Available online at www.sciencedirect.com

SCIENCE DIRECT®

Finite Elements in Analysis and Design 42 (2006) 1150–1164

FINITE ELEMENTS IN ANALYSIS AND DESIGN

www.elsevier.com/locate/finel

An angle-based optimization approach for 2D finite element mesh smoothing

Hongtao Xu, Timothy S. Newman*

Department of Computer Science, University of Alabama in Huntsville, Huntsville, AL 35899, USA

Received 15 June 2004; received in revised form 3 October 2005; accepted 8 January 2006
Available online 9 June 2006

Abstract

A new mesh smoothing algorithm that can improve poor-quality meshes, such as meshes with badly shaped elements, is presented. Such meshes are problematic for finite element analysis since the presence of poorly formed mesh elements can reduce the accuracy of the analysis. The new algorithm introduced here improves mesh quality by adjusting the position of the mesh's internal nodes based on optimization of a torsion spring system. The Gauss–Newton method is used to optimize this spring system's objective function to obtain the optimal location of each internal node. Demonstration of the improvement offered by application of the algorithm to real meshes is also made.

Keywords: Finite element meshes; Badly shaped elements; Mesh quality; Optimization

1. Introduction

Finite element (FE) analysis acts on mesh elements that are usually generated by applying mesh generation algorithms. To obtain accurate FE analysis results, the mesh must be of a high quality. Specifically, it must be valid (i.e., have valid elements), have no badly shaped elements, and must conform to the boundary of the given domain. For triangle meshes, any element with a width much smaller than its length is badly shaped (thus, an element is badly shaped if it has at least one small angle [1]). The two types of badly shaped triangle elements are known as daggers and blades [1] and are shown in Fig. 1. If adjacent elements vary greatly in size, one of them or a nearby element will be badly shaped. Low-quality meshes can exhibit sub-par stability in the FE analysis computations [2]. In this paper, we focus on mesh quality improvement via smoothing. Other problems, for example, control of mesh density variation for meshes with large grading, are also important and have typically been addressed separately (e.g., [3,4]).

Meshes generated by mesh generation algorithms can often be optimized with a mesh smoothing algorithm, which relocates nodes to improve mesh quality. Mesh smoothing is often done in an iterative process that does not change element connectivity. One popular 2D mesh smoothing algorithm is the Laplacian smoothing algorithm [5] (which we discuss in detail in Section 2.2). One alternate approach is Zhou and Shimada's [6] torsion spring-based smoothing algorithm (which we discuss in detail in Section 2.2). Its advantage over the Laplacian approach is that it tends to better avoid generation of badly shaped elements. However, it does not optimally utilize the torsion spring system that is its basis.

In this paper, we present an optimization approach that smoothes a mesh of triangular elements in a manner that is provably optimal for the torsion spring system. By provably optimal, we mean our method produces a result guaranteed to be within any desired tolerance for the given formulation.

This paper is organized as follows. Section 2 considers related work. Section 3 derives the problem formulation. Section 4 introduces the new algorithm. Section 5 presents results and an evaluation of the algorithm. Section 6 presents the conclusion.

2. Previous work and motivations

A number of mesh smoothing methods have been presented previously (e.g., [5,7–12]). These methods include approaches for producing acceptable quality meshes by solving a optimization problem (e.g., as in [7,8,13–15]), by minimizing a distortion metric (e.g., as in [9]), by disconnecting invalid elements

* Corresponding author.
E-mail address: tnewman@cs.uah.edu (T.S. Newman).

0168-874X/$ - see front matter © 2006 Elsevier B.V. All rights reserved.
doi:10.1016/j.finel.2006.01.016

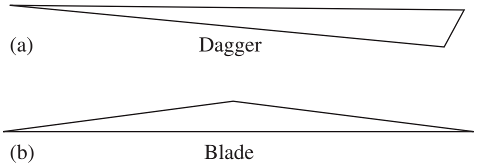


Fig. 1. Badly shaped element types.

from the rest of the mesh (e.g., as in [10]), by solving a generalized linear programming problem (e.g., as in [11], which shows how the optimization techniques in [14,15] can be converted to a linear programming problem), or by combined strategies (e.g., as in [5,12]). Some popular smoothing strategies include Laplacian smoothing [16], Zhou and Shimada's smoothing [6], and optimal Delaunay triangulation-based smoothing [17]. Statistical approaches have also been used (e.g., [18]). The Laplacian and Zhou and Shimada works are discussed in more detail later in this section.

The method introduced here, like [14,15], is an optimization-based approach. However, in [14,15] (as well as in [11]), quality measures are used to guide the process. Here, a torsion spring method guides the process.

Measures to assess mesh quality have also been considered. Such measures can be useful in assessing the relative goodness of a mesh smoothing approach. Readers interested in this topic may consult the comparative study of Philippe and Baker [19], which considers triangle quality measures of extreme angles, radii-ratio, edge ratio, edge to circumradius length ratio, edge to in-radius length ratio, and matrix norms. One popular metric for assessing mesh quality is the minimum angle of any mesh element [5,6].

It should be noted that since FE mesh smoothing is used to improve the quality of finite element analysis (FEA) results, another way to assess mesh quality is to examine the relationship of mesh quality with the FEA results. Shewchuk [20] has studied this relationship. His work showed that the conditioning of the stiffness matrices depends on element shape. It also stated the connection between mesh geometry, interpolation error, and stiffness matrix conditioning and expressed these relationships with error bounds and element quality measures that determine triangle element fitness for achieving low condition numbers. He concluded that elements with smaller angles give worse matrix conditioning. Thus, his findings support using of the minimum angle as a key metric to assess mesh quality; minimum angle can be used as a proxy for FEA error checking.

It should also be noted that FEA solution-based error estimates are often unavailable or unreliable [21], which makes use of mesh quality metrics attractive predictors of FEA error. Berzins [21] has suggested that since the knowledge of how to make such error estimates for a particular problem has typically lagged the problem itself, user-supplied mesh quality measures are suitable proxies to FE analysis error estimates. Hence a primary focus in this paper is mesh quality evaluation using mesh quality measures.

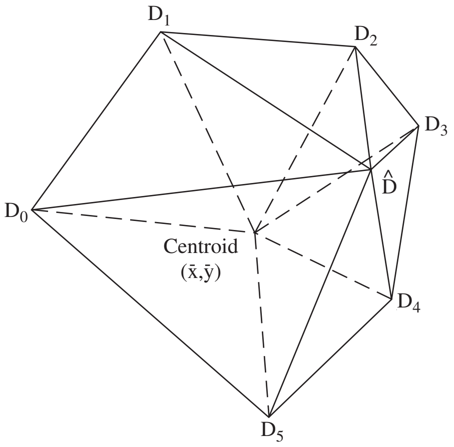


Fig. 2. Illustration of Laplacian smoothing in mesh with internal node $\hat{D}$.

2.1. Laplacian smoothing algorithm

The Laplacian smoothing algorithm [16] is a popular iterative smoothing method. It is widely used due to its simplicity in application and its usual production of satisfactory results. However, it can generate badly shaped or invalid elements in some cases. It attempts to improve mesh quality by moving internal mesh nodes in one or more smoothing passes. In each pass, it moves each internal node (e.g., the node $\hat{D}$ in Fig. 2) to the centroid of the polygon about the internal node. In 2D meshes, the new coordinate $(\bar{x}, \bar{y})$ of the internal node is thus

$$(\bar{x}, \bar{y}) = \left(\frac{1}{n} \sum_{i=0}^{n-1} x_i, \frac{1}{n} \sum_{i=0}^{n-1} y_i \right), \tag{1}$$

where n is the number of nodes connected to the internal node by an edge and (x_i, y_i) is the location of the ith node in the set of n nodes.

For example, for the mesh shown in Fig. 2, internal node $\hat{D}$ will be moved to $(\bar{x}, \bar{y})$, which is the centroid of Polygon $D_0D_1D_2D_3D_4D_5$ about the internal node. In this figure, the original mesh is shown as solid lines and the new mesh that results from moving $\hat{D}$ is shown as dashed lines. In this case, moving $\hat{D}$ to $(\bar{x}, \bar{y})$ makes the new mesh more regular than the original mesh (e.g., the shape of element $\hat{D}D_3D_4$ is improved).

If the edges that connect a set of nodes D_i connected to an internal node are viewed as translational springs, it is easy to prove that minimizing the energy of the translational spring system leads to Eq. (1) [6]. The energy E of such a translational spring system [6] is

$$E = \sum_{i=0}^{n-1} \frac{1}{2} k_D L_i^2, \tag{2}$$

where k_D is a constant, L_i is the displacement of the ith translational spring and n is the number of nodes that are connected to the internal node by an edge.

2.2. Zhou and Shimada's smoothing algorithm

Zhou and Shimada [6] have presented a physically based algorithm that accomplishes smoothing by moving each internal node to a better location based on modeling the nodes as a torsion spring system. The approach seeks to make the local shape properties of adjacent elements more similar. Like the Laplacian approach, it can be implemented easily. It tends to produce better meshes than does the Laplacian approach [6]. The torsion spring system model is attractive for smoothing because it may overcome shortcomings that arise from use of a translation spring system, such as in Laplacian smoothing. In this section, the formulation and the steps of the algorithm are presented.

2.2.1. Formulation

A *torsion spring system* is a system that is composed of springs, each of which can only be applied with torsional force. A *torsional force* is a turning force that causes an object to twist. If the edges that connect a set of nodes D_i connected to an internal node are viewed as a torsion spring system, then this torsion spring system's energy can be expressed as follows:

$$E = \sum_{i=0}^{2n-1} \frac{1}{2} k_T \theta_i^2, \tag{3}$$

where n is the number of nodes that are connected to the internal node by an edge, k_T is a constant and each connected node D_j has associated angles θ_{2j} and θ_{2j+1}, where θ_i is the angle formed by the line from the internal node to $D_{\lfloor i/2 \rfloor}$ and the polygon edge between $D_{(\lceil i/2 \rceil -1) \bmod n}$ and $D_{\lceil i/2 \rceil \bmod n}$.

2.2.2. Algorithm steps

Zhou and Shimada's smoothing algorithm follows a four-step process to calculate the new location of each internal node. These steps are as follows.

Step 1: Two angles, θ_{2j} and θ_{2j+1}, are calculated at one node D_j connected to an internal node $\hat{D}$. These angles are defined in Eqs. (4) and (5):

$$\theta_{2j} = \cos^{-1}\left(\frac{v_j \cdot v_{0,j}}{\|v_j\| \| v_{0,j}\|}\right), \tag{4}$$

$$\theta_{2j+1} = \cos^{-1}\left(\frac{v_j \cdot v_{1,j}}{\|v_j\| \|v_{1,j}\|}\right), \tag{5}$$

where the vectors $v_{0,j}, v_j, v_{1,j}$ originate from node D_j and connect D_j to other (adjacent) nodes (v_j connects D_j to the internal node $\hat{D}$) (Fig. 3).

Step 2: Angle β_j is calculated, as defined in Eq. (6):

$$\beta_j = (\theta_{2j+1} - \theta_{2j})/2. \tag{6}$$

Step 3: The edge v_j is rotated by βj about node D_j. This rotation relocates the edge v_j to the bisectional line of the internal angle at the node D_j.

Step 4: Similarly, for each other node connected to the internal node $\hat{D}$, the new position of the node can be found by applying Steps 1–3. The result is a set of new node positions. By taking the average position of the set of these new positions, the location $(\bar{x}, \bar{y})$ of the new node $\hat{D}'$ is found, as expressed in Eq. (7). This position replaces the node $\hat{D}$'s original position; $\hat{D}$ is moved to $(\bar{x}, \bar{y})$. The new position's coordinates are

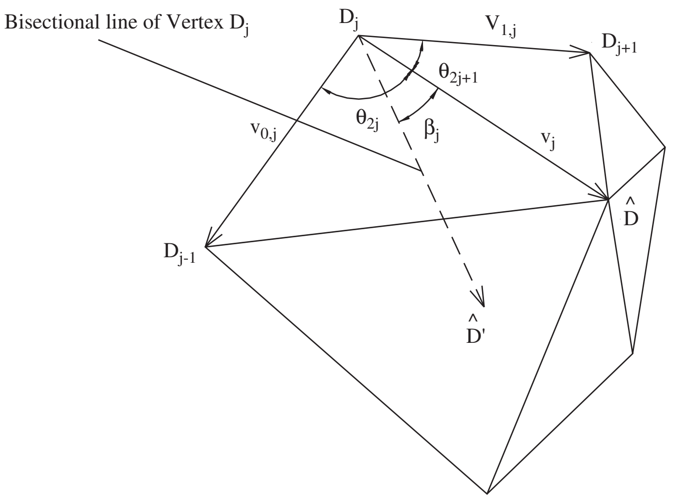


Fig. 3. New position $\hat{D}'$ of internal node with old position $\hat{D}$ in Zhou and Shimada's smoothing.

$$(\bar{x}, \bar{y}) = \left(\frac{1}{n}\sum_{j=0}^{n-1} x'_j, \frac{1}{n}\sum_{j=0}^{n-1} y'_j\right), \tag{7}$$

where (x'_j, y'_j) is the node location that is reached by the rotation of the edge v_j, $(\bar{x}, \bar{y})$ is the new location of the node $\hat{D}$ and n is the number of nodes linked to the internal node.

3. Problem formulation

Although the Zhou and Shimada smoothing algorithm described in Section 2.2 can typically produce better quality meshes than Laplacian smoothing, especially for meshes that contain badly shaped finite elements, the Zhou and Shimada algorithm directly uses a torsion spring system only in its initial processing; after minimizing the system's energy, it uses a heuristic to relocate the interior nodes of the mesh. However, it is possible to use an optimization approach rather than a heuristic to optimally relocate each internal node. In this paper, an optimization approach that accurately minimizes the energy of the torsion spring system to produce a well-smoothed mesh is presented. The approach better optimizes the new locations of all internal nodes.

3.1. Optimal angle derivation

The energy E of the torsion spring system that was shown in Eq. (3) can also be expressed in the following form for a

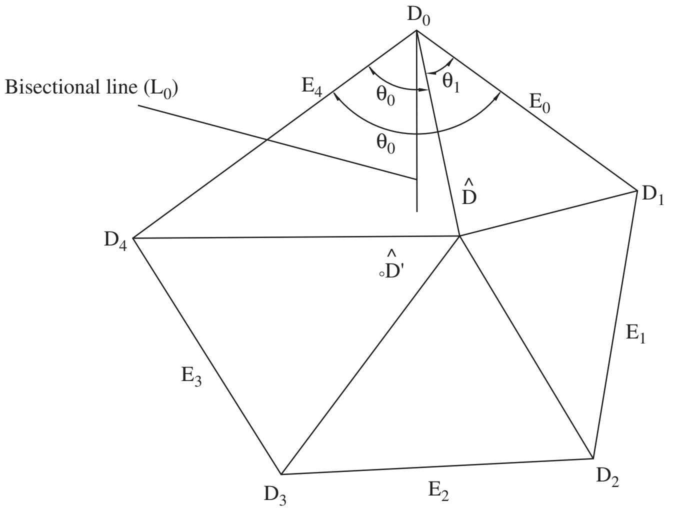


Fig. 4. A polygon about an internal node $\hat{D}$ illustrating torsion spring system parameters.

polygon P about an internal node $\hat{D}$:

$$E=\sum_{i=0}^{n-1}\frac{1}{2}k\left(\theta_{2i}^2+\theta_{2i+1}^2\right), \tag{8}$$

where k is a constant, n is the number of vertices in P, θ_{2i} is the angle between polygon edge $E_{(i-1) \bmod n}$ and the segment linking the internal node $\hat{D}$ with node D_i and θ_{2i+1} is the angle between polygon edge E_i and the segment linking the internal node D with node D_i. (Fig. 4 illustrates these measures for a polygon P about an internal node $\hat{D}$).

For easy notation, we have assumed that the two angles (as shown in Fig. 4) at a node D_i are θ_{2i} and θ_{2i+1}, and their summation is Θ_i. Thus, we have

$$\Theta_i=\theta_{2i}+\theta_{2i+1}. \tag{9}$$

By using the constraint

$$\theta_{2i+1}=\Theta_i-\theta_{2i}, \tag{10}$$

obtained from Eq. (9), and substituting Eq. (10) into Eq. (8), we have

$$E=\sum_{i=0}^{n-1}\frac{1}{2}k\left[\theta_{2i}^2+(\Theta_i-\theta_{2i})^2\right], \tag{11}$$

where θ_{2i} is a variable and Θ_i is a constant. By taking the partial derivative of E with respect to θ_{2i}, we get

$$\frac{\partial E}{\partial\theta_{2i}}=k(2\theta_{2i}-\Theta_i) \quad \forall i=0,1,\ldots,n-1. \tag{12}$$

To minimize E, we should have

$$\frac{\partial E}{\partial\theta_{2i}}=0 \quad \forall i=0,1,\ldots,n-1. \tag{13}$$

Thus, at E's critical points

$$2\theta_{2i}-\Theta_i=0 \quad \forall i=0,1,\ldots,n-1. \tag{14}$$

From Eq. (14), we also have the result

$$\theta_{2i}=\frac{1}{2}\Theta_i \quad \forall i=0,1,\ldots,n-1. \tag{15}$$

Thus, since E is a concave-down parabola, E is minimal when $\theta_{2i}=\theta_{2i+1}$.

3.2. *Objective function for optimal angle*

From the derivation in the previous section, it is known that the energy of a torsion spring system is minimal when $\theta_{2i}=\theta_{2i+1}$. Based on this minimization result, we might expect that drawing a bisectional line for each node of a polygon P about an internal node, then getting the point N where all of these bisectional lines intersect would produce an optimal new location N of the internal node. Unfortunately, these bisectional lines seldom intersect at the same point. The averaging method used by Zhou and Shimada to generate the new location of the internal node produces a reasonable approximation. The new location can be computed in a manner more consistent with the torsion spring system model, however, which we introduce next.

To find the optimal new locations, our approach uses the following objective function s:

$$s=\sum_{i=0}^{n-1}\left[distance(\hat{D}',L_i)\right]^2, \tag{16}$$

where L_i is the bisectional line of the internal angle at node D_i, D_i is one node of the polygon about an internal node $\hat{D}$, and $\hat{D}'$ is the new (i.e., optimized) position of the internal node. (Fig. 4 contains an illustration of these parameters).

We can define $t=(p,q)$ as the coordinate of an internal node with scalars p and q as the coordinate components. Given such a notation, since $distance(\hat{D}',L_i)$ is a function of the coordinates of an internal node, the distance can be expressed as a function f_i of the coordinate t, allowing the objective function s in Eq. (16) to be re-written as follows:

$$s=\sum_{i=0}^{n-1}f_i(t)^2. \tag{17}$$

4. Angle-based optimization smoothing

In this section, a least-squares formulation for the objective function s is given and a solution for this formulation is derived. The solution leads to our algorithm for mesh smoothing.

4.1. *Least-squares formulation*

The least-squares formulation for an objective function s is

$$\min\ s(t)=\tfrac{1}{2}f(t)^{\mathrm{T}}f(t), \tag{18}$$

where $t=(t_1,t_2,\ldots,t_n)$ is a vector with n components, $f(t)$ is a column vector of m functions $f_i(t)$ and $f(t)^{\mathrm{T}}$ is the row vector $f(t)^{\mathrm{T}}=(f_1(t),\ldots,f_m(t))$.

The formulation of Eq. (18) can be solved by any optimization that minimizes the objective function $s(t)$. When $m = n$, this problem becomes a set of non-linear equations in the following form:

$$f(t) = 0. \tag{19}$$

4.2. Derivation of linear equation system

By taking the derivative of s in Eq. (18), the following equation can be obtained:

$$\frac{\partial s}{\partial t} = \sum_{i=1}^{n} f_i \frac{\partial f_i}{\partial t} = \frac{\partial f^{\mathrm{T}}}{\partial t} f. \tag{20}$$

We assume that

$$J^{\mathrm{T}} = \frac{\partial f^{\mathrm{T}}}{\partial t}, \tag{21}$$

where J^{T} is the transpose of the Jacobi matrix.

The Jacobi matrix can be expanded as follows:

$$J = \begin{pmatrix} \dfrac{\partial f_1}{\partial t_1} & \dfrac{\partial f_1}{\partial t_2} & \cdots & \dfrac{\partial f_1}{\partial t_n} \\ \dfrac{\partial f_2}{\partial t_1} & \dfrac{\partial f_2}{\partial t_2} & \cdots & \dfrac{\partial f_2}{\partial t_n} \\ \vdots & \vdots & \vdots & \vdots \\ \dfrac{\partial f_m}{\partial t_1} & \dfrac{\partial f_m}{\partial t_2} & \cdots & \dfrac{\partial f_m}{\partial t_n} \end{pmatrix}. \tag{22}$$

We assume that

$$P = \frac{\partial s}{\partial t}, \tag{23}$$

or, equivalently, that

$$P = \begin{pmatrix} \dfrac{\partial s}{\partial t_1} \\ \dfrac{\partial s}{\partial t_2} \\ \vdots \\ \dfrac{\partial s}{\partial t_n} \end{pmatrix}. \tag{24}$$

From Eqs. (20), (21) and (23), we have

$$P = J^{\mathrm{T}} f. \tag{25}$$

To minimize the objective function in Eq. (18), we should have

$$P = 0. \tag{26}$$

By expanding P at point $t = t^n$ with a Taylor series that deletes derivatives of second and above order, we obtain

$$P(t) = P(t^n) + \left.\frac{\partial P}{\partial t}\right|_{t^n} (t - t^n). \tag{27}$$

Then, substituting Eq. (26) into Eq. (27), we have

$$0 = P(t^n) + \left.\frac{\partial P}{\partial t}\right|_{t^n} (t - t^n). \tag{28}$$

From Eq. (28), we obtain

$$t = t^n - \left(\left.\frac{\partial P}{\partial t}\right|_{t^n} \right)^{-1} P(t^n). \tag{29}$$

Clearly,

$$\left.\frac{\partial P}{\partial t}\right|_{t^n} = \left.\frac{\partial (J^{\mathrm{T}} f)}{\partial t}\right|_{t^n}. \tag{30}$$

By deleting the derivatives of second and above order, as is done in Eq. (27), we obtain

$$\left.\frac{\partial P}{\partial t}\right|_{t^n} = J_n^{\mathrm{T}} J_n, \tag{31}$$

where J_n is the value of J at $t = t^n$.

By substituting Eq. (31) and (25) into Eq. (29), and using t^{n+1} as the next step value, the following equation can be obtained:

$$t^{n+1} = t^n - \left(J_n^{\mathrm{T}} J_n \right)^{-1} J_n^{\mathrm{T}} f_n. \tag{32}$$

Next, we define

$$d^n = -\left(J_n^{\mathrm{T}} J_n \right)^{-1} J_n^{\mathrm{T}} f_n, \tag{33}$$

which allows Eq . (32) to be simplified to

$$t^{n+1} = t^n + d^n. \tag{34}$$

In Eq. (34), it can be proved that d^n is always negative. Thus, t^n is always decreasing with the increasing n [22]. Although d^n can be very small, there are still chances that a better point exists in the interval [0, d^n].To further optimize Eq. (34) in the interval, a scalar parameter λ^n can be introduced into the equation. The domain of λ^n is [0, 1] . Introduction of λ^n produces the following equation:

$$t^{n+1} = t^n + \lambda^n d^n. \tag{35}$$

The step in the steepest decreasing direction can be obtained by performing a one-dimensional search of λ^n with Eq. (35). After a specified precision has been reached during the search, the variable vector t is obtained. Substituting the obtained value t into Eq. (18) allows the optimal value s to be found, thus solving the problem.

4.3. Optimization algorithm

Next, we describe how our approach uses the Gauss–Newton optimization to optimize the objective function s. Optimization of s requires finding vector t. To solve Eq. (17), it is necessary to find the optimal λ^n in Eq. (35). Therefore, the following problem needs to be solved:

$$\min\, s(t^n + \lambda^n d^n), \tag{36}$$

where only λ^n is unknown.

To solve this problem, the following steps are used. These steps require the solution's precision ε to be pre-specified. That is

$$\left|\frac{s(t^{n+1}) - s(t^n)}{s(t^n)}\right| < \varepsilon. \tag{37}$$

Step 1: Calculate $s(t^n + d^n)$ and $s(t^n)$.

Step 2: If predefined solution precision ε has been reached or it is iteration $j_{\max}$, go to Step 8.

Step 3: Set $\lambda^n = 1$.

Step 4: If $s(t^n + d^n) < s(t^n)$, set $t^n = t^n + d^n$ and go to Step 1.

Step 5: Assume $s(t^n + \lambda^n d^n)$ with respect to λ^n is quadratic and find the coefficients of the quadratic polynomial using the values of s at $\lambda^n = 0$, 1, and the derivative of s at $\lambda^n = 0$.

Step 6: Find minimum value of $s(t^n + \lambda^n d^n)$ for $0 \leqslant \lambda^n \leqslant 1$.

Step 7: Perform the following series of steps:

(a) If $s(t^n + d^n) < s(t^n)$, go to Step 1.
(b) Set $\lambda^n = \lambda^n/2$.
(c) Set $t^n = t^n + \lambda^n d^n$.
(d) Go to Step 7a.

Step 8: Stop.

In practice, we have used 10 iterations ($j_{\max} = 10$) which appears to lead to solutions that are good enough.

5. Results and discussion

In this section, the qualitative characteristics and computational results of the new smoothing algorithm on triangulated meshes are presented. The algorithm's characteristics and performance are also compared with the Laplacian and Zhou and Shimada smoothing algorithms.

5.1. Algorithm comparison criteria

First, we describe seven metrics useful for quantitatively comparing the quality of meshes produced by the algorithms. Visual assessment of the mesh can also be made, although metrics allow for more consistency in evaluation. Two of these metrics are simple measures from which the other metrics are derived.

The first metric is the *angle ratio*. The angle ratio is the ratio of the minimum angle (of the angles of a triangle element) to the maximum angle (of the angles of the same triangle element). The ideal triangular facet (an equilateral triangle, which has a maximal minimum angle) has an angle ratio of 1.0.

The second metric is the *length ratio*. The length ratio expresses the relative length and positioning of mesh edges. For a triangle with edges that have lengths $l1$, $l2$ and $l3$, and which are distances $h1$, $h2$ and $h3$, respectively, from the vertex opposite the edge, we define the triangle's length ratio as $\min(h1/l1, h2/l2, h3/l3)$. The length ratio is also a shape metric. The ideal triangle (an equilateral triangle) has a length ratio of 0.866.

The third type of metric is the *average ratio* of the whole mesh. We have considered both the *average ratio of angles* as well as the *average ratio of lengths*. The average ratio of angles is the mean of all the mesh's angle ratios. The average ratio of lengths is the mean of all the mesh's length ratios. The average ratio of angles (or lengths) is used to measure the average overall quality of the mesh. A mesh with only ideal triangles (i.e., equilateral triangles) has an average angle ratio of 1.0 and an average length ratio of 0.866. The closer the average ratio is to the ideal value, the better the average mesh element.

The fourth type of metric are *median ratios* of the whole mesh. We have considered the *median angle ratio*, which is the median of all the angle ratios for the triangles that exist in the mesh, and the *median length ratio*, which is defined analogously for the length ratio. The median ratio expresses the quality of a typical mesh element, and ideally the median angle ratio would be 1.0 and the median length ratio would be 0.866. To conserve space, we report median ratios for only a few of the cases.

The fifth type of metric considers the fifth percentile of the mesh's angle and length ratios. This metric considers the goodness of the worst (i.e., lowest quality) elements of the mesh. A mesh of ideal triangles (equilateral triangles) would have a fifth percentile of the angle ratio of 1.0 and a fifth percentile of the length ratio of 0.866.

The sixth metric is the *minimum angle* of the whole mesh, which is a key quality metric, since it has been widely used by others [5,6]. The ideal mesh (composed entirely of equilateral triangles) has a minimum angle of 60°. While this metric does describe the poorness of the worst element, it is not sufficient to use only this metric as a measure of a mesh's overall goodness (since it only describes how badly shaped the very worst element is).

The seventh metric is the *minimum area* of the whole mesh. This metric is used to measure the uniformness of the mesh element size variation.

5.2. Improvement on invalid elements

Next, we report an experiment that considers a mesh that contains invalid elements when smoothed by Laplacian smoothing. This mesh scenario, called Case 1, is illustrated in Fig. 5. Fig. 5(a) shows the original mesh (i.e., the mesh before smoothing) for this scenario, and Figs. 5(b)–(d) show the meshes created by applying the Laplacian smoothing, Zhou and Shimada's smoothing and the new angle-based smoothing, respectively, to the original mesh. The mesh produced by the Laplacian smoothing has several elements that are invalid because they overlap. The mesh produced by the new angle-based smoothing algorithm has no invalid elements and has elements that are more uniform in shape than the mesh generated by Zhou and Shimada's smoothing. The original mesh had three badly shaped elements out of eight total elements, and in the mesh smoothed by the Zhou and Shimada and new smoothings, the mesh elements are much more uniform; there are no badly shaped elements.

The quantitative metrics, average angle ratio, average length ratio, median angle ratio, median length ratio, fifth percentile of the angle ratio, fifth percentile of the length ratio, and the

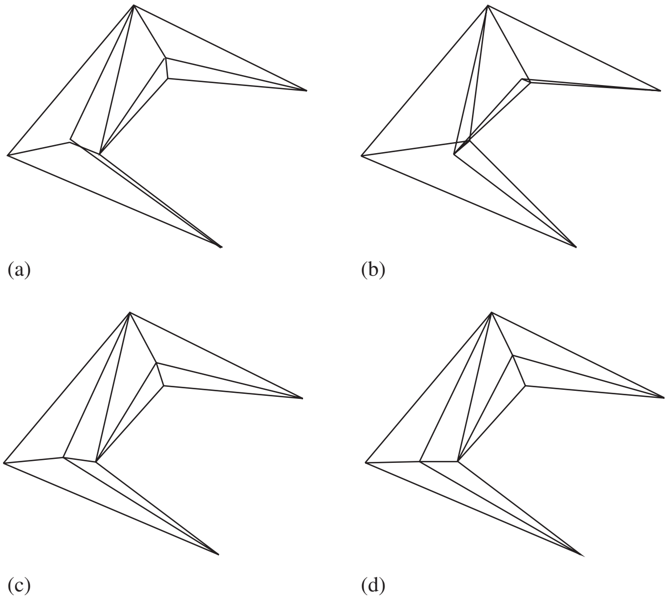


Fig. 5. Case 1 mesh and results of smoothing algorithms applied to it: (a) original mesh; (b) Laplacian smoothing; (c) Zhou and Shimada's smoothing; (d) new smoothing.

Table 1
Metrics for the invalid element experiments (Case 1 Scenario)

Metric name	Original mesh	Z&S	New method
Average angle ratio	0.098	0.095	0.094
Average length ratio	0.163	0.165	0.167
Median angle ratio	0.097	0.097	0.115
Median length ratio	0.170	0.172	0.197
Fifth perc. angle ratio	0.015	0.035	0.047
Fifth perc. length ratio	0.037	0.076	0.097
Minimum angle	2.560	5.180	6.990

minimum angle for the original mesh, the mesh generated with the Laplacian smoothing, the mesh generated with Zhou and Shimada's smoothing (labeled as "Z&S") and for the mesh generated with the new smoothing algorithm are shown in Table 1. Since the Laplacian smoothing generates invalid elements, its metric values have been omitted. Except for the average angle ratio, the metric values for the new smoothing algorithm are all greater than (and closer to the ideal values than) the metric values for the original and the Zhou and Shimada-smoothed meshes; the new algorithm produces a mesh that has better overall shape uniformity. In particular, the much larger minimum angle and fifth percentile values mean that the worst elements have been significantly improved, which is important in enabling the FEA to obtain acceptable accuracy. The reason that the average angle ratio is not really changed by either Zhou and Shimada's algorithm or by the new algorithm may be that there are few elements in the mesh. This small number of elements may make the node relocation difficult due to the limited degrees of freedom in the nodes' positional changes.

5.3. Badly shaped elements for Zhou and Shimada vs. new algorithm

In this section, we report experiments that test the mesh quality of the new smoothing algorithm. Results are compared only to the Zhou and Shimada smoothing since their results [6] and our confirmatory experiments have demonstrated that method's superiority over Laplacian smoothing. Five scenarios are considered. Focus is on determining if the new algorithm tends to better avoid badly shaped elements than does the Zhou and Shimada algorithm. We consider visual inspection of the meshes as well as metrics of mesh quality.

5.3.1. Shape improvement

The first scenario studied, called Case 2, is shown in Fig. 6. Fig. 6(a) shows the original mesh for this scenario, and Figs. 6(b) and 6(c) show the meshes created by applying the Zhou and Shimada smoothing and the new angle-based smoothing, respectively, to the original mesh. The mesh produced by Zhou and Shimada's smoothing in Fig. 6(b) appears to be more uniform in shape than the original mesh. It can also be seen from the figure that the mesh generated with the new smoothing algorithm in Fig. 6(c) is also quite uniform in shape. Fig. 6(c) mesh is actually better than the mesh generated with the Zhou and Shimada's smoothing, as the quantitative measures confirm.

The quantitative metrics for the meshes formed for the Case 2 scenario are shown in Table 2, which presents clear evidence of the new algorithm's advantage. In this table, the average angle ratio, average length ratio, median angle ratio, median length ratio, fifth percentile of the angle ratio, fifth percentile of the length ratio, and minimum angle of the original mesh, the mesh generated by the Zhou and Shimada's smoothing, and the mesh generated by the new smoothing algorithm are shown. The metric values for the Zhou and Shimada's smoothing are all greater than the metric values for the original mesh, which means that the overall mesh uniformity in shape is improved by the Zhou and Shimada's smoothing, although the smaller minimum angle and fifth percentile of the angle ratio suggests that the Zhou and Shimada smoothing algorithm's worst elements are more badly shaped than in the original mesh.

The new smoothing algorithm's metric values are all greater than the values of the metrics for the Zhou and Shimada's smoothing, which means that the new algorithm produced a mesh that has improved uniformity in shape. In particular, the much larger fifth percentile values and minimum angle indicate that the worst elements generated by the new method are much better than those generated by the Zhou and Shimada's smoothing.

The second and third scenarios studied, called Cases 3 and 4, respectively, are shown in Figs. 7 and 8. Part (a) of the figures shows the original mesh and parts (b) and (c) shows the meshes created by applying the smoothing methods to the original mesh. The meshes produced by the Zhou and Shimada smoothing appear to be more uniform in shape than the original meshes, and the meshes generated with the new smoothing

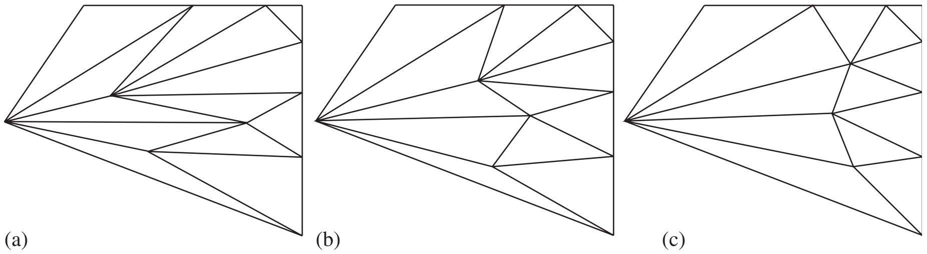


Fig. 6. Case 2 mesh and results of smoothing on it: (a) original mesh; (b) Zhou and Shimada's smoothing; (c) new smoothing.

Table 2
Metrics for Case 2 scenario

Metric name	Original mesh	Z&S	New method
Average angle ratio	0.222	0.255	0.380
Average length ratio	0.268	0.324	0.430
Median angle ratio	0.137	0.217	0.393
Median length ratio	0.207	0.320	0.477
Fifth perc. angle ratio	0.047	0.039	0.067
Fifth perc. length ratio	0.074	0.064	0.124
Minimum angle	7.670	6.400	9.750

algorithm appear to be more uniform in element shape than the meshes generated with the Zhou and Shimada's smoothing. Selected quantitative metrics for Cases 3 and 4 scenarios are shown in Tables 3 and 4, which present clear evidence of this improvement. The metric values for the Zhou and Shimada smoothing are all greater than the metric values for the original mesh. However, the new smoothing algorithm's metric values are all greater than the values of the metrics for the Zhou and Shimada's smoothing, which means that the new algorithm improved overall mesh shape uniformity more than did Zhou and Shimada's smoothing. In particular, the larger fifth percentile values and minimum angle indicate that the worst elements generated by the new method are better than those generated by the Zhou and Shimada's smoothing.

The fourth scenario studied, called Case 5, is shown in Fig. 9. In this figure, (a.1), (b.1), (c.1), and (d.1) show four different triangulations of the region; (a.2), (b.2), (c.2), and (d.2) show the meshes created by applying the Zhou and Shimada smoothing to the original mesh; and (a.3), (b.3), (c.3), and (d.3) show the meshes created by applying the new angle-based smoothing. While the mesh produced by the Zhou and Shimada smoothing is more uniform in shape than the original mesh, the mesh produced by the new smoothing algorithm appears to be more uniform in element shape.

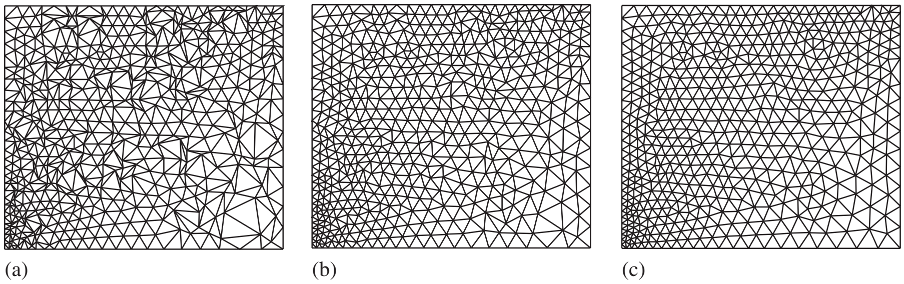


Fig. 7. Case 3 mesh and results of smoothing on it: (a) original mesh; (b) Zhou and Shimada's smoothing; (c) new smoothing.

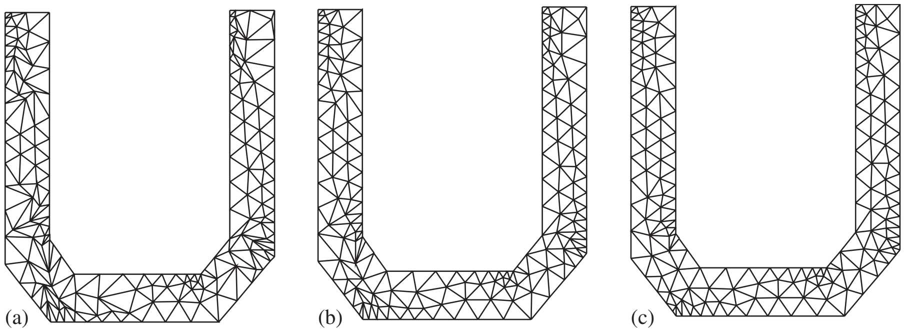


Fig. 8. Case 4 mesh and results of smoothing on it: (a) original mesh; (b) Zhou and Shimada's smoothing; (c) new smoothing.

Table 3
Metrics for Case 3 scenario

Metric name	Original mesh	Z&S	New method
Average angle ratio	0.531	0.680	0.760
Average length ratio	0.552	0.673	0.726
Fifth perc. angle ratio	0.148	0.421	0.544
Fifth perc. length ratio	0.221	0.487	0.580
Minimum angle	6.600	24.180	31.080

Table 4
Metrics for Case 4 scenario

Metric name	Original mesh	Z&S	New method
Average angle ratio	0.478	0.568	0.615
Average length ratio	0.509	0.59	0.626
Fifth perc. angle ratio	0.114	0.252	0.327
Fifth perc. length ratio	0.169	0.332	0.416
Minimum angle	1.500	10.820	18.310

Selected quantitative metrics for the meshes formed for the Case 5 scenario are shown in Table 5. It is evident that the new method is best at improving overall mesh shape uniformity and at improving the worst elements.

For the fourth scenario, an optimal initial mesh was also considered. The original mesh and generated meshes are shown in Fig. 10. All metrics have the same (optimal) value (1) for the original mesh and for the smoothed meshes for this case; all meshes are optimal.

The fifth scenario studied, called Case 6, is a relatively good large mesh of Lake Superior. We have used two variants on this case—a subregion (with 986 elements, shown in Fig. 11) and the whole mesh (with 4378 elements, shown in Fig. 12). For a more complete comparison, we also computed the Laplacian smoothing results and have displayed a zoomed-in view of a sub-region containing badly shaped elements under Laplacian smoothing. In these figures, (a) shows the original mesh, (b) shows the Laplacian smoothed mesh, (c) shows the Zhou and Shimada smoothed mesh, and (d) shows the mesh created by applying the new angle-based smoothing. Near the top of the zoomed part, there is a very badly shaped element for the Laplacian case.

Selected quantitative metrics for the meshes formed for the Case 6 scenario are shown in Tables 6 and 7. The worst angle ratio is improved more by the new smoothing than it is by Zhou and Shimada smoothing while the other metrics are about the same.

5.3.2. Size improvement

Next, element sizes for the meshes produced by Zhou and Shimada's smoothing and the new smoothing are considered. We use minimum element areas as the measure of size uniformity, as was used in [6]. For easy comparison, the *normalized area*, which is the element's area divided by some specified element's area is used. The specified area is the area of an element in the original mesh. Specifically, the minimum normalized area is the ratio of the area of the mesh's minimum-sized element to the area of the original mesh's minimum-sized element. Table 8 shows, for the six scenarios discussed earlier in this paper, the minimum normalized element areas in the original mesh, Zhou and Shimada-smoothed mesh, and in the mesh smoothed by the new method. The minimum normalized areas for meshes produced by the new method are all at least as large as those for the Zhou and Shimada-smoothed meshes. Thus, the meshes generated by the new smoothing algorithm tend to have less extreme very small or very large elements; maximum variation in size tends to be smaller for the new smoothing algorithm.

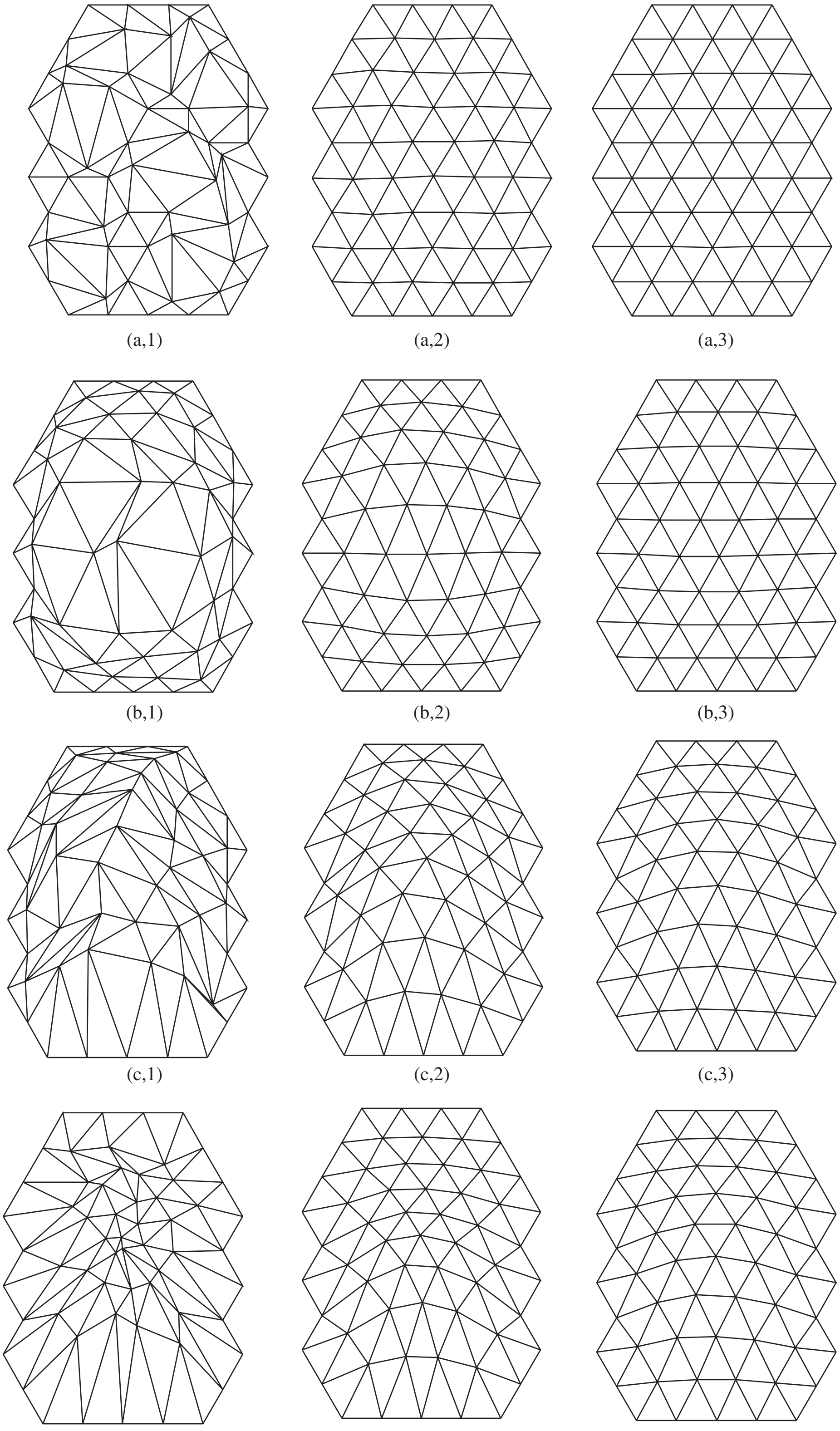
While our examples demonstrate that the Zhou and Shimada's smoothing can improve mesh quality, it can be further seen that the new smoothing algorithm can generate a mesh with less extreme variation in element size. In all the experiments conducted, applying one iteration of the new method to the original mesh produced a better mesh than did applying one iteration of the Zhou and Shimada smoothing.

5.4. Convergence issues

Often, optimization-based smoothing algorithms are applied in a sequence of passes, where each pass smoothes the last pass's result, in the hope of getting better results (e.g., [6]). Often, this process can improve results, although usually only for a few passes, after which results deteriorate. This deterioration is a result of the optimization approaches acting locally; when a sub-mesh is smoothed to an optimal local condition, the whole mesh has not been optimized globally. Next an example that illustrates this point is shown. The example considers the Case 3 mesh. Key mesh quality metrics are reported in Table 9. In the table, the normalized areas use as their specified element the element of the one-iteration mesh which has the minimum angle. Although the minimum normalized element areas for the meshes from both the Zhou and Shimada and new smoothings are similar, the maximum normalized element areas in the mesh generated with the new smoothing are larger. Thus, as more smoothing iterations are applied, the mesh generated with the new smoothing tends to become uniform in size, but the rate of size improvement is slower for the new algorithm than it is for Zhou and Shimada's smoothing. The best smoothing achieved by Zhou and Shimada's smoothing was not as good as the best smoothing achieved by the new smoothing. On the other hand, as the number of iterations increases, the minimum angle of the new smoothing gets worse—the best results were given by applying one iteration. For Zhou and Shimada's smoothing, results improved up to six iterations, then got worse. The new smoothing tends to give the best results on the first iteration and this result is better than the one given by Zhou and Shimada's smoothing when one iteration of smoothing is performed.

5.5. Effect of badly shaped element on finite element analysis results

Next, two experiments that consider the effect of badly shaped elements on finite element analysis results of the


(a,1) (a,2) (a,3)

(b,1) (b,2) (b,3)

(c,1) (c,2) (c,3)

(d,1) (d,2) (d,3)

Fig. 9. Case 5 mesh and results of smoothing on different initial tessellations (first column: original; second column: Zhou and Shimada's smoothing; third column: new smoothing).

Table 5
Metrics for Case 5 scenario

	Metric name	Original mesh	Z&S	New method
Experiment 1	Worst length ratio	0.315	0.872	0.988
	Average angle ratio	0.402	0.931	0.992
	Worst angle ratio	0.073	0.798	0.981
	Minimum angle	10.948	53.328	59.357
Experiment 2	Worst length ratio	0.290	0.721	0.908
	Average angle ratio	0.340	0.738	0.928
	Worst angle ratio	0.071	0.564	0.851
	Minimum angle	10.642	42.816	54.716
Experiment 3	Worst length ratio	0.210	0.341	0.680
	Average angle ratio	0.268	0.499	0.722
	Worst angle ratio	0.019	0.243	0.546
	Minimum angle	3.233	19.735	40.969
Experiment 4	Worst length ratio	0.195	0.454	0.707
	Average angle ratio	0.388	0.599	0.747
	Worst angle ratio	0.094	0.337	0.575
	Minimum angle	10.954	26.324	42.810

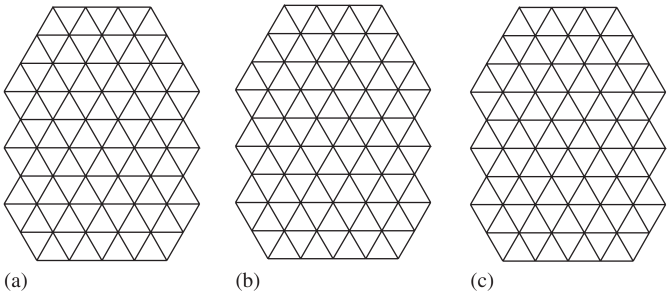


Fig. 10. Case 5 mesh with optimal initialization smoothing on it.

Laplace's equation, $\Delta u = 0$, are reported. These experiments are evaluated on three error norms: $\|A\|_1$, $\|A\|_2$, $\|A\|_\infty$, which for square matrices A are: $\|A\|_1 = \max_j \sum_{i=1}^{n} |a_{ij}|$, $\|A\|_2 = (\text{max eigenvalue of } A_H A)^{1/2}$, and $\|A\|_\infty = \max_i \sum_{j=1}^{n} |a_{ij}|$, where $\|A\|_H$ is conjugate transpose.

The first experiment applies Laplace's equation on a rectangular domain with a Dirichlet boundary condition. We used a good and bad mesh of the domain, as shown in Fig. 13. The domain's boundary condition is that on boundary segment cd (shown in Fig. 13), $u=100$. For all other boundary points, $u=0$.

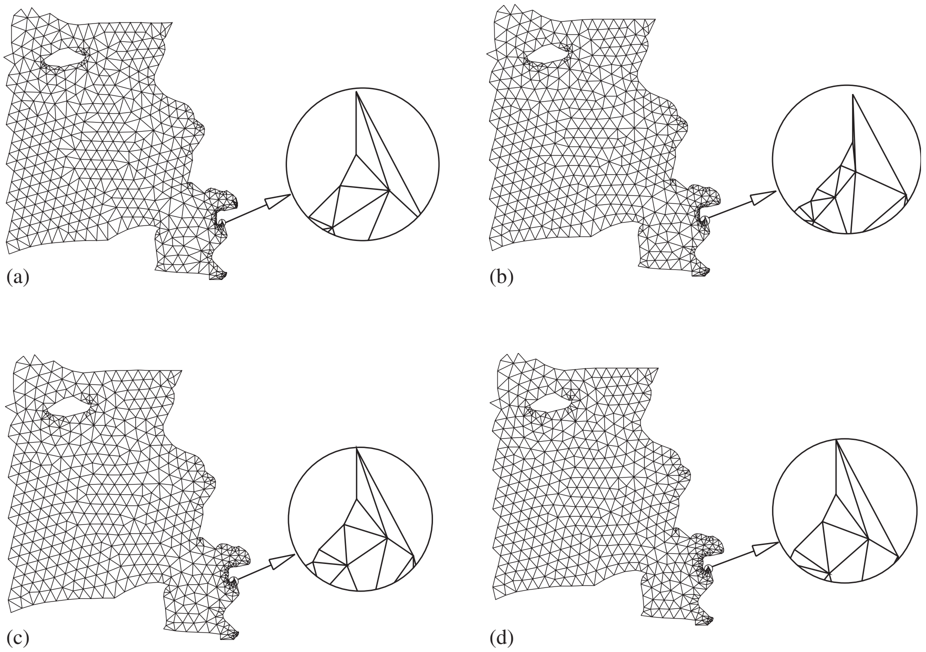


Fig. 11. Case 6 smoothing of partial Lake Superior mesh: (a) original; (b) Laplacian; (c) Zhou and Shimada; (d) new angle-based smoothing.

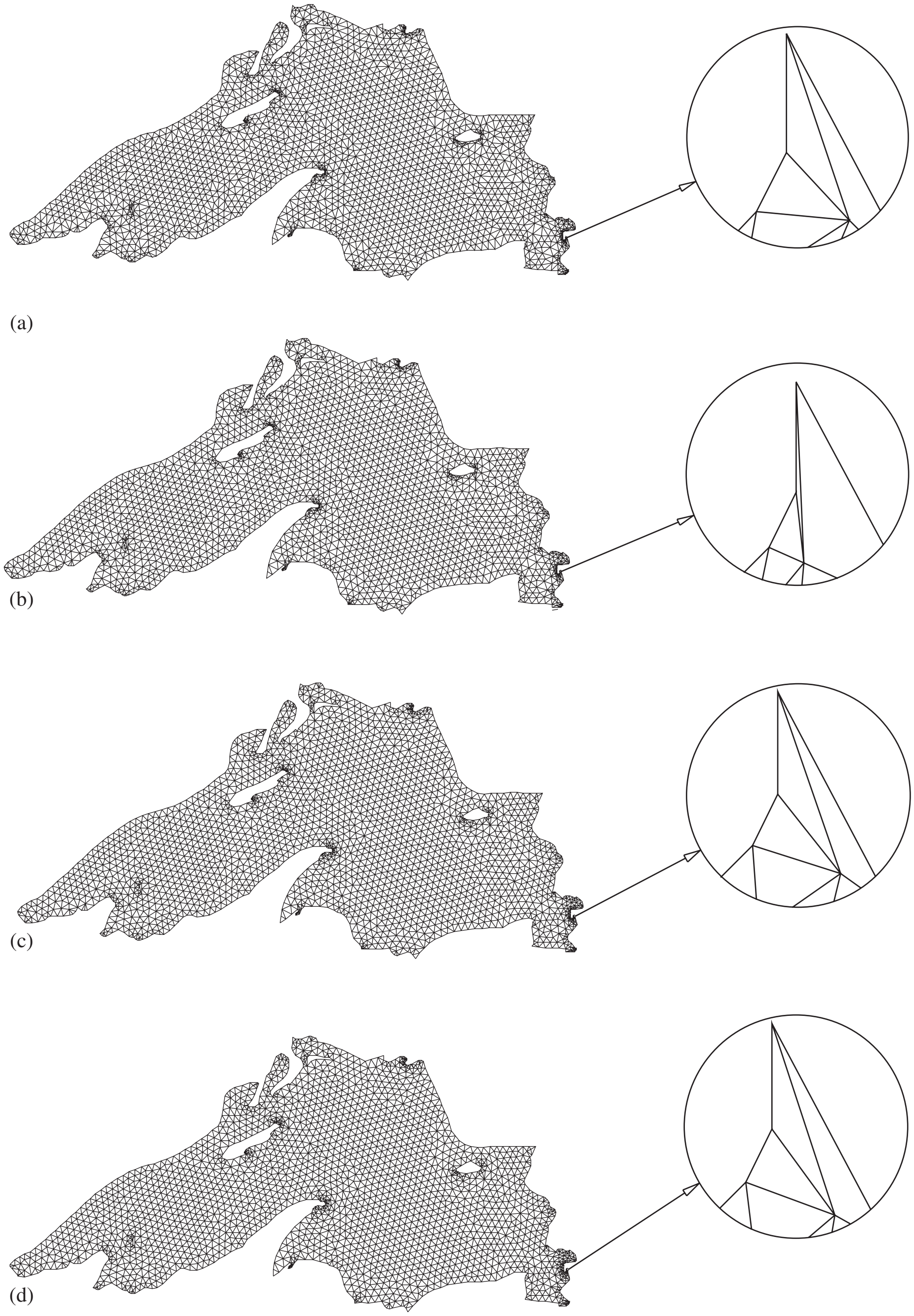


Fig. 12. Case 6 smoothing of full Lake Superior mesh: (a) original mesh and highlighted area; (b) Laplacian; (c) Zhou and Shimada; and (d) new angle-based smoothing to highlighted area.

Table 6
Metrics for Case 6 (partial) scenario

Metric name	Original mesh	Laplacian	Z&S	New method
Worst length ratio	0.210	0.204	0.224	0.220
Average angle ratio	0.708	0.721	0.704	0.694
Worst angle ratio	0.061	0.009	0.062	0.071
Minimum angle	9.088	1.545	8.943	9.876

Table 7
Metrics for Case 6 (full) scenario

Metric name	Original mesh	Laplacian	Z&S	New method
Worst length ratio	0.210	0.204	0.204	0.220
Average angle ratio	0.770	0.786	0.773	0.766
Worst angle ratio	0.061	0.009	0.062	0.070
Minimum angle	9.088	1.545	8.926	9.760

A high-precision FEA result is shown in Fig. 14. The error in FEA solution on the good mesh relative to the high-precision solution is plotted in Fig. 15. The error in FEA solution on the bad mesh relative to the high-precision solution is plotted in Fig. 16. Table 10 shows the error 1-norm, 2-norm, and infinity-norm values for those meshes. The plots of errors and the error

Table 8
Element size improvement measured with minimum normalized area

		Original mesh	Z&S	New method
Case 1	Min. norm. area	1	1.676	2.391
Case 2	Min. norm. area	1	1.000	1.000
Case 3	Min. norm. area	1	1.597	1.677
Case 4	Min. norm. area	1	10.252	10.252
Case 5	Min. norm. area	1	4.494	4.809
	Min. norm. area	1	2.689	3.777
	Min. norm. area	1	4.044	6.733
	Min. norm. area	1	10.375	14.083
	Min. norm. area	1	1.000	1.000
Case 6	Min. norm. area	1	1.571	1.857
	Min. norm. area	1	1.167	1.333

Table 9
Effect of the number of iterations on smoothing, Case 3 mesh

# iterations	Z&S			New method		
	Min. angle	Min. norm. area	Max. norm. area	Min. angle	Min. norm. area	Max. norm. area
1	24.180	1.000	1.000	31.080	1.000	1.000
2	29.019	1.090	0.747	29.590	1.076	1.030
3	30.110	1.106	0.644	29.100	1.076	1.027
4	30.900	1.129	0.614	28.720	1.076	1.023
5	31.430	1.129	0.619	28.500	1.076	1.020
6	31.550	1.129	0.622	28.310	1.076	1.016
7	31.030	1.129	0.623	28.140	1.076	1.014
10	29.670	1.129	0.623	27.690	1.076	1.007

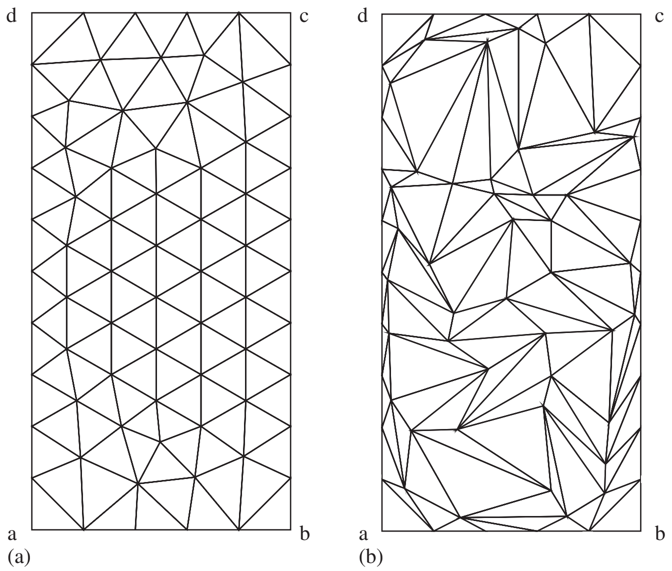


Fig. 13. Rectangle domain with meshes and boundary conditions: (a) good mesh; (b) bad mesh.

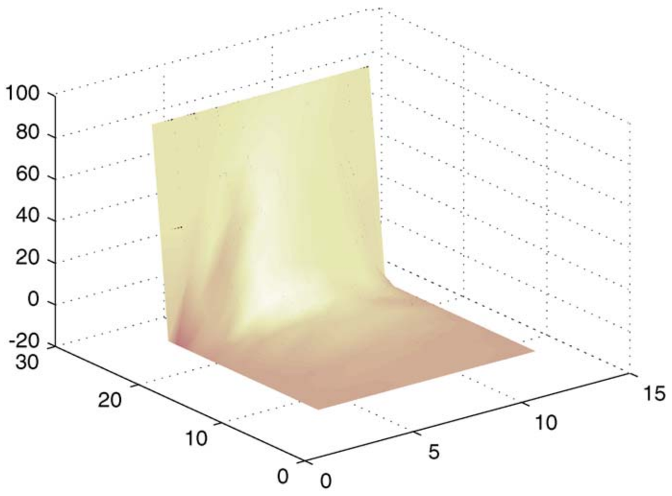


Fig. 14. High-precision solution of Laplace's equation on rectangle domain.

norms suggest the improvement in FEA quality from use of a higher quality mesh.

Finally, a similar experiment was performed on the partial Lake Superior domain after smoothing by Laplacian smoothing and the new method. The meshing results are shown in Fig. 17. The FEA Dirichlet boundary condition has $u = 100$ on boundary segment *abc* (this segment is shown in the circled region of Fig. 17) and $u = 0$ for all other boundaries. The solution error norms for the smoothed meshes are shown in Table 10. The mesh produced by the new method exhibits the lowest error.

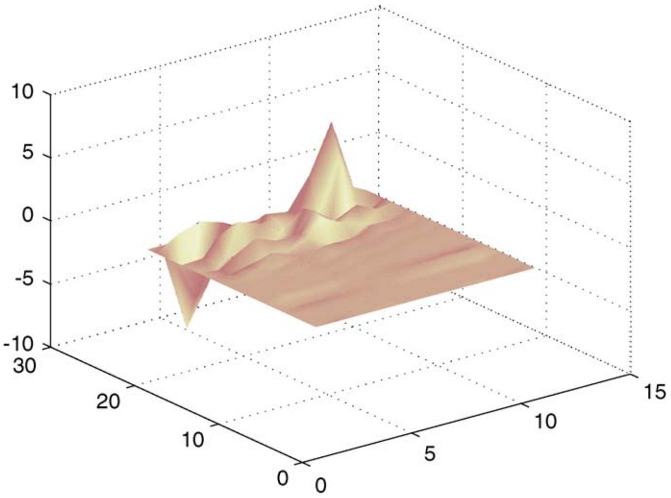


Fig. 15. Solution error of Laplace's equation on the good mesh.

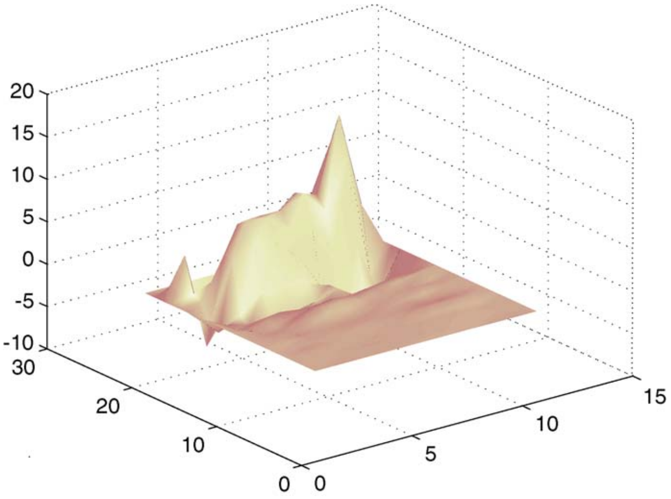


Fig. 16. Solution error of Laplace's equation on the bad mesh.

Table 10
Error norm values for meshes of domains

		1-norm	2-norm	Infinity-norm
Rectangle domain	Good mesh	13.691	9.099	17.552
	Bad mesh	55.716	33.589	60.928
Partial Lake Superior	New method	60.366	41.570	71.243
	Laplacian	60.872	42.102	73.512

5.6. *Timing comparison between algorithms*

The run times of the Laplacian, Zhou and Shimada, and new smoothing methods on a 1.8 GHz Pentium 4 CPU are shown in Table 11 for the full and partial Lake Superior meshes. Although the new method is slower than the other algorithms, the new method's goal is to achieve a quality mesh, not to achieve fast smoothing.

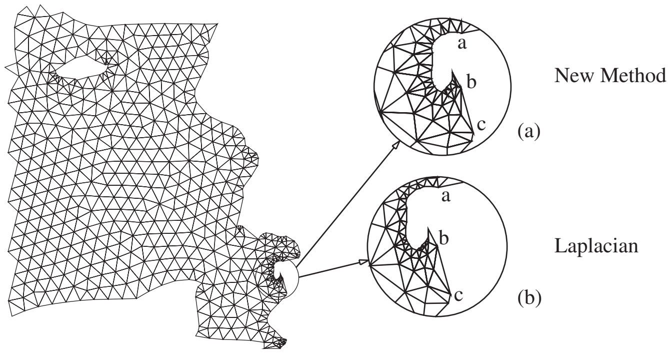


Fig. 17. Partial Lake Superior boundary conditions.

Table 11
Run time comparison of three algorithms(s)

	Laplacian	Z&S	New method
Partial Lake Superior	0.156	0.156	0.453
Full Lake Superior	1.703	1.765	3.359

6. Conclusion

In this paper, a new smoothing algorithm for mesh smoothing has been presented. In this new smoothing algorithm, an angle-based optimization method is used to optimize a torsion spring system. The formulation is set up to optimize the locations of all internal nodes. The solution is found based on the Gauss–Newton optimization.

The new mesh smoothing algorithm was applied to several meshes for testing purposes, including a mesh with the potential to produce invalid elements and meshes with badly shaped elements. The results obtained were measured with a set of metrics: average angle ratio, average length ratio, median angle ratio, median length ratio, the fifth percentile of the angle ratio, the fifth percentile of the length ratio, and the minimum angle. The advantages of the new method include the consistency of its formulation, its creation of meshes with less badly shaped elements, and its decreased likelihood of producing invalid elements.

Our future work will include extending the new smoothing algorithm to 3D grids.

References

[1] S.W. Cheng, T.K. Dey, H. Edelsbrunner, M.A. Facello, S.H. Teng, Sliver exudation, in: Proceedings of 15th Annual Symposium on Computational Geometry, Miami Beach, FL, 1999, pp. 1–13.

[2] M. Bern, P. Plassmann, Mesh generation, in: J. Sack, J. Urrutia (Eds.), Handbook of Computational Geometry, Elsevier Science, Amsterdam, 1998.

[3] S. Owen, S. Saigal, Neighborhood-based element sizing control for finite element surface meshing, in: Proceedings of the Sixth International Meshing Roundtable, Park City, UT, 1997, pp. 143–154.

[4] A. Cunha, S. Canann, S. Saigal, Automatic boundary sizing for 2D and 3D meshes, in: AMD-vol. 220 Trends in Unstructured Mesh Generation, ASME, 1997, pp. 65–72.

[5] L. Freitag, On combining Laplacian and optimization-based mesh smoothing techniques, in: Proceedings of the Sixth International Meshing Roundtable, AMD-vol 220, London, 1997, pp. 375–390.

[6] T. Zhou, K. Shimada, An angle-based approach to two-dimensional mesh smoothing, in: Proceedings of the Ninth International Meshing Roundtable, New Orleans, 2000, pp. 373–384.

[7] V. Parthasarathy, S. Kodiyalam, A constrained optimization approach to finite element mesh smoothing, Finite Elements Anal. Design 9 (1991) 309–320.

[8] Z. Chen, J. Tristano, W. Kwok, Construction of an objective function for optimization-based smoothing, Eng. Comput. (London) 20 (3) (2004) 184–192.

[9] S. Canann, M. Stephenson, T. Blacker, Optismoothing: an optimization-driven approach to mesh smoothing, Finite Elements Anal. Design 13 (1993) 185–190.

[10] T. Li, S. Wong, Y. Hon, C. Armstrong, R. McKeag, Smoothing by optimisation for a quadrilateral mesh with invalid element, Finite Elements Anal. Design 34 (2000) 37–60.

[11] N. Amenta, M. Bern, D. Eppstein, Optimal point placement for mesh smoothing, in: Proceedings of the Eighth ACM–SIAM Symposium on Discrete Algorithms, New Orleans, 1997, pp. 528–537.

[12] S.A. Canann, J.R. Tristano, M.L. Staten, An approach to combined Laplacian and optimization-based smoothing for triangular, quadrilateral, and quad-dominant meshes, in: Proceedings of the Seventh International Meshing Roundtable, Dearborn, MI, 1998, pp. 479–494.

[13] R. Bank, PLTMG, A software package for solving elliptic partial differential equations, in: Users' Guide 7.0, Frontiers in Applied Mathematics, vol. 15, SIAM, Philadelphia, PA, 1994.

[14] L. Freitag, C. Ollivier-Gooch, A comparison of tetrahedral mesh improvement techniques, in: Proceedings of the 15th International Meshing Roundtable, Albuquerque, NM, 1996, pp. 87–100.

[15] L. Freitag, M. Jones, P. Plassmann, An efficient parallel algorithm for mesh smoothing, in: Proceedings of the Fourth International Meshing Roundtable, Albuquerque, NM, 1995, pp. 47–58.

[16] D.A. Field, Laplacian smoothing and Delaunay triangulations, Comm. Appl. Numer. Methods 4 (1988) 709–712.

[17] L. Chen, Mesh smoothing schemes based on optimal Delaunay triangulations, in: Proceedings of the 13th International Meshing Roundtable, Williamsburg, VA, 2004, pp. 109–120.

[18] O. Egorova, N. Kojekine, I. Hagiwara, M. Savchenko, I. Semonova, V. Savchenko, Improvement of mesh quality using a statistical approach, in: Proceedings of Visualization, Imaging, and Image Processing, Benalmadena, Spain, 2003, pp. 1016–1021.

[19] P. Philippe, T. Baker, A comparison of triangle quality measures, in: Proceedings of the 10th International Meshing Roundtable, Newport Beach, CA, 2001, pp. 327–340.

[20] J. Shewchuk, What is a good linear element? Interpolation, conditioning, and quality measures, in: Proceedings of the 11th International Meshing Roundtable, Ithaca, NY, 2002, pp. 115–126.

[21] M. Berzins, Mesh quality: a function of geometry error estimates or both, in: Proceedings of the Seventh International Meshing Roundtable, Dearborn, MI, 1998, pp. 229–238.

[22] J. Nocedal, S.J. Wright, Numerical Optimization, Springer, New York, 1999.