A 2D Combined Advancing Front-Delaunay Mesh Generation Scheme

A. El-Hamalawi

1. Introduction

The finite element method is a powerful and flexible numerical modelling tool in the engineering field. However, the mesh generation stage has always hampered its efficiency. One of the main reasons for seeking to automate mesh generation was the time and money spent in the process. This became especially crucial when the problems analysed had complicated irregular geometries or large gradient variations across the domain. In recent years, many advances have been made in an attempt to facilitate the mesh generation process, with the current literature being very extensive and an increasing degree of automation in mesh generators available.

The large variety of mesh generators, along with rapid changes in the field, have prompted many researchers to produce literature reviews of the available technology, such as Buell and Bush [Buell, Feb. 1973 #257], Shephard [Shephard, 1988 #329], Ho-Le [Ho-Le, 1988 #284], and more recently El-Hamalawi [El-Hamalawi, 1997 #206] and Frey and George [Frey, 2000 #369]. The mesh generation process may be classified into two broad classes; structured and unstructured mesh generators. During the initial phases of establishing the finite element method, when more emphasis was placed on the mathematical basis of the method rather than auxiliary processes, structured mesh generation was the norm. Unstructured mesh generators have now superseded structured techniques in most engineering applications, due to the flexibility and power they possess over structured generators when the object domain is complicated and irregular. Unstructured methods also facilitate the production of variable-sized elements from densities specified at random points in the mesh, which are the usual end-products of an error-estimation process in adaptive remeshing. This is in contrast to the more uniform element size variation across the mesh when based upon the densities on the edges of parent elements alone.

In adaptive mesh refinement, most analysts favour either the Delaunay technique or the advancing front method over other techniques when generating meshes due to the quality of unstructured meshes generated. The most popular method available for generating

triangular unstructured meshes is the Delaunay method [Delaunay, 1934 #372], due to it being the most efficient meshing technique available today in terms of speed. A Delaunay triangulation has the property that the circumcircle (or circumsphere in three dimensions) of every triangle (or tetrahedral) does not contain any other points of the triangulation. Several algorithms using the Delaunay property are available, Watson [Watson, 1981 #341] and Loze and Saunders [Loze, 1993 #306] algorithms being a favourite choice. Sibson [Sibson, 1973 #331] introduced the flipping algorithm for two-dimensional domains, where an arbitrary triangulation is converted into a Delaunay triangulation by flipping the diagonals of two neighbouring triangles. Chew [Chew, 1989 #266] proposed a constrained version of the Delaunay method for domains with irregular non-convex boundaries. Several other algorithms and variations of the Delaunay method exist.

The second popular method used to generate unstructured meshes is the advancing front method, first proposed by George [George, 1971 #370], with the more current form described by Lo [Lo, 1985 #245] and Peraire at al. [Peraire, 1987 #315]. Nodes are generated along the domain's boundary, which constitutes a front. Internal nodes are then generated to link sets of double nodes on the front, creating a layer of elements having roughly the same shape as the front. The internal nodes are positioned based on the element densities at these positions. The front is updated, with the new element edges forming the new front. This process is continued, working into the interior of the domain until the whole of the object domain has been meshed. Most of the mesh generators utilising advancing front techniques have been developed in the past few years. Peraire et al. [Peraire, 1987 #315], along with Zienkiewicz and Wu [Zienkiewicz, 1994 #355], managed to control the direction of mesh stretching in addition to the element size. More recent triangular mesh generation schemes using the advancing front method with modifications have been developed [Cescotto, 1989 #264; Jin, 1990 #287; Lo, 1994 #242; Qian, 1994 #320; Lau, 1996 #241; Peraire, 1997 #371].

The main advantage of the advancing front method is that they tend to produce nicely graded meshes and high quality triangles that are usually very close in shape to equilaterals. The boundary integrity is also preserved, since the discretisation of the domain boundary constitutes the initial front. This is in contrast to the Delaunay triangulation, where boundary integrity is not usually preserved for complicated domains, which is a key requirement for mesh generation procedures. Local mesh modifications can be applied to remedy the situation when meshed using the Delaunay method. Exterior elements are removed, and internal points

can be created and inserted into the current mesh. The resulting mesh is then optimised. The main disadvantage of the advancing front method lies with its efficiency, where processes such as checking the intersection of edges/overlapping of elements consume the majority of mesh generation time. Complicated domains may also pose some further problems such as the method of identifying and selecting optimal points based on a selected front, followed by validation of the newly created element. All these problems may affect the convergence of the advancing front method.

To overcome the various problems associated with each of the two methods described, an approach combining the Delaunay and advancing front techniques is used. Mavriplis [Mavriplis, 1992 #373] suggested an algorithm that utilises the Delaunay method to generate elements based on points located by the advancing front method. It allows the construction of several elements at a time, and the proper merging of two fronts of different element length densities, which would cause the classical advancing front method to fail. Recently Frey et al. ([Frey, 1996 #375], [Frey, 1998 #374]) developed a 3D approach based on the coupled Delaunay-advancing front method for tetrahedrals.

In this paper, a new unstructured triangular mesh generator will be described, which incorporates and combines both the Delaunay and advancing front methods. The domain boundary comprising the initial front is discretised by adding nodes based on pre-specified element densities. Internal optimal nodes are then computed from the front such that the latter are each located inside the circumscribed Delaunay circle, and an optimum equilateral triangle, or near equilateral, is created. The mesh is finally optimised by smoothing, and associated boundary conditions found by interpolation. Examples of generated meshes are described, with element quality factors presented and used to generate quadrilaterals for a mixed element mesh. A stand-alone mesh generator and its source code, *DBMesh*, based on this method, are available and can be obtained free of charge for non-commercial use by contacting the author.

2. Preview of the mesh generator

The mesh generation method implemented in this paper comprises seven stages, depicted in detail in Figure 1. The process starts with the input of the geometry and boundary conditions, along with the element densities h_i, which are defined in section 3.1. The domain

to be meshed is represented by $\Omega = \bigcup_i \Omega_i$ in the two-dimensional Euclidean space E^2, with boundary Γ, and sub-domains Ω_i, where $\Omega_i \subset \Omega$. The domain geometry is also input, with all internal boundaries Γ_{int} provided in a clockwise direction and external boundaries Γ_{out} provided in an anti-clockwise direction, where boundaries $\Gamma_{int} \cup \Gamma_{out} = \Gamma$. The boundary conditions such as fixities and loads are also input by the user, along with various other properties. Boundary nodes $P_b \in \Gamma$ are then generated along the internal and external domain boundaries, followed by the interior nodes P_{int}, which are created using a new method modified from the advancing front method. The triangulation algorithm used to form the elements is a hybrid of the two most popular methods; the advancing front and the Delaunay triangulation methods. Triangular quality factors [El-Hamalawi, 2000 #365] are calculated and the mesh smoothed accordingly. If quadrilaterals are required, the extra step of calculating quadrilateral factors and then merging triangles into quadrilaterals are taken, as will be described in section 5. The mixed mesh is then re-smoothed to ensure the elements are not too distorted. The final stages involve generating mesh boundary conditions by interpolation from the initial geometry ones, and producing the output files.

Figure 1. A flowchart showing the processes involved in the mixed element mesh generator

3 The node generation algorithm

3.1 Element size calculation at a point

Before any nodes and elements can be generated in a mesh, the element size h_i at any point i in the mesh has to be obtained. Several methods exist depending on which definition of h_i is used. The widely used definition of h for a triangle as the diameter of the smallest circle containing the element, as shown in Figure 2(a), is used here.

Figure 2. Definition of element size h

After some mathematical manipulation, h is found to be equal to 1.755 times the square root of the triangular area. No similar definition exists for a quadrilateral element, so h has been defined as equivalent to the length of a side of a square, which is the square root of the square's area. This can be thought of as being roughly equivalent to the square root of a quadrilateral's area. The definition of h is not of great importance as long as the same

definition is adhered to throughout, such as when transferring computed element sizes from an old mesh to the new one when adaptively remeshing, and during the mesh generation process.

A commonly used method in the literature to find h_i for an element i from adjacent elements is via equation (1), where r_{ij} is the distance between points i and j. Another modified form that could be used is one proposed by Cescotto and Wu [Cescotto, 1989 #264], defined by equation (2), where extra weights w_j are added. Equation (1) will be used with j ranging from 1 to 4. Using only the four closest points to node i was found by the author to provide better density values than the use of all the density points in the mesh, as some previous workers had done, since the local behaviour tends to be represented more strongly.

$$h_i = \frac{\sum_j \frac{h_j}{r_{ij}^2}}{\sum_j \frac{1}{r_{ij}^2}} \tag{1}$$

$$h_i = \frac{\sum_j \frac{h_j}{(1+\omega_j) r_{ij}^2}}{\sum_j \frac{1}{(1+\omega_j) r_{ij}^2}} \tag{2}$$

3.2 Boundary nodes generation

Various schemes exist for generating boundary nodes, however the method adopted in this paper is a modification of the one by Cescotto and Wu [Cescotto, 1989 #264]. Nodes are generated along the internal openings' boundaries in a clockwise direction, followed by external boundary nodes in an anticlockwise direction. No nodes are however generated along the connector lines, which are described in more detail in section 3.3. A typical segment AB obtained from splitting the boundary during the data input stage is shown in Figure 3(a).

Figure 3. Boundary node generation

The starting node (i.e. A or B) is chosen based on the lower element size h. The next nodal position s is then calculated using a corrector-predictor method in the form of equations (3). A predicted provisional position for node i+1 is computed using h_i, and then a corrected

position found. This is continued until s exceeds segment AB's length. However, the preferred element size will rarely be matched exactly. The nodal positions are thus finalised by distributing the discrepancy Δs amongst the various nodes according to their respective element sizes as shown in equation (4).

$$\begin{aligned} s_{i+1}^{\mathrm{p}} &= s_i + h_i \\ s_{i+1}^{\mathrm{c}} &= s_i + \frac{h_i + h_{i+1}^{\mathrm{p}}}{2} \end{aligned} \tag{3}$$

$$h_{ij}^{\mathrm{new}} = h_{ij}^{\mathrm{old}} \left(1 + \frac{\Delta s}{\mathrm{L}_{\mathrm{AB}}} \right) \tag{4}$$

The nodes are numbered according to the directions previously mentioned, and their co-ordinates are stored. This procedure is also applied to straight-lined as well as curved segments. One important point to notice is that the order of nodal numbers has to be reversed when dealing with nodes on a common boundary between different sub-domains.

3.3 Interior nodes generation

Before interior nodes are generated, *connector lines* are introduced, linking internal openings to the external boundaries, as shown in Figure 4. This forces the holes to become a continuous part of the external boundary, and incorporates them into the generation front. Sezer and Zeid [Sezer, 1991 #328] provide a method of generating connector lines, where the main objective is to connect an interior boundary node to an exterior one via the shortest distance.

In general, mesh quality is a function of the internal point distribution. This stage of the mesh generation process is thus very significant. The first step in this new method of generating interior nodes is to assign all the boundary segments of the sub-domain Ω_i to the generation front Γ_f. Segment AB is commenced with, and its midpoint M is calculated. A candidate node C is then proposed along the left perpendicular bisector of AB by assuming that an isosceles triangle ABC is to be formed, with base AB and height MC. The distance d_{MC} is obtained by calculating the element size h_{M} at M, and then using equation (5). The orientation of the point C with respect to segment AB (i.e. whether left or right) has to be checked using the procedure in Section A.1. Section A.2 describes how to check whether a proposed node is inside or outside a domain.

Figure 4. Addition of connector lines

$$d_{\mathrm{MC}} = \frac{\mathrm{AB}}{2\tan\left(\alpha/2\right)} \text{ where } \alpha = \frac{\cos^{-1}\left(1 - \frac{2\mathrm{AB}^2}{h_M^2}\right)}{2} \tag{5}$$

Figure 5. Addition of proposed node C

The feasibility of node C's position is checked by seeing whether edges AB and AC intersect any segments in the generation front Γ_f using the method in Section A.3. If they do intersect, then point C is discarded and the next segment is moved onto after removing segment AB from Γ_f . This method of finding an intersection is also applicable to curved edges, which are approximated by short straight-lined boundary segments. Assuming that this first check is complied with, the distances between all nodes $P_i \in P_{int} \cup P_b$ and C are computed and compared with the minimum element size specified by the user. This is to ensure that C is not too close to the boundary or other nodes, which would produce poor quality triangles. The distance between a point and a segment can be found using equation (A2) in section A.4. If the distance between a boundary node and C is less than the minimum spacing, node C is discarded, AB removed from Γ_f, and the next segment is moved onto. If however $P_i \in P_{int}$ and this check is not satisfied, the average position between C and P_i is found and a replacement node for the latter two is located at that position. In this case, the position of the node with respect to the sub-domain being triangulated is checked to ensure the node is inside. The edge AB is then removed from Γ_f , the new edges AC and AB added to the front, and C is then added to P_{int}, i.e. $\Gamma_f^{new} = \Gamma_f^{old} \cup \{\mathrm{AC}, \mathrm{CB}\}$ and $P_{int}^{new} = P_{int}^{old} \cup \{\mathrm{C}\}$. This procedure is repeated until no edges remain in the front Γ_f .

The splitting of the advancing front method into two phases in this paper is useful in order to avoid problems associated with the way that interior nodes and elements are generated. During the standard advancing front method, nodes and elements are generated simultaneously. The main problems encountered during such an approach include overlapping triangles upon the creation of a new node in small domain corners and connecting regions of elements with large variations in element densities, as shown in Figure 6. The first problem is avoided by moving the proposed node only, rather than the previously generated triangles. The second difficulty is solved by generating the interior nodes first, followed by the triangulation procedure.

Figure 6. Some common problems with the standard advancing front method

4 The Delaunay triangulation

A popular Delaunay triangulation approach in two dimensions is a version of an algorithm proposed by Watson [Watson, 1981 #341]. However, due to the Delaunay triangulation representing a convex hull, special consideration has to be paid when dealing with non-convex domains. This concern is easily addressed by rejecting elements that are not within the domain, which is especially important for domains with internal holes. The use of an advancing front from which elements must be on the left of the front, also facilitates this.

The triangulation algorithm used here is a modification of Lo's method [Lo, 1989 #304]. The domain boundary Γ is represented by a disjointed union of simple closed loops of segments. The generation front Γ_f is split into two sets; a Delaunay (Γ_d) and non-Delaunay front (Γ_{nd}), i.e. $\Gamma_f = \Gamma_d \cup \Gamma_{nd}$ and $\Gamma_d \cap \Gamma_{nd}=\varnothing$. Initially, Γ_f consists of the sub-domain boundary segments and $\Gamma_{nd} = \Gamma_f$ and $\Gamma_d=\varnothing$. Γ_d is the set of all generation front segments that are part of triangles satisfying the Delaunay property, while Γ_{nd} contains the remainder of the generation front segments.

The method constitutes examining each segment $AB \in \Gamma_{nd}$, and trying to select a node $C \in P_{in} \cup P_b$ such that triangle ABC's circumcircle is the smallest and ABC does not intersect the generation front. When such a node is found, the triangle ABC's Delaunay property is checked using the method in section A.5 and the new triangular edges AC and CB are classified accordingly. If triangle ABC is non-Delaunay, i.e. some point $P \in P_{in} \cup P_b$ exists inside the triangle's circumcircle, then the non-Delaunay set becomes $\Gamma_{nd}^{new} = \Gamma_{nd}^{old} \cup \{AC, CB\}$, otherwise $\Gamma_d^{new} = \Gamma_d^{old} \cup \{AC, CB\}$. Section A.6 describes how to determine the position of a point with respect to a triangle circumcircle. Segment AB is then removed from the set Γ_{nd} and the element connectivities and nodal arrays are updated. When all the non-Delaunay segments have been used, i.e. $\Gamma_{nd}=\varnothing$, the segments in Γ_d are treated in a similar manner until the generation front $\Gamma_f =\varnothing$. This process is repeated for all the sub-domains in the problem.

5 Converting triangles to quadrilaterals

Quadrilateral elements are desirable in mechanical-type finite element analyses, as they result in increased computational performance and numerical accuracy [Strang, 1973 #333]. The ability to produce quadrilaterals based on the triangles generated by the new meshing scheme has therefore been implemented into the mesh generator *DBMesh*, and is described below.

Figure 7. Two methods of creating quadrilaterals from triangular elements

Figure 7(a) shows the easiest way of producing a quadrilateral from two adjacent triangles. An element quality factor is computed for each triangle in the mesh and the neighbouring triangles, and the triangles combined to form quadrilaterals based on the optimum element quality factor. Another method is shown in figure 7(b), where a centroidal node of a patch of triangles is introduced and new quadrilateral edges (shown as dotted lines) added to form three quadrilaterals from the original four triangles. The process of transforming triangles into quadrilaterals means that for certain combinations, quadrilaterals of inferior quality might form. The use of shape quality measures is therefore a necessary step in order to ensure that the elements' shapes do not act as an extra cause of deterioration in finite element analysis accuracy. Several quality factors exist in the literature (e.g. [Frey, 2000 #369], [Lo, 1985 #245], [Robinson, 1987 #247; Robinson, 1988 #246; Lo, 1989 #244; Potyondy, 1995 #316]). The quality factors used here will be ones devised by the author [El-Hamalawi, 2000 #365], which are easier to visualise and which relate directly to the element interior angles. A brief description of this factor follows.

The optimum shapes for quadrilaterals and triangles are squares (interior angles of 90°) and equilaterals (interior angles of 60°) respectively. The main objective would therefore be to minimise the deviation $\delta\theta_i$ (equation (6)) for all interior angles θ_i. Shape factors $\left\|\vec{f}_Q\right\|$ and $\left\|\vec{f}_T\right\|$, defined by equations (7) and (8), are therefore proposed as quality measures for quadrilaterals and triangles respectively.

$$\delta\theta_Q = \left|\frac{\pi}{2} - \theta_i\right| \text{ for quadrilaterals, } \delta\theta_T = \left|\frac{\pi}{3} - \theta_i\right| \text{ for triangles} \tag{6}$$

$$\left\|\vec{f}_Q\right\| = \sqrt{\sum_{i=1}^{4} (\delta\theta_i)^2} \tag{7}$$

$$\left\|\vec{f}_T\right\| = \sqrt{\sum_{i=1}^{3}(\delta\theta_i)^2} \tag{8}$$

It can be seen that $\left\|\vec{f}_Q\right\|$ would attain a minimum value of zero for a perfect square and the acceptable range of 90°±45° defined by Zhu et al. [Zhu, 1991 #240] would correspond to $(\delta\theta_i)_{max}$ equal to $\pi/4$ radians, or $\left\|\vec{f}_Q\right\| \leq \pi/2$. Similarly, $\left\|\vec{f}_T\right\|$'s minimum value is zero for a perfect equilateral, and an arbitrary value of 60°±30° for triangles would lead to $(\delta\theta_i)_{max}$ equal to $\pi/6$, or $\left\|\vec{f}_T\right\| \leq \pi/\sqrt{12}$. The factors $\left\|\vec{f}_Q\right\|$ and $\left\|\vec{f}_T\right\|$ have been used in section 6 for various meshes.

After a mesh is produced, the shapes of the elements are improved using what is known as a smoothing process. Sezer and Zeid [Sezer, 1991 #328] developed the centroid smoothing method, which is the method used in this work with slight modifications, due to its suitability with mixed as well as pure element meshes. The formula used to calculate the interior node's new position P_i is given in equation (9), where A_j is the area of element j, P_{cj} is element j's centroid, and P_c is the centroid of the polygon, and in turn the new position of node i. The area A_j is calculated using equation (10) and the centroid using the general convex polygon formula (11) with n vertices in anticlockwise direction.

$$P_i = P_c = \frac{\sum_{j=1}^{N_i} P_{cj} A_j}{\sum_{j=1}^{N_i} A_j} \tag{9}$$

$$A_j = \frac{1}{2}\sum_{k=1}^{n}(x_k - x_{k+1})(y_k + y_{k+1}) \ , \ x_{n+1} = x_1 \ , \ y_{n+1} = y_1 \tag{10}$$

$$x_{cj} = \frac{\sum_{k=1}^{n}[x_{k+1} - x_k][y_k(y_k + y_{k+1}) + y_{k+1}^2]}{6A_j} \tag{11a}$$

$$y_{cj} = \frac{-\sum_{k=1}^{n}[y_{k+1} - y_k][x_k(x_k + x_{k+1}) + x_{k+1}^2]}{6A_j} \tag{11b}$$

The process is repeated for the rest of the nodes in a similar manner until the difference between the old and new nodal positions is small. From the analyses done in this paper, it has been found that although the number of iterations is dependent on the mesh, three to four iterations of smoothing were sufficient. For hybrid meshes, the extra step of

smoothing twice was taken; after triangulation had occurred, and then after the conversion of triangles to quadrilaterals. This was to ensure that the elements produced had improved aspect ratios, and having well-shaped triangles aids the creation of well-shaped quadrilaterals during conversion.

In some cases, a pure quadrilateral mesh may be required. In general, the existence of a mesh composed of only quadrilaterals is not guaranteed for an arbitrarily shaped domain. A solution resulting in a purely quadrilateral mesh based on a triangular mesh can be formed via subdividing the remaining triangles into three by introducing centroids and edge midpoints. The resulting mesh will comprise pure quadrilaterals and will also be a conforming mesh. However the quality of some of the quadrilaterals will be inferior, especially boundary elements, and it is advisable to use a pure quadrilateral mesh scheme if that is required.

6 Examples of meshes generated

In this section, sample example meshes generated using the techniques described are presented. The quality of the elements generated for each mesh are also assessed using the quality factors previously defined. Sudden changes in the element sizes h_i have been included in the examples by specifying them manually to check the method's ability to handle such situations, which are common-place in adaptive mesh refinement. The examples provided here thus serve to act as controlled tests, which rigorously test the mesh generator's capabilities. In doing so, they satisfy the main objective of ensuring that the elements produced are compliant with any pre-specified element sizes and are simultaneously of high quality.

Figure 8. Domain with internal openings and narrow edges

The geometry in Figure 8 has been designed to test the algorithm's performance when handling narrow edges and to ensure that the produced elements do not intersect. The geometry of the problem could easily be a machine part with two holes of different shapes. Initially, a purely triangular mesh was produced, shown in Figure 9, which consisted of 1221 elements. Figures 10 and 11 show the mesh after one and two smoothing iterations respectively. It can be seen that the mesh produced is of a better quality after smoothing than before. The maximum and minimum triangular quality factors $\left\|\vec{f}_T\right\|$ in the final mesh were found to be equal to 1.013 and 0.004, corresponding to angle ranges of 60°±33.5° and

60°±0.1° respectively. These are excellent angle ranges, with the overall mean mesh quality factor 0.376 corresponding to an angle range of 60°±12.4°.

Figure 12 depicts the same mesh as Figure 11 after the mesh was passed through the TRI2QUAD subroutine. This subroutine computes the quadrilateral quality factors $\left\|\vec{f}_Q\right\|$ resulting from testing different combinations of triangles and converts the latter to quadrilateral elements if a pre-specified quadrilateral factor limit is not exceeded. The conversion techniques used are the ones described in section 5. A maximum quadrilateral interior angle range of 90°±50° was used to generate the mesh in Figure 12, and as can be seen in the enlarged section in Figure 13, the quality of most of the quadrilaterals is good, with only a few being poor. These few would only be problematic if these elements were located in regions of high stress gradients where the variables calculated are of interest to the analyst. This could be remedied by using a lower pre-specified quadrilateral factor limit.

	Triangles	**Quadrilaterals (90°±50°)**	**Quadrilaterals (90°±45°)**
$\left\|\vec{f}_T\right\|_{min}$ **(angle range)**	0.004 (60°±0.1°)	0.004 (60°±0.1°)	0.004 (60°±0.1°)
$\left\|\vec{f}_T\right\|_{max}$ **(angle range)**	1.013 (60°±33.5°)	1.013 (60°±33.5°)	1.201 (60°±39.7°)
$\left\|\vec{f}_Q\right\|_{min}$ **(angle range)**		1.523 (90°±43.6°)	0.815 (90°±23.3°)
$\left\|\vec{f}_Q\right\|_{max}$ **(angle range)**		1.620 (90°±46.4°)	1.499 (90°±42.9°)
$\left\|\vec{f}_T\right\|_{mean}$ **(angle range)**	0.376 (60°±12.4°)	0.479 (90°±15.8°)	0.412 (90°±13.6°)
$\left\|\vec{f}_Q\right\|_{mean}$ **(angle range)**		1.552 (90°±44.4°)	1.408 (90°±40.3°)

Table 1. Quality factors for the problem in Figure 7

Figure 9. Mesh of domain in Figure 8

Figure 10. Mesh from Figure 8 after one smoothing iteration

Figure 11. Final mesh from Figure 8 after two smoothing iterations

Figure 12. Hybrid mesh from Figure 8 with maximum interior angles of 90°±50°

Figure 13. A zoom view of the upper part of Figure 12

Figure 14. Hybrid mesh from Figure 8 with maximum interior angles of 90°±45°

A lower angle range of 90°±45° was used to produce the mesh depicted in Figure 14. Figure 15 demonstrates how the quality of the quadrilaterals has improved by using a lower limit on $\left\|\vec{f}_Q\right\|$, with the quality factors shown in Table 1. One point of significance is that some of the triangular elements have had their quality factors increased; this is due to applying the smoothing procedure after quadrilaterals are produced, but they are still below the prescribed maximum $\left\|\vec{f}_T\right\|_{max}$. However the mean mesh triangular quality factor is still relatively low. The number of triangles and quadrilaterals in Figure 12 were 567 and 327 respectively, with 1107 and 57 in Figure 14. It can also be seen from the previous figures how high quality triangular and quadrilateral elements which satisfy the angle ranges specified by Zhu et al. [Zhu, 1991 #240] have been generated in the narrow edges and around the non-convex edges at the internal holes. These are usually problematic when using conventional mesh generation methods.

The domains shown in Figures 16(a) and 17(a) represent typical practical problems with different materials and openings such as underground tunnels, and have been meshed into Figures 16(b) and 17(b) respectively. In Figure 16(b), materials A and C have been meshed as mixed element materials, with material B having only triangles. It can be seen that the transition of elements between the different materials is gradual even though the changes in densities are more sudden. The use of several non-convex edges has also not had any marked effect on the quality of elements generated.

Figure 15. A zoom view of the upper part of Figure 14

Figure 17(b) has both materials meshed as hybrid element areas. A low $\left\|\vec{f}_Q\right\|_{max}$ limit has been imposed, which explains the small number of quadrilaterals generated. However,

the quality of quadrilaterals produced is very high, having $\left\|\vec{f}_Q\right\|_{mean}$ equal to 0.209, which is equivalent to a quadrilateral interior angle range of 90°±5.9°.

Figure 16. Opening in a multi-material domain

Figure 17. Second example of openings in a multi-material domain

A final example mesh has been produced which included many curved concave edges and a rapid variation of segment lengths. Figure 18 shows that elements of a very high quality can be generated for highly concave and curved regions having internal openings. Very small element sizes h_i were imposed around the "U" hole intentionally compared to the rest of the mesh to observe the effect of doing so. The maximum and minimum triangular quality factors $\left\|\vec{f}_T\right\|$ in the final mesh were found to be equal to 1.050 and 0.031, corresponding to angle ranges of 60°±34.7° and 60°±1.0° respectively. The overall mean triangular mesh quality factor 0.421 corresponds to an angle range of 60°±13.9°, which along with the angle ranges above, indicates that the element shapes still remain roughly equiangular even though the transition in element sizes is very rapid. As can be seen from the previous examples, the mesh generation algorithm is very versatile and can handle any complicated geometries.

Figure 18. Domain with multi-curved boundaries

7 Discussion

As can be seen from the previous sections, two-dimensional mesh generation is a well-developed area and different mesh generation methods have different features. However by combining both the Delaunay and advancing front methods, the new mesh generation technique has the advantage of being much more efficient. Several elements are created at a time since the elements in the neighbourhood of a point or element being created are known, which would require a high computational effort to determine using the classical advancing front method. Using the Delaunay criterion facilitates the process of avoiding any potential collision of two fronts, and allows them to be merged easily, with a gradual change across differing element densities, as shown in the various examples in section 6.

The mesh generator *DBMesh* can act as a stand-alone mesh generator, where the mesh geometry of the external and internal domains, along with the boundary conditions, and element densities h_i are defined and used to generate the mesh. Alternatively, the element densities h_i are generated using an error criterion from an adaptive finite element program, and a new mesh is generated. The mesh generator only requires specifying the stress and boundary conditions either at individual points or on defined segments of the problem geometry (internal or external). After element generation, the respective values are interpolated accordingly. The mesh generator also accounts for multi-material layers and any number of internal holes of any geometry or curvature. These are present in most mechanical/structural mesh generators, but are especially important in soil mechanics, where soil-interaction problems involving underground structures and variable soil layers are the norm. The extra option of specifying the type and polynomial order of elements to be generated in the various layers has been added, which is facilitated by the method in which nodes and elements are generated. Excavation and construction are an integral part of civil engineering processes, and simulating them requires the ability to remove and add layers of elements during the analysis. This is achieved by initially defining the different constructed layers in question as superimposed domains on the original layers to be excavated, without the user having to know the elements/nodes in question. Elements and nodes in these additional superimposed layers are then numbered and element properties assigned.

The quality of the triangular elements created by the new mesh generation scheme has been shown to be of near-equilateral quality. The use of the quality factors is necessary if a hybrid element mesh is required, due to the way in which quadrilateral elements are generated. Despite the mesh generator having the capability of generating pure quadrilateral meshes, it is advisable to use a pure quadrilateral generation scheme. The large variety of meshes tested during the development stage, accompanied by the generator's ability to handle with ease any arbitrarily shaped multi-material domain having internal openings and vigorously changing element sizes, have thus proven the generator to be both robust and versatile. The mesh generator and source code may be obtained free of charge for non-commercial use by contacting the author.

Appendix : Some computational geometry concepts

In this appendix, several definitions and concepts will be introduced. These are required for the mesh generation algorithm.

A.1 Orientation of a point C with respect to segment AB

When moving around the sub-domains in an anticlockwise direction, one has to ensure that the points are to the left of the boundary. If a point C is to the left of a segment AB on the boundary, then the area Δ of the counter-clockwise triangle ABC, given by equation (A1), would be positive. One point to note is that if point C is co-linear with AB, then 2Δ reduces to zero.

$$\text{Area } \Delta = \left\| \vec{A} \times \vec{B} \right\| \Big/ 2 = \frac{\begin{vmatrix} x_C & y_C & 1 \\ x_A & y_A & 1 \\ x_B & y_B & 1 \end{vmatrix}}{2} > 0 \Leftrightarrow \text{point C is left of AB} \qquad \text{(A1)}$$

A.2 Is a point C inside a polygon ?

At first sight, a simple solution to such a problem would be to loop over the polygon edges and ensure that the point is to the left of the polygon boundary transversed anticlockwise. This would be fine for convex polygons, but would not work with non-convex ones. A way around this problem would be to make use of the ray-tracing technique. A horizontal ray in the positive x-axis direction emanating from point C is assumed and the number of intersections of the ray with the polygon boundary edges are recorded. The point is inside the domain if the number of intersections is odd and outside if the number is even. Two special conditions however have to be taken account of; if the ray is co-linear with one of the edges, or hits a vertex. A way of treating these two cases is to ensure that the boundary edge-points are on alternate sides of the ray when checking their intersection, as described in section A.3. This procedure also applies to domains with internal holes.

A.3 Intersection of two line segments AB and CD

While generating nodes/elements, one must check that a prospective element does not cross the domain boundaries. This in effect is a problem of checking the intersection of the element edge AB with the domain boundary edge CD. Assuming that no three points of the four points are co-linear, point C is compared with segment AB to check whether it is to the left of it using equation (A1). If this is the case, then AB intersects CD if point B is left of CD. If C is not left of AB but point D is, then the two segments intersect if A is left of CD. In Boolean logic, this is written as [(C left of AB) $\oplus$ (D left of AB)] $\wedge$ [(A left of CD) $\oplus$ (B

left of CD)]. An alternative form would be to ensure that ΔABC*ΔABD and ΔCDA*ΔCDB are both negative if AB and CD intersect, where Δijk is triangle ijk's area. However, Figure A.1(a) depicts a situation where the above criterion would be satisfied but the two segments do not actually intersect, and the need for another check is thus necessary.

Figure A.1 Special cases of intersection

This problem is resolved by checking that one endpoint of one of the segments is in between the two endpoints of the other segment. This second check would also account for the case shown in Figure A.1(b), where three points are in fact co-linear.

A.4 Distance from a point C to segment AB

The distance *d* between a point C having co-ordinates (x_C,y_C) and segment AB connected by points (x_A,y_A) and (x_B,y_B) respectively, can be found using equation (A2).

$$\begin{vmatrix} x & y & 1 \\ x_A & y_A & 1 \\ x_B & y_B & 1 \end{vmatrix} = 0 \Rightarrow ax + by + c = 0 \qquad \text{(A2a)}$$

$$d = \left| \frac{ax_C + by_C + c}{\sqrt{a^2 + b^2}} \right| \qquad \text{(A2b)}$$

A.5 Delaunay triangulations

For a set of points *S*, a triangulation is mathematically defined as a set of segments whose endpoints are in *S*, which only intersect each other at endpoints, and which partition the convex hull of *S* into triangles. The convex hull of *S* may be defined as the smallest convex polygon, having the smallest perimeter, that encloses all the points in this set. The basic property of a Delaunay triangulation is that no points exist inside the circumcircle defined by the three corners of the triangles in 2D and no points inside the circumsphere defined by the 4 corners of the tetrahedra in 3D. The effect of such a property is to maximise the minimum angle over the triangulation.

A clever way of checking the Delaunay property was suggested by O' Rourke [O' Rourke, 1995 #313]. An alternative interpretation of the Delaunay triangulation is its equivalence to being the projection onto the xy-plane of the lower convex hull of the transformal points in three dimensions, transformed by mapping upwards to the paraboloid z

$= x^2 + y^2$. This means that instead of checking that the triangles' circumcircle is free of points, one could just check that the points are on or above the plane containing the triangle in order for it to be Delaunay. This may be done by computing the normal to the triangle ABC, and then deciding based on the sign of the dot-product of the normal and a vector between one of the triangle's vertices to the point in question.

A.6 Position of a point D with respect to triangle ABC's circumcircle

The circumcircle of a triangle is a circle that passes through a triangle's three vertices and is unique to that triangle. Figure A.2 shows a typical triangle ABC, surrounded by its circumcircle of centre O. The equation of the circumcircle may be found using the determinant in equation (A3).

Figure A.2 Circumcircle of triangle ABC with centre O

$$\begin{vmatrix} x^2+y^2 & x & y & 1 \\ x_A^2+y_A^2 & x_A & y_A & 1 \\ x_B^2+y_B^2 & x_B & y_B & 1 \\ x_C^2+y_C^2 & x_C & y_C & 1 \end{vmatrix} = 0 \qquad \text{(A3)}$$

The circumcentre is located at the intersection of the perpendicular bisectors of the triangles' sides, which may also be outside the triangle. After mathematical manipulation, the circumcentre (x_O,y_O) is found to be equal to equation (A4), with the diameter equal to the product of the three triangle sides' lengths divided by twice the triangle's area.

$$\begin{gathered} x_O = \frac{m_1 m_2 (y_A - y_B) + m_2 (x_A + x_C) - m_1 (x_C + x_B)}{2(m_2 - m_1)}, \; m_1 = \frac{y_C - y_A}{x_C - x_A} \text{ and } m_2 = \frac{y_C - y_B}{x_C - x_B} \\ y_O = -\frac{1}{m_1}\left(x_O - \frac{x_A + x_C}{2}\right) + \frac{y_A + y_C}{2} = \frac{m_2 (y_B + y_C) - m_1 (y_A + y_C) + x_B - x_A}{2(m_2 - m_1)} \end{gathered} \qquad \text{(A4)}$$

A simple check on the position of any arbitrary point D with respect to the circumcircle may be made by computing the distance d_{DO} = d(D,O) between D and the circumcentre. If d_{DO} is less than, equal to, or greater than the radius of the circumcircle, then the point is inside, on, or outside the circumcircle respectively.

References