

A MESH WARPING ALGORITHM BASED ON WEIGHTED LAPLACIAN SMOOTHING*

Suzanne M. Shontz[†]

Stephen A. Vavasis[‡]

[†]*Center for Applied Mathematics
Cornell University, Ithaca, NY 14853
shontz@cam.cornell.edu.*

[‡]*Department of Computer Science
Cornell University, Ithaca, NY 14853
vavasis@cs.cornell.edu.*

ABSTRACT

We present a new mesh warping algorithm for tetrahedral meshes based upon weighted laplacian smoothing. We start with a 3D domain which is bounded by a triangulated surface mesh and has a tetrahedral volume mesh as its interior. We then suppose that a movement of the surface mesh is prescribed and use our mesh warping algorithm to update the nodes of the volume mesh. Our method determines a set of local weights for each interior node which describe the relative distances of the node to its neighbors. After a boundary transformation is applied, the method solves a system of linear equations based upon the weights to determine the final position of the interior nodes. We study mesh invertibility and prove a theorem which gives sufficient conditions for a mesh to resist inversion by a transformation. We prove that our algorithm yields exact results for affine mappings and state a conjecture for more general mappings. In addition, we prove that our algorithm converges to the same point as both the local weighted laplacian smoothing algorithm and the Gauss-Seidel algorithm for linear systems. We test our algorithm's robustness and present some numerical results. Finally, we use our algorithm to study the movement of the canine heart.

Keywords: moving meshes, optimization-based mesh smoothing, unstructured mesh generation, tetrahedral meshes, cardiology

1. INTRODUCTION

Moving meshes arise in cardiology, computer graphics, animation, and crash simulation, among other applications in science and engineering. With moving meshes, the mesh is updated at each step in time due to a moving domain boundary, thus resulting in potentially drastically varying mesh quality from step to

step. One problem that can occur at each timestep is element inversion. We focus on maintaining good-quality tetrahedral meshes throughout the mesh warping process in this paper.

It is well-known that poor quality elements affect the stability, convergence, and accuracy of finite element and other solvers because they result in poorly conditioned stiffness matrices [1]. If well-shaped elements are not the result of updating the mesh boundary, the mesh quality must be improved by topological or geometrical means after each time step.

Research has shown that mesh smoothing (or r-

*THE WORK OF THE FIRST AUTHOR IS SUPPORTED BY THE NATIONAL PHYSICAL SCIENCE CONSORTIUM, SANDIA NATIONAL LABORATORIES, AND CORNELL UNIVERSITY. THE WORK OF THE SECOND AUTHOR IS SUPPORTED IN PART BY NSF GRANT ACI-0085969.

refinement) methods can be applied to improve the quality of a mesh. These methods adjust the positions of the vertices in the mesh while preserving its topology.

Laplacian smoothing is the most popular method for node-based mesh smoothing. In an iterative manner, it repositions the vertices of the mesh by moving each interior node to the geometric center of its neighbors. It is often used because it is computationally inexpensive and is very easy to implement. However, the method has several undesirable properties. One of them is that the method is not guaranteed to work, i.e., sometimes it inverts mesh elements. A second drawback is that the resulting mesh depends upon the order the nodes are smoothed.

A related type of smoothing, namely Winslow smoothing, is more resistant to mesh folding due to the requirement that the logical variables be harmonic functions. See [2] for more details on Winslow smoothing.

Other, more accurate methods for r-refinement are possible. Most of these methods are based upon optimization. Optimization-based methods are used with the goal of guaranteeing an improvement in the mesh quality by minimizing a particular mesh quality metric. Their main drawback, however, is their computational expense. Examples of optimization-based methods for r-refinement can be found in the following papers: [3], [4], [5], [6], [7], [8], [9], [10], [11], [12], [13], and [14]. For a theory of algebraic mesh quality metrics see [15].

To address the above issues, Baker [16] developed a three-step method for the metamorphosis of tetrahedral meshes. Each cycle of the method involves a combination of r-refinement, mesh coarsening, and mesh enrichment to adapt the mesh. The first step in the cycle is to move the interior nodes as far as possible using r-refinement while avoiding element inversion. The second step is to remove the poorly shaped elements in the mesh using mesh coarsening. The final step is the addition of elements to improve the mesh quality by mesh refinement. This dynamic procedure was shown to be more cost-effective than just r-refinement. One disadvantage of this technique is that it comes with no theoretical guarantees as it is difficult to analyze because each cycle is a combination of three very different techniques.

We study a different mesh warping problem where the connectivity of the mesh is not allowed to change, which is important for some applications. Seeking simplicity, preservation of the mesh's combinatorial structure, theoretical guarantees, and low computational expense, we developed a linear weighted laplacian smoothing (LWLS) method for mesh warping in two and three dimensions. A full description of the

algorithm is given in Section 2. In Section 3, we prove a theorem that gives sufficient conditions for when a mesh is able to resist inversion by a transformation. We also study mesh inversion within the context of our algorithm. In particular, we prove that our algorithm yields exact results for affine boundary transformations, and we state a conjecture for more general mappings. In Section 4, we show that our algorithm converges to the same point as both the local version of weighted laplacian smoothing and Gauss-Seidel. In Section 5, we test our algorithm on several types of mesh deformations. In Section 6, we apply our algorithm to study the motion of the beating canine heart under normal conditions. In Section 7, we summarize our work and identify directions for future research.

2. LINEAR WEIGHTED LAPLACIAN SMOOTHING

The problem that we address is as follows: Given a 3D domain, bounded by a triangulated surface mesh, and given an interior volume mesh composed of unstructured tetrahedra, suppose the triangulated surface mesh is displaced. Is there an algorithm to move the nodes of the volume mesh so that it continues to conform to the surface mesh and to be a good quality mesh?

Our mesh warping algorithm to address this question is based upon weighted laplacian smoothing. The first step in the algorithm is to generate a set of local weights for each interior node that represent the relative distances of the node to each of its neighbors. We use an interior point method from nonlinear programming in order to generate these weights. Next we apply a transformation to the boundary nodes. Using these new positions for the boundary nodes and the sets of weights from the original mesh, we solve a system of linear equations to determine new positions for the interior nodes. We now give a more detailed description of this algorithm.

Here we describe a nonlinear programming method for computing the weights for 2D meshes. Note that our method can be extended to 3D in the straightforward manner; however, we have not done extensive testing of our 3D implementation. In addition, this paper analyzes primarily the 2D version.

Let (x_i, y_i) denote the x - and y -coordinates respectively of the i th interior node in the initial mesh. In addition, let the x - and y -coordinates of its adjacent vertices be given by $\{(x_j, y_j) : j \in N_i\}$, where N_i denotes the set of neighbors of node i .

In order to find the set of weights w_{ij} , where w_{ij} is the weight of node j on interior node i , we use the log barrier function from linear programming to formulate the following optimization problem for each i :

$$\begin{aligned}
& \max_{w_{ij}, j \in N_i} \sum_{j \in N_i} \log(w_{ij}) & (1) \\
\text{subject to } w_{ij} & > 0, & (2) \\
\sum_{j \in N_i} w_{ij} & = 1, & (3) \\
x_i & = \sum_{j \in N_i} w_{ij} x_j, & (4) \\
\text{and } y_i & = \sum_{j \in N_i} w_{ij} y_j. & (5)
\end{aligned}$$

We note that the objective function together with the constraints form a strictly convex optimization problem for which there is a unique optimum. The optimum can thus be found by an interior point method. The use of convex optimization in mesh smoothing is not a new idea. Amenta et al. developed a framework for formulating mesh smoothing problems as quasi-convex programs so that they can be optimized as generalized linear programming problems [9]. In addition, Freitag et al. developed methods for local mesh smoothing and untangling using mesh quality metrics with convex function levels sets in [7] and [8].

By starting with an initial feasible point in the interior of the space of all possible weights, we are able to solve (1)-(5) using the Projected Newton Method. This method can be applied to solve optimization problems of the following form:

$$\begin{aligned}
& \min_x f(x) \\
\text{subject to } Ax & = b.
\end{aligned}$$

The optimality conditions for the above problem are:

$$\nabla f(x) - A^T \lambda = 0 \quad (6)$$

$$b - Ax = 0. \quad (7)$$

Here λ denotes the vector of Lagrange multipliers. Note that this is a system of nonlinear equations in x and λ ; thus Newton's method can be used to solve for x and λ . However, if the initial point is feasible (i.e., $b - Ax_0 = 0$) then p_k must satisfy $Ap_k = 0$ where $x_{k+1} = x_k + p_k$, i.e., p_k is the Newton direction at iteration k . This implies that $p_k = Zv_k$ for an $(n - m)$ -dimensional vector v_k , where Z is a basis for the null space of A . It is easy to see that the resulting equation for p_k is given by

$$p_k = - \left(H^{-1} - H^{-1} A^T [A H^{-1} A^T]^{-1} A H^{-1} \right) \nabla f(x_k), \quad (8)$$

where $H = \nabla^2 f(x_k)$ [17].

In order to compute an initial feasible point in the interior of the solution space, we choose three of the interior node's adjacent vertices and write (x, y) as a convex combination of the positions of the three nodes. This yields three positive weights; call them w_1, w_2 , and w_3 . In order to find a set of n positive weights, the rest of the w_i are initialized to ϵ , a small positive constant. Then once ϵ has been specified, linear equations (3) through (5) can be solved for new values of w_1, w_2 , and w_3 . For small values of ϵ , all of the constraint equations above will be satisfied (using the new values for the weights). An inexact linesearch is used to make ϵ as large as possible so that w_1, w_2 , and w_3 are still reasonably sized, i.e., each of them should be greater than some small tolerance.

Once the weights have been generated, the input surface deformation is applied to move the boundary nodes to new locations. The final step is to use the positions of the boundary nodes and the sets of weights to simultaneously determine the final positions for the interior nodes by solving a linear system of equations. The linear equations that must be solved are of the following form:

$$\sum_{j \in N_i} w_{ij} x_j = x_i \quad (9)$$

$$\sum_{j \in N_i} w_{ij} y_j = y_i. \quad (10)$$

The resulting linear systems from (9) and (10) are both $m \times m$, where m is the number of interior nodes.

We first prove that the resulting linear system has a unique solution.

Theorem 1 *The linear system (11), which expresses the position of each interior node as a convex combination of its neighbors, has a unique solution.*

PROOF.

We first establish some necessary notation and terminology. Let b and m represent the numbers of boundary and interior nodes, respectively. Next, define x_B and y_B to be vectors of length b that contain the initial x - and y -coordinates of the boundary nodes. Similarly, let x_I and y_I be vectors of length m that contain the initial x and y -coordinates of the interior nodes. Then $[x_B, y_B]$ and $[x_I, y_I]$ contain the original positions of the boundary and interior nodes respectively. Next, define the weighted Laplacian matrix, L , for a weighted graph $G(V; E; w)$ as follows:

$$L(i, j) = \begin{cases} -w_{ij} & \text{if } i \neq j, \\ \sum_{k \in V} w_{ik} & \text{if } i = j, \end{cases}$$

where $w_{ij} = 0$ iff $(i, j) \notin E$. The boundary nodes are assumed to be numbered last.

Denote by $A = [A_I, A_B]$ the matrix that is derived from the weighted Laplacian matrix in LWLS by deleting its last b rows. Note that A_I contains all of the weights corresponding to the interior neighbors and is $m \times m$. In addition, A_B contains all of the weights corresponding to the boundary neighbors and is $m \times b$. Linear systems (9)–(10) then can be expressed

$$A_I[x_I, y_I] = -A_B[x_B, y_B]. \quad (11)$$

To argue that we have uniqueness when LWLS is used to generate the weights, we classify A_I as an M -matrix which is defined as follows:

Definition: An $n \times n$ real matrix A is said to be an M -matrix if $a_{ij} \leq 0$ for all $i \neq j$ and $A^{-1} \geq 0$.

To see that A_I is an M -matrix, we make use of the following characterization theorem:

Theorem 2 [18] *If $A \in \mathbb{R}^{n \times n}$ satisfies $a_{ii} > 0$, $a_{ij} \leq 0$ for all $i \neq j$, is weakly row diagonally dominant (i.e., $Ae \geq 0, \neq 0$, where e is the vector of all 1's), and is irreducible, then A is an M -matrix.*

Because the mesh is connected and a positive weight is associated with each edge, we see that A_I is irreducible. Because the diagonal entries of A_I are 1, and the off-diagonal are negative and sum to a number in $[-1, 0]$, we see that A_I is diagonally dominant. Therefore, A_I satisfies the definition of an M -matrix by the above argument [19]. Thus, A_I is invertible, and (11) has a unique solution. ■

Because the above system has a unique solution, it is next solved via Gaussian elimination. Because the matrix is weakly diagonally dominant, partial pivoting is not necessary. This is a major advantage, since sparse matrix algorithms are much more efficient without pivoting. The sparsity structure is apparent, since, on average, an interior node has 6 neighbors in 2D, whereas a typical 2D mesh may have hundreds, thousands, or even millions of nodes. In addition, because the nonzero pattern is symmetric, we can first apply the symmetric minimum degree ordering to the appropriate matrix in an attempt to improve the speed of the computations. GMRES could also be used to solve the linear system instead of Gaussian elimination in an attempt to further improve the algorithm's speed although we have not tried this.

Another big advantage of our method is that if a continuous deformation of the boundary is given, then LWLS is a valid algorithm for computing the resulting trajectory that specifies the movement of the interior nodes. In addition, these trajectories will be continuous. This is vital for some applications where continuity of motion is required.

Consider the following application of the LWLS Method. Figure 1 shows the three steps of the mesh warping process. Note that the original mesh is shown in the top graph. A rather large deformation is then

applied to the boundary, and the result is shown in the middle graph. Finally, the interior nodes are moved to new positions as is shown in the bottom graph. The final mesh is a valid mesh, that is, it has no inverted elements.

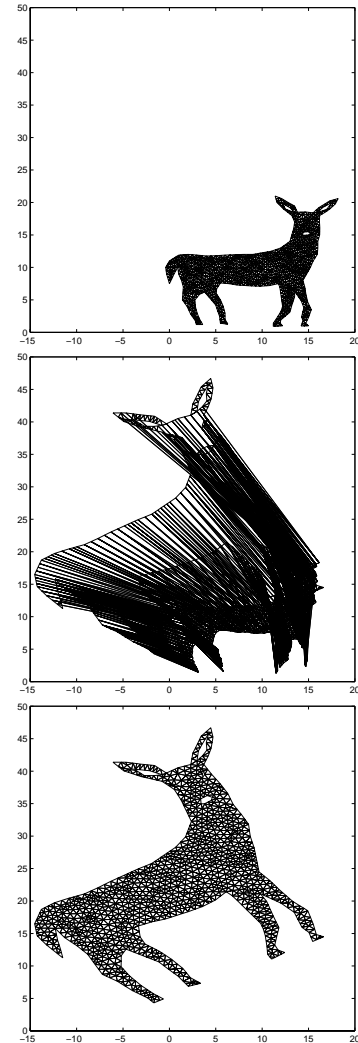


Figure 1: The three stages of the LWLS algorithm [20].

These figures show that the method demonstrates promise for use as both a mesh untangler and a smoother in the context of the mesh warping problem.

However, the method will not always be successful. If the boundary itself becomes tangled (self-intersecting) under the deformation, then there is no possibility that LWLS could recover an untangled mesh. But even if the boundary is not tangled, LWLS may still fail as indicated by our computational experiments below.

Because LWLS is not guaranteed to work for all types of boundary transformations, we seek to determine exactly what conditions will guarantee that LWLS will

be successful. In order to meet this goal, we first prove a theorem about mesh invertibility and then give a conjecture (with supporting theory) on sufficient conditions for LWLS to be successful on the mesh warping problem.

3. THEORY OF MESH INVERTIBILITY

In this section we consider sufficient conditions to guarantee that elements are not inverted in the mapped mesh. The material in this section focuses on the 2D case, but we also indicate results that extend to 3D. First, we prove a theorem about mesh invertibility in the case when a known transformation $f : \mathbb{R}^2 \rightarrow \mathbb{R}^2$ is applied to all the nodes. Then we conjecture that a similar theorem holds in the case when f is applied only to the boundary and the LWLS algorithm is used to position interior nodes (i.e., the situation under consideration). We provide some partial analysis in support of the conjecture.

In order to prove Theorem 3 below, we first prove the following useful lemma, which has appeared in other forms in the previous literature (e.g., [21]). This lemma extends to 3D as well if “angle” in 3D means solid angle.

Lemma 1 *Let T be a triangle with vertices $v_1, v_2, v_3 \in \mathbb{R}^2$. Let $A(T) = [v_2 - v_1; v_3 - v_1]$, and define the aspect ratio of T , $a(T)$, by $\frac{1}{a}$, where a is the minimum angle given in radians. Then, $a(T) \leq \kappa(A(T)) \leq 16a(T)$.*

Sketch of Proof: First, we argue that without loss of generality, we can assume that the smallest angle of T is at v_1 and that $\overline{v_2v_1}$ is longer than $\overline{v_3v_1}$. The second step in the proof is to compute $\|A(T)\|$ and $\|A(T)^{-1}\|$ and show that the quantities are related to the area and edge lengths of T . Throughout the proof we use $\|\cdot\|_\infty$ for our choice of norm. The result is that $\|A(T)^{-1}\| \leq \frac{\sqrt{2}}{\text{area}} l_{\max}$, where l_{\max} is the length of the longer of the edges $\overline{v_1v_2}$ and $\overline{v_2v_3}$. Similarly, it can be shown that $\|A(T)\| \leq 2\sqrt{2}l_{\max}$. Putting this together, we obtain that $\kappa(A(T))$ is bounded above by $2\frac{l_{\max}^2}{\text{area}(T)}$. Similarly, it can be shown that $\kappa(A(T))$ can be bounded below by $\frac{1}{2\text{area}(T)}l_{\min}l_{\max}$. Recall the product of an edge with its altitude is always twice the area of T . Using the relationships between side-lengths, we can rewrite our bounds as $\frac{l_{\min}}{h_{\min}} \leq \kappa(A(T)) \leq \frac{8l_{\min}}{h_{\min}}$. Now define θ to be the angle at v_1 . Because we assumed that the smallest angle is at v_1 , the longest edge is $\overline{v_2v_3}$. Therefore, $\frac{l_{\min}}{h_{\min}} = (\sin \theta)^{-1}$ and $a(T) = \frac{1}{\theta}$. So, we obtain $\frac{1}{\sin \theta} \leq \kappa(A(T)) \leq \frac{8}{\sin \theta}$. Note that $0 < \theta \leq \frac{\pi}{3}$ because θ is the smallest angle. Using basic calculus, we see that for $\theta \in [0, \frac{\pi}{3}]$, $\frac{1}{\theta} \leq \frac{1}{\sin \theta} \leq \frac{2}{\theta}$. Therefore, $a(T) = \frac{1}{\theta} \leq \frac{1}{\sin \theta} \leq \kappa(A(T)) \leq \frac{8}{\sin \theta} \leq \frac{16}{\theta} = 16a(T)$ which proves the lemma.

We are now ready to prove a theorem about mesh resistance to invertibility in the case that a transformation

is applied to all nodes. This theorem is stated for 2D but extends to 3D.

Theorem 3 *Suppose that $f : \mathbb{R}^2 \mapsto \mathbb{R}^2$ is bijective and differentiable on the entire mesh, with f' nonsingular. Suppose f is applied to all nodes of the mesh. Then the triangles in the mesh are not flipped by f if inequalities (14) - (16) below hold.*

PROOF.

For simplicity, assume for the rest of this section that $\det(f') > 0$ on the whole mesh since the case $\det(f') < 0$ is handled symmetrically. We give sufficient conditions for the noninvertibility of the mesh elements. Let t be a triangle in the triangulation T with vertices v_1, v_2 , and v_3 .

Note that t is flipped by $f \iff$

$$\frac{\begin{vmatrix} (v_2 - v_1)_x & (v_2 - v_1)_y \\ (v_3 - v_1)_x & (v_3 - v_1)_y \end{vmatrix}}{\begin{vmatrix} (f(v_2) - f(v_1))_x & (f(v_2) - f(v_1))_y \\ (f(v_3) - f(v_1))_x & (f(v_3) - f(v_1))_y \end{vmatrix}} < 0. \quad (12)$$

Using Taylor's Theorem, we obtain:

$$\begin{aligned} f(v_2) &= f(v_1) + J_{v_1}(v_2 - v_1) \\ &+ \frac{1}{2}v^T \begin{pmatrix} \frac{\partial^2 f}{\partial x^2}(\epsilon, \mu) & \frac{\partial^2 f}{\partial x \partial y}(\epsilon, \mu) \\ \frac{\partial^2 f}{\partial x \partial y}(\epsilon, \mu) & \frac{\partial^2 f}{\partial y^2}(\epsilon, \mu) \end{pmatrix} v, \end{aligned}$$

where $v = \begin{bmatrix} (v_2 - v_1)_x \\ (v_2 - v_1)_y \end{bmatrix}$ for some (ϵ, μ) lying on the line connecting v_1 and v_2 .

Substituting the above into (12) we obtain an inequality of the following form giving necessary and sufficient conditions for invertibility:

$$\frac{\begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix}}{|J_{v_1}| \begin{vmatrix} a_{11} + \gamma_1 & a_{12} + \gamma_2 \\ a_{21} + \gamma_3 & a_{22} + \gamma_4 \end{vmatrix}} < 0 \quad (13)$$

So, Lemma 2.7.1 in [22] applies to (13) once we have sufficient conditions for the hypotheses to hold. Note that the portion of the lemma we used states that if A is invertible, if $M = A + \delta A$ satisfies $\|\delta A\| \leq \epsilon \|A\|$, and if $\epsilon \kappa(A) < 1$, then M is nonsingular. Here $\kappa(A)$ denotes the condition number of A .

Using algebra and Lemma 1 above, we find that the following conditions ensure the hypotheses of the above lemma are satisfied:

$$\frac{1}{J_{v_1}} \left| \frac{\partial^2 f_1}{\partial x^2}(\epsilon) \right| \leq \frac{2}{51} \frac{h_{\min}}{a(T)h_{\max}^2} \quad (14)$$

$$\frac{1}{J_{v_1}} \left| \frac{\partial^2 f_1}{\partial x \partial y}(\epsilon) \right| \leq \frac{1}{51} \frac{h_{\min}}{a(T)h_{\max}^2} \quad (15)$$

$$\frac{1}{J_{v_1}} \left| \frac{\partial^2 f_1}{\partial y^2}(\epsilon) \right| \leq \frac{2}{51} \frac{h_{\min}}{a(T)h_{\max}^2}, \quad (16)$$

where h_{\min} and h_{\max} are the smaller and larger altitudes (respectively) from v_2 and v_3 . Similar conditions are obtained for f_2 . ■

Note that this proof generalizes to higher dimensions and is thus applicable to unstructured mesh generation in 3D.

Next, we state our conjecture for the LWLS method, which is formally reminiscent of Theorem 3. After stating the conjecture, we provide some supporting theory.

Conjecture 1 *Suppose that $f : \mathbb{R}^2 \mapsto \mathbb{R}^2$ is bijective and differentiable on the entire mesh, with f' non-singular. Consider transforming the mesh by applying f only to the boundary nodes of the triangulation, and then using LWLS to position the interior nodes. Then no triangles are flipped provided that an upper bound on the second derivative of f in terms of the first derivative and mesh aspect ratio is satisfied.*

The condition in the conjecture is analogous to the hypotheses in Theorem 3, namely, a condition for invertibility that relates the first and second derivatives of f . There is a qualitative difference, however, between the theorem and the conjecture: The theorem states that as long as f is bijective, then for a sufficiently refined mesh (with bounded aspect ratio), there will be no inversions. On the other hand, the conjecture and supporting theory here apparently indicate that the condition for occurrence of inversions in the LWLS algorithm depends on f and the mesh aspect ratio but not so much on the level of refinement.

In considering how to prove the conjecture, let us fix a particular triangle T in the original mesh's interior, centered at (x_c, y_c) with vertices v_1, v_2, v_3 . As noted above, the positions of the nodes in the updated mesh are obtained from solving $A_I[\hat{x}_I, \hat{y}_I] = -A_B[\hat{x}_B, \hat{y}_B]$, for $[\hat{x}_I, \hat{y}_I]$ given that $[\hat{x}_B, \hat{y}_B]$ are obtained from the original $[x_B, y_B]$. In fact, each two-element row of $[\hat{x}_B, \hat{y}_B]$ is f applied to the corresponding row of $[x_B, y_B]$. Let W_T be the $2 \times m$ matrix with 1's in the $(1, v_2)$ and $(2, v_3)$ positions and -1 's in the $(1, v_1)$ and $(2, v_1)$ positions. Then $W_T[\hat{x}_I, \hat{y}_I]$ is the 2×2 matrix, call it $A(T)$, whose determinant must be positive to avoid inversion. Observe that

$$A(T) = -W_T A_I^{-1} A_B [\hat{x}_B, \hat{y}_B]. \quad (17)$$

Next, expand f as a Taylor series about (x_c, y_c) , and for simplicity assume f is a quadratic function. The Taylor expansion therefore has the form

$$\begin{aligned} f(x, y) = & f(x_c, y_c) + D_x(f)(x_c, y_c)(x - x_c) \\ & + D_y(f)(x_c, y_c)(y - y_c) \\ & + \frac{1}{2} D_{x,x}(f)(x_c, y_c)(x - x_c)^2 \\ & + D_{x,y}(f)(x_c, y_c)(x - x_c)(y - y_c) \\ & + \frac{1}{2} D_{y,y}(f)(x_c, y_c)(y - y_c)^2 \end{aligned}$$

We can write the first three terms of this expansion as $v^T + [x, y]L$, where L is a 2×2 matrix and v^T is 2-vector, i.e., in the general form for an affine mapping, where L and v^T depend on the choice of triangle T . (The dependence on T is not explicitly denoted; the superscript T on v indicates "transpose.") Let us write the remaining terms as $Q((x, y) - (x_c, y_c))$, where Q is a pure quadratic function also depending on T . Thus,

$$f(x, y) = v^T + (x, y)L + Q((x, y) - (x_c, y_c))$$

We substitute this into (17) to obtain

$$\begin{aligned} A(T) = & -W_T A_I^{-1} A_B \left(ev^T + [x_B, y_B]L \right. \\ & \left. + \begin{bmatrix} Q((x_{B1}, y_{B1}) - (x_c, y_c)) \\ \vdots \\ Q((x_{Bb}, y_{Bb}) - (x_c, y_c)) \end{bmatrix} \right) \quad (18) \end{aligned}$$

where $(x_{B1}, y_{B1}), \dots, (x_{Bb}, y_{Bb})$ is an explicit enumeration of the nodes of $[x_B, y_B]$. We can analyze the sum of the first two terms on the right-hand side explicitly using the following lemma.

Lemma 2 *Suppose the boundary transformation applied to the initial mesh is an affine mapping. Then if LWLS is used to reposition the interior nodes, the resulting mesh is the same as if the affine mapping were applied to all nodes in the initial mesh.*

This lemma extends to 3D as well.

PROOF.

Continuing to use the above notation, we note that the position of the interior nodes in the deformed mesh is given by

$$[\hat{x}_I, \hat{y}_I] = -A_I^{-1} A_B ([x_B, y_B]L^T + ev^T). \quad (19)$$

In order to show that affine mappings yield exact results with LWLS, we want to show that (19) is the same as:

$$[\hat{x}_I, \hat{y}_I] = [x_I, y_I]L^T + ev^T. \quad (20)$$

Observe that if (20) is the same as (19), then it will follow that:

$$A_I([x_I, y_I]L^T + ev^T) = -A_B([x_B, y_B]L^T + ev^T). \quad (21)$$

Thus, it remains to check that (21) holds.

Because the weights for each interior node sum to 1, $Ae = \mathbf{0}$; i.e., $A_Ie + A_Be = \mathbf{0}$. Hence $(A_Ie + A_Be)v^T = \mathbf{0}$. Also, because $[x_I, y_I]$ and $[x_B, y_B]$ denote the original positions of the nodes, we know that $A_I[x_I, y_I] + A_B[x_B, y_B] = \mathbf{0}$. So, $(A_I[x_I, y_I] + A_B[x_B, y_B])L^T = \mathbf{0}$.

Putting these together, we see that

$$(A_I[x_I, y_I] + A_B[x_B, y_B])L^T + (A_Ie + A_Be)v^T = \mathbf{0}. \quad (22)$$

Therefore, (21) holds, and the lemma is proven. ■

For an example of using LWLS in conjunction with an affine boundary transformation, see Figure 2.

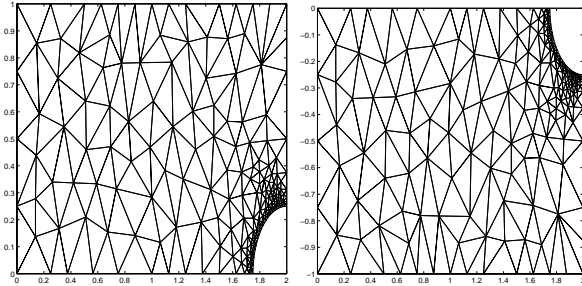


Figure 2: Exact results from LWLS for affine mapping [20].

Substituting the result of this lemma into (18) and using the fact that $W_Te = \mathbf{0}$ (by definition of W_T) yields

$$A(T) = W_T[x_I, y_I]L - W_TA_I^{-1}A_B \begin{bmatrix} Q((x_{B1}, y_{B1}) - (x_c, y_c)) \\ \vdots \\ Q((x_{Bb}, y_{Bb}) - (x_c, y_c)) \end{bmatrix}. \quad (23)$$

Observe that this equation is in a form analogous to the proof of Theorem 3: the matrix whose determinant we must take is written as the product of a determinant in the original mesh (namely $W_T[x_I, y_I]$) by the first derivative of f locally (namely, L) plus a quadratic term. We now wish to argue that the linear term dominates the quadratic term under suitable assumptions. Let us write the quadratic term as $q_1 + \dots + q_b$, where q_l denotes the contribution from the l th boundary node. In other words, $q_l = -W_TA_I^{-1}A_B(Q((x_{Bl}, y_{Bl}) - (x_c, y_c))e_l)$ where e_l denotes the l th column of the identity matrix. Fix l and let $r = -A_I^{-1}A_Be_l$, so that $A_Ir + A_Be_l = \mathbf{0}$ and

$$|q_l| \leq \|W_T r\| \cdot |Q((x_{Bl}, y_{Bl}) - (x_c, y_c))|. \quad (24)$$

If all the weights were equal and the mesh were uniform, the equation $A_Ir + A_Be_l = \mathbf{0}$ would be a discretization of Laplace's equation on the mesh domain,

in which the boundary condition is a Dirichlet condition of 1 at a single boundary node (node l) and 0 at all other boundary nodes. The quantity under consideration is $W_T r$, which (by definition of W_T) is a 2-vector each of whose entries is a finite difference of two entries of r at neighboring mesh nodes of triangle T centered at x_c, y_c . Therefore, we analyze this quantity by considering the model problem of solving Laplace's equation on the unit disk in which a Dirichlet boundary condition of 1 is imposed on boundary segment S of length h and a Dirichlet condition of zero is imposed on the rest of the boundary. Here h is the typical mesh cell size for the mesh under consideration. Let u be the Laplace solution for this boundary condition. The Poisson integral formula may be used to show that for any point t inside the unit disk, $\|\nabla u(t)\| \leq \text{const} \cdot h / \text{dist}(t, S)^2$. Therefore, if t_1, t_2 are two points in the unit disk satisfying $\|t_1 - t_2\| \leq h$, then

$$|u(t_1) - u(t_2)| \leq \text{const} \cdot h^2 / \min(\text{dist}(t_1, S), \text{dist}(t_2, S))^2.$$

If we hypothesize that a bound of the same form holds for the LWLS algorithm, we conclude that

$$\|W_T r\| \leq \text{const} \cdot h^2 / \text{dist}((x_c, y_c), (x_{Bl}, y_{Bl}))^2.$$

This takes care of the first factor in (24). Because the second factor is quadratic, it is bounded above by $\text{const}_2 \cdot \text{dist}((x_c, y_c), (x_{Bl}, y_{Bl}))^2$. Combining shows that $|q_l| \leq \text{const}_3 \cdot h^2$ since the dist^2 terms cancel each other. Since there are $b = O(1/h)$ terms in the sum $q_1 + \dots + q_b$, we conclude that the overall contribution from the quadratic term of (23) is $O(h)$ and that the constant factor in this $O(h)$ bound depends on the second derivative of f . On the other hand, the first term in (23) is also $O(h)$ (since W_T takes a finite difference of locations of mesh neighbors), but here the constant depends on the first derivative. This concludes our justification of the conjecture.

We have less insight into how the mesh aspect ratio affects (23), but we can show that no weight is too close to zero as long as the mesh has bounded aspect ratio.

Theorem 4 *There is a positive-valued function $q(a)$ such that $w_{ij} \geq q(a)$, where w_{ij} is any weight computed by the LWLS algorithm and a is the worst (largest) aspect ratio in the mesh.*

PROOF.

First, for this proof we need the following claim.

Claim. Let $p = (1, 0)$ be an interior node in a bounded-aspect ratio triangulation, and suppose that all of its neighboring nodes have nonnegative x -coordinates. Let x_{\max} be the maximum x -coordinate among neighbors of p . Then $x_{\max} \leq q_0(a)$, where q_0 is a fixed function and a is the mesh aspect ratio.

The proof of claim is omitted, but we give a sketch of its main idea here. For more details, see [24]. Consider the triangle t_1 containing p and also the line segment starting at p and proceeding in the negative x

direction. Since no neighbor of p has a negative x -coordinate, we conclude that this triangle t_1 has an altitude at most 1. Therefore, its maximum side-length is at most a , using the following definition of the aspect ratio: $AR = \frac{l_{\max}}{alt_{\min}}$, where l_{\max} and alt_{\min} denote the maximum side-length and minimum altitude. The triangle immediately adjacent to p , possessing one side of length at most a , therefore has maximum side-length at most a^2 , and so on. The number of triangles adjacent to p is bounded by ca , where c is a universal constant, since the minimum angle of each triangle is at least $1/(c'a)$, using the alternative definition of the aspect ratio, $AR = 1/\theta_{\min}$, where θ_{\min} is the minimum angle in the triangle. (Note that the two definitions of aspect ratio are equivalent up to a constant.) Therefore, the longest side-length among neighbors of p is a^{ca} , which is the function q_0 . A similar argument appears in [25].

Once the claim is established, we can return to the proof of the theorem, which is as follows. Fix an interior point p in the initial mesh with n neighbors, and let the optimal weights computed by the LWLS algorithm be w_1, \dots, w_n . (Omit the second subscript since p is fixed throughout the proof.) Let the neighbors of p be $(x_1, y_1), \dots, (x_n, y_n)$, and let p itself be located at (x^p, y^p) . Since the LWLS problem is a strictly convex optimization problem on a bounded convex set, the Lagrange multiplier conditions are necessary and sufficient for optimality [26]. These conditions are as follows. There exist three multipliers λ, μ, ν such that

$$\begin{aligned} w_1 x_1 + \dots + w_n x_n &= x^p, \\ w_1 y_1 + \dots + w_n y_n &= y^p, \\ w_1 + \dots + w_n &= 1, \\ \lambda x_1 + \mu y_1 + \nu &= 1/w_1, \\ &\vdots \\ \lambda x_n + \mu y_n + \nu &= 1/w_n. \end{aligned}$$

First, let us consider a very simple case that $\lambda = \mu = 0$. In this case, each w_i is equal to $1/\nu$. Since the sum of the w_i 's is 1, this means in fact that $1/\nu = 1/n$ and hence $w_i \geq 1/n$ for all i . Furthermore, $1/n > 1/(ca)$ using the argument about the number of neighbors from the proof of the claim. This shows that $w_i > q(a)$ in this special case.

The generic case is that at least one of λ, μ is nonzero. Without loss of generality, we can assume $\lambda = 1, \mu = \nu = 0$. The reason this assumption is WLOG is as follows. We can apply an arbitrary transformation of the form

$$(x, y) \mapsto (x, y) \begin{pmatrix} u & v \\ -v & u \end{pmatrix} + (k_1, k_2)$$

(with at least one of u, v nonzero) uniformly to the original data $(x_1, y_1), \dots, (x_n, y_n), (x^p, y^p)$; applying such a transformation has the effect of transforming λ, μ, ν arbitrarily as long as not both λ, μ are zero but does not change any of the aspect ratios in the original triangulation (since the above transformation is a combination of a rigid motion, a translation, and a uniform scaling).

After this transformation is applied, we can further apply a uniform translation to all the y -coordinates in the original data to force $y^p = 0$. Since $\lambda = 1$ and $\mu = \nu = 0$, we conclude from the KKT system that $x_i = 1/w_i$ so that $w_i x_i = 1$. This means that $w_1 x_1 + \dots + w_n x_n = n$, hence $x^p = n$ so $p = (n, 0)$. Also, the relationship $x_i = 1/w_i$ means that all the x_i 's are positive. Finally, the relationship $x_i = 1/w_i$ means that $\min w_i = 1/\max x_i$. In particular, in order to have a very small w_i means that there must be a very large x_i . It follows from the claim that $\max x_i \leq nq_0(a)$, hence $\min w_i \geq 1/(nq_0(a))$. Therefore, there exists a positive-valued function $q(a)$ such that all the weights are at least $q(a)$. ■

4. CONVERGENCE THEORY

Next, we study the convergence of LWLS and compare it with the convergence of the local weighted Laplacian smoothing and Gauss-Seidel algorithms.

Theorem 5 *The LWLS Algorithm converges to the same point as the local weighted Laplacian smoothing method and the Gauss-Seidel method taken one pass at a time.*

PROOF.

First, we note that as proved in Theorem 1, unique positions result when using LWLS to solve for positions of the interior nodes. Throughout this discussion we assume that the input transformation has already been applied to the boundary.

Now, we discuss the convergence of the Gauss-Seidel algorithm when applied to $A_I[\hat{x}_I, \hat{y}_I] = -A_B[\hat{x}_B, \hat{y}_B]$. It is well-known in the numerical analysis community that this algorithm is guaranteed to converge for an M -matrix. (See [27] for the details.) Thus, according to Theorem 2, it will converge when applied to this system.

Next, we observe that it is easy to show the iterates produced by applying the local version of weighted Laplacian smoothing to the above linear system are the same. (The details are omitted due to space constraints.)

Thus, we can compute the iterates for the local version of weighted Laplacian smoothing as follows:

$$[x_I, y_I]^{(k+1)} = C[x_I, y_I]^{(k)} + F[\hat{x}_B, \hat{y}_B],$$

where $C = (M_G^{-1}N_G)$, $F = (-M_G^{-1}A_B)$, $M_G = D + L$, $D = \text{diag}(A_I)$ and $L = \text{tril}(A_I)$, $N_G = -\text{triu}(A_I)$, and $[x_I, y_I]^{(0)} = [x_I, y_I]$.

Finally, we show that the local version of weighted Laplacian smoothing converges to the same point as LWLS. To this end, we consider the convergence of the local version of the weighted Laplacian smoothing method. In order to study its convergence, we need to concern ourselves with

$$\lim_{k \rightarrow \infty} \left(\begin{pmatrix} C & F \\ 0 & I \end{pmatrix}^k \right). \quad (25)$$

Note that (25) is equal to

$$\begin{pmatrix} \lim_{k \rightarrow \infty} C^k & \left(\sum_{j=0}^{\infty} C^j \right) F \\ 0 & I \end{pmatrix}. \quad (26)$$

Now, observe that $\left(\sum_{j=0}^{\infty} C^j \right) F = (I - C)^{-1} F$. In addition, observe that A_I and $-A_B$ can be obtained from $I - C$ and F via elementary row operations since $I - C = I - M_G^{-1} N_G$ and $F = -M_G^{-1} A_B$. Thus, the desired matrices are obtained by left multiplying by M_G . This proves that $(I - C)^{-1} F = -A_I^{-1} A_B$.

The final observation to make is that the eigenvalues of C satisfy $\rho(\lambda) < 1$, which can be seen as follows. First, we note that all of the entries of C are nonnegative by definition of C . Second, we note that each row sum is less than or equal to 1 since each row is a convex combination of rows with nonnegative entries and row sums that are less than or equal to 1. In addition, there is at least one row with row sum strictly less than 1 since there is at least one interior node with a boundary neighbor. Thus, the eigenvalues of C are less than 1 in absolute value, i.e., $\lim_{k \rightarrow \infty} C^k = 0$.

Putting everything together, we see that (26) is equal to

$$\begin{pmatrix} 0 & -A_I^{-1} A_B \\ 0 & I \end{pmatrix}. \quad (27)$$

This proves the theorem, since the solution to the second linear system, $A_I[\hat{x}_I, \hat{y}_I] = -A_B[\hat{x}_B, \hat{y}_B]$, can be written as follows:

$$\begin{pmatrix} \hat{x}_I & \hat{y}_I \\ \hat{x}_B & \hat{y}_B \end{pmatrix} = \begin{pmatrix} 0 & -A_I^{-1} A_B \\ 0 & I \end{pmatrix} \begin{pmatrix} x_I & y_I \\ \hat{x}_B & \hat{y}_B \end{pmatrix}.$$

■

5. NUMERICAL TESTS

In order to test the robustness of LWLS, we designed three numerical tests each based upon a series of mesh deformations. These tests have been designed to mimic the basic elements of a beating heart's motion, namely translation and rotation. We study the deformation of an annulus on each timestep. The annulus is composed of four, equally-spaced concentric rings of triangles. Its inner radius is 1, and its outer radius is 10. The initial annulus mesh is shown in the upper, left-hand corner of Figure 3.

In the first test, the inner circle was deformed via an outward translation on each iteration. In particular, its radius was increased by 0.5 each timestep. The test showed that when the inner radius of the annulus became 6.5, too much deformation had occurred, since

the mesh had inverted, and the minimum mean ratio of a triangle in the mesh was less than 1. (See [3] for the mean ratio definition.) However, as this test demonstrates, LWLS tolerates a large amount of translation is tolerated before inversion occurs.

Rotation of the inner circle occurs in the second deformation series. On each iteration, the inner circle was rotated counterclockwise 10 degrees. While performing this test, we observed that as the inner circle rotates counterclockwise, the quality of the triangles in the inner ring decreases. The mesh remained untangled until it had been rotated 70 degrees.

The goal of the third test was to combine translation and rotation of the inner circle. At each timestep, the motions of the first two tests were combined, i.e., the radius of the inner circle was increased by 0.5, and it was rotated by 10 degrees. Figure 3 shows the results. The new radius for the inner circle, its amount of rotation, and the minimum mean ratio of the triangles are given. Observe that the mesh remains untangled until the radius of the inner circle is 3.5 and the circle has been rotated by 50 degrees.

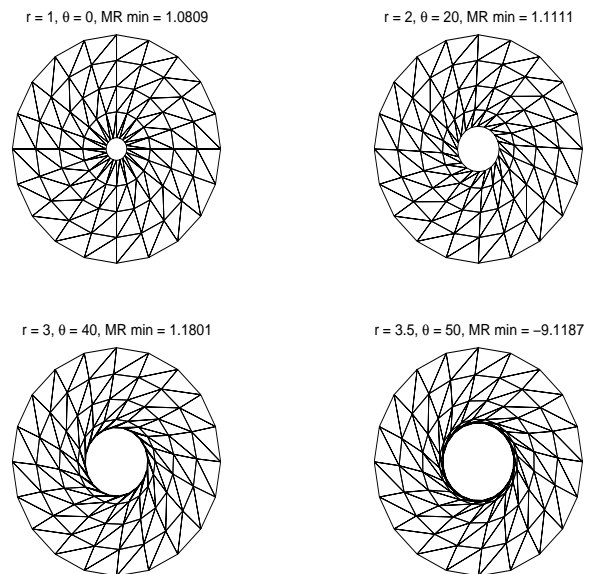


Figure 3: Translation and Rotation of Inner Circle of Annulus.

From these three tests, we conclude that LWLS can withstand relatively large amounts of deformation of various kinds while resisting inversion. However, whenever the deformations either tangle the boundary or are too large, LWLS fails because inversion of elements occurs.

6. APPLICATION TO CARDIOLOGY

We now use LWLS in order to study the movement of the beating canine heart under normal conditions. To do this, we obtained data from the Laboratory of Cardiac Energetics at the National Institutes of Health (NIH) [28]. We were given (x, y, z, t) data for 192 points on the inner surface of the left and right ventricles of the beating canine heart from a physiological pacing experiment. The data frames are 14.6 milliseconds apart with the first frame occurring 12 milliseconds before the pacing spike.

The first step in the simulation of the ventricular movement was to generate a mesh for the initial position of the ventricles. In order to do this, we first noted that the 192 points we were given were arranged in 8 slices with 24 points each. Thus, in order to generate the initial mesh, we decided to create a mesh for the top slice and then use LWLS to do the mesh warping necessary to create meshes for the remaining slices. Note that this uses LWLS in one and two dimensions as we describe in detail below. Once we have the meshes for all of the levels, we connect the triangular meshes into a tetrahedral mesh for the ventricles. The procedure to do this is also described below.

We now give a more detailed description of the method we used to create the initial mesh of the canine ventricles. We first used the two-dimensional quality mesh generation package called Triangle [29] to generate an initial mesh of the top slice (after projecting it into the x - y plane). Note that this yielded a good-quality mesh in the x - y plane with several additional nodes. Second, we computed the z -coordinates for the new points on the boundary of the top slice using 1D LWLS. Third, we used the weight-finding portion of our LWLS algorithm to compute the weights for the appropriate 2D linear system obtained from the x - and y -coordinates. Fourth, we determined the z -coordinates for the mesh of the top slice by forcing the z -coordinates to satisfy the appropriate 3D linear system using the 2D weights. At this point, we had the mesh for the top slice.

Our second task was to generate meshes for each of the remaining 7 slices. This was done using our LWLS algorithm to warp the mesh for the top slice into meshes for each of the remaining slices. In order to accomplish this, the first step was to determine the coordinates of the additional boundary nodes for the mesh of the appropriate slice. This was done using 1D LWLS. Then, the (x, y) coordinates of the interior nodes of that slice were determined using 2D LWLS. The z -coordinates for the interior nodes were found by forcing them to satisfy the appropriate 3D linear system using these weights.

The third step was to connect the triangular meshes for each of the 8 slices into one tetrahedral mesh for the canine ventricles. To do this, the corresponding triangles between two slices were connected to form a triangular prism. After a temporary mesh of triangular prisms was created, the triangular prisms were subdivided into tetrahedron using the method outlined in [30].

After the initial tetrahedral mesh was created, we checked its quality using the mean ratio mesh quality metric. Using this test, it was determined that the initial mesh was of poor quality. This was because the 8 slices were equally-spaced even though the curvature of the ventricles changes much more rapidly near the bottom of the heart. Thus, we decided to use linear interpolation in conjunction with LWLS in the obvious way in order to add two additional slices of nodes near the bottom of the heart. Because the curvature of the ventricles changes more rapidly near the bottom of the heart, the first additional slice was placed halfway between levels 7 and 8, and the second additional slice was placed halfway between the first new level and level 8. The resulting initial mesh is shown in Figure 4.

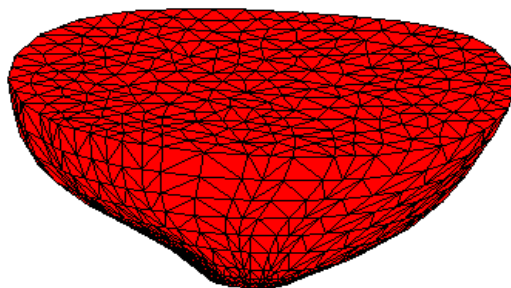


Figure 4: Initial canine ventricular mesh.

After the initial mesh was created, the heart data was used to move the 192 data points on the boundary of the mesh to their new positions. The same process as above (i.e., LWLS in 1D, LWLS in 2D, and using 1D LWLS to add the two new levels of data near the bottom) was used in order to reposition the remaining boundary points. Once the boundary nodes were relocated to their positions for timestep $t = 2$, the 3D version of LWLS was used to move the interior nodes to their new positions for this timestep. This process was performed iteratively in order to study the movement of the heart at timesteps $t = 3, \dots, 32$.

The simulation of the canine ventricular movement produced a series of meshes that show the ventricles twisting, expanding, and then contracting over the cardiac cycle. This is consistent with what occurs in nature. These meshes were combined into an animation which can be seen at [31]. Because the dynamic range of the motion is small, it cannot be detected in single figures (separate from an animation).

During each timestep, the mean ratios of the tetrahedra in the mesh were computed. The mean ratio computations showed that the heart remained untangled throughout the entire simulation. This is not very surprising since the heart mesh is composed of elliptical rings that seem to undergo less movement on

each timestep than the circular rings in the test cases. However, the motion of the heart is anisotropic which makes it difficult to predict in advance how it will tolerate deformations. Interestingly enough, the values of the minimum and average mean ratios were relatively constant across all timesteps. Only the value of the maximum mean ratio changed a significant amount. The mean ratio computations are also a good indication that the heart meshes are of sufficiently good quality for use with a numerical PDE solver that requires moving meshes.

In order to further test our mesh warping algorithm, the motion of the ventricles was exaggerated by a factor of 3. In this case, the motion was large enough to detect in separate figures and is shown in Figure 5. We note that the LWLS algorithm also performed successfully in this case, which is encouraging given the much larger deformations.

7. CONCLUSIONS

In summary, we developed a new mesh warping algorithm for tetrahedral meshes based upon weighted laplacian smoothing.

Our method determines a set of local weights for each interior node which describe the relative distances of the node to its neighbors. A deformation is then applied to the boundary, and the method solves a system of linear equations based upon the weights to determine the final positions of the interior nodes.

Furthermore, LWLS is simple, preserves the mesh's combinatorial structure, is computationally inexpensive, and provides theoretical guarantees for use on the mesh warping problem.

We proved a theorem regarding mesh invertibility and a lemma showing that LWLS yields exact results for affine boundary transformations. In addition, we proved a convergence result comparing LWLS to the local weighted laplacian smoothing and Gauss-Seidel algorithms.

We also used LWLS to study the motion of the canine ventricles under normal conditions.

We now describe some possibilities for future research. First, we would like to extend the above theory to prove Conjecture 1. Second, we would like to use our canine ventricle meshes to study the bioelectricity of the heart. To do this, we will couple the heart's electrical activity with its mechanical motion (from the LWLS meshes). We will then use the finite element method to simulate the electricity on the beating heart. This will yield new knowledge of the heart's activity since most of the current models have not coupled the heart's electricity with its mechanical motion.

8. ACKNOWLEDGEMENTS

The authors wish to thank O. Faris and E. McVeigh from the Laboratory of Cardiac Energetics at the

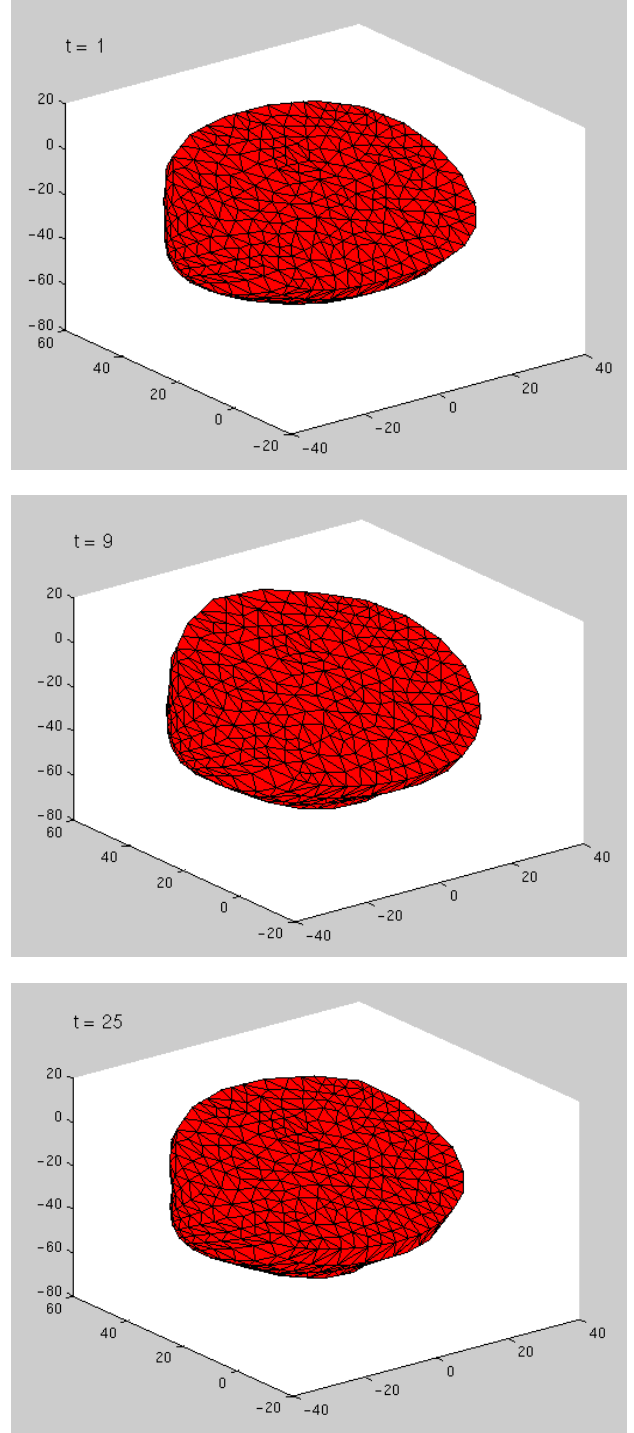


Figure 5: Exaggerated Movement of the Heart as it Beats. Motion for timesteps $t = 1, 9,$ and 25 .

National Institutes of Health for the moving canine heart data that they provided. In addition, they wish to thank Dr. Lori Freitag of Lawrence Livermore National Laboratory for providing us with two 2D test meshes. They benefited from conversations with G. Bailey and especially H. Kesten of Cornell. Finally, they thank the two anonymous referees for very helpful comments on the first version of this manuscript.

References

- [1] Shewchuk J. “What is a Good Linear Element? Interpolation, Conditioning, and Quality Measures.” *Proceedings of the Eleventh International Meshing Roundtable*, pp. 115–126. Sandia National Laboratories, Albuquerque, NM, 2002
- [2] Knupp P. “Winslow Smoothing on Two-Dimensional Unstructured Meshes.” *Engineering with Computers*, vol. 15, 263–268, 1999
- [3] Freitag L., Knupp P., Munson T., Shontz S. “A comparison of optimization software for mesh shape-quality improvement problems.” *Proceedings of the Eleventh International Meshing Roundtable*, pp. 29–40. Sandia National Laboratories, Albuquerque, NM, 2002
- [4] Freitag L., Knupp P. “Tetrahedral mesh improvement via optimization of the element condition number.” *Int'l. J. Numer. Meth. Engr.*, vol. 53, 1377–1391, 2002
- [5] Freitag L., Knupp P. “Tetrahedral Element Shape Optimization via the Jacobian Determinant and Condition Number.” *Proceedings of the 8th International Meshing Roundtable*, pp. 247–258. Sandia National Laboratories, Albuquerque, NM, 1999
- [6] Freitag L., Ollivier-Gooch C. “Tetrahedral mesh improvement using swapping and smoothing.” *Int'l. J. Numer. Meth. Engr.*, vol. 40, 3979–4002, 1997
- [7] Freitag L. “On Combining Laplacian and Optimization-based Mesh Smoothing Techniques.” *AMD - Vol. 220 Trends in Unstructured Mesh Generation*, pp. 37–43. ASME, 1997
- [8] Freitag L., Plassmann P. “Local optimization-based simplicial mesh untangling and improvement.” *Intl. J. Num. Methods in Engr.*, vol. 49, 109–125, 2000
- [9] Amenta N., Bern M., Eppstein D. “Optimal Point Placement for Mesh Smoothing.” *Proceedings of 8th ACM-SIAM Symposium on Discrete Algorithms*, pp. 528–537. 1997
- [10] Amezua E., Hormaza M., Hernández A., Ajuria M. “A method for the improvement of 3D solid finite-element meshes.” *Advances in Engineering Software*, vol. 22, 45–53, 1995
- [11] Parthasarathy V.N., Kodiyalam S. “A constrained optimization approach to finite element mesh smoothing.” *Finite Elements in Analysis and Design*, vol. 9, 309–320, 1991
- [12] Canann S.A., Stephenson M.B., Blacker T. “Optismoothing: An optimization-driven approach to mesh smoothing.” *Finite Elements in Analysis and Design*, vol. 13, 185–190, 1993
- [13] White D., Rodrigue G. “Improved Vector FEM Solutions of Maxwell’s Equations Using Grid Pre-Conditioning.” *Int'l. J. Numer. Meth. Engr.*, vol. 40, 3815–3837, 1997
- [14] Zavattieri P., Dari E., Buscaglia G. “Optimization Strategies in Unstructured Mesh Generation.” *Int'l. J. Numer. Meth. Engr.*, vol. 39, 2055–2071, 1996
- [15] Knupp P. “Algebraic Mesh Quality Metrics.” *SIAM Journal on Scientific Computing*, vol. 23, 193–218, 2001
- [16] Baker T.J. “Mesh Movement and Metamorphosis.” *Proceedings of the Tenth International Meshing Roundtable*, pp. 387–396. Sandia National Laboratories, Albuquerque, NM, 2001
- [17] Nash S.G., Sofer A. *Linear and Nonlinear Programming*. McGraw-Hill, Inc., 1996
- [18] Windisch G. *M-matrices in numerical analysis*. Teubner, 1989
- [19] Ortega J. *Numerical Analysis: A Second Course*. Academic Press, 1972
- [20] Freitag L. *Personal Communication*
- [21] Mitchell S., Vavasis S. “Quality Mesh Generation in Higher Dimensions.” *SIAM J. Computing*, vol. 29, 1334–1370, 2000
- [22] Golub G.H., Van Loan C.F. *Matrix Computations*. John Hopkins University Press, third edn., 1996
- [23] Doyle P.G., Snell J.L. *Random walks and electric networks*. The Mathematical Association of America, 3.02 edn., 2000
- [24] Shontz S.M. *A Mesh Warping Algorithm Based on Weighted Laplacian Smoothing*. Ph.D. thesis, Cornell University, Expected 2004
- [25] Mitchell S.A. *Mesh generation with provable quality bounds*. Ph.D. thesis, Cornell University, 1993
- [26] Nocedal J., Wright S.J. *Numerical Optimization*. Springer, New York, 1999
- [27] Varga R. *Matrix Iterative Analysis*. Springer-Verlag, second edn., 2000
- [28] McVeigh E., Faris O. *Personal Communication*, February 2003
- [29] Shewchuk J. “Triangle: Engineering a 2D Quality Mesh Generator and Delaunay Triangulator.” *Proceedings of the First Workshop on Applied Computational Geometry*, pp. 124–133. Association for Computing Machinery, New York, NY, 1996
- [30] Dompierre J., Labbé P., Vallet M., Camarero R. “How to subdivide pyramids, prisms and hexahedra into tetrahedra.” *Proceedings of the 8th International Meshing Roundtable*, pp. 195–204. Sandia National Laboratories, Albuquerque, NM, 1999
- [31] Shontz S. “www.cam.cornell.edu/~shontz/heart.html.” 2003