Automatic unstructured Cartesian mesh generation method and its application for diffusion problems with finite volume method

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

For the application of finite volume method for complex geometries, considering the effects of mesh orthogonality on the discretization of diffusion terms, unstructured Cartesian mesh might be more appropriate if accurate numerical solutions were needed. For the generation of unstructured Cartesian mesh, tree data structures were widely used for obtaining neighbourhood relationship of elements during the mesh generation processes. This paper provided an automatic unstructured Cartesian mesh generation method and also introduced the application of unstructured Cartesian mesh for diffusion problems with finite volume method. (1) For the automatic unstructured Cartesian mesh generation method, list data structure (rather than the more complicated tree data structures) is used as obtaining neighbourhood relationship of elements are not needed during the whole processes of unstructured Cartesian mesh generation. Meanwhile, with the using of background grid, neighbourhood relationship could be easily obtained after the unstructured Cartesian mesh generation. (2) For the application unstructured Cartesian mesh for diffusion problems, the advantages of unstructured Cartesian mesh are that complex geometries can be well approximated and the finite volume method on the mesh is consistent. The disadvantage is that the unstructured Cartesian mesh needs far more elements for region boundary approximation (compared with unstructured triangular mesh and structured Cartesian mesh).

Keywords: unstructured Cartesian mesh; mesh generation; finite volume method; diffusion problems.

1. Introduction

In my previous researches, we [1] provided a discretized pressure Poisson algorithm for the steady incompressible flow on structured mesh. In order to make this method suitable for complex geometries, we [2] also developed a method to make the discretized pressure Poisson algorithm to be applied on unstructured (triangular) mesh. However, there was a problem when finite volume method was applied to unstructured mesh: the discretization of diffusion term would bring errors at nonorthogonal triangular mesh which are not existed at orthogonal rectangular mesh [3] (pp. 316-320). Some discretization methods of diffusion term [4–6] have been provided to reduce the errors caused by nonorthogonal unstructured mesh. However, these methods are too complicated and the errors caused by nonorthogonal grids are actually not eliminated.

As rectangular meshes are orthogonal, we think that Cartesian mesh [3] (pp. 304-305) might be more appropriate for diffusion problems if accurate numerical solutions were needed. Cartesian mesh can be structured or unstructured. The advantage of unstructured Cartesian grids [7] (pp. 275-277) is that complex geometries can be well approximated by refined elements around geometry boundaries. In fact, unstructured Cartesian meshes have been used in many problems of heat transfer and fluid dynamics [8–10].

For the application of unstructured Cartesian mesh, the first problem is how the unstructured Cartesian meshes were generated. For unstructured mesh (triangular elements), some books [11–13] and even more articles (such as [14, 15]) have introduced different mesh generation methods. My previous research [16] also provided an automatic Delaunay mesh generation method and physically-based mesh optimization method on two-dimensional regions. However, there are relatively fewer references about the unstructured Cartesian mesh generation methods. Previously, De Zeeuw and Powell [17] (see also in the PhD Thesis of De Zeeuw [18]) developed an adaptively refined Cartesian mesh generation method by using quadtree data structure. Similarly, Coirier and Powell [19] (see also in the PhD Thesis of Coirier [20]) used binary data structure for unstructured Cartesian mesh generation. In

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

these methods, tree data structures were used for obtaining neighbourhood relationship of elements conveniently in the later steps of mesh refinement. In fact, quadtree-octree based methods have been also used in mesh generation. The books of Lo [11] (pp. 86-91) and Frey [13] (pp. 163-199) provided detailed introduction about these methods.

The author think that obtaining neighbourhood relationship of elements are not essential during the processes of unstructured Cartesian mesh generation. Thus, simpler data structure (such as list) can be used to replace the more complicated tree data structures. This paper would provide such a method to generate the unstructured Cartesian mesh without using tree data structures. Meanwhile, with the using of background grid, neighbourhood relationship could be easily obtained after the unstructured Cartesian mesh generation.

Then, this paper would introduce the application of unstructured Cartesian mesh for diffusion problems with finite volume method. The errors of finite volume method for diffusion problems under unstructured Cartesian mesh would be analyzed in details. The errors under unstructured Cartesian mesh would be further compared with the errors under unstructured triangular mesh and structured Cartesian mesh, which will be meaningful for choosing appropriate mesh for diffusion problems under complex geometries.

2. Automatic unstructured Cartesian mesh generation method

2.1. Geometry

Providing the geometry (or region) is the first step of mesh generation. In the Green's Theorem [21] (pp. 1154-1161), the two-dimensional region is bounded by a counterclockwise outer curve and several clockwise inner curves. In this paper, similar to that method, the inner and outer curves are discretized to many directional line segments. As shown in the Figure 1, the region between edge 1 and edge 2 is the region defined in the example.

In order to control the local mesh coarseness and fineness, this paper should identify the "fine edges", then the mesh elements around these edges would be refined at later steps. As shown in the Figure 1, the edges with number "2" are the refined edges defined in the example. Figure 2 presented the detailed steps during unstructured Cartesian mesh

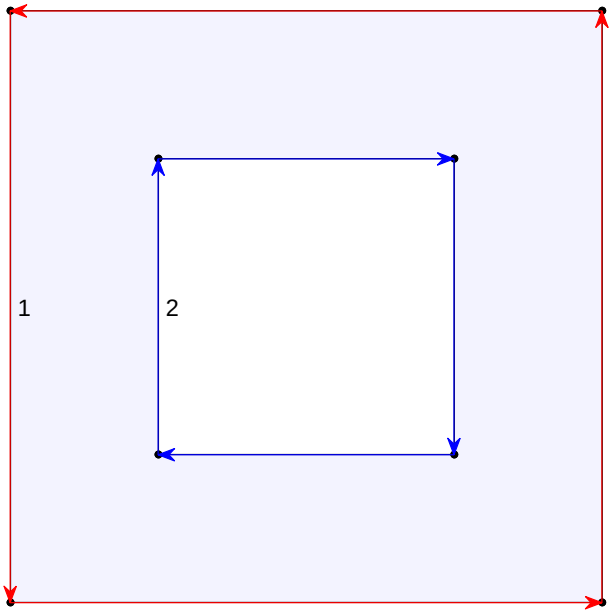


Figure 1: A square region with a square hole.

generation for the square region with a square hole. These steps would be introduced in details at the following subsections.

2.2. Background grid

As introduced in the book of Lo [11] (pp. 53-75), background grid is widely used in mesh generation. Background grid is a partition of space to facilitate searching or to establish a neighbourhood relationship. In this paper, the regular grid provided the initial mesh. Figure 2(a) presented the regular background grid of the square region with a square hole. As background grid can be used to facilitate searching or to establish a neighbourhood relationship, the background grid will be used in the later to obtain the neighbourhood relationship of the unstructured mesh elements before be used in the finite volume method.

The data structure of the unstructured Cartesian mesh should be considered before the usage of background grid. Previously, the quadtree data structure are widely used for unstructured Cartesian mesh [17, 18]. However, in this paper, list data structure is used as it is simpler than quadtree data structure and it is also very convenient to delete

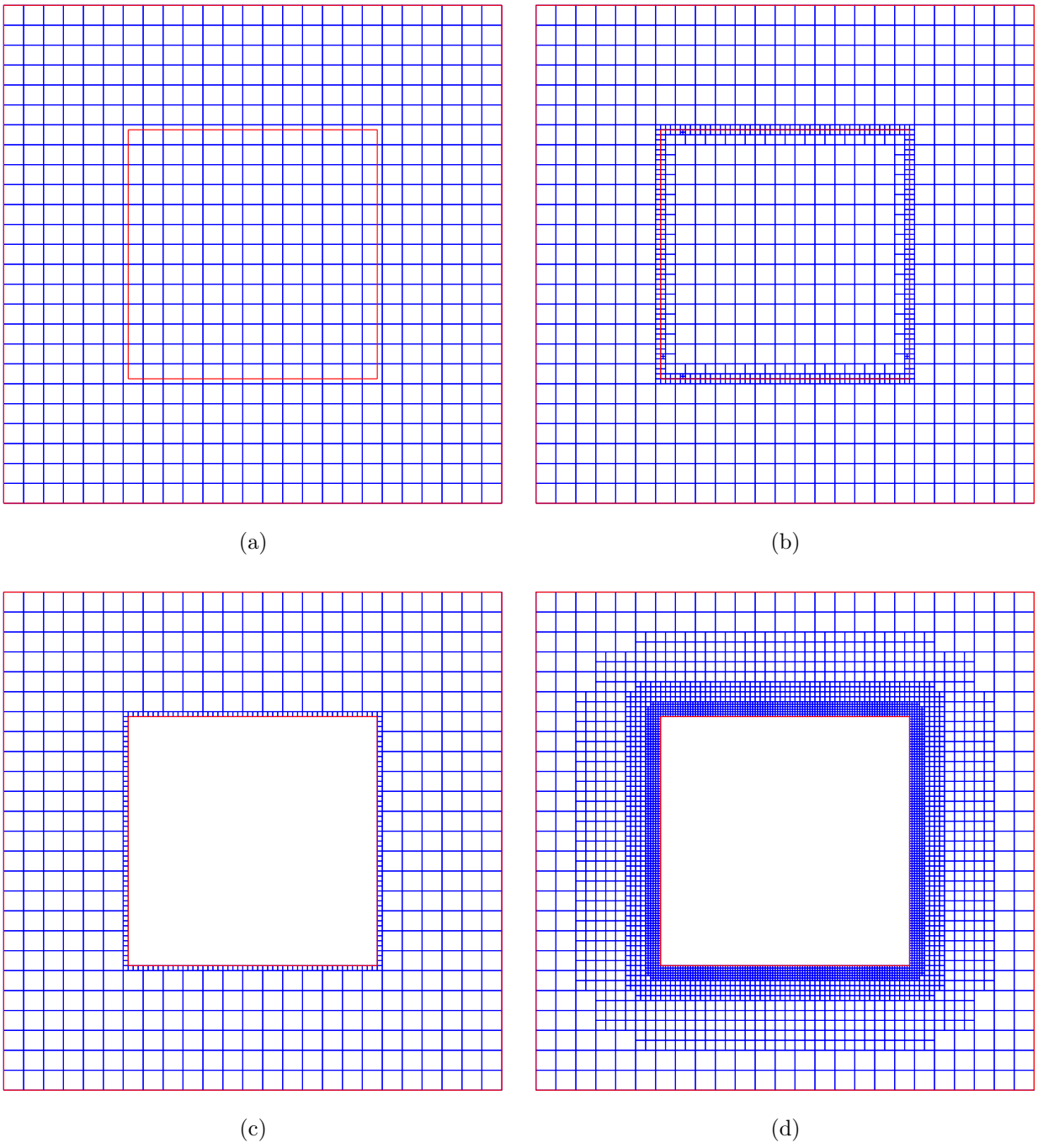


Figure 2: Unstructured Cartesian mesh generation for a square region with a square hole.

unwanted mesh elements and insert new mesh elements in the list data structure. Another important reason for using list data structure (rather than quadtree data structure) is that

obtaining neighbourhood relationship of elements are not needed during the whole processes of unstructured Cartesian mesh generation.

2.3. Initial mesh refinement

In this step, the mesh elements who are crossed with the fine edges are needed to be refined. The level of refinement should be provided in advance. For example, the refinement level can be three or four, then the side length of the finest mesh will be 1/8 or 1/16 of the normal mesh. Figure 2(b) presented the initial refined mesh for the square region with a square hole. In this figure, the level of refinement is three.

In this step, how to identify whether a square mesh element is crossed with the fine edges or not is a problem. The method used in this paper is shown in the Figure 3. The position of nearest point (in the fine edges) between the center of the square mesh element and fine edges should be calculated.

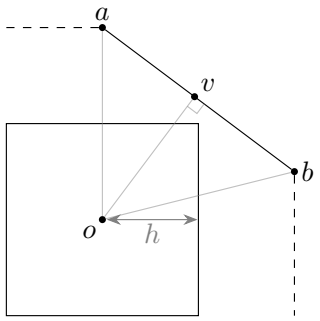


Figure 3: The diagram to identify whether an element is crossed with edges.

From Figure 3, for the center point o and edge ab, the nearest point n is the perpendicular foot v rather than a or b. Then, the square mesh element is crossed with the fine edges certainly, if the nearest point n is located in the region of the square mesh element, which means the following equations should be satisfied simultaneously.

$$\begin{cases} x_o - h \leqslant x_n \leqslant x_o + h \\ y_o - h \leqslant y_n \leqslant y_o + h \end{cases} \tag{1}$$

2.4. Deletion of out-region elements

In this step, the out-region elements would be deleted. The problem is whether a square element is out of the region or not. This can be realized by calculate the signed distance from the square element center to region edges. In this paper, as the edges are oriented line segments, the point which is located at the right side of the nearest edge should have "negative" distance. The distance from the square element to the region edges can be calculated by calculate the minimum distance from a point to the line segments of edges. The method to calculate the distance from a point to a line segment is introduced in the book of Lo [11] (pp. 19-20). If the distance from the center of the square element to the region edges is negative, the element would be certainly out of the region. Then, the element should be deleted from the list data structure of mesh elements. Figure 2(c) presented the mesh after delete out-region elements for a square region with a square hole.

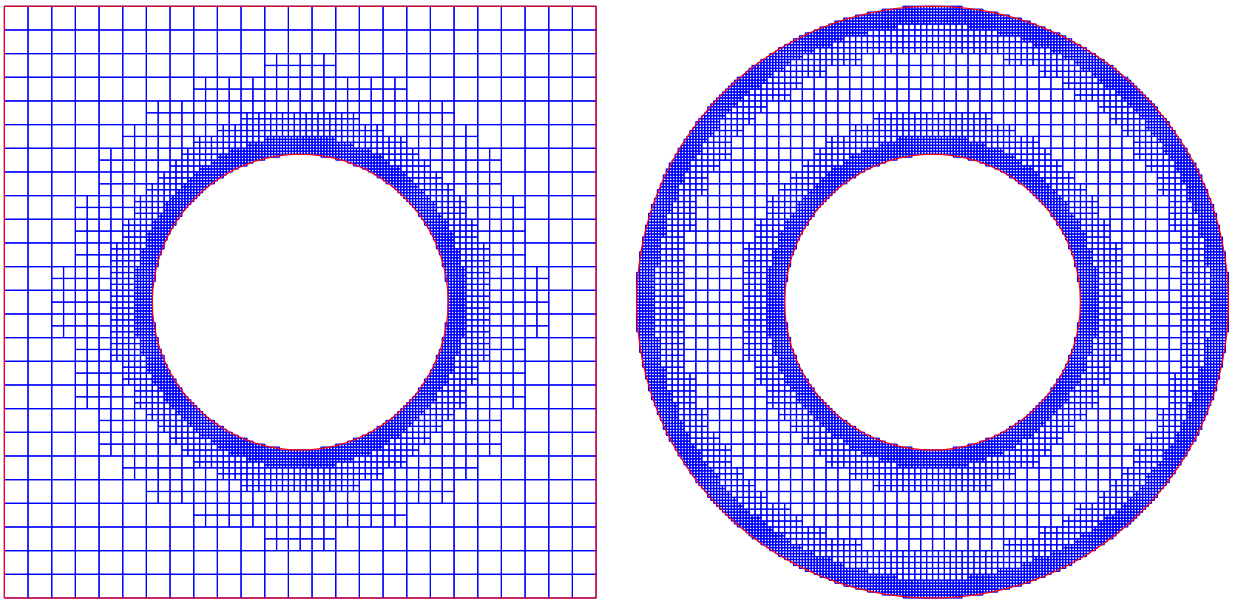
2.5. Further mesh refinement

In this step, the mesh elements around the fine edges would be further refined. The problem is which square elements should be refined. This paper provided a criterion: the element should be refined if the following equation is satisfied.

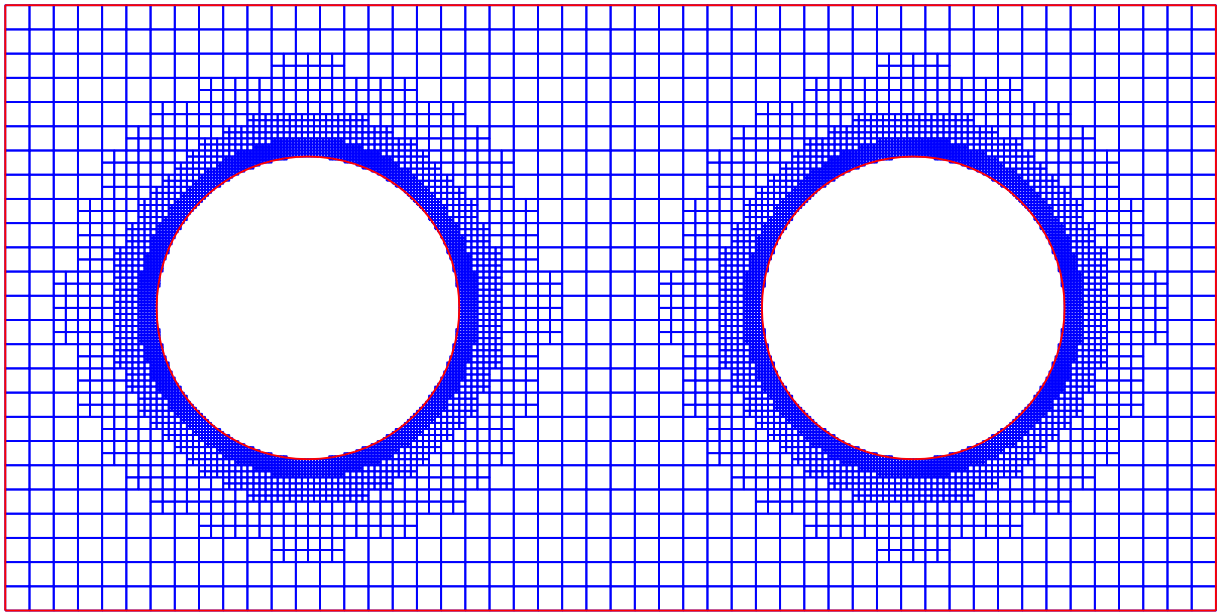
$$d < \alpha \cdot h_{\min} \cdot (2^{L-l+1} - 1) \tag{2}$$

In the above equation, d is the distance from the center of square element to fine edges; $h_{\min}$ is the half side length of the minimum square element; L is the predefined level of refinement; l is the current level of refinement ($l \leqslant L$), α is a parameter to control the width of each refinement level. Figure 2(d) presented the mesh after further mesh refinement for a square region with a square hole. In the figure, α is given as 4.2.

Figure 4 presented some examples of unstructured Cartesian meshes generated by the method provided by this paper. These unstructured Cartesian meshes could be generated automatically by provided their geometries and several parameters.



(a) (b)



(c)

Figure 4: Some examples of unstructured Cartesian meshes generated by the method provided by this paper.

112 *2.6. Neighbourhood relationship*

113 For the convenient usage of the unstructured Cartesian mesh, the method to calculate
114 neighbourhood relationship of unstructured Cartesian mesh are suggested to be provided
115 after the mesh have been generated. In the book of Lo [11] (pp. 33-34), it introduced a
116 method to find the neighbouring elements of a triangular mesh. But, as the data structure
117 of the unstructured Cartesian mesh is different from the traditional triangular mesh, this
118 subsection would use background grid method to calculate neighbourhood relationship of
119 the elements of unstructured Cartesian mesh. This background grid method is widely used
120 in Smoothed Particle Hydrodynamics (SPH) method for neighbouring particles search [22]
121 (pp. 28-29). Figure 5 presented the diagram to obtain the neighbouring elements of the a
square element by using background grid.

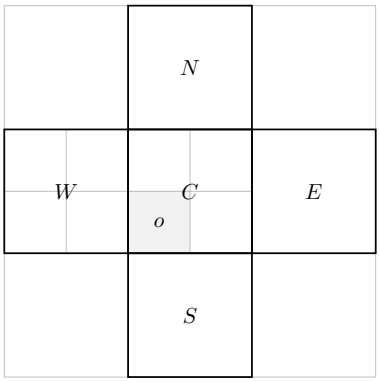


Figure 5: The diagram to obtain the neighbouring elements of the a square element by using background grid.

122

123 In the Figure 5, if the center of a square element o is located in the region of a background
124 grid element C, the neighbouring elements of the square element o must be some square
125 elements contained in the region of background grid elements C, N, S, W and E. Then,
126 the square element i is the neighbouring element of square element o if the either of the
127 following two equations is satisfied.

$$\begin{cases} |(h_o + h_i) - |x_o - x_i|| < r \\ (h_o + h_i) - |y_o - y_i| > r \end{cases} \text{or} \begin{cases} (h_o + h_i) - |x_o - x_i| > r \\ |(h_o + h_i) - |y_o - y_i|| < r \end{cases} \tag{3}$$

In the above equation, (x_o, y_o) and (x_i, y_i) are the coordinates of the center of square elements o and i respectively. h_o and h_i are respectively the half side length of square elements o and i. As errors were introduced when the floating-point arithmetics are implemented, the error r is introduced in the equation to offset the effects of round-off errors. In the programs, the value of r is given as $(h_o + h_i)/10^4$.

3. Application of unstructured Cartesian mesh for diffusion problems with finite volume method

3.1. Diffusion problem and finite volume method

Poisson's equation on unit disk, which is a typical diffusion problem, is used to evaluate finite volume method for diffusion problems under different meshes. The governing equation is

$$\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} + 1 = 0. \tag{4}$$

For the Poisson equation on a unit disk with zero Dirichlet boundary $T = 0$, the exact solution is $T = (1 - x^2 - y^2)/4$. As there is a exact solution for the shape of unit disk (a typical complex geometry), this example is a very suitable example to compare the errors of finite volume method for diffusion problems under different meshes.

Finite volume method is well introduced in the book of Versteeg and Malalasekera [3]. This subsection will briefly introduce the discretization and interpolation methods of finite volume method according to my previous research [2].

Discretization of diffusion term:

$$\int_{\Omega_P} \mathrm{div}(\mathbf{grad}\phi) d\Omega = \oint_{\Gamma_P} \mathbf{n} \cdot \mathbf{grad}\phi d\Gamma \approx \sum_{nb} \frac{\phi_{nb} - \phi_P}{l_{Pnb}} \Gamma_{nb} \tag{5}$$

In the above equation, l_{Pnb} is the distance from element edge to element center; Γ_{nb} is the length of element edge. ϕ_{nb} is the quantity value at element edge, its value can be calculated with linear interpolation [2]:

$$\phi_{nb} = \frac{l_{Anb}}{l_{Pnb} + l_{Anb}} \phi_P + \frac{l_{Pnb}}{l_{Pnb} + l_{Anb}} \phi_A \tag{6}$$

Discretization of source term:

$$\int_{\Omega_P} S d\Omega \approx S_P \Omega_P \tag{7}$$

In the above equation, Ω_P is the area of element.

According to finite volume method, the results of discretization for Poisson's equation on unit disk are shown as follows.

$$T_P = \hat{T}_P + \Delta T_P \tag{8}$$

In the above equation,

$$\hat{T}_P = \frac{1}{a_P} \sum_{nb} a_{nb} T_{nb}, \Delta T_P = \frac{A_P}{a_P} \tag{9}$$

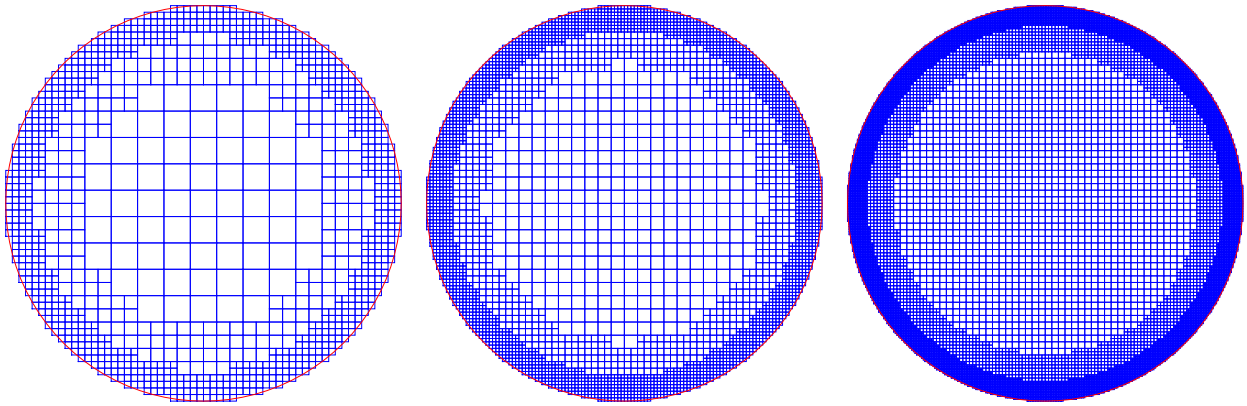
$$a_{nb} = \frac{l_{nb}}{r_P}, a_P = \sum_{nb} a_{nb} \tag{10}$$

r_P is the radius of incircle (triangle or square); A_P is the area of element. Boundary conditions are very simple: $T_{nb} = 0$. Whether the iteration is convergent or not can be judged from whether $\max\left(\left|T_P^n - T_P^{n-1}\right|\right) < \varepsilon$. In this paper, ε is given as 10^{-10}.

3.2. Unstructured Cartesian mesh and numerical errors

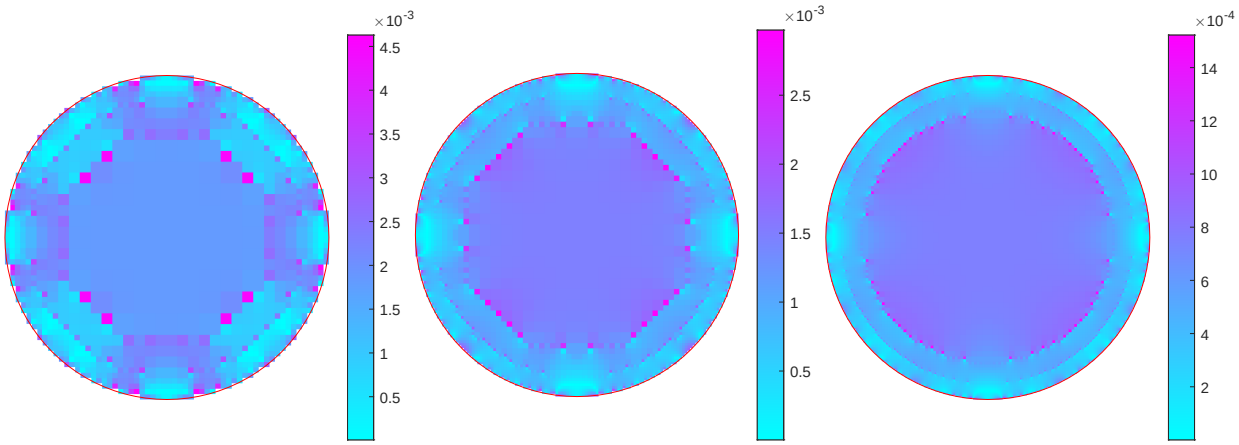
Figure 6 shows the unstructured Cartesian mesh on unit disk. There are 1121, 3816 and 12836 square elements respectively. Figure 7 shows the absolute errors between the exact solutions and numerical results for Poisson's equation on unit disk under unstructured Cartesian mesh.

From Figure 7, it can be known that, with the increasing of element numbers, the numerical errors decreased simultaneously. Meanwhile, there are relatively larger errors around those elements with one or more larger or smaller neighbouring elements. Obviously, these relatively larger errors are caused by the nonorthogonality of local elements.



(a) 1121 elements (b) 3816 elements (c) 12836 elements

Figure 6: The unstructured Cartesian mesh on unit disk.



(a) 1121 elements (b) 3816 elements (c) 12836 elements

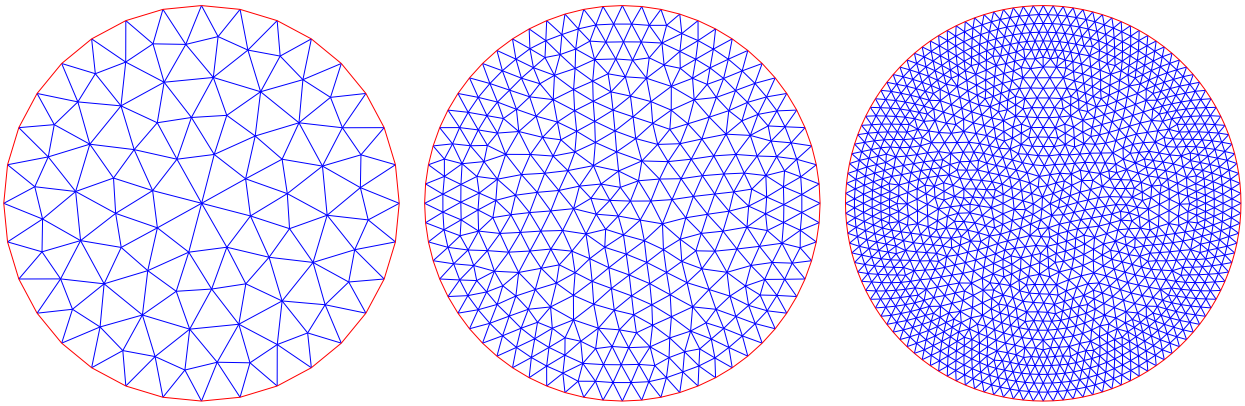
Figure 7: The absolute errors between exact solutions and numerical results for Poisson's equation on unit disk under unstructured Cartesian mesh.

3.3. Comparison of numerical errors with other meshes

3.3.1. Comparison of numerical errors with unstructured triangular mesh

Figure 8 shows the unstructured triangular mesh on unit disk. The unstructured triangular grids were generated with MATLAB Partial Differential Equation Toolbox. There are 176, 694 and 2862 triangular elements (or called as cells) respectively. Figure 9 shows the absolute errors between the exact solutions and numerical results for Poisson's equation on

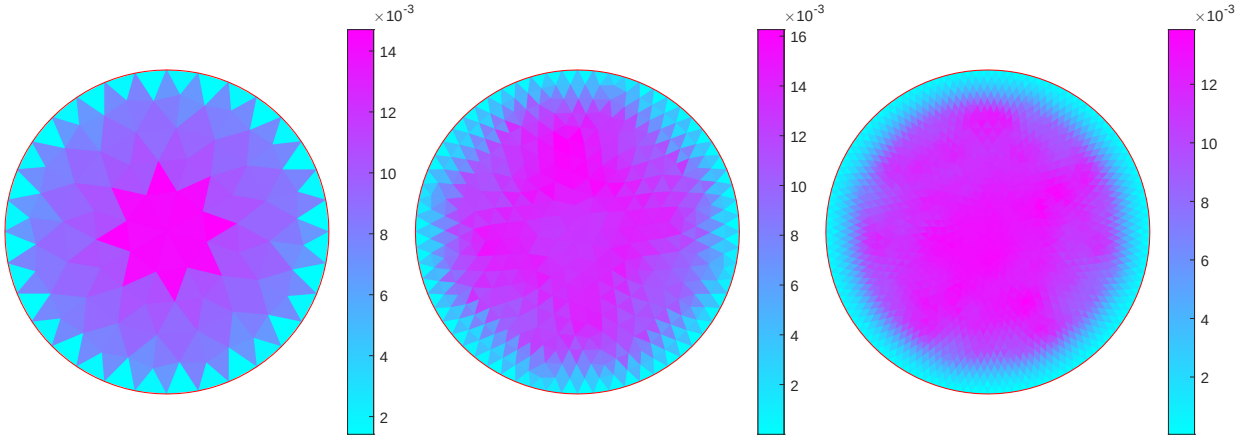
unit disk under unstructured triangular mesh.



(a) 176 elements (b) 694 elements (c) 2862 elements

Figure 8: The unstructured triangular mesh on unit disk.

174



(a) 176 elements (b) 694 elements (c) 2862 elements

Figure 9: The absolute errors between exact solutions and numerical results for Poisson's equation on unit disk under unstructured triangular mesh.

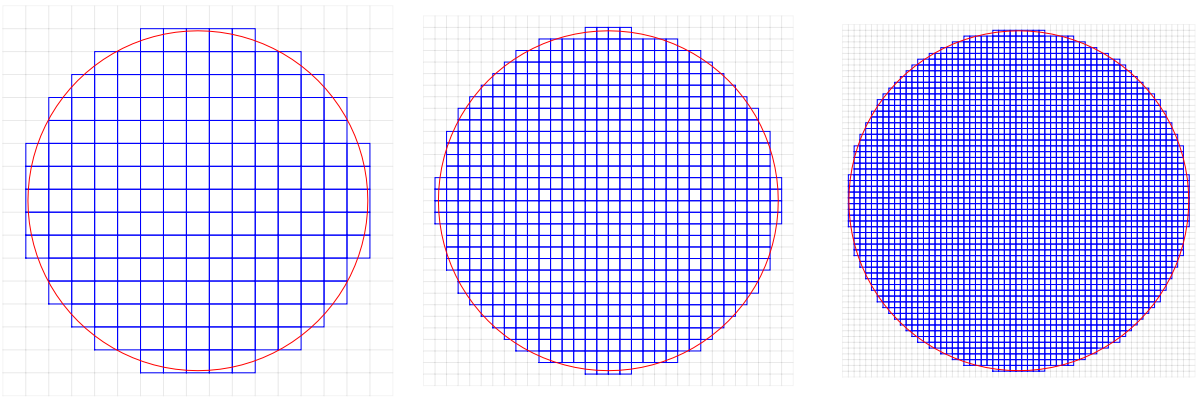
175 From Figure 8 and Figure 9, for diffusion problems, mesh skewness or non-orthogonality
176 of unstructured triangular mesh resulted in that there were larger errors at the interior
177 regions of the unit disk than at the boundary regions. More importantly, with the increasing
178 of element numbers, the errors did not decrease simultaneously. According to the definition
179 of consistency [7] (pp. 34-35), the discretization method under triangular unstructured mesh

is inconsistent. Previously, some papers [23–25] have pointed that traditional finite volume discretization method of diffusion term under triangular unstructured mesh would result in inconsistent problem. Thus, for diffusion problems which need accurate numerical solutions under complex geometries, it cannot be realized by refinement of unstructured triangular mesh.

In summary, the finite volume method for diffusion problems under unstructured Cartesian mesh is consistent while the method under unstructured triangular mesh is not consistent. However, unstructured Cartesian mesh needs far more elements for region boundary approximation than unstructured triangular mesh.

3.3.2. Comparison of numerical errors with structured Cartesian mesh

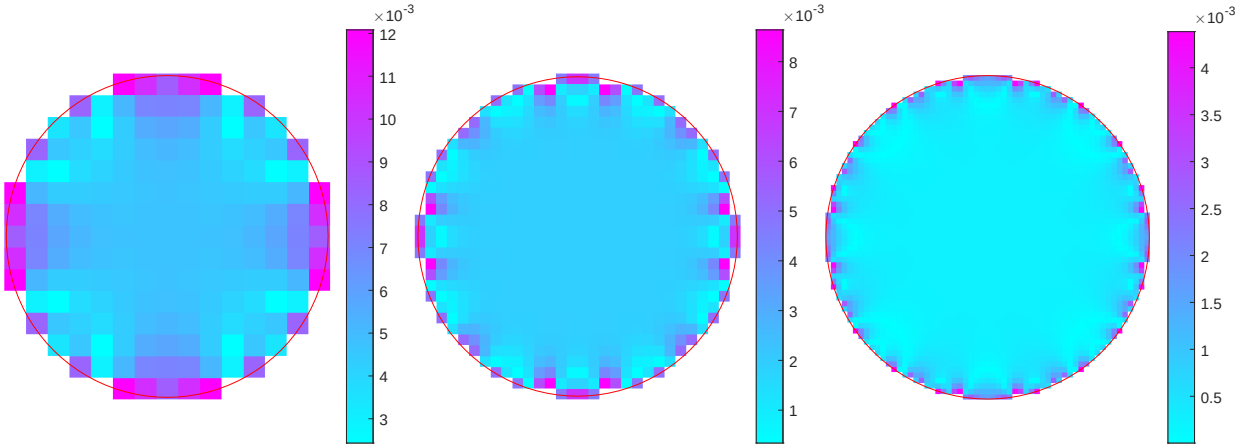
Figure 10 shows the structured Cartesian mesh on unit disk. For the structured Cartesian mesh, the blue elements are the elements which should be calculated and the gray elements should be ignored during calculation. There are 177, 680 and 2725 square elements respectively. Figure 11 shows the absolute errors between the exact solutions and numerical results for Poisson's equation on unit disk under structured Cartesian mesh.



(a) 177 elements (b) 680 elements (c) 2725 elements

Figure 10: The structured Cartesian mesh on unit disk.

From Figure 10 and Figure 11, with the increasing of element numbers, the numerical errors decreased simultaneously. Meanwhile, as there is no the problem of non-orthogonality



(a) 177 elements (b) 680 elements (c) 2725 elements

Figure 11: The absolute errors between exact solutions and numerical results for Poisson's equation on unit disk under structured Cartesian mesh.

under structured Cartesian mesh, the errors of the problem mainly come from boundaries of the mesh which are unconformable with the geometry boundaries

In summary, as the elements around boundaries are refined at unstructured Cartesian mesh, there are smaller errors at boundary regions for the finite volume method for diffusion problems under unstructured Cartesian mesh than under structured Cartesian mesh. However, for unstructured Cartesian mesh, with the refinement of boundary elements, the number of elements increased tremendously.

4. Conclusions

This paper provided an automatic unstructured Cartesian mesh generation method and also introduced the application of unstructured Cartesian mesh for diffusion problems with finite volume method. From this paper, it can be known that:

(1) Unstructured Cartesian mesh can be generated automatically with the method provided by this paper, and the mesh elements around the refine edges we defined can be automatically refined. In the method, list data structure (rather than the more complicated tree data structures) is used for as obtaining neighbourhood relationship of elements are

not needed during the whole process of unstructured Cartesian mesh generation. Meanwhile, with the using of background grid, neighbourhood relationship could be easily obtained after the unstructured Cartesian mesh generation.

(2) For diffusion problems, compared with unstructured triangular mesh and structured Cartesian mesh, the advantages of unstructured Cartesian mesh are that complex geometries can be well approximated and the finite volume method on the mesh is consistent. The disadvantage is that the unstructured Cartesian mesh needs far more elements for region boundary approximation.

Acknowledgments

This work was supported by Guangdong University of Technology. The source codes of this paper can be download from GitHub[1].

References

[1] Q. Luo, T. Ren, D. Liang, Discretized pressure poisson algorithm for the steady incompressible flow on a nonstaggered grid, Numerical Heat Transfer, Part B: Fundamentals 71 (6) (2017) 549–559. arXiv:http://dx.doi.org/10.1080/10407790.2017.1326773, doi:10.1080/10407790.2017.1326773.
URL http://dx.doi.org/10.1080/10407790.2017.1326773

[2] Q. Luo, Discretized pressure poisson algorithm for steady incompressible flow on two-dimensional triangular unstructured grids, European Journal of Mechanics - B/Fluids 80 (2020) 187–194. doi:10.1016/j.euromechflu.2019.11.003.

[3] H. K. Versteeg, W. Malalasekera, An introduction to computational fluid dynamics: the finite volume method, 2nd Edition, Pearson, Harlow, 2007.

[4] S. R. Mathur, J. Y. Murthy, A pressure-based method for unstructured meshes, Numerical Heat Transfer, Part B: Fundamentals 31 (2) (1997) 195–215. arXiv:http://dx.doi.org/10.1080/10407799708915105, doi:10.1080/10407799708915105.
URL http://dx.doi.org/10.1080/10407799708915105

[5] D. Kim, H. Choi, A second-order time-accurate finite volume method for unsteady incompressible flow on hybrid unstructured grids, Journal of Computational Physics 162 (2) (2000) 411 – 428. doi:http:

[1] https://github.com/Quanbing-Luo/UnstructuredCartesianMesh-2D

//dx.doi.org/10.1006/jcph.2000.6546.
URL http://www.sciencedirect.com/science/article/pii/S002199910096546X

[6] Y.-Y. Tsui, Y.-F. Pan, A pressure-correction method for incompressible flows using unstructured meshes, Numerical Heat Transfer, Part B: Fundamentals 49 (1) (2006) 43–65. arXiv:http://www.tandfonline.com/doi/pdf/10.1080/10407790500344084, doi:10.1080/10407790500344084.
URL http://www.tandfonline.com/doi/abs/10.1080/10407790500344084

[7] J. H. Ferziger, M. Perić, R. L. Street, Computational Methods for Fluid Dynamics, 4th Edition, Springer, Cham, 2020. doi:10.1007/978-3-319-99693-6.

[8] D. Hartmann, M. Meinke, W. Schröder, An adaptive multilevel multigrid formulation for cartesian hierarchical grid methods, Computers & Fluids 37 (9) (2008) 1103–1125. doi:10.1016/j.compfluid.2007.06.007.

[9] Y. Kidron, Y. Mor-Yossef, Y. Levy, Robust cartesian grid flow solver for high-reynolds-number turbulent flow simulations, AIAA Journal 48 (6) (2010) 1130–1140. doi:10.2514/1.45817.

[10] K. Nakahashi, Aeronautical CFD in the age of petaflops-scale computing: From unstructured to cartesian meshes, European Journal of Mechanics - B/Fluids 40 (2013) 75–86. doi:10.1016/j.euromechflu.2013.02.005.

[11] D. S. Lo, Finite Element Mesh Generation, CRC Press, London, 2015. doi:10.1201/b17713.

[12] S.-W. Cheng, T. K. Dey, J. Shewchuk, Delaunay Mesh Generation, Chapman and Hall/CRC, New York, 2013. doi:10.1201/b12987.

[13] P. L. G. Pascal Frey, Mesh Generation: Application to Finite Elements, ISTE LTD, 2008.

[14] B. Yu, M. J. Lin, W. Q. Tao, Automatic generation of unstructured grids with delaunay triangulation and its application, Heat and Mass Transfer 35 (5) (1999) 361–370. doi:10.1007/s002310050337.
URL https://doi.org/10.1007/s002310050337

[15] P.-O. Persson, G. Strang, A simple mesh generator in MATLAB, SIAM Review 46 (2) (2004) 329–345. doi:10.1137/s0036144503429121.

[16] Q. Luo, Automatic delaunay mesh generation method and physically-based mesh optimization method on two-dimensional regions, Engineering with Computersdoi:10.1007/s00366-020-01262-x.

[17] D. DeZeeuw, K. G. Powell, An adaptively refined cartesian mesh solver for the euler equations, Journal of Computational Physics 104 (1) (1993) 56–68. doi:10.1006/jcph.1993.1007.

[18] D. DeZeeuw, A quadtree-based adaptively-refined cartesian-grid algorithm for solution of the euler equations, Ph.D. thesis, The University of Michigan (1993).
URL http://hdl.handle.net/2027.42/103745

[19] W. J. Coirier, K. G. Powell, An accuracy assessment of cartesian-mesh approaches for the euler equations, Journal of Computational Physics 117 (1) (1995) 121–131. doi:10.1006/jcph.1995.1050.

[20] W. J. Coirier, An adaptively-refined, cartesian, cell-based scheme for the euler and navier-stokes equations, Ph.D. thesis, The University of Michigan (1994).
URL http://hdl.handle.net/2027.42/129435

[21] J. Stewart, D. K. Clegg, S. Watson, Calculus: Early Transcendentals, Cengage Learning, Boston, 2020.

[22] C. A. D. F. Filho, Smoothed Particle Hydrodynamics: Fundamentals and Basic Applications in Continuum Mechanics, Springer-Verlag GmbH, 2019. doi:10.1007/978-3-030-00773-7.

[23] M. Svärd, J. Gong, J. Nordström, An accuracy evaluation of unstructured node-centred finite volume methods, Applied Numerical Mathematics 58 (8) (2008) 1142 – 1158. doi:https://doi.org/10.1016/j.apnum.2007.05.002.
URL http://www.sciencedirect.com/science/article/pii/S0168927407000992

[24] A. Jalali, M. Sharbatdar, C. Ollivier-Gooch, Accuracy analysis of unstructured finite volume discretization schemes for diffusive fluxes, Computers & Fluids 101 (2014) 220 – 232. doi:https://doi.org/10.1016/j.compfluid.2014.06.008.
URL http://www.sciencedirect.com/science/article/pii/S0045793014002473

[25] C. B. Sejekan, C. F. Ollivier-Gooch, Improving finite-volume diffusive fluxes through better reconstruction, Computers & Fluids 139 (2016) 216 – 232. doi:https://doi.org/10.1016/j.compfluid.2016.08.002.
URL http://www.sciencedirect.com/science/article/pii/S0045793016302389