

QUADRILATERAL MESHING WITH ANISOTROPY AND DIRECTIONALITY CONTROL VIA CLOSE PACKING OF RECTANGULAR CELLS

Naveen Viswanath¹, Kenji Shimada², Takayuki Itoh³

¹ *Mechanical Engineering, Carnegie Mellon University, Pittsburgh, PA, U.S.A. <naveen@biosys.net>*

² *Mechanical Engineering, Carnegie Mellon University, Pittsburgh, PA, U.S.A. <shimada@cmu.edu>*

³ *IBM Research, Tokyo Research Laboratory, Yamato, Kanagawa, Japan. <itot@jp.ibm.com>*

ABSTRACT

A new technique for automatically generating anisotropic quadrilateral meshes is presented in this paper. The inputs are (1) a 2-D geometric domain and (2) a desired anisotropy – defined as a metric tensor over the domain – specifying mesh sizing in two independent directions. Node locations are obtained by closely packing rectangles in accordance with the inputs. These nodes are then connected using anisotropic Delaunay triangulation that takes into account the desired anisotropy. The obtained triangular mesh is converted into a quadrilateral mesh using mesh conversion templates. The novelty of the method is that closely packed rectangles resemble a pattern of Voronoi polygons corresponding to a well-shaped quadrilateral mesh. The result is a high quality mesh that conforms well to the input. As an application, this method is used to generate a mesh to solve a steady state heat transfer problem.

Keywords: mesh generation, anisotropy, computational geometry, quadrilateral, circumcircle test,

1. INTRODUCTION

Mesh generation is used in a variety of areas like finite element method and computer graphics. Today, such computational techniques form an integral part of design and analysis. Mesh generation using quadrilateral elements is computationally expensive due to constraints on element size and shape, mesh directionality control and adaptive remeshing capabilities. The problem is further restricted by specifications on anisotropy. Therefore a high quality, automatic mesh generation algorithm can increase productivity by reducing the time spent on generating the mesh.

This paper describes a new computational method by which a two-dimensional domain can be meshed using anisotropic quadrilateral elements. It has been shown that quad elements perform better in the FEM analysis of plane stress [1]. They are also preferred in computational fluid dynamics, sheet metal bending, automobile crash simulation. In addition, an anisotropic mesh aligned to specific directions, is better in terms of computational cost and solution accuracy when the physical phenomena being analyzed has a strong directionality or when the material properties are anisotropic. For example analyses involving shock propagation, or analysis of fiber-reinforced glass or plastics.

The method used is an extension of the bubble mesh method [2, 3] in which bubbles or spheres were packed to obtain node locations suitable for an isotropic triangular mesh. Later ellipsoids were packed to generate anisotropic triangular meshes[4]. Square packing was done to obtain isotropic quadrilateral meshes[5]. In the current method, rectangular cells are packed to obtain the desired anisotropy.

This method is readily extensible to three dimensions by packing parallelepipeds instead of rectangles.

The advantage of this approach is that node spacing and mesh directionality can be precisely controlled, independent of the boundary edges. This can be important to capture physical phenomena like high stress gradients in stress analysis or shocks in fluid flow. Locally, the size and orientation of rectangles are adjusted based on the input sizing and direction information. This results in a quadrilateral mesh that is anisotropic, well-shaped and well-aligned.

Another advantage is that adaptive remeshing is not computationally expensive because dynamic simulation can be continued from the existing mesh, instead of starting from scratch. This is an advantage of using the bubble mesh technique of physically based dynamic simulation for close packing.

As a demonstration of the capabilities of this technique, a steady state heat transfer problem is solved using a mesh generated in this way. Since the central theme of this paper is anisotropic quadrilateral mesh generation, this is described separately in section 6.

2. PROBLEM STATEMENT

The problem addressed in this paper can be stated as follows

Given:

1. a two dimensional geometric domain
2. a desired mesh anisotropy - element sizing and mesh directionality given as a 2x2 tensor field \mathbf{M}

Obtain:

a well-shaped, graded, anisotropic quadrilateral mesh, conforming to the input anisotropy – specified using node spacing and mesh directionality.

3. PREVIOUS WORK

There have been several reviews of mesh generation algorithms. [6-9]. Recently there have been surveys of mesh generation algorithms and software using them, published on the world wide web [10-12].

Quadrilateral meshing has been implemented using a variety of techniques. One of the popular methods is node placement followed by connection. This is popular because of the existence of a robust scheme for connection called Delaunay Triangulation. The obtained triangular mesh is converted into a quadrilateral mesh. Many such conversion methods have been proposed [13-16]. A CSG based approach for node placement was proposed by Lee [17, 18]. In paving [19], quadrilateral elements are created one by one, starting from the boundary. The Q-Morph algorithm proposed by Owen [20] works similarly, converting a triangular mesh to a quadrilateral mesh starting from the boundary. In these advancing front methods, mesh directionality cannot be controlled independent of the boundary, which may be needed to reflect load conditions or material properties.

Castro-Diaz et al. showed the use of a metric tensor to improve the quality of adapted meshes in flow computations [21]. Borouchaki et al. demonstrated the use of a metric tensor to generate an anisotropic triangular mesh [22]. Bossen and Heckbert used a 2×2 metric tensor to generate an anisotropic triangular mesh using a system of interacting particles [23]. Shimada et al. used ellipse packing to obtain anisotropic triangular meshes [4]. The biting ellipses scheme [24] uses a method that combines paving and packing to generate an anisotropic triangular mesh. Though this method has a theoretical time bound for convergence, it takes a long time for meshing even simple geometries.

This work draws on both isotropic quadrilateral meshing techniques and anisotropic triangular meshing methods and incorporates new techniques to realize efficient, high-quality anisotropic quadrilateral meshing.

4. ANISOTROPIC QUADRILATERAL MESH GENERATION

4.1 Outline of Technical approach

The anisotropic quadrilateral meshing problem is solved in the following way.

- Step 1 : Place rectangular cells on the all vertices
- Step 2 : Pack rectangular cells on the all edges
- Step 3 : Pack rectangular cells on the faces
- Step 4 : Place nodes at the center of rectangles
- Step 5 : Generate triangular mesh topology using anisotropic Delaunay Triangulation

Step 6 : Selectively combine pairs of triangles to obtain a quad-dominant mesh

Step 7 : Use mesh conversion templates to obtain an all-quad mesh

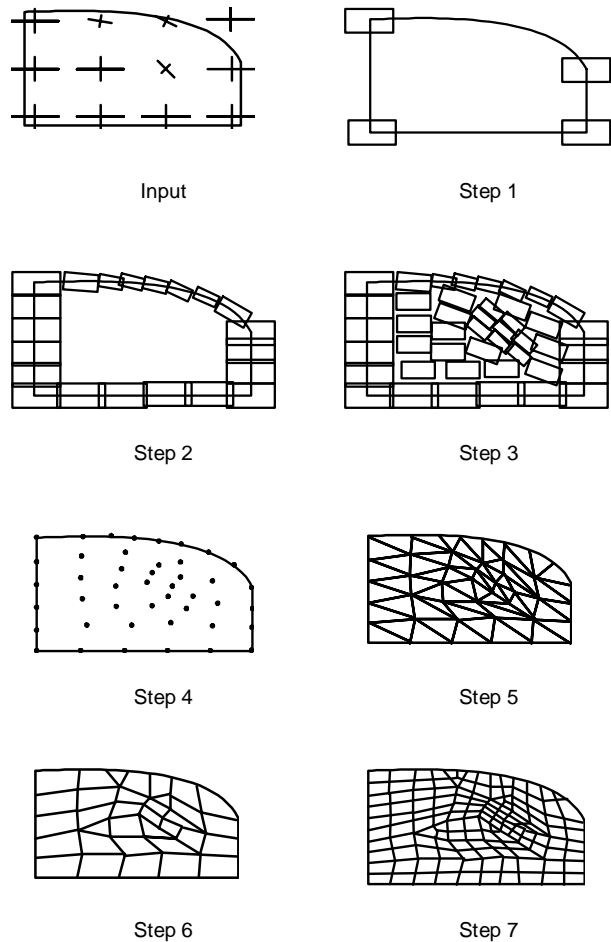


Figure 1 The mesh generation process

Figure 1 shows this step by step process. The first three steps generate suitable node locations by closely packing rectangles, which are in accordance with the given input. This is done because packed rectangles mimic a pattern of Voronoi polygons for a well-shaped quadrilateral mesh (figure 2). The sizes and directions of the rectangles are adjusted based on the given mesh sizing and mesh directionality information.

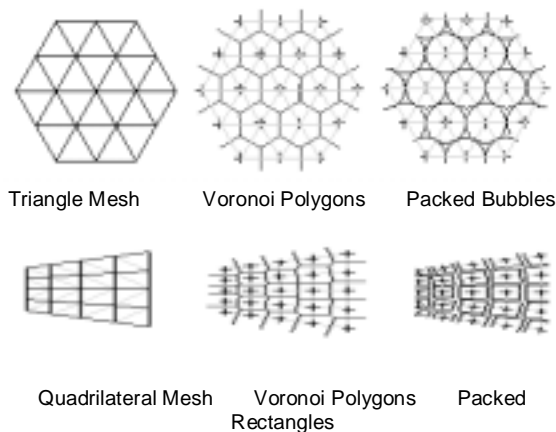


Figure 2 Voronoi Polygons and Mesh

Some of the issues that arise are

1. What are optimal locations of rectangles
2. How many rectangles are needed to fill the domain
3. How to modify Delaunay triangulation to produce anisotropic triangles
4. How to convert the obtained rectangle packed triangles to quads.

For the first issue we use a physically-based model, similar to a particle system in computer graphics. A proximity-based force field is defined between the rectangles so that either an attracting force or a repelling force is applied based on the distance between the two rectangles in consideration. The equations of motion are solved numerically assuming a point mass at the center of the rectangles and viscous damping.

The second issue is resolved by checking the local population density and adaptively adding or removing rectangles during the dynamic simulation of the motion of the rectangles. After an iteration, if a region has an excessive number of rectangles, then rectangles are removed, and if there are significant holes, rectangles are added.

Delaunay triangulation tries to locally maximize the minimum angle in pairs of triangles, which ideally produces equilateral triangles. This is often not the case in anisotropic meshing, where long, slender triangles maybe preferred. So, the circumcircle test is suitably modified using the input metric tensor to evaluate which of the two possible triangulations of 4 nodes are desirable.

Once the triangular mesh is obtained, it is first converted to a quad-dominant mesh using the input directionality data, and then to an all-quad mesh using mesh conversion templates. Each of these topics is addressed in the following subsections.

4.2 Mesh Anisotropy - Specification, Interpretation

An input 2x2 metric tensor field is used to specify the desired anisotropy. As in previous work [4, 21-23] this tensor is

symmetric, positive-definite. For completeness, the following topics are explained:

1. How to calculate values of mesh sizes and directions, given a metric tensor
2. How to specify the metric tensor for a required anisotropy

Mesh directionality, mesh sizing and therefore, anisotropy can be specified in a single compact form using tensor notation. Anisotropy implies that one edge of the element has significantly different length than the other(s) – resulting in elements that are stretched. Though these three concepts are inter-related, it is important to understand that anisotropy cannot be obtained using mesh directionality control and mesh sizing control together. We can obtain anisotropy only if the sizing can be controlled independently in different directions. This is different from just using isotropic quads and orienting it along a given direction. Figure 3 explains this distinction.

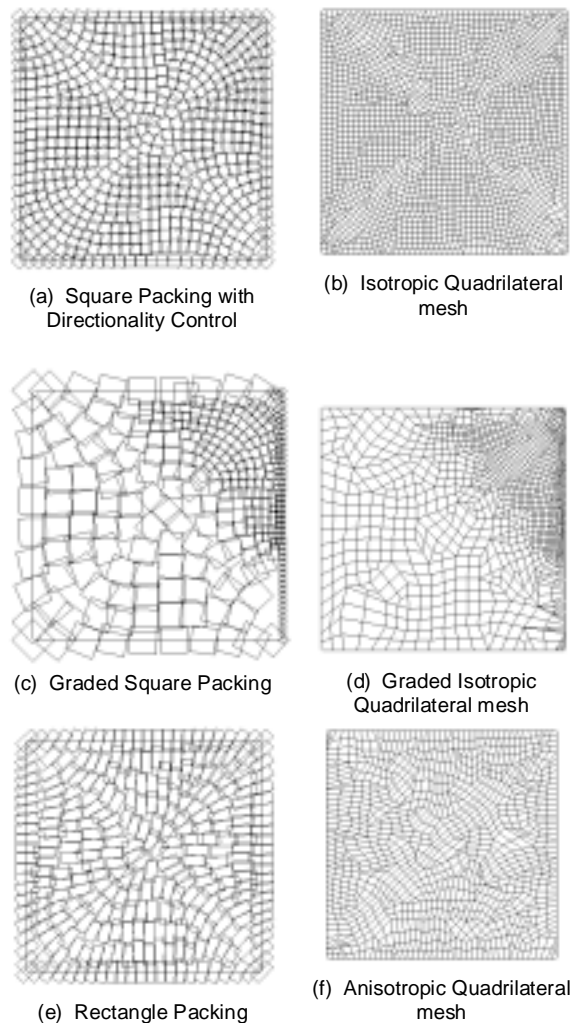


Figure 3 Square Packing vs. Rectangle Packing

The meshes in figure 3 have the same input geometry and mesh directionality. In the figure 3(a), squares of the same size are packed in the domain according to specified mesh directions. This results in an isotropic quadrilateral mesh 3(b). Even if the size of the squares are varied as shown in figure 3(c), node locations suitable for a graded quadrilateral mesh are obtained, but the mesh 3(d) is still be isotropic. By virtue of packing rectangles with specified aspect ratios 3(e), an anisotropic quadrilateral mesh is obtained by controlling the node spacing in two different directions. This is shown in figure 3(f).

The procedure to calculate the required anisotropy from the given metric tensor is described first. Following that, the method used to specify the metric tensor, given the anisotropy is explained.

Given a 2x2 tensor \mathbf{M} , we obtain the two eigen values λ_i by solving[25]

$$|\mathbf{M} - \lambda \mathbf{I}| = 0 \quad (1)$$

where \mathbf{I} is the 2x2 identity matrix. Once the eigen values are found, the eigen vectors \mathbf{x}_i are found using

$$\mathbf{M}\mathbf{x}_i = \lambda_i \mathbf{x}_i, \quad i = 1, 2 \quad (2)$$

The eigen vectors are the directions of the major and minor axes specifying the mesh directionality and the eigen values are the inverse of the squares of the major and minor radii.

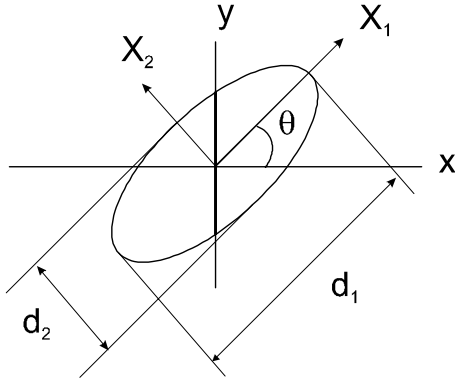


Figure 4 Specification, interpretation of a tensor as an ellipse

Analogously, given the directions and magnitudes of the axes (the desired mesh sizing and directionality as described above), the tensor can be calculated using

$$\mathbf{M} = \mathbf{R}\mathbf{A}\mathbf{R}^T \quad (3)$$

$$\mathbf{R} = \begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix} \quad (4)$$

$$\mathbf{A} = \begin{pmatrix} (2/d_1)^2 & 0 \\ 0 & (2/d_2)^2 \end{pmatrix} \quad (5)$$

As shown in figure 4, θ is the angle between the major axis of the ellipse and the positive x-axis and d_1 , d_2 are the major and minor axis diameters.

In our implementation, we specify this by

1. a direction \mathbf{X}_2 specifying the minor axis
2. a base size along \mathbf{X}_2
3. an aspect ratio

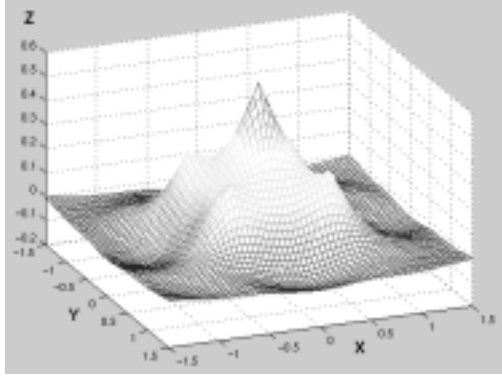
The single direction specifies the orientation of the minor axis. Since the principal axes are perpendicular to each other, the direction of the major axis can be easily calculated. Mesh sizing information is given as a base size along the minor axis and an aspect ratio – a ratio of the diameter of the major axis to that of the minor axis. The specified “stretching” can easily be visualized using this aspect ratio.

4.3 Close Packing of Oriented Rectangular Cells

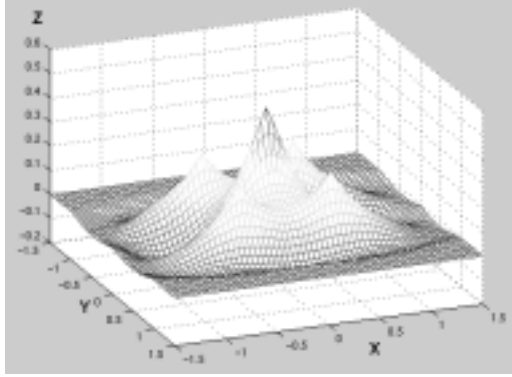
To get appropriate node locations, rectangles are packed closely according to the given input. A physically-based particle simulation approach is used to solve this rectangular packing problem efficiently.

The input metric tensor field is used to specify the length, breadth and orientation of quadrilateral elements in the domain. This data is stored in a background grid at discrete locations. At intermediate points, the data at the four grid nodes enclosing the point are linearly interpolated. Alternatively, if input is given only at a few points, then values at all nodes are calculated using a Laplacian smoothing [26] type approach to obtain a smoothly varying field.

In the triangular bubble mesh technique [2, 3] a force similar to Van der Waals force produces hexagonal packing of bubbles. This field produces an attractive force if two particles are farther than a stable distance, and a repulsive force if they are located closer than the stable distance. Node locations obtained this way create a well-shaped isotropic triangular mesh. This field was modified in square packing to obtain node locations suitable for isotropic quadrilateral mesh generation [5] by adding four sub-fields to the main force field. In the proposed method, the field used for square packing has been “stretched” using the input aspect ratio, to obtain rectangular packing.



(a) Force field used for square packing



(b) Force field used for rectangle packing

Figure 5 Force fields used for packing

Figure 5 (a) shows the force field used in square packing. In 5 (b) the given aspect ratio is locally constant and is 2 along the x-axis. The corresponding potential function can be obtained by integrating the force. The rectangles try to occupy minimum energy positions in this field. Packing is complete when the geometry is covered sufficiently, without any significant gaps or overlaps.

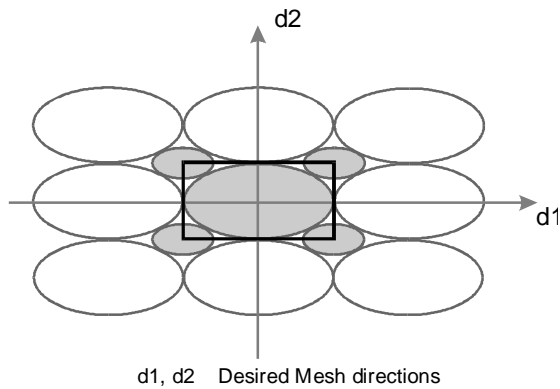


Figure 6 Rectangle packing using ellipses

Ellipses can be used to achieve rectangle packing as shown in figure 6. After the central rectangle, other stable positions for rectangles are the along the directions d_1 and d_2 . Once these are also occupied, the ones in between can be filled. For a

graded anisotropic quadrilateral mesh (the base size of the rectangles and the aspect ratio may vary), the fields and the sub-fields have to be suitably adjusted to obtain packing of rectangles of desired sizes which would in turn give us a mesh consistent with the input.

Using this proximity-based force, a physically-based relaxation method is used to find a closely packed configuration of rectangles. Due to the non-linear nature of the force and complicated geometric constraints, force equilibrium equations become highly non-linear. Therefore it is difficult to solve the equation using a multi-dimensional root finding technique such as the Newton-Raphson method.

The solution to this problem is to assume a point mass m at the center of each rectangular cell and a viscous damping c and solve these equations of motion (equation 6) using a numerical integration scheme like the fourth-order Runge-Kutta method. The positions of rectangles \mathbf{x}_i are thus obtained.

$$m \ddot{\mathbf{x}}_i(t) + c \dot{\mathbf{x}}_i(t) = \mathbf{f}_i(t) \quad i = 1, 2, \dots, n \quad (6)$$

While solving this equation numerically, we adaptively adjust the number of rectangles in the domain. This is necessary as the number of rectangles needed for packing is unknown at the start. We generate the initial configuration using octree subdivision. During the simulation, we use a procedure of adding rectangles in sparse areas and deleting rectangles in dense areas, to get closely packed rectangles. This dynamic simulation and adaptive population control approach makes adaptive remeshing very efficient because we can just continue the simulation process from previous node locations without starting from a totally fresh configuration, when the geometry, node spacing or mesh directionality is changed.

4.4. Anisotropic Delaunay Triangulation

Once a force-balancing configuration of rectangles is obtained, the centers of the rectangles must be connected to form a complete triangular mesh, which is then converted into a quad mesh. In connecting nodes, Delaunay triangulation is considered suitable for finite element analysis, as the triangulation maximizes the smallest angles of the triangles. Ideally, it creates triangles as equilateral, or isotropic, as possible for the given set of points; thus thin, or anisotropic triangles are avoided whenever possible.

One important property of Delaunay triangulation is that a circumcircle of a Delaunay triangle, must not contain other nodes inside it. To check this, many Delaunay triangulation algorithms use the circumcircle test. This test is also used in Sloan's algorithm [27] which was implemented in the original 2-D isotropic bubble mesh. The circumcircle test is performed on a pair of adjacent triangles that forms a convex quadrilateral. Given such a set of four points, the test checks if the fourth point lies inside the circumcircle of the triangle formed by the other three points. If it is, the four points are then reconnected into the other possible configuration of two triangles. This test is not suitable for anisotropic meshing.

Delaunay triangulation is modified to incorporate anisotropy in the circumcircle test.

Assuming the metric tensor is locally constant, the circumcircle test is done in normalized space. The coordinates of the four nodes under consideration are transformed so that an ellipse is mapped back to a unit circle [23]. A local average tensor can be determined by first calculating the barycenter of the four nodes and then finding the metric tensor at this barycenter¹

$$\mathbf{M} = \mathbf{M} \left(\frac{\mathbf{x}_1 + \mathbf{x}_2 + \mathbf{x}_3 + \mathbf{x}_4}{4} \right) \quad (7)$$

where $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4$ are position vectors of the four nodes in question.

This metric tensor is used to transform the coordinates of the four nodes under consideration. A rotational and scaling transformation is used to map the ellipse corresponding to this tensor into a unit circle. The new coordinates are given by

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \mathbf{T} \begin{pmatrix} x \\ y \end{pmatrix} \quad (8)$$

$$\mathbf{T} = \mathbf{P}\mathbf{R}^T \quad (9)$$

(x,y)	Coordinates of nodes
(x',y')	Transformed coordinates
\mathbf{T}	Transformation matrix
\mathbf{R}	$\begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix}$
\mathbf{P}	$\begin{pmatrix} (2/d_1) & 0 \\ 0 & (2/d_2) \end{pmatrix}$
d_1, d_2, θ	Major, Minor diameters and angle of inclination obtained from \mathbf{M} , as before

So, the circumcircle test is applied in transformed normalized coordinates. This modifies the Delaunay triangulation used in the original bubble mesh program so that the input anisotropy is taken into consideration.

¹ Slightly different anisotropic Delaunay triangulation schemes are used by other researchers [21-23]. For example, an alternative way to take an average of four metric tensors is:

$$M = \frac{\sum_{i=1}^4 M(x_i)}{4}$$

Figure 7 shows how a different pair of triangles is selected when the anisotropic circumcircle test is performed after transforming the positions of the four nodes.

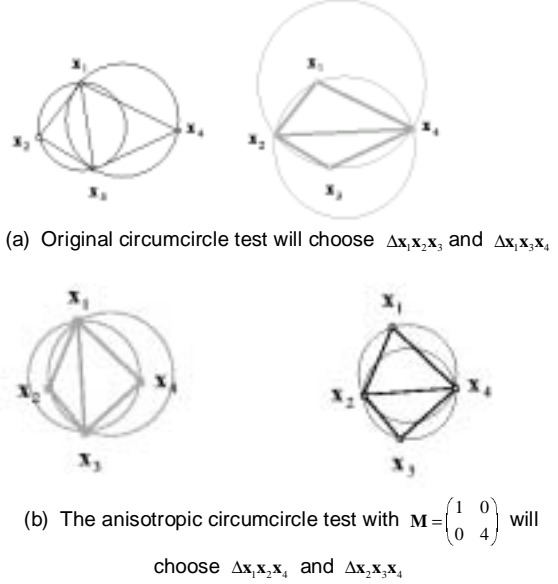
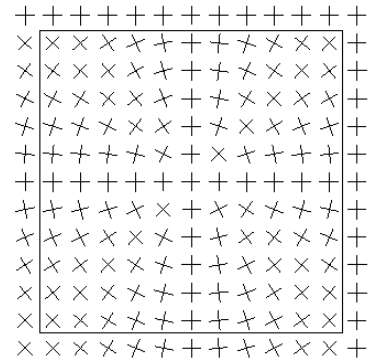


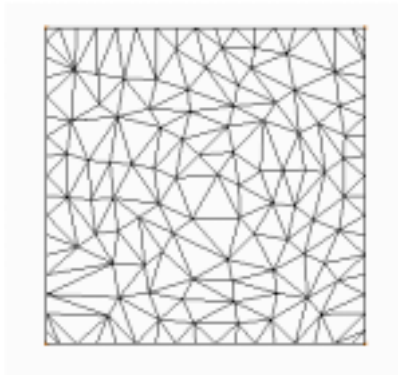
Figure 7 Anisotropic circumcircle test

In the example shown in figure 7, an aspect ratio of 2 is desired along the horizontal direction. The original circumcircle test 7(a) does not account for this and chooses the pair $\Delta_{x_1x_2x_3}$ and $\Delta_{x_1x_3x_4}$ although this does not orient the triangles according to the desired anisotropy. The modified approach first transforms the coordinates and then applies the circumcircle test. As shown in figure 7(b), this chooses the right pair of triangles.

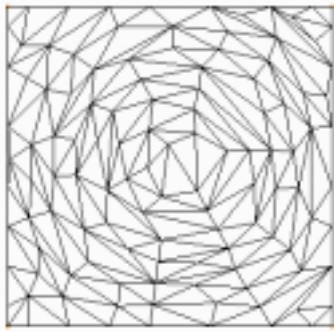
To demonstrate the effectiveness of this procedure, figure 8 contrasts the original Delaunay triangulation and the anisotropic Delaunay triangulation. Given the same inputs, the anisotropic Delaunay triangulation creates a mesh that is stretched and “flows” along the input mesh directionality and conforms better to the desired anisotropy.



(a) Input Mesh Directionality



(b) Normal Delaunay Triangulation



(c) Anisotropic Delaunay Triangulation

Figure 8 Effect of anisotropic Delaunay triangulation

4.5. Conversion to Quadrilateral Mesh

After an anisotropic triangular mesh topology is obtained, it is first converted into a quad-dominant mesh by selectively merging two triangular elements into a quadrilateral element along the given vector field. This mesh conversion algorithm consists of three stages:

1. Calculate score Ω_i that measures how well the resultant quadrilateral mesh element aligns along the specified mesh directions if the i^{th} non-boundary edge of a triangular element is removed to form a quadrilateral.
2. Sort all the non-boundary edges using a priority queue
3. Delete the edges successively from the top of the priority queue. The deletion of one edge creates one quadrilateral.

The score Ω_i is calculated by comparing the directions of the four edges of the resultant quadrilateral element with the input mesh direction vectors at the centers of the edges. For side j of the quadrilateral element i , we calculate ω_{ij} using

$$\omega_{ij} = \begin{cases} k_j |\mathbf{u}_{ij} \cdot \mathbf{v}_{ij}| & |\mathbf{u}_{ij} \cdot \mathbf{v}_{ij}| \geq \frac{1}{\sqrt{2}} \\ k_j \sqrt{1 - (\mathbf{u}_{ij} \cdot \mathbf{v}_{ij})^2} & |\mathbf{u}_{ij} \cdot \mathbf{v}_{ij}| < \frac{1}{\sqrt{2}} \end{cases} \quad (10)$$

where the subscript i is the index of a quadrilateral element and $j = 1, 2, 3, 4$ is the index of the side edge of the element. The inner product ω_{ij} is computed using \mathbf{u}_{ij} the unit vector of the edge j of quadrilateral i , and \mathbf{v}_{ij} the input mesh direction vector at the center of that edge. The product is weighted by k_j so that the input anisotropy is taken into account. For the two shorter sides of a rectangle k_j is 1 and for the longer sides it is the aspect ratio. From this definition of ω_{ij} the score Ω_i is calculated as follows

$$\Omega_i = \frac{1}{4} \sum_{j=1}^4 \omega_{ij} \quad (11)$$

After going through the above stages, a quad-dominant mesh is obtained. This is converted into an all-quad mesh using one of two mesh conversion templates [15]

1. One triangle to three quads
2. One quad to four quads

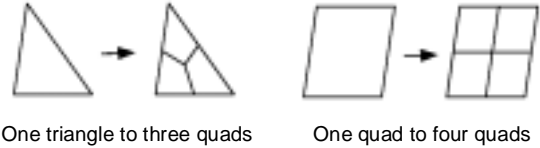


Figure 9 Mesh Conversion Templates

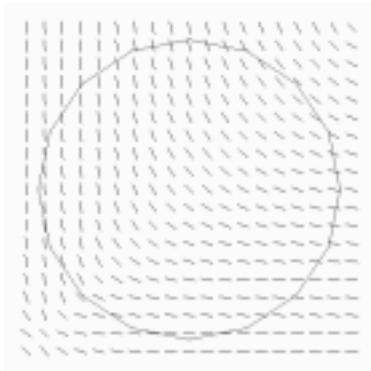
In both these methods, a new node is inserted at the centre using linear interpolation of midpoints of the edges. This is joined to each of the applicable edges to form quadrilateral mesh elements. So, the final element size is roughly half the size of elements at the previous step.

5. RESULTS

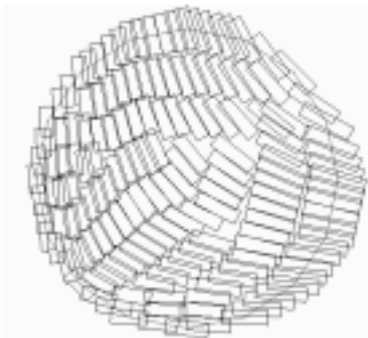
This section contains the results of the automatic anisotropic quadrilateral mesh generation technique proposed in this paper. The sequence of pictures in figures 10 and 11 is as follows. The first picture (a) is the desired mesh directionality. Picture (b) has rectangles packed in accordance with the input. The third picture is the triangular mesh obtained after anisotropic Delaunay triangulation (c). The anisotropic quadrilateral mesh is the final image (d).

The domain in figure 10 is a circle approximated using line segments. Mesh directions 10(a) are those corresponding to a family of rectangular hyperbolas. The aspect ratio is varied from 2 at the left of the domain, to 4 at the right. The rectangle packing 10(b) shows this. The intermediate

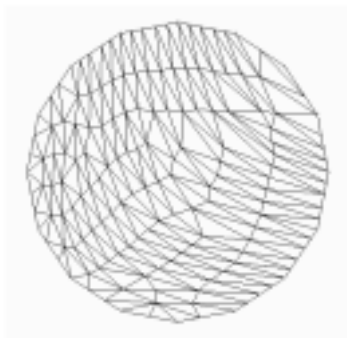
triangle mesh 10(c) conforms to the given input. The variation in aspect ratio of quadrilateral elements is seen in 10 (d).



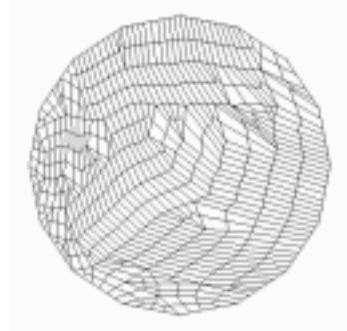
(a) Input mesh directions



(b) Packed Rectangles

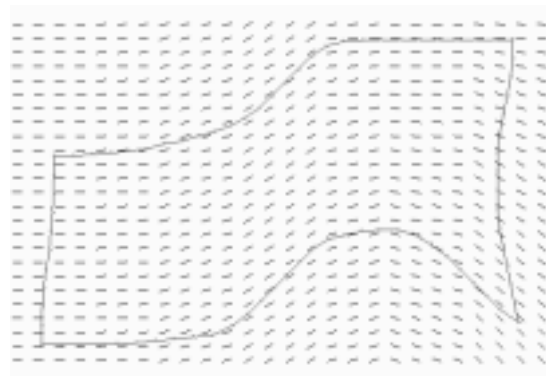


(c) Delaunay Triangulation



(d) Anisotropic Quadrilateral Mesh, Nodes 621, Elements 564

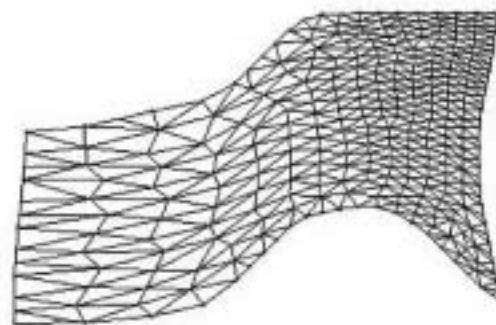
Figure 10 Mesh generation result – 1



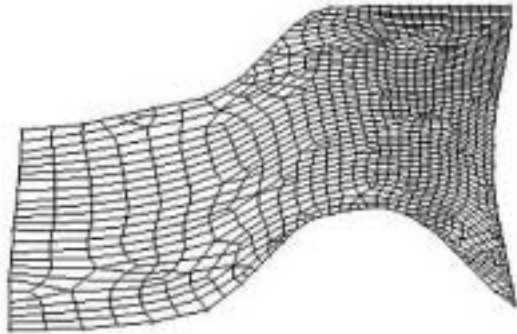
(a) Input mesh directions



(b) Packed Rectangles



(c) Delaunay Triangulation



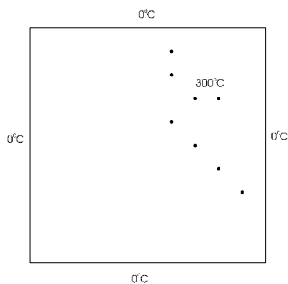
(d) Anisotropic Quadrilateral Mesh
Nodes 1303, Elements 1212

Figure 11 Mesh generation results – 2

The results shown in figure 11 showcase the full functionality of the presented technique. Mesh directionality and sizing is independently controlled in two directions – parallel and perpendicular to the specified direction (base size of rectangle and aspect ratio). The aspect ratio is linearly varied from 2 at the right end to 3 at the left end. The base size of the rectangles is 5 at the right and increases to 10 at the left. Mesh directions are specified to coincide with the boundary. This shows how a boundary-aligned mesh may be obtained using this method, without front collision problems typical of advancing front methods.

6. APPLICATION – MESH GENERATION FOR STEADY STATE HEAT TRANSFER

This section describes a procedure which can generate meshes for real world problems, using the proposed method. A mesh is generated to solve a steady-state heat-transfer problem. A preliminary analysis is used to specify the inputs required for this technique. Once the mesh is generated, it is used to solve the problem. Comparing the temperature contours of regular meshes with those of the generated mesh shows that the computational cost can be reduced without losing solution accuracy.



(a) Input geometry, loads

(b) Coarse mesh

Nodes 121

Elements 100

Figure 12 Preliminary analysis using a coarse mesh

Figure 12(a) shows the problem being solved. The eight points shown in the interior of the geometry are at 300°C . The four sides are at 0°C . The initial coarse mesh is 12(b). The contour plot of the solution is 12(c). The directions of the temperature gradients are shown in figure 12(d).

To specify the inputs required for the procedure, a preliminary analysis is done using a coarse mesh of 100 square elements. Even though a few important details may be missed if the mesh is too coarse with reference to the features in the problem, it is better than generating a mesh with no consideration to the boundary conditions, loads, or material properties.

Data exchange interfaces are used to exchange mesh and solution data between ANSYS and the proposed meshing scheme. The command line interface of ANSYS is used to read in scripts to import the generated mesh. The nodal solution and the list of nodes and elements in plain text format are used for export.

The flexibility of the process allows arbitrary specification of anisotropy – given as mesh sizes, aspect ratios and directions. Mesh directions are specified using temperature gradients calculated using the coarse mesh solution. In heat transfer problems, there are large temperature gradients near regions of high temperature. So, a small mesh size and aspect ratio is specified at regions where the high temperature is applied. This demonstrates how an expert can use this technique. A better way would be to automate this based on error bounds or other desired criteria.

The FEA code ANSYS was used to solve the problem and to make the contour plots in figure 13. The plots all have the same scale – the same shade represents the same range of temperature in the contour plots 13(b), 13(d), and 13(f).

The mesh thus generated is shown in figure 13(e). As a comparison, figures 13(a) and 13(c) show a regular mesh of 900 and 1600 elements respectively. The temperature contours for the generated mesh 13(f) seem to be better than those for the 900 element regular mesh 13(b), even though it has fewer number of elements – 828. This is a qualitative comparison, assuming the 1600 element mesh has the better solution. This is a fair assumption because, in such a straightforward problem, when the number of uniform elements are increased, better solutions are obtained. For instance, the contours are smoother and the resolution of the high temperatures is better. The second contour from high temperature spots of the generated mesh 13(f) resembles those of 13(d), more than the corresponding contour of 12(b) resembles 13(d). The number of elements is a good measure of the computational cost. Since better results are obtained with fewer elements, this leads us to believe that the proposed technique results in a reduction in computational cost without loss in accuracy of analysis.

7. CONCLUSION AND FUTURE WORK

This paper presents a new technique for generating anisotropic quadrilateral meshes in two-dimensions, using a physically - based method for obtaining a close packing of rectangles. The novelty of this technique is that closely packed rectangles resemble a pattern of Voronoi polygons that correspond to a well-shaped, well-aligned, anisotropic quadrilateral mesh.

The centers of the rectangles give the node locations. Though a triangular mesh topology is obtained initially, it is well suited for conversion to a quad mesh because rectangles have been packed using an appropriate force-field.

A significant advantage of our technique is the ability to control the mesh sizing and direction independent of the boundary. This can be used to achieve effective adaptive remeshing because meshes can be tailored to suit specific application domains and needs – like load conditions, properties of materials in stress analysis. Also, remeshing is easy as dynamic simulation can be resumed from previous node locations rather than start afresh.

REFERENCES

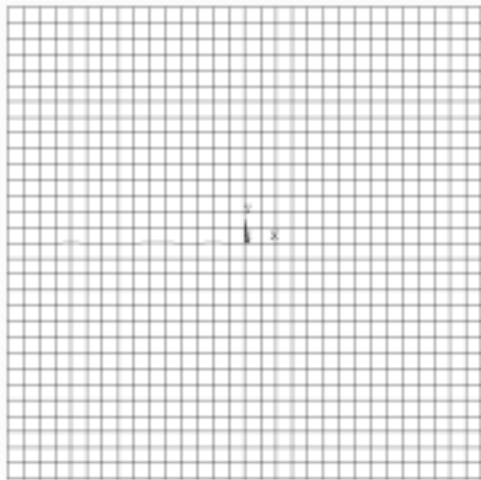
- [1] A. Malanchara and W. Gerstle, "Comparative study of unstructured meshes made of triangles and quadrilaterals," presented at 6 th International Meshing Roundtable., 1997.
- [2] K. Shimada, "Physically-Based Mesh Generation: Automated Triangulation of Surfaces and Volumes via Bubble Packing," in *Mechanical Engineering*. Cambridge, MA, U.S.A.: Massachusetts Institute of Technology, 1993.
- [3] K. Shimada and D. C. Gossard, "Bubble Mesh: Automated triangular meshing of non-manifold geometry by sphere packing," presented at Third Symposium on Solid Modeling and Applications, 1995.
- [4] K. Shimada, A. Yamada, and T. Itoh, "Anisotropic Triangulation of Parametric Surfaces via Close Packing of Ellipsoids," *To appear in International Journal on Computational Geometry and Applications*.
- [5] K. Shimada, J.-H. Liao, and T. Itoh, "Quadrilateral Meshing with Directionality Control through the Packing of Square Cells," presented at 7 th International Meshing Roundtable, 1998.
- [6] K. Ho-Le, "Finite element mesh generation method : A review and classification," *Computer Aided Design*, vol. 20(1), pp. 27-38, 1988.
- [7] N. Sapidis and R. Perucchio, "Advanced techniques for automatic finite element meshing from solid models," *Computer Aided Design*, vol. 8(4), pp. 248-253, 1989.
- [8] M. S. Shephard and e. al., "Trends in automatic three-dimensional mesh generation," *Computers and Structures*, vol. 30(1/2), pp. 421-429, 1988.
- [9] W. C. Thacker, "A brief review of techniques for generating irregular computational grids," *International Journal for Numerical Methods in Engineering*, vol. 15, pp. 1335-1341, 1980.
- [10] S. Owen, "A Survey of Unstructured Mesh Generation Technology," <http://www.andrew.cmu.edu/user/sowen/survey/index.html>.
- [11] A. Hines, "Comparison of finite element meshing software packages," 1999, <http://www.andrew.cmu.edu/~sowen/hines99.html>.
- [12] R. Schneiders, "List of meshing software," <http://www-users.informatik.rwth-aachen.de/~roberts/software.html>.
- [13] E. A. Heighway, "A mesh generator for automatically subdividing irregular polygons into quadrilaterals," *IEEE transactions on Magnetics, Mag-19*, 1983.
- [14] B. P. Johnston, J. M. S. Jr., and A. Kwasnik, "Automatic conversion of triangular finite element meshes to quadrilateral elements," *International Journal for Numerical Methods in Engineering*, vol. 31, 1991.
- [15] K. Shimada, A. Yamada, and T. Itoh, "Automated conversion of 2d triangular meshes into quadrilateral meshes," presented at International Conference on Computational Engineering Science, 1995.

This method is naturally extensible to three dimensions by packing parallelepipeds instead of rectangles to generate anisotropic hexahedral meshes.

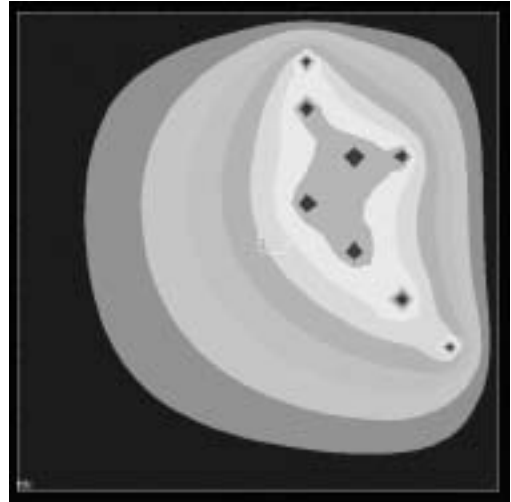
Since a large number of inputs are required to use the full capabilities of an anisotropic quadrilateral mesh generator, an adaptation scheme is desirable which can automatically produce the input, in this case – size, aspect ratio, and mesh directions. It is difficult to specify this input in complicated problems without prior knowledge. So a preliminary analysis would present a picture of the loads, material properties, and boundary conditions. Even though, it is possible to miss fine features, when using a coarse mesh, even this information, when used for mesh generation, is much better than none at all. When the mesh better reflects the problem being solved, computational cost can be lowered for a desired accuracy, or equivalently for the same cost, a better solution can be obtained. The design process is streamlined and cycle times are reduced.

Qualitative methods are used to ascertain how good a mesh is for a specific application. A quantitative approach will give a definite indication of how much one mesh is better than another. This is an important issue that should be addressed later.

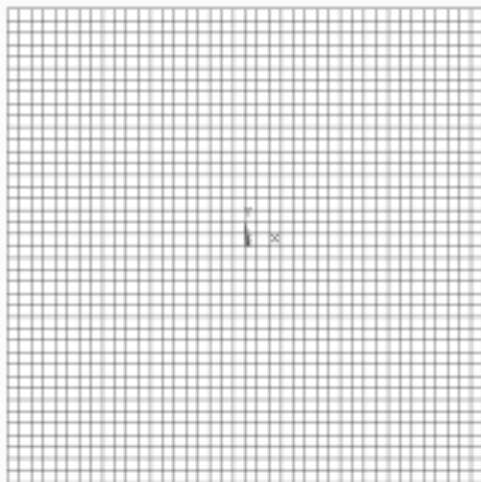
- [16] T. Itoh and K. Shimada, "Automatic conversion of triangular meshes into quadrilateral meshes with directionality," *submitted to International journal for numerical methods in engineering*, 2000.
- [17] Y. T. Lee, "Automatic finite-element mesh generation," *ACM Transactions on Graphics*, vol. 3, pp. 287-311, 1984.
- [18] Y. T. Lee, "Automatic Finite Element Mesh Generation based on Constructive Solid Geometry," . Leeds, England: University of Leeds, 1983.
- [19] T. D. Blacker and M. B. Stephenson, "Paving: A new approach to automated quadrilateral mesh generation," *International Journal for Numerical Methods in Engineering*, vol. 32, pp. 811-847, 1991.
- [20] S. J. Owen, M. L. Staten, S. A. Canann, and S. Saigal, "Q-Morph: An Indirect Approach to Advancing Front Quad Meshing," *International Journal for Numerical Methods in Engineering*, vol. 9, pp. 1317-1340, 1999.
- [21] M. J. Castro-Diaz, F. Hecht, and B. Mohammadi, "New progress in anisotropic grid adaptation for inviscid and viscous flow simulations," presented at 4th International Meshing Roundtable, 1995.
- [22] H. Borouchaki, P. J. Frey, and P. L. George, "Unstructured triangular-quadrilateral mesh generation application to surface meshing," presented at 5th International Meshing Roundtable, 1996.
- [23] F. J. Bossen and P. S. Heckbert, "A Pliant Method for Anisotropic Mesh Generation," presented at 5th International Meshing Roundtable, 1996.
- [24] X.-Y. Li, S.-H. Teng, and A. Ungor, "Biting Ellipses to Generate Anisotropic Mesh," presented at 8 th International Meshing Roundtable, 1999.
- [25] E. Kreyszig, *Advanced engineering mathematics*, 5 ed: New Age International (P) Limited., 1997.
- [26] D. A. Field, "Laplacian smoothing and delaunay triangulation," *Communications in Applied Numerical Methods*, vol. 4, pp. 709 - 719, 1988.
- [27] S. W. Sloan, "A fast algorithm for constructing Delaunay triangulations in plane," *Advanced Engineering Software*, vol. 9, pp. 34-42, 1987.



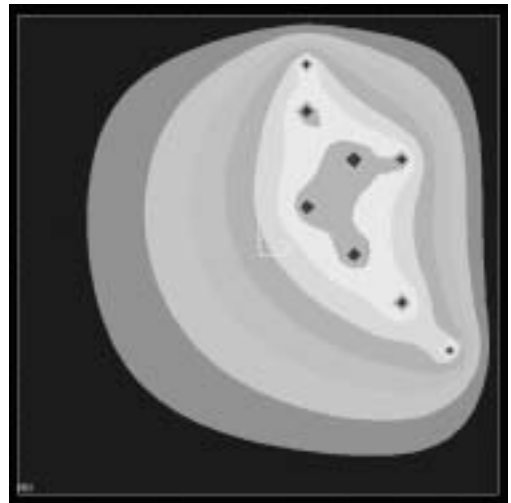
(a) Regular mesh Nodes 961, Elements 900



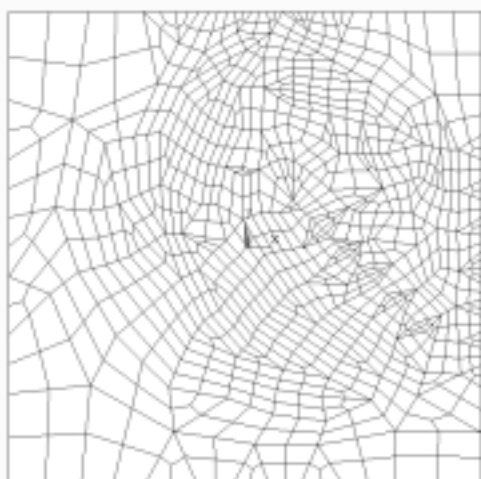
(b)



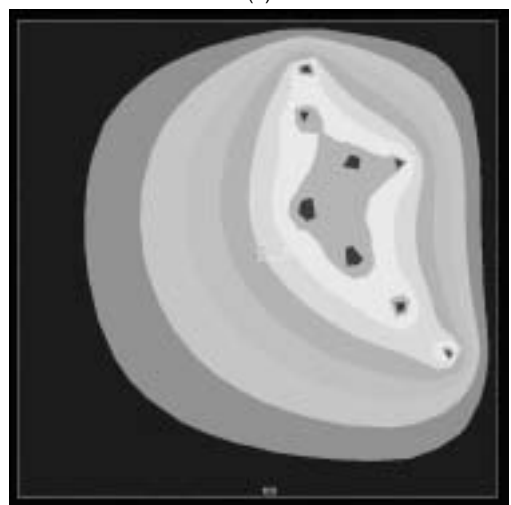
(c) Regular mesh Nodes 1681, Elements 1600



(d)



(e) Generated mesh Nodes 879, Elements 828



(f)

Figure 13 Meshes and temperature contours – Steady-state heat-transfer

