# International Journal of Applied Mathematics

Volume 29 No. 6 2016, 759-773

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)  $\mathbf{doi}$ : http://dx.doi.org/10.12732/ijam.v29i6.7

# QUALITATIVE ANALYSIS OF MESH REFINEMENT METHODS ON NON-CONVEX DOMAINS

Fernanda Paula Barbosa Pola<sup>1</sup>, Ives Renê Venturini Pola<sup>2</sup> §

1,2 Faculdade de Ciências e Tecnologia

UNESP - Universidade Estadual Paulista

Departamento de Matemática e Computação

Presidente Prudente - SP - BRAZIL

**Abstract:** As many numeric simulations are mesh based approaches, a domain decomposition into triangular elements is a essential process in order to produce a good representation of the target domain. But, many methods were proposed and the commonly used approaches have different behaviors depending of the domain to mesh. Convex domains are easier to form good triangulations by most methods because of the well formed borders, but non-convex domains may be a problem to represent correctly. This paper performs a qualitative analysis of two known triangulation methods, the Ruppert and Force Balance refinement methods. We present some results of mesh refinement for non-convex domains by testing the minimal restriction angle.

AMS Subject Classification: 65L50, 65M50, 65N50 Key Words: non-convex domain meshing, mesh refinement methods

#### 1. Introduction

Finding an appropriated domain discretization is essential to the quicker convergence of a numeric simulation. In this discretization, generally known as a triangulation, a series of equations describing physical laws must be solved. There was some proposed triangulation methods in literature, but most of them

Received: September 16, 2016

© 2016 Academic Publications

<sup>§</sup>Correspondence author

were build to work mainly on exact models, usually generated by CAD systems. In this situations the generated triangulation must be corresponded exactly to the border limits of the the original model, because specific details can not be lost in representation.

Starting on the 80's, mesh generation techniques were proposed such as Delaunay triangulation, Mesh Propagation and Quadtrees/Octrees. A Quadtree algorithm consists in recursively dividing a quadrilateral which contains the geometry until hitting the desired solution. The Mesh Propagation algorithm is based on the building of triangles at the border of the geometry and progressively heading to the surface center. But, the most popular of those techniques – the Delaunay Triangulation, consists on creating elements on a way that does not exist a vertex of a polyhedral that is inside of the circumscribed sphere of another polyhedral, [6].

After the mesh generation, many times it is necessary to refine them by including a set of well posed internal points. One of the first refinement techniques consisted on connecting the barycenter of the element to its vertexes building new elements. However, the generated mesh was not quality guaranteed. Another technique that produced satisfactory results was introduced by Hermeline [5], proposing to refine the imperfections by calculated points though weights.

Many other methods were proposed in order to reduce the mesh errors through refinements to guarantee good elements. Some of these techniques can be seen in works of Chew [1],[2], Frey [4], and Shewchuk [11]. Most surged techniques for the mesh generation in literature derived from the Delaunay agorithm [3], [8], [9], [10]. However, the Delaunay algorithm have the property to maximize the smaller angle of the triangles making it very popular among researchers. The purpose to raise the smaller element angle is to remove the thin triangles, increasing the mesh quality, because bad quality elements lead to a loss in results precision.

Tools were developed by applying the Delaunay algorithm, such as the *Triangle* as one of the most cited [10]. The employed algorithm is adapted from the works of Chew [3] and Ruppert [8, 9] and does not need a previous processing at the border. Points are added during the refinement process to preserve the domain accordance. The resulting mesh is usually consisted of few elements and triangles preserves quality.

This paper investigates the Delaunay triangulation with Ruppert refinement, by also applying the Force Balance method dealing with cases where initial polygons presents very acute angles. Therefore, we investigated the advantages and disadvantages of each method for non-convex domain cases.

The structure of this paper is the following. Section 2 presents the background concepts of the Delaunay algorithm; Section 3 presents the mesh refinement algorithms used in this work, including Ruppert and Force Balance method; Section 4 presents the numeric results achieved in experiments and Section 5 draws conclusions about the obtained results.

### 2. Delaunay Algorithm

The general idea of the Delaunay refinement algorithms is to modify an initial triangulation by adding new points to improve the mesh quality. Every triangle must keep its circumscribed circle (circumcircle) empty, i.e., no other points can be inside it. The quality of the generated triangle is evaluated by the ratio between the circumscribed circle radius r and the shortest edge l, as can be seen in Figure 1. The ratio r/l is directly associated to the smallest angle  $\theta$  of the triangle.

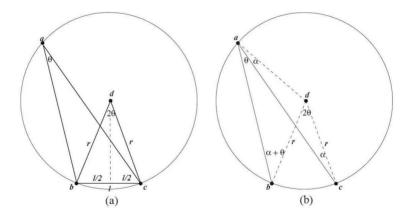


Figure 1: a) A triangle  $\Delta abc$  circumscribed circle centered at d, where  $\theta$  is the triangle smallest angle, l its smaller edge, and r the circle radius. (b) The two isosceles triangles  $\Delta adb$  e  $\Delta adc$ .

According to the Figure 1, we can derive that  $\angle bdc = 2\theta$ . Following, let  $\alpha = \angle cad$ , we have

$$\angle adb = 180^0 - 2(\theta + \alpha) \tag{1}$$

and

$$\angle adc = 180^0 - 2\alpha \tag{2}$$

as  $\triangle adb$  and  $\triangle adc$  are isosceles and subtracting (1) of (2), we have that  $\angle bdc = 2\theta$ . Note that  $sen\theta = \frac{l}{2r}$ , consequently, for  $r/l \leq \mathbf{L}$  the minimal angle  $\theta$  cannot be greater than  $arcsen\frac{1}{2\mathbf{L}}$ . A triangle with ratio greater than the limit  $\mathbf{L}$  is called of thin triangle.

The main operation in the Delaunay refinement is the insertion of a point at the triangle circumcenter in order to improve the mesh quality. In order to remove the thin triangles of the mesh, a new vertex is inserted at the circumcenter. After each operation the mesh edges must be remade. As the circumcircle of any triangle must remain empty in a Delaunay triangulation, by inserting the each new vertex the associated triangle is eliminated from the mesh. Moreover, no new edge will have smaller width than the circumradius. The process is repeated until no triangle have ratio greater than  ${\bf L}$ .

#### 3. Bidimensional Delaunay Refinement Algorithms

In this section are presented the Ruppert and the Force Balance algorithms. Some algorithms build the Delaunay triangulation with a posed restriction, thus not being limited to an entry composed of only vertexes. The restriction occurs when the entry also presents edges that must be preserved in the final mesh. Those entry data are known as planar straight line graph (PSLG).

# 3.1. The Ruppert Refinement Algorithm

The refinement algorithm of Jim Ruppert [9] is employed for the generation of bidimensional meshes with a satisfactory level of quality with an iterative method based on Delaunay triangulation concepts. A segment is called wrapped when the diameter calculus contains some vertex that its not one of the vertexes that define the main segment. The diameter circle is the smallest possible circle that contains the segment. Figure 2 shows an example of a wrapped segment s by vertex v.

In order to generate a triangulation regarding the limit L, the cases of wrapped segments and thin triangles must be processed. A wrapped segment is divided by adding a new vertex in its center point. A thin triangle is usually eliminated by adding a new vertex in its circumcenter. However, if the new vertex generate one or more wrapped segments, instead of add them, all those segments that would be wrapped are divided.

Ruppert has proven that for a limit  $\mathbf{L} = \sqrt{2}$ , the algorithm have the convergence guarantee. The limit  $\mathbf{L} \geq \sqrt{2}$  generate a triangulation with angles

between  $20.7^{0}$  and  $138.6^{0}$ . However, this guaranty is only valid for polygons with minimal angle of at least  $90^{0}$ . Shewchuk [11] has proven that this restriction can be reduced to  $60^{0}$ .

For angles smaller than  $60^0$  the algorithm usually not generate well formed triangles, specially near small angles from entry set. Moreover, Shewchuk [11] shows that in some cases, the refinement algorithms of Delaunay could not complete when there are small angles in polygons. Figure 3 illustrate one of the generated problems by small angles of the polygon, particularly when there is any less than  $45^0$ . The segment qr is wrapped by the vertex p. Then qr is divided by inserting a new vertex s in its center point. However, the segment qp will be wrapped by the new vertex s, and also must be divided. Following, the segment qs will be wrapped by the new vertex and also will be divided. This process can continue and the algorithm may not converge.

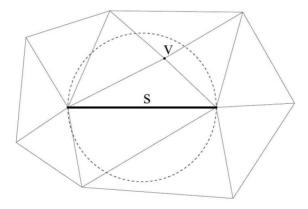


Figure 2: The segment s is wrapped by vertex v which is inside its diameter circle.

The Ruppert approach creates concentric circles around each vertex of the entry set. Each circle has the double of the radius of the most internal circle, where all radius are in power of 2. When dividing a segment, where one of the extremes is an entry vertex, the segment is divided in the intersection with the closer circle (not the center point as before). Figure 4 illustrates the Ruppert method. Eventually, the segments separated by a small angle will be divided with same sizes. As segments with equal sizes does not wrap each other, division loop does not happens. The algorithm 1 illustrates the steps necessary to do the refinement.

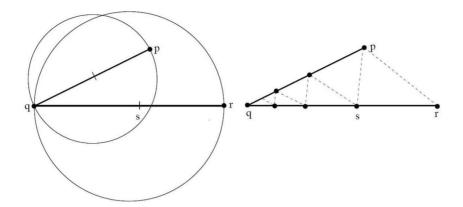


Figure 3: Illustration of a problem caused by small angles of a polygon. As any vertex inserted turns some segment wrapped, the algorithm keeps dividing edges.

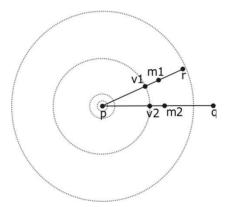


Figure 4: When the segment pr is subdivided, instead of use the mean point m1, the point v1 in the closer circle is used. Analogously, the segment pq is subdivided at point v2. The subdivided segments pv1 e pv2 have the same size, and then v1 does not wrap pv2 neither v2 on pv1.

#### 3.2. Force - Balance Method

An alternative method of adaptive mesh generation was proposed by Per-Olof Person et al. [7]. The domain to be triangulated is defined though an implicit continuum function f(x, y). Such function can be also combined to form more complex domains through union, intersection and subtraction operations. The

# Algorithm 1 Rupert refinement algorithm

```
Require: G \leftarrow PSLG
Require: \alpha: minimal quality
 1: Generate Delaunay triangulation T for vertexes in G
 2: while exists an encroached segment s \in G do
 3:
      Split s by inserting a midpoint
      Update G and T
 5: end while
 6: while exists some triangle t \in T with quality q where q_t < \alpha do
 7:
      Let p the circumcenter of t
      Identify segments s_1, s_2, \ldots, s_k encroached by p
 8:
      if k \ge 1 then
 9:
         Split segments s_1, s_2, \ldots, s_k inserting the k midpoints
10:
11:
         Update G and T
      else
12:
         Add p
13:
      end if
14:
15: end while
```

main concept of the method is the modeling of the mesh as a spring structure. The mesh vertexes are the connecting nodes while the edges are the springs connecting nodes. Each spring presents an associated force that provides either its stretching or shrinking. Considering a spring set, the attractive and repulsive forces sum into a resulting vector force in the structure and, in each iteration, the system is solved for the balance. The spring forces move nodes by distributing them in the domain.

Each vector has horizontal and vertical components for each mesh vertex, being constituted of internal forces from springs and external forces from border reactions, as can be seen in the equation below:

$$F(p) = [F_{int,x}(p), F_{int,y}(p)] + [F_{ext,x}(p), F_{ext,y}(p)].$$
(3)

The forces depend directly of the topology of springs structure. This topology is obtained through a Delaunay triangulation, which is generated based on the entry points. Note that the force vector F(P) is not a continuum function over p seen that the topology can be changed by the Delaunay triangulation according with the node points positions. The system F(p) = 0 must be solved for a set of a equilibrium position set p. A simple approach for the solution of this problem is to consider an artificial temporal element. For some  $p(0) = p_0$ 

we consider a differential equation system:

$$\frac{dp}{dt} = F(p), \quad t \ge 0. \tag{4}$$

If a balance situation is found, the solution for the differential equation 4 will be a solution for the system F(p) = 0. The system 4 can be approximated through the Euler method. At the time  $t_n = n\Delta t$  the approximated solution  $p_n \approx p(t_n)$  is updated through the following equation

$$p_{n+1} = p_n + \Delta t F(p_n). \tag{5}$$

During the evaluation of the force F(p), the points position  $p_n$  are known as the structure topology, due to the Delaunay triangulation of the initial points set. The external reaction forces are handled in the following manner. All points that are moved outside the domain during the positions update of  $p_n$  for  $p_{n+1}$  are re-positioned back to the closer border point. The points can be moved along the border but never outside it.

The force function is modeled in a manner to allow only repulsive forces, not allowing the attractive forces, where the constant k has a role on the unity correction. The force function is then given by

$$f(l, l_0) = \begin{cases} 0, & l \ge l_0 \\ k(l_0 - l), & l < l_0 \end{cases}$$
 (6)

The border treatment in this method aim at spreading the points along the domain. For that to happen, it is important that there are repulsive forces f > 0. This means that the function  $f(l, l_0)$  must be positive when l is close to the desired edges width, and it can be done by the choice of a width  $l_0$  a little higher than the desired. The edges width is given to the method through a width function h(x, y). The function h(x, y) does not represents the absolute width of an edge but its relative width, and in this way it can be avoided an implicit relationship between the number of nodes and the edges width. Due to the relative nature of the edges width through the function h(x, y), it is considered a scale value, evaluated as follows:

$$S = \sqrt{\frac{\sum l_i^2}{\sum h(x_i, y_i)^2}},\tag{7}$$

where  $l_i$  is part of the collection of all edges.

#### Algorithm 2 Force- Balance Method

- 1: Create initial distribution of points in the bounding box of domain
- 2: Remove points outside of domain
- 3: Perform an initial Delaunay triangulation
- 4: while moving points still achieves bad quality do
- 5: **if** there was considerable points movement **then**
- 6: Save old points positions
- 7: Perform Delaunay Triangulation of the points
- 8: Evaluate the triangle centroids
- 9: Remove triangles outside of the domain
- 10: end if
- 11: Evaluate the current edges width
- 12: Evaluate the desired widths
- 13: Evaluate the forces vector
- 14: Apply the forces and update the points positions.
- 15: Bring outside points to domain inside
- 16: end while

#### 4. Numeric Results

Basing on the presented methods, it is possible to establish qualitative comparisons between them, allowing to discover the advantages and disadvantages from any method to another. We will present in this section a qualitative comparison between the presented methods.



Figure 5: Acute angle domain.

First, a comparison was made based on two mesh files, allowing to observe the precision and quality of each method. Figure 5 presents the domain of the first mesh with an acute angle in one of its borders. Both methods generated the mesh correctly, but the Ruppert method took around two seconds for 1507 points, once the force balance method took a greater time in the order of minutes, due

to the need of a new Delaunay mesh each time the points move. Therefore, more than half of the time spent for the force balance method was due to the successive Delaunay triangulations, performing 3646 at total with 1461 points. The result meshes generated by both methods can be seen in Figures 6(a) and 6(b).

The quality of the generated triangles was evaluated by verifying the angles in triangles. Therefore, a histogram was evaluated with the distribution of angles in each resulting mesh, as can be seen in Figures 7 and 8. The mesh generated by the Ruppert method present angles that vary from 25 to 119 degrees. We can observe that the major concentration in the histogram is between 50 and 60 degrees, followed by smaller concentrations between 40 50 degrees and between 60 and 70 degrees. These results show that the mesh present a good quality relationed to angles. As higher the concentrations of angles next to 60 degrees, the better is the shape of the triangles. In this case, the Ruppert method was executed with an additional parameter guaranteeing the mesh quality with a minimal angle of 30 degrees. The mesh generated by the force balance method has minimal angle of 32 degrees and maximal angle of 105 degrees, being those values in a smaller interval when compared with the angles produced by the Ruppert method, showing a better quality of the mesh in general. The concentration of angles in the interval of 50 and 60 degrees is considerably higher than in the Ruppert results.

Based on these results, it can be concluded that the force balance method, besides more computationally expensive, generated better results when considering non convex domains and had no need of a special treatment for acute angles in the domain when compared to the Ruppert method. It is important to consider that the mesh contains similar number of elements, 2838 triangles for the Ruppert mesh and 2794 triangles for the force balance mesh.

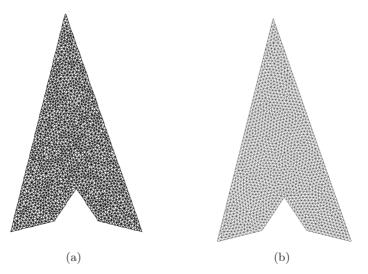


Figure 6: a) Ruppert generated mesh, b) Force balance generated mesh.

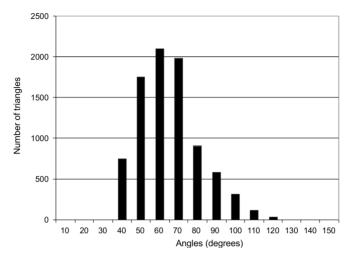


Figure 7: Angles histogram distribution for the mesh generated by the Ruppert method.

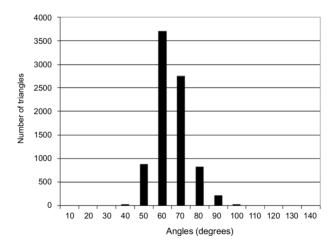


Figure 8: Angles histogram distribution for the mesh generated by the Force Balance method.

In the second experiment, a domain that is defined by four acute corners, as can be seen in Figure 9. The generated mesh by Ruppert and Force Balance methods can be seen in Figures 10(a) and 10(b), respectively.

The Ruppert method generated the mesh in a time around two seconds for 1050 points. The Force balance method spent 75 seconds to generate the mesh, with the Delaunay reconstruction consuming half of the total spent time, with 600 iterations for 1453 points. The generated Ruppert mesh has a minimal angle around 22 degrees and a maximal angle of 116 degrees. In this experiment, it was also used

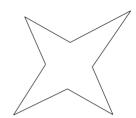


Figure 9: Mesh Domain 2.

a parameter guaranteeing a minimal angle of 30 degrees. The Force Balance generated mesh has minimal and maximal angles of 28 and 112 degrees, respectively. It can be seen again that the values for angles are in an interval smaller than those in Ruppert results for non convex domains.

Figures 11 and 12 show the histogram of the angle distribution. It can be seen that the major concentration of angles are between 50 and 60 degrees, followed by smaller angles between 40-50 degrees and 60-70 degrees, revealing the good mesh quality. Those angles are 26.3% of all angles in Force Balance method, while they are 23.6% in Ruppert's. On the other hand, in this iteration the Force Balance method had a higher number of triangles with angles between 20 and 30 degrees when comparing with the Ruppert method. However, the generated mesh by the Force Balance method has a smaller minimal angle and a higher maximal angle, when compared with Ruppert method.

It can be concluded that the generated mesh by Force Balance method is slightly superior in terms of quality with relation to the generated mesh by the Ruppert method. This is due to the smaller interval between the smaller and the greater angle degree, which reflects in the shape of the triangles generated. The spent time is higher when comparing to Ruppert, demanding more computational power.

#### 5. Conclusion

In this paper we performed a qualitative analysis of two known triangulation methods, the Ruppert and Force Balance refinement methods, specifically on mesh refinement for non-convex domains and testing the minimal restriction angle.

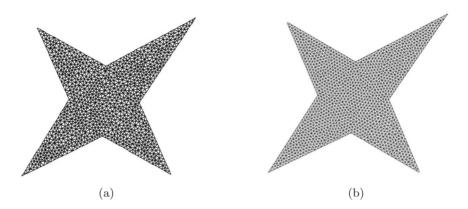


Figure 10: a) Mesh generated by Ruppert method, b) Mesh generated by Force Balance method. Case 2.

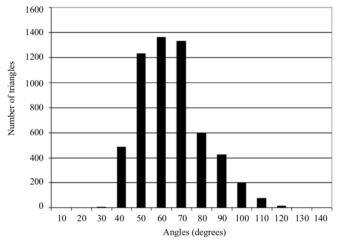


Figure 11: Angles histogram distribution for the mesh generated by the Ruppert method. Case 2.

The Ruppert method provides an alternative to the problem of applications where there is a need of refinement in specific parts of the domain. This alternative has as base the creation of a mesh less uniform and with variable density. In the boundaries the elements are smaller, following the typical irregularity wanted. This type of behavior can offer advantages with relation to numeric simulations that are focused in the behavior next to the borders. However, the internal parts of the domain or portions that are close to the border will have a refinement with greater elements, then reducing the precision in those regions.

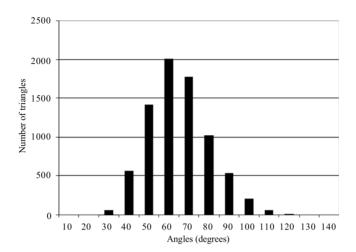


Figure 12: Angles histogram distribution for the mesh generated by the Force Balance method. Case 2.

The Force Balance method is a interesting approach in this case because uses as a basis a physical analogy with a spring balance system, where the springs are the edges in the mesh connecting vertexes. Although having a higher computational complexity, the Force Balance method produced better quality meshes when dealing with non convex domains.

#### References

- [1] L. P. Chew, Constrained delaunay triangulation, Algorithmica, 4 (1987), 97–108.
- [2] L. P. Chew, Guaranteed-quality triangular meshes, *Cornell University*, 4 (1989), 89–983, Technical Report.
- [3] L. P. Chew, Guaranteed-quality mesh generation for curved surfaces, *Proc.* of the Ninth Annual Symposium on Computational Geometry (1993), 274–280.
- [4] W. H. Frey, A new strategy for automatic node placement in graded triangular meshes, *International J. of Numerical Methods in Engineering*, **24** (1987), 2183–2200.
- [5] F. Hermeline, Triangulation d'un poliedre en dimension n.r.a.i.r.o., Analyse Numérique, Numerical Analysis, 16, No 3 (1982), 211–242.

- [6] S. J. Owen, A survey of unstructured mesh generation technology, Proc. 7<sup>th</sup> International Meshing Roundtable in Applied Computational Geometry, 22, No 1-3 (1998), 239–267.
- [7] P. O. Persson and G. Strang, A simple mesh generator in Matlab, SIAM Review, 46 (2004), 329–345.
- [8] J. A. Ruppert, A new and sample algorithm for quality 2-dimensional mesh generation, *Int.* 4<sup>th</sup> ACM-SIAM Symposium on Discrete Algorithms (1993), 83–92.
- [9] J. A. Ruppert, A Delaunay refinement algorithm for quality 2-dimensional mesh generation, *Journal of Algorithm*, 18, No 3 (1995), 548–585.
- [10] J. R. Shewchuk, Triangle: Engineering a 2d quality mesh generator and Delaunay triangulator. Proc. 1<sup>st</sup> Workshop in Applied Computational Geometry (1996), 123–133.
- [11] J. R. Shewchuk, *Delaunay Refinement Mesh Generation*, PhD Thesis, School of Computer Science, Carnegie Mellon University, Pittsburgh (1997).