

Automatic Delaunay mesh generation method and physically-based mesh optimization method on two-dimensional regions

Quanbing Luo^{a,*}

^a*Guangzhou Key Laboratory of Environmental Catalysis and Pollution Control, Guangdong Key Laboratory of Environmental Catalysis and Health Risk Control, School of Environmental Science and Engineering, Institute of Environmental Health and Pollution Control, Guangdong University of Technology, Guangzhou 510006, China*

Abstract

Delaunay mesh generation method is a common method for unstructured mesh (or unstructured grid) generation. Delaunay mesh generation method can conveniently add new points to existing mesh without remeshing the whole domain. However, the quality of the generated mesh is not high enough if compared with some mesh generation methods. In order to obtain high-quality mesh, this paper developed an automatic Delaunay mesh generation method and a physically-based mesh optimization method on two-dimensional regions. For the Delaunay mesh generation method, boundary-conforming problem was ensured by create nodes at centroid of mesh elements. The definition of node bubbles and element bubbles was provided to control local mesh coarseness and fineness automatically. For the physically-based mesh optimization method, the positions of boundary node bubbles are predefined, the positions of interior node bubbles are adjusted according to interbubble forces. Size of interior node bubbles is further adjusted according to size of adjacent node bubbles. Several examples shows that high-quality meshes are obtained after mesh optimization.

Keywords: mesh generation; unstructured grid generation; Delaunay triangulation; mesh optimization.

¹ 1. Introduction

*Corresponding author.

Email address: quanbing.luo@foxmail.com (Quanbing Luo)

2 Unstructured mesh (or called as unstructured grids) are widely used in finite volume
3 method of computational fluid dynamics for complex geometries [1] (pp. 311-336). How-
4 ever, for the discretization of the diffusion term under finite volume method [1] (pp. 316-320),
5 mesh skewness or non-orthogonality (low-quality mesh) would bring errors during discretiza-
6 tion, and this error increases with increasing skewness and aspect ratio. Thus, it is important
7 that every effort is made to control skew and aspect ratios in unstructured mesh generation.

8 Delaunay mesh generation method is a common method for unstructured mesh genera-
9 tion [2](pp. 450-468). Delaunay mesh generation method can conveniently add new points
10 to existing mesh without remeshing the whole domain, which makes it particularly suit to
11 a self-adaptive solution strategy [3]. However, the quality of the generated mesh is not high
12 enough if compared with some mesh generation methods, such as bubble packing method.
13 Bubble packing method is a physically-based mesh generation method which was provided
14 by Shimada [4]. Previously, some paper improved bubble packing method from different
15 aspects. Shimada and Gossard [5] solved the two basic problems of triangulation – where
16 to place nodes and how many nodes to place in the domain – via dynamic simulation with
17 attractive/repulsive interbubble forces and adaptive bubble population control. Wu et al.
18 [6] improved bubble packing method from the aspects of grid density of the whole region and
19 local mesh refinement. Qi et al. [7] and Guo et al. [8] provided some acceleration strategies
20 for bubble packing method. Wang et al. [9] provided a bubble-like method which is applica-
21 ble to various surfaces in high automation level. In my mind, bubble packing method is not a
22 perfect mesh generation method though this method can generate high-quality unstructured
23 meshes. In fact, the method just provided a method to generate the interior nodes for the
24 giving geometries, it still needs other method (such as Delaunay method) to connect these
25 nodes.

26 As Delaunay method can generate mesh independently and automatically (though of low
27 quality), this paper think that optimizing the mesh generated by Delaunay method could
28 bring high-quality mesh conveniently than bubble packing method. For mesh optimization,
29 there are many different methods. The optimization methods can be broadly grouped into
30 two categories [10] (pp. 327-383): (i) optimization by shifting of nodes; (ii) optimization

31 by topological operations . The mesh optimization methods by shifting of nodes are often
32 based on geometry properties, such as Laplace smoothing methods [11–13]. For topologi-
33 cal optimization operations [14–16], swap of diagonals and elimination of short edges and
34 small elements are the common strategies employed. These mesh optimization methods
35 mentioned above are geometric and topological methods. As high-quality mesh could be
36 generated by physically-based methods [4, 5], this paper think mesh optimization could also
37 be implemented with physically-based methods after Delaunay mesh generation.

38 In my previous paper [17], discretized pressure Poisson algorithm was successfully applied
39 on triangular unstructured mesh. In the paper, mesh was generated from MATLAB Partial
40 Differential Equation Toolbox, which is a black box. This paper would provide an automatic
41 Delaunay mesh generation method and a physically-based mesh optimization method on
42 two-dimensional regions. High-quality meshes would be obtained after mesh optimization.

43 **2. Automatic Delaunay mesh generation method**

44 *2.1. Boundary-conforming problem and initial boundary mesh*

45 Boundary-conforming problem (boundary integrity problem) is a common problem of
46 Delaunay mesh generation. If no special method is applied, the resulting meshes may not
47 coincide with the boundaries of the computational domain. Previously, some papers [18–20]
48 solve the boundary-conforming problem by generating initial boundary mesh with different
49 methods. In these methods, refined boundary nodes and their connective information should
50 be provided before generating initial boundary mesh. Meanwhile, constrained Delaunay
51 method [21] (pp. 46–52) is a widely-used method in computational geometry and mesh
52 generation. According to this method, this paper would generate initial boundary mesh with
53 “delaunayTriangulation” and “isInterior” function of MATLAB. Then, creation of interior
54 nodes and node insertion would be the next steps for Delaunay mesh generation.

55 *2.2. Node insertion with Delaunay triangulation method*

56 Delaunay triangulation method just provides a way to connect points, but not provide
57 the method of generating nodes within computational domain. There are some different

58 methods for the creation of interior nodes, such as refinement method, use of background
 59 grid and other techniques [10] (pp. 111-121). For refinement method, interior nodes could
 60 be created at circumcenter, centroid and edge. This paper would use the refinement method
 61 and creat interior points at centroid. The reason is that: centroid would be always inside
 62 of triangular element, boundary-conforming problem would be always satisfied during node
 63 insertion procedures of mesh generation.

64 If a interior node has been provided, there are still three steps for the node insertion
 65 with Delaunay triangulation method (according to Lo [10] (pp. 97-98) and Borouchaki and
 66 Lo [22]): (1). identify the triangles whose circumcircles contain the new node; (2) delete
 67 common edges of these triangles (obtain a cavity); (3) obtain new triangles from the node
 68 and the cavity. Figure 1 shows the node insertion with Delaunay triangulation method.

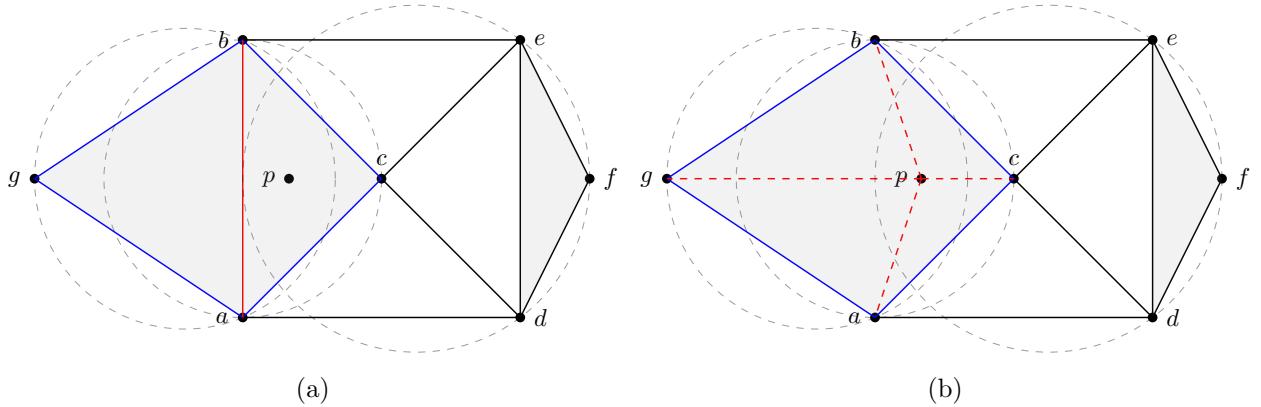


Figure 1: Point p insertion with Delaunay triangulation method.

69 Figure 1a shows the steps to identify the triangles whose circumcircles contain the node
 70 p and identify common edges of these triangles. A simple method to find all the triangles
 71 whose circumcircle contains the node p is to scan through all the existing triangles. There
 72 are three triangles whose circumcircles contain the node p : Δabc , Δabg and Δdef . However,
 73 as the Δdef is far from node p , there are no common edges between this triangle and other
 74 two triangles. The reason comes from that the original meshes are not Delaunay triangles.
 75 Thus, the cavity needed for Delaunay triangulation should not include the Δdef . The
 76 better approach to obtain the cavity is that: start the search from the triangle containing

77 the inserted centroid node p and then find the other triangles by means of the adjacency
 78 relationship [10] (pp. 97-98). In this way, the boundary of the polygon cavity a-c-b-g is
 79 given by the non-common edges of these adjacent triangles whose circumcircle contains the
 80 node p . Figure 1b shows the steps to delete common edges of these triangles and obtain new
 81 triangles from the node and the cavity. These steps is very simple, it just need to delete
 82 original triangles Δ_{abc} and Δ_{abg} and add new Δ_{pbc} , Δ_{pbg} , Δ_{pac} and Δ_{pag} .

83 *2.3. Automatic creation of interior nodes*

84 For the creation of interior nodes, there is still a problem: as there are many centroids for
 85 triangular elements, which centroid should be inserted with priority. Previously, Anderson
 86 [19] and Yu et al. [3] provided different strategies. Inspired by bubble packing method for
 87 mesh generation [5-7], this paper would create interior nodes at certain priority with the
 88 help of some bubbles. For the creation of interior nodes, the definition of node bubbles and
 89 element bubbles was provided for this paper.

90 *2.3.1. Node bubbles*

91 Node bubbles are bubbles whose centers are located at mesh nodes. For boundary nodes,
 92 the area of node bubble is defined as the mean area of two regular triangles A_1 and A_2 whose
 93 side lengths (l_1 and l_2) are the lengths of neighboring boundary edges.

$$A_n = \frac{1}{2} (A_1 + A_2) = \frac{1}{2} \left(\frac{\sqrt{3}}{4} l_1^2 + \frac{\sqrt{3}}{4} l_2^2 \right) \quad (1)$$

94 The above equation reflected that the areas of boundary node bubbles are close to the
 95 area of regular triangles of boundary edges. Then, the radius of boundary node bubble can
 96 be calculated from the following equation:

$$r_n = \sqrt{\frac{\sqrt{3}}{8\pi} (l_1^2 + l_2^2)} \quad (2)$$

97 For the node bubbles of interior nodes, as the interior nodes are created from centroids of
 98 triangular elements, the node bubbles would be generated form bubbles of interior centroids
 99 (element bubbles) after node insertion. Then, the next problem is what is the definition of
 100 element bubbles.

101 2.3.2. *Element bubbles*

102 Element bubbles are bubbles whose centers are located at the element centroids. For
103 the element bubbles, their radii can be calculated by interpolation of node bubbles. This
104 paper provided the following reciprocal interpolation equation for radii calculation of element
105 bubbles:

$$r_e = \frac{r_a/l_{ap} + r_b/l_{bp} + r_c/l_{cp}}{1/l_{ap} + 1/l_{bp} + 1/l_{cp}} \quad (3)$$

106 Where, vertices a , b and c are the vertices of a mesh triangle, e is its centroid. r_e , r_a , r_b
107 and r_c are the radii of the points e , a , b and c respectively; l_{ae} , l_{be} , and l_{ce} are the length
108 between these vertices respectively. This interpolation rule enhances the effect of boundary
109 nodes distribution character to the interior nodes coarseness [3]. Then, the area of element
110 bubble can be calculated with the following equation:

$$A_e = \pi r_e^2 \quad (4)$$

111 The area of element bubble reflects the size of the mesh element it should have during
112 mesh generation.

113 2.3.3. *Bubble density*

114 This paper defined density of element bubbles to control local mesh density according to
115 following equation:

$$D_e = \frac{A_e}{A_m} \quad (5)$$

116 In the above equation, A_m is the area of triangular mesh element. Apparently, if the
117 value of relative mesh density D_e is too small, the mesh element shoud be refined further.

118 In the refinement process, a new interior node would be created at centroid of the element
119 which has smallest bubble density.

120 2.3.4. *Mesh population control*

121 There are many different methods to control the population of mesh. For Delaunay
122 method, Yu et al. [3] and Anderson [19] controlled the population of mesh according to

123 dimensionless circumcircle radius and maximum aspect ratio, respectively. For bubble pack-
124 ing method, Shimada and Gossard [5] and Wu et al. [6] controlled the population of mesh
125 according to overlapping ratio.

126 This paper developed a mesh population control method according to the area of element
127 bubbles. Mesh population control can be realized by that: the total area of element bubbles
128 is larger than the area of whole geometry region. That means if the following equation is
129 satisfied, the process of creation and insertion of interior nodes should be stopped.

$$\sum A_e \geq A_{geo} \quad (6)$$

130 In the above equation, A_{geo} is the area of whole geometry region.

131 2.4. Procedures of the Delaunay mesh generation method

132 The procedures of the automatic Delaunay mesh generation method include several steps,
133 these steps are listed as follows.

- 134 (a) Generate initial boundary mesh with refined boundary (solve boundary-conforming
135 problem);
- 136 (b) Compute the node bubbles, element bubbles and bubble density for the mesh;
- 137 (c) Search the mesh element whose bubble density is smallest;
- 138 (d) Creating a new node at the centroid of the triangular mesh element;
- 139 (e) Node insertion with Delaunay triangulation method (node bubbles, element bubbles
140 and bubble density should be updated simultaneously);
- 141 (f) Mesh population control: if $\sum A_e \geq A_{geo}$, stop, else, return to the step c.

142 In the next section, the mesh optimization method would be introduced, and the data
143 of node bubbles would be further used in the mesh optimization.

144 3. Physically-based mesh optimization method

145 The mesh optimization methods by shifting of nodes are often based on geometry prop-
146 erties, such as Laplace smoothing methods [11–13]. This paper would provide a physically-
147 based mesh optimization method, the method is also inspired by bubble packing mesh
148 generation method [5–7].

149 3.1. Initial node bubbles

150 Node bubbles can be divided into boundary node bubbles and interior node bubbles. For
151 the physically-based mesh optimization method, the positions and sizes of boundary node
152 bubbles are predefined, the positions and sizes of interior node bubbles can be adjusted
153 according to interbubble forces. In this paper, only repulsive forces are provided for the
154 interbubble forces among bubbles. Thus, the giving radii of initial node bubbles in the mesh
155 optimization should be large enough to make them overlapped partly.

156 In this paper, the initial node bubbles of physically-based mesh optimization method are
157 calculated from the node bubbles of the Delaunay mesh generation method in the previous
158 section. As the sizes of node bubbles in Delaunay mesh generation are too small, the radii
159 of these node bubbles were amplified to 1.7 times in the physically-based mesh optimization
160 to make these initial node bubbles overlapped partly.

161 3.2. Position adjustment of interior nodes

162 In this paper, position adjustment of interior nodes is based on force equilibrium [23, 24],
163 the new node position \mathbf{x}_i^{n+1} could be calculated with the following equation.

$$\mathbf{x}_i^{n+1} = \mathbf{x}_i^n + \alpha \sum_j \mathbf{f}_{i,j} \quad (7)$$

164 In the above equation, $\mathbf{f}_{i,j}$ is the interbubble forces between two bubbles, α is an impor-
165 tant parameter to control the new interior node position in each iteration, its value is given
166 as 0.1 in this paper. The next problem is how to set the interbubble forces.

167 In the bubble packing mesh generation method [5–7], for the interbubble forces, repulsive
168 forces and attractive forces were considered simultaneously. In this paper, only repulsive
169 forces is considered for position adjustment of interior nodes, which is similar to the mesh
170 generation method of Persson and Strang [23]. That is the reason why the node bubbles
171 of Delaunay mesh should be partly overlapped in the previous subsection. The equation of
172 interbubble forces is given as follows.

$$\mathbf{f}_{i,j} = \begin{cases} \sum_j (r_i + r_j - l_{ij}) \mathbf{n}_{ij}, & (r_i + r_j > l_{ij}) \\ 0, & (r_i + r_j \leq l_{ij}) \end{cases} \quad (8)$$

173 In the above equation, r_i and r_j are the radii of current bubble and nearby bubble. \mathbf{n}_{ij} is
 174 the unit vector from current node bubble to nearby node bubble. The repulsive forces are
 175 similar to spring forces in vibration problems [25] (pp. 643-666)

176 *3.3. Nearby node bubbles*

177 In the previous subsection, for position adjustment of interior node bubbles, nearby
 178 node bubbles of giving node should be provided for calculation. As nearby node bubbles of
 179 current node bubble would change during the process of mesh generation, there are different
 180 strategies to obtain nearby node bubbles. In the bubble packing mesh generation method
 181 [5–7], all other node bubbles were regarded as the nearby node bubbles. Thus, it was very
 182 inefficient for the calculation of new position of interior node bubbles. In the paper of Persson
 183 and Strang [23], the nearby node bubbles were identified by the topological relationship of
 184 mesh. As Delaunay method was needed to get new topological relationship before each step
 185 of interior nodes position adjustment, the mesh generation method is also very inefficient.

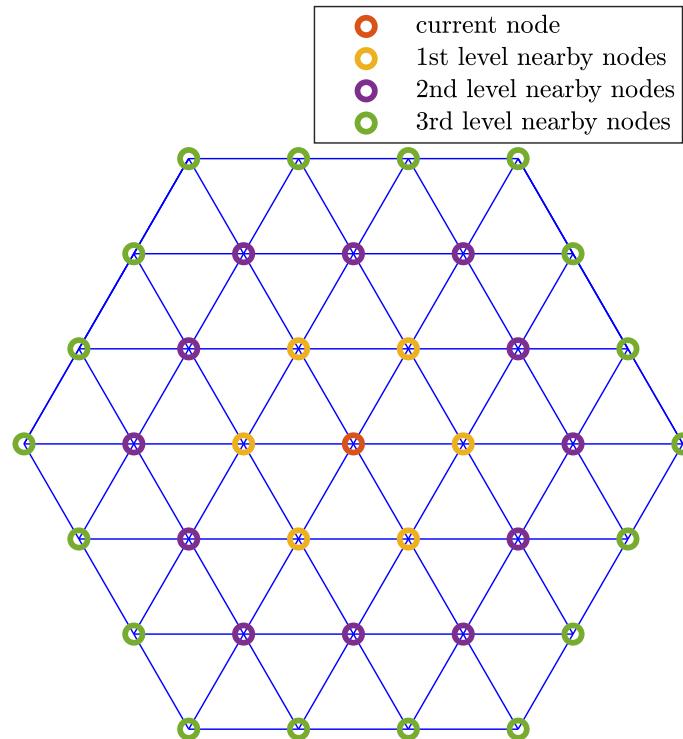


Figure 2: The current node and its 1st, 2nd and 3rd level nearby nodes according to topological relationship.

186 In this paper, as mesh optimization method is used to optimize the mesh generated by
 187 Delaunay method, the position of interior nodes would be not adjusted greatly. Thus, in
 188 this paper, nearby node bubbles would be given as some unchangeable node bubbles which
 189 obtained according to topological relationship. Figure 2 shows the current node and its 1st,
 190 2nd and 3rd level nearby nodes according to topological relationship. In this paper, nearby
 191 node bubbles are defined as the total node bubbles of its 1st, 2nd and 3rd level nearby node
 192 bubbles.

193 *3.4. Size adjustment of node bubbles*

194 After position adjustment of interior nodes, if the size of there node bubbles were not
 195 adjusted further, the quality of generated mesh would be still very low. Thus, the size of
 196 interior node bubbles should be adjusted with the position adjustment of interior nodes.
 197 Previously, Shimada and Gossard [5] and Persson and Strang [23] provided their methods to
 198 control the selement size by size function. However, it was very hard to provide reasonable
 199 size function for complicated geometries. Persson [26] even write a paper to introduce how
 200 to generate the required size function.

201 This paper provide a simpler methods for size adjustment of node bubbles. The size of
 202 boundary node bubbles is predefined, the size of interior node bubbles can be adjusted by
 203 the size of adjacent node bubbles in the mesh optimization method. The radii of interior
 204 node bubbles can be calculated with reciprocal interpolation equation, which is similar to
 205 the interior point creation in the Delaunay mesh generation method [3].

$$r_p = \frac{\sum_i r_i / l_{p,i}}{\sum_i 1 / l_{p,i}}, (r_i + r_p > l_{p,i}) \quad (9)$$

206 In the above equation, the adjacent node bubbles are chosen from nearby node bubbles,
 207 these adjacent node bubbles should partly overlapped with the current node bubble. Thus,
 208 only those nearby node bubbles who fulfilled the equation $r_i + r_p > l_{p,i}$ are regarded as
 209 adjacent node bubbles. From the above equation, it can be known that the sizes of interior
 210 node bubbles are controlled by initial node bubbles (especially by boundary node bubbles).
 211 As there are similar reciprocal interpolation strategy of element bubbles in the Delaunay

212 mesh generation, the node bubbles would be still partly overlapped after size adjustment of
213 interior node bubbles (with the reciprocal interpolation) if the initial node bubbles are partly
214 overlapped. Obviously, the initial node bubbles have been ensured to be partly overlapped
215 during the first step of mesh optimization.

216 *3.5. Mesh generation with constrained Delaunay method*

217 As position of nodes have been adjusted, this paper would then generate mesh with
218 constrained boundaries with “delaunayTriangulation” and “isInterior” function of MAT-
219 LAB. The method is based on constrained Delaunay method [21] (pp. 46-52), which is a
220 widely-used method in computational geometry and mesh generation.

221 *3.6. Procedures of the mesh optimization method*

222 The procedures of the mesh optimization method include several steps, these steps are
223 listed as follows.

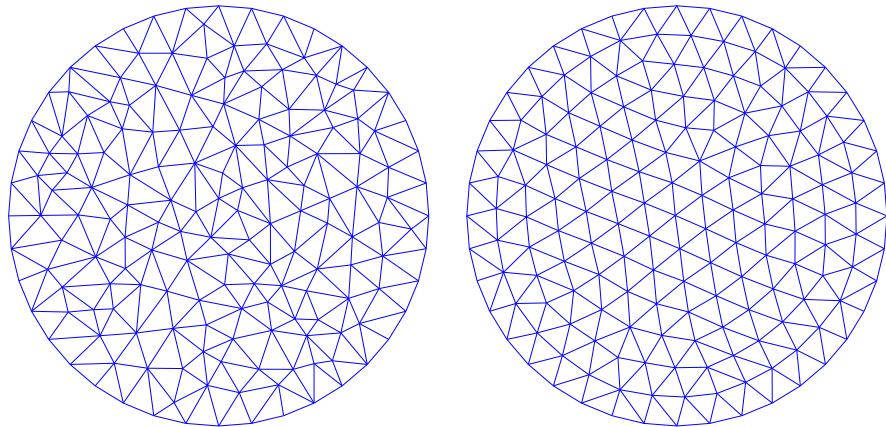
- 224 (a) Calculation of the radii of initial node bubbles and their nearby nodes
225 (b) Position adjustment of interior nodes;
226 (c) Size adjustment of node bubbles;
227 (d) Return to step (b) if more iterations are needed;
228 (e) Mesh generation with constrained Delaunay method;

229 **4. Results and discussion**

230 Figure 3, Figure 4, Figure 5 and Figure 6 shows the results of Delaunay mesh and
231 optimized mesh for different regions. Radius ratio was widely used to evaluate the mesh
232 quality [7, 23, 27, 28], the detailed introduction of radius ratio was given in the book of Lo
233 [10] (p. 331). According to the references [7, 23], the definition of radius ratio is giving as
234 follows.

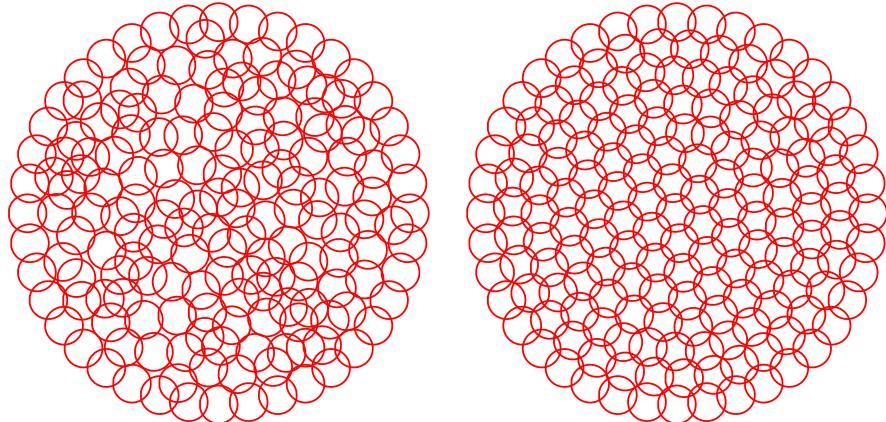
$$\rho = \frac{2r}{R} = \frac{(l_c + l_b - l_a)(l_c + l_a - l_b)(l_a + l_b - l_c)}{l_a l_b l_c} \quad (10)$$

235 In the above equation, r and R are inradius and circumradius of triangular mesh element;
236 l_1 , l_2 and l_3 are the side lengths of triangular mesh element.



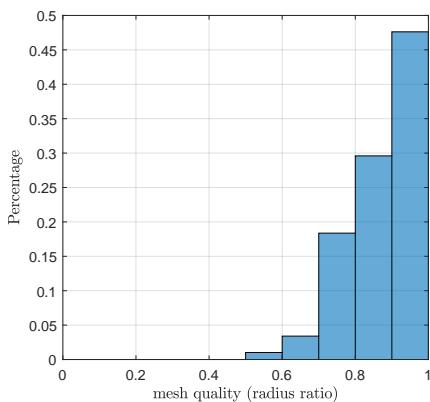
(a) Delaunay mesh

(b) Optimized mesh

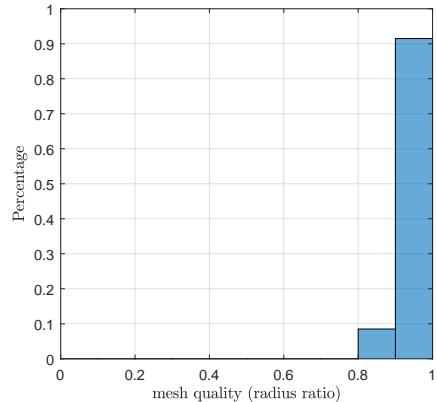


(c) Initial node bubbles

(d) Optimized node bubbles

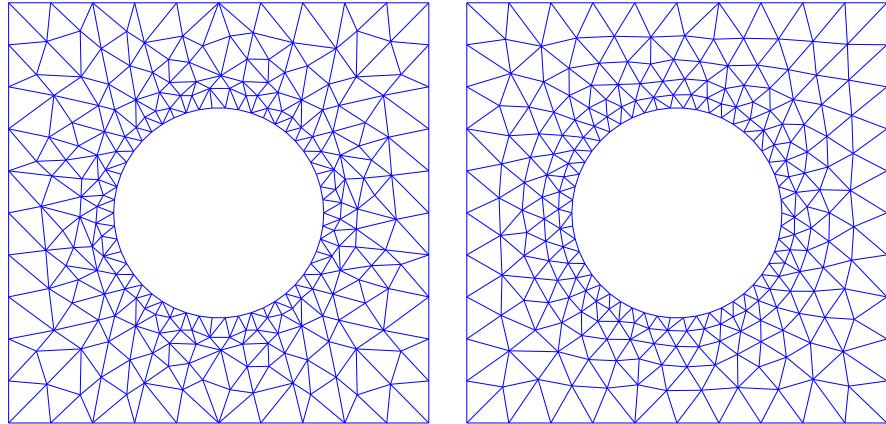


(e) Mesh quality (Delaunay mesh)



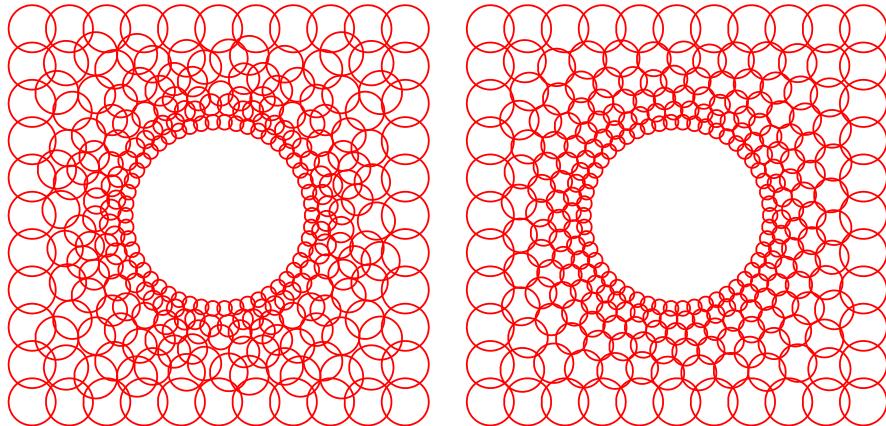
(f) Mesh quality (Optimized mesh)

Figure 3: Delaunay mesh and optimized mesh for circular region.



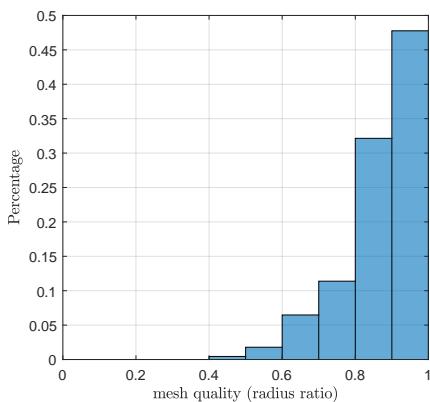
(a) Delaunay mesh

(b) Optimized mesh

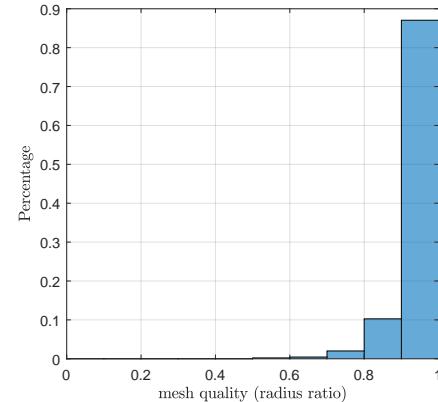


(c) Initial node bubbles

(d) Optimized node bubbles



(e) Mesh quality (Delaunay mesh)



(f) Mesh quality (Optimized mesh)

Figure 4: Delaunay mesh and optimized mesh for square region with a circular hole.

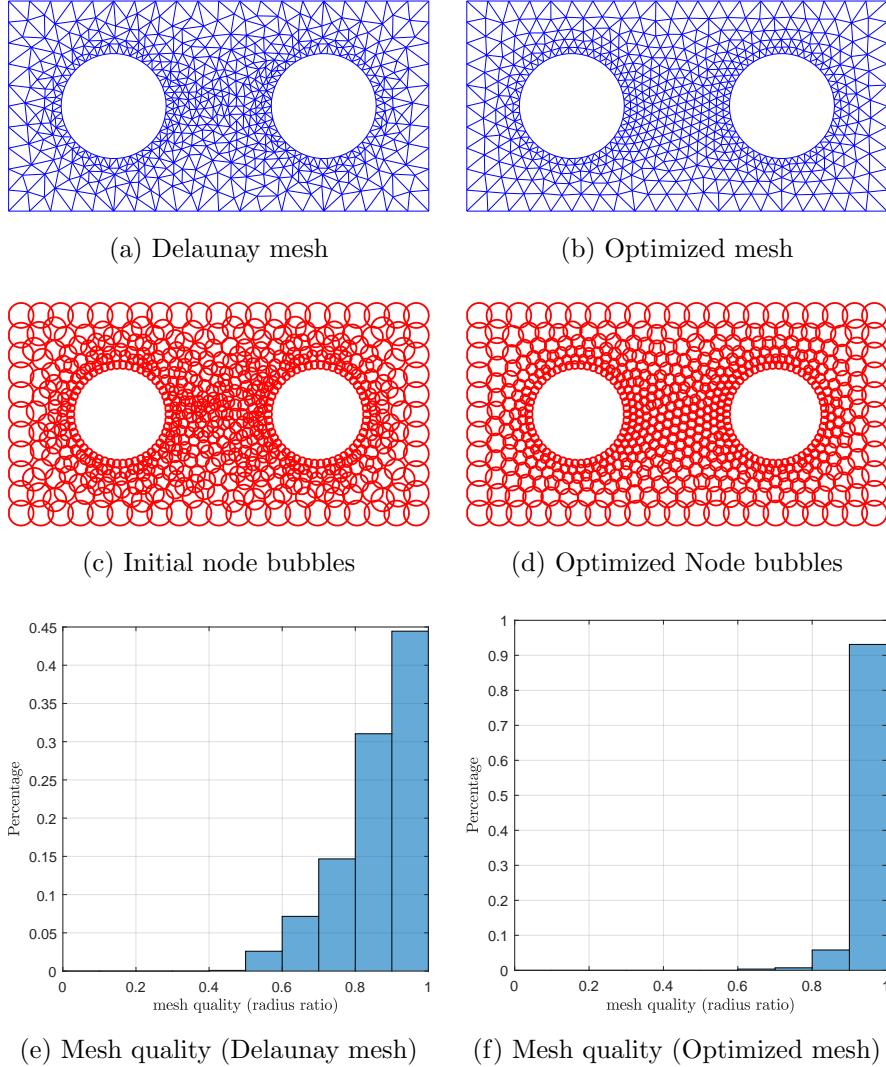


Figure 5: Delaunay mesh and optimized mesh for rectangular region with two circular hole.

From these figures, the quality of the mesh generated by Delaunay method is very low, which should be improved further by mesh optimization method. After mesh optimization, the positions of interior nodes are evidently optimized by physical model, high-quality meshes are then generated. From Figure 4 and Figure 5, the coarseness or fineness of the generated mesh can be well controlled by the distribution of boundary node bubbles. From Figure 6, the physically-based mesh optimization method is also feasible for the mesh of complicated geometries.

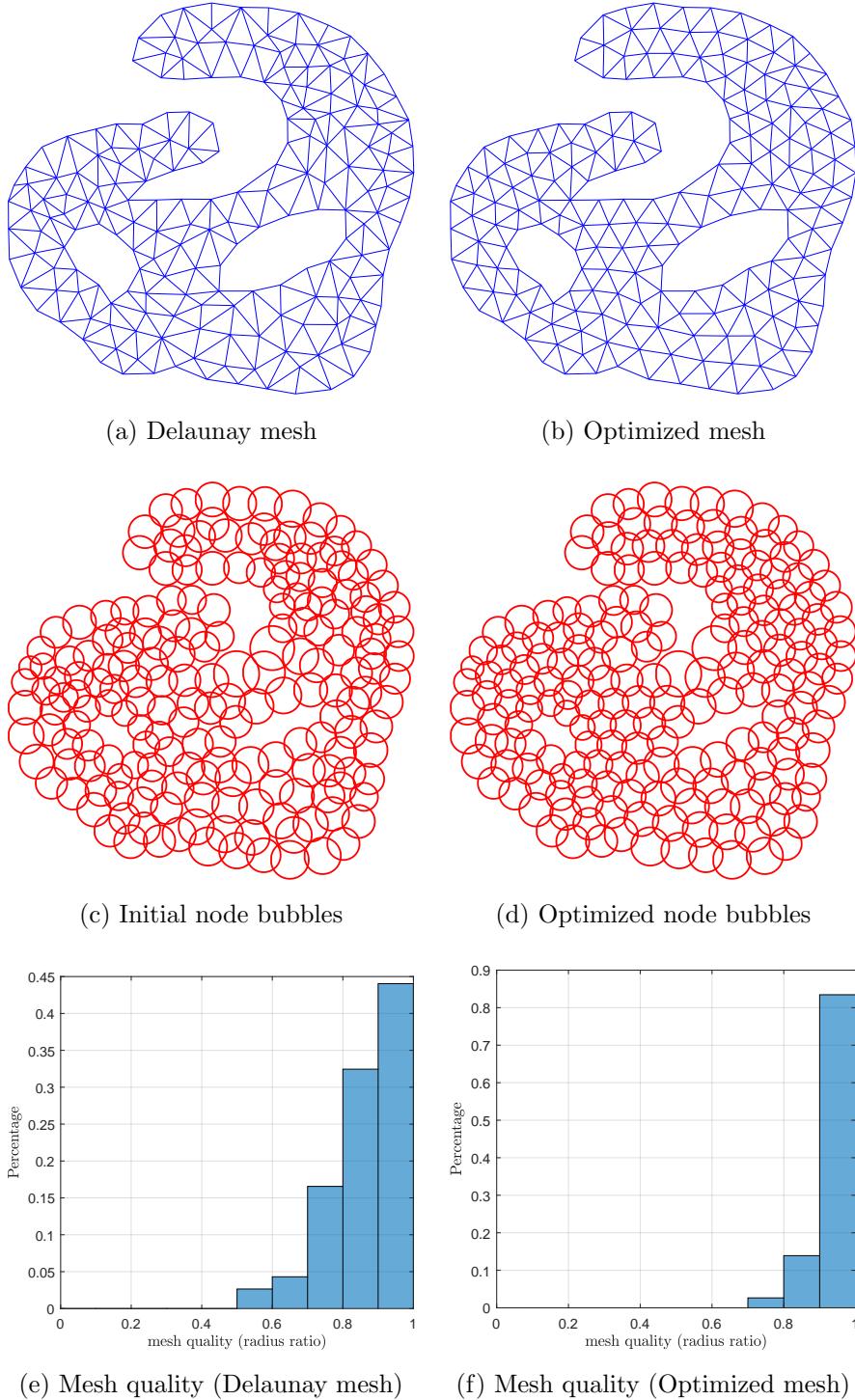


Figure 6: Mesh generation of arbitrary region with two hole.

244 However, from Figure 4 and Figure 5, there are still few elements with low radius ratio
245 (less than 0.7). The possible reasons are given as follows.

- 246 (1) The distance from some exterior boundaries (with coarse mesh) to some interior bound-
247 aries (with fine mesh) is too short, which makes the size of node bubbles between them
248 cannot change slowly. Then, that would generate few low-quality mesh elements between
249 them.
- 250 (2) The population of mesh nodes is not well controlled. Population control of mesh nodes
251 is the responsibility of mesh generation method. The population of mesh nodes was pro-
252 vided by mesh generation method in this paper. From the results of mesh optimization
253 method, the population of original Delaunay mesh nodes might have a little effect on
254 the mesh quality of optimized mesh.
- 255 (3) The size of node bubbles near interior boundaries (with fine mesh) is too large, which
256 makes there are more low-quality mesh elements. Though the size of node bubbles near
257 interior boundaries (with fine mesh) can be adjusted easily in this paper, there are still
258 few low-quality mesh elements in other place if the above two problems cannot well
259 solved.

260 For the listed three possible reasons, the first reason might be the main reason for few
261 low-quality mesh elements in this paper. If reasonable Delaunay mesh is provided for mesh
262 optimization, the method of this paper would certainly obtained optimized mesh with higher
263 quality.

264 **5. Conclusions**

265 This paper developed an automatic Delaunay mesh generation method and a physically-
266 based mesh optimization method on two-dimensional region. For the Delaunay mesh gen-
267 eration method, boundary-conforming problem was ensured by create nodes at centroid of
268 mesh. The definition of node bubbles and element bubbles was provided to control local
269 mesh coarseness and fineness automatically. For the physically-based mesh optimization
270 method, the positions of boundary node bubbles are predefined, the positions of interior

271 node bubbles are adjusted according to interbubble forces. Size of interior node bubbles is
272 further adjusted according to size of adjacent node bubbles. Several examples shows that
273 high-quality meshes are obtained after mesh optimization.

274 **Acknowledgments**

275 This work was supported by Guangdong University of Technology.

276 **References**

- 277 [1] H. K. Versteeg, W. Malalasekera, An introduction to computational fluid dynamics: the finite volume
278 method, 2nd Edition, Pearson, Harlow, 2007.
- 279 [2] V. D. Liseikin, Grid Generation Methods, third edition Edition, Springer, Cham, 2017. doi:10.1007/
280 978-3-319-57846-0.
- 281 [3] B. Yu, M. J. Lin, W. Q. Tao, Automatic generation of unstructured grids with delaunay triangulation
282 and its application, Heat and Mass Transfer 35 (5) (1999) 361–370. doi:10.1007/s002310050337.
283 URL <https://doi.org/10.1007/s002310050337>
- 284 [4] K. Shimada, Physically-based mesh generation: automated triangulation of surfaces and volumes via
285 bubble packing, Ph.D. thesis, Massachusetts Institute of Technology (1993).
286 URL <http://hdl.handle.net/1721.1/12332>
- 287 [5] K. Shimada, D. C. Gossard, Automatic triangular mesh generation of trimmed parametric surfaces for
288 finite element analysis, Computer Aided Geometric Design 15 (3) (1998) 199 – 222. doi:10.1016/S0167-8396(97)00037-X.
289 URL <http://www.sciencedirect.com/science/article/pii/S016783969700037X>
- 290 [6] L. Wu, B. Chen, G. Zhou, An improved bubble packing method for unstructured grid generation with
291 application to computational fluid dynamics, Numerical Heat Transfer, Part B: Fundamentals 58 (5)
292 (2010) 343–369. doi:10.1080/10407790.2010.511970.
293 URL <https://www.tandfonline.com/doi/abs/10.1080/10407790.2010.511970>
- 294 [7] N. Qi, Y. Nie, W. Zhang, Acceleration strategies based on an improved bubble packing method, Communications in Computational Physics 16 (1) (2014) 115–135. doi:10.4208/cicp.080213.151113a.
- 295 [8] W. Guo, Y. Nie, W. Zhang, Acceleration strategies based on bubble-type adaptive mesh refinement
296 method, Mathematics and Computers in Simulation 170 (2020) 143–163. doi:10.1016/j.matcom.
297 2019.10.014.
- 298 [9] Q. Wang, B. Gao, H. Wu, Triangular mesh generation on free-form surfaces based on bubble dynamics
299 simulation, Engineering Computations 36 (2) (2019) 646–663. doi:10.1108/ec-09-2017-0352.

- 302 [10] D. S. Lo, Finite Element Mesh Generation, CRC Press, London, 2015. doi:10.1201/b17713.
- 303 [11] S. H. Lo, A new mesh generation scheme for arbitrary planar domains, International Journal for
304 Numerical Methods in Engineering 21 (8) (1985) 1403–1426. doi:10.1002/nme.1620210805.
305 URL <https://onlinelibrary.wiley.com/doi/abs/10.1002/nme.1620210805>
- 306 [12] W. H. Frey, D. A. Field, Mesh relaxation: A new technique for improving triangulations, Interna-
307 tional Journal for Numerical Methods in Engineering 31 (6) (1991) 1121–1133. doi:10.1002/nme.
308 1620310607.
- 309 [13] S. Lo, Generation of high-quality gradation finite element mesh, Engineering Fracture Mechanics 41 (2)
310 (1992) 191 – 202. doi:10.1016/0013-7944(92)90180-M.
311 URL <http://www.sciencedirect.com/science/article/pii/001379449290180M>
- 312 [14] L. A. Freitag, C. Ollivier-Gooch, Tetrahedral mesh improvement using swapping and smoothing, In-
313 ternational Journal for Numerical Methods in Engineering 40 (21) (1997) 3979–4002. doi:10.1002/
314 (sici)1097-0207(19971115)40:21<3979::aid-nme251>3.0.co;2-9.
- 315 [15] C. Lee, R. Hobbs, Automatic adaptive finite element mesh generation over arbitrary two-dimensional
316 domain using advancing front technique, Computers & Structures 71 (1) (1999) 9–34. doi:10.1016/
317 s0045-7949(98)00215-6.
- 318 [16] R. Feuillet, A. Loseille, F. Alauzet, Optimization of p2 meshes and applications, Computer-Aided
319 Design 124 (2020) 102846. doi:10.1016/j.cad.2020.102846.
- 320 [17] Q. Luo, Discretized pressure poisson algorithm for steady incompressible flow on two-dimensional
321 triangular unstructured grids, European Journal of Mechanics - B/Fluids 80 (2020) 187–194. doi:
322 10.1016/j.euromechflu.2019.11.003.
- 323 [18] G. Subramanian, V. V. S. Raveendra, M. G. Kamath, Robust boundary triangulation and delaunay
324 triangulation of arbitrary planar domains, International Journal for Numerical Methods in Engineering
325 37 (10) (1994) 1779–1789. doi:10.1002/nme.1620371009.
326 URL <https://www.onlinelibrary.wiley.com/doi/abs/10.1002/nme.1620371009>
- 327 [19] W. Anderson, A grid generation and flow solution method for the euler equations on unstructured
328 grids, Journal of Computational Physics 110 (1) (1994) 23 – 38. doi:10.1006/jcph.1994.1003.
329 URL <http://www.sciencedirect.com/science/article/pii/S0021999184710035>
- 330 [20] K. S. V. Kumar, A. V. R. Babu, K. N. Seetharamu, T. Sundararajan, P. A. A. Narayana, A generalized
331 delaunay triangulation algorithm with adaptive grid size control, Communications in Numerical Meth-
332 ods in Engineering 13 (12) (1997) 941–948. doi:10.1002/(sici)1099-0887(199712)13:12<941::
333 aid-cnm121>3.0.co;2-m.
- 334 [21] S.-W. Cheng, T. K. Dey, J. Shewchuk, Delaunay Mesh Generation, Chapman and Hall/CRC, New
335 York, 2013. doi:10.1201/b12987.

- 336 [22] H. Borouchaki, S. Lo, Fast delaunay triangulation in three dimensions, Computer Methods in Applied
337 Mechanics and Engineering 128 (1-2) (1995) 153–167. doi:10.1016/0045-7825(95)00854-1.
- 338 [23] P.-O. Persson, G. Strang, A simple mesh generator in MATLAB, SIAM Review 46 (2) (2004) 329–345.
339 doi:10.1137/s0036144503429121.
- 340 [24] P.-O. Persson, Mesh generation for implicit geometries, Ph.D. thesis, Massachusetts Institute of Tech-
341 nology (2004).
- 342 URL <http://hdl.handle.net/1721.1/27866>
- 343 [25] R. C. Hibbeler, Engineering Mechanics: Dynamics, 14th Edition, Pearson, 2015.
- 344 [26] P.-O. Persson, Mesh size functions for implicit geometries and PDE-based gradient limiting, Engineering
345 with Computers 22 (2) (2006) 95–109. doi:10.1007/s00366-006-0014-1.
- 346 [27] A. Liu, B. Joe, Relationship between tetrahedron shape measures, BIT 34 (2) (1994) 268–287. doi:
347 10.1007/bf01955874.
- 348 [28] S. Lo, Optimization of tetrahedral meshes based on element shape measures, Computers & Structures
349 63 (5) (1997) 951–961. doi:10.1016/s0045-7949(96)00399-9.