Flattening 3D Triangulations for Quality Surface Mesh Generation

Kirk Beatty and Nilanjan Mukherjee*

Meshing & Abstraction Group
Digital Simulation Solutions
SIEMENS
2000 Eastman Dr., Milford, Ohio 45244 USA
kirk.beatty@siemens.com, mukherjee.nilanjan@siemens.com

Abstract. A method of flattening 3D triangulations for use in surface meshing is presented. The flattening method supports multiple boundary loops and directly produces planar locations for the vertices of the triangulation. The general nonlinear least-square fit condition for the triangle vertices includes conformal (angle preserving) and authalic (area preserving) conditions as special cases. The method of Langrange multipliers is used to eliminate rotational and translation degrees of freedom and enforce periodic boundary conditions. Using matrix partitioning, several alternative sets of constraints can be efficiently tested to find which produces the best domain. A surface boundary term is introduced to improve domain quality and break the symmetry of indeterminate multi-loop problems. The nonlinear problems are solved using a scaled conformal result as the initial input. The resulting 2D domains are used to generate 3D surface meshes. Results indicate that best mesh quality is achieved with domains generated using an intermediate *altitude preserving* condition. Apart from an admirable robustness and overall efficiency, the 2D developed domains are particularly suited for structured transfinite/mapped meshes which often reveal wiggly irregularities with most conventional developed domains. Flattening and meshing (both free and transfinite/mapped) results are presented for several 3D triangulations.

Keywords: flattening, free-boundary, interpolation, Langrange multipliers, parameterization, periodic, transfinite, triangle.

1 Introduction

Although direct methods of generating meshes on 3D surfaces abound, a large majority of mesh generation programs used in both the industry and the research laboratories continue to use indirect method of generating the mesh on a flattened 2D parameter space first, transforming the mesh back to the original 3D surface later. Finite element analyses of structures, fluid flow and interaction problems have increased in complexity over the last few decades. The mesh to solve (especially quad-form meshes) often have pressing quality requirements. Crash analyses limit the variation of element size in anisotropic meshes. A large range of boundary-flow and structural problems require meshes to be boundary structured or partially or wholly transfinite.

* Corresponding author.

A general purpose mesh generation engine needs to be able to cater to most classes of problems. Such meshes of high quality and constraint are often virtually impossible to generate directly on real, complex 3D geometry efficiently.

Thus the problem of generating a low distortion 2D parameter space from the 3D surface continues to enthrall engineers and mathematicians. The goal is to create a 2D flattened domain of a crack-free triangulated (tessellated) 3D surface where the mesh could be generated. A discrete, continuous transformation function should be made available to finally transform the mesh nodes to the 3D surface. The same transform can be also used in 2D space to check the quality of the 3D elements being generated in 2D space.

1.1 Use of 2D Domains Created from 3D Triangulation

Two-dimensional domains created from 3D triangulations have a variety of applications. These include the fitting of parametric surfaces, the use and creation of texture maps, and as a parameterization to solve differential equations on 3D surfaces. In our case, the domains are used in the meshing of tessellated faces, which are based on a 3D triangulation. These faces can be derived by combining triangulations of several NURBS surfaces. The topology associated with these faces can include many boundary loops and/or repeated edges. To see how we use the 2D domains, we give a simplified version of our meshing process:

1.2 Simplified Meshing Method – Free or Transfinite Meshing

Before we start the meshing process, we abstract the tessellated faces. This geometry abstraction toolbox includes a host of basic discrete geometry operators which involve splitting and combining the facetted faces, cutting sharp corners that cause meshing issues. First mesh all edges at 3D locations. (For mapped meshes one must match the number of vertices on matched edges for the whole system.) For each surface:

1) Create 2D domain from 3D triangulation and topology. Split edge uses of same edge in domain (to create non-periodic surfaces from periodic) if needed for quality.
2) Transform nodes on 3D edges to 2D domain.
3) Mesh domain in 2D using boundary nodes.
4) Map 2D node positions to 3D using triangle coordinates.

In this paper we will focus on how the domains produced in step 1 affect our meshing result.

2 Past Research

There is an extensive literature for the parameterization of 3d triangulations [1]. Some techniques require a fixed convex boundary in UV space as an initial condition. Floater [2] has developed an improvements to this approach by considering harmonic function theory. Using his approach, the results vary smoothly with input data, are fast to solve, and guarantee no triangle flips in UV space. Though the technique is

powerful, the requirements of a non-convex boundary (and the filling in of holes) can often cause significant distortion. Instead we focus on free boundary methods.

One of the first free-boundary approaches was developed by Hormann and Greiner [3]. They developed the Most Isometric Parameterization (MIPS) based on minimizing a metric of distortion. They realized that directly using the Dirichlet energy did not punish the collapsing of triangles, or a degenerate solution. Instead they found a nonlinear criterion that was more difficult to solve. They had good results on small models, but the overall approach was computationally expensive. Levy et al. [4] considered a quasi-conformal approach (Least square conformal mapping -LSCM) in the context of automatic atlas generation. For smaller regions this method was quite fast and could produce good results. They found they could avoid degeneracy by fixing two points of the system in UV space. They chose two far apart in geodesic distance along the surface. Desbrun et al [5] developed a similar least squares conformal approach independently. They found they could linearly combine the results with area preserving (authalic) solution that was not formulated for a free boundary. They developed a general approach of constraints using Lagrange multipliers. Even though that constraint approach was quite general, in practice they used a more simplistic approach to avoiding degeneracy than Levy et al. They fixed two points farthest apart in 3d space. An angle based flattening (ABF) approach was found by Sheffer and de Sturler [6]. They considered equations for angles that were consistent with the plane. When they found a solution to their nonlinear problem with constraints, they could guarantee a no triangle flip answer. Sheffer et al. [7] developed a hierarchical approach for large systems and pointed out their approach produced significantly less stretch than the LSCM method. We suggest that in the comparison the LSCM procedure suffered from a simplistic pinning condition and a “collapsing triangle” problem evident with large triangulations. Recently Zayer et al [8] have found a fast linear approach that gives very similar results to the ABF method. After the angles of the triangles are determined by the ABF (or derivative) procedure, it is still necessary to find a solution in UV space. Here at Siemens, we have used a flattening approach based on preserving the lengths of triangle sides. Chief architects of the technology included Evan Sherbrooke and David Gossard (formerly of New Technologies Inc.) Similarly to the ABF method, constraint equations are developed such that the result is consistent with the restriction of the plane. The constraints were based on the law of cosines for the wheel of triangles around all interior vertices. Once the lengths of the triangles are found, they must be rolled out into the 2D plane. (The work in this paper originated from improvements to the robustness of the roll-out procedure used in NTI’s method.) A global parameterization based on gradient fields was developed by Gu and Yau [9]. In their method they found the underlying basis vectors of the linear space. The bounding conditions were based on periodic boundary conditions on cuts of closed surfaces or a double covering of the non-closed surface. By tying together the boundaries of the surface together, and avoiding pinning of points, very uniform conformal solutions were found.

3 Problem Statement

Our meshing method requires a procedure to create a 2D domain for meshing from a 3d triangulation of a face. Associated with the facetted face there is a topology which

includes loops and edges. This topology is a important component of our meshing process, which in turn puts requirements on our flattening procedure. The facetted faces consist of triangles defined by facet vertices. The facetted faces to be flattened have a loop and edge topology. The edges are defined on edges of facets. The edges can be both on the exterior and interior of the 3d face. Edges on the interior of a face have two uses by the loops, which are in different orientations. Edges interior to a face can be used to make non-manifold connections between different faces or enforce meshing along an edge. They can also be added to help the creation of a low distortion domain in two dimensions. A cylinder can be represented with one loop of edges with a repeated edge use. If the repeated edge is allowed to split apart in a UV domain, then we can get a rectangle where the vertices along the repeated edge have different uses with different UV values. An edge so used is commonly referred to as a "seam". Originally, we distinguished between interior edges that were from cutting a cylinder, and edges that were added for other purposes. However, there were two problems with this approach. The first is that an edge could have different uses on different faces. (This problem could be rectified by considering edge uses.) The second problem was that a small modification of the topology of a surface could require reevaluating the status of all edges of the surface. These considerations lead us to the following requirements for our flattening process.

i) Produces a 2D domain with minimal global and local distortion. Ability to trade-off shape distortion, dimension distortion, and performance depending on application.
ii) Supports multiple loops and non-convex boundaries
iii) Automatically finds constraints between the repeated edges of a face's topology that lower distortion in the 2d domain.
iv) Overall procedure must be fast and robust.

4 Weighted Edge Flattening Method (WEFM)

The WEFM will be presented as both a local and a global solution. The local solution is a simpler framework, but can become unstable with large systems of badly shaped triangles. It also requires explicit constraints. A global solution that overcomes these problems is presented.

4.1 Rolling Out a 3D Triangulation into 2D Using Iteration

Consider creating a 2D domain from a 3D triangulation surface with the five triangles shown in Figure 1. All of the triangles of the surface are equilateral, and four of them are from the top of a square based pyramid. An additional (equilateral) triangle is added with one side attached to one side of the square at the base of the pyramid. In the case of this example, it is impossible to put the vertices into 2-d without distorting the lengths of the edges and the angles of the triangles. Notice the angles of the four triangles the meet the vertex of the pyramid are 60 degrees. However, as this point is interior to the surface the total angle must be 360 degrees. But only triangles with all angles the same have all sides of the same length, so the sides can not be all the same length in 2D. If we put a cut at the thick solid line in Figure 1a, and let these sides

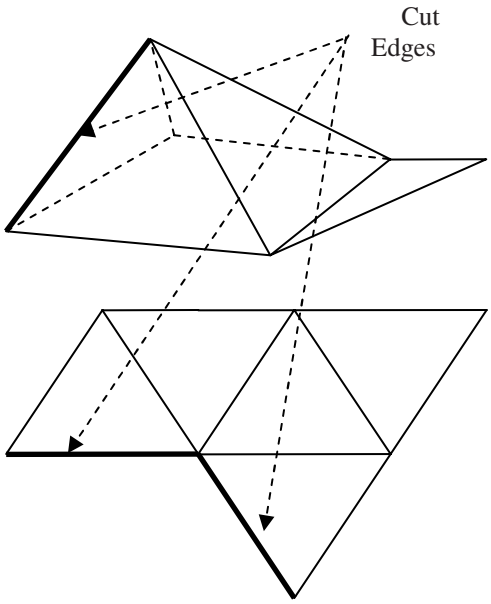

Fig. 1a. Simple 3D Triangulation

Fig. 1b. Flattened 2D domain with cut (5 equilateral triangles)

separate in the UV plane, then it is possible to find a UV domain with no area/length distortion except measurements that involve the cut. The result is given in Figure 1b.

When rolling the triangulation out to the plane requires deformation, averages/compromises are used for determining the positioning of vertices. The first step of the iterative procedure is to lay a long perimeter edge of the triangulation along the x-axis. The edge is given the same length that it has in 3D. For the given figure, the edge is placed from vertex 1 to vertex 2 on the x – axis. Then the vertex is placed opposite to the edge in the domain to keep the same shape and orientation. This process is continued, picking a vertex that neighbors the edges placed in 2D. In Fig. 2a, this is done with vertices 3 and 4. When vertex 5 comes, it is found that it is neighboring two different triangles, and the coordinates found for the vertex are different. The vertex is placed at weighted average of these results. Each triangle is given a weight which is determined by the length of the opposite side divided by the altitude to the vertex we are placing in 2D. Up until vertex 5 is placed in 2D, all of the edges have their original length. This makes it possible for each triangle to have a position for a neighboring vertex that does not change any lengths or angles. Once the results are averaged, vertex 5 is no longer at each desired position, and distortion results.

When we come to placing vertex 6 in the plane we have a new problem. The length from vertex 4 to vertex 5 in the plane is different then they are in 3D. Vertex 6 could be placed such that:

1) The triangle in 2D has the same angles as in 3D.
2) The altitude to the vertex has the same length in 2D and 3D.
3) The triangle in 2D has the same (absolute) area as in 3D.

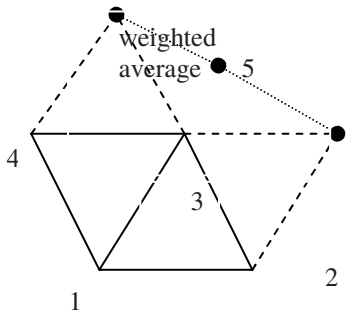

Fig. 2a. Simple triangulation roll-out where vertex 5 is weighted average from triangles 4-3-5 and 3-2-5

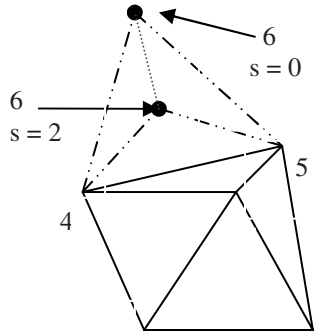

Fig. 2b. Simple triangulation roll-out where vertex 6 placement can vary from preserving angles to preserving area

To calculate the location of vertex 6 for these different cases a notation based on half-edges is used. (Choices 2 and 3 will be made unique with further restrictions.) A half-edge is associated with one and only one side of a triangle. There are one half-edge for each exterior edge of the triangulation, and two half-edges for each interior edge of the triangulation. A special notation is used to indicate the vertex indices associated with an half-edge with index *h*. The vertex indices along a half-edge with index *h* are designated in the counter-clockwise order of the triangle by *I(h,1)* and *I(h,2).* The vertex opposite the half-edge with index h is designated by *I(h,3).* Consider this notation applied to Fig. 2b for the half-edge the goes from vertex 4 to vertex 5. This half-edge is in triangle 4-5-6, since is assumed the triangles are numbered consistently and counter-clockwise. If the half-edge from vertex 4 to vertex 5 has an index of 13, then *I(13,1) =4, I(13,2)=5*, and *I(13,3) = 6*.

Associated with each half-edge are two parameters that give the shape of the triangle. The first gives how far along the half-edge is the opposite vertex. Using the dot product to find the parameter (α_h) along the half-edge:

$$\alpha_h = \frac{\left(v_{I(h,2)} - v_{I(h,1)}\right)\cdot\left(v_{I(h,3)} - v_{I(h,1)}\right)}{\left[L_h\right]^2} \tag{1}$$

where L_h= length of the half edge with index *h*. The second parameter (β_h) is a ratio of the altitude to the opposite vertex to the length of the half-edge with index *h*. Using the cross product for the parameter perpendicular to the half-edge:

$$\beta_h = \left|\frac{\left(v_{I(h,2)} - v_{I(h,1)}\right)\times\left(v_{I(i,3)} - v_{I(h,1)}\right)}{\left[L_h\right]^2}\right| f\left(I\left(h,3\right)\right) \tag{2}$$

where *f(I(h,3))* is an adjustment for Gaussian curvature around the opposite vertex. For a vertex on the boundary it is set to 1.0. For an interior vertex it is the sum of the angles around the vertex divided by 2π. Also associated with an half-edge is the ratio

(ρ_h) of the geodesic length of the half-edge with index h (which we take to be the 3D length) to the length of the edge in the UV domain. That is

$$\rho_h = \frac{L_h}{\left| z_{I(h,2)} - z_{I(h,1)} \right|} \tag{3}$$

where the each z refers to the UV position of the vertex with the same index. The formula to calculate a new UV location of a vertex from anopposite half-edge is given by eqn.(4) where the UV coordinates have been

$$z_{I(h,3)} = z_{I(h,1)} + \alpha_h (z_{I(h,2))} - z_{(I(h,1))}) + \beta_h (z_{I(h,2)} - z_{I(h,1)})(\rho_h)^s i \tag{4}$$

represented as complex numbers, i is the imaginary unit vector, and s is a global parameter which gives different types of "roll outs". Choices 1,2, and 3 correspond to s values of 0, 1, and 2 respectively. These choices are referred to as conformal, altitude preserving, and authalic. Notice for the s =0 solution the value of ρ does not matter. When one vertex is across from many half-edges, then the weight for each half-edge is proportional to the reciprocal of its associated β. The weights are normalized to sum to 1.0. Taking half-edge 13 which goes from vertex 4 to vertex 5, and using an s value of 1.0 the equation (4) becomes:

$$z_6 = z_4 + \alpha_{13}(z_5 - z_4) + \beta_{13}(z_5 - z_4)(\rho_{13})i \tag{5}$$

The order in which points are selected for putting into 2d affects the solution that is found. Ordering the vertices by the ratio of the 3d lengths of the edges which are in 2d that ring a vertex to the perimeter of the ring in 3d provides a good heuristic for sorting the list of possible vertices. Once all the points are in the UV plane, we can continue to iterate through all vertices updating their position. Choice (1) can lead to degenerate solutions, and choice (3) is often unstable.Though choice (2) often works well for domain generation, it is not always stable and can produce domains that are of poor quality. It is also order dependent; heuristic choices of which vertex to calculate next can dramatically affect the result. Though it is straightforward to constrain points of the iterative solution, creating periodic boundary conditions is more problematic.

4.2 A Global Method for Triangle Roll-Out

To overcome the limitations of the iterative scheme, we move to a global solution using a least squares fit. The solution is directly in terms of planar coordinates, and the constraints can be any linear combination of vertex positions. The function to be minimized consists of four terms. The first term is a surface energy term based on the same formulation as the iterative method. The second term uses Lagrange multipliers to take out the translation degree of freedom by constraining a weighted average of the UV points to be at the origin. The third term is a more general constraint term used to enforce boundary conditions and eliminate a rotational degree for freedom. The rotational degree of freedom can be eliminated by enforcing the difference between the UV position of two vertex uses to be a vector. (At repeated edges the

vertices can be split in the domain) The last term is based on angles in UV space between edges on the boundary loops. The angles are for a counterclockwise direction in the designated outer loop, and for a clockwise direction in all other loops. This convention breaks the symmetry of the problem for uncut cylinders. The upper limit n is for the number of vertex uses of the model. The upper limit m is for the number of constraints in the model. The upper limit d is for the loops of the model. l_i is the number of facet segments for loop i. The four term function to be minimized for the vector of unknown UV positions:

$$F(Z)=\sum_{i=1}^{n} q_i \overline{q_i} + \lambda_1 \sum_{i=1}^{n} \eta_i z_i + \sum_{k=2}^{m} \lambda_k \left(\left[\sum_{i=1}^{n} g_i^k z_i \right] - c_k \right) + \sum_{i=1}^{d} \mu_d \sum_{j=1}^{l_i} \frac{p_{i,j}\overline{p_{i,j}}}{\left| l_{i,j} \right|} \quad (6)$$

qi of the first term is a 2d vector (or complex number). Multiplying q_i by its complex conjugate gives a distance squared. The vector q_i starts at a weighted average of UV positions determined by all half-edges opposite the vertex use. (There is one such half-edge for each triangle that contains vertex use i.) It ends at the UV position of the vertex use times the sum of the weights of all half-edges for which it is an opposite.

$$q_i = \sum_{\substack{h \in H \\ \forall I(h,3)=i}} w_h \left[z_i - \left[z_{I(h,1)} + \sigma_h \left(z_{I(h,2)} - z_{I(h,1)} \right) \right] \right] \quad (7)$$

The index notation for vertices based on half-edges, $I(h,1)$ etc., was described in the previous section. The scaling-rotation vector, which depends on the type of "roll out, is:

$$\sigma_h = \alpha_h + \beta_h (\rho_h)^s \mathrm{i}. \quad (8)$$

In the special case s is set to zero, a value for ρ is not needed. The weight of each half-edge is given by

$$w_h = \frac{1}{\beta_h} \left(\sum_{\substack{h' \in H \\ \forall I(h',3)=I(h,3)}} \frac{1}{\beta_h'} \right)^{-k} \quad (9)$$

where β_h is defined in eqn. 2. The sum is over the set of all half-edge indices that have the same opposite vertex use. Setting k to 1 normalizes the sum of the weights around a vertex use to 1. Improved results are often found when k is set to ½. (In the iterative scheme k is always set to 1.) For the second term of eqn. (6), η_i is the area of all 3D triangles connected to vertex $\boldsymbol{i}$. For the second and third terms the λ_k are Lagrange multipliers. For the third term, g_i^k is a weight for a point of the constraint, and c_k is a fixed vector [5]. In our current usage, g_i^k is usually -1, 0, or 1 and c_k is zero. But these constants can be any complex value. The last term of the global equation

involves loop segments of different loops. $p_{i,j}$ is a vector relation for loop *i* segment *j*. The term preserves boundary angle and length ratios:

$$p_{i,j} = \left(z_{i,j-1} - z_{i,j}\right) \cdot r_{l,-1} \cdot e^{i\varphi_{i,j}} + (z_{i,j+1} - z_{i,j}) \cdot r_{i,1} \cdot e^{-i\varphi_{i,j}} \quad (10)$$

φ is half the goal angle between two neighboring segments of a loop. $r_{l,+-1}$ is the ratio of segment length before/after the vertex to their average. In the last term, $l_{i,j}$ is the average length of two consecutive segments of the loop. μ_d is a weighting factor for the boundary. $z_{i,j}$ is the UV position for the point on loop i at the end of segment j. Solve the system by setting the partial derivatives for the points and Lagrange multipliers to zero.

4.3 General Solution Process

The solution of the least squares minimization problem is found by solving the equations resulting from setting the partials for the points and the Lagrange Multipliers to zero. Approaching the general nonlinear solution is done in three steps:

1) Solving the (linear) conformal system without repeated edge constraints. Return if the solution is of sufficient quality.
2) Compare solutions for repeated edges to find if each repeated edge should be periodic, free, or together (scar). Again, can return with solution if domain is of sufficient quality.
3) Repeat solve with increasing *s* value using the constraint decision found in step 2. Iterate to an s=1.0 solution. Use the boundary energy term in the nonlinear solution. When s is not 0.0, a value of ρ_h determined from equation 3 must be used in equation 8. The value is determined from the solution at the previous s.

During the solution process we often make use of a score to assess the quality of the domain. The score is from zero to one based on a weighted average of the square of the ratio of areas for the triangles in UV and 3d space. (The reciprocal of the ratio is taken if it is above 1.0). One is subtracted from the total weighted score for all triangles if there is any triangle flip. Two is subtracted for boundary intersection. (A flip of a triangle would result in a mapping such that a position in the UV domain would not correspond to one point in the 3-d triangulation. This could lead to an unusable mesh.).

To eliminate these degrees of freedom for step 1 of the process, Levy fixed two points that were far away in a 3D geodesic sense. This requires finding geodesic distances in 3D with holes. Instead of fixing two points *apriori*, we do most of the work of solving the linear system, and add the constraints later. To do this extensive use was made of the method of Lagrange Multipliers and matrix partitioning. To eliminate the translational degree of freedom, a weighted average of the vertex UV positions was constrained to be at the origin. To eliminate a rotational degree of freedom, we fix the vector between two points that are on the boundary. Once this solution is found, we can find two points on the boundary that are far apart in the UV domain. We eliminate the constraint fixing the facet edge, and add the constraint for the two points that are far apart in UV. We reuse the LU decomposition of the part of our

matrix that involves points, so the second solution is found very quickly. One example where the two-step approach works well, while the methods of Levy and Desbrun do not, is in the flattening of the triangulation of a sphere with a small hole. A two level matrix partitioning procedure significantly speeded up step 2 of the process.

5 Surface Mesh Generation

Once a 2D triangulated domain is generated, the boundary of the domain is discretized with nodes (driven by a global element size) and a variety of 2D mesh generation algorithms are used to fill the area. The algorithms used can be broadly classified into three categories, namely

a. Unstructured free mesh generation using subdivision technique [10,11]
b. Unstructured free mesh generation using a combination of subdivision and advancing loop-front methods [12]
c. Structured mesh generation using boundary-blended transfinite interpolation techniques [13,14]

5.1 Mesh Quality Measure

For free, unstructured meshes, mesh distortion can be measured by an element quality metric, λ, where

$$\lambda_{mesh} = (\Pi_{i=0}^{n} \lambda_i)^{1/n} \quad \text{denotes the quality of the mesh} \qquad (11)$$

λ_i (mesh quality of the i-th triangle) where λ_i is as proposed by Lo[15].
All elements need to pass a critical threshold defined by $\lambda_{cr} = 0.6$.

Both λ_{mesh} and λ_i of the worst element in the mesh can be compared for different domains to decide which is better.

For transfinite meshes however, element or mesh quality, as determined by λ_{mesh} or λ_i is not necessarily a good measure. Bad domains can cause the transfinite nature (structured) of the mesh to be destroyed even for the same number of nodes/elements and element connectivity. The rail lines of the mesh can be wiggly, or unparallel – all of these could destroy the properties of a transfinite mesh. To measure the wiggle of the rail lines a "wiggle factor" is used. The wiggle factor, ε_{fi} for the i-th rail line is an error norm computed for each interior mesh node on the line summed over the length of the rail line in j direction made up of n nodes.

$$\varepsilon_{fi} = 1/n \sum_{j=0}^{n} (L_{ij} - L_{ij-1})^2 / \bar{L}_i^2 \qquad (12)$$

Lij denotes the length vector at node ij (jth node on the i-th rail line). L_i denotes the average length vector of all nodes on the i-th rail line.

6 Examples and Discussion

Three different examples will be discussed to show the benefit of using an efficient flattening or domain development technique for high quality surface meshing. Certain

specific aspects of mesh quality or pattern will be discussed in this connection. In all of the results presented a k value of 1 was used in eqn. (9), though it has subsequently found a k value of ½ slightly improved the results for the boundary structured free mesh.

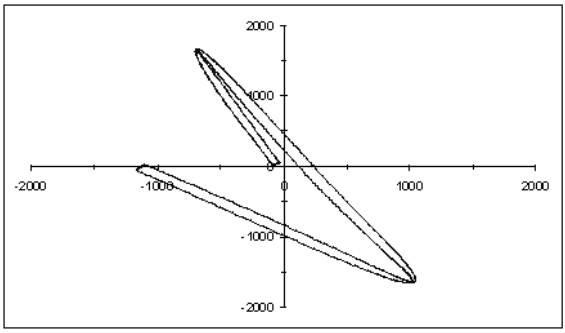

Fig. 3a. 2D domain of a helical surface developed by WFEM at s=0 (conformal) with no boundary constraint

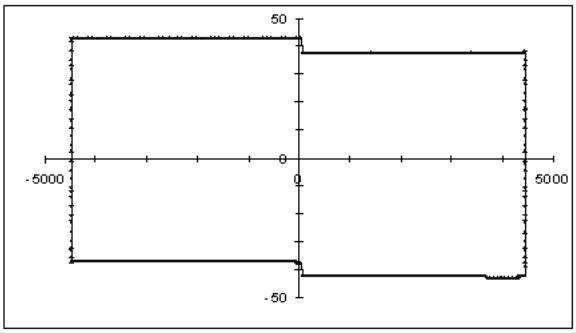

Fig. 3b. 2D domain of the same helical surface by WFEM at s=1 with boundary constraint

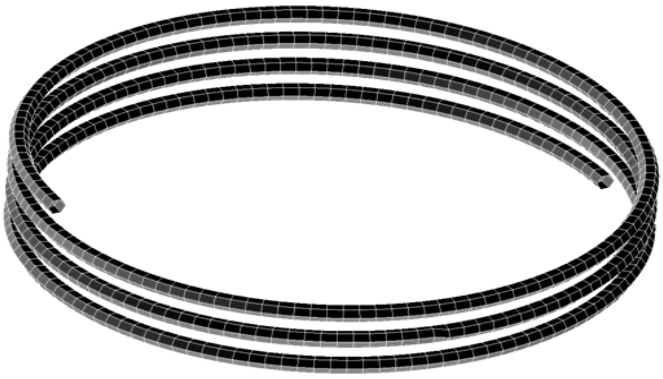

Fig. 3c. 3D final transfinite mesh on the helical surface

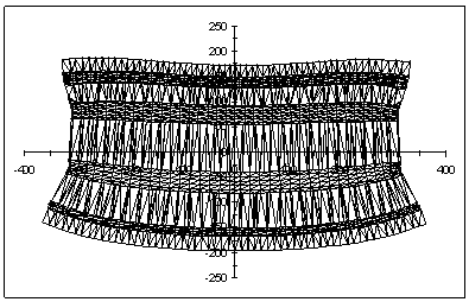

Fig. 4a. 2D domain of a pressure vessel wall surface by WEFM at s = 0 (conformal)

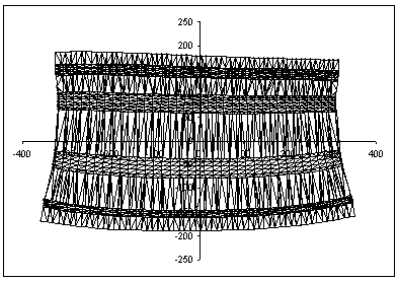

Fig. 5a. 2D domain of a pressure vessel wall surface by WEFM at s = 0.3 with boundary energy term

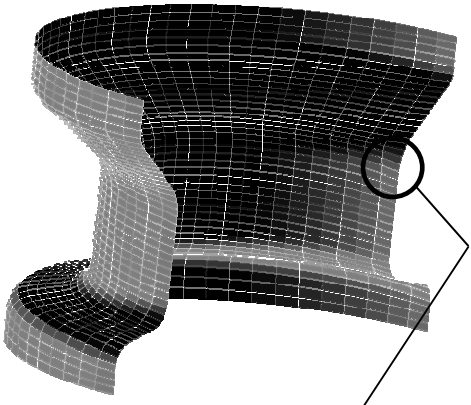

Fig. 4b. Corresponding final 3D transfinite mesh with wiggly ends

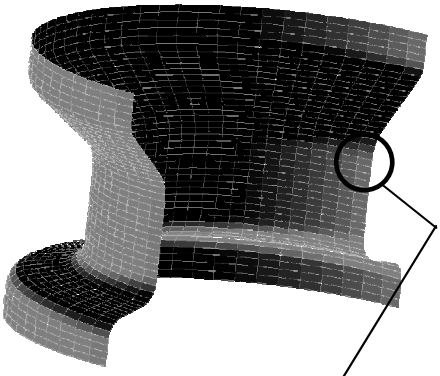

Fig. 5b. Corresponding final 3D transfinite mesh

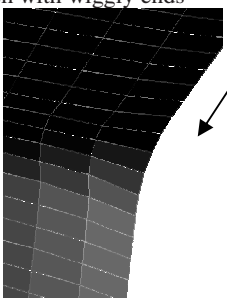

Fig. 4c. Rail lines at near face boundary showing wiggles

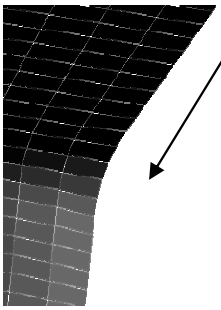

Fig. 5c. Smooth rails

6.1 Helical/Spiral Surface Meshing

Meshing is often challenged by merged tessellated surfaces that are helically/spirally coiled. Such surfaces abound in threaded mechanical enginering parts and are often encountered in finite element analysis. Fig. 3a shows a typical rolled out 2D domain

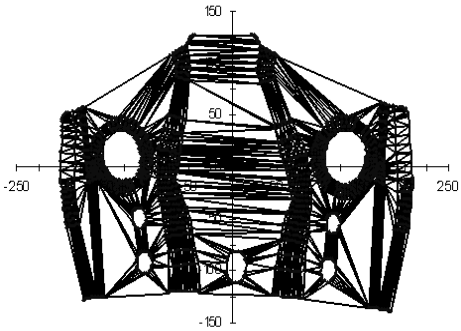

Fig. 6a. 2D domain of an automobile frame by WEFM at s=0.8 with boundary energy term

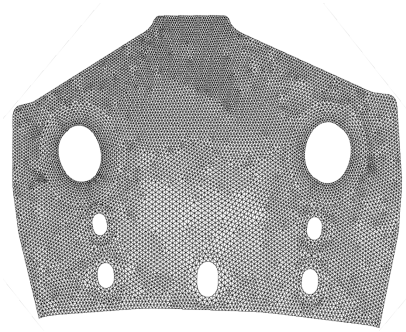

Fig. 6b. 2D triangular CSALF mesh on the 2D domain

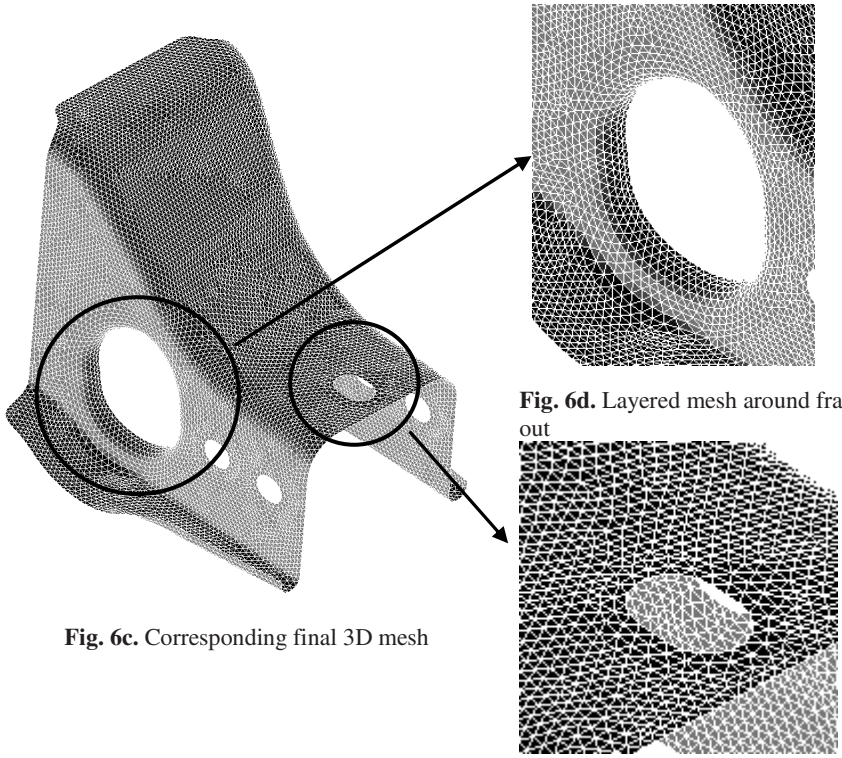

Fig. 6c. Corresponding final 3D mesh

Fig. 6d. Layered mesh around frame cut-out

Fig. 6e. Boundary structured mesh around holes

which has appreciable distortion but is not suitable for map meshing. Fig. 3b presents a continuous section of a 2D domain developed by WFEM with s=1. This domain is regular and can be meshed by transfinite interpolation (mapped meshing) which is

depicted in Fig. 3c. Note, the domain has a small kink resulting from an irregular cut line chosen along the facet edges. The 3D mapped mesh reflects a small local distortion at that location.

6.2 Accurate Transfinite Meshing Needs

Transfinite or mapped meshes are structured and mathematically accurate. The Coon's surface is not easily available on discrete geometry. Thus the efficiency and the accuracy of the flattening strategy is important for generating such meshes. Fig. 4a shows the 2D domain of a pressure vessel wall surface developed by WFEM (s=0.0) with no boundary energy term. The corresponding 3D transfinite mesh is shown in Fig. 4b. When we zoom in on the elements near the vertical edge, as is evident in Fig. 4c, we notice a lateral waviness in a certain region. This region corresponds to the two longitudinal kinks that exist on the vertical edges of the 2D domain. The wiggle factor of the transfinite distortion parameter for this mesh is 0.12 which is far above the limiting value of 0.01. With a domain developed by WFEM (s = 0.3) with a boundary energy term as pictured by Fig. 5a, these irregularities vanish and we get a perfect 3D transfinite mesh as shown in Fig. 5b-5c. The wiggle factor, described by eqn. (12) is 0.007.

6.3 Boundary-Structured Free Meshing on Complex Geometry

Good quality boundary-structured free meshes are important for a host of structural and fluid flow analysis problems. Generation of such surface and volume meshes on complex industrial geometry is key to the success of an engineering design process. Fig. 6a shows the 2D domain of such a frame structure developed by WFEM (s=0.8). A triangular CSALF mesh is shown on the 2D domain in Fig. 6b. These zones are of importance to the designer as they are usually susceptible to high stresses. The final 3D mesh shown in Fig. 6c reveals the same. It is important to note that the mesh, although unstructured in a global sense, is extremely layered/structured around all the inner circular/oval boundaries (see insets in Fig. 6d -6e).

7 Conclusion

It was found the conformal (s=0) solution for the WEFM can be used for meshing processes based on a 2d domain. However, as in all conformal based solutions, the dimensions for non-developable triangulations are changed when they are mapped to the UV domain. This can be a problem when high quality meshes are required. It was found an intermediate "altitude preserving" condition (s=1) of the WEFM, which compromises shape and dimension distortion, could give superior results when used with the presented map meshing process, and with forms of "mesh paving". It was found that automatically choosing from all the valid solutions gave us the best overall strategy. An automatically procedure for creating and evaluating constraints for repeated edges was developed. It was used to determine if each repeated edge should be left free, treated as a scar, or be part of a periodic boundary condition. A boundary energy term in the WEFM formulation enabled the solution of indeterminate multi-loop problems, and generally improved the quality of the resulting domain. The difference in quality was readily apparent in the resulting 3D mapped mesh produced from the different domains.

References

1. Floater, M.S., Hormann, K.: Surface parameterization: A tutorial and survey. In: Dodgson, M.S.E.N., Sabin, M. (eds.) Advances on Multiresolution in Geometric Modelling, Springer, Heidelberg (2004)
2. Floater, M.S.: Mean value coordinates. Computer Aided Geometric Design 14(3), 231–250 (1997)
3. Hormann, K., Greiner, G.: MIPS: An Efficient Global Parameterization Method. In: Laurent, J.-P., Sablonniere, P., Schumaker, L.L. (eds.) Proceedings, Curve and Surface Design, pp. 153–162. Vanderbilt University Press (2000)
4. Levy, B., Petitjean, S., Ray, N., Maillot, J.: Least Squares Conformal Maps for Automatic Texture Atlas Generation. In: Proceedings, ACM, SIGGRAPH (July 2002)
5. Desbrun, M., Meyer, M., Alliez, P.: Intrinsic Parameterization of Surface Meshes. Computer Graphics Forum (EUROGRAPHICS 2002) 21(3), 209–218 (2002)
6. Sheffer, A., de Sturler, E.: Parameterization of faceted surfaces for meshing using angle based flattening. Engineering with Computers 17, 326–337 (2001)
7. Sheffer, A., Levy, B., Mogilnitsky, M., Bogomyakov, A.: ABF++: Fast and Robust Angle Based Flattening. Proceedings, ACM Transactions on Graphics 24(2), 311–330 (2005)
8. Zayer, R., Levy, B. Seidel, H.-P.: Linear Angle Based Parameterization. In: Proceedings, Eurographics Symposium on Geometry Processing (2007)
9. Gu, X., Yau, S.-T.: Global Conformal Surface Parameterization. In: Proceedings, Eurographics Symposium on Geometry Processing, pp. 127–137 (2003)
10. Cabello, J.: Towards Quality Surface Meshing. In: Proceedings, 12th International Meshing Roundtable, pp. 201–213 (2003)
11. Schoof, A.J.G., Th, L.H., Van Beukering, M., Sluiter, M.L.C.: A general purpose two-dimensional mesh generator. Advances in Engineering Software 1(3), 131–136 (1979)
12. Mukherjee, N.: CSALF: A combined subdivision and advancing loop-front approach to generating controlled boundary structured free meshes. International Journal of CAD/CAM (submitted, 2007)
13. Perronnet, A.: Triangle, tetrahedron, pentahedron transfinite interpolations. Application to the generation of C0 or G1-continuous algebraic meshes. In: Proceedings. Internationa Conference of Numerical Grid Generation in Computational Field Simulations, Greenwich, England, July 6-9, 1998 vol. 7, pp. 467–476 (1998)
14. Mukherjee, N.: High Quality Bi-Linear Transfinite Meshing with Interior Point Constraints. In: Pebay, P.P. (ed.) Proceedings, 15th International Meshing Roundtable, pp. 309–323. Springer, Heidelberg (2006)
15. Lo, S.H.: A New Mesh Generation Scheme For Arbitrary Planar Domains. International Journal For Numerical Methods in Engineering (21), 1403–1426 (1985)