Mesh Generation

1. Grid generation

A good grid generation methodology must generate well shaped grids on domain of any level of complexity. The skewness of the grids must be low. It should also have an adaptive nature to the boundaries. It should be easily programmable.

When every point in a group of points is a node of any triangle of group of non-intersecting triangles then such a group of triangles is called the triangulation of the given set of points. There can be many ways of triangulating a given set of points.

If circum-circle of any triangle does not enclose any other node in the domain, then such a triangulation is called the Delaunay triangulation of the given set of points. This property ensures optimum shapes of the elements generated inside the domain. This is also called the "in-circle" property or criteria to identify a triangulation as Delaunay.

Rupert's Algorithm (Wikipedia) is one such grid generation methodology available. Bowyer 1981 and Watson 1981 have developed a procedure to generate a Delaunay triangulation from n points called Bowyer algorithm.

Bowyer algorithm

Our aim is to generate the Delaunay triangulation of the given set of points P_1 to P_n such that the given triangulation maps a convex domain. Instead of triangulating the points all at once we employ point by point methodology.

Let us say we have Delaunay triangulated points from P_1 to P_m. $4<m<n+1$ A new point P_{m+1} is considered. We then track the changes that need to be done to add P_{m+1} to the existing triangulation such that the new triangulation which includes the P_{m+1} also is Delaunay. As m goes from 5 to n we have Delaunay triangulated the given set of points.

The changes are done according to the bowyer algorithm.

1. If the newly added point lies inside the circum-circle of any triangle, then this triangle will be deleted.
2. A list of all faces common between deleted triangles and undeleted neighbors is obtained.
3. New triangles are created consisting of the newly added point and the points of the edges obtained in step 2.

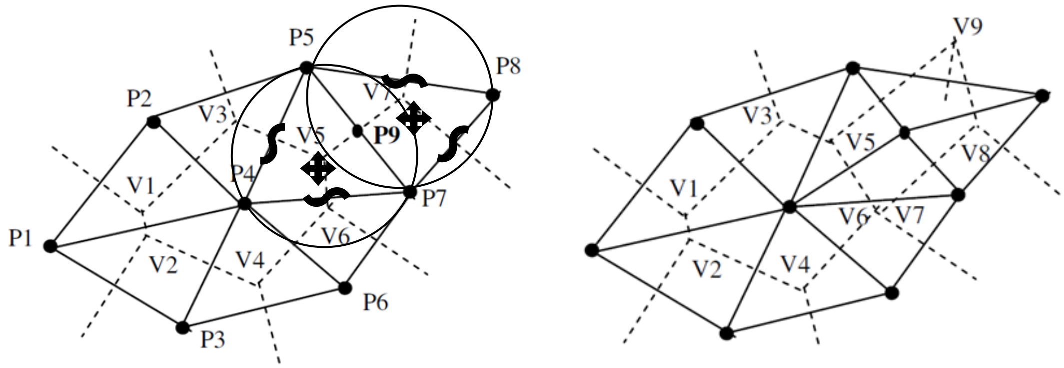

Figure A1.1 A point being inserted in an existing Delaunay triangulation.

In the Figure A1.1, we want generate the Delaunay triangulation of set of points P_1 to P_9 .We have Delaunay triangulated points from P_1 to P_8 .To do this we employ a point by point methodology. A point P_9 needs to be added some where inside an existing Delaunay triangulation such that the new triangulation also has Delaunay in-circle property. This is done by the Bayer algorithm given below. Point P_9 is added such that it lies is the circum-circle of triangles V_5 and V_7. In the above example triangles marked by '+' are deleted while edges marked by '~' satisfy step 2. New triangles are created consisting of the edges marked by '~' and P_9.

Note that if a triangle is to be **deleted** or created, only the data structure of the triangle is deleted or altered. The data structure of the points forming the triangle, remain intact and unaffected by deletion or alteration of the triangle.

Weatherill 1988 and 1992 developed a mesh generation methodology based on the above discussed concept of sequential addition of points in the domain. The method starts with only the boundary mesh and gradually progresses inside the domain by generating new nodes and adding them using the algorithm described above. The method can be easily extended to three dimensional domains. This method appears attractive as it minimizes the inputs from the user and ensures good mesh quality. Hence this method was used in the code developed for triangulation. The steps involved are listed below.

1. Initiating the Delaunay triangulation

We are given a set of points that define a boundary of a 2-D domain that have to be triangulated using the Delaunay procedure. The Bowyer algorithm needs some initial Delaunay triangles to start triangulation such that they cover the entire domain. Hence four temporary points are added to this set such that, a convex quadrilateral is formed from them and the set of boundary points lies inside this quadrilateral. Two Delaunay triangles are created from them by joining appropriate points. The Bowyer algorithm is initiated from these two triangles. The final mesh is unaffected by the initial points.

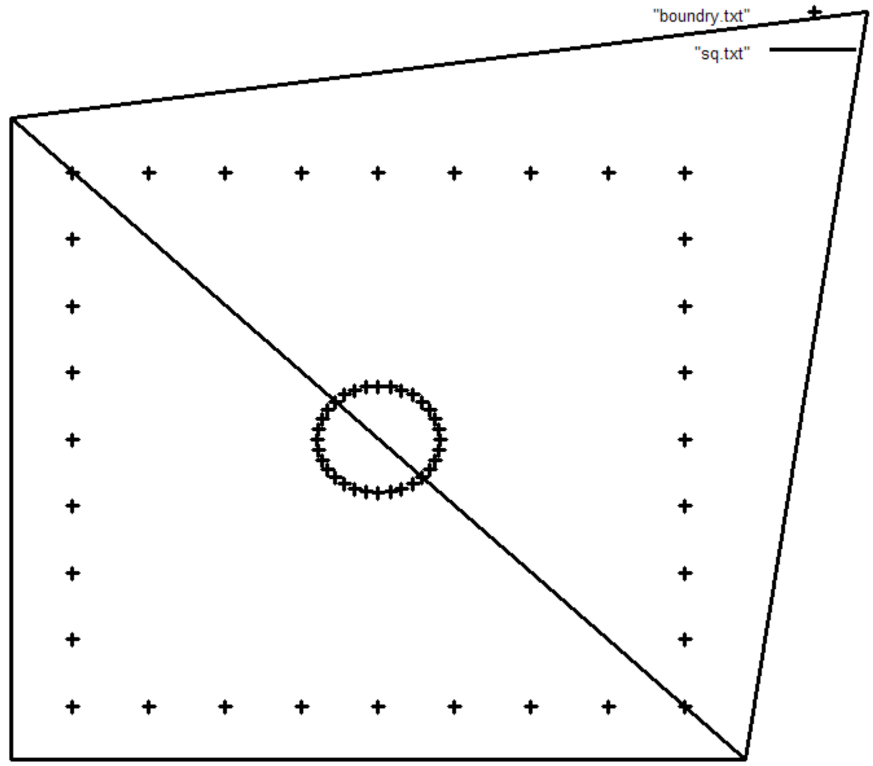

Figure A1.2 Convex quadrilateral around points to be triangulated

2. Generating the boundary mesh

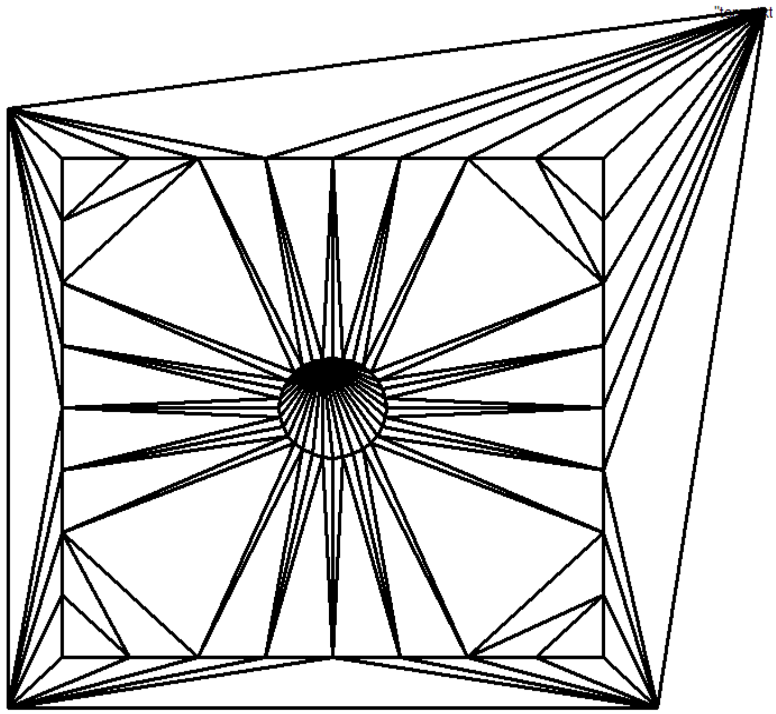

Figure A1.3 Boundary mesh of the triangulated points

The set of boundary points is fed one by one into the initiated Delaunay triangulation. The mesh is modified using the Bowyer algorithm described above. After this step is complete, triangulated mesh consisting of set of boundary points and the four additional points of the initial quadrilateral is obtained.

3. Cleaning process to delete triangles outside the actual domain

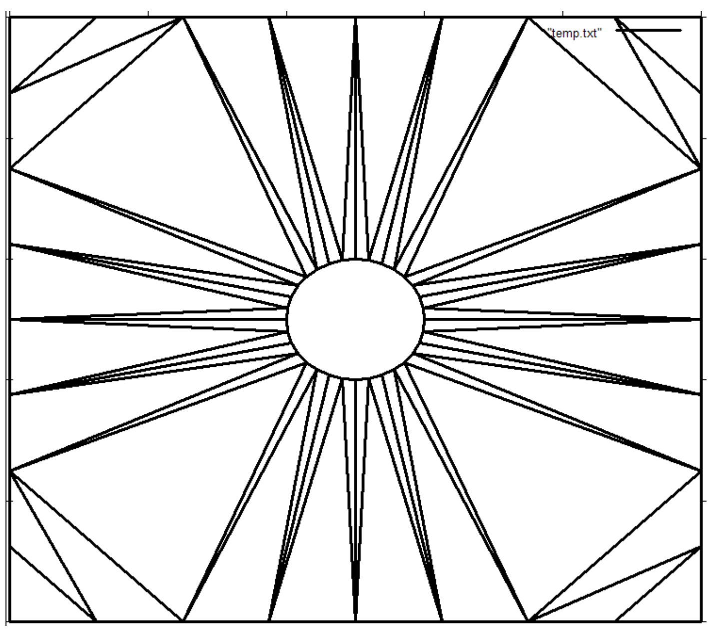

Figure A1.4 Cleaned up boundary mesh of the triangulated points

All those triangles which are outside the domain are deleted. After this step is complete, triangulated mesh consisting of only the boundary points is obtained. This serves as the initial mesh for automatic point insertion step. This triangulation has the Delaunay property because of the Bowyer algorithm.

4. **Automatic point insertion inside the domain**

This step is the most crucial step as it decides the location of the nodes inside the domain, and triangulates it.

Active triangles are those that generate a new set of prospective points at their centroids for next iteration. **Inactive triangles** are those that do not generate new points. For the first iteration all triangles of the initial mesh are initialized to be active.

Point Distribution Function (PDF) is defined for every node, which represents the local mesh scale. For every boundary node it is calculated as average of the lengths of the two adjacent edges sharing this node. PDF for the prospective centroidal point is obtained by taking average of the PDF at the nodes of the triangle. Following are the steps involved in the triangulation.

With this we begin the first iteration of the automatic point insertion algorithm.

1. New prospective points are generated at the centroids of the active triangles for the current iteration.
2. PDF for the new prospective points is obtained for the current iteration.
3. A prospective point must satisfy the following conditions.

- Distances of this centroidal point from the any node of the triangle that created it must be greater than 0.7 times the PDF at the node.
- Consider any other point (that include established points of the previous iteration as well as newly considered non-rejected prospective points) that has a PDF smaller than the prospective point being examined. The distance between these two points must be greater than 0.7 times the PDF of the prospective point

4. A set of prospective points passing step three is obtained. This set is inserted one by one into the existing Delaunay triangulation using the Bowyer algorithm for boundary triangulation.
5. Consider the average PDF of any two nodes of a triangle. If the average PDF is less than $2/3^{rd}$ the distance between them, then the triangle is made inactive. This step is done for all triangles.

The algorithm converges when either of the following situations arises.

- All the triangles remain inactive after iteration.
- All the prospective points get rejected by step 3 or 4.

Else the next iteration begins from step 1.Figure below shows meshes after number of iterations

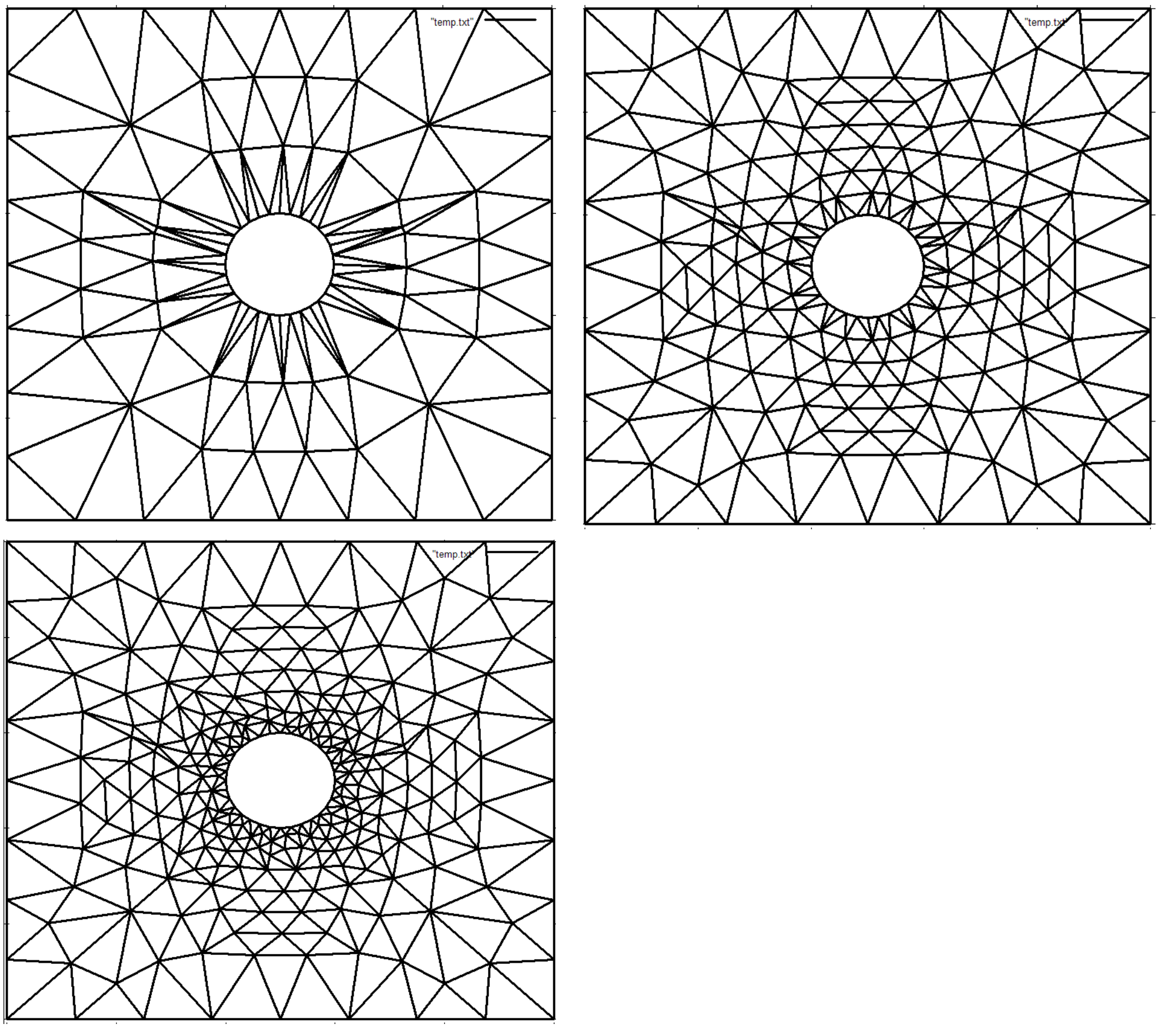

Figure A1.5 A Progressive Iterations of the Automatic point insertion method

5. Smoothing the mesh to improve mesh shape

This above process is followed by a smoothing process (Laplacian smoothing), where every point in the domain is slowly and iteratively moved to the centroid of its surrounding points using the following method. This improves the mesh quality by stretching the triangles to get them close to an equilateral shape.

$$x_i = x_i + \sum_{NB} (x_{NB} - x_i) \frac{\alpha}{N}$$

Where x_i is the coordinate of the point i α is an under-relaxation parameter, x_{NB} is a surrounding point coordinate, and N is the number of surrounding points. A similar equation appears for 'y' coordinate. These iterations are carried out till the difference in the distance moved by any point during the two consecutive smoothing iterations is less than 0.001 times the PDF at that point.

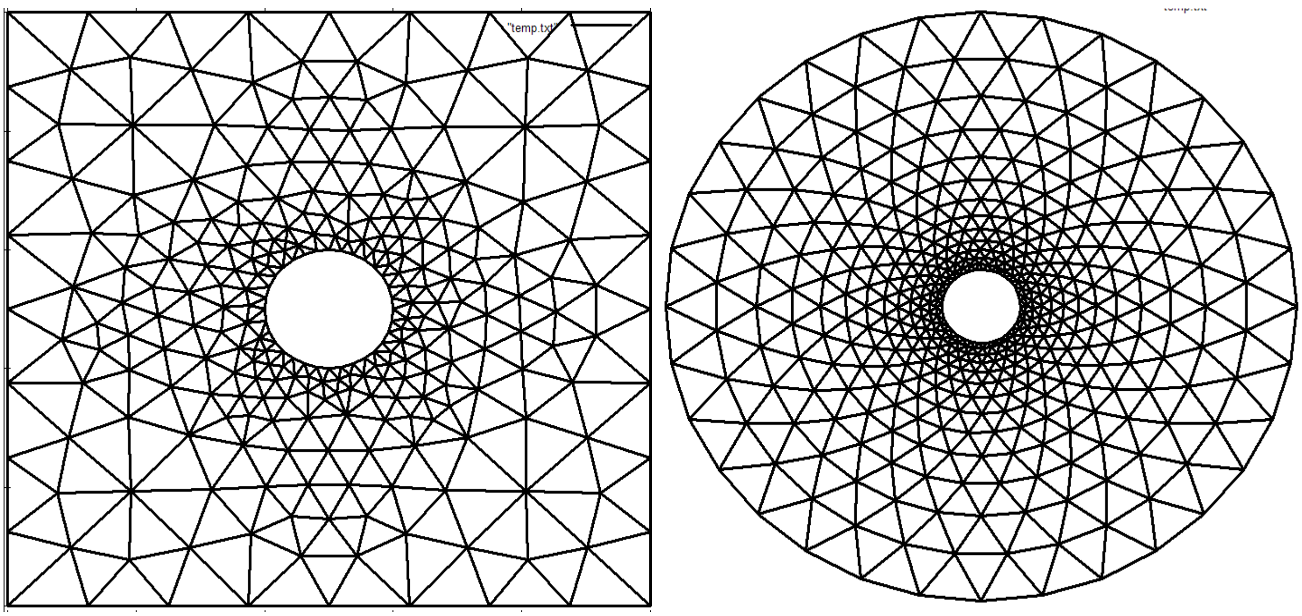

Figure A1.6 Final smoothened meshes