46th AIAA Aerospace Sciences Meeting and Exhibit
7 - 10 January 2008, Reno, Nevada

AIAA 2008-923

Automatic Creation of Quadrilateral Patches on Boundary Representations

John F. Dannenhoffer, III[1]
Syracuse University, Syracuse, NY, 13244

Michael J. Aftosmis[2]
NASA Ames Research Center, Moffett Field, CA, 94035

A major long-standing problem in the grid generation field has been the automatic creation of suitable computational grids directly on the boundary representations (BReps) that are produced by computer-aided design (CAD) systems. Traditionally one of the difficulties has been that the BRep is not necessarily consistent with the engineering intent of the designer, but is strongly influenced by the CAD operations used in the generation of the model. This problem has largely been circumvented by the introduction of quilts and chains that are defined so as to be consistent with the engineering intent. The other major difficulty that still exists in the creation of block-structured grids is the decomposition of the quilts into a set of logically-rectangular patches (quad patches) that completely cover the configuration. ***The objective of this paper is to introduce such a quadrilateral patching technique.*** **This paper describes the process used to automatically generate quad patches, followed by examples from several aerospace configurations. The paper concludes with a description of several benefits of quad patches, namely that they can be used as the basis either for block-structured volume grids or for overset grid systems**

I. Introduction

COMPUTER-AIDED Design (CAD) systems have undergone several paradigm shifts over the past three decades in order to significantly improve their usefulness throughout the entire product development process. The current generation of systems employs solid models as the way parts are defined. A major advantage of a solid model (over its predecessors) is that only physically realizable (although not necessarily manufacturable) objects can be generated. But one of main problems with solid models is that their boundary representations (BREPs) are as much a function of the shape of the body being modeled as it is of the CAD operations used to form the model. It is not uncommon to have different CAD operators generate very different BREPs of the same part.

It is these BREPs that are used to drive structured and unstructured grid generators. The differences in the BREPs can cause enormous difficulties in automated grid generation processes, especially those that generate grids along the edges first, and then generate the grids (triangles) in the faces

II. Summary of Quilts and Chains

To circumvent this problem, a new construct called a “quilt” was introduced in [1]. Just as its name implies, a quilt is a collection of faces (or patches) that are “stitched together” to form a larger entity. As originally implemented, quilts possessed no geometry of their own, but instead referenced the geometry of the underlying faces. From the user’s perspective, a quilt could be viewed as a super-face, extending operations such as “nearest on face” to “nearest on quilt”, which operates on many faces simultaneously.

Subsequently, the concept of a “chain”, which is a series of edges that bound quilts, was introduced in [2]. In particular, a chain is comprised of the complete set of edges that are shared by two quilts. As implemented, chains were generated automatically as a part of the quilt creations process [3].

The two major processes associated with quilts and chains are described in the following paragraphs.

[1] Associate Professor, Mechanical and Aerospace Engineering, AIAA Associate Fellow
[2] Senior Research Scientist, AIAA Senior Member

A. Quilt definition

Given a new configuration, the first decision that needs to needs to be made is which faces should be combined into a quilt. Early implementations of quilts required that the user graphically "select" the faces; this was quickly discarded for two reasons. First, it was a very time-consuming process. But more importantly, it was a very error-prone process, because it was easy for the user to overlook very small faces amongst very large faces – one of the very problems quilts was intended to fix.

In subsequent implementations, automated techniques were employed. At a very high level, the strategy was to examine all the edges in the BREP and determine which of those were "quiltable" (to be described below). Then, all the faces that were separated only by quiltable edges were automatically selected to be combined into a quilt.

An edge was deemed to be "quiltable" if the surface normal of the faces bounded by the edge were nearly parallel. More specifically, the normalized dot-product of the surface normals for each pair of tessellation facets adjacent to an edge were examined, and a parameter α, defined by

$$\alpha \equiv \min_{\text{all facets}} (\text{normalized dot products})$$

is computed. An edge was determined to be quiltable if $\alpha \leq \cos(10^{o})$, although the results tended to be fairly insensitive to the exact angle chosen.

B. Quilt (and chain) creation

Once the faces to be combined into a quilt have been identified, a quilt is created using the following steps:

- Initialize
    - Check the inputs (for example, ensure that selected faces are not disjoint).
    - Classify all edges associated with the faces as either "interior" or "exterior", depending on whether they were shared by the quilt's prospective faces or were along the outside of the quilt.
- Create data structure
    - Concatenate the data structures of the underlying faces into a single quilt data structure.
    - Combine duplicate points (along the "interior edges").
    - Ensure that all neighbor and other information is consistent.
    - Build the smallest number of chains possible out of the "exterior" edges that are not currently part of a chain.
    - Ensure that all pre-existing chains either contain all or no "exterior" edges associated with the new quilt; split any chain that contains both.
- Generate global parameterization
    - Apply each of the following until one succeeds:
        - If quilt is composed of only one face, use the face's parameterization as the global parameterization.
        - If a quilt can be projected onto a plane such that every triangle in the plane has a positive area, use the projected parameterization as the global parameterization.
        - Starting at an arbitrary triangle in he quilt, combine the underlying parameterizations by concatenating the face (u,v) parameters (after appropriately translating, stretching, and rotating)., as proposed by Jones[4].
        - If a quilt can be projected onto a cylinder such that every triangle in the plane has a positive area, use the projected parameterization as the global parameterization.
        - Unroll using the technique described in [3].
    - If all these techniques fail, then the quilt is deemed to be "too complex" and it is divided into sub-quilts and the process continues. Note that this recursive process is guaranteed to terminate since eventually the sub-quilts will be such that they are composed of only one face.
- Finalize
    - Report statistics.

III. Generation of Quad Patches

Once the quilts and chains have been generated for a configuration, each of the quilts is subdivided into a set of **quad** patches. Each quad patch is bounded by four **seams**, and each seam is terminated by two **corners** (which means that each quad is associated with four corners).

The process for generating quads is executed using the quilt's parametric coordinates. Consider Figure 1, in which the triangulation created by CAPRI and used by the quilts is shown in the physical domain in part (a). The same triangulation is shown in Figure 1(b) in the quilt's parametric coordinates. The chains are shown as either red or orange lines (which in this case surround the quilt); the color depends on the direction of the chain. The small black symbols show the endpoints of the chains.

There are three separate processes that are attempted in order, until one generates suitable quads.

A. Quad generation process 1

The first process is a very simple one that works for a very large number of quilts. It turns out that the quilting process frequently generate quilts without holes that have either 1 bounding chain (for example, a circular disc), 3 bounding chains (for example, a triangular region), or 4 bounding chains. In each of these cases, a suitable set of quads can be immediately created, as shown in Fig 2. In each of these cases, the number of vertices along each seam of each quad is computed automatically such that the spacings along the chains match (in a least-squares sense) the nominal spacing specified for the configuration.

B. Quad generation process 2

The basic idea of the second process is to overlay a piece of graph paper on the quilt (in parametric space), and then carve out, or i-blank out, regions that are either exterior to the quilt or lie within a hole in the quilt.

The first step here is to determine the size and orientation of the graph paper that is superimposed on the quilt. This is done by first finding the orientation that maximizes the amount of the quilt boundary that is aligned with either a horizontal or vertical graph-paper line. The graph paper is then trimmed to surround the quilt. Nominally the graph paper resolution is set up so that there are 16 by 16 cells; this resolution can be refined (below) if it turns out that the quilt outer- or hole-boundaries contain more features than can be resolved with the current resolution.

The outer boundary is then traversed, with each facet characterized as being either horizontal or vertical. For every contiguous group of facets that are horizontal, the facets are mapped to a suitable single horizontal line on the graph-paper; an analogous process is followed for the vertical segments. Once the entire outer boundary is mapped, the graph-paper is adjusted to match the quilt's outer boundary and then the interior is elliptically smoothed.

If the quilt contains holes, the process is repeated for each inner boundary (ie, hole) based upon the current (elliptically-smoothed) graph paper.

Finally, the graph-paper is broken into quads that are face-matched, as required above The resulting quads for a typical quilt is shown in Fig. 3.

As one can see, this second process is the most complicated of the processes; but when it works, it generates better quads that the third process (below) will generate and hence is always applied second.

C. Quad generation process 3

The first step in the quad creation is the creation of a ring of seams. For the simplest case in which the quilt has only one (outer) loop of chains, the ring of seams is created by bisecting each chain in the outer loop. The bisection guarantees that there is an even number of seams in the loop. Hence for the configuration shown in Figure 1(b), which consists of 10 chains, the initial ring will consist to 20 seams. (Note that there is apparently a symbol in the middle of the left-hand boundary in Figure 1(b). In actuality there are two symbols very close to each other due to a very small "step" in the definition of the configuration.)

The next step for generating the quads involves filling in the ring with quads for well-known configurations. The well known configurations are:

- if the ring consists of four seams, it is simply filled in with a single quad.
- if the ring consists of six seams, then there are five possible subdivisions (three of which involve two quads each and two of which involve the three quads each), as shown in Fig. 4; the best of these is chosen based upon the "quality" of the resulting of the quads.
- if the ring consists of eight seams, then there are 14 possible subdivisions that result in three or four quads each; again the best "quality" situation is chosen.
- if the ring consists of ten seams, then a subdivision that resembles a pizza pie is attempted and is used if the "quality" is high enough.

- if none of the above succeed, then the ring is split into two rings, each of which is filled with the above techniques. The ring is split by finding a cut that:
  - intersects the seams in the ring the minimum number of times;
  - if a tie, reduced the number of duplicated seams in the sub-rings
  - if a tie, reduces the difference in areas enclosed by the sub-rings (while rejecting splits that cause a negative area)
  - if a tie, has the shortest length
  - if still a tie, chosen arbitrarily.

Once the quilt has been subdivided into quilts, a diagonal-swapping technique is applied to improve the overall "quality". This step is needed in cases where a ring is split and then recursively filled; since the split location is somewhat arbitrary, the diagonal-swapping allows the seams associated with the split to be improved.

Finally, an elliptic grid generation scheme is applied that smooths the quads and all interior seams (and their corners) simultaneously. For the configuration in Fig. 1, the resulting quads are shown in part (c) in the parametric coordinate plane and in part (d) in the physical space.

In the above discussions, the "quality" of a quad is measured by

$$\text{qual} \equiv \frac{4\,\text{area}}{\text{side}_1^2 + \text{side}_2^2 + \text{side}_3^2 + \text{side}_4^2}$$

such that the quality is 1 for a square.

This third process is guaranteed to always generate topologically-valid quads, although the geometric-regularity of the quads is often not as good as the first two processes; it is for this reason that this technique is attempted only if the first two techniques fail.

IV. Sample Configurations

Fig. 5 shows the generation of quad patches for the front part of the Space Shuttle. The configuration, which was modeled in Parasolids, consists of 385 faces that are bounded by 862 edges. As can be seen by part (a), only a few of the CAD edges (such as those around the windscreens) actually represent some engineering intent; the vast majority of the edges are the result of the CAD operations used to create the model. With about 5 minutes of engineering labor, the engineering intent was defined, resulting in the 10 quilts (and 30 chains) shown in part (b). It is easy to see the engineering intent from the figure.

One of the powerful features of quilts is that each one is supported by a global parameterization (g,h) such that there is a one-to-one mapping between (g,h) and each location in the quilt. Part (c) of the Figure show lines of constant g and of constant h one each of the quilts.

Finally part (d) shows the 71 quad patches that were automatically generated. Note that there is grid-point continuity from patch to patch, even when the patches are associated with different quilts. (For example, there is grid-line continuity between the vehicle surface and the windscreen.) For this case, the only user input was the nominal cell size in the quads, which was taken to be 1 percent of the length of the vehicle.

Fig. 6 shows a wing/body configuration composed of 429 faces surrounded by 998 edges in part (a). With less than 5 minutes of engineering labor, the engineering intent was defined and the 11 quilts and 24 chains, as shown in part (b), were generated. Again part (c) shows lines of constant g and h in each of the quilts. Finally, part (d) shows the 52 quad patches that were generated automatically.

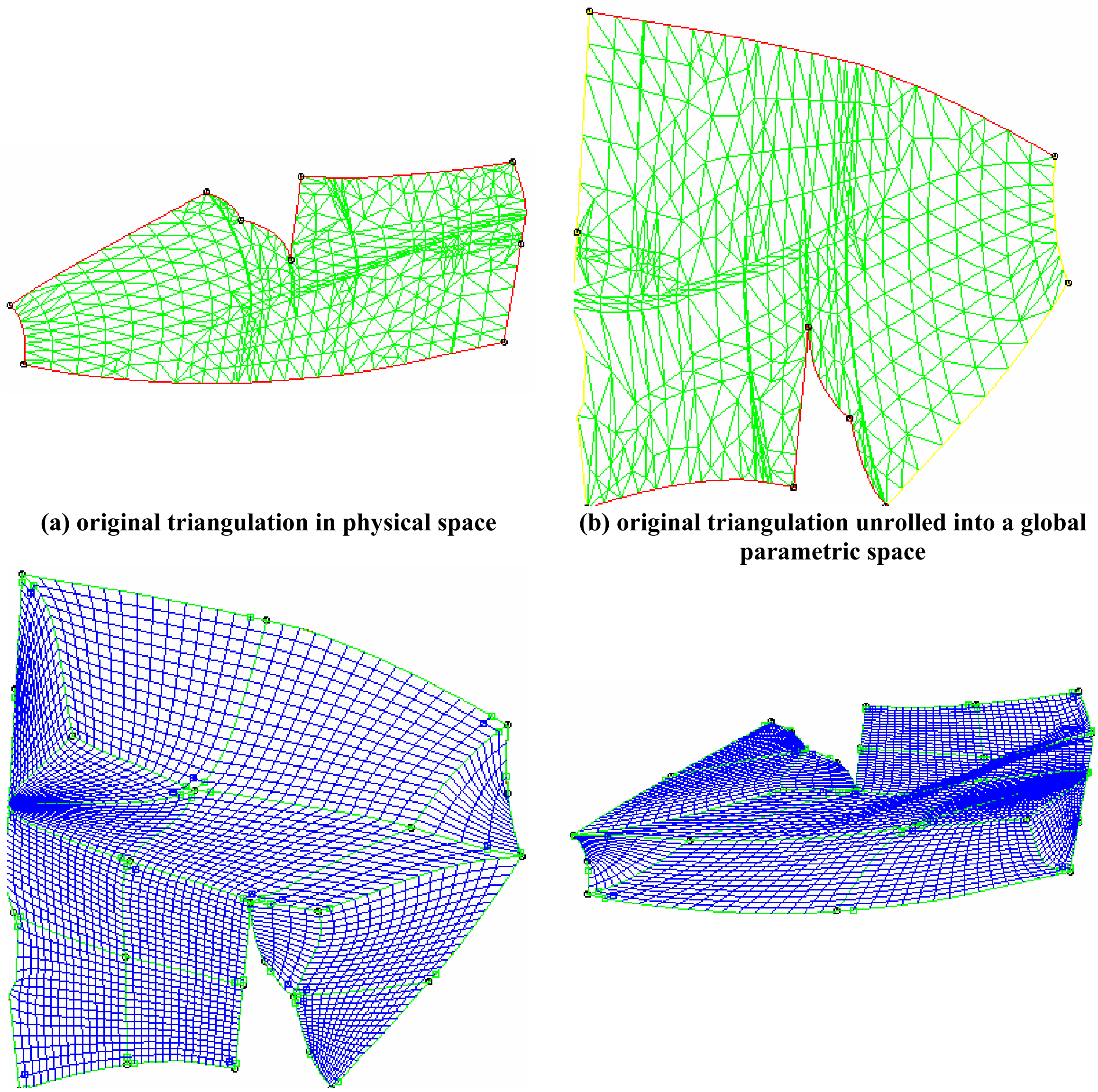
Figs. 7 and 8 show a variety of other configurations to which the quadding algorithm was applied. In each of the cases, one can see the grid-line continuity at all seams between the quad patches.

V. Summary

This paper has described a method for automatically generating logically-rectangular quadrilateral patches that completely cover a CAD-generated boundary representation. The technique is built upon the quilts/chains package that was developed earlier and described elsewhere. Three different techniques are applied, in order, until a suitable set of quad patches exist for each quilt. These quad patches can be used as the basis for a block-structured volume grid generator. While the generation of a suitable volume-grid topology is not trivial, it is expected that many of the tools described here can be extended to the 3D problem. Also, the availability of quad patches on a configuration is a necessary ingredient for the automatic creation of overset grids.

Acknowledgments

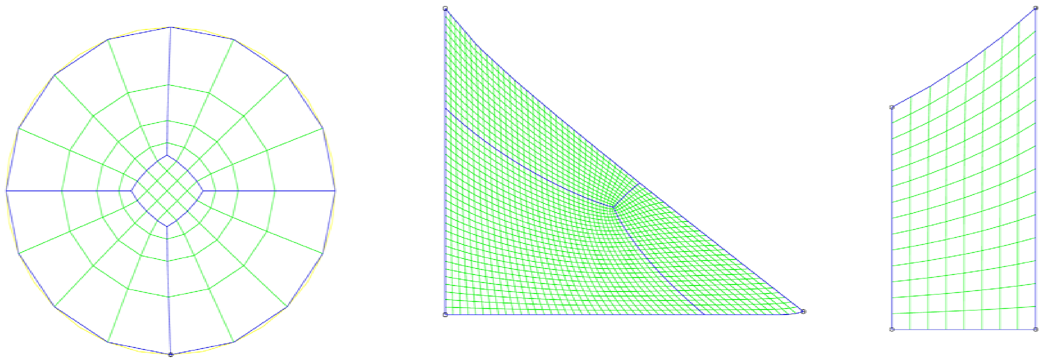
This work was sponsored by the NASA Ames Research Center under grant NNA06CB43G.



(a) original triangulation in physical space **(b) original triangulation unrolled into a global parametric space**

(c) quad patches in the global parametric space **(d) quad patches in physical space**

Figure 1. Various steps in the creation of quad patches for a quilt.



(a) one chain (b) three chains (c) four chains

Figure 2. Templates for creating quads from quilts with 1, 3, or, 4 chains.

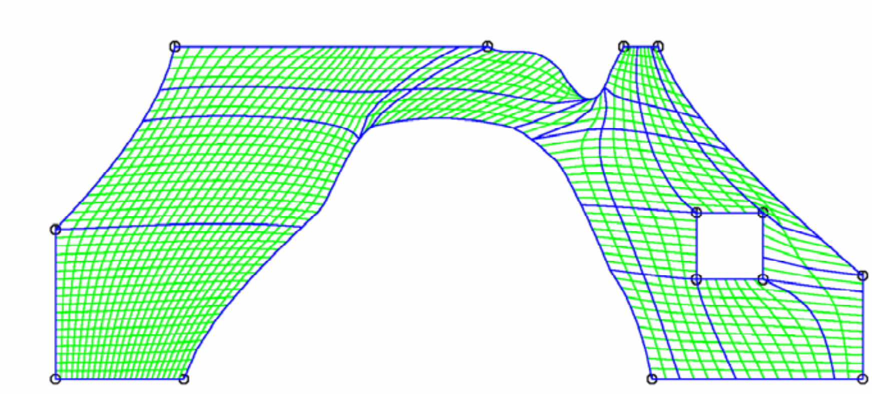


Figure 3. Quads generated for a quilt with many chains and holes.

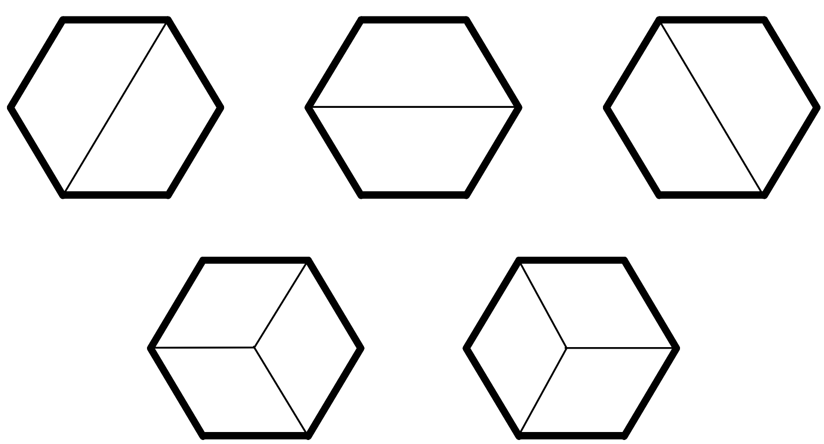
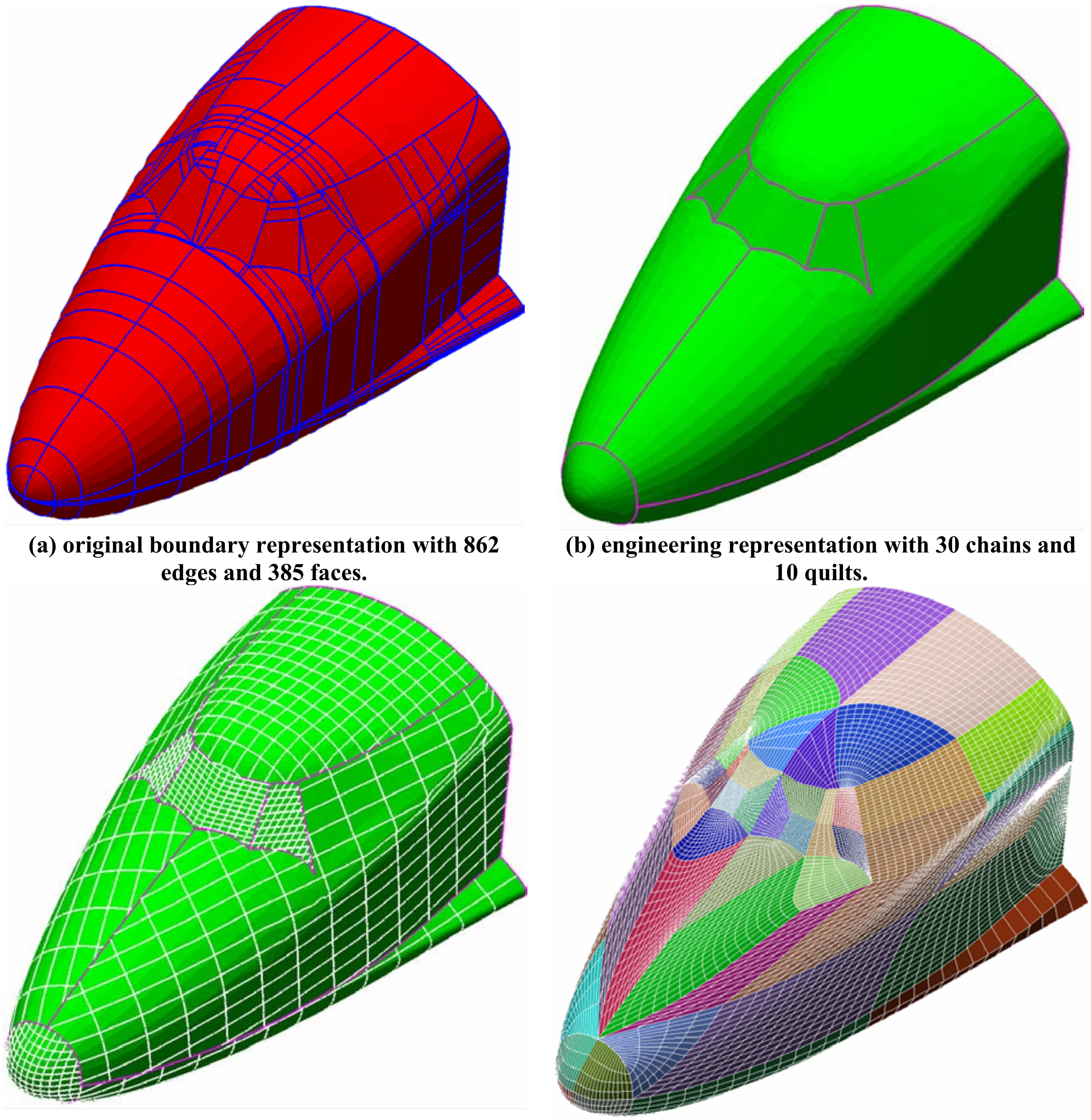


Figure 4. Possible transformation from 6 seams into 2 or 3 quads.



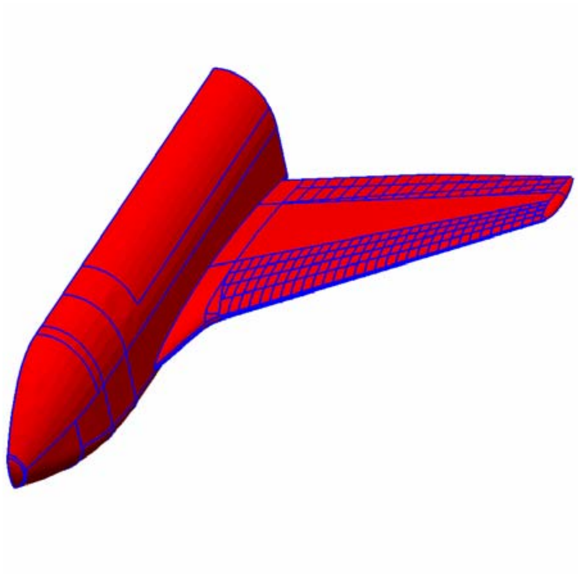
(a) original boundary representation with 862 edges and 385 faces.

(b) engineering representation with 30 chains and 10 quilts.

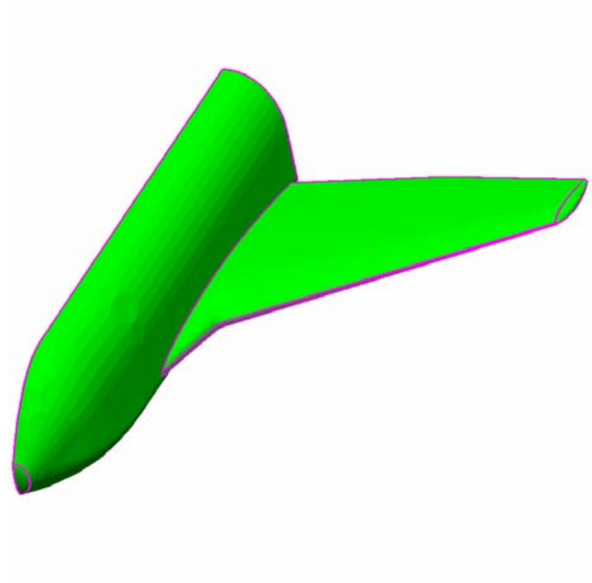
(c) lines of constant parametric values in each of the quilts.

(d) 71 automatically generated quad patches

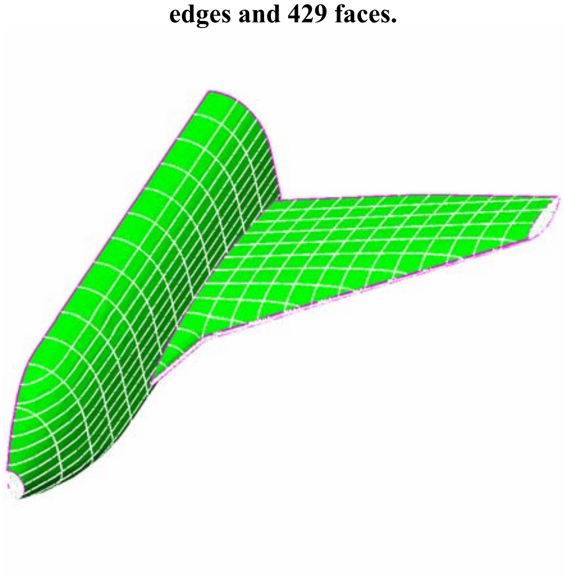
Figure 5. Nose of the Space Shuttle.



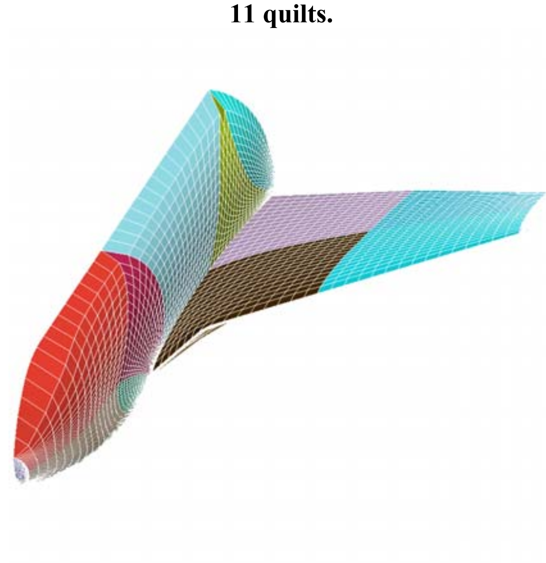
(a) original boundary representation with 998 edges and 429 faces.



(b) engineering representation with 24 chains and 11 quilts.

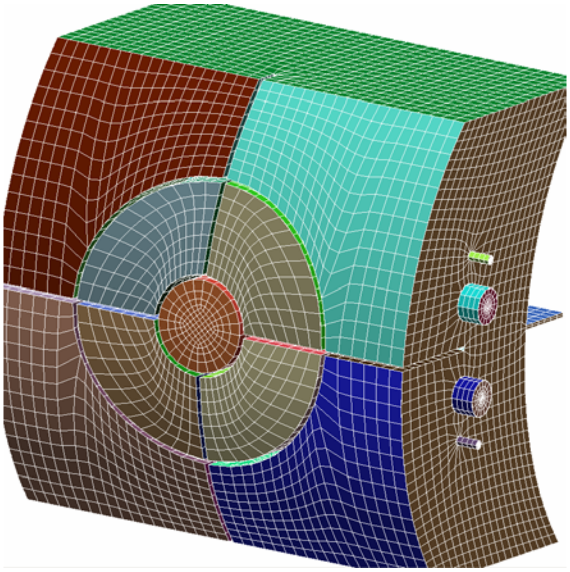


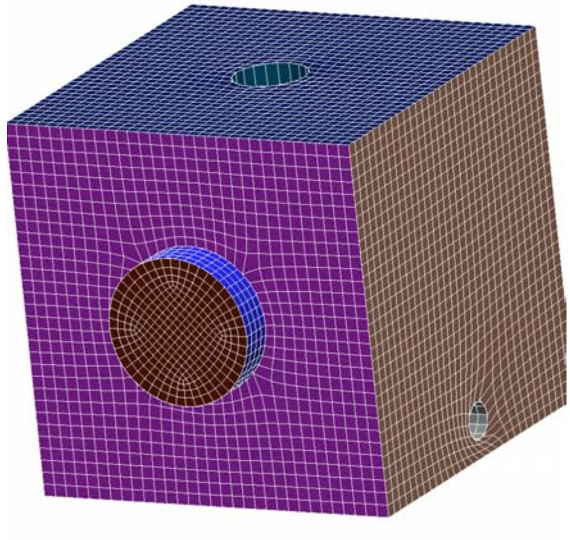
(c) lines of constant parametric values in each of the quilts.



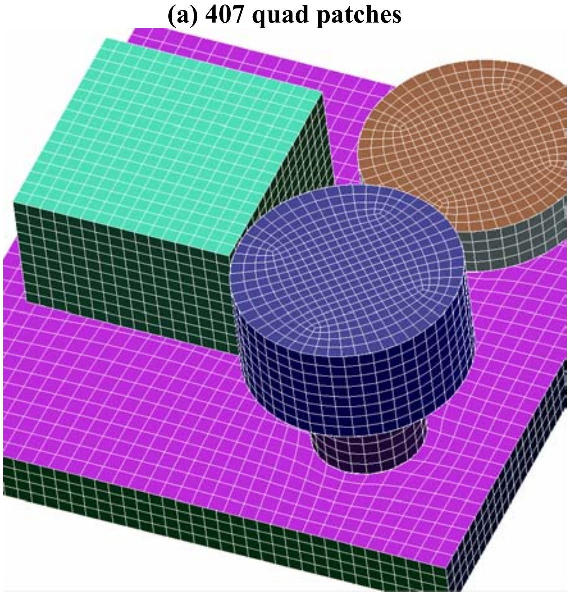
(d) 52 automatically generated quad patches

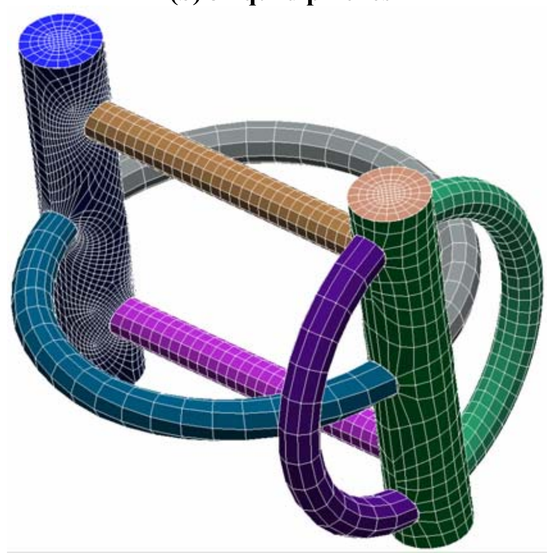
Figure 6. Spaceplane vehicle.





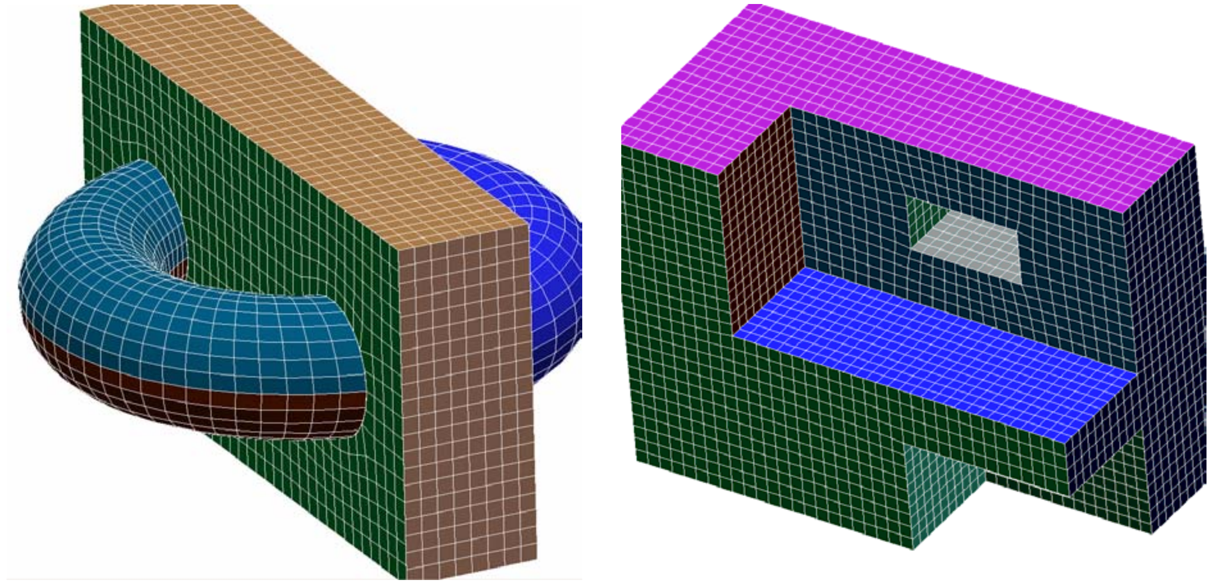
(a) 407 quad patches **(b) 54 quad patches**



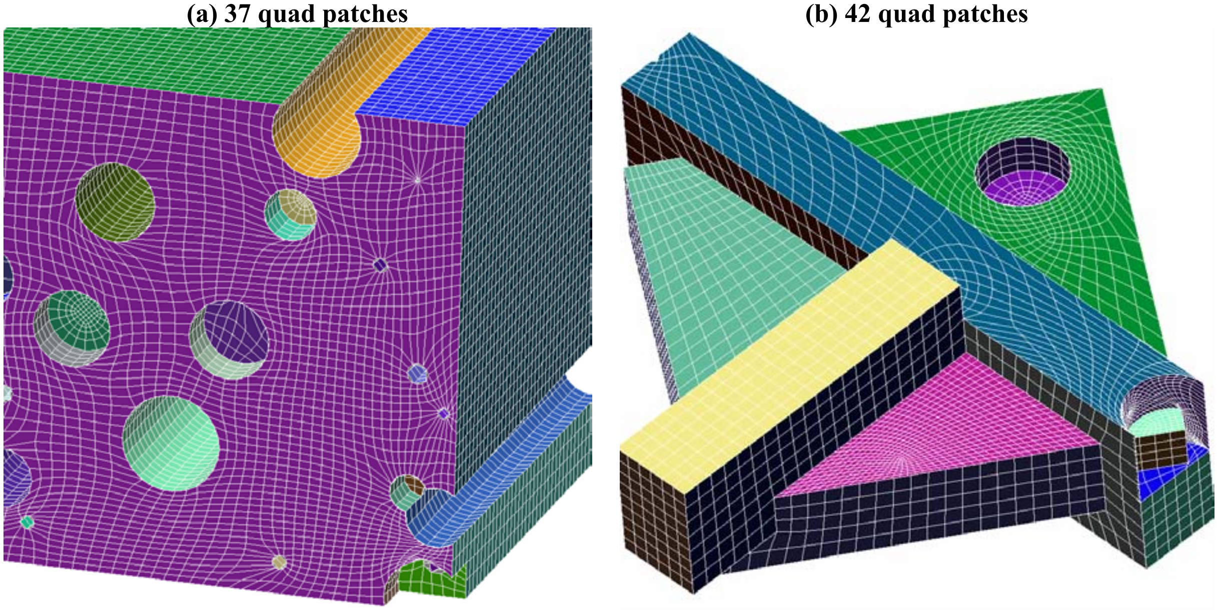


(c) 90 quad patches **(d) 131 quad patches**

Figure 7. Other sample configurations showing automatically-generated quad patches.



(a) 37 quad patches **(b) 42 quad patches**



(c) 767 quad patches **(d) 103 quad patches**

Figure 8. Other sample configurations showing automatically-generated quad patches.

References

[1]Dannenhoffer, J.F. and Haimes, R., "Quilts: A Technique for Improving Boundary Representations for CFD", AIAA-2003-4131, June 2003.

[2]Dannenhoffer, J.F., and Haimes, R., "Using Quilts and Chains to Improve Structured and Unstructured Surface Grids", AIAA-2004-0610, January 2004.

[3]Dannenhoffer, J.F. and Haimes, R., "Robust Algorithms for Generating Quilts and Chains", AIAA-2006-0964.

[4]Jones, W.T., "Toward a Global Parameterization for Quilted CAD Entities," AIAA-2004-0611, January 2004.