states an equivalence relationship between β-sphere packing and well-shaped (bounded aspect ratio) meshes. Based on this result several algorithms are suggested to construct well-shaped meshes in 2D and bounded circumradius shortest edge ratio meshes in 3D [32, 33, 34]. The four step algorithm we suggest below is similar to simultaneous refinement and coarsening work in [32].

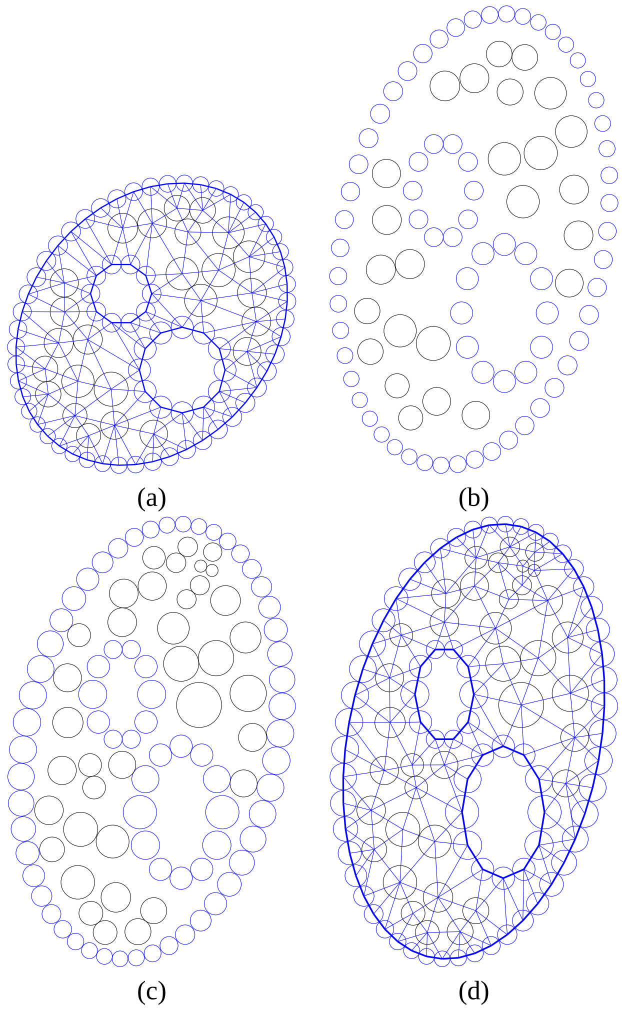


Figure 9: Maintaining a sphere packing (a) Original mesh generated by a sphere packing. (b) Packing is no longer good as big gaps in the domain are introduced. (c) Packing is modified to be a good packing. (d) Delaunay Triangulation of the points in the packing gives a good quality mesh.

1. Copy the mesh of the original model to the new model as explained in Section 4.
2. Recompute the $gns()$ function to reflect the new geometry and the numerical spacing needs.

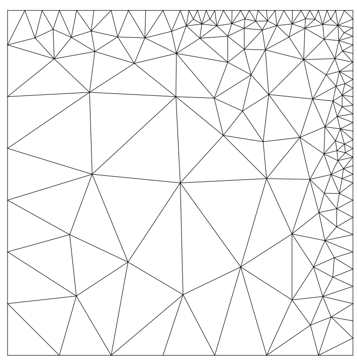
3. Modify the packing to become a good packing. This can be achieved in various ways and discussed below in detail.

4. Construct the Delaunay triangulation of the centers of the new packing spheres.

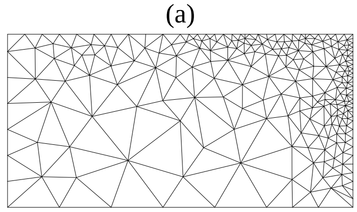
There are various ways to maintaining a good packing. In [32], the oversampling idea is shown to generate a good packing for adaptive meshes. First, the method inserts new points (and their corresponding spheres) to the domain to ensure there are no big gaps. Next, overlapping spheres are eliminated resulting in a good packing. Alternatively, one can use particle simulation approach of Shimada and Gossard together with insertion and removal of spheres (particles) during the simulation [35] . A modified version of the *biting method* in [33] can also be used for maintaining a packing.
Figure 9 illustrates the packing based approach. The model is stretched in the Y direction with a factor of 2, implying larger gaps but no overlaps in the distorted packing. Also note that, since only a few point (sphere) insertions are sufficient to get a good packing, the triangulation does not need to be recomputed. In this case the new triangulation can be computed by point insertion/removal operations on the distorted triangulation, eliminating complete Delaunay re-meshing.

5.1.3 Coarsening via edge contraction

When neither the parametric geometry changes nor the numerical spacing impose smaller elements in the new mesh model, refinement based methods, discussed in Section 5.1.1, might generate over-refined meshes. An example is given in Figure 10. The sphere packing based methods discussed in Section 5.1.2 avoid this problem. This section suggests coarsening strategies to avoid the over-refinement.



(a)



(b)

Figure 10: Refinement algorithm may create over-refined meshes. (a) Original mesh. (b) Refinement based method computes mesh of the same model shrunk in Y direction by a factor of 1/2.

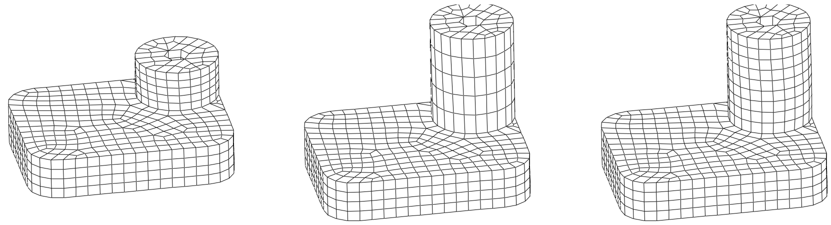
Coarsening strategies are very popular in computer graphics community. These methods aim at mesh simplification for efficient visualization. Both the *progressive meshes* of Hoppe [36] and the *quadratic error metric based simplification* of Garland and Heckbert [37] use heavily the edge contraction operation to coarsen the meshes. This operation removes an edge, merging the two endpoints of it and modifying the connectivity accordingly. The methods can be applied directly to our problem.

5.2 Hexahedral Meshes

When dealing with adjustment of hexahedral meshes the modification options available are much more limited. Automatic hexahedral mesh generation continues to be an open problem [12]. Grid based mesh generation [38] is the only automated hexahedral meshing algorithm for general solid models, mature enough to be used in practice. However, this algorithm tends to generate low quality elements near model boundaries and does not preserve conformity between meshes on adjacent solids. These are severe drawbacks, which prevent its wide use. Tools provided by existing mesh generation software [12, 9], are limited in scope, i.e. they only apply to a subset of geometries. Models that can be meshed automatically typically include uni-axial combinations of swept solids, and sub-mappable solids, i.e. solids which can be decomposed into topological cubes. Due to the difficulty of generating hexahedral meshes in the first place, the available research on modifying such meshes is limited to template based methods [9]. Hence, for the more general cases, new techniques have to be developed.
Below, we propose several new ideas on handling partial re-meshing and modification of hexahedral meshes, given the limited options available.

5.2.1 Partial re-meshing

Partial re-meshing consists of removing a part of the model mesh, followed by generation of a new mesh for the region conforming to the existing mesh. It can be performed when a mesh in a specific region of the model is distorted. This approach was used for adaptation of tetrahedral meshes by Hassan *et. al*[39].
An algorithm which can mesh the empty region conforming to the existing boundary mesh is necessary to perform the re-meshing. The request to maintain conformity with the rest of the mesh, prevents the use of grid based methods [38]. This, since in grid based meshing the mesh is generated from the solid interior outwards, prescribing the boundary mesh. Therefore, only the regions that can be meshed by mapping/sub-mapping or sweep can be re-meshed using most existing software [9]. Hence, as opposed to tetrahedral meshing, a region of poor quality elements can not be re-meshed directly. A larger region, containing it, which fits the constraints imposed by sub-mapping or sweep has to be used. Clearly we would like this region to remain relatively small. Given a region of the mesh it is possible to check if it can be meshed, as many hexahedral mesh-generation codes, such as Gambit [17] by Fluent, Inc. or CUBIT [40] by Sandia National Laboratories, are able, given a solid (with or without some of the model faces meshed ahead), to check if the solid can be meshed by available meshing algorithms. The question to be answered is how to find such meshable regions. An extensive search starting with the badly shaped mesh elements and adding elements one or more at a time, can clearly be exponential.
A possible approach is, given the mesh of the original model, to subdivide it into parts meshable by basic algorithms. Then, given a region of poor shaped elements, find which part (or parts) contain it. Such subdivision is usually generated when meshing the original model. The models are often decomposed by the user in order to use one of those basic algorithms for the meshing. The more advanced meshing techniques, such as methods for meshing a combination of swept volumes [41], are generating such subdivision as part of the meshing process. By taking advantage of this subdivision the re-meshing can be performed automatically. An example of this approach is shown in Figure 11, where the original mesh is subdivided into two sweepable regions. While, this approach often does not produce the smallest possible regions, for complex models it results in significant savings compared to re-meshing the entire model. Methods to detect smaller meshable regions should be investigated to make the partial re-meshing more efficient.



(a) (b) (c)

Figure 11: Adjusting the mesh for a model composed of two sweepable parts. The parameter that changes is the tube height (from 2 to 5). Applying only copy and smoothing will generate highly stretched elements on the pipe. (a) Original model. (b) Mesh after copy. (c) Mesh after re-meshing the pipe part.

5.2.2 Adaptive Hexahedral Meshing

As mentioned earlier, published research on mesh adaptivity is mostly restricted to tetrahedral meshes. Due to the structure of hexahedral meshes, local modifications typically propagate themselves to more than the immediate vicinity. As a result most works on modification of hexahedral meshes, as reviewed by Owen [9], are limited to local, template based changes [42, 43]. Those methods usually have very strict requirements on the mesh structure, and as a result a very limited scope. In this work, we suggest taking advantage of the mesh structure to enable global modifications of the hexahedral mesh, while ensuring the mesh validity.

The Spatial Twist Continuum (STC) was introduced by Tautges *et. al* [44] as a representation of the connectivity of a hexahedral mesh. The STC is an interpretation of the geometric dual of a hexahedral mesh as an arrangement of intersecting surfaces, which bisect hexahedral elements in each direction. Tautges *et. al*[44] called those surfaces *whisker sheets* (Figure 12). The mesh dual is the cell complex induced by the intersection of the whisker sheets. Each whisker sheet is dual to a layer of mesh elements. Each hexahedral mesh element belongs to three whisker sheets corresponding to its three axes. A curve formed by the intersection of two sheets is called a *whisker chord*. A chord represents a column of elements that propagates through the mesh. The STC fully defines the connectivity of a hexahedral mesh. The model mesh can be modified by modifying the STC structure, subject to validity constraints [44]. The constraints for topological validity of a hexahedral mesh based on the STC are:

1. each intersection of a sheet with the geometric model is a closed curve;
2. the intersection of two sheets is a curve which is either circular or has both ends on the outside surface of the model;
3. no more than three sheets intersect together at a point and no more than two along an intersection curve;
4. each sheet has at least one intersection point with two other sheets.

Here, we consider two such modification operations, based on manipulating a whisker sheet (Figure 14). To define operations on a sheet we use the term *whisker sheet axis* as the direction perpendicular to the the mesh layer represented by it (Figure 13).

The two operations are:

- *Sheet removal* - removal of an entire mesh layer associated with a given whisker sheet (Figure 14 (a) to (b));
- *Sheet split* - separation of a layer into two layers, by splitting each hexahedral element into two in the direction of the whisker sheet axis (Figure 14 (a) to (c)).

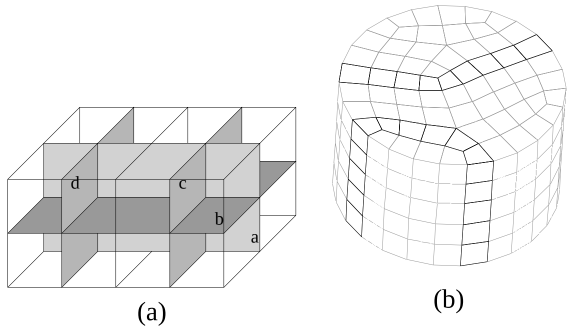


Figure 12: (a) Four whisker sheets (a-d) in a two element hexahedral mesh. (b) Two whisker sheets in a cylinder mesh (highlighted).

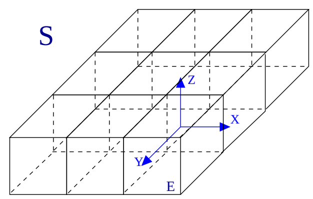


Figure 13: A whisker sheet S with Z pointing in the direction of the sheet axis.

To ensure that the operations preserve a valid mesh connectivity, we need to verify that the four conditions above hold for the mesh after the operation. For a sheet removal, the first three conditions will hold for any removing any sheet. The only condition which needs to be ensured is that each mesh sheet has to have at least one intersection point with two other sheet. Hence a sheet can be removed only if all other sheets have intersection points which do not include it. Example of a sheet which can't be removed is a sheet containing an element with two opposite faces on the model boundary.

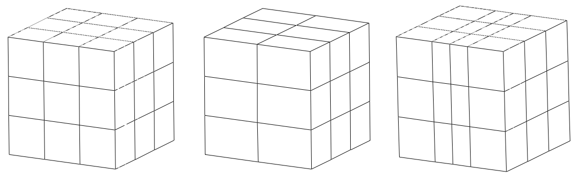
For the sheet split operation, the new sheets are copies of the old one and hence all properties of the old sheet still hold, i.e. they intersect the same sheets the old one did at identical curves and points, hence all the conditions hold. Sheet split can be performed for any whisker sheet in the mesh.

To decide when a whisker sheet should be split or removed we introduce a measure of sheet aspect ratio. It is defined as the average of the aspect ratios of the sheet elements in the direction of the sheet axis. For the example in Figure 13, the sheet ratio of element E is

$$AR = \frac{\sum e_x + \sum e_y}{2 \sum e_z} \tag{3}$$

where $e_j, j \in \{x, y, z\}$ are the lengths of the element edges in the j direction.

The removal and split operations are applied to sheets with extremely high or low aspect ratios respectively. The whisker sheet removal operation leads to the deletion of pressed elements in regions where the parameter change leads to contraction of the model. The sheet splitting leads to the addition of new elements in regions where the model is expanded. A more advanced application of those operations can rely directly on the parameters change, using the ratio of the new and original sizes to decide which operations to perform and how many times.



(a) (b) (c)

Figure 14: Whisker sheet removal and split. (a) Initial model. (b) Model after a sheet removal. (c) Model after a sheet split.

After a sheet removal or split operation, the mesh needs to be smoothed again to diffuse the change to the surrounding area. The whisker sheet based mesh modification is used for the models in Figure 2, Figure 15, and Figure 17. In Figure 2 the sheet split operation is used to split stretched elements into two. In Figure 15 the sheet removal is used to remove layers of pressed elements. The effect of the smoothing can be seen in Figure 15(c) to (d).
These operations are highly successful when the model parameters changed are perpendicular to some of the mesh layers in the modified region. They will be less effective when no such layers exist.

6 Examples

Figure 17 shows the re-meshing process for a crank-shaft model. The meshing of the original model required significant model adjustment. The blends had to be suppressed and the model had to be decomposed into five parts to enable hexahedral meshing using the Cooper Tool algorithm [41] (Figure 17 (b)). The decomposition included separating the two "tips" of the model by using splitting planes and separating the central board from the two side "wheels". Once the editing was completed the mesh was generated automatically. The editing process and mesh generation took about thirty minutes. The editing and meshing were performed using the Gambit analysis pre-processor [17].
The new model (Figure 17 (d)) differs from the original in three parameters. First, the width of the central board was reduced by third. Second, the cylindrical tip on the right was modified, reducing its radius by third and increasing its height by a factor of two. And third, the angle of the blend on the left "wheel" was increased from $10°$ to $15°$. The copied mesh is shown in Figure 17 (e)). Mesh adjustment was required to restore the mesh quality in the central board and the right tip cylinder. The adjustment used the whisker sheet editing operations. The entire procedure can be fully automated as described above, as opposed to the initial mesh generation that required manual model simplification and decomposition.
Figure Figure 16 shows another example of remeshing after parametric modification. In this example a tetrahedral mesh of the model is constructed. The figure shows a meshed part of a booster rocket. It includes the solid propellant and the cavity inside it. The cavity is star-shaped to optimize the burning rate of the propellant. The cavity is also meshed to simulate the behavior of the gases generated during propellant burning. The original mesh of the model in Figure 16 (a) has 6829 nodes and 36726 tetrahedra. Figure 16 (b) shows the model stretched by a factor of 1.5 along X and Z directions. This parametric change causes 1085 of the tetrahedra to become bad, i.e. their circumradius to shortest-edge ratio is larger than 2.5. Delaunay refinement is performed on the distorted mesh by inserting 2170 circumcenter points and all of the bad simplices are removed (see Figure 16 (c)).

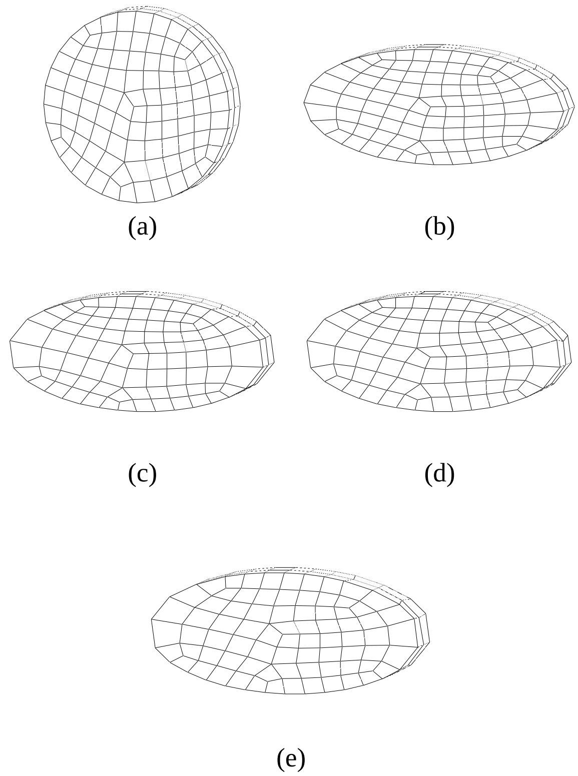


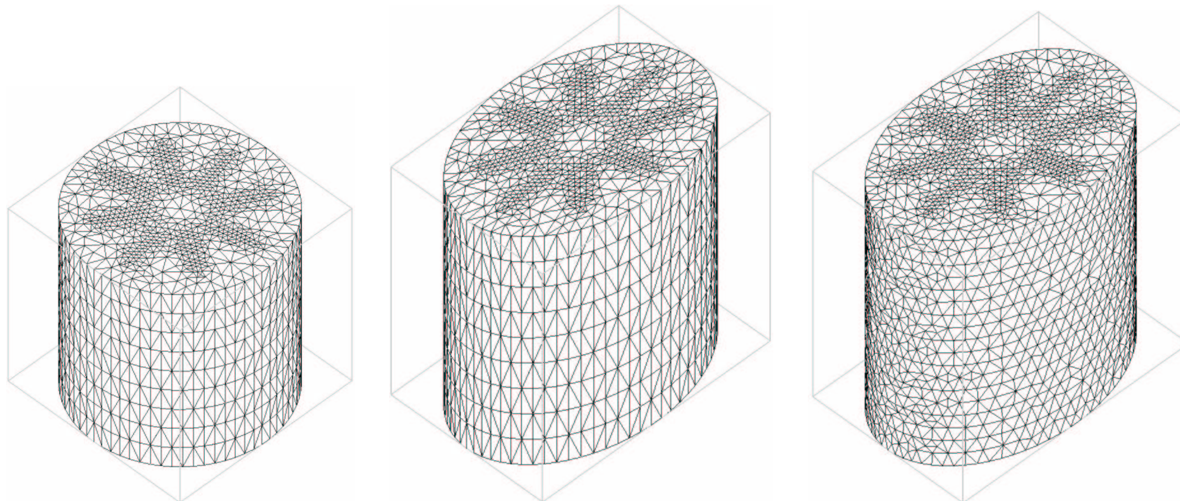
Figure 15: Adapting a hexahedral mesh using the whisker sheet removal operation. (a) Mesh of the original model. (b) Copied mesh of the new model (cylinder radius of 5, replaced by two radii of 7 and 3). (c) Mesh after single sheet removal. (d) Mesh after smoothing. (e) Final mesh after one more sequence of sheet removal and smoothing.

7 Summary

The paper addressed the issues involved in re-meshing CAD models after parameter modification. It demonstrated that using parametric information can eliminate user interaction and significantly shorten the mesh generation process.

The paper presents a comprehensive scheme for re-meshing a model after parameter changes, starting with the geometric adjustments required and concluding with the mesh adaptation necessary to maintain the mesh quality. It examines the application of several existing meshing techniques to this new problem and introduces several new methods for topological mesh adjustment for hexahedral and tetrahedral meshes. Many of the methods referred in the paper were not implemented by the authors and a large amount of work is still required to fully implement the automated mesh adjustment procedure. Implemented and tested parts of the method include geometry adjustment using the virtual topology approach, mesh copy and smoothing, Delaunay refinement, sphere packing, and the whisker sheet based editing of hexahedral meshes. The main ares for future research are quantitative analysis and comparison of the suggested approaches.

This work further encourages the use of the parametric and dual model representations throughout the product data management cycle, by showing the advantages of using the parametric information for applications not considered earlier.



(a) (b) (c)

Figure 16: Volume mesh of part of a booster rocket containing the solid propellant and the star shaped cavity inside it. The fins of the star are about two elements thick. (a) Original mesh with 36726 tetrahedra. (b) Stretching the model along X and Z directions makes about 3% of the tetrahedra bad. (c) Inserting 2170 circumcenters removes all of the bad simplices.

8 Acknowledgments

The work reported here was supported, in part, by the Center for Process Simulation and Design (NSF DMS 98-73945) and the Center for Simulation of Advanced Rockets (DOE LLNL B341494). We would like to thank Damrong Guoy for providing his 3D refinement software to test our algorithms.

References

[1] R. Light and D. Gossard. Modification of geometric models through variational geometry. *CAD*, 14(4):209–214, 1982.

[2] D. Roller. Advances methods for parametric design. *Geometric Modeling Methods and Applications*, pages 251–266, 1991.

[3] S. Raghothama and V. Shapiro. Consistent updates in dual representation systems. *Solid Modeling'99*, pages 65 – 75, 1999.

[4] J.C.H. Chung, T-S. Hwang, C-T. Wu, Y. Jiang, J-Y. Wang, Y. Bai, and H. Zou. Extended variational design technology–foundation for integrated design automation. *Solid Modeling'99*, pages 13 – 22, 1999.

[5] Parametric Technology Corporation. *Pro/ENGINEER*. http://www.ptc.com.

[6] SDRC. *Ideas*. http://www.sdrc.com.

[7] Unigraphics Solutions. *UG/Solid-Modeling*. http://www.ugsolutions.com.

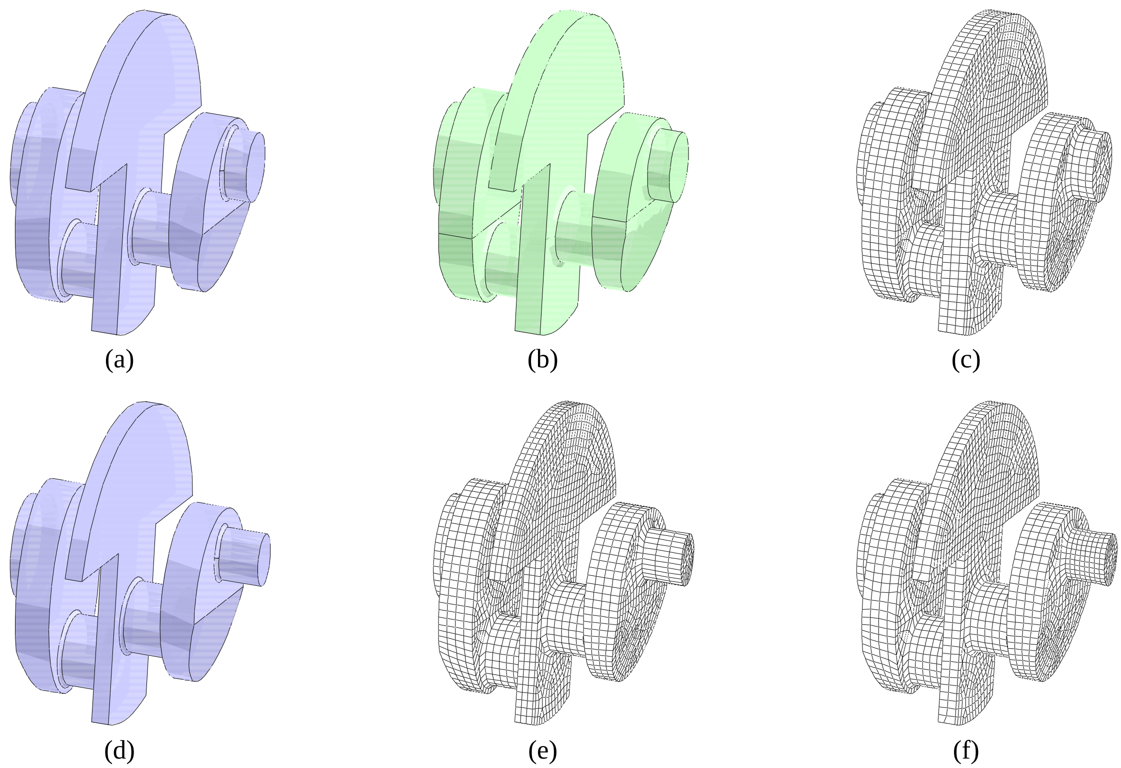


Figure 17: Meshing a crank-shaft model after parameter modification. (a) Original design model. (b) Mesh model, after suppression of blends and subdivision into 5 sub-volumes. (c) Mesh of the original model. (d) New design model. (e) New mesh model with mesh mapped from (c). (f) Final mesh of the new model. The mesh adjustment included removal of a mesh sheet in the central board and several sheet splits in the right cylindrical tip.

[8] M. Halpern. Industrial requirements and practices in finite element meshing: A survey of trends. *6th International Meshing Roundtable*, pages 399–411, 1997.

[9] S. J. Owen. A survey of unstructured mesh generation technology. *7th International Meshing Roundtable*, pages 239–267, 1998.

[10] A. Sheffer, T. D. Blacker, and M. Bercovier. Towards the solution of fundamental issues in cad-fem integration. *Proc. 4th Word Congress on Computational Mechanics*, 1998.

[11] CAD/FE Integration Working Group, NAFEMS Finite Element Methods & Standards. *How to - Integrate CAD and Analysis*, 1996.

[12] T. Blacker. Meething the challenge for automated conformal hexahedral meshing. *9th International Meshing Roundtable*, pages 11–20, 2000.

[13] A. Sheffer, T. D. Blacker, and M. Bercovier. Virtual topology operators for meshing. *International Journal of Computational Geometry and Applications*, 10(2), 2000.

[14] G. Butlin and C. Stops. Cad data repair. *5th International Meshing Roundtable*, pages 7–12, 1996.

[15] J. P. Steinbrenner, N. J. Wyman, and J. R. Chawner. Fast surface meshing on imperfect cad models. *9th International Meshing Roundtable*, pages 33–42, 2000.

[16] V. Capoyleas, X. Chen, and C. M. Hoffman. Generic naming in generative, constraint-based design. *Computer Aided Design (CAD)*, 28(1):17–26, 1996.

[17] Fluent Inc. *GAMBIT CFD Pre-Processor*. http://www.fluent.com/software/gambit/index.htm.

[18] M. L. Staten, S. A. Canann, and S. J. Owen. Bmsweep: Locating interior nodes during sweeping. *7th International Meshing Roundtable*, pages 7–18, 1998.

[19] P. M. Knupp. Applications of mesh smoothing: Copy, morph, and sweep on unstructured quadrilateral meshes. *International Journal for Numerical Methods inEngineering*, 45:37–45, 1999.

[20] H. Edelsbrunner. Triangulations and meshes in computational geometry. *Acta Numerica*, pages 133–213, 2000.

[21] Shang-Hua Teng and Chi Wai Wong. Unstructured mesh generation: Theory, practice, and perspectives. *Int. J. Computational Geometry & Applications*, 10(3):227–266, Jun 2000.

[22] N. A. Calvo and S. R. Idelsohn. All-hexahedral mesh smoothing with a normalized jacobian metric combining gradient driven and simulated annealing. *MECOM 99, Sexto Congreso Argentino de Mecanica Computacional. Mendoza - Argentina*, 1999.

[23] P. M. Knupp. Hexahedral mesh untangling and algebraic mesh quality metrics. *9th International Meshing Roundtable*, pages 173–183, 2000.

[24] P. M. Knupp. Winslow smoothing on two-dimensional unstructured meshes. *7th International Meshing Roundtable*, pages 449–458, 1998.

[25] L. A. Freitag and P. M. Knupp. Tetrahedral element shape optimization via the jacobiandeterminant and condition number. *8th International Meshing Roundtable*, pages 247–258, 1999.

[26] J. Ruppert. A new and simple algorithm for quality 2-dimensional mesh generation. In *Proc. 4th ACM-SIAM Symp. on Disc. Algorithms*, pages 83–92, 1993.

[27] J. R. Shewchuk. Tetrahedral mesh generation by delaunay refinement. In 14^{th} *Annual ACM Symposium on Computational Geometry*, pages 86–95, 1998.

[28] S.-W. Cheng, T.K. Dey, H. Edelsbrunner, M.A. Facello, and S.-T. Teng. Sliver exudation. In *Proc.* $15^t h$ *ACM Symp. Comp. Geometry*, 1999.

[29] H. Edelsbrunner, X.-Y. Li, G. Miller, A. Stathopoulos, D. Talmor, S.-H. Teng, A. Üngör, and N. Walkington. Smoothing and cleaning up slivers. In *Proc.* 32^{nd} *ACM Symp. on Theory of Computing*, 2000.

[30] H. Edelsbrunner and D. Guoy. Sink-insertion for mesh improvement. *Proc.* 17^{th} *ACM Symp. on Computational Geometry*, 2001.

[31] M.-C. Rivara. New longest-edge algorithms for the refinement and/or improvement of unstructured triangulations. *International Journal for Numerical Methods in Engineering*, 40:3313–3324, 1997.

[32] X. Y. Li, S. H. Teng, and A. Üngör. Simultaneous refinement and coarsening adaptive meshingwith moving boundaries. *Engineering with Computers*, 15:292–302, 1999.

[33] X. Y. Li, S. H. Teng, and A. Üngör. Biting: Advancing front meets sphere packing. *International Journal for Numerical Methods in Engineering*, 49:61–81, 2000.

[34] G. L. Miller, D. Talmor, and S.-H. Teng. Optimal coarsening of unstructured meshes. *Journal of Algorithms*, 31:29–65, 1999.

[35] K. Shimada and D.C. Gossard. Bubble mesh: Automated triangular meshing of non-manifold geometry by sphere-packing. *ACM Symposium on Solid Modeling Foundations and Applications*, pages 409–419, 1995.

[36] H. Hoppe. Progressive meshes. *SIGGRAPH96*, pages 99–108, 1996.

[37] M. Garland and P.S. Heckbert. Surface simplification using quadric error metrics. *Proc. SIGGRAPH'97*, pages 209–221, 1997.

[38] R. Schneiders. Automatic generation of hexahedral finiteelement meshes. *4th International Meshing Roundtable*, pages 103–114, 1995.

[39] O. Hassan, K. Sorenson, K. Morgan, and N.P. Weatherill. An adaptive unstructured mesh method for transient flows involving moving boundaries. *Proc. 7th International Conference on Numerical Grid Generation in Computational Field Simulations*, 2000.

[40] Sandia National Laboratories. *Cubit Mesh Generation Toolkit*. http://endo.sandia.gov/cubit.

[41] T. D. Blacker. The cooper tool. *5th International Meshing Roundtable*, pages 13–29, 1996.

[42] S. Mitchell and T. J. Tautges. Pillowing doublets: Refining a mesh to ensure that faces share at most one edge. *4th Internatinal Meshing Roundtable*, pages 231–240, 1995.

[43] R. Schneiders. Refining quadrilateral and hexahedral element meshes. *5th International Conference on Numerical Grid Generation in Computational Field Simulations*, pages 679–688, 1996.

[44] T. J. Tautges, T. D. Blacker, and S. A. Mitchell. The whiskerweaving algorithm: A connectivity-based method for constructingall-hexahedral finite element meshes. *International Journal of Numerical Methods in Engineering*, 39 (19):3327–3349, 1996.