

Research Article

A Derivative-Free Mesh Optimization Algorithm for Mesh Quality Improvement and Untangling

Jibum Kim,¹ Myeonggyu Shin,¹ and Woochul Kang²

¹Department of Computer Science and Engineering, Incheon National University, Incheon 406-772, Republic of Korea

²Department of Embedded Systems Engineering, Incheon National University, Incheon 406-772, Republic of Korea

Correspondence should be addressed to Woochul Kang; wchkang@incheon.ac.kr

Received 10 November 2014; Revised 19 March 2015; Accepted 19 March 2015

Academic Editor: Lu Zhen

Copyright © 2015 Jibum Kim et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

We propose a derivative-free mesh optimization algorithm, which focuses on improving the worst element quality on the mesh. The mesh optimization problem is formulated as a min-max problem and solved by using a downhill simplex (amoeba) method, which computes only a function value without needing a derivative of Hessian of the objective function. Numerical results show that the proposed mesh optimization algorithm outperforms the existing mesh optimization algorithm in terms of improving the worst element quality and eliminating inverted elements on the mesh.

1. Introduction

It is well known that mesh qualities affect both the accuracy and efficiency of partial differential equation (PDE) solutions [1, 2]. Elements with poor mesh qualities (also, inverted elements) often occur during mesh generation [3], mesh deformation [4, 5], and mesh optimization [5]. Researchers have proposed many mesh quality improvements and untangling algorithms to solve these problems [5–9]. Mesh smoothing algorithms improve mesh qualities simply by moving vertex positions, while fixing element connectivities. One of the most popular mesh smoothing algorithms is Laplacian smoothing [10], where new vertex position becomes a geometric center of adjacent vertices. However, Laplacian smoothing does not guarantee generating good element qualities and instead often generates inverted elements [11]. Several variants of Laplacian smoothing, such as smart Laplacian smoothing, have been proposed to overcome the limitations of Laplacian smoothing [12, 13].

Optimization-based mesh quality improvement algorithms [2, 11, 13] are now becoming more popular, because these methods are able to offer high-quality meshes though with high computational costs. These methods formulate the nonlinear objective function over the entire mesh domain and improve mesh qualities by minimizing the objective

function, while keeping mesh topologies [5, 14]. On the other hand, topological change methods improve mesh qualities by iteratively performing edge splitting, edge swapping, and edge collapsing [15, 16]. Researchers have pointed out that mesh smoothing is preferred to topological changes for many PDE-based applications, since it is able to improve element qualities, while keeping mesh topologies [5]. For many PDE problems, consistent mesh topology is desirable for PDE solution accuracy [5, 17]. Several optimization-based simultaneous untangling and smoothing algorithms have been proposed [5–7]. These methods are known to be able to simultaneously eliminate inverted elements, while improving mesh qualities.

Most previous mesh quality improvement methods have focused on improving the average element qualities on the mesh. However, it is well known that a few poor quality elements significantly diminish the efficiency and accuracy of PDE solutions [1, 7]. The inverted elements in the mesh can even result in invalid PDE solutions [4, 9]. In this paper, we focus on improving the worst element quality in the mesh and also focus on eliminating inverted elements in the mesh. The optimization problem, which improves the worst quality elements, is formulated as a nonlinear optimization problem, which is nonsmooth, and thus, the computation of derivative or Hessian is not available. For

this reason, traditional nonlinear optimization solvers such as conjugate gradient, steepest descent, Newton, quasi-Newton, and trust-region methods are not directly applicable for solving these nonsmooth mesh optimization problems [7, 13]. Several derivative-free mesh optimization algorithms have been proposed [7, 12, 18]. For example, Freitag and Plassmann proposed an active set algorithm for solving mesh optimization problems [12]. This method improves element qualities by maximizing the minimum area of an element and solves the problem using linear programming. However, this method requires the mesh quality to be convex. Also, other derivative free mesh optimization algorithms need many initial parameters to be chosen or are quite complex to use in practice [7, 18].

We propose an efficient derivative-free mesh optimization algorithm for mesh quality improvement and untangling. Our proposed method employs a downhill simplex method for improving the worst element quality and eliminating inverted elements on the mesh. Our method does not require the computation of a derivative or Hessian of the objective function, but it iteratively performs reflection, expansion, and contraction, to reach the local optimal point (vertex location). Our algorithm is simple to apply and does not need many initial parameters to be determined. We will show numerically that our proposed mesh optimization algorithm outperforms traditional mesh optimization algorithms in terms of worst element quality and speed.

2. Problem Formulation

We mathematically formulate mesh quality metrics and objective functions to optimize. We also describe the local mesh optimization method we used.

2.1. Mesh Quality Metric. There are many mesh quality metrics for triangles and tetrahedra in the literature. See [1] for more details. Among them, we choose an inverse mean ratio (IMR) quality metric to improve the element shape [19]. The IMR quality metric defines an ideal (reference) element and measures how similar the actual (physical) element to the reference element is. The reference triangle is often considered to be an equilateral triangle for isotropic PDE problems. Let the coordinates of three vertices in the triangle element be $[a, b, c]$. The incidence matrix, A , of this (physical) element is defined as

$$A = [b - a, c - a]. \quad (1)$$

Let the incidence matrix for the reference element be W . For the isotropic PDE problem, the ideal element is considered to be an equilateral triangle or tetrahedron. For the equilateral triangle, W is defined as

$$W = \begin{bmatrix} 1 & \frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} \end{bmatrix}. \quad (2)$$

The IMR quality metric is defined as

$$q_{\text{imr}} = \frac{\|AW^{-1}\|_F^2}{2|\det(AW^{-1})|}, \quad (3)$$

where $\|\cdot\|_F$ is a Frobenius norm. When the actual element and the ideal element are identical, A is the same as W and thus, AW^{-1} becomes an identity matrix. In this case, q_{imr} becomes one.

We choose an untangling beta (q_{ub}) quality metric to eliminate inverted elements on the mesh [8]. Inverted elements on the mesh have negative signed areas (volume). Untangling beta quality metric gives a high cost (penalty) to the element, which has a negative signed area. Let the area of the triangle be A . The untangling beta quality metric is defined as

$$q_{\text{ub}} = \frac{1}{2} (|A - \beta| - (A - \beta)), \quad (4)$$

where β is a user-defined value, which is close to zero. For a valid (noninverted) element, q_{ub} becomes zero. For the above two quality metrics, a smaller value indicates better element quality.

2.2. Objective Function. We focus on improving the worst element quality on the mesh. Let the i th element quality on the mesh be q_i . The element with the worst quality is denoted as

$$\max_{1 \leq i \leq n} (q_i), \quad (5)$$

where n is the number of elements. Then, our goal is to minimize (5). The optimization problem is formulated as

$$\min \max_{1 \leq i \leq n} (q_i). \quad (6)$$

Note that the mesh optimization problem in (6) is a nonsmooth objective function. Thus, traditional nonlinear optimization solvers, which require the computation of derivative and/or Hessian, cannot be used to solve it.

2.3. Local Mesh Optimization. In the literature, there are essentially two kinds of mesh optimization strategies: local (Gauss-Seidel type) optimization and global (Jacobi type) mesh optimization. For local mesh optimization, an entire mesh is decomposed into multiple patches. A patch is defined as a set of adjacent elements around the free vertex to optimize, as shown in Figure 1. When local mesh optimization is performed, we visit one free vertex at a time and iteratively move and optimize other vertices. For global mesh optimization, all vertices on the mesh move simultaneously. That is, the entire mesh becomes a single path.

We use local mesh optimization to improve the mesh quality, because previous researches have shown the efficiency of local mesh optimization compared to the global mesh optimization [3, 20]. When global mesh optimization is used, the update of all vertex positions often gets truncated to satisfy mesh validity. Also, the objective function and corresponding matrices in the problem become too large, when the mesh size is too big (e.g., if the mesh includes millions of vertices).

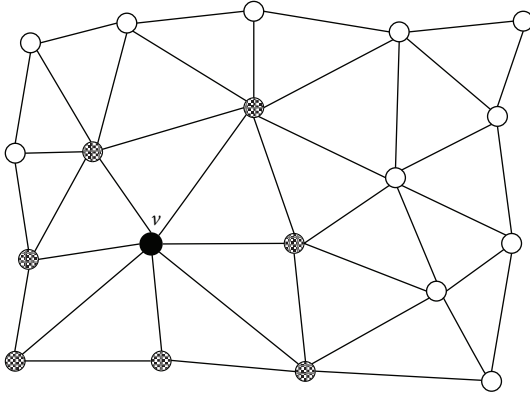


FIGURE 1: This figure shows an example of local mesh optimization. The seven vertices and elements, which are adjacent to the free vertex, v , are optimized at this time. This local mesh optimization is repeated for other vertices on the mesh until the entire mesh is fully optimized.

3. A Downhill Simplex Method for Mesh Quality Improvement and Untangling

The optimization problem in (6) is not a smooth objective function due to the maximum function. Thus, traditional nonlinear optimization solvers cannot be used to solve the optimization problem, because the computation of a gradient and Hessian value is not available. Thus, we propose to use a downhill simplex (amoeba) method to solve (6), which does not use either function derivatives or a Hessian but only uses function evaluations.

The basic idea of a downhill simplex method is to move a free vertex by repeatedly performing three actions: reflections, expansions, and contractions. More details on these three actions will be described in the following sections.

3.1. Initial Simplex. The downhill simplex method requires an initial simplex to begin. A simplex is a triangle and tetrahedron in 2D and 3D, respectively. For the local mesh optimization, an initial simplex is generated around the free vertex to optimize by choosing two (virtual) points. It is natural to assume that the initial simplex size should consider the edge length of the mesh or mesh size. We use the minimum edge length information of the patch around the free vertex to determine the initial simplex size. We expect that the initial simplex size should be smaller than the minimum edge length of the patch. Otherwise, the initial simplex location is placed beyond the patch where the free vertex places and two chosen (virtual) points could be too far away from the optimal vertex position. On the other hand, if the initial simplex is too small, the convergence of mesh optimization could be very slow.

Figure 2 shows the initial simplex around the free vertex, v . Two (virtual) points, v_1 and v_2 , and v , form an initial simplex. Here, α is computed as

$$\alpha = (\text{simplex diameter}) * (\text{minimum edge length}), \quad (7)$$

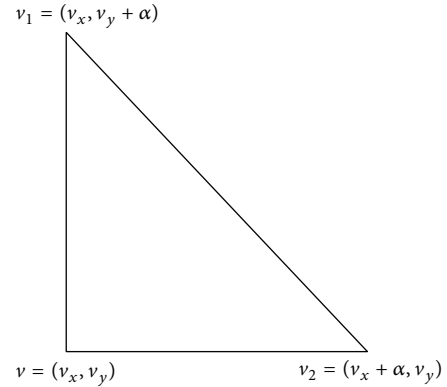


FIGURE 2: This figure shows an initial simplex (triangle) to begin when a free vertex v is given. A free vertex v and two (virtual) points, v_1 and v_2 , compose an initial simplex. The α is a user-defined parameter defined in (7).

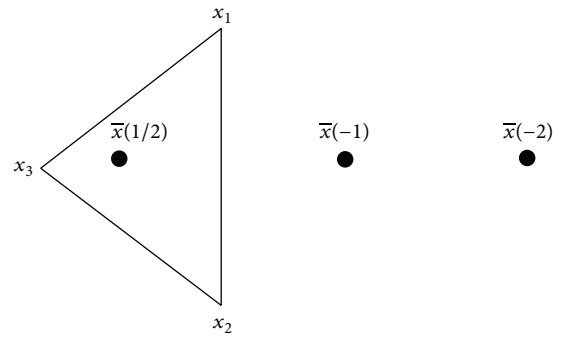


FIGURE 3: This figure shows one step of the downhill simplex method in 2D. The current simplex has three points, x_1 , x_2 , and x_3 . The reflected point $\bar{x}(-1)$ is located at the opposite side of the point, x_3 , with the worst cost function. Here, $\bar{x}(-2)$ and $\bar{x}(1/2)$ are expansion and contraction points, respectively.

where the simplex diameter is a user-defined parameter. We will discuss the effect of the simplex diameter on the convergence and the final mesh quality in more detail in Section 4.

3.2. Downhill Simplex Method. In a single iteration of the downhill simplex method, it seeks to eliminate the (virtual) point with the worst cost function and replaces it with a new one by repeatedly performing three different actions: reflection, expansion, and contraction. Let the three points of the current simplex be $\{x_1, x_2, x_3\}$ such that

$$f(x_1) \leq f(x_2) \leq f(x_3). \quad (8)$$

We find the vertex with the worst cost function on the simplex (i.e., x_3) and compute the reflected point of this vertex using the centroid of the current simplex (see Figure 3). The centroid of three points is computed as

$$\bar{x} = \frac{\sum_{i=1}^3 x_i}{3}. \quad (9)$$

```

Compute the reflected point,  $\bar{x}(-1)$  and function value at this point,
 $f(\bar{x}(-1))$ . There are 3 possibilities:
Case 1 (reflection):
if  $f(x_1) \leq f(\bar{x}(-1)) < f(x_2)$  then
    replace  $x_3$  by  $\bar{x}(-1)$ .
end if
Case 2 (expansion):
if  $f(\bar{x}(-1)) < f(x_1)$  then
    compute the expansion point,  $\bar{x}(-2)$  and evaluate  $f(\bar{x}(-2))$ .
    if  $f(\bar{x}(-2)) < f(\bar{x}(-1))$  then
        replace  $x_3$  by  $\bar{x}(-2)$ .
    else
        replace  $x_3$  by  $\bar{x}(-1)$ .
    end if
end if
Case 3 (contraction):
if  $f(\bar{x}(-1)) > f(x_2)$  then
    compute the contraction point,  $\bar{x}(1/2)$  and evaluate  $f(\bar{x}(1/2))$ .
    if  $\bar{x}(1/2) < f(x_3)$  then
        replace  $x_3$  by  $\bar{x}(1/2)$ .
    else
        replace  $x_i$  by  $(1/2)(x_1 + x_i)$  for  $i = 2, 3$ 
    end if
end if

```

ALGORITHM 1: Downhill simplex algorithm.

Now, these three points, including the reflected point, $\bar{x}(-1)$, compose a new simplex. Here, $\bar{x}(p)$ is denoted by

$$\bar{x}(p) = \bar{x} + p(x_3 - \bar{x}). \quad (10)$$

We consider three possibilities for this reflected point, $\bar{x}(-1)$. (1) If the function value at the reflected point is neither worst nor best in the new simplex, then replace the worst point by the reflected point. (2) If function value at the reflected point is better than the one at the current point, we perform expansion, since this means that the direction of reflection is good. (3) If function value at the reflected point is worse than the one at the current point, we perform contraction, since this means that the triangle is too large.

Figure 3 shows a current simplex with three points, x_1 , x_2 , and x_3 , reflected point, $\bar{x}(-1)$, expansion point, $\bar{x}(-2)$, and contraction point, $\bar{x}(1/2)$. We repeatedly perform the downhill simplex algorithm, which is illustrated in Algorithm 1, until all vertices converge to the local optimum.

4. Numerical Results

In this section, we describe our numerical results to show the effectiveness of our proposed algorithm, which improves the worst element quality on the mesh using the downhill simplex method.

4.1. Test Meshes. We consider four test meshes to evaluate our algorithm. Cylinder and bar meshes (see Figures 4(a) and 4(b)) include many poor quality elements, which were produced during mesh deformation. We use these two meshes

TABLE 1: The test mesh configurations.

Mesh name	Number of vertices	Number of elements	Number of inverted elements
Cylinder	1,196	2,240	0
Bar	4,167	7,232	0
Hydrocephalus	4,311	8,166	13
Airfoil	5,252	10,200	42

TABLE 2: Mesh quality statistics.

Mesh name	Min.	Avg.	Max.	std.dev
Cylinder	1.00	3.09	678.92	15.48
Bar	1.00	2.15	71.12	2.97
Hydrocephalus	-27.24	1.89	551.52	8.42
Airfoil	-4.30e - 06	37.05	132,540	1316.26

to improve the worst element quality on the mesh, because these meshes include very skinny (poor quality) elements. Hydrocephalus and airfoil meshes (see Figures 4(c) and 4(d)) include some inverted elements. The hydrocephalus mesh is provided by Park et al. [21]. These inverted elements were produced during a mesh deformation process.

Properties of four test meshes and mesh quality statistics are shown in Tables 1 and 2, respectively. The inverse mean ratio quality metric is used to measure the overall mesh quality. The value of the inverse mean ratio quality metric lies between 1 and ∞ for valid (noninverted) elements. A smaller

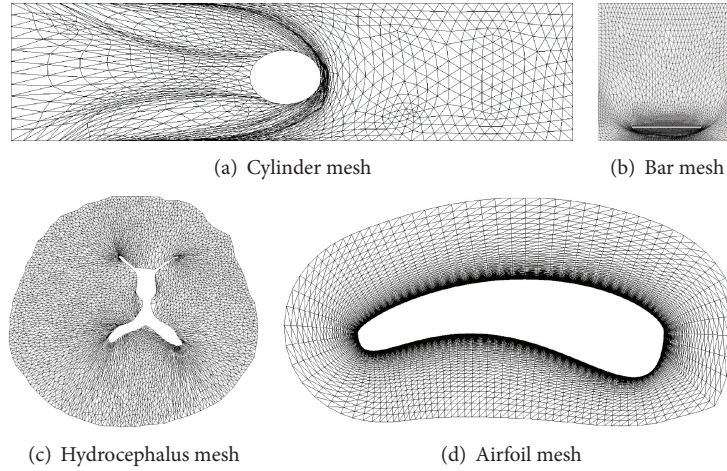


FIGURE 4: (a) Cylinder mesh, (b) bar mesh, (c) hydrocephalus mesh, and (d) airfoil mesh.

value indicates a better mesh quality and the ideal element has the value of one. Elements with negative values indicate inverted elements on the mesh. The initial hydrocephalus and airfoil meshes have 13 and 42 inverted elements, respectively.

4.2. Effect of Initial Simplex Diameter on Algorithm Performance. When the downhill simplex method is used, the initial simplex is composed of one free vertex to optimize and two (virtual) points as discussed in Section 3.1. Other than the free vertex to optimize, the location of the two points is determined by the initial simplex diameter and the minimum edge length in the patch. For example, if simplex diameter is 1, the initial simplex size becomes a minimum edge length in the patch where a free vertex belongs. We investigate the effect of the initial simplex diameter on mesh optimization performance. Here, the mesh optimization performance means both the mesh quality after performing mesh optimization and the time to optimize (time to converge).

Figure 5 shows the worst element quality and timing to optimize as a function of various initial simplex diameters for test meshes. For these meshes, we observe that the best initial simplex diameters lie between 0.01 and 0.1. For this initial simplex size, the output meshes have the smallest worst element quality with the fastest convergence rate. If the initial simplex diameter is too small (e.g., simplex diameter = 0.001), we observe that the convergence time is very slow. For this case, the initial simplex size is too small compared with the mesh size and thus, the movement of one step of the downhill simplex method is very small. If the initial simplex diameter is too big (e.g., simplex diameter = 5), the optimization problem fails to converge or fails to untangle inverted elements. For these cases, we expect that the initial simplex size is chosen to be too big compared with the element size or edge lengths on the mesh. Thus, we chose $\alpha = 0.1 * (\text{minimum edge length})$. Table 3 shows the minimum and maximum edge lengths on the mesh and the best initial simplex diameter. For all test meshes, the best simplex diameters were placed between the minimum and the maximum edge lengths on the mesh.

TABLE 3: Connection between the edge length on the mesh and the best initial simplex diameter.

Mesh name	Minimum edge length	Maximum edge length	Best simplex diameter
Cylinder	$2e - 03$	0.19	0.1
Bar	$2e - 03$	2.15	0.01
Hydrocephalus	$4e - 02$	36.51	0.1
Airfoil	$3e - 06$	0.29	0.1

TABLE 4: Summary of the proposed and traditional mesh optimization methods.

Method	Problem	Optimization solver
Proposed method	$\min \max_{1 \leq i \leq n} (q_i)$	Downhill simplex method
Traditional method	$\min \sum_{i=1}^n q_i$	Nonlinear conjugate method

4.3. Comparison with Existing Mesh Quality Improvement Methods. We compare our proposed method with a traditional (existing) mesh optimization method that focuses on improving the average element quality on the mesh. The traditional mesh optimization method, which has been used in many previous research articles [2, 5–7], is to minimize

$$\sum_{i=1}^n q_i \quad (11)$$

or to minimize $\sum_{i=1}^n q_i^2$, where n is number of elements. We used the nonlinear conjugate gradient (NLCG) method to minimize (11), because many previous papers have shown the effectiveness of using the NLCG method to minimize (11). Table 4 summarizes both our proposed and traditional mesh optimization methods.

The NLCG method updates the vertex position using a line search technique. Let x_k be a vector of vertex positions at

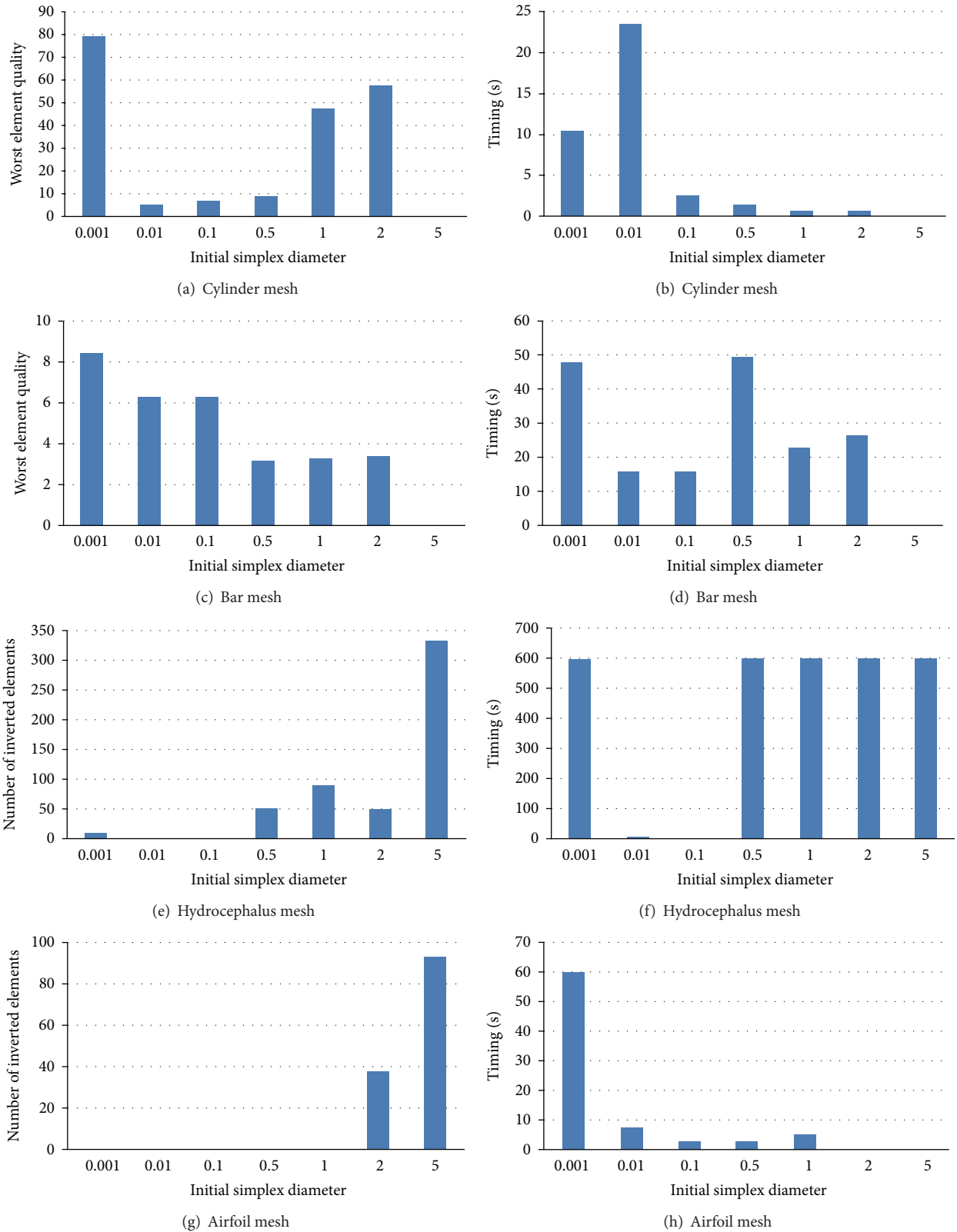


FIGURE 5: These figures show the worst element quality and timing results to converge for various initial simplex diameters. A smaller value indicates a better element quality for the inverse mean ratio quality metric. In (a) and (b), the optimization problem does not converge when the initial simplex diameter is five. In (c) and (d), the optimization problem does not converge when the initial simplex diameter is five. In (e) and (f), all inverted elements are eliminated after optimization when the initial simplex diameters are 0.01 and 0.1. Other initial simplex diameters fail to eliminate inverted elements. In (g) and (h), the optimized meshes fail to untangle inverted elements when the initial simplex diameter is two and five.

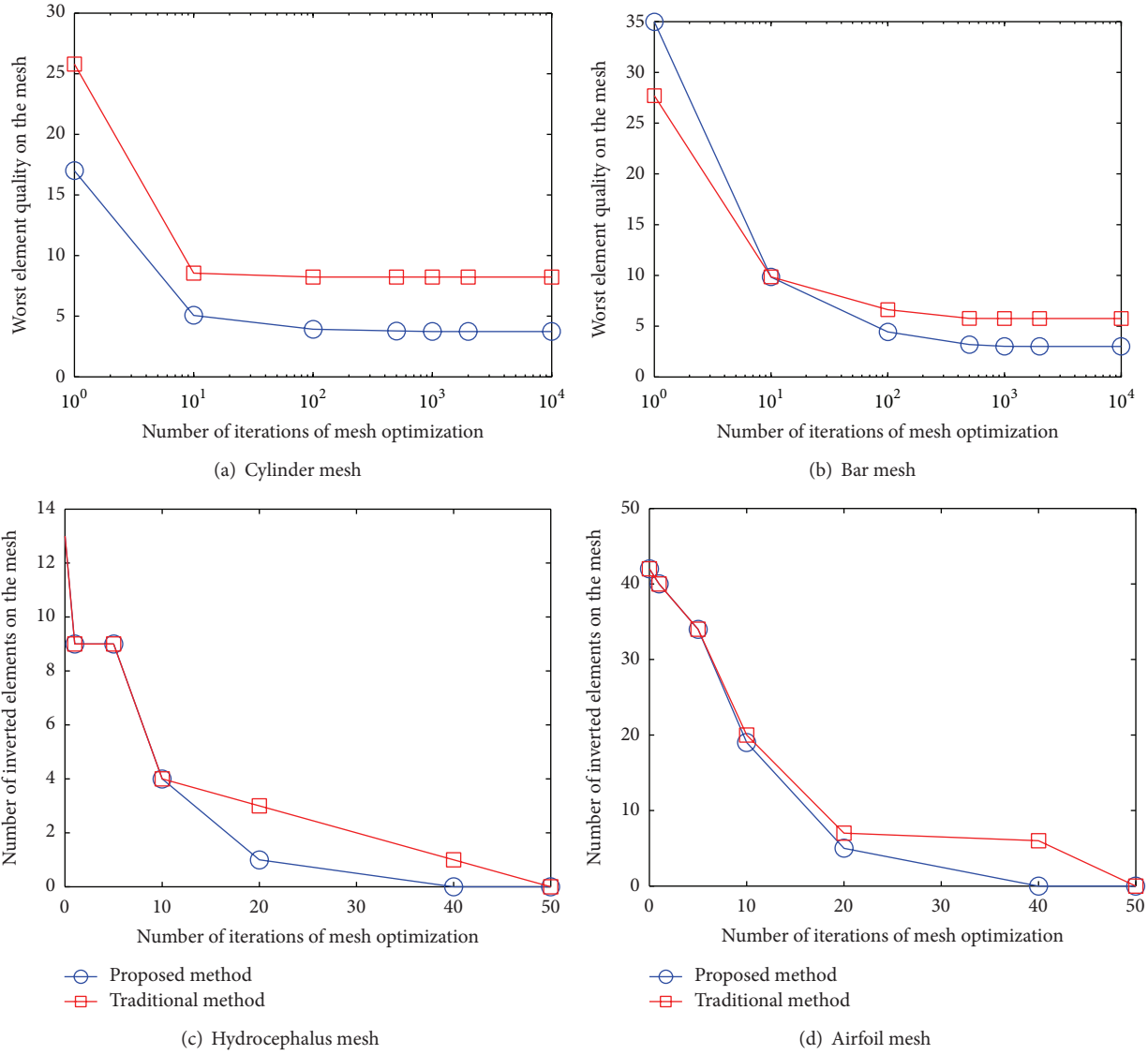


FIGURE 6: Note that a smaller value indicates a better element quality for these figures. The inverse mean ratio quality metric is employed to measure the element quality for (a) and (b): (a) worst element quality as a function of number of iterations of mesh optimization on the cylinder mesh. (b) Worst element quality as a function of number of iterations of mesh optimization on the bar mesh. (c) Number of inverted elements on the hydrocephalus mesh as a function of number of iterations of mesh optimization. (d) Number of inverted elements on the airfoil mesh as a function of number of iterations of mesh optimization.

the k th iteration and let p_k be the search direction. Then, the updated vertex position, x_{k+1} , is computed by

$$x_{k+1} = x_k + \alpha_k p_k, \quad (12)$$

where α_k is a step length. The search direction, p_k , is denoted by

$$p_k = -\nabla F(x_k) + \beta_k^{\text{PR}} p_{k-1}, \quad (13)$$

where β_k^{PR} is a parameter given by

$$\beta_k^{\text{PR}} = \frac{\nabla \tilde{F}(x_k)^T (\nabla \tilde{F}(x_k) - \nabla \tilde{F}(x_{k-1}))}{\nabla \tilde{F}(x_{k-1})^T \nabla \tilde{F}(x_{k-1})} \quad (14)$$

for Polak-Ribière NLCG method. See [5, 14] for more details about the nonlinear conjugate gradient method for solving mesh optimization problems. Note that the NLCG method is not applicable to solve our mesh optimization problem, (6), which improves the worst element quality on the mesh, since NLCG requires the gradient computation of the objective function.

Figures 6(a) and 6(b) show the worst element quality with respect to the number of iterations of mesh optimization. We observe that our proposed method, which improves the worst element quality using the downhill simplex method, outperforms the traditional method, which improves the average element quality on the mesh. Our method improves the worst element quality up to 54.6% compared with

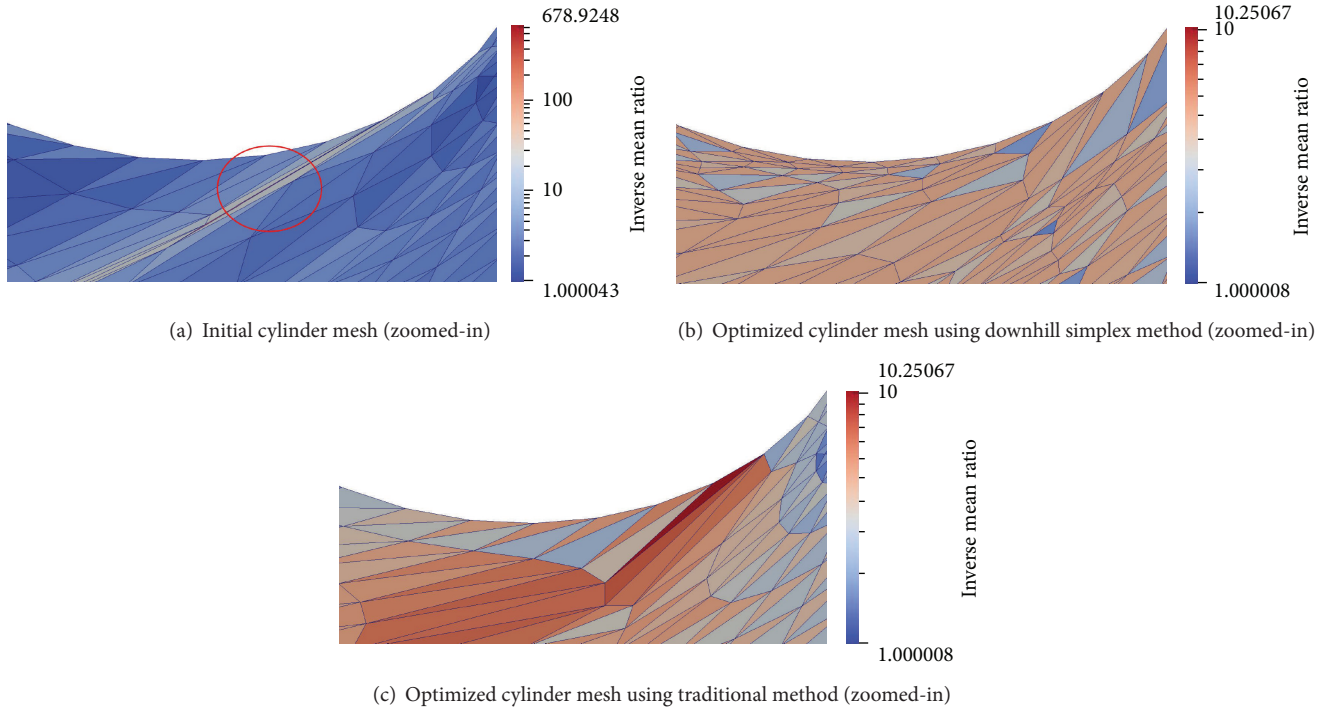


FIGURE 7: (a) Initial cylinder mesh (zoomed-in). This mesh has a very skinny element (red) with the element quality of 678.9248 enclosed by the red circle. (b) Optimized cylinder mesh (zoomed-in) using the proposed approach. The downhill simplex method is used to improve the worst element quality. The worst quality element has the worst element quality with 3.73. (c) Optimized cylinder mesh (zoomed-in) using the traditional approach. The traditional approach focuses on improving the average element quality using the nonlinear conjugate gradient method. The worst quality element has the worst element quality with 8.23.

the traditional mesh optimization method. Figure 7 shows zoomed-in mesh, where the element with the worst element quality is located. The initial mesh has the element with the worst element quality in the middle (see a skinny red element enclosed by the circle). This element is very skinny and the element quality is poor with 678.9248 when the inverse mean ratio (IMR) quality metric is used to measure the element quality. Note that a smaller value indicates a better element quality for the IMR quality metric and the ideal element has value of one. After performing the downhill simplex method to improve the worst element quality, these skinny triangles are eliminated and the worst element quality becomes 3.73 (see Figure 7(b)). The traditional mesh optimization method using NLCG also improves the worst element quality but shows inferior performance compared with the proposed method. The output mesh quality using the proposed algorithm is 54.6% better than the output mesh using the traditional method for the cylinder mesh. The output mesh using the traditional method shows the worst element quality with 10.25 (see red elements in Figure 7(c)).

Figures 6(c) and 6(d) show the number of inverted elements on the mesh as a function of number of iterations of mesh optimization. We observe that the proposed mesh optimization method takes fewer iterations to untangle inverted elements compared with the traditional mesh optimization method. For these numerical experiments, the untangling beta quality metric was used to eliminate inverted

elements on the mesh. When the initial mesh includes inverted elements, these elements become the worst quality elements on the mesh. For the hydrocephalus mesh, there are several inverted elements, which are very skinny (see Figure 8(a)). These elements are eliminated after performing mesh optimization (see Figures 8(b) and 8(c)). Both output meshes have no inverted elements, but the proposed method results in better worst element qualities than the traditional method.

4.4. Comparison with a Log-Barrier Interior Point Method.

We compare our derivative-free mesh optimization method with a log-barrier interior point method (simply, log-barrier method) [7]. The log-barrier method is a popular derivative-free mesh optimization method, which focuses on improving the worst element quality on the mesh. The log-barrier method is flexible in that it can be used to untangle inverted elements on the mesh [7]. Different from other mesh quality optimization methods, it uses the log-barrier objective function to optimize meshes, which is defined as

$$F(\mu, t) = t + \mu \sum_{i=1}^n \log\left(\frac{1}{q_i} - t\right), \quad (15)$$

where μ and t are auxiliary parameters and n is the number of elements. A quantity t is smaller than $1/q_i$ and μ is greater than 0. The log-barrier method maximizes F with respect to

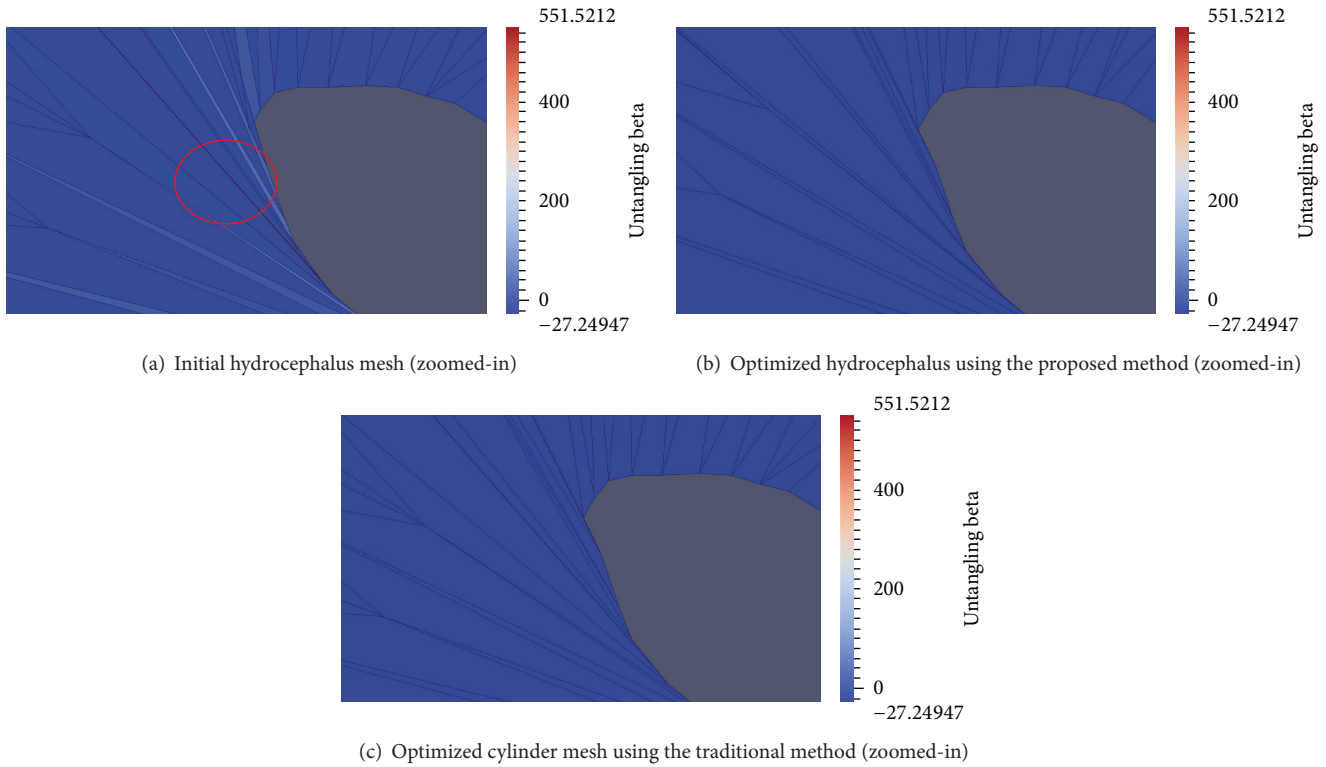


FIGURE 8: (a) Initial hydrocephalus mesh (zoomed-in). This mesh has a very skinny inverted element (red) with the element quality of 551.5212 enclosed by the red circle. (b) Optimized cylinder mesh (zoomed-in) using the proposed approach. The downhill simplex method is used to improve the worst element quality. After performing mesh optimization, all inverted elements are eliminated. (c) Optimized cylinder mesh (zoomed-in) using the traditional approach. The traditional approach focuses on improving the average element quality using the nonlinear conjugate gradient method. The optimized mesh does not have any inverted elements.

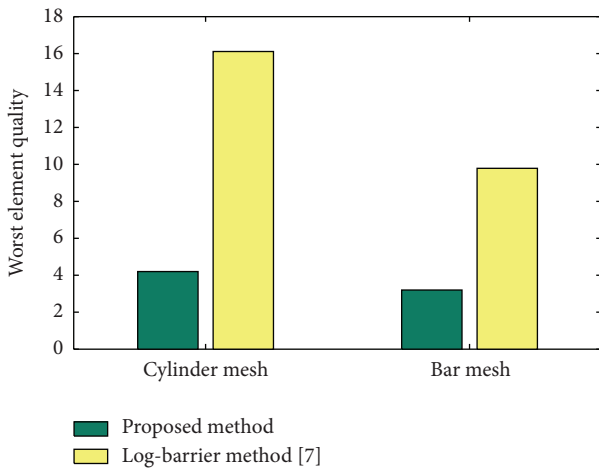


FIGURE 9: Comparison of the worst element quality on the optimized mesh between the proposed method and the log-barrier method. The inverse mean ratio quality metric is used to measure the mesh quality. A smaller value indicates a better mesh quality.

t until convergence. Readers refer to [7] for more details on the log-barrier method. Figure 9 shows comparison between the proposed method and the log-barrier method with respect to the worst element quality on the mesh. Note that

some parameters of the log-barrier method could not be fully optimized. We observe that both methods significantly improve the worst element quality. However, the proposed method improves the worst element quality up to 75.0% better than the log-barrier method.

We also compare our derivative-free mesh optimization method with the log-barrier method for the initially tangled meshes (see Figures 4(c) and 4(d)). For the tangled airfoil mesh shown in Figure 4(c), our method takes 3.1 seconds to untangle inverted elements while the log-barrier method takes 323.2 seconds to untangle inverted elements. For the tangled hydrocephalus mesh shown in Figure 4(d), both methods take approximately 3 seconds to untangle inverted elements.

5. Conclusions

We have proposed a derivative-free mesh optimization algorithm, which improves the worst element quality on the mesh. We have formulated the mesh optimization problem as a nonlinear optimization problem and solved it using a downhill simplex (amoeba) method. The downhill simplex method was able to solve the nonlinear optimization problem by computing just a function value without needing a derivative or Hessian of the objective function. We have compared our proposed method with the existing mesh optimization

algorithms, in terms of the worst element quality and the convergence time.

Numerical results show that our proposed algorithm is more effective in improving the worst element quality on the mesh compared with the one using the existing mesh optimization algorithms. Specifically, the worst element quality using the proposed algorithm is up to 75.0% better than the existing mesh optimization methods. Moreover, the proposed algorithm is more efficient and faster in eliminating inverted elements on the mesh compared with the existing mesh optimization algorithms.

In this paper, we have focused on improving only one aspect of the mesh: element shape. We plan to apply our derivative-free mesh optimization algorithm on simultaneously improving more than two aspects of the mesh. We also plan to parallelize the downhill simplex algorithm using an OpenMP library.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

The authors were supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT and Future Planning (NRF-2014R1A1A1036677). The authors would like to thank Shankar P. Sastry for providing the source code of the log-barrier interior point method.

References

- [1] J. Shewchuk, "What is a good linear element? Interpolation, conditioning, and quality measures," in *Proceedings of the 11th International Meshing Roundtable*, pp. 115–126, Sandia National Laboratories, Ithaca, NY, USA, September 2002.
- [2] J. Kim, S. P. Sastry, and S. M. Shontz, "A numerical investigation on the interplay amongst geometry, meshes, and linear algebra in the finite element solution of elliptic PDEs," *Engineering with Computers*, vol. 28, no. 4, pp. 431–450, 2012.
- [3] R. Garimella, J. Kim, and M. Berndt, "Polyhedral mesh generation and optimization for non-manifold domains," in *Proceedings of the 22nd International Meshing Roundtable*, pp. 239–250, Sandia National Laboratories, October 2013.
- [4] J. Danczyk and K. Suresh, "Finite element analysis over tangled meshes," in *Proceedings of the ASME International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, vol. 2, pp. 89–95, Chicago, Ill, USA, August 2012.
- [5] J. Kim, T. Panitanarak, and S. M. Shontz, "A multiobjective mesh optimization framework for mesh quality improvement and mesh untangling," *International Journal for Numerical Methods in Engineering*, vol. 94, no. 1, pp. 20–42, 2013.
- [6] D. Benitez, E. Rodriguez, J. M. Escobar, and R. Montenegro, "Performance evaluation of a parallel algorithm for simultaneous untangling and smoothing of tetrahedral meshes," in *Proceedings of the 22nd International Meshing Roundtable*, pp. 579–598, Sandia National Laboratories, October 2013.
- [7] S. P. Sastry, S. M. Shontz, and S. A. Vavasis, "A log-barrier method for mesh quality improvement," in *Proceedings of the 20th International Meshing Roundtable Conference (IMR '11)*, pp. 329–346, Sandia National Laboratories, Paris, France, October 2011.
- [8] P. M. Knupp, "Hexahedral and tetrahedral mesh untangling," *Engineering with Computers*, vol. 17, no. 3, pp. 261–268, 2001.
- [9] S. Bhowmick and S. M. Shontz, "Towards high-quality, untangled meshes via a force-directed graph embedding approach," *Procedia Computer Science*, vol. 1, no. 1, pp. 357–366, 2010, Proceedings of the International Conference on Computational Science (ICCS '10).
- [10] D. A. Field, "Laplacian smoothing and Delaunay triangulations," *Communications in Applied Numerical Methods*, vol. 4, no. 6, pp. 709–712, 1988.
- [11] L. A. Freitag and P. M. Knupp, "Tetrahedral mesh improvement via optimization of the element condition number," *International Journal for Numerical Methods in Engineering*, vol. 53, no. 6, pp. 1377–1391, 2002.
- [12] L. A. Freitag and P. Plassmann, "Local optimization-based simplicial mesh untangling and improvement," *International Journal for Numerical Methods in Engineering*, vol. 49, no. 1–2, pp. 109–125, 2000.
- [13] M. Brewer, L. F. Diachin, P. Knupp, T. Leurent, and D. Melander, "The mesquite mesh quality improvement toolkit," in *Proceedings of the 12th International Meshing Roundtable*, pp. 239–250, Sandia National Laboratories, 2003.
- [14] S. P. Sastry and S. M. Shontz, "Performance characterization of nonlinear optimization methods for mesh quality improvement," *Engineering with Computers*, vol. 28, no. 3, pp. 269–286, 2012.
- [15] L. A. Freitag and C. Ollivier-Gooch, "Tetrahedral mesh improvement using swapping and smoothing," *International Journal for Numerical Methods in Engineering*, vol. 40, no. 21, pp. 3979–4002, 1997.
- [16] J. Kim, D. McLaurin, and S. M. Shontz, "A 2D topology-adaptive mesh deformation framework for mesh warping," in *Proceedings of the 4th Tetrahedron Workshop on Grid Generation for Numerical Computation*, May 2014.
- [17] P. Knupp, "Updating meshes on deforming domains: an application of the target-matrix paradigm," *Communications in Numerical Methods in Engineering*, vol. 24, no. 6, pp. 467–476, 2007.
- [18] J. Park and S. M. Shontz, "Two derivative-free optimization algorithms for mesh quality improvement," *Procedia Computer Science*, vol. 1, no. 1, pp. 387–396, 2010, Proceedings of the International Conference on Computational Science (ICCS '10).
- [19] T. Munson, "Mesh shape-quality optimization using the inverse mean-ratio metric," *Mathematical Programming*, vol. 110, no. 3, pp. 561–590, 2007.
- [20] L. Freitag Diachin, P. Knupp, T. Munson, and S. Shontz, "A comparison of two optimization methods for mesh quality improvement," *Engineering with Computers*, vol. 22, no. 2, pp. 61–74, 2006.
- [21] J. Park, S. M. Shontz, and C. S. Drapaca, "A combined level set/mesh warping algorithm for tracking brain and cerebrospinal fluid evolution in hydrocephalic patients," in *Image-Based Geometric Modeling and Mesh Generation*, vol. 3 of *Lecture Notes in Computational Vision and Biomechanics*, pp. 107–141, Springer, 2013.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

