AUTOMATIC BLOCKING SCHEME FOR STRUCTURED MESHING IN 2D MULTIPHASE FLOW SIMULATION

Damrong Guoy[1] Jeff Erickson[2]

[1] *Center for Simulation of Advanced Rockets, Computational Science and Engineering Program, University of Illinois at Urbana-Champaign, guoy@uiuc.edu*
[2] *Department of Computer Science, University of Illinois at Urbana-Champaign, jeffe@uiuc.edu*

ABSTRACT

We present an automatic algorithm to construct blocking scheme for multiblock structured meshes in 2D multiphase flow problems. Our solution is based on the concepts of medial axis and Delaunay triangulation. We show that the quality of the blocking scheme strongly depends on the quality of Delaunay triangulation. Therefore, well-known techniques and issues like Delaunay refinement and geometric degeneracy resurge again in multiblock structured meshes.

Keywords: multiblock structured mesh, medial axis, Delaunay triangulation, Voronoi diagram, Delaunay refinement, multiphase flow simulation.

1. INTRODUCTION

We study a fully automatic algorithm to construct multiblock structured meshes in two dimensions for a certain class of multiphase flow with moving particles. Structured meshes have been widely used in computational fluid dynamics (CFD). For a certain class of physical domains, there is an extensive list of literature that generates a structured mesh by mapping the domain to a square in a parametric domain. For a domain with complex shape, however, manual editing is required to subdivide the domain into simple blocks. This manual step can be tedious and time consuming.

Target applications of our meshing scheme are direct numerical simulations (DNS) of turbulence in multiphase flow with the presence of particles, bubbles, or droplets. The simulations resolve small-scale flow features around particles, turbulence in carrier-phase flow, and interaction between the two phases. Figure 1 on the left gives an example of such simulations in 2D. Recent advanced 3D simulation [1] considers one stationary particle in ambient turbulence. The grand-challenge simulation will consider a distribution of freely moving particles in the ambient flow. High fidelity of the simulation requires body-fitted grid around the particles. Motion of particles requires fully automatic meshing algorithms.

This paper focuses on meshing of the dispersed-phase flow around particles. The carrier-phase flow can be treated with a cartesian grid. The dispersed-phase flow requires a body-fitted grid for high-fidelity simulation. Its domain consists of a bounding circle with circular holes representing the particles. The time-dependent simulation determines motion of the particles. Figure 1 on the right gives an example of our meshes. The two particles are enclosed by a bounding circle. Transition from the bounding circle to the cartesian grid is performed by a standard template. Notice that the bounding circle has been transformed to an ellipse by mesh smoothing.

We can formulate our problem in the following way.

Problem 1: Structured Meshes of Circles.
Given a domain described by a bounding circle and several circular holes, generate a multiblock structured mesh automatically.

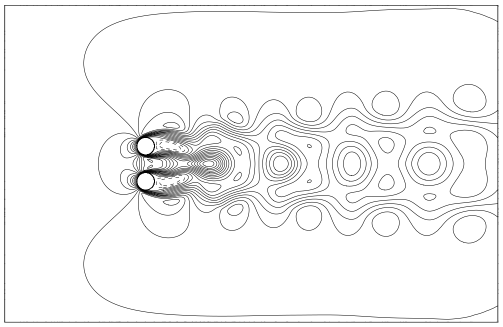
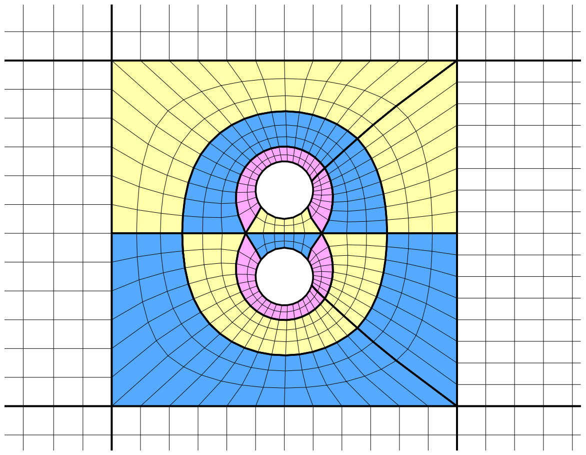



Figure 1: Flow simulation (courtesy of Lin Zhang) on the left and the mesh around two particles on the right.

Our proposed solution is based on the concepts of medial axis and Delaunay triangulation. Medial axis has been used for quadrilateral and hexahedral meshing by many researchers [2, 3, 4, 5, 6] since 1991. All the previous works focused on unstructured meshes. Our solution, on the other hand, applies medial axis to multiblock structured meshes. Another contribution of our work is the simplicity of the algorithm. By restricting ourselves to circular domains, our algorithm becomes straightforward. The algorithms in [2, 3, 4, 5, 6] are more general but require complicated case analysis.

In general, meshing algorithms based on medial axis have their own issues and limitations. We devote a significant part of our work to handle such problems. These problems are caused by bad features in medial axis like poor angles and short arcs. We show that the idea of Delaunay refinement can be helpful here.

The rest of this paper is organized into five sections. Section 2 reviews the concept of medial axis and its relation to Delaunay triangulation in two dimensions. Section 3 presents our automatic blocking scheme and explains its appealing properties. Section 4 discusses how to detect special cases and how to handle them. Section 5 presents our results to justify the approach. Section 6 concludes the paper and discusses future research.

2. MEDIAL AXIS AND DELAUNAY TRIANGULATION

We adopt the definition of medial axis from [7, 8, 9]. Given a two-dimensional domain, its medial axis is the locus of centers of maximal empty circles inside the domain. Figure 2 gives an example of medial axis. Some of the empty circles are shown as dotted circles. These empty circles stay inside the domain. Each dotted circle is maximal in the sense that it is tangent to domain boundary at two or more points. Most of the dotted circles in Figure 2 touch domain boundary in exactly two points. Their centers lie on an arc of medial axis. Figure 2 also shows one empty circle that touches domain boundary in three points. Its center lies on a vertex shared by three medial arcs.

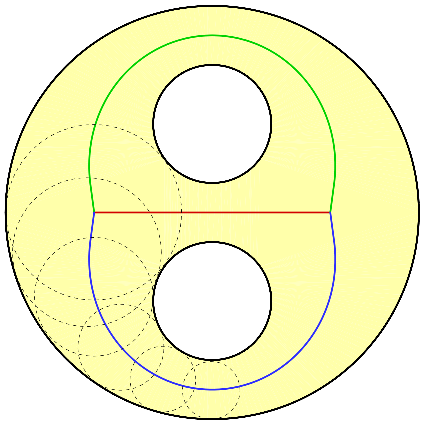


Figure 2: Medial axis is the locus of centers of maximal empty circles inside the domain.

Delaunay triangulation [10, 11] of a finite point set consists of triangles with empty circumcircles (circumscribing circles). Three points in the set form a Delaunay triangle when their circumcircle encloses no other points. We are interested in Delaunay triangulation of sample points from domain boundary, and we consider only triangles that lie inside the domain. We call the set of such triangles *Delaunay triangulation of the domain.* Figure 3 shows an example of Delaunay triangulation of sample points and Delaunay triangulation of the domain.

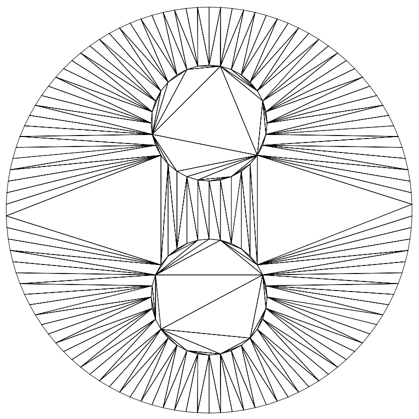
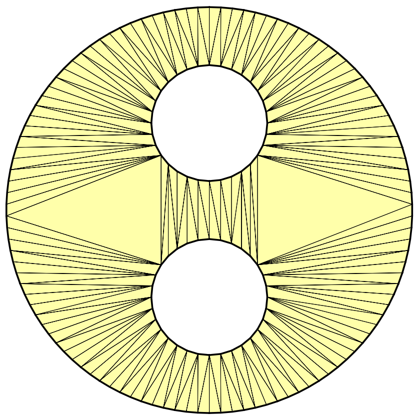



Figure 3: On the left, Delaunay triangulation of sample points. On the right, Delaunay triangulation of the domain.

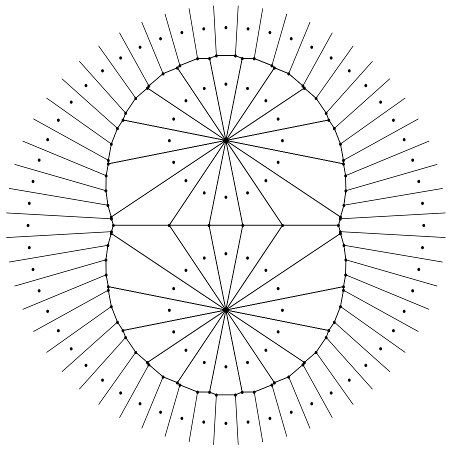
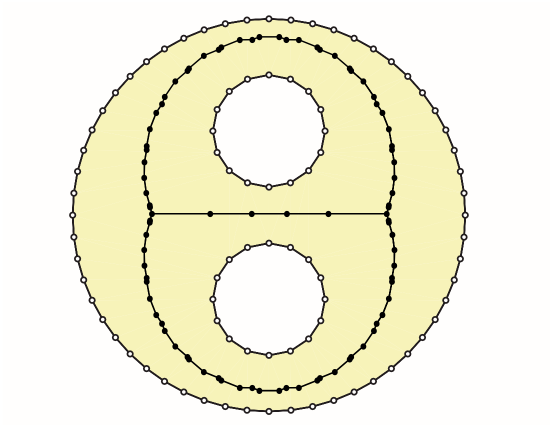



Figure 4: On the left, Voronoi diagram of sample points. On the right, Voronoi diagram of the domain.

We assume that the set of sample points is good enough that its Delaunay triangulation can represent the domain appropriately. Coming up with a good sampling strategy is not trivial for domains with all possible shapes, especially the ones with sharp angles. Many works [12, 13, 14, 15, 16] have been done to address this problem. In our case, however, our domain has smooth boundary, and the problem becomes manageable.

Delaunay triangulation of a point set has a dual complex called *Voronoi diagram* [17, 11]. Their duality is inverse in dimension. In two dimensions, a vertex in Voronoi diagram corresponds to a triangle in Delaunay triangulation. Specifically, Voronoi vertex resides at the circumcenter of Delaunay triangle. Two Voronoi vertices are connected when two Delaunay triangles share an edge. Voronoi polygons partition space by proximity to Delaunay vertices.

Given sample points from domain boundary, we can consider Delaunay triangulation of the domain as a subcomplex of Delaunay triangulation of the point set. Similarly, we can define *Voronoi diagram of the domain* as the part of Voronoi diagram that lie completely inside the domain. Figure 4 gives an example of Voronoi diagram of sample points and Voronoi diagram of the domain.

In two dimensions, dense sample points can give Voronoi diagram that approximate medial axis well. In terms of implementation, several authors [2, 18, 19, 20, 21] have proposed several algorithms similar to the following one. It approximates medial axis by traversing Delaunay triangulation.

The algorithm classifies Delaunay triangles into two classes. *Interior triangles* have no edge on domain boundary. *Boundary triangles* have exactly one edge on domain boundary. In our case, we do not have a triangle with two boundary edges. By convexity of circular holes, such a triangle cannot appear in the domain around a hole. By empty circle property, the triangle cannot appear around the bounding circle either.

Interior triangles have three neighboring triangles, and boundary triangles have two neighboring triangles. We can construct an arc of medial axis from a sequence of successive neighboring triangles. The sequence starts and ends at interior triangles with boundary triangles in between. Figure 5 gives an example of the construction.

Algorithm 1: Approximate Medial Axis.
Vertices of medial axis are approximated by circumcenters of interior triangles. An arc of medial axis is approximated by circumcenters of boundary triangles between two interior triangles.

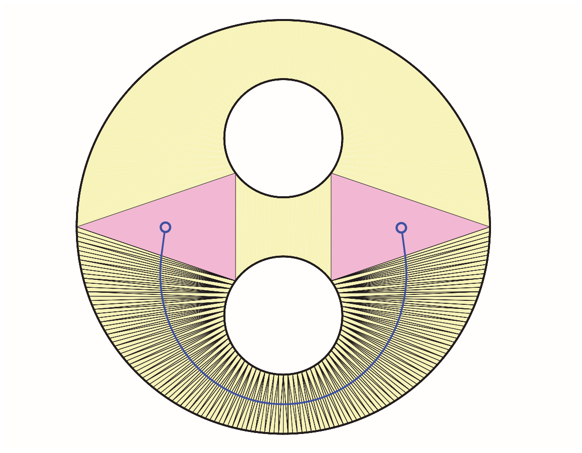


Figure 5: Vertices of medial axis correspond to interior triangles. An arc of medial axis corresponds to a sequence of triangles between two interior triangles.

3. BLOCKING SCHEME

A block in a structured mesh can be described by four bounding curves. The four curves are mapped to the four sides of a square $[0,1] \times [0,1]$ in a parametric domain (ξ, η). We define ξ_0-curve to be the bounding curve with constant $\xi = 0$. The other three curves ξ_1, η_0, η_1 are defined similarly. Our blocking scheme will describe the four bounding curves for each block.

Each arc of medial axis serves as a ξ_0-curve. We construct η_i-curves by connecting each medial vertex to vertices of the corresponding interior triangle. This implies that our η_i-curves are always straight lines. Finally, each ξ_1-curve is a part of domain boundary between end points of η_0- and η_1-curves. Figure 6 on the left shows a block created this way. Figure 6 on the right shows the whole blocking scheme. The ξ_0-curves from medial axis are drawn with thick lines. The η_i-curves are drawn with dotted lines.

In terms of implementation, we construct the blocks by traversing Delaunay triangulation from one interior triangle to another. Two blocks sharing the same medial arc are created at the same time. Each pair of blocks is described by three polygonal curves C_1, M, and C_2. The curves C_1 and C_2 are parts domain boundary. The curve M is an arc of medial axis. The following algorithm assumes that the three edges in each triangle are oriented in counterclockwise order.

Algorithm 2: Blocking2D.

Input: Delaunay triangulation of the domain.

Output: Decompose domain into pairs of blocks. Each pair is described by three polygonal curves C_1, M, and C_2.

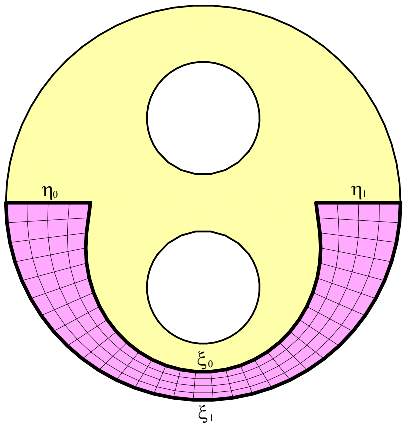
```
1.  for each interior triangle $\Delta$
2.    for each unvisited edge $\lambda$ of triangle $\Delta$
3.      $a$, $b$ = vertices of $\lambda$
4.      $C_1 = \{a\}$, $C_2 = \{b\}$, $M = \{$circumcenter$(\Delta)\}$

5.      loop forever  (invariance: $\lambda$ is an interior edge)
6.        $\Delta$ = next triangle sharing edge $\lambda$
7.        $p$ = vertex of $\Delta$ not owned by $\lambda$
8.        $\lambda_0, \lambda_1, \lambda_2$ = three edges of $\Delta$
            with $\lambda_0$ mirror of $\lambda$

9.        add circumcenter$(\Delta)$ to $M$
10.       if $\lambda_1$ is on boundary, add $p$ to $C_1$
11.       if $\lambda_2$ is on boundary, add $p$ to $C_2$

12.       if both $\lambda_1, \lambda_2$ are interior edges,
13.         exit loop
13.       if $\lambda_1$ is an interior edge,
14.         continue with $\lambda = \lambda_1$
15.       else continue with $\lambda = \lambda_2$
16.     end loop
17.   end for
18. end for
```

The states of the algorithm at line 3 and line 8 are depicted in Figure 7. We maintain a loop invariance that the edge λ is always an interior edge. The loop exits at line 13 when we reach an interior triangle, which corresponds to an endpoint of a medial arc. Notice that the curve M always receives a new point at line 9, but the curves C_1 (C_2) receives a new point at line 10 (line 11) only when we are traversing a boundary



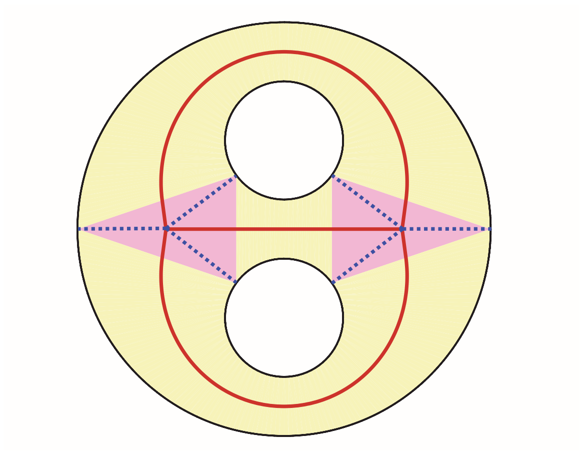


Figure 6: On the left, an example of a block created by Algorithm 2. On the right, the whole blocking scheme.

triangle. An example of a mesh generated by our algorithm is shown in Figure 8. The algorithm generates the blocking scheme automatically, and quadrilateral elements inside each block are generated using transfinite interpolation.

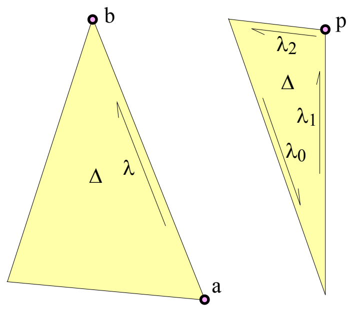


Figure 7: States of Blocking2D at line 3 on the left and at line 8 on the right.

Our blocking scheme has several appealing properties. Every block has its ξ_1-curve aligned with domain boundary, so **boundary layer** can be captured naturally. The blocks are always **orthogonal** to domain boundary because each η_i-curve is a radial ray from the center of an empty circle to a tangent point on domain boundary. The fact that η_i-curves are straight lines guarantees **applicability of transfinite interpolation**(TFI). Applying TFI to our blocks will never create self intersection or folding of grid lines.

In our blocking scheme, two adjacent blocks always share the whole ξ_i-curve or the whole η_i-curve. There is no partial overlap between any two blocks. In other words, the blocks are **conforming** and permit **regular communication pattern** in parallel simulation. In fact, each block always communicate to only three

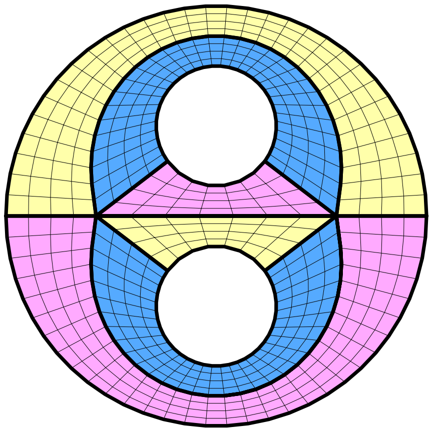


Figure 8: An example of a multiblock structured mesh from Algorithm 2.

neighboring blocks on ξ_0-, η_0-, and η_1-curves. Each corner point on domain boundary is always shared by two blocks. Each corner point in the interior, located at each medial vertex, is always shared by six blocks.

The last property is a geometric relation between the blocks and the Delaunay triangulation. The four corners of a block can be classified into two classes. *Boundary corners* are the ones located on domain boundary. *Interior corners* are the ones located inside the domain. They reside at medial vertices. As mentioned above, the angles of boundary corners are 90°. We will show that angles of interior corners equal angles of interior triangles in Delaunay triangulation.

Lemma 1: Angle Lemma. Angles of interior corners of the blocks from Algorithm 2 equal angles of interior triangles in Delaunay triangulation (Figure 9).

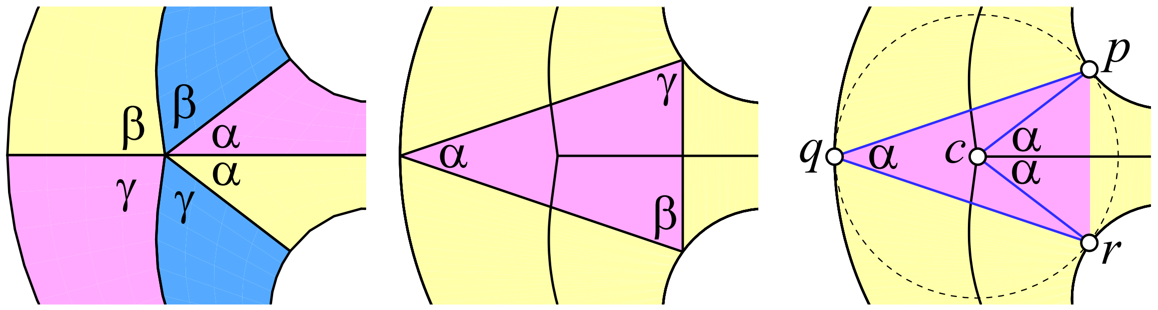


Figure 9: An illustration of Angle Lemma on the left and its proof on the right.

Proof Consider an interior triangle pqr in Delaunay triangulation. Let c be its circumcenter, which is a vertex of medial axis. The angle $p\hat{c}r$ is twice the angle $p\hat{q}r$ by Euclid's theorem. Finally, medial axis subdivides the angle $p\hat{c}r$ into two corners of adjacent blocks.

EOP

4. SPECIAL CASES

We describe two important special cases that can happen in the blocking scheme from Algorithm 2. The first one is about poor angle at an interior corner of a block. The second one is the possibility of very thin blocks caused by degeneracy in Delaunay triangulation.

4.1 Poor angle

The Angle Lemma suggests that poor quality interior triangles in Delaunay triangulation will induce poor block angles. In two dimensions, there are two ways a triangle can become very poor. Either it has a very small angle, or it has a very large angle. The first row of Figure 10 gives examples of small angle and large angle of interior triangles in Delaunay triangulation. These poor triangles influence the poor quality of the blocks in the second row. The poor triangles are drawn with thick lines.

There is an extensive list of literature on removing poor quality triangles in Delaunay mesh [12, 22, 13, 11, 23]. The technique is commonly called Delaunay refinement. The main idea of Delaunay refinement is to repeatedly insert new points at circumcenters of poor triangles. The process is guaranteed to terminate because poor triangles have large circumcircles. The new points are always placed far away from existing points.

We would like to apply the idea of Delaunay refinement to our problem. Instead of inserting circumcenters, we will insert circular holes at circumcenters of interior triangles with poor quality. We call this algorithm *macro Delaunay refinement.*

Algorithm 3: Macro Delaunay refinement.

```
while ∃ poor quality interior triangle $\sigma$
  $c$ = circumcenter of $\sigma$
  $r$ = circumradius of $\sigma$
  insert circular hole centered at $c$ with radius $r/2$
endwhile
```

In Algorithm 3, we define poor quality triangle as the one with the ratio r/ℓ of circumradius per shortest edge greater than 2.0. It can be shown that having the ratio r/ℓ greater than 2 is equivalent to having smallest angle less than $\arcsin(1/4) \approx 14.48°$. Improving smallest angle above 14° implies having largest angle below 152°. The following lemma argues for the termination of macro Delaunay refinement. The proof is very similar to the one for a typical Delaunay refinement [12, 22, 13, 11, 23].

Lemma 2: Macro Delaunay refinement terminates.

Proof We define the *gap* between two circles as the shortest distance between the two circles. For example, it is the distance $\parallel c_1 - c_2 \parallel - (r_1 + r_2)$ where c_1, c_2 and r_1, r_2 are centers and radii of the two circles, assuming that one circle is not inside another. We will show that macro Delaunay refinement preserves the minimum gap G among all circular holes and the bounding circle.

Consider an interior triangle with poor quality $r/\ell > 2$. The length ℓ of its shortest edge is greater than the minimum gap G, i.e. $\ell > G$. Thus, the newly inserted hole of radius $r/2$ has a gap of at least $r/2 > \ell > G$ to all existing holes and the bounding circle.

EOP

The last row of Figure 10 shows the structured meshes after insertion of a circular hole. The newly inserted holes are filled with two standard templates.

4.2 Degeneracy

It is customary to assume non-degeneracy for Delaunay triangulation. The non-degenerated assumption is that the point sample contains no four co-circular points. A technique like symbolic perturbation [24] can be applied to create a valid triangulation of a point set with degeneracy. In our case, however, two interior triangles sharing the same circumcircle will create a medial arc of zero length. Similarly, two interior triangles that are nearly co-circular will create a very short medial arc. These anomalies will give rise to very thin blocks with zero or almost zero area (Figure 11).

We say four or more holes are *co-circular* when there is an empty circle tangent to all of them. A group of n co-circular holes create $(n-2)$ co-circular interior triangles, and Algorithm 2 will create $2(n-3)$ blocks with zero area.

In the case of many co-circular holes, an interior triangle with very small angle will reside in the triangulation. We can expect an angle as small as $180°/n$, where n is the number of co-circular holes. We can use macro Delaunay refinement to resolve this problem when n is larger than 12.

If only a few holes are co-circular, collapsing the very short medial arc would be preferable. The disadvantage of this approach is that high degree vertex will be created. One collapsing operation will replace two vertices of degree d_1, d_2 with a new vertex of degree $(d_1+d_2)-4$. Collapsing an arc of four co-circular holes will create a vertex of degree (6+6)-4 = 8. Collapsing two arcs of five co-circular holes will create a vertex of degree (8+6)-4 = 10.

It is not clear what is the best strategy to choose between macro Delaunay refinement and collapsing of medial arcs. A reasonable strategy is to first perform macro Delaunay refinement to eliminate all angles smaller than 15°. This will eliminate the short medial arcs resulted from 12 or more co-circular holes. Then, the collapsing operations are performed on the remaining short medial arcs.

5. RESULTS

We present three examples of our results. In each case, the mesh is generated by applying Algorithm 2 followed by a simple transfinite interpolation inside each block. No mesh smoothing is performed in order to evaluate the direct outcome of our algorithm.

Figure 12 shows `Hexagon` data. It consists of seven circular holes arranged in a regular pattern of hexagonal packing. This data is ideal for our algorithm. All the interior triangles in Delaunay triangulation have good quality. The mesh contains 36 blocks. The quadrilateral elements have minimum angle 46° and maximum angle 130°. On average, an element has smallest angle 75° and largest angle 104°. Every block is orthogonal to the circular boundary.

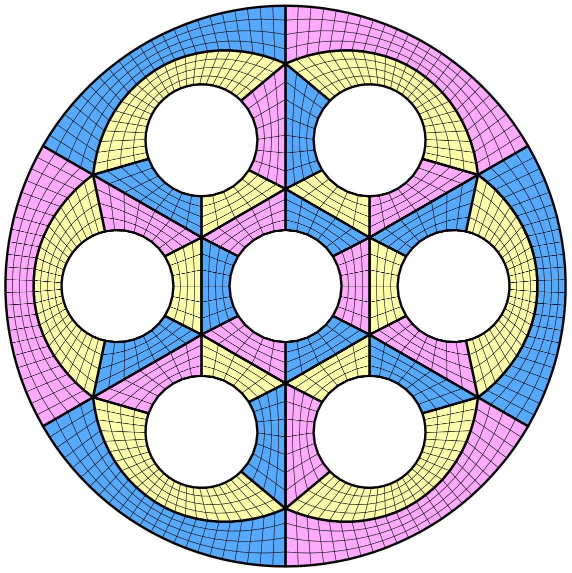


Figure 12: `Hexagon` data consists of 7 holes. The mesh contains 36 blocks.

Figure 13 shows the mesh of `Bubble` data. It represents 20 random bubbles in a periodic domain. The mesh was generated by tiling the twenty circles nine times in a three-by-three array. Then, we choose a section of the mesh in the middle. This technique yields a mesh with periodic boundary condition. Each block on one side of the zigzag boundary matches another block on the opposite zigzag side. In other words, we can tile the space by infinite copies of this mesh. This property is useful for several kinds of flow simulation.

Notice that there are a few areas where four or more bubbles are almost co-circular. They result in a number of very thin blocks. In principle, we could have "zipped" them away.

The mesh of `Bubble` data consists of 120 blocks. The quadrilateral elements have minimum angle 31° and maximum angle 148°. On average, an element has smallest angle 65° and largest angle 112°.

Figure 14 presents our last data set `Lattice`. It consists of 12 large particles interleaved with 9 small particles. The mesh consists of 120 blocks. Each small particle is adjacent to four blocks. Each large particle in the inner ring is adjacent to eight blocks. Each large particle in the outer ring is adjacent to five blocks. The quadrilateral elements have minimum angle 39° and maximum angle 138°. On average, an element has smallest angle 73° and largest angle 106°. The number of blocks is about six times the number of particles. Generating the 120 blocks manually could have taken days of user interaction. In this particular example, we can generate the mesh in a few minutes.

6. DISCUSSION

We have presented an automatic algorithm to construct blocking schemes for multiblock structured meshes in multiphase flow problems. The capability to generate the meshes automatically without human intervention is desirable for our target applications of multiphase flow with moving particles. To complete the task, more development will include application of mesh smoothing and a better scheme for node distribution. All the meshes presented in this report are generated using transfinite interpolation, and the mesh quality is acceptable. Some preliminary tests showed that applying PDE-based mesh smoothing will further improve the mesh quality.

Currently we distribute the nodes equally spacing along the four sides of each block. This forces an artificial constraint on matching which point m along a medial arc to which point b along the domain boundary. A more natural way is probably to let the maximal empty circle determine the matching. If the empty circle centered at m touches a point a on domain boundary, the node m on one side of the block should be matched to the node a on the opposite side of the block. We already satisfy this property in the case of medial vertex c with its empty circle touching three points. We would like to extend this property to the case of the empty circle touching two points. However, it is not straightforward to apply this continuous idea to the discrete setting of Delaunay triangulation.

A natural extension of a 2D problem is a 3D problem. The 3D version of our problem will consider a set of spherical particles in a bounding sphere. It is unfortunate that Voronoi diagram in three dimensions do not approximate medial surface due to the presence of sliver tetrahedra in Delaunay triangulation. Many researchers are working on a remedy of this problem [20, 21]. Assuming that this issue can be resolved, the next step would be generating structured meshes on each patch of the medial surface and projecting the nodes on each patch to domain boundary using the principle of empty sphere. It remains to be seen how well this approach will perform.

ACKNOWLEDGEMENT

The authors thank Cecil Armstrong for sending the papers [2, 3, 4]. We also thank Lin Zhang of Theoretical and Applied Mechanics at UIUC for the result of the flow simulation. The work of the first author is supported by the Center for Simulation of Advanced Rockets (CSAR) at the University of Illinois at Urbana-Champaign. The CSAR research program is supported by the US Department of Energy through the University of California under subcontract B523819.

References

[1] P. Bagchi and S. Balachandar. Effect of turbulence on the drag and lift of a particle. *Phys. Fluids.* **15** (2003), 3496–3513.

[2] T. K. H. Tam and C. G. Armstrong. 2D Finite element mesh generation by medial axis subdivision. *Adv. Eng. Software* **13** (1991), 313–324.

[3] M. A. Price and C. G. Armstrong. Hexahedral mesh generation by medial surface subdivision: part I. solids with convex edges. *Int. J. Numer. Meth. Eng.* **38** (1995), 3335–3359.

[4] M. A. Price and C. G. Armstrong. Hexahedral mesh generation by medial surface subdivision: part II. solids with flat and concave edges. *Int. J. Numer. Meth. Eng.* **40** (1997), 111–136.

[5] A. Sheffer, M. Etzion, A. Rappoport, and M. Bercovier. Hexahedral Mesh Generation Using the Embedded Voronoi Graph. *Engineering with Computers* **15** (1999), 248–262.

[6] A. Sheffer, M. Bercovier. Hexahedral Meshing of Non-Linear Volumes Using Voronoi Faces and Edges. *Int. J. Numer. Meth. Eng.* **49** (2000), 329–351.

[7] H. Blum. A transformation for extracting new descriptions of shape. *Models for the Perception of Speech & Visual Form,* Ed. W. Wathen Dunn, MIT Press, 1967, 362–380.

[8] D. T. Lee and R. L. Drysdale. Generalization of Voronoi diagrams in the plane. *SIAM J. Comp.* **10** (1981), 73–87.

[9] D. T. Lee. Medial axis transformation of a planar shape. *IEEE Trans. PAMI.* **4** (1982), 363–369.

[10] B. Delaunay. Sur la sphère vide. *Izv. Akad. Nauk SSSR, Otdelenie Matematicheskii i Estestvennyka Nauk* **7** (1934), 793–800.

[11] H. Edelsbrunner. *Geometry and Topology for Mesh Generation.* Cambridge Univ. Press, England, 2001.

[12] J. Ruppert. A Delaunay refinement algorithm for quality 2-dimensional mesh generation. *J. Algorithms* **18** (1995), 548–585.

[13] J. Shewchuk. Tetrahedral mesh generation by Delaunay refinement. *In* “Proc. 14th Ann. ACM Sympos. Comput. Geom.”, 1998, 86–95.

[14] P. P. Pebay. A New Approach Towards Delaunay-Conformity in 3 Dimensions. *In* “Proc. 9th Int. Meshing Roundtable”, 2000, 283–292.

[15] S.-W. Cheng and S.-H. Poon. Graded Conforming Delaunay Tetrahedralization with Bounded Radius-Edge Ratio. *In* “Proc. 14th Ann. ACM-SIAM Sympos. Disc. Algo.”, 2003, 295–304.

[16] S.-W. Cheng, T. K. Dey, E. Ramos, and T. Ray. Quality Meshing for Polyhedra with Small Angles. *In* “Proc. 20th Ann. ACM Sympos. Comput. Geom.”, 2004, 290–299.

[17] G. F. Voronoi. Nouvelles applications des paramètres continus à la theéorie des formes quadratiques. *J. Reine Angew. Math.* **133** (1907), 97–178, and **134** (1908), 198–287.

[18] X. Yu, J. A. Goldak, and L. Dong. Constructing 3D discrete medial axis. *In* “Proc. ACM Sympos. Solid Modeling Found. and CAD/CAM Applic”, 1991, 481–492.

[19] P. J. Frey and P.-L. George. *Mesh Generation. Application to finite elements.* Hermès Science Publ., Paris, Oxford, 2000.

[20] N. Amenta, S. Choi, and R. K. Kolluri. The power crust, unions of balls, and the medial axis transform. *Comput. Geom. : Theory and Applications* **19** (2001), 127–153.

[21] T. K. Dey and W. Zhao. Approximating the medial axis from the Voronoi diagram with a convergence guarantee. *Algorithmica*, **38** (2004), to appear.

[22] L. P. Chew. Guaranteed-quality Delaunay meshing in 3D. *In* “Proc. 13rd Ann. Sympos. Comput. Geom.”, 1997, 391–393.

[23] H. Edelsbrunner and D. Guoy. Sink Insertion for Mesh Improvement. *Int. J. Found. Comput. Sci.*, **13** (2002), 223–242.

[24] H. Edelsbrunner and E. P. Mücke. Simulation of Simplicity: a technique to cope with degenerate cases in geometric algorithms. *ACM Trans. Graphics* **9** (1990), 66–104.

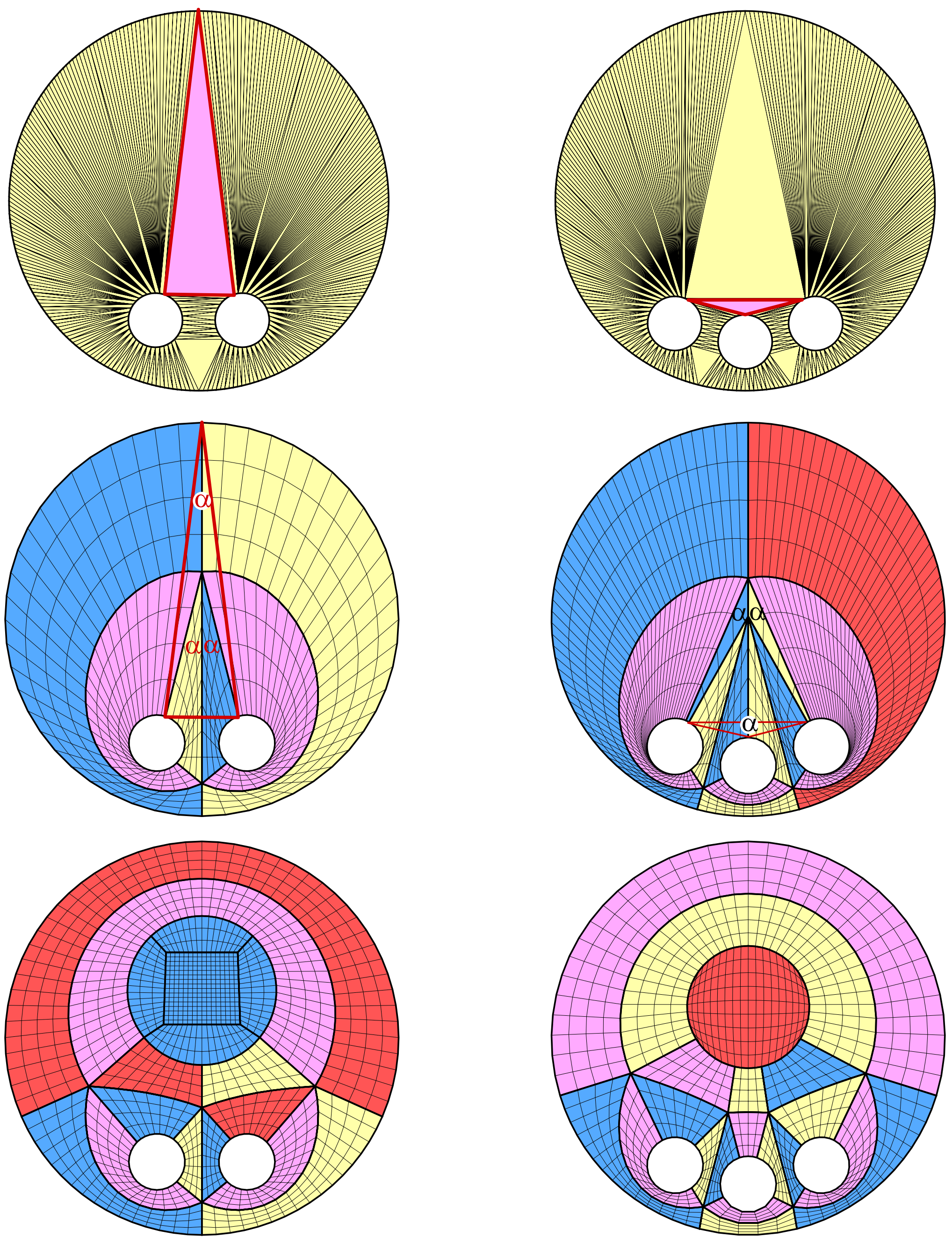


Figure 10: Small angle and large angle of interior triangles in Delaunay triangulation (first row), induced poor blocks (second row), and improvement after macro Delaunay refinement (last row).

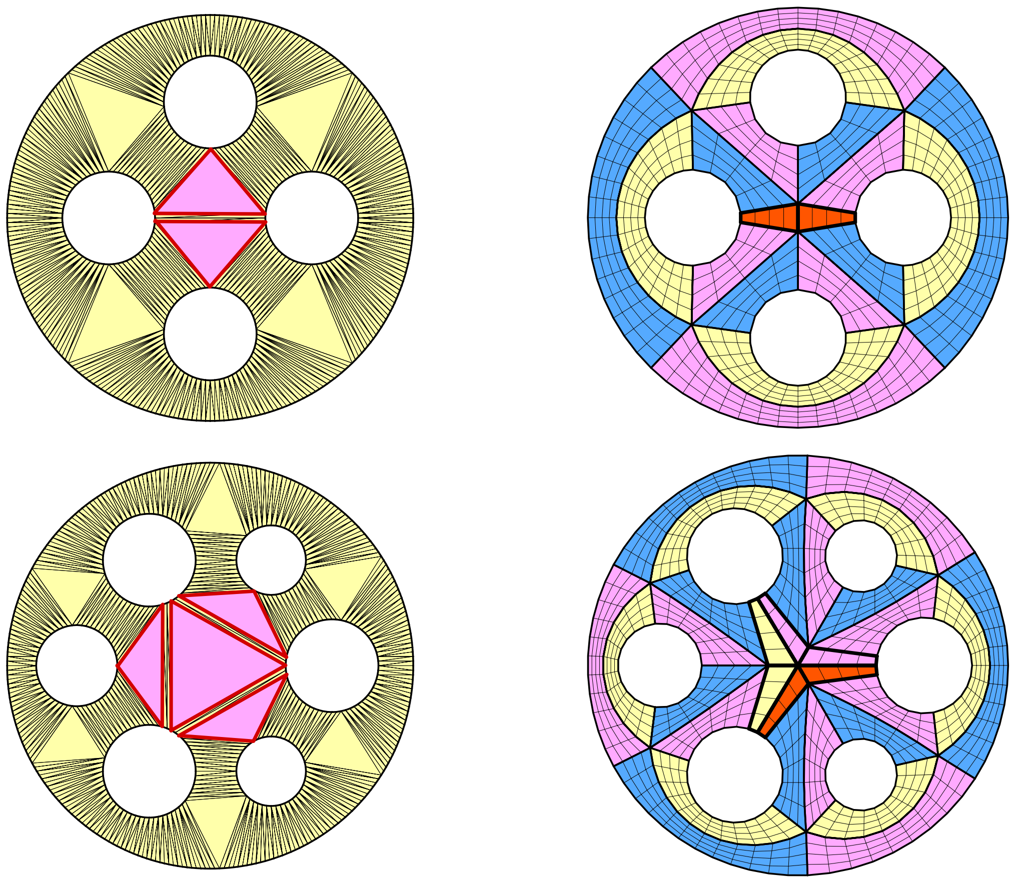


Figure 11: On the top, two almost co-circular interior triangles induce one pair of very thin blocks. On the bottom, four almost co-circular interior triangles induce three pairs of very thin blocks

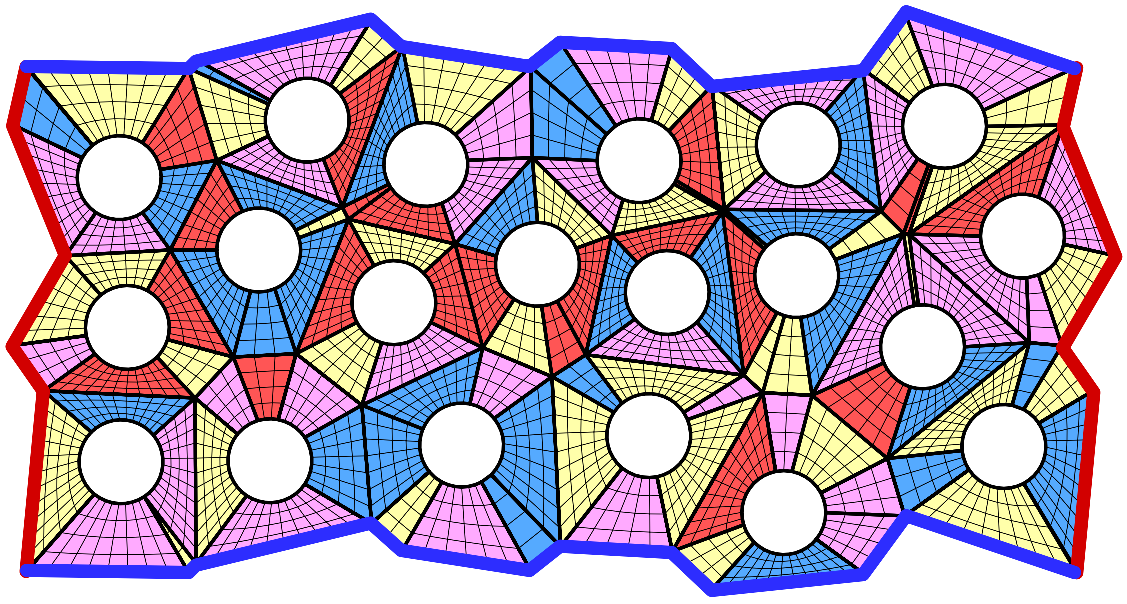


Figure 13: `Bubble` data consists of 20 circles. The periodic mesh contains 120 blocks.

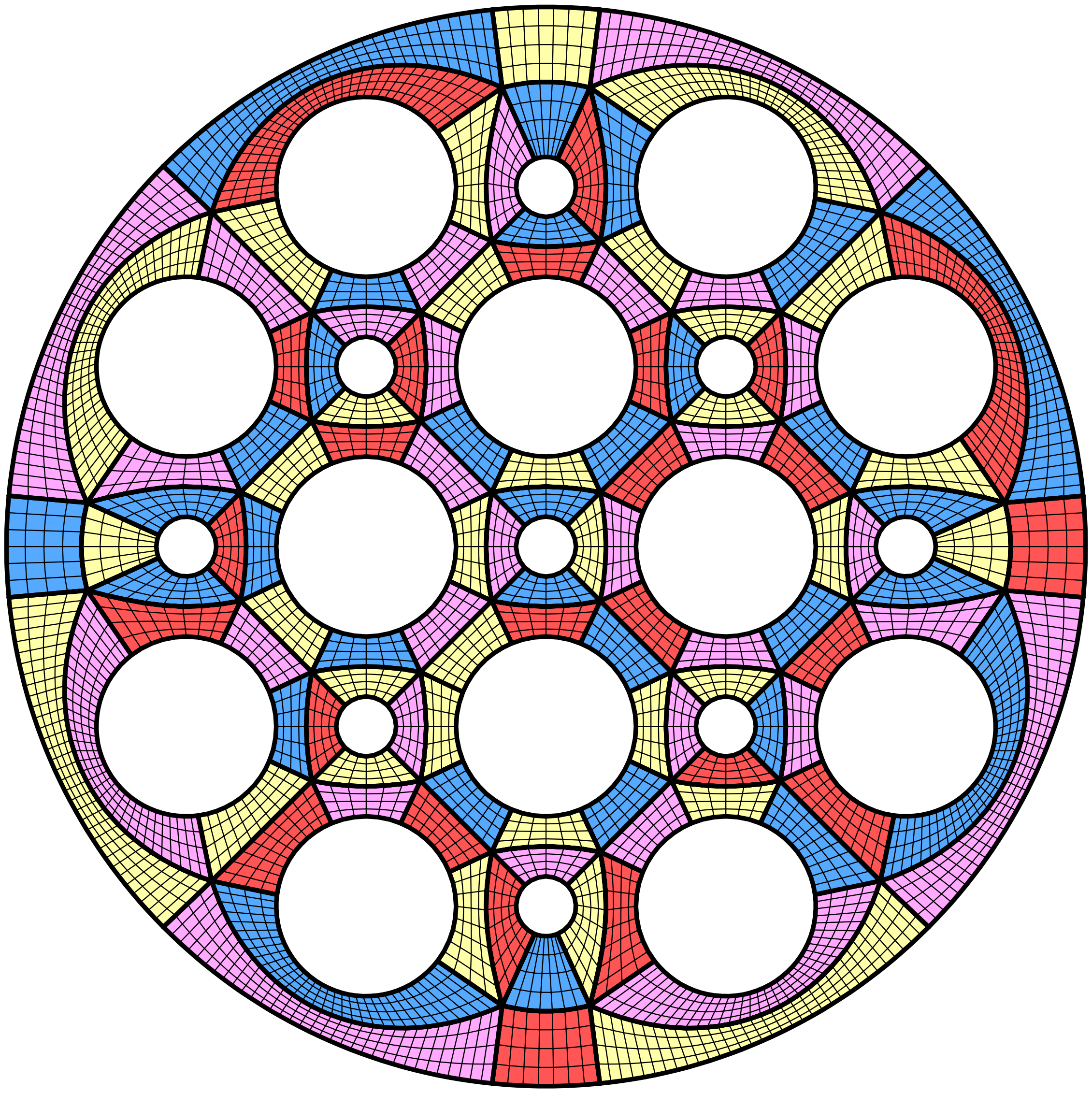


Figure 14: `Lattice` data consists of 12 large holes and 9 small holes. The mesh contains 120 blocks.