Finite Element Mesh Smoothing Using Cohort Intelligence

Mandar S. Sapre, Anand J. Kulkarni and Sumit S. Shinde

Abstract Several approaches have been developed for mesh quality improvement by considering only node movement, without disturbing the element connectivity. Laplacian smoothing is well known and easiest one to implement among all, but this approach might lead to inverted elements in concave regions. In this work, optimization-based smoothing method is proposed for the improvement of hexahedral mesh without any inverted elements. Optimization is carried out using condition number-based objective function which defines qualities of individual elements in the hexahedral mesh. Numerical optimization method used is the cohort intelligence (CI) algorithm which is socio-inspired algorithm developed by Dr. Anand Kulkarni et al. (Cohort intelligence: a self-supervised learning behavior, pp. 1396–1400, 2013 [10]). The approach is demonstrated with three examples.

Keywords Hexahedral mesh · Condition number
Optimization-based smoothing · Cohort intelligence

1 Introduction

The finite element analyses (FEAs) are required to verify the suitability of an engineering design before actual manufacturing. There are two big challenges, one is to build a sufficiently accurate model and other is to carry out its analysis, in the available time. Meshing is the essential component of any analysis or simulation and obtaining useful meshes is an important issue. All commercial FEA softwares

M. S. Sapre (✉) · A. J. Kulkarni · S. S. Shinde
Symbiosis Institute of Technology,
Symbiosis International University, Pune 412 115, MH, India
e-mail: mandar.sapre@sitpune.edu.in

A. J. Kulkarni
e-mail: anand.kulkarni@sitpune.edu.in

S. S. Shinde
e-mail: sumit.shinde@sitpune.edu.in

A. J. Kulkarni et al. (eds.), *Proceedings of the 2nd International Conference on Data Engineering and Communication Technology*, Advances in Intelligent Systems and Computing 828, https://doi.org/10.1007/978-981-13-1610-4_48

are based on interpolation methods which produce approximate results and the error needs to be minimum. There are various techniques of mesh generation which are either automatic or manual [1]. Automatic mesh generators often generate meshes that are not well-shaped, and hence, a mesh smoothing technique is essential for mesh quality improvement.

Smoothing or *r* refinement method is a technique of correcting the poorly shaped elements. No one smoothing technique works all of the time [2]. Mesh smoothing provides a high quality of mesh and increases reliability of the solution. The most popular technique is Laplacian smoothing which uses internal nodes of the element and its connected nodes for relocation of nodes at desired place via centroid approach [3].

Knupp [4] has described the mesh quality metrics in the algebraic form which can be used as the objective functions for optimization-based smoothing. Element quality is a function of invertibility, size, and shape. Invertibility gives the measure of the positive local volume of given mesh. The quality of a mesh is defined in terms of these mesh metrics [4]. Knupp [5] proved that mesh element shape can be defined by using condition number parameter and can be improved by aspect ratio improvement and element skew reduction [5]. The mesh metrics generally used are inverse mean-ratio metric [6], aspect ratio metric [7], condition number [8], and distortion metric [9] etc.

In the present work, we discuss the application of cohort intelligence (CI) algorithm for developing a new efficient mesh smoothing technique. Objective function for individual qualities of hexahedral elements in the mesh based on the condition number is used as a quality measure. CI is based on natural tendency of the individuals or candidates, to improve its behaviour by observing and implementing behaviour of peers in the group known as cohort. This helps in the growth of overall cohort behaviour as every candidate absorbs certain qualities from one another [10].

2 Literature Review

Finite element analysis is a very powerful tool and a necessity of mechanical engineering applications like structural analyses and fluid dynamics but its usefulness is controlled by mesh quality. Meshing is a critical step in FEA and is generally carried out in two stages—point placement and mesh improvement. It is an iterative process. Kovalev [11] has shown that auto-generation of good-quality quadrilateral/hexahedral meshes is difficult [11]. Various methods have been developed for improvement of triangular, quadrilateral, tetrahedral, or hexahedral meshes. Laplace smoothing is suitable only for 2D meshes but does not guarantee good element quality 3D meshes. It produces slivered tetrahedron for 3D elements [12].

Mesh smoothing algorithms also apply optimization techniques for movement of nodes [13]. Knupp [4] explained how the quality of mesh is an important aspect in mesh smoothing algorithms and can be improved by numerical optimization of the mesh metric [4]. The optimization methods are better than Laplacian smoothing methods as they remove inverted elements on mesh improvement but are more expensive and time-consuming than Laplacian smoothing [14]. Hence, combinations of both result in effective tools for mesh improvement [8].

Both classical and heuristic techniques have been used in the literature for optimization-based smoothing. Freitag and Ollivier [15] proposed a technique which used minimum angle metric and iterative steepest descent for optimization [15]. Other classical optimization algorithms used are Newton's method, Hessian method [7], steepest gradient method [16], and linear programming [17]. Holder et al. (1998) have employed genetic algorithm for mesh optimization using distortion mesh criteria [18]. Acikgoz et al. [19] used simulated annealing technique for optimization [19]. Dittmer et al. [20] used optimization to control mesh creation parameters. [20]. Sastry and Shontz [7, 17] discussed the performance of nonlinear optimization methods for mesh quality improvement and showed that optimization solver behaviour can be improved by varying mesh size and accuracy level [7, 17].

Knupp [5] discussed condition number as a shape quality measure for hexahedral elements designed using a set of Jacobian matrices related to the particular element [5]. Yilmaz and Kuzuoglu [8] used particle swarm optimization technique for hexahedral mesh smoothing using condition number. The shape quality of all hexahedral elements is taken into account to design objective function which is modified in order to get the minimum value of objective function as zero [8].

The proposed work aims to apply condition number optimization for solving mesh smoothing using CI algorithm. The CI algorithm was proposed by Kulkarni et al. [10] and is inspired from the self-supervised learning behaviour of the candidates in a cohort [10]. All of them have the same target, and to achieve it, every candidate tries to study the behaviour of peers and improve by following a certain candidate's behaviour. Several parameters like sampling interval and reduction factor govern CI. The algorithm has been successfully applied in the areas of healthcare and logistics [21], knapsack problem [22], and mechanical design problems [23], and the present work investigates the application of CI for mesh smoothing. Throughout the journey, it will help to explore and validate various characteristics of CI [24].

3 Mesh Smoothing Algorithm Using Cohort Intelligence (CI)

Consider an objective function for hexahedral mesh smoothing (in the minimization sense) as described in Eq. 3.1. Here, k denotes the condition number of individual element and $T_{n,i}$ is the transformation matrix defined for the ith node of the nth hexahedral element. N denotes total number of elements in the problem. The attributes of the candidates are considered as the three coordinates of the eight vertices of the cube, i.e. $x, y, z = (x_1, y_1, z_1, \ldots, x_k, y_k, z_k, \ldots, x_8, y_8, z_8)$. Similarly $f(x^c y^c, z^c)$ denotes the behaviour of individual candidate $c(c = 1 \text{ to } C)$.

$$\text{Minimize } F = f(x, y, z) = \frac{1}{8N}\sum_{n}\sum_{i}\left(k\left(T_{n,i}\right)/3\right)^{2} - 1 \tag{3.1}$$

3.1 Procedure

The geometry of cube is generated using ANSYS and chosen for the preliminary analysis of the new algorithm proposed in this paper. The cube is discretized into 27 elements, 64 elements, and 729 elements, respectively. 27 elements and 64 elements have been chosen to investigate the applicability of the algorithm while 729 elements have been chosen for the comparison with previously published results. Creation of geometry file and auto-generated mesh using ANSYS APDL is the preliminary step. The analysis steps are as follows:

Step 1: (Conversion of data from ANSYS into a text file): Nodal connection data is obtained using **nlist** command and element data is obtained using **elist** command from hexahedral mesh.

Step 2: Extract node and element data: The data in this file is in complex form consisting of spaces, and alphanumeric characters. It contains data about element connectivity, real constants, and material assignment. Geometric coordinates of the nodes and elements need to be extracted from this huge piece of information. This data is initially stored in an excel file which is then extracted using textscan command. Then, the mesh data is read line by line for the removal of entries like NaN and zero values from the file. Then, the element connectivity in all directions (i.e. x, y, z) is used to define solution space.

Step 3: Develop the boundary of the object in MATLAB by edge and surface identification: The boundary of the object is required to be defined. So from the set of nodes extracted, the nodes lying on the surfaces, edges, and corners are identified. The equations of the planes for the surfaces of the cubes are developed using the three-point form of the equation of the plane. Redundant planes are excluded using the perpendicularity condition. Edges of the cube are identified as the intersection of the planes. This helps us to define external boundary of object.

Step 4: Identify the mesh metric for optimization which is the condition number: Let (x_i, y_i, z_i) be the vertices of the adjacent corners of a vertex (x, y, z), then its condition number (Eq. 3.2) is given by

$$k(T) = \text{cond}\begin{bmatrix} x_1 - x & x_2 - x & x_3 - x \\ y_1 - y & y_2 - y & y_3 - y \\ z_1 - z & z_2 - z & z_3 - z \end{bmatrix} \tag{3.2}$$

Condition number is calculated for each corner. The maximum value represents the condition number of the element which needs to be minimized. Store the minimum condition number candidate as global best.

Step 5: Check for bad-quality element: The element is said to have good quality if condition number lies between 1 and 3 [5]. The candidates C (number of bad elements in auto-generated mesh data) are initialized. The bounds of the variables are given by Eq. 3.3

$$\begin{aligned} x_i\,(\text{solution space})_{\text{lower}} &< x_i < x_i\,(\text{solution space})_{\text{upper}} \\ y_i\,(\text{solution space})_{\text{lower}} &< y_i < y_i\,(\text{solution space})_{\text{upper}} \\ z_i\,(\text{solution space})_{\text{lower}} &< z_i < z_i\,(\text{solution space})_{\text{upper}} \end{aligned} \tag{3.3}$$

Step 6: The behaviour selection probability p_c is calculated using Eq. 3.4. The behaviour of every associated candidate $c(c = 1, \ldots, C)$ to be followed is given by $f^*(x^c y^c, z^c)$, thus

$$p_c = \frac{1/f^*(x^c, y^c, z^c)}{\sum_c 1/f^*(x^c, y^c, z^c)}, (c = 1 \ldots C) \tag{3.4}$$

The roulette wheel approach is used by every candidate $c(c = 1\,\text{to}\,C)$ to generate random number to follow corresponding behaviour $f(x^{c[f]}, y^{c[f]}, z^{c[f]})$ and associated qualities $x^{c[f]}, y^{c[f]}, z^{c[f]} = \left(x^{c[f]}, y^{c[f]}, z^{c[f]} = x_1^{c[f]}, y_1^{c[f]}, z_1^{c[f]}, x_2^{c[f]}, y_2^{c[f]}, z_2^{c[f]} \ldots x_8^{c[f]}, y_8^{c[f]}, z_8^{c[f]}\right)$

Step 7: Every candidate shrinks or expands attributes depending on element edge length of candidate to be followed. The relations in Eq. 3.5 are obtained using condition number criteria. Specific nodes are not permitted to move to preserve the shape of volume mesh. Corner nodes are fixed. Surface nodes are allowed to move along surface only. Edge nodes are allowed to move along edges only.

$$\begin{aligned} x_i &= x_1^{c[f]} - x_{i\,(\text{edge})^{c[f]}} + x_{i-1(\text{edge})^{c[f]}} \\ y_i &= y_1^{c[f]} - y_{i\,(\text{edge})^{c[f]}} + y_{i-1(\text{edge})^{c[f]}} \\ z_i &= z_1^{c[f]} - z_{i\,(\text{edge})^{c[f]}} + z_{i-1(\text{edge})^{c[f]}} \end{aligned} \tag{3.5}$$

Step 8: Decide cohort behaviour to be followed by candidates and improve the quality of mesh using CI. Each candidate $c(c = 1, \ldots, C)$ tries to enrich its behaviour by following the other candidates having better behaviour associated with modified attributes $xc^{[f]}, yc^{[f]}, zc^{[f]}$. This makes the cohort available with C updated behaviours represented as $F^C = \left(f\left(x^1, y^1, z^1\right), \ldots, f(x^c, y^c, z^c)\right)$. After few learning attempts, all candidates follow global best behaviour from available behaviours which are helpful for internal mesh smoothing of hexahedral elements.

Step 9: Recheck mesh quality: The results using surface mesh plots before and after mesh improvement are compared. The objective function for global mesh quality improvement is evaluated. The cohort behaviour could be considered to be sat-

urated if there is no significant improvement in $f^*(x^c y^c, z^c)$ of every candidate $c(c = 1, \ldots, C)$ in the cohort, and the difference between individual behaviours is not very significant for a large number of iterative learning attempts, then the convergence is achieved.
Step 10: Import the improved mesh to ANSYS. The complete flow is summarized in flowchart 1.

4 Results and Discussions

The problem was coded in MATLAB R2016a on Windows Platform with Intel Core processor with 4 GB RAM and plot mesh function developed by [25]. Mesh improvement can be validated by checking the condition number improvement within range 1–3 for number of nodes of modified mesh data.

4.1 Stage-Wise Improvements

Mesh Improvement in stage I: In the proposed work, stage 1 is 3D mesh improvement using roulette wheel approach for deciding following behaviour of candidates in the cohort. Table 1 shows the condition number analysis for the Mesh with 27 elements. Several nodes do not lie within the range of 1–3 after improvement, so current scheme needs more refinement in further stages.

Mesh Improvement in stage II: Stage 2 is 3D or overall mesh improvement using global best behaviour to be followed by all candidates in the cohort. As shown in Table 2, the maximum number of nodes lies within 1–3 after mesh improvement. It proves that current optimization-based smoothing; that is, cohort algorithm gives promising results. The positive values show that elements are not inverted.

Table 1 Comparison based on condition number count

Condition number range (k)	Original number of nodes	Condition number optimization	
		Stage 1	Stage 2
1–1.5	35	79	216
1.5–2	82	93	0
2–3	56	42	0
3–4	16	2	0
4–5	11	0	0
5–8	12	0	0
8 and above	4	0	0

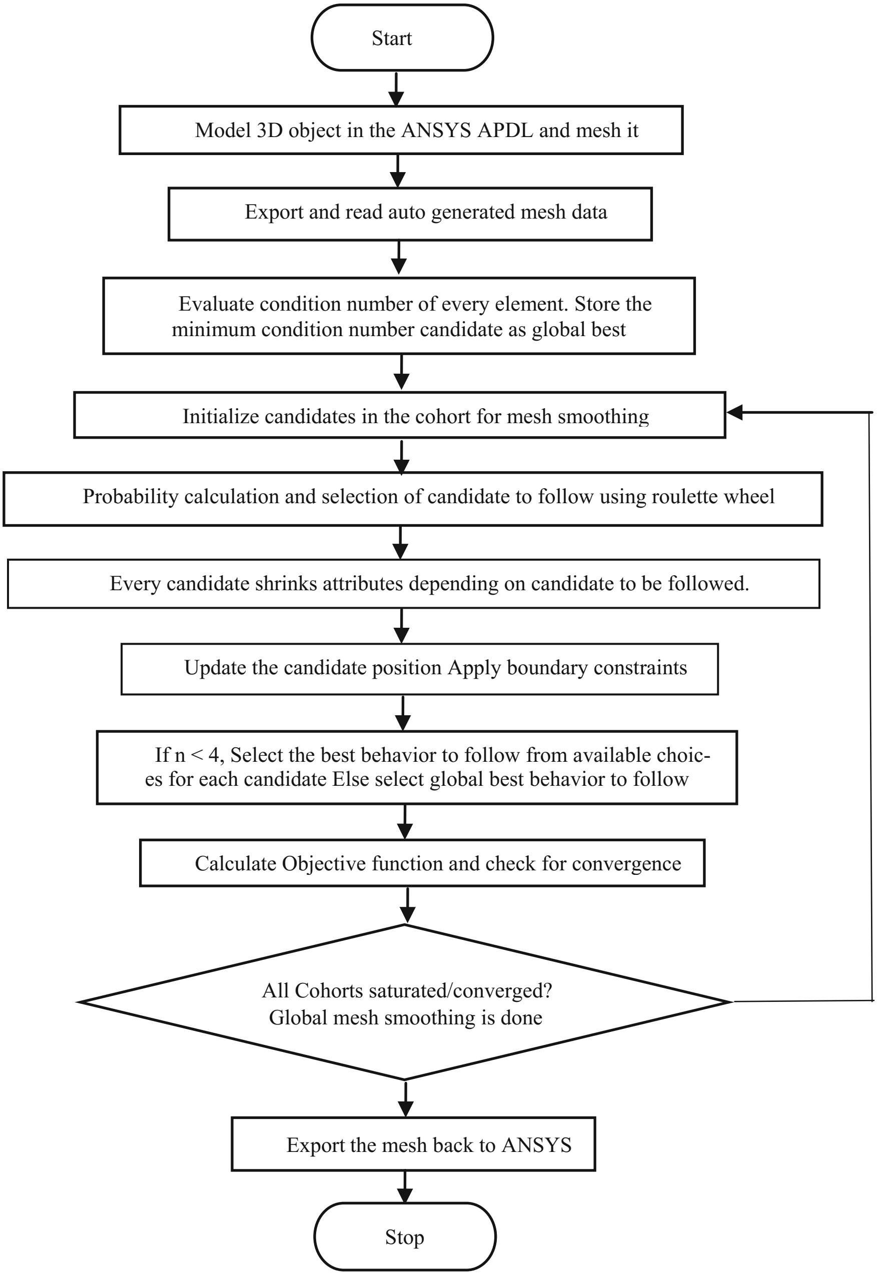
Start
Model 3D object in the ANSYS APDL and mesh it
Export and read auto generated mesh data
Evaluate condition number of every element. Store the minimum condition number candidate as global best
Initialize candidates in the cohort for mesh smoothing
Probability calculation and selection of candidate to follow using roulette wheel
Every candidate shrinks attributes depending on candidate to be followed.
Update the candidate position Apply boundary constraints
If n < 4, Select the best behavior to follow from available choices for each candidate Else select global best behavior to follow
Calculate Objective function and check for convergence
All Cohorts saturated/converged? Global mesh smoothing is done
Export the mesh back to ANSYS
Stop

Fig. 1 Flow chart for condition number optimization using CI

Table 2 Condition number reduction

Mesh data	Results	Condition number
27 elements, 64 nodes	Before	510.1303
	After	1.0117
64 elements, 125 nodes	Before	26.0429
	After	1.2136
729 elements, 1000 nodes	Before	12.1118
	After	1.7188

4.2 Graphical and Statistical Analysis

As shown in surface mesh plot followed by convergence plot in Figs. 2 and 3, the irregular mesh having 27 and 64 elements is smoothened using CI in two stages.

The graphical plot gives an idea about condition number optimization. Y axis gives a measure of objective function value and X axis refers to the learning attempts needed. Results are plotted for a various number of element data and all shows fine refinement after several learning attempts. The plots are also obtained for 729 elements. As results stated above shows about 95% improvement in the noisy version of mesh, we can say that CI can be used for hexahedral mesh smoothing.

The mean solution, worst solution, best solution, and standard deviation (Std. Dev.), mean run time (in seconds) over the 30 runs of the algorithm for three cases of mesh improvement problems are represented in Tables 2 and 3, respectively. Condition number is a node-based quality measure for hexahedral elements while the objective function shows the overall mesh improvement. If the value of condition number lies between 1 and 8, it is considered as acceptable range for meshing, but value between 1 and 3 is the good-quality measure for the same. Since the value of the highest condition number is close to 1 and standard deviation (σ) of the objective function is closer to 0, we can say that CI can be applied for mesh smoothing if objective function and condition number are taken into account.

5 Conclusions and Future Directions

A method for mesh improvement based on condition number optimization by using CI is validated in this work. Hexahedral meshes of 3D for a cubical model of various element sizes can be smoothed with this technique. This work needs to be updated to handle complex geometries such as prismatic elements with hexahedral mesh, meshing of geometry with stress concentration regions, and enhancement in convergence time by means of adjustments in the candidate behaviour.

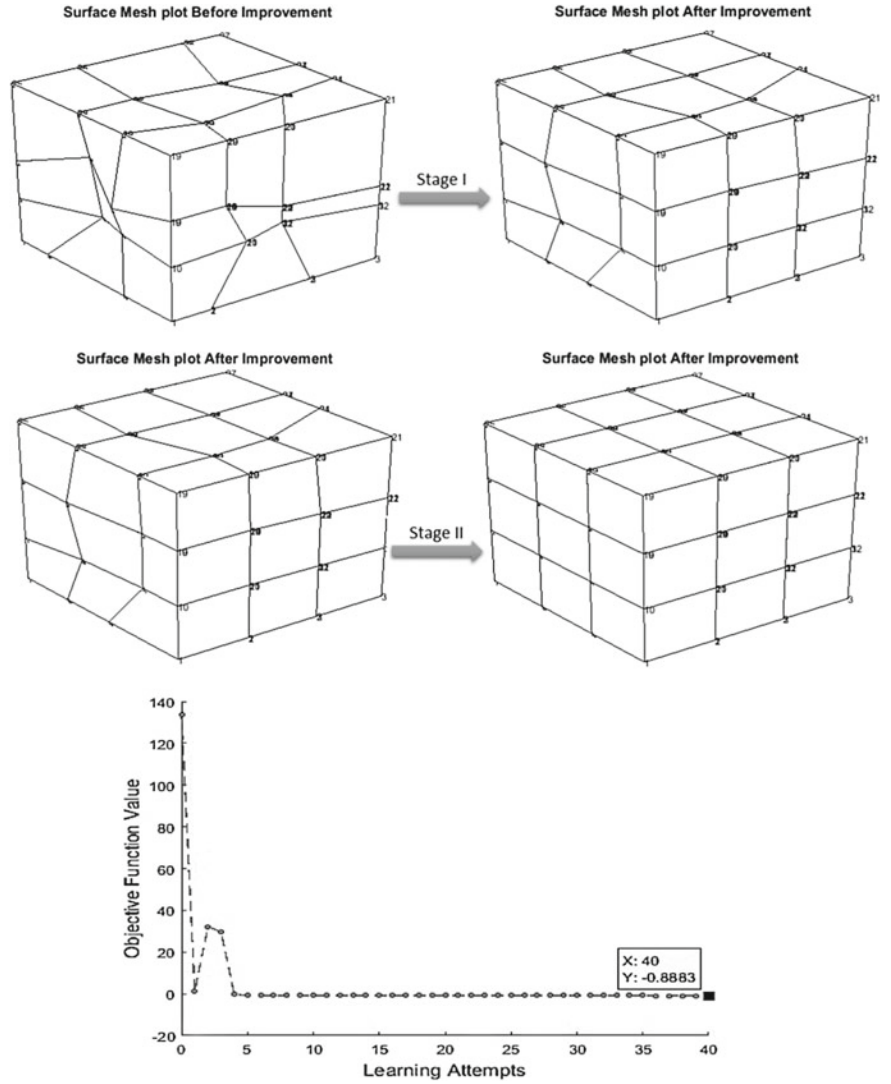

Fig. 2 Surface mesh plot and convergence plot: 27 elements, 64 nodes

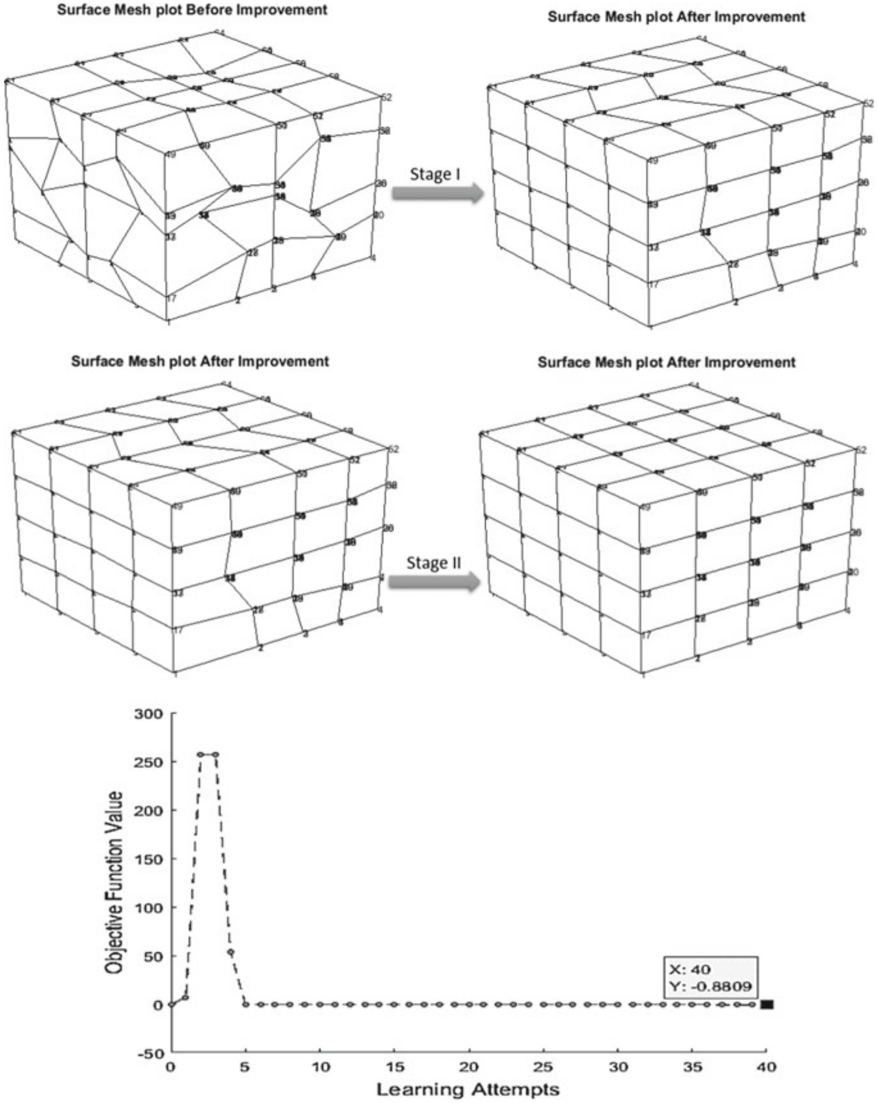

Fig. 3 Surface mesh plot and convergence plot: 64 elements, 125 nodes

Table 3 Condition number-based robustness check

Mesh data	Results	Condition number	σ
27 elements, 64 nodes	Worst	1.0117	0.001429098
	Best	1.0014	
	Mean	1.0065	
64 elements, 125 nodes	Worst	1.2136	0.006551462
	Best	1.0045	
	Mean	1.1041	
729 elements, 1000 nodes	Worst	1.7188	0.00012563
	Best	1.0216	
	Mean	1.3702	

References

1. Ho-Le K (1988) Finite element mesh generation methods: a review and classification. Comput Aided Des 20(1):27–38
2. Bern M, Eppstein D (1992) Mesh generation and optimal triangulation. Comput Euclidean Geom 1:23–90
3. Field DA (1988) Laplacian smoothing and Delaunay triangulations. Int J Numer Methods Biomed Eng 4(6):709–712
4. Knupp PM (2001) Algebraic mesh quality metrics. SIAM J Sci Comput 23(1):193–218
5. Knupp PM (2003) A method for hexahedral mesh shape optimization. Int J Numer Meth Eng 58(2):319–332
6. Munson T (2007) Mesh shape-quality optimization using the inverse mean-ratio metric. Math Program 110(3):561
7. Sastry SP, Shontz SM (2012) Performance characterization of nonlinear optimization methods for mesh quality improvement. Eng Comput 28(3):269–286
8. Yilmaz AE, Kuzuoglu M (2009) A particle swarm optimization approach for hexahedral mesh smoothing. Int J Numer Meth Fluids 60(1):55–78
9. Sarrate, J, Huerta A (2004) A new approach to minimize the distortion of quadrilateral and hexahedral meshes. In: 4th European congress on computational methods in applied sciences and engineering
10. Kulkarni AJ, Durugkar IP, Kumar M (2013) Cohort intelligence: a self-supervised learning behavior. In: 2013 IEEE international conference on systems, man, and cybernetics (SMC), pp 1396–1400
11. Kovalev K (2005) Unstructured hexahedral non-conformal mesh generation. Faculty of Engineering, Vrije Universiteit Brussel
12. Amenta N, Bern M, Eppstein D (1999) Optimal point placement for mesh smoothing. J Algorithms 30(2):302–322
13. Parthasarathy VN, Kodiyalam S (1991) A constrained optimization approach to finite element mesh smoothing. Finite Elem Anal Des 9(4):309–320
14. Canann SA, Tristano JR, Staten ML (1998) An approach to combined Laplacian and optimization-based smoothing for triangular, quadrilateral, and quad-dominant meshes. InIMR, pp 479–494
15. Freitag LA, Ollivier-Gooch C (1997) Tetrahedral mesh improvement using swapping and smoothing. Int J Numer Meth Eng 40(21):3979–4002

16. Freitag L, Jones M, Plassmann P (1999) A parallel algorithm for mesh smoothing. SIAM J Sci Comput 20(6):2023–2040
17. Sastry SP, Shontz SM (2009) A comparison of gradient-and hessian-based optimization methods for tetrahedral mesh quality improvement. In: Proceedings of 18th international meshing roundtable, pp 631–648
18. Holder M, Richardson J (1998) Genetic algorithms, another tool for quad mesh optimization. In: IMR, pp 497–504
19. Acikgoz N, Bottasso CL, Detomi D (2004) Metric based mesh optimization using simulated annealing. In: 4th European congress on computational methods in applied sciences & engineering ECCOMAS 2004
20. Dittmer JP, Jensen CG, Gottschalk M, Almy T (2006) Mesh optimization using a genetic algorithm to control mesh creation parameters. Comput-Aided Des Appl 3(6):731–740
21. Kulkarni AJ, Baki MF, Chaouch BA (2016) Application of the cohort-intelligence optimization method to three selected combinatorial optimization problems. Eur J Oper Res 250(2):427–447
22. Kulkarni AJ, Shabir H (2016) Solving 0–1 knapsack problem using cohort intelligence algorithm. Int J Mach Learn Cybernet 7(3):427–441
23. Kulkarni O, Kulkarni N, Kulkarni AJ, Kakandikar G (2016) Constrained cohort intelligence using static and dynamic penalty function approach for mechanical components design. Int J Parallel Emergent Distrib Syst, pp 1–19
24. Kulkarni AJ, Krishnasamy G, Abraham A (2017) Cohort intelligence: a socio-inspired optimization method. Springer, Heidelberg
25. Kolukula SS (2016) PlotMesh source code (version 2), Mathsworks (Source code) https://sites.google.com/site/kolukulasivasrinivas/