42nd AIAA Aerospace Sciences Meeting and Exhibit
5-8 January 2004, Reno, Nevada AIAA 2004-0611

TOWARD A GLOBAL PARAMETERIZATION FOR QUILTED CAD ENTITIES

William T. Jones*
NASA Langley Research Center
Hampton, VA 23681-2199
w.t.jones@larc.nasa.gov

ABSTRACT

A contiguous global parameterization suitable for numerical mesh generation is developed for collections of Computer Aided Design entities. The method is dependent on an underlying closed tessellation of the geometric components which are gathered into quilts to be subsequently treated as a single entity. The result allows for the navigation, evaluation, and calculation of entity derivatives based on the single unified global parameterization for the quilt. The technique has application in the field of numerical mesh generation. Included are unstructured meshing examples which utilize the parameterized quilts to suppress various aspects of the underlying topology defined for the model of interest.

INTRODUCTION

A number of engineering disciplines rely on numerical solutions computed by means of a discrete mesh about the domain of interest. The creation of such a mesh however remains a bottle neck to the overall analysis and design of engineering systems. Much of the time consumed in the mesh construction can be attributed to the definition of the target domain topology and the associated interface to the geometric model of interest. To further complicate matters, there is an ever increasing desire to expand the role of numerical simulation in the overall production process.

The proliferation of Solid Modeling (SM) Computer Aided Design (CAD) systems has brought with it the possibility for the analysis disciplines to leverage existing modeling efforts used in the overall production process. Analysis applications can make direct use of the geometry modeling kernel of the native CAD system through an Application Programming Interface (API) thereby employing the same description and access methods to the model as used by the remaining production components. Basing the analysis (i.e. mesh generation) on the same CAD model used for production maintains accuracy and consistency throughout the overall process.

NASA must contend with a variety of diverse commercial CAD systems as part of its technology assessment and problem solving role. It cannot therefore tailor its analysis capability to the API of a single commercial modeling kernel. A viable alternative is to access the specific modeling kernel through a middleware API that provides a single interface to common operations needed by analysis and design applications. One such example is that of the Computational Analysis and Programming Interface (CAPRI)[1,2]. CAPRI is CAD-vendor neutral yet provides the same access to SM related information through an abstraction that in turn communicates the requirements of the operation in question to the appropriate modeling kernel. CAPRI contains a set of software drivers that contain the actual calls to the modeling kernel API of the originating CAD system. The abstraction however, insulates a derivative application from the nuances in invocation of common operations across the multitude of supported modeling kernels. Support for a new kernel requires implementation of a CAPRI driver for the new system, but does not require any source code changes to derivative applications. CAPRI operations are restricted to manifold solid geometry, such as that defined within a SM system. As such, it provides a closed topological description of the domain of interest. CAPRI also provides a closed tessellation[3] of the subject part that will prove crucial to the current work.

The modeling kernel, and therefore CAPRI, expose the SM constructs inherent in the model and typically used to drive manufacturing. These constructs can also be used to automate the domain definition required by numerical mesh generation and thereby reduce the primary bottle neck to the analysis and design components. Within the definition of manifold solid models is a closed topology suitable for manufacture, but also required by many analysis disciplines. This topology can be automatically extracted for use in the domain definition during mesh generation. The

*Computer Engineer, Data Analysis and Imaging Branch primary benefit of automated topology extraction is a significant reduction in the turnaround time required for mesh generation.[4] Unfortunately, when a CAD model is designed solely for manufacture, or another component process, it may possess features or design intent that are contrary to that desired for other disciplines. The topological definition automatically extracted to drive the mesh generation might be overly complex or possess features whose detail is not required by the target analysis. Traditionally this would force topological changes to the part in order to facilitate analysis. Such changes negate the benefits of accuracy and consistency as distinct models must be constructed for the various phases of the production process.

A recent enhancement to the CAPRI interface is the development of an additional package for quilting topological entities together.[5] The quilting process serves to collect and combine like entities which meet a set of specified constraints (such as mating angle). The collection can then be viewed as a single entity thereby suppressing interior entities lower in the topological hierarchy (nodes when grouping edges, and edges for groups of faces). This allows the analysis applications to modify not the model proper, but simply the topology that is to be extracted for automated domain definition.

With the addition of quilts, the analysis application is afforded a great deal of latitude in making the necessary modifications to the topology that are needed to drive the analysis itself. The quilting technique provides a mapping of functionality for the associated CAPRI methods (evaluation, snap, etc.). Quilts as currently defined, however, lack a single underlying parameterization. As such, there is not a one-to-one mapping of topology to geometry. Evaluations must be conducted with an understanding of which underlying component (edge/face) to use along with its local parameters. Many applications, such as mesh generation, may also benefit or even depend on such a global parameterization of the entity. A global quilt parameterization makes possible consistent derivative calculation that can be used when navigating the physical space and provides for an argument-for-argument mapping of the associated CAPRI functions. With a consistent global parameterization, it is also possible to map a mesh following a geometry change that does not alter the topology of the abstracted quilt. What follows is a description of the development of a unified global parameterization for CAD quilts which provides not only simple navigation, but also evaluation of entity derivatives used to drive the mesh generation itself. Unstructured meshing examples are provided to demonstrate the applicability of the resulting parameterizations.

CAPRI

The CAPRI software provides access to the geometric model in question via the native API of the originating modeling kernel used for model construction. Access to the CAD model is provided through a consistent software interface that is independent of the underlying modeling kernel. Both geometric and topological access is provided. The model topology is represented by a hierarchical series of dependent entities consistent with Computational Solid Modeling (CSM) terminology. Primary topological entities include 0-D nodes, 1-D edges, and 2-D faces. Additional topological entities include closed collections of edges called loops and collections of faces that are used to define a 3-D manifold region known as a volume. The hierarchical nature of the topology definition provides entities that are bounded by those of the next lower dimension. For example, each edge is bounded by two nodes, faces are bounded by loops and therefore by their respective edges, and volumes are bounded by the collection of faces. The manifold nature of construction ensures that within a given volume, each face has a single reference and every edge is used by exactly two faces with each use being of opposite sense. Each primary topological entity is supported by an underlying geometric entity of the same dimension providing the numerical definition of shape. The geometric entities include points, curves, and surfaces that support the node, edge, and face topological entities. The geometric entities are defined parametrically and may be analytic, polynomial, B-Spline, and so on. Topological queries of the model are provided by the CAPRI API. These queries provide a means of navigating through the model space. Physical coordinates and properties may be evaluated from the parameter space of the supporting geometry. The API also allows snapping points to topological entities, and thereby their supporting geometry, along with the evaluation of entity derivatives, tangent and normal vectors, curvatures, and so on.

The CAPRI software also provides a closed (i.e. watertight) tessellation for each entity in the SM. The tessellation is closed in the sense that the number and physical location of the tessellation points on the boundary of each face are the same as those of the corresponding edge tessellations that make up the bounding loops for the face. Likewise, the points at the beginning and end of the edge tessellation share the same physical coordinates as the points constituting the bounding nodes. Therefore, there are no gaps or overlaps between entities. The tessellation provides not only the physical coordinates of the vertices, but also the parametric coordinates relative to the underlying geometry supporting the topological entity. The initial CAPRI tessellation loosely follows from the distribution of surface curvature, but may be refined through the API via the specification of additional criteria such as normal deviation, chordal deviation, and maximum element side length. The CAPRI tessellation, including the parametric coordinates, will prove essential to the current work.

QUILTS

An additional package is available to supplement the basic CAPRI functionality with the ability to create topological quilts. Quilts are used to further abstract the model topology in terms of an "engineering topology" used for analysis. A quilt is a grouping of topological faces from the SM. The faces are joined together based on common edge entities. Quilts themselves are purely topological entities and possess no geometry of their own, but instead rely on the underlying geometry supporting the constituent faces. As such, the quilting process does not alter the geometry, but rather offers up an alternative view of the connectivity of entities to the analyst. Through quilting, topological entities can be grouped together to form a larger virtual entity that essentially suppresses the lower level components contained within (e.g. common edge to two faces).

Quilts are bounded by chains. Chains are the 2D analogue of quilts and represent a contiguous collection of edges. Unlike the loops defined by CAPRI, however, chains need not be closed. Topological nodes internal to the chain may be considered suppressed by the user. As the first quilt is created, it is bounded by a single chain (provided there are no internal holes in the collection of faces). The newly created chain maintains a pointer to the quilt on its left relative to its orientation. The addition of subsequent quilts will automatically subdivide chains at quilt junctions such that each chain has a unique left and right quilt association with regard to the natural chain orientation. The chains resulting from the split will have the right association assigned to the newly added quilt that forced the split. The right quilt association will naturally use the chain in the opposite sense of its orientation.

The current quilts package assembles the topological entities into a contiguous collection. As entities are collected and grouped into quilts and chains, the associated tessellations are also combined into a single comprehensive tessellation for the collection. This process includes the removal of duplicate vertices along lower level entities and the corresponding update of tessellation connectivity. Like the basic CAPRI tessellations, quilt tessellations will prove crucial to the current work.

PARAMETERIZATION

Evaluation of the quilt via the standard quilts package involves an extension to the equivalent CAPRI routines. The extension requires additional information regarding a particular basic component (edge or face) and relative local parameters for which the evaluation is to take place. Many operations, such as mesh generation, can benefit from the additional information provided by a global parameterization for quilts and chains. For example, geometric quantities (physical coordinates, normal and tangent vectors, curvatures, etc.) can be evaluated directly by way of global parameters. This in turn makes possible the computation of the derivatives of the physical coordinates with respect to the global parameters. It also makes possible a direct argument-for-argument mapping of the respective CAPRI API for the related function. As such, it facilitates the incorporation of quilting technology into an existing CAPRI based algorithm. Operations are greatly simplified by working in the global parameter space. It is no longer necessary to provide a basic component when evaluating a parameterized quilt or chain as parameters are consistent across the entire collection. Algorithms may track changes in the physical coordinates with respect to changes in the global parameters thereby greatly simplifying their implementation. It is therefore advantageous to extend the existing quilts package of CAPRI in order to provide a global parameterization of quilted entities.

Each component entity of a quilt is in fact defined with a separate local parameterization. However, the parameter space tessellations of the separate entities may be disjoint, overlapping, and of different orientation. We seek to join these tessellations into a single contiguous parameter space triangulation that can be used to define the desired global parameter space.

Other works in this area have used "flattening" techniques to define similar parameterizations for piecewise representations of physical objects.[6,7] Fortunately, the triangulations that we are confronted with are already defined in the parameter plane. An additional advantage afforded by the tessellations provided with CAPRI and the quilting package is the closed nature of the overall tessellation of the model. Therefore, for each disjoint entity we have a direct mapping of the boundary vertices with those on the adjacent entity.

The approach used here is to select one of the tessellations within the quilt as the "basis" tessellation. The basis tessellation defines the initial global parameter space. Along the boundary of the basis tessellation where an adjacent entity is to be added, we are provided with both the basis (global) parameters and the local parameters of the new entity.

When collecting edges into chains, the basis is simply selected as the first edge in the chain. The construction of a global parameter starts at the last vertex of the basis tessellation (taking into account edge orientation). The procedure is to step through the remaining vertices in the chain's tessellation and add the difference in the local parameter over each interval to the global value of the previous vertex. Note that at the bounding nodes of an edge, two local parameter values are known relative to the two edges of the chain that are incident to the node. The local parameter used depends on the context of the interval used to compute the difference.

The basis for a quilt is arbitrarily selected as the tessellation with the largest number of vertices. Triangles of additional component face tessellations are added in turn starting with those that share a segment with the basis. Such a triangle has known local parameters at all three vertices, but global parameters are known only at the two vertices constituting the side of the triangle shared with the basis triangulation. We can determine the unknown global parameters by inserting a similar triangle in the global tessellation. A similar triangle is defined by maintaining the ratio of the lengths of the triangle sides. Given two triangles: $\Delta \boldsymbol{ABC}$ in the global parameter space and $\Delta \boldsymbol{DEF}$ in the local parameter space, if $\Delta \boldsymbol{ABC} \cong \Delta \boldsymbol{DEF}$ then

$$\frac{\|\boldsymbol{AB}\|}{\|\boldsymbol{DE}\|} = \frac{\|\boldsymbol{BC}\|}{\|\boldsymbol{EF}\|} = \frac{\|\boldsymbol{AC}\|}{\|\boldsymbol{DF}\|} \qquad [1]$$

Assuming that the known global parameters are defined for two points $\boldsymbol{A}$ and $\boldsymbol{B}$ with the corresponding local triangle points $\boldsymbol{D}$ and $\boldsymbol{E}$, equation [1] can be solved for the unknown global parameters at point $\boldsymbol{C}$.

The tessellation of each entity in the quilt is combined with that of the basis by means of an advancing front technique initiated at the basis boundary. The unknown global parameter values for triangles incident to the front are calculated such that the global parameter values define a similar triangle to that defined by the original local parameterization. Vertices with multiple incident triangles that provide a side on the current front are resolved using a linear combination of the candidate global values determined from each triangle. The coefficients of the combination are based on a measure of the local triangle distortion to be described in the next section.

While the technique for generating the topological quilt abstraction is extremely robust, the current method of constructing its associated global parameterization is not without limitation. The use of similar triangles described above assumes that scaling of the local parameter space is uniform. If any shearing is required to map the local parameter space into the global space the method can fail. This can arise from the need to attach two adjacent faces which have different scaling in the traverse direction with respect to the basis. Such is depicted in Figure 1. Here two triangles are to be added to the basis. Triangle 1 is to share the segment **ab** and triangle 2 is to share segment **bc** with the basis (shown as the bold line). As can be seen in the figure,

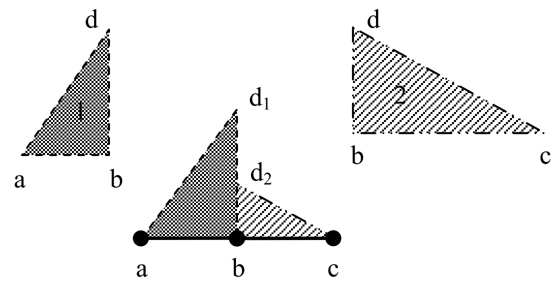


Figure 1 – Limitation of Similar Triangles

the independent use of similar triangles for the insertion of elements 1 and 2 results in quite different solutions for the global parameters at **d**. In some instances it may be possible to resolve this difference with the before mentioned distortion based weighting, however this is not always suitable.

To circumvent this limitation, the current methodology requires intelligent choice of the candidate faces for quilting such that this situation is avoided (i.e. simply tiling faces end to end). Typically this will result in the desired suppression of unwanted topology making mesh improvement possible.

To facilitate the construction and use of the global parameterization, the quilt assembly process has been augmented such that it preserves the local parametric coordinates relative to the supporting geometry of each component of the quilt. The local parameters of every vertex in the original combined tessellation are stored, including boundary duplicates. The unaltered connectivity of the combined tessellation is also stored to reference the local parameter list while maintaining a one-to-one mapping of triangles to the altered connectivity that references unique vertices. Also preserved is the entity identifier for each element in the combined tessellation. As such, a piecewise local parameterization is maintained for the quilted entities. A projection to the combined tessellation is used to interpolate the local parameters relative to the underlying component entity and to retrieve the identifier. The local parameters and the identifier are then used in standard evaluations via conventional CAPRI invocation. As such no approximation of the geometry supporting the quilt is needed.

MAPPING AND DERIVATIVES

In order for the global parameterization to be of full use, derivatives with respect to the global parameters must also be provided. The global parameterization defines an affine mapping of the local parameter space[6,7] with two dimensional simplifications afforded by the parameter plane.

Let $\boldsymbol{u}_i = (\mathrm{u}_i, \mathrm{v}_i)^T$ represent a point in the local parameter space and $\boldsymbol{s}_i = (\mathrm{s}_i, \mathrm{t}_i)^T$ be a point in the global parameterization. Let $\boldsymbol{U}(\boldsymbol{s}_i) = \boldsymbol{u}_i$ define an affine map between the global and local parameter spaces. We denote the area of the triangle $\Delta \boldsymbol{s}_i \boldsymbol{s}_j \boldsymbol{s}_k$ as

$$\langle \boldsymbol{s}_i, \boldsymbol{s}_j, \boldsymbol{s}_k \rangle = \frac{(\mathrm{s}_j - \mathrm{s}_i)(\mathrm{t}_k - \mathrm{t}_i) - (\mathrm{s}_k - \mathrm{s}_i)(\mathrm{t}_j - \mathrm{t}_i)}{2} \qquad [2]$$

Given any global point $\boldsymbol{s}$ bounded by the triangle $\Delta \boldsymbol{s}_i \boldsymbol{s}_j \boldsymbol{s}_k$, a local point $\boldsymbol{u}$ is then determined as the barycentric combination of the values defining the same triangle in the local parameter space, triangle $\Delta \boldsymbol{u}_i \boldsymbol{u}_j \boldsymbol{u}_k$.

$$\boldsymbol{U}(\boldsymbol{s}) = \frac{\langle \boldsymbol{s}, \boldsymbol{s}_j, \boldsymbol{s}_k \rangle \boldsymbol{u}_i + \langle \boldsymbol{s}, \boldsymbol{s}_k, \boldsymbol{s}_i \rangle \boldsymbol{u}_j + \langle \boldsymbol{s}, \boldsymbol{s}_i, \boldsymbol{s}_j \rangle \boldsymbol{u}_k}{\langle \boldsymbol{s}_i, \boldsymbol{s}_j, \boldsymbol{s}_k \rangle} \qquad [3]$$

Thus the partial derivatives of $\boldsymbol{U}$ are:

$$\frac{\partial \boldsymbol{U}}{\partial \mathrm{s}} = \frac{(\mathrm{t}_j - \mathrm{t}_k)\boldsymbol{u}_i + (\mathrm{t}_k - \mathrm{t}_i)\boldsymbol{u}_j + (\mathrm{t}_i - \mathrm{t}_j)\boldsymbol{u}_k}{2\langle \boldsymbol{s}_i, \boldsymbol{s}_j, \boldsymbol{s}_k \rangle} \qquad [4]$$

$$\frac{\partial \boldsymbol{U}}{\partial \mathrm{t}} = \frac{(\mathrm{s}_k - \mathrm{s}_j)\boldsymbol{u}_i + (\mathrm{s}_i - \mathrm{s}_k)\boldsymbol{u}_j + (\mathrm{s}_j - \mathrm{s}_i)\boldsymbol{u}_k}{2\langle \boldsymbol{s}_i, \boldsymbol{s}_j, \boldsymbol{s}_k \rangle} \qquad [5]$$

As stated above, a measure of the local triangle distortion is needed to resolve conflicts encountered during the addition of vertices. The local geometric distortion of the mapping is defined in terms of the singular values γ of the Jacobian matrix for the map $\boldsymbol{U}(\boldsymbol{s})$.[6]

We define the Jacobian matrix

$$\boldsymbol{J} = \begin{bmatrix} \frac{\partial \mathrm{u}}{\partial \mathrm{s}} & \frac{\partial \mathrm{u}}{\partial \mathrm{t}} \\ \frac{\partial \mathrm{v}}{\partial \mathrm{s}} & \frac{\partial \mathrm{v}}{\partial \mathrm{t}} \end{bmatrix} \qquad [6]$$

Therefore,

$$\gamma_{\min} = \sqrt{\frac{1}{2}\left((a+c) - \sqrt{(a-c)^2 + 4b^2}\right)} \qquad [7]$$

$$\gamma_{\max} = \sqrt{\frac{1}{2}\left((a+c) + \sqrt{(a-c)^2 + 4b^2}\right)} \qquad [8]$$

where,

$$a = \frac{\partial \boldsymbol{U}}{\partial s} \cdot \frac{\partial \boldsymbol{U}}{\partial s}, \quad b = \frac{\partial \boldsymbol{U}}{\partial s} \cdot \frac{\partial \boldsymbol{U}}{\partial t}, \quad c = \frac{\partial \boldsymbol{U}}{\partial t} \cdot \frac{\partial \boldsymbol{U}}{\partial t} \qquad [9]$$

We use the ratio of the $\gamma_{\max}$ to $\gamma_{\min}$ to construct the coefficients so as not to penalize the isotropic scaling of the triangles as a result of their embedding in the global space.

Determination of derivatives with respect to the global parameters is achieved by simple application of the Chain Rule. Given a scalar function $\boldsymbol{F}(\boldsymbol{u})$, its derivative with respect to the vector $\boldsymbol{u}$ is the vector,

$$\nabla_{\boldsymbol{u}} \boldsymbol{F} = \left(\frac{\boldsymbol{dF}}{\boldsymbol{du}}\right)^{\boldsymbol{T}} = \begin{bmatrix} \frac{\partial \boldsymbol{F}}{\partial \mathrm{u}} \\ \frac{\partial \boldsymbol{F}}{\partial \mathrm{v}} \end{bmatrix} \qquad [10]$$

The derivative of $\boldsymbol{F}$ with respect to $\boldsymbol{s}$ can be expressed explicitly in terms of the [6] and [10] via the chain rule,

$$\frac{\partial \boldsymbol{F}}{\partial \boldsymbol{s}_i} = \frac{\partial \boldsymbol{u}_j}{\partial \boldsymbol{s}_i} \frac{\partial \boldsymbol{F}}{\partial \boldsymbol{u}_j} = \boldsymbol{J}_{ji} \frac{\partial \boldsymbol{F}}{\partial \boldsymbol{u}_j} \qquad [11]$$

Equation [11] is equivalent to,

$$\left(\frac{\boldsymbol{dF}}{\boldsymbol{ds}}\right)^{\boldsymbol{T}} = \boldsymbol{J}^{\boldsymbol{T}} \left(\frac{\boldsymbol{dF}}{\boldsymbol{du}}\right)^{\boldsymbol{T}} \qquad [12]$$

The second and cross derivatives with respect to $\boldsymbol{u}$ can be written in matrix form as the outer product of the derivative vectors operating on $\boldsymbol{F}$,

$$\frac{d^2\boldsymbol{F}}{du^2} = \left(\frac{d}{du}\right)^T \left(\frac{d}{du}\right)\boldsymbol{F} = \begin{bmatrix} \frac{\partial^2 \boldsymbol{F}}{\partial u^2} & \frac{\partial^2 \boldsymbol{F}}{\partial u \partial v} \\ \frac{\partial^2 \boldsymbol{F}}{\partial v \partial u} & \frac{\partial^2 \boldsymbol{F}}{\partial v^2} \end{bmatrix} \qquad [13]$$

Introducing the Jacobian from [6], we can rewrite equation [13] in terms of the variables $\boldsymbol{s}$ as,

$$\frac{d^2\boldsymbol{F}}{ds^2} = \left(\frac{d}{ds}\right)^T \left(\frac{d}{ds}\right)\boldsymbol{J}^T \left(\frac{d}{du}\right)^T \left(\frac{d}{du}\right)\boldsymbol{JF}$$

$$= \boldsymbol{J}^T \frac{d^2\boldsymbol{F}}{du^2}\boldsymbol{J} \qquad [14]$$

Use of the Chain Rule allows the computation of geometric derivatives relative to the global parameterization. Both first and second derivatives with respect to the global parameters are calculated. As stated above, the derivative information can simplify many algorithms and once provided, completes the direct argument-for-argument mapping of operations on parameterized quilts to the respective operations for basic CAPRI entities of the same dimension.

MESHING

Mesh generation on quilted entities is facilitated by the definition of strict argument-for-argument replacements for the respective CAPRI functions. Operations on parameterized quilts use the same interface as the CAPRI entity of the same dimension (i.e. chains->edges, quilts->faces). To implement an algorithm with the parameterized quilts, a developer need only change the name of the function, formally called from the standard CAPRI API, to that of the parameterized quilts API. Return values (coordinates, derivatives, length, tangent, curvature, etc.) can be treated as if they were calculated from a single underlying CAPRI entity.

The examples to follow employ a derivative of the FELISA[8] curve and surface meshing algorithms. These algorithms utilize the parameter space to conform to the target geometry. The mesh is generated using a localized parameter mapping that requires the computation of entity derivatives and subsequent evaluation of curve tangent and surface normal vectors. These algorithms would require extensive modification without the use of a globally parameterized quilt. However, the current work makes possible the direct substitution of the parameterized quilt routines for the traditional CAPRI functions as mentioned above. In fact, to produce the results to follow, a simple wrapper is used to determine which routine to invoke (standard CAPRI or parameterized quilts) based on the entity identifier that is requested. The algorithms previously employed wrappers for CAPRI routines, to allow for error handling and the like, resulting in seamless integration of parameterized quilts capability into the existing mesh generators.

An additional step in making the parameterized quilts mimic the standard CAPRI API required the assembly of chains into closed boundary loops of the quilt. The loops of chains are used to orient the quilt and assemble the boundary discretization for input to the FELISA surface meshing algorithm.

EXAMPLES

The following examples represent the results of both the globally parameterized quilting capability and its use to generate unstructured curve and surface discretizations. Figure 2 represents a sample geometry used for testing the quilts package of CAPRI. This geometry has been specifically constructed to contain

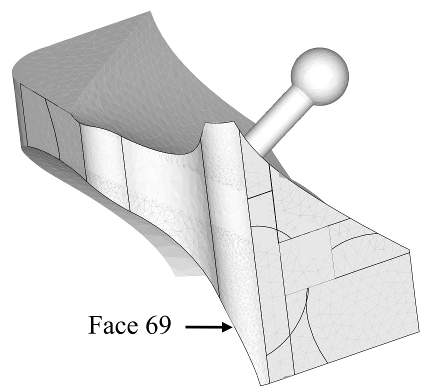


Figure 2 – Sample quilting part

numerous distinct faces resulting in a topology that is far more complex than that required for mesh generation. We consider a collection of 22 solid model faces outlined in the figure. The square omitted in the center of the collection represents a small wafer like protrusion from the main body that is not selected as part of the current quilt. The associated local parameter space for each face of the quilt is shown in Figure 3. In this figure, the tessellation is relative to the underlying support surface of each face. Note that the faces are irregular, disjoint, and in some cases overlap when viewed in the parameter plane. Figure 4 shows the resulting quilt for this collection of faces. Notice that the edges of the component faces internal to the resulting final quilt are suppressed. Figure 5 shows the unified triangulation in the computed global parameter space for the quilt. The tessellation for face 69

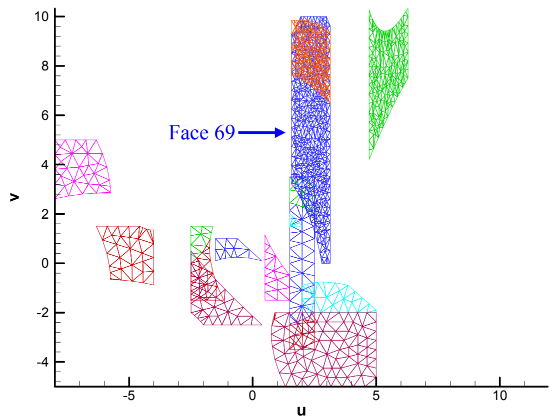


Figure 3 – Local parameterizations

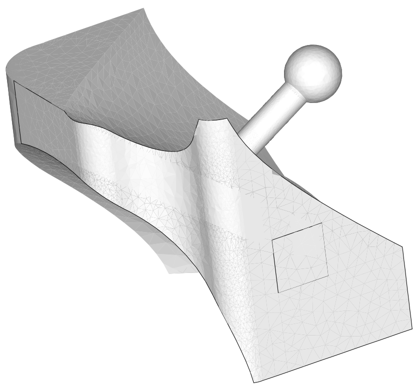


Figure 4 – Quilted topology

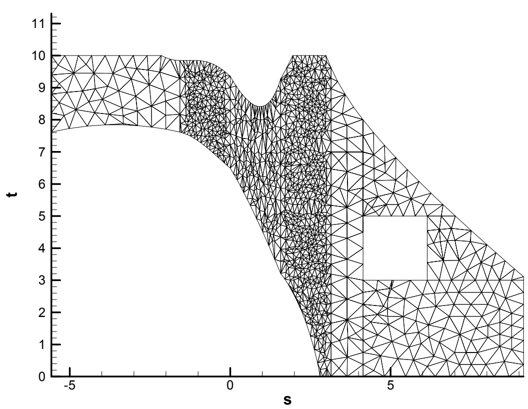


Figure 5 – Global parameterization for Quilt

contained the most nodes and was selected to define the basis for the global parameterization.

Figures 6 through 9 show a uniform unstructured surface mesh generated on the nose of the Space Shuttle Obiter. The mesh was generated with the FELISA curve and surface meshing algorithms. The model as delivered for analysis contained 192 topological faces for this portion of the vehicle.

Figures 6 and 7 show a surface mesh generated based on the raw topology extracted from the model. A uniform spacing constraint has been imposed. The bold lines represent the topological boundary of the individual faces. The mesh generated using the raw topology is implicitly and unnecessarily restricted by the minimum edge length of many of the faces. In many locations this length falls well below the uniform spacing value. Although a uniform spacing constraint was imposed on the mesh generation, the small edge lengths extracted from the topology produce numerous elements with aspect ratios much greater than unity. As a result element quality and the overall smoothness of this "uniform" mesh are reduced.

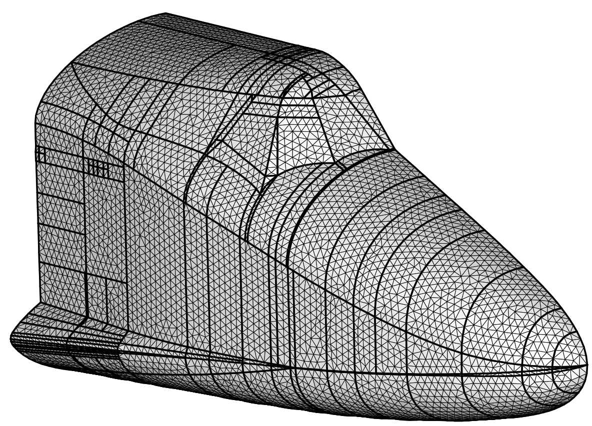
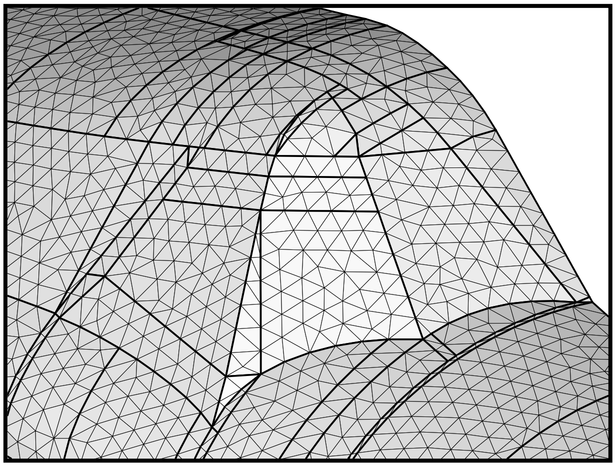


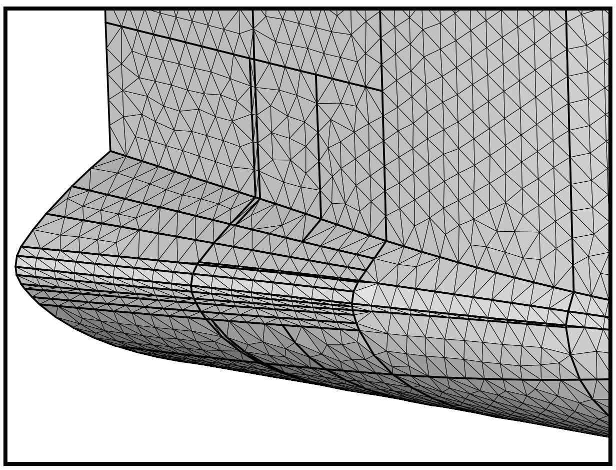
Figure 6 – Raw Topology Uniform Mesh

Numerous undesirable areas can be identified in these figures. Areas to note in figure 7a include: the crew windows; the small bands just ahead, as well as along the windows on the vehicle; and complex, often degenerate regions, on the top of the crew area. Figure 7b details the wing leading edge and the numerous sliver faces used in its definition. Finally Figure 7c represents the nose of the vehicle. Prominent in this region are the degenerate faces making up the nose cone and the small sliver that continues down the length of the body continuing onto the wing leading edge.

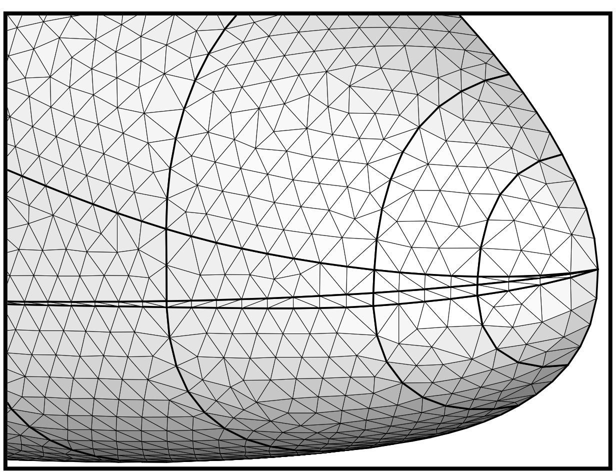
The basic quilts package is capable of reducing this topology to 5 quilts while maintaining adequate resolution of model discontinuities. The resulting quilts are: the symmetry plane; the exit; the majority of the orbiter; two small faces required to model protrusions



a) Crew Area



b) Wing Leading Edge



c) Nose

Figure 7 – Raw Topology Mesh Details

located over the payload bay door attachments; and a forward facing step behind the starboard window of the crew area.

Parameterization of the quilts forced the construction of 32 distinct quilts. This is due to the limitations of the current method described above. Faces were collected such that the limitations were circumvented and the results are sufficient to produce an improved uniform mesh over that of the extracted raw topology. The resulting uniform mesh was again generated with the FELISA algorithms and is shown in figures 8 and 9.

It is readily apparent from figure 8 that a significant improvement is provided by the construction and use of the parameterized quilts. These quilts are used to drive mesh generation with the same uniform spacing constraints.

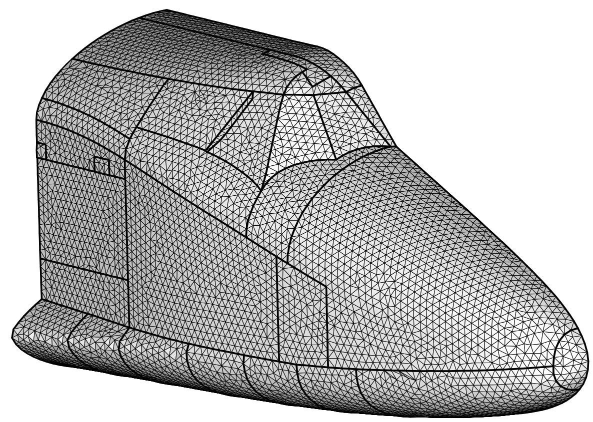
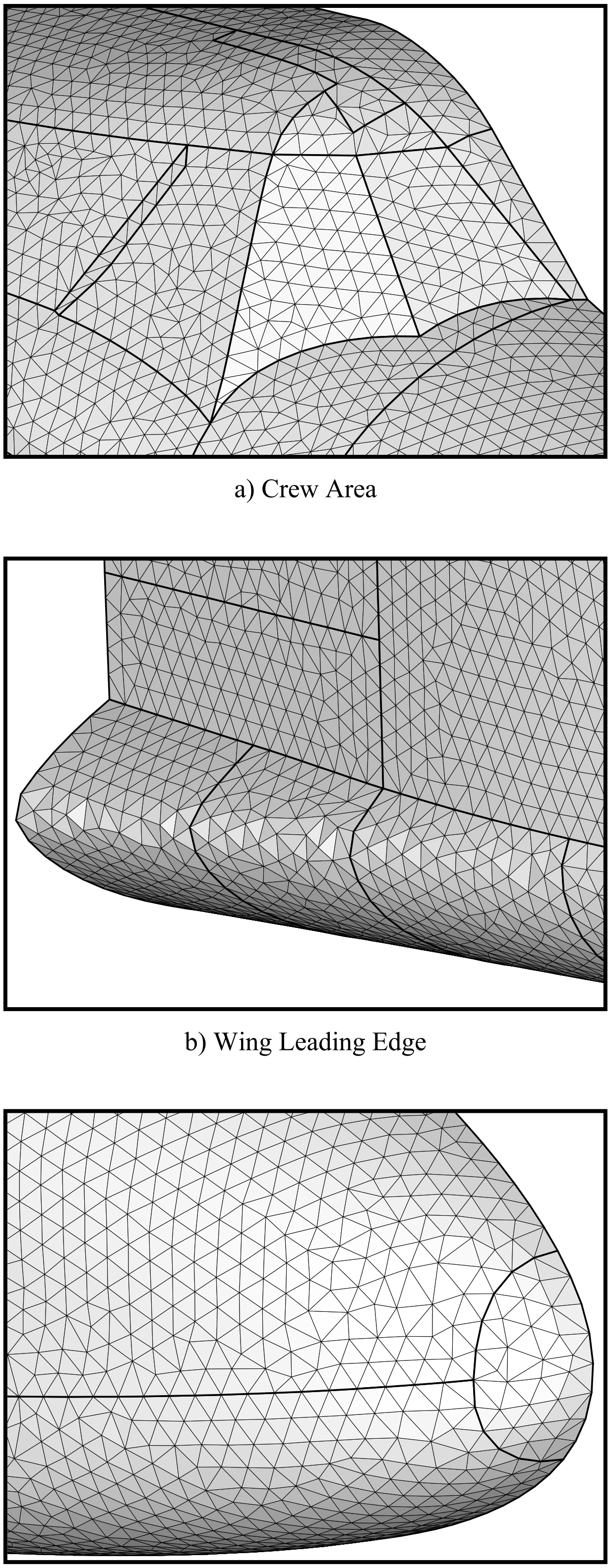


Figure 8 – Mesh on Parameterized Quilt

Figures 9 show the same details as those found in Figures 7. Significant improvement is represented by the use of the parameterized quilt to generate the mesh with a uniform spacing constraint. The abundant sliver faces observed in Figure 7a around the crew windows are now removed as evidenced by the corresponding Figure 9a. The exception is the small rectangular region aft of the starboard window which was intentionally left to resolve the fidelity of a forward facing step in the geometry. The high aspect ratio cells formerly in a band just ahead of, as well as emanating from, the crew windows have been removed as a result of using the parameterized quilt. The complexity of the orbiter above the crew windows has also been greatly reduced with the removal of many slivers and degeneracies. Recalling that a uniform spacing constraint was requested of the mesh generator, Figure 9b shows a great improvement in the mesh quality



a) Crew Area

b) Wing Leading Edge

c) Nose

Figure 9 – Parameterized Quilt Mesh Details

along the leading edge of the wing. The multitude of small high aspect ratio faces (translating to high aspect ratio elements in the mesh) of 7b are now suppressed resulting in a better distribution of points and element quality. Finally, the detail of the nose region provided in 9c reveals the improvement in mesh quality afforded by the modified topology defined by the quilts. Again, sliver and degenerate topology is suppressed resulting in a highly uniform mesh in agreement with the imposed spacing constraint.

CONCLUSION

A method for the calculation of a unified global parameterization for quilted CAD entities has been presented. Though the method is not without limitation, it has been shown to provide a significant improvement when used to alter the topology extracted to drive numerical mesh generation. The method utilizes an underlying tessellation of the respective CAD entities to define a contiguous parameterization that is suitable for navigation over the entity and also provides derivative information for use in geometry dependent analysis applications. No approximation of the supporting geometry is required. The resulting parameterization makes the use of the resulting quilts virtually transparent to an employing algorithm.

ACKNOWLEDGEMENTS

The author would like to thank Robert Haimes and Dr. J. F. Dannenhoffer for the extensive development of the CAPRI API and the Quilts enhancement which make this work possible. Assistance in the use of their respective packages, and their willing support of the same are of continuing help to the author. Appreciation is also conveyed to Dr. N. Alexandrov, W. L. Kleb, M. A. Park, and other members of the Langley Fast Adaptive AeroSpace Tools (FAAST) project team for their assistance and willingness to discuss this problem area.

REFERENCES

[1]Haimes, R., “Computational Analysis Programming Interface,” Proceedings of the 6th International Conference on Numerical Grid Generation in Computational Field Simulations, pp. 663-672, 1998.

[2]Haimes, R., “CAPRI: Computational Analysis PRogramming Interface User’s Guide”, Massachusetts Institute of Technology, 2001.

[3]Haimes, R., Aftomis, M., J., “On Generating High Quality Watertight Triangulations Directly from CAD,” Proceedings of the 8th International Conference on Numerical Grid Generation in Computational Field Simulations, 2002.

[4]Jones, W., T., "GridEx – An Integrated Grid Generation Package for CFD," Proceedings of the 16th AIAA Computational Fluid Dynamics Conference, AIAA Paper 2003-4129, American Institute of Aeronautics and Astronautics, Orlando, 2003.

[5]Dannenhoffer, J., F., Haimes, R., "Quilts: A Technique for Improving Boundary Representations for CFD," Proceedings of the 16th AIAA Computational Fluid Dynamics Conference, AIAA Paper 2003-4131, American Institute of Aeronautics and Astronautics, Orlando, 2003.

[6]Sander, P., V., Snyder, J., Gortler, S., J., Hoppe, H., "Texture Mapping Progressive Meshes", Proceedings of ACM SIGGRAPH 2001, Computer Graphics Proceedings, Annual Conference Series, pp. 409-416, August, 2001.

[7]Sorkine, O., Cohen-Or, D., Goldenthal, R., Lischinski, D., "Bounded-distortion Piecewise Mesh Parameterization", IEEE Visualization, October, 2002.

[8] Peiro, J., Peraire, J., and Morgan, K., "FELISA System Reference Manual and User's Guide, Volume 1," University of Wales Swansea Report, CR/821/94, 1994.