OPTIMIZATION OF SURFACE MESH QUALITY USING LOCAL PARAMETRIZATION

Rao V. Garimella[1] Mikhail J. Shashkov[2] Patrick M. Knupp[3]

[1] *MS B284, T-7, Los Alamos National Lab., Los Alamos, NM USA,* `rao@lanl.gov`
[2] *MS B284, T-7, Los Alamos National Lab., Los Alamos, NM USA,* `shashkov@lanl.gov`
[3] *Parallel Computing Sciences Dept., Sandia National Labs., Albuquerque, NM, USA,* `pknupp@sandia.gov`

ABSTRACT

Improvement of the quality of surface meshes is important for mesh generation and numerical simulation. The challenge with surface mesh improvement is to improve element quality while preserving the surface characteristics as much as possible. A procedure is presented here to optimize the quality of elements in surface meshes by node repositioning while keeping the nodes on the original mesh faces and close to their original locations. The nodes are repositioned in a series of local parametric spaces derived from individual mesh elements rather than a global parametric space constructed from the complete mesh. The local parametric spaces are derived from barycentric mapping of triangles and isoparametric mapping of quadrilaterals. The procedure has been tested successfully on a number of complex triangular and quadrilateral meshes. Quantitative measures are presented to prove that the mesh quality is improved and the deviation of the optimized mesh from the original mesh is small.

Keywords:mesh generation, surface meshes, quality improvement, condition number, triangles, quadrilaterals

1. INTRODUCTION

Improvement of mesh quality is a very important problem for mesh generation and numerical simulation. The quality of a surface mesh heavily influences the ability of mesh generation algorithms to generate good quality solid meshes. Since surface meshes define the boundaries of computational domains where boundary conditions are imposed, they also influence the accuracy of numerical simulations.

One method to improve the quality of a mesh is to reposition its nodes such that the shape of its elements is improved. In this case the topology of the mesh is not altered, a desirable feature when properties defined over the original mesh must be transferred to the new mesh. Other methods to improve meshes include edge swapping, vertex insertion, vertex deletion and local retriangulation [1, 2]. However, these methods are primarily applicable for simplicial (triangular,tetrahedral) meshes. Also, since these methods modify mesh topology, they may not be suitable for all applications.

When improving the quality of a surface mesh, an important consideration is to minimize changes in the discrete surface characteristics like discrete normals and curvature. Preservation of such characteristics is critical for large deformation of free boundaries in metal forming and interfaces in multi-material gas dynamics. Sometimes, in such applications, a severely deformed part of the mesh may have to be improved and the solution transferred from the old mesh to the new before resuming the simulation. In such a case, it is clearly desirable that the new mesh be of improved quality and yet remain close to old mesh. Preserving the surface characteristics also prevents drastic changes in forces like surface tension that depend on surface characteristic.

When repositioning nodes of a surface mesh, changes in the surface properties can usually kept small by constraining the nodes to the smooth surface underlying the mesh or to the discrete surface described by the faces of the original mesh. A common approach to constrain the nodes to the smooth or discrete surface is to reposition them in a 2D parametrization of the surface. When nodes that are repositioned in the parametric space of the surface are mapped back to 3D, they lie on the original smooth or discrete surface.

If the mesh has an underlying smooth surface, the parametric space of the surface is usually available from the geometric modeler that provided the surface definition. However, if such a surface is not available, then one approach is to create a global 2D parametric space from the discrete surface. Several researchers have developed techniques to build such global parametric spaces for triangular meshes [3, 4, 5]. However, all these methods involve substantial computational cost since they often require solution of a system of nonlinear equations. Also, they cannot be used directly to parametrize closed surfaces. Instead, the closed surfaces must be cut into one or more pieces which are then parametrized separately.

In this paper, a procedure is presented to improve the quality of a mesh by repositioning its nodes such that they are constrained to remain on the surface. The nodes are repositioned in a series of local parametric spaces derived from individual mesh elements (faces,edges) rather than a global parametric space derived from the complete mesh. The local parametric spaces are constructed either by barycentric mapping of triangles and by isoparametric mapping of quadrilaterals. The repositioning procedure keeps track of the original mesh element that each node is moving in and if a node moves out of the parametric space of the element, the procedure switches to the parametric space of an adjacent element. When the repositioned nodes are mapped back to the real space, each node lies on the mesh element whose local parametric space it is in. The method imposes no restrictions on the nature of the discrete surface or how far nodes may move on it. The procedure has been implemented for surface meshes containing triangles and quadrilaterals. Using a recent publication on the barycentric mapping of general polygons, it is expected that the procedure can be extended to handle general mesh elements easily [6].

Improvement of the surface mesh quality is achieved by a two-step optimization process that seeks to improve the shape of the elements but keep the modified mesh close to the original mesh. A similar procedure for optimization of planar meshes has been described earlier by Shashkov and Knupp [7]. The shape measure used to evaluate the quality of elements is based on the Jacobian matrix condition number of elements presented by Knupp [8]. The procedure has been tested on a number of complex triangular and quadrilateral surface meshes. Several quantitative measures have been incorporated into the procedure to evaluate the deviation of the modified mesh from the original and the preservation of surface characteristics like surface normals.

The rest of the paper is organized as follows. Section 2. below describes the method of optimizing a function with respect local parametric coordinates. The section describes the element based local parametrization, line search with respect to local parametric coordinates and moving nodes from one parametric space to another. Section 3. describes two methods for improving the condition number quality measure of surface mesh faces using optimization with respect to local parametric coordinates. The first method aims only to improve the condition number quality measure of all elements in the mesh as much as possible. The second method aims to improve the quality of all elements in the mesh and to minimize the deviation of the improved mesh from the original mesh. Section 4. presents several examples of triangular and quadrilateral meshes to demonstrate the capabilities of the features of both optimization methods.

2. OPTIMIZATION WITH RESPECT TO PARAMETRIC COORDINATES

Consider the gradient-based minimization of an objective function, $\Phi(\mathbf{x})$, defined in terms of the real coordinates, $\mathbf{x}$, of all the vertices of a surface mesh. In any gradient-based optimization process, each iteration involves finding the gradient of the objective function, computing a search direction based on the gradient, conducting a 1D minimization or line search along the search direction and updating the optimization variables [9, 10].

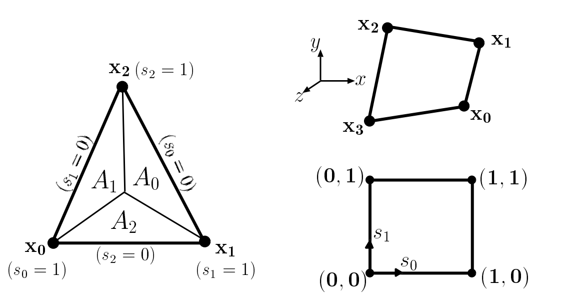
If an optimization procedure is applied directly to the objective function with respect to the real coordinates of the vertices, the gradient direction may indicate vertex movement off the original surface mesh. To constrain the movement of the vertices to the original surface mesh, the optimization is done with respect to the coordinates of the mesh vertices in a series of local parametric spaces derived from the faces of the original mesh. The parametric coordinates of a vertex in the interior of the mesh is derived from a local parametrization of the original mesh face it is contained in. The parametric coordinates of a vertex on the surface mesh boundary (model edge) is derived from the parametrization of the original boundary mesh edge that it is contained in. The optimization process moves each vertex in its appropriate local parametric space and when mapped back to real space, the vertex lies on the original mesh face corresponding to the local parametric space. If the optimization process drives the vertex out of bounds of a local parametric space, the vertex switches to the parametric space of an adjacent element.

In the following sections, element based local parametrization, 1D minimization in local parametric spaces and parameter updating for repositioning vertices are described in more detail.

2.1 Element based Local Parametrization

In this work, a local parametric space for a mesh triangle is derived using a barycentric mapping [11], resulting in parametric coordinates $0 \le (s_1, s_2) \le 1$ as shown in Figure 1a. A local parametric space for a quadrilateral is derived using isoparametric mapping [11], giving rise to parametric coordinates $0 \le (s_1, s_2) \le 1$, shown in Figure 1b. Meyer et. al. [6] have proposed a new barycentric mapping method which can be used to extend this procedure for parameterizing general straight sided polygonal faces.

Any procedure using the local parametrization defined above must keep track of which mesh element of the base mesh each node is in (referred to as the ***base*** element) and the coordinates of the node in the parametric space of the base element. During the optimization process, all objective function evaluations are done after mapping the parametric coordinates of the vertices to real coordinates. Also, the gradient of the objective function is computed with respect to the parametric coordinates by numerical differentiation.



$$
\begin{aligned}
A &= A_0 + A_1 + A_2 \\
s_i &= A_i / A, \quad 0 \le s_i \le 1 \\
s_0 &+ s_1 + s_2 = 1 \\
\mathbf{x} &= (x, y, z) \\
&= s_0\mathbf{x_0} + s_1\mathbf{x_1} + s_2\mathbf{x_2}
\end{aligned}
$$

(a)

$$
\begin{aligned}
& 0 \le s_0, s_1 \le 1 \\
\mathbf{x} &= (x, y, z) \\
&= (1 - s_0)(1 - s_1)\mathbf{x_0} + \\
& \quad s_0(1 - s_1)\mathbf{x_1} + s_0 s_1 \mathbf{x_2} + \\
& \quad (1 - s_0) s_1 \mathbf{x_3}
\end{aligned}
$$

(b)

Figure 1: (a) Barycentric mapping for triangle, (b) Isoparametric mapping for quadrilateral.

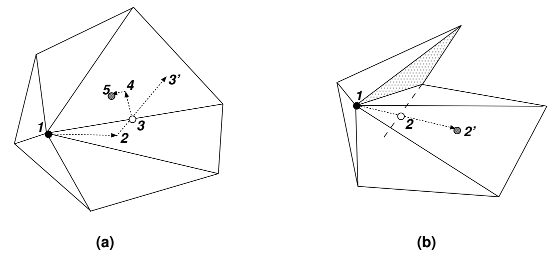


Figure 2: Line search constraints: (a) Parameter bounds, (b) Invalid Mesh.

2.2 Line Search or 1D minimization

In the optimization procedure described here, the gradient with respect to the local parametric coordinates is used to compute a search direction in the local parametric space. Then a line search is conducted to find a distance α, to move along the parametric search direction, $\mathbf{d}$, until the objective function is minimized or the constraints of the line search are encountered. For surface optimization with local parameterization, the line search is subject to two constraints, parametric bounds and mesh validity, as discussed next.

During a line search, a node that travels out of the parametric space of a base mesh face, moves out of the face and off the original surface mesh. In such a case, computing quantities such as the gradient of the objective function with respect to the current parametric coordinates of the vertex becomes meaningless. Therefore, if a node tries to move out of bounds of the local parametric space, the line search is stopped at the boundary of the base face. For example, in Figure 2a, the line search tries to proceed from point 2 to point 3', which is outside the triangle and off the surface triangulation. However, it encounters the parametric bounds of the triangle at point 3 (which is on an edge of the triangle) and therefore, the line search is stopped at that point.

Also, it is possible that one of the elements connected to the vertex becomes invalid due to the movement along the search direction in which case the line search must be stopped. This is shown in Figure 2b where the line search must be stopped at point 2 because further movement toward point 2' renders the shaded triangle invalid.

The line search procedure is implemented as an incremental stepping algorithm with step size control. The line search starts with a very small step size and checks if the function has decreased, the parameters are within bounds and if the mesh is valid. If so, the step size is increased and the process is repeated; if not, the step size is cut in half (up to a minimum) and the checks are repeated. The algorithm has additional refinements for zeroing in on the minimum with better accuracy.

At the start of the optimization, the base mesh face for initial movement of a node is chosen arbitrarily from the set of faces connected to the node. Therefore, it is possible that the objective function does not decrease along any direction in the chosen face and that a line search in the face will terminate without any movement from the current location. In such a case, an adjacent face connected to the node is chosen as the base face and the optimization iteration is repeated. Also, if the node is at a common edge of two faces of the original mesh, it is possible that the gradient in one face points into the adjacent face and vice versa, leading to the search switching infinitely between the two faces. This condition is recognized in the algorithm and resolved by moving the node along the edge. The line search direction along the edge is taken to be the one closer to the negative of the gradient direction.

2.3 Parameter Update and Parametrization Change

Once the line search along a direction has terminated, the step size (α) obtained from it is used to update the parametric coordinates of the vertex as $\mathbf{s}_{new} = \mathbf{s}_{old} + \alpha\mathbf{d}$. If the line search terminated normally at a minimum or because further movement in the search direction would have made the mesh invalid, a new optimization iteration is started with a new gradient calculation. However, if the line search terminated because the parametric bounds were reached, then it is assumed that the vertex is trying to move out of the current mesh face. In such a case, the optimization iteration is terminated and the node is switched to the parametric space of an adjacent mesh face. Since the node is moved into a different parametric space, the optimization procedure is restarted from the parametric location of the node in the new face, discarding the previous search direction and any saved gradient information (in a conjugate gradient method).

Figure 3 illustrates the movement of vertices during an optimization with respect to parametric optimization for a planar triangulation. The mesh was improved by minimizing an objective function based on the condition number quality measure (See Sec. 3.2) over the entire mesh. Note the node movements across several elements of the original mesh as well as movements along mesh edges.

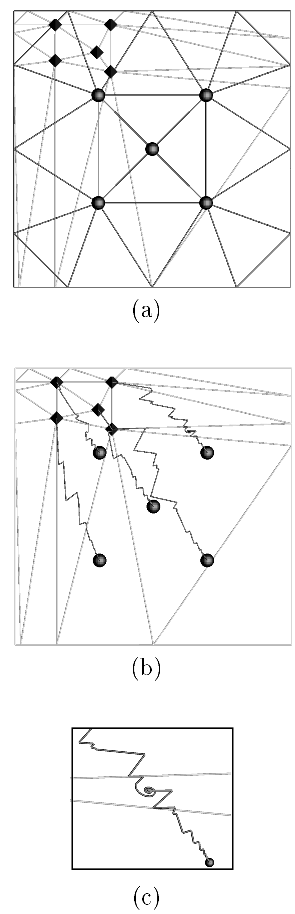


Figure 3: (a) Original (light lines) and final (dark lines) mesh, (b) Paths taken by vertices from their original positions (shown as $\diamond$) to their final positions (shown as $\bullet$), (c) Zoom-in of one of the paths.

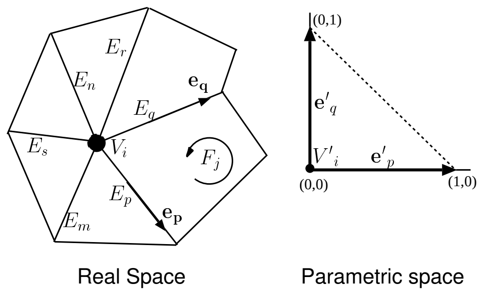


Figure 4: Definition of edge vectors, $\mathbf{e}_p$, $\mathbf{e}_q$ for calculating the Jacobian of an element F_j at vertex V_i.

3. OPTIMIZATION OF SURFACE MESH QUALITY

3.1 Condition Number Shape Measure for Mesh Faces

Of the many measures for evaluating the shape (or quality) of triangular and quadrilateral elements, the Condition Number Shape Measure [8] is one with a strong mathematical foundation. This measure is derived from the Jacobian matrix of a triangle mapping as described below.

Consider a vertex V_i, connected to a set of of edges, $\{E(V_i)\}$, and triangles, $\{F(V_i)\}$ as shown in Figure 4. Assume that one of the triangles $F_j \in \{F(V_i)\}$ has edges $E_p \in \{E(V_i)\}$ and $E_q \in \{E(V_i)\}$ connected to vertex V_i. This triangle can always be mapped to a right triangle in 2D space with V_i at the origin, a unit vector representing E_p along the x-axis and a unit vector representing E_q along the y-axis. Then, the *Jacobian matrix*, $\mathbf{J}_{ji}$, of the mapping of F_j to the right triangle, evaluated at vertex V_i, is given by $\mathbf{J}_{ji} = [\mathbf{e}_p\ \mathbf{e}_q]$ where, $\mathbf{e}_p$ and $\mathbf{e}_q$ are edge vectors representing edges E_p and E_q, of lengths l_p and l_q respectively.

Since $\mathbf{J}_{ji}$ is a 3x2 matrix for a triangle in 3D, its condition number must be calculated by singular value decomposition methods. On the other hand, the Jacobian matrix of a triangle in 2D space is a 2x2 matrix whose condition number can be calculated more easily as $\kappa(\mathbf{J}_{ji}) = (l_p^2 + l_q^2)/A_j$, where A_j is twice the area of face F_j [7, 8]. This condition number is only a function of triangle lengths[1]; therefore, it is invariant with rotation of the triangle in the plane. Since there always exists a coordinate system in which an arbitrarily oriented triangle lies on one of its coordinate planes, it suggests that the condition number is also useful for measuring the quality of arbitrarily oriented triangles in space.

[1] A_j is function of the lengths of the triangle sides

3.2 Condition Number Based Optimization

Consider the minimization of the sum of condition number measures of faces incident at a given vertex as given by the expression below:

$$\psi_i^c(\mathbf{x}_i) = \sum_j \kappa(\mathbf{J}_{ji}(\mathbf{x}_i)) = \sum_j \frac{l_p^2(\mathbf{x}_i) + l_q^2(\mathbf{x}_i)}{A_j(\mathbf{x}_i)}, \quad j \in \{j \mid F_j \in \{F(V_i)\}\} \tag{1}$$

where l_p and l_q are the lengths of edges E_p and E_q respectively and $\mathbf{x}_i$ is the coordinate vector of V_i. Note the presence of area A_j in the denominator as a barrierfunction which discourages node movements that tend to make the triangle formed by E_p and E_q degenerate.

The objective function ψ_c is designed such that its minimization attempts to smooth the distribution of face angles and edge lengths around a vertex. Based on this property, a strategy can be formed for improving the quality of a mesh by minimizing a global condition number based objective function, Ψ^c, defined as:

$$\Psi^c = \sum_i \psi_i^c, \quad i \in \{i \mid V_i \in \{V\}\} \tag{2}$$

where $\{V\}$ is the set of all mesh vertices. To minimize the global function, each vertex of the mesh is visited in turn and the position of the vertex is optimized using the local objective function, ψ_i^c. The local optimization can be done by any optimization method such as the non-linear conjugate gradient method. For surface meshes, the optimization is conducted with respect to local parametric coordinates as described in Section 2. Several optimization iterations are made over all the vertices of the mesh leading to a minimization of the global function, Ψ^c. The iterations are stopped when the movement of all vertices becomes negligible.

3.3 Reference Jacobian based Mesh Improvement Method

3.3.1 Motivation

The global condition number minimization procedure allows mesh vertices to move along the surface as much as necessary to minimize the objective function. However, in certain applications, it is of interest to keep the nodes of the original mesh as close as possible to their original locations while improving the shape of the mesh elements. The Arbitrary Lagrange-Eulerian (ALE) method [12] is one instance where it is important to minimize the deviation of the optimized mesh from the starting mesh. In ALE methods, the

Lagrangian step dictates a certain movement for the nodes based on the physics of the problem. This can cause the mesh to be distorted enough that the simulation cannot proceed unless the quality of the elements is improved. After the mesh is improved, the solution from the distorted mesh must be transferred to the improved mesh before continuing the simulation. Since the accuracy of the solution transfer depends strongly on the similarity of the two meshes, it is important to devise a procedure that improves mesh quality but also limits the extent that nodes can move from their original locations. Such an optimization procedure, referred to here as *Reference Jacobian based Mesh Improvement*, has been described earlier by Shashkov et. al. [7, 12] for planar meshes. In this work, the reference Jacobian based mesh improvement procedure has been combined with optimization with respect to local parametrizations, resulting in a strategy for improving surface mesh quality while keeping the nodes of the mesh on the faces of the original mesh and close to their original positions.

3.3.2 Local Condition Number based Optimization (Step I)

This is the first stage of the Reference Jacobian based mesh improvement strategy. In this step, the locally optimal position of each mesh vertex is computed with respect to the fixed position of its neighbors. The objective function for optimization is the local condition number function, ψ_i^c, described in Eq. 1, Section 3.2. However, in this step, the node is not moved to its locally optimal position. Rather, the optimal position of each node, described by a base face and the parametric coordinates of the node in the base face, is stored as a virtual position for use in the second stage of the mesh improvement procedure.

3.3.3 Reference Positions, Reference Edges and the Reference Jacobian Matrix

The locally optimal position computed and stored for each vertex in the first stage of the procedure is known as the *reference position* for the vertex. After reference positions are calculated for all mesh vertices, two *reference edge vectors* are calculated for each edge in the mesh; each reference edge vector goes from the reference position of one vertex of the edge to the original position of the other. The idea of reference edges is illustrated in Figure 5, where E_k is an edge with vertices V_i and V_j. The reference positions of V_i and V_j are V_i^R and V_j^R respectively. The two reference edge vectors for E_k are $(\mathbf{e}_k^R)_i$ and $(\mathbf{e}_k^R)_j$, where the outer subscript indicates which of the vertices is at its reference position.

Using the concept of reference edge vectors, it is now possible to define *Reference Jacobian Matrices* just as Jacobian matrices were defined for a mesh without reference positions. Therefore, if the edges of F_j connected to vertex V_i are E_p and E_q, then the reference Jacobian of F_j at V_i is defined as $\mathbf{J}_{ji}^R = \left[(\mathbf{e}_p^R)_i \ (\mathbf{e}_q^R)_i\right]$.

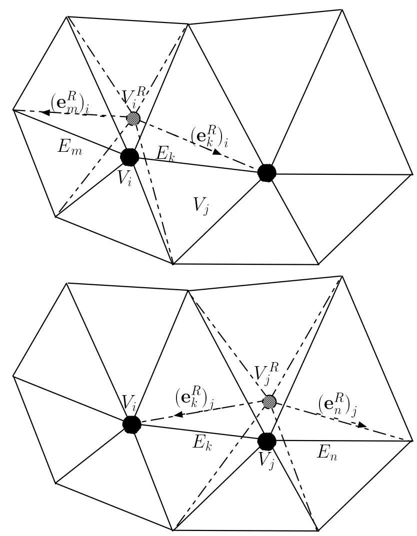


Figure 5: Reference positions and reference edge vectors.

3.3.4 Global Optimization based on Reference Jacobian Matrix (Step II)

The second stage of the mesh improvement procedure is a global optimization based on the definition of reference Jacobian matrices. The goal of this step is to find a valid mesh configuration such that each edge is in a compromise configuration between its pair of reference edge vectors. It is expected that such a configuration for the edges will improve mesh quality, since the reference edge vectors were formed by locally improving mesh quality at each mesh vertex. It is also expected that the optimized mesh will not deviate drastically from the base mesh, since each reference edge vector has one of its vertices at its original position and the other at the locally optimal position.

The objective function for the global optimization quantifies the difference between the Jacobian matrices of the mesh being optimized and the reference Jacobian matrices as shown below:

$$\Psi^R = \sum_i \sum_j \frac{\|\mathbf{J}_{ji} - \mathbf{J}^R_{ji}\|^2}{A_j / A^R_{ji}},$$
$$i \in \{i \mid V_i \in \{V\}\},\ j \in \{j \mid F_j \in \{F(V_i)\}\} \quad (3)$$

where, $\{V\}$ is the set of all mesh vertices, $||.||$ is the Frobenius norm[2], A^R_{ji} is the twice the area of the triangle formed by edge vectors, $(\mathbf{e}^R_p)i$ and $(\mathbf{e}^R_q)i$. Note that, similar to the objective function for local optimization, the objective function includes a barrier term A_j in the denominator in the form of the triangle area to prevent mesh invalidity. Since the Jacobian matrix and the reference Jacobian matrix are formed from the mesh edges and the reference edges respectively, optimization of Ψ^R makes the edges of the final mesh as close as possible to their respective reference edge vectors.

It would be most efficient if a global procedure could be used to minimize the objective function Ψ^R so that all the mesh vertices could be moved toward their optimal position simultaneously. However, the use of local parameterization for node movement imposes strong constraints on a global optimization process. The line search necessary in the global optimization seeks a single step size for the parametric coordinates of all the nodes. However, if a parametric coordinate for even a single vertex goes out of bounds, the line search must end for all the parameters in the problem and the optimization restarted, making the optimization very inefficient.

Therefore, the second optimization step is modified so that the mesh vertices are repositioned one vertex at a time using a local piece of the global objective function. Consider a vertex V_i, connected to the set of faces $\{F(V_i)\}$. Then the piece of the global objective function that involves the real and reference positions of V_i is given as:

$$\psi^R_i = \sum_j \sum_k \frac{\|\mathbf{J}_{jk} - \mathbf{J}^R_{jk}\|^2}{A_j / A^R_{jk}},$$
$$j \in \{j \mid F_j \in \{F(V_i)\}\},$$
$$k \in \{k \mid V_k \in \{V(F_j)\} \cap \{V(\{E(V_i)\}\}\ \}$$

In the expression, the outer sum is over all faces connected to the vertex and the inner sum is over all vertices of a face that include V_i itself or are edge-connected to V_i. Figure 6 shows the vertices involved in the expression ψ^R_i for V_i.

[2]Frobenius norm of matrix A is defined as $||A|| = \sqrt{tr(A^TA)}$ where $tr(A) = \sum_i a_{ii}$

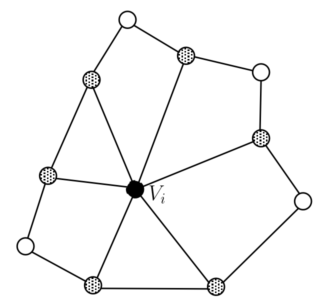


Figure 6: Vertices involved in the local objective function expression, ψ^R_i, for V_i. The shaded circles along with the black circle (V_i) represent the vertices at which real and reference Jacobians are computed for use in ψ^R_i. The white circles represent vertices whose real locations contribute to the Jacobians at the vertices with shaded circles.

Thus, the second stage of optimization visits each mesh vertex, V_i, and conducts a minimization of the local function, ψ^R_i by repositioning V_i. Minimization of the local function results in a reduction of the global function, Ψ^R. The procedure loops over all the mesh vertices several times until the optimization converges to a solution. The criteria for convergence is that the movement of all the nodes is negligible for several iterations. It can be seen that the first and second stage optimizations are similar except for the use of different objective functions.

4. RESULTS

Figure 7 shows a simple example to illustrate the effects of a condition number optimization (**CN Opt.** or **CNO**) and reference Jacobian based optimization (**RJ Opt.** or **RJO**) on a non-planar surface mesh. Figure 7a shows the original pyramid shaped mesh on which the two optimization techniques are applied. Figure 7b shows the effect of optimizing the CN objective function and Figure 7c shows the effect of optimizing the RJ objective function. In both cases, the apex node lies on the left lateral surface of the original pyramid. It can be seen that the CN optimization improves the shapes of the triangles more than the RJ optimization. On the other hand, the RJ optimization results in lesser movement of the apex node from its original position.

Figure 8 shows a quadrilateral mesh and the results of the CN optimization and RJ optimization. It is again clear from the example that the CN optimization improves the shape of mesh elements more than the RJ optimization, but it also causes much more movement of the nodes. This is verified by the following data about the change in the mesh characteristics between

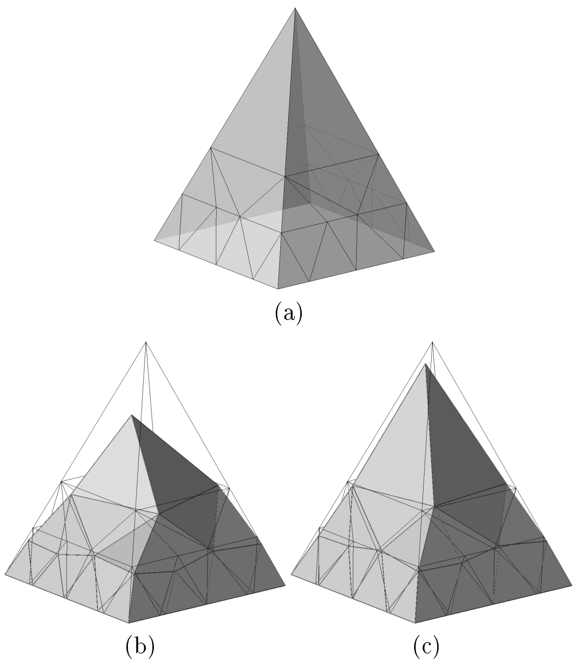


Figure 7: (a) Original Mesh, (b) Mesh optimized with condition number objective function, (c) Optimized with reference Jacobian objective function. Note that in both cases, the apex node is on the lateral surface of the original pyramid.

$\bar{\mathcal{K}}$			Original	CNO	RJO
1.0	–	1.5	84	114	112
1.5	–	2.0	26	0	2
2.0	–	3.0	3	0	0
3.0	–	4.0	1	0	0
4.0	–		0	0	0

Table 1: **Histograms of Normalized Average Condition Number in Original and Optimized Meshes for example shown in Figure 8**

the two meshes. Table 1 shows the improvement in the distribution of *normalized average condition number*, $\bar{\mathcal{K}}$, of elements in the mesh with the two types of optimization. The normalized average condition number for an element is defined as the mean of the condition numbers at the vertices of an element, normalized so that an equilateral triangle or square quadrilateral will produce a value of 1.

Table 2 shows various quantities computed to measure the change in the mesh and the surface using the two methods of optimization. In the computation, the normalized Hausdorff distance is computed by computing the minimum distance from each node of the original mesh to the new mesh, taking its maximum [13, 14] and normalizing it by the problem size. The problem size is defined as the maximum length of the domain

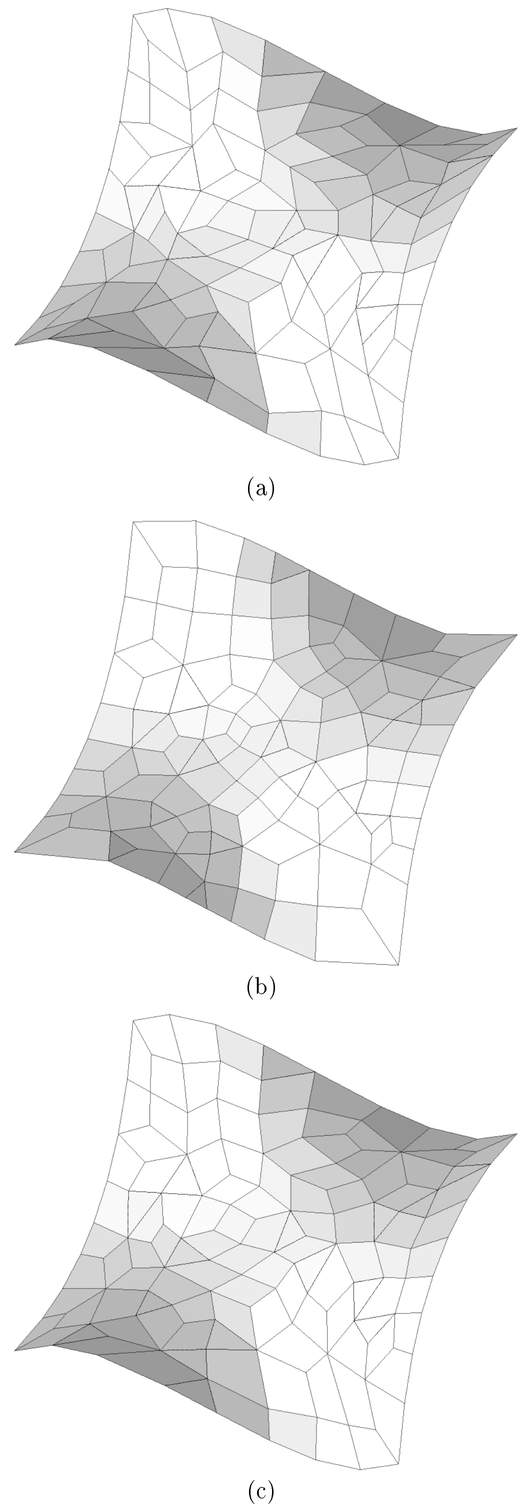


Figure 8: (a) Quadrilateral mesh of curved surface, (b) Mesh optimized with global condition number function, (c) Mesh optimized with reference Jacobian function.

Measure	CN Opt.	RJ Opt.
Hausdorff Distance	5.6%	1.0%
Max. Node Movement	19.1%	5.7%
Ave. Node Movement	5.2%	1.5%
Max. Change in Normals	13.3°	8.9°
Ave. Change in Normals	3.2°	1.9°

Table 2: Quantitative measures of the change in the mesh and discrete surface characteristics for CN optimization and RJ optimization for example shown in Figure 8; distances are presented as a percentage of the problem size

along the three coordinate directions. The maximum node movement is the maximum distance traveled by any node from its original position and the average node movement is the mean of the distance traveled by all nodes from their original positions; these are also normalized by the problem size. The change in surface normals is computed as the deviation between the discrete normal at a node of the optimized mesh and the normal at the corresponding location in the old mesh as shown in Figure 9. The average change in normals is simply the mean of the deviation in the normals over the entire mesh and the maximum change in normals is the maximum of the deviation in normals.

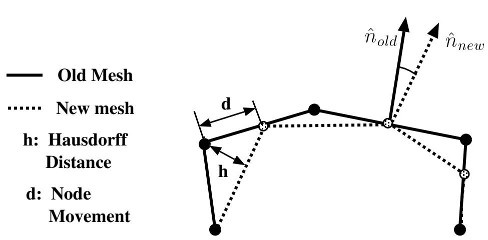


Figure 9: Comparing normals of the original and optimized meshes illustrated using an edge mesh.

Finally, a complex mesh of a sculpture is presented in Figure 10 to illustrate the effectiveness of this procedure on large surface meshes. The original mesh for this model was obtained from the Cyberware, Inc. (`http://www.cyberware.com/samples`) which was then coarsened using software from the Scientific Computation Research Center at Rensselaer Polytechnic Institute. The coarsened mesh (Figure 10a) was used to obtained the optimized meshes shown in the example. A CN optimization resulted in the mesh shown in Figure 10b and a RJ optimization yielded the mesh shown in Figure 10c. Figure 11a,b,c show zoomed in views of the original mesh, CN based optimized mesh and RJ based optimized mesh of the same example.

The condition number histograms for the three meshes are presented in Table 3 and the measures for change in surface characteristics are presented in Table 4. In this

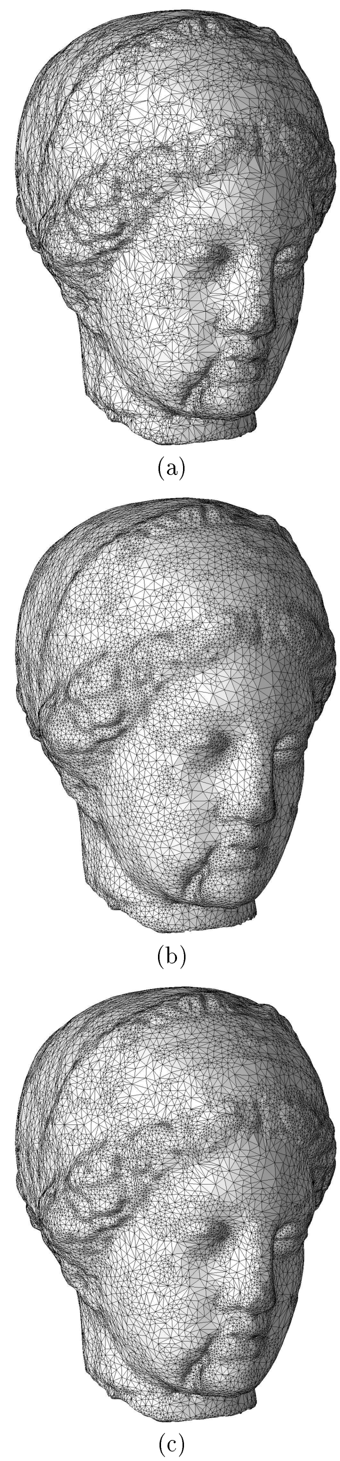
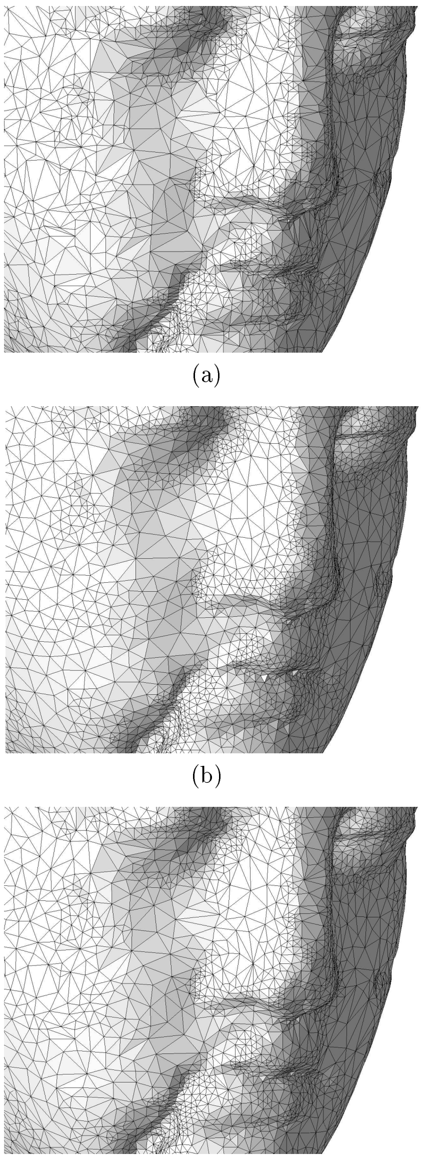


Figure 10: (a) Mesh of the Igea artifact (from Cyberware, Inc.), (b) Mesh optimized with CN objective function, (c) Mesh optimized with RJ objective function.



(a) (b) (c)

Figure 11: Zoom in of mesh of Igea artifact (Cyberware, Inc.), (a) Original mesh, (b) Mesh optimized with CN objective function, (c) Mesh optimized with RJ objective function.

$\mathcal{K}$			Original	CN Opt.	RJ Opt.
1.0	–	1.5	29572	39764	37432
1.5	–	2.0	7325	277	2371
2.0	–	3.0	2683	0	232
3.0	–	4.0	335	1	5
4.0	–	5.0	64	0	1
5.0	–	7.5	50	0	1
7.5	–	10.0	9	0	0
10.0	–	15.0	3	0	0
15.0	–		1	0	0

Table 3: Histograms of Normalized Average Condition Number in Original and Optimized Meshes for Igea artifact (Figure 10).

Measure	CN Opt.	RJ Opt.
Hausdorff Distance	0.5%	0.2%
Max. Node Movement	3.0%	1.3%
Ave. Node Movement	0.4%	0.2%
Max. Change in Normals	77.3°	66.3°
Ave. Change in Normals	6.2°	7.0°

Table 4: Quantitative measures of the change in the mesh and discrete surface characteristics for CN optimization and RJ optimization for Igea artifact (Figure 10); distances are presented as a percentage of the problem size

example too, it can be seen that the CN optimization causes more surface distortion but improves the shape of the elements more than the RJ optimization.

5. CONCLUSIONS

A procedure was presented to improve the quality of complex surface meshes without an underlying smooth surface using numerical optimization. The optimization is designed to improve the quality of the mesh faces without distorting the discrete surface too much. The vertices are kept on the original surface triangulation using movement in local parametric spaces of mesh faces. Two forms of optimization were proposed for improving the quality of the mesh. The first type of optimization was a global condition number optimization in which the condition numbers of faces at each mesh vertex were locally improved. The second type of optimization was a reference Jacobian based optimization in which the local condition number improvement is used only to calculate the locally optimal, virtual position for each mesh vertex. These virtual or reference positions are then used to form a reference Jacobian based objective function that effects a compromise between improving quality and keeping the nodes close to their original positions.

The procedure has been successfully tested on a number of complex triangular and quadrilateral surface meshes. Several quantitative measures were presented to show that both types of optimizations do not do not distort the surface much. The reference Jacobian based optimization strategy improves the mesh quality considerably but also keeps the nodes of the original mesh close to their original positions. On the other hand, the global condition number based optimization can cause considerable movement of the nodes from their original positions in order to provide a small improvement in mesh quality beyond what is possible by the reference Jacobian based method.

Future work will attempt to extend the procedure to general polygonal meshes.

6. ACKNOWLEDGMENTS

The work of Rao V. Garimella and Mikhail J. Shashkov was performed at Los Alamos National Laboratory operated by the University of California for the US Department of Energy under contract W-7405-ENG-36. Los Alamos National Laboratory strongly supports academic freedom and a researcher's right to publish; as an institution, however, the Laboratory does not endorse the viewpoint of a publication or guarantee its technical correctness. The work of Patrick M. Knupp was funded by the Department of Energy's Mathematics Information and Computational Sciences Program (SC-31) and was performed at Sandia National Laboratory. Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the US Department of Energy under contract DE-AC04-94AL85000.

The authors also acknowledge use of software tools from the Scientific Computation Research Center, Rensselaer Polytechnic Institute, Troy, NY that enabled the testing of the ideas presented in this paper.

References

[1] H. Hoppe and et.al. Mesh optimization. In *Computer Graphics (SIGGRAPH 93)*, pages 19–26, 1993.

[2] H. L. de Cougny. Refinement and coarsening of surface meshes. *Engineering with Computers*, 14(3):214, 1998.

[3] M.S. Floater. Parametrization and smooth approximation of surface triangulations. *Computer Aided Geometric Design*, 14:231–250, 1997.

[4] A Sheffer and E de Sturler. Parameterization of faceted surfaces for meshing using angle-based flattening. *Engineering with Computers*, 17(3):326–337, 2001.

[5] M. Desbrun, M. Meyer, and P. Alliez. Intrinsic parameterizations of surface meshes. In *Proceedings of Eurographics 2002 Conference*, Saaebrücken, Germany, September 2002.

[6] M. Meyer, H. Lee, M. Desbrun, and A.H. Barr. Generalizing barycentric coordinates for irregular n-gons. *Journal of Graphics Tools*, 2002. to appear.

[7] M. Shashkov and P.M. Knupp. Optimization-based reference-matrix rezone strategies for arbitrary lagrangian-eulerian methods on unstructured grids. In *Proceedings of the 10th Anniversary International Meshing Roundtable*, pages 167–176, Newport Beach, CA, 2001. Sandia National Laboratories. `http://math.lanl.gov/šhashkov`.

[8] P.M. Knupp. Achieving finite element mesh quality via optimization of the jacobian matrix norm and associated quantitites. Part I - a framework for surface mesh optimization. *International Journal for Numerical Methods in Engineering*, (48):401–420, 2000.

[9] J. Nocedal and S.J. Wright. *Numerical Optimization*. Springer, 1999.

[10] Reklaitis, Ravindran, and Ragsdell. *Engineering Optimization - Methods and Applications*. Wiley Interscience, 1983.

[11] T. J. R. Hughes. *The Finite Element Method: Linear Static and Dynamic Finite Element Analysis*. Prentice Hall, 1987.

[12] P. Knupp, L.G. Margolin, and M. Shashkov. Reference jacobian optimization-based rezone strategies for arbitrary lagrangian eulerian methods. *Journal of Computational Physics*, 176:93–128, 2002.

[13] P. Cignoni, C. Rocchini, and R. Scopigno. Metro: Measuring error on simplified surfaces. *Computer Graphics Forum*, 17(2):167–174, June 1998.

[14] D.P. Huttenlocher, G.A. Klanderman, and W.J. Rucklidge. Comparing images using the Hausdorff distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(3):850–863, September 1993.