COMMUNICATIONS IN APPLIED NUMERICAL METHODS, Vol. 6, 101–109 (1990)

THE INTEGRITY OF GEOMETRICAL BOUNDARIES IN THE TWO-DIMENSIONAL DELAUNAY TRIANGULATION

N. P. WEATHERILL
Institute for Numerical Methods in Engineering, University College of Swansea, Singleton Park, Swansea, U.K.

SUMMARY

The Delaunay triangulation has recently received attention as a viable method for constructing computational meshes. However, an arbitrary boundary definition which must be preserved in the triangulation process will not, in general, satisfy the geometrical definition on which the Delaunay construction is founded. The effect of this is that the integrity of the given boundary edges will be violated and the computational mesh will not conform to the applied geometrical shape. A method is proposed whereby boundary data are supplemented with points to ensure that imposed boundary edges are preserved during the Delaunay triangulation. The method is illustrated on a geometry of an estuary which exhibits highly complex geometrical features.

INTRODUCTION

As numerical techniques for the solution of engineering problems become more advanced there is an increasing demand for the simulation to incorporate realistic geometrical information. In fact, the geometrical details of many engineering problems are essential for a meaningful understanding of the physical processes. Hence, in parallel with algorithm development, new innovative methods for the automatic generation of computational meshes have received much attention.

Many approaches have been followed and the literature is now extensive.[1] A simple but useful classification is that of methods which generate structured meshes, which possess curvilinear co-ordinate networks of points and associated connectivities, and methods which generate unstructured meshes which, in general, are assemblages of triangles in two dimensions and tetrahedra in three dimensions. Meshes of the latter type require a connectivity matrix to be introduced to explicitly define the connections between points. This is not required for a structured mesh since the points map to a matrix where it is assumed that neighbours in the matrix are neighbours in the physical space in which the mesh is constructed. Traditionally, finite difference methods have employed structured curvilinear meshes whilst finite element methods have utilized unstructured meshes. The relative advantages and disadvantages from the view point of computational fluid dynamics together with details of some methods have been discussed elsewhere.[2]

One of the major advantages of the unstructured approach is that it is applicable to very complex geometrical domains. This is related to the fact that unstructured meshes do not possess any global property and elements can be added or deleted locally as the geometrical features of the domain dictate. One technique for the construction of unstructured meshes which has received much attention is that based upon the Voronoi diagram.[3–8] Interestingly,

0748–8025/90/020101–09$05.00

Received 6 June 1989
Revised 5 October 1989

this construction has been widely used in the field of computer science, where it is used to compute nearest-neighbour problems efficiently.[9]

Although the method is reliable for connecting together arbitrary point sets, the nature of the geometrical criterion on which the Voronoi construction is based does not lend itself for application as a mesh generator for arbitrary geometrical boundaries. As the application of the Voronoi construction in mesh generation becomes more widespread this problem receives more attention.[7,8] In previous work[7] it was proposed that this problem could be overcome by constructing an initial Delaunay triangulation and then testing if the required boundaries are contained in the triangulation. If some edges or faces of the boundary are found to be missing, additional points are added to 'stitch' the hole in the boundary. This approach requires that the integrity of the boundary is checked within the Delaunay algorithm. Here a method is proposed whereby, through an analysis of the boundary points and connectivities performed prior to the Delaunay construction, a set of points is derived which ensures that the resulting Delaunay triangulation contains the required boundary definition. The method is simple to apply and proves to be robust. Examples are given for two-dimensional domains which demonstrate how the Delaunay triangulation method can be applied to generate meshes in regions with highly complex boundaries.

VORONOI NEIGHBOURHOODS AND THE DELAUNAY TRIANGULATION

Given a set of points in the plane, a region can be assigned to each point such that the area of the region is closer to that point than to any other point in the set. This tessellation of a closed domain results in a set of non-overlapping convex polygons which are called Voronoi regions. More formally, if a set of points is denoted by $\{P_i\}$, then the Voronoi region $\{V_i\}$ can be defined as

$$\{V_i\} = \{p : \| P - P_i \| < \| P - P_j \|, \forall j \neq i\}$$

i.e. the Voronoi region $\{V_i\}$ is the set of all points of p that are closer to P_i than to any other point. The sum of all points p forms a Voronoi polygon. From this definition it is clear that, in two dimensions, the territorial boundary which forms a side of a Voronoi polygon must be mid-way between the two points which it separates and is thus a segment of the perpendicular bisector of the line joining these two points. If all point pairs which have some boundary in common are joined by straight lines, the result is a triangulation of the convex hull of the data

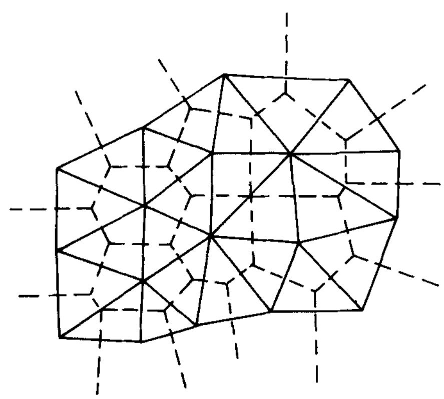


Figure 1. A Voronoi diagram and the Delaunay triangulation

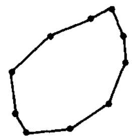


Figure 2. A boundary point distribution which automatically satisfies the Delaunay criterion

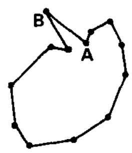


Figure 3. A boundary which does not satisfy the Delaunay criterion. In this case the imposed edges will not be formed in the Delaunay triangulation of the points

points. This triangulation is known as the Delaunay triangulation of the set of points $\{P_i\}$. It is this triangulation which, in general, is of value as the computational mesh. Further details of the properties and algorithm construction can be found elsewhere.[10,11]

It is apparent that for each triangle there is an associated vertex of the Voronoi diagram which is at the circumcentre of the three points which form the triangle. In other words, each Delaunay triangle has a unique vertex of the Voronoi diagram, and no other point within the Voronoi structure lies within the circle centred at this vertex. Figure 1 shows a schematic version of a Voronoi diagram and associated Delaunay triangulation of a small set of points.

For a triangulation which conforms to a given set of boundary edges this definition requires that a circle is assigned to every triangle with a side which forms an edge of the boundary. For many geometrical shapes this is possible. In fact, for example, in applications of the Delaunay triangulation in computational aerodynamics the geometrical nature of aerodynamic shapes is such that, in general, after the introduction of the points which define the boundaries, the triangulation results in boundary conforming edges. At this stage, before the introduction of additional interior domain points, the circles corresponding to the triangles whose sides form the desired boundaries are flagged and protected such that they are not broken by the introduction of any new point. Hence, the final triangulation is boundary conforming.[3-5] A boundary point distribution for which this strategy is applicable is shown in Figure 2. Although this approach is widely used it is clear that for some boundary point distributions it is not possible to construct a Delaunay triangulation which results in a boundary conforming set of triangles. For example, in Figure 3 it is not possible to ensure that edge AB is contained within a Delaunay triangulation since it proves impossible to construct a circle through AB and any other point which does not include at least one other point in the set $\{P_i\}$. In this paper a procedure is proposed to alleviate this problem.

ALGORITHM

The basis for the algorithm with which arbitrary boundary shapes can be made to conform to the Delaunay criterion can be found in the work of Peraire[12] on the mesh generation procedure based on the concept of the advancing front. In this work an algorithm was proposed whereby triangles were formed by connecting edges on the front to new points or existing points. The procedure ensures that no triangles could be formed which contain any other point or that edges intersect any other edge (except at the given nodes of the triangulation). It is clear that if this first criterion is replaced by one which ensures that no other points lie within the circumcircle through the three points which form the triangle then the resulting triangulation will be Delaunay-satisfying. This observation motivates the following algorithm, used to ensure that given boundaries can be preserved during the Delaunay triangulation:

STEP 1
Input boundary edges $\{N_1, N_2\}^j$, $j = 1, 2, \ldots, M$ and $\delta_{1\max}$.
STEP 2
Set up linked list of boundary connectivities

FOR EACH BOUNDARY EDGE
STEP 3
Set $\delta_1 = \delta_{1\max}$
STEP 4
Find mid-point C of the edge (N_1, N_2).
STEP 5
Find the intersection of the perpendicular bisector of (N_1, N_2) with the circle centred on C and radius $\delta_1 \times (CN_1)$. Two roots are obtained: $(X_{\text{new}}, Y_{\text{new}})$ and $(X'_{\text{new}}, Y'_{\text{new}})$.

STEP 6
Find all existing nodes $\{a_i\}$ which lie inside the circle centred on $(X_{\text{new}}, Y_{\text{new}})$ with a radius $\delta_1 \times (CN_1)$.
STEP 7
Order points $\{a_i\}$ such that the first in the list is nearest to $(X_{\text{new}}, Y_{\text{new}})$.
STEP 8
Decide if $(X_{\text{new}}, Y_{\text{new}})$ is to be added to the list of $\{a_i\}$.
If $N_1 a_i < 1{\cdot}5 \times (\delta_1 CN_1)$

and

$$N_2 a_i < 1{\cdot}5 \times (\delta_1 CN_1)$$

then place $(X_{\text{new}}, Y_{\text{new}})$ in the first position in the list $\{a_i\}$.
If not, do not include $(X_{\text{new}}, Y_{\text{new}})$.

DETERMINE THE CONNECTING POINT $\{a_i\}$
FOR EACH i.
STEP 9
Find centre of circle passing through $\{a_i, N_1, N_2\}$
STEP 10
Does the circle through $\{a_i N_1 N_2\}$ contain any other points?
If yes: next i and set KFLAG1 = 1.
If no: check if new point lies within existing circles.
If yes: next i and set KFLAG2 = 1 and store edge number related to this circle E_j.
If no: form connection with point $\{a_i\}$ and check linked list to test if any other connection can be made automatically.

NEXT EDGE – RETURN TO STEP 3

IF NO VALID CONNECTION FOR ALL i THEN
STEP 11
If both roots $(X_{\text{new}}, Y_{\text{new}})$ and $(X'_{\text{new}}, Y'_{\text{new}})$ have been used go to STEP 12.
If not set $(X_{\text{new}}, Y_{\text{new}})$ to $(X'_{\text{new}}, Y'_{\text{new}})$ and return to STEP 6.

STEP 12
If KFLAG1 = 1 subdivide edge creating new point and return to STEP 2.
If KFLAG2 = 1 subdivide edge first encountered i.e. E_j and return to STEP 2.
Modify δ_1: $\delta_1 = \alpha\, \delta_1$, $\alpha < 1$, and return to STEP 5.

By contrast, it is noted that for the advancing front method of Peraire[12] STEPS 9, 10, 11 and 12 are replaced by:

STEP 9
DETERMINE THE CONNECTING POINT $\{a_i\}$.
FOR EACH i:
Does the triangle $\{N_1 N_2 a_1\}$ contain any of the remaining nodes $\{a_i\}$ excluding C
If yes: next i.
If no: check if line segment from $\{a_i\}$ to mid-point of $(N_1 N_2)$ intersects any other boundary edge.
If no: form connection with point $\{a_i\}$. If (X_{new}, Y_{new}) is chosen as the connecting point add new node to list.

Steps 4–7 are shown diagramatically in Figure 4. δ_1 controls the size of the region away from the edge $N_1 N_2$ within which possible connecting points are located. It has proved advantageous to choose δ_{1max} to be quite large, typically 2. In determining the appropriate connecting point $\{a_i\}$ it is necessary to ensure any candidate does not violate the circumcircle property of the Delaunay triangulation. Firstly, the circle through the points $\{N_1 N_2 a_i\}$ must not contain any other point in the set $\{P_i\}$, and conversely the point $\{a_i\}$ must not lie in any circle already associated with an edge. If either of these two criteria is violated then the next point in the list is chosen. When the list of possible points is exhausted then a number of possible options are available. Firstly, it proves advantageous to attempt the connectivity with the new set $\{a_i\}$ defined by choosing (X_{new}, Y_{new}) to be the other intersection point of the circle and the perpendicular bisector of the edge $N_1 N_2$. If this has also been attempted and no valid connectivity found, then STEP 12 is followed.

If KFLAG1 = 1 this indicates that the edge $\{N_1 N_2\}$ has a length which is such that the circle which passes through the points N_1, N_2 and a_i contains other points. In this case the edge is split into two equal parts by the creation of a new point and the boundary connectivities are redefined.

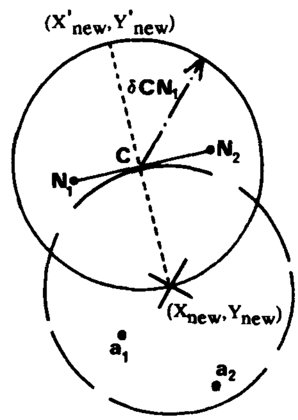


Figure 4. A schematic representation of STEPS 4–7 of the proposed algorithm

If KFLAG2 = 1 this indicates that a potential new point lies inside an existing circle. This implies that the edge associated with that circle is again too large and must be reduced such that a valid new point and connectivity can be found for the present edge j. In this case, as previously, a new point is created on edge j, the boundary connectivities redefined and the procedure returns to STEP 2.

If neither KFLAG1 and KFAG2 have been set, then this implies that no valid point has been found for the value of δ_1 chosen. Hence, δ_1 is reduced for that edge and an attempt is made again to identify a valid point.

With sufficient subdivisions of an edge it is seen that it will always be possible to find or create a point such that a triangle is formed which satisfies the Delaunay criterion, and hence the algorithm will always converge.

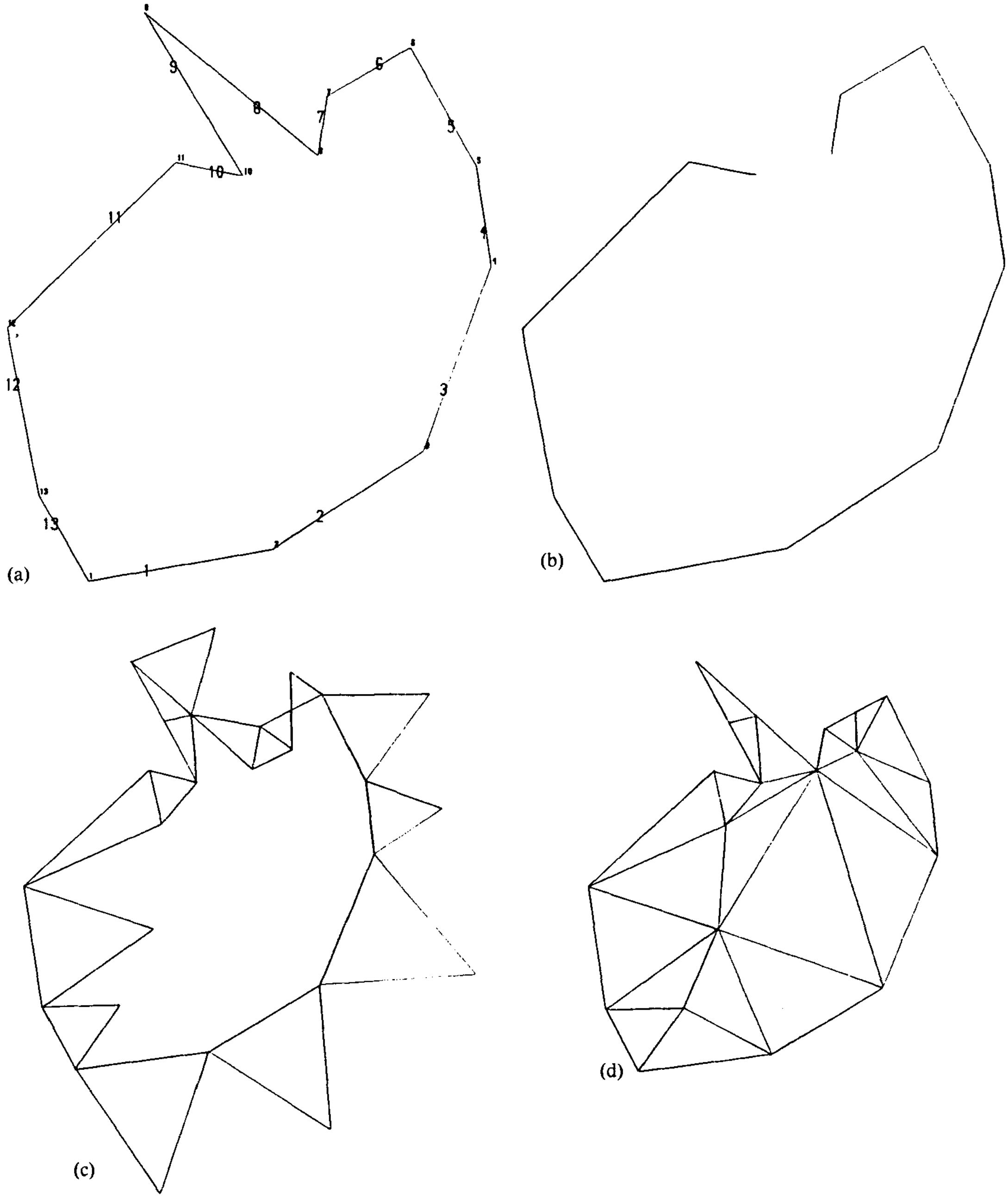
APPLICATIONS

Two examples of the application of the algorithm just described are now presented. The first is a test case designed to illustrate the approach, whilst the second is a realistic example in which a triangulation of an estuary with a particularly complicated geometrical definition is obtained.

Figure 5(a) shows a closed contour consisting of 13 points and 13 edges. It is necessary to construct a triangulation within this closed domain using the Delaunay approach. It is noted that the distribution of points 7, 8, 9, 10, 11 is such that a Delaunay triangulation would not contain some of the edges specified in the boundary definition. In fact, if the points given are connected using a Delaunay algorithm, the resulting boundary edges contained in the triangulation do not contain edges 8 and 9 (Figure 5(b)). At this state the triangulation, since not boundary conforming, would not be suitable as a computational mesh.

If the boundary points and edge definitions are substituted into the proposed algorithm it is found that in order for the specified edges to be preserved two additional boundary points must be added at the mid-point of edges 8 and 9 (Figure 5(c)). This is consistent with expectations as a consequence of the results from Figure 5(b). Also, noted in Figure 5(c) are the additional field points introduced such that other edges on the boundary will be contained in a Delaunay triangulation. With the boundary points and additional points the resulting Delaunay connections involving these points are shown in Figure 5(d) (only points interior to the domain are shown). It is noted that the triangles in Figure 5(c) constructed to ensure integrity of the boundary are present in the Delaunay triangulation of Figure 5(d). At this stage a valid Delaunay-satisfying triangulation of the domain has been achieved. In the conventional way the boundary triangles can be protected and additional points added for a finer discretization of the domain. However, two points have been added on the boundary to achieve this Delaunay triangulation, and since these did not form part of the original given data they should, ideally, be removed. To achieve this all triangles which contain an edge which connects to the added boundary points are deleted and the edges which form a closed contour around the added points are given as boundary data to the advancing front algorithm for connectivity. The effect on the triangulation presented in Figure 5(d) is now shown in Figure 5(e). A Delaunay algorithm has been used to generate a triangulation in which the given boundary data were non-Delaunay-satisfying.

As a more realistic application of the ideas presented here, Figure 6(a) shows the boundary definition of an estuary. A triangulation of the interior region is sought from the Delaunay method. In the usual way, if the boundary points are input to a Delaunay algorithm it is shown in Figure 6(b) that the resulting triangulation does not produce the desired boundary edges. With the application of the proposed algorithm it is seen from Figure 6(c) that additional



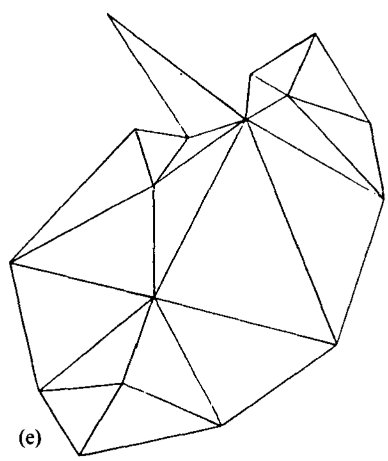


Figure 5. Application of the algorithm to a boundary which does not satisfy the Delaunay criterion: (a) edges of the domain; (b) edges of domain as obtained after Delaunay triangulation; (c) point distribution derived from algorithm; (d) Delaunay triangulation; (e) final triangulation after removal of added boundary points

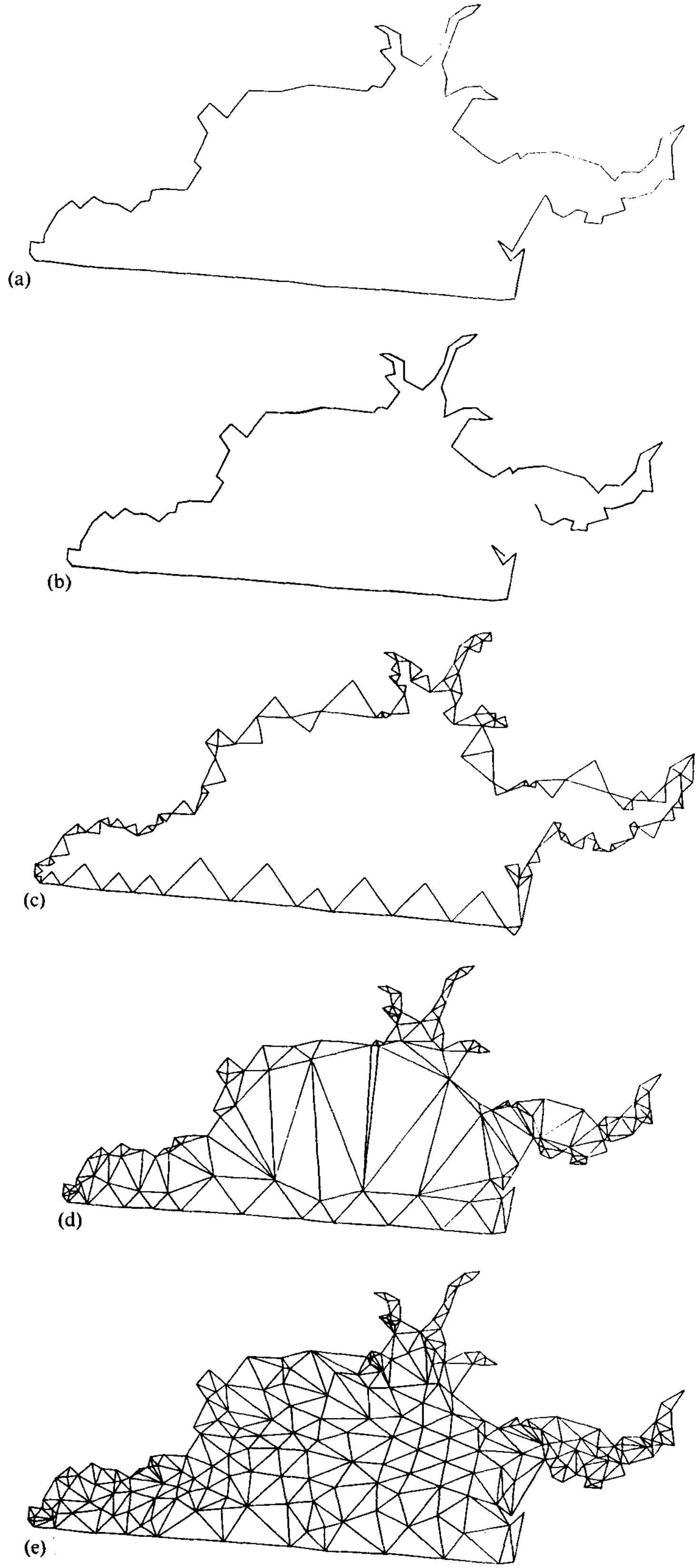


Figure 6. Application of the algorithm to the profile of an estuary (Carmarthen Bay, Severn Estuary, U.K.: (a) boundary of the estuary; (b) edges of estuary as obtained after Delaunay triangulation; (c) point distribution derived from algorithm; (d) Delaunay triangulation; (e) final triangulation after addition of points and smoothing

boundary and field points have been added to ensure that after a Delaunay triangulation the imposed boundary edges will be preserved. A Delaunay triangulation of the points in Figure 6(c) gives rise to the results shown in Figure 6(d). It is clear that the imposed boundary edges have been preserved. At this stage it is possible to add field points to obtain a finer discretization of the estuary, and following this the points added on the boundary can be removed. The resulting mesh, where interior points have been smoothed using a Laplacian filter[3,4], is shown in Figure 6(e).

CONCLUSIONS

If mesh generation based upon the criterion of the Voronoi diagram is to gain widespread appeal in computational methods it must be shown to be applicable to a wide range of highly complex geometrical problems. It is not always possible to construct a Voronoi diagram in which the Delaunay triangulation preserves the given boundary connectivities. Here a systematic method for two-dimensional applications has been proposed for which the Delaunay approach can be adopted with the assurance that the resulting triangulation will conform to the imposed boundary data. The method is very simple to apply, is efficient and robust and can be extended into three dimensions. In this way it should extend the applicability of the Delaunay approach to mesh generation for arbitrarily shaped domains.

ACKNOWLEDGEMENTS

This work was supported by the Aircraft Research Association Ltd, Bedford, UK and the Procurement Executive, Ministry of Defence, Royal Aerospace Establishment, Farnborough, UK. The author acknowledges the input to this work of Dr J. Peraire.

REFERENCES

1. M. S. Sheppard, 'Approaches to the automatic generation and control of finite element meshes', *Appl. Mech. Rev.*, **41**, No. 4 (1988).
2. N. P. Weatherill, 'Mesh generation in computational fluid dynamics', Von Karman Lecture Notes 04, March 1989.
3. N. P. Weatherill, 'The generation of unstructured grids using Direchlet tessellations', Princeton University, MAE Report, No. 1715, July 1985.
4. N. P. Weatherill, 'A method for generating irregular computational grids in multiply connected planar domains', *Int. j. numer. methods fluids*, **8**, 181–197 (1988).
5. T. J. Baker, 'Three-dimensional mesh generation by triangulation of arbitrary point sets', Proc. of the AIAA 8th CFD Conference, Hawaii, June 1987.
6. J. C. Cavendish, D. A. Field and W. H. Frey, 'An approach to automatic three-dimensional finite element mesh generation', *Int. j. numer. methods eng.*, **21**, 329–347 (1985).
7. W. J. Schroeder, and M. S. Shephard, 'Geometry-based fully automatic mesh generation and the Delaunay triangulation', *Int. j. numer. methods eng.*, **26**, 2503–2515 (1988).
8. P. L. George, F. Hecht, and E. Saltel, 'Constraint of the boundary and automatic mesh generation', in Proceedings of second international conference on numerical grid generation in computational fluid mechanics, Miami, December 1988, Pineridge Press, pp. 589–597.
9. R. Sedgewick, *Algorithms*, Addison-Wesley Publishing Co., 1988.
10. A. Bowyer, 'Computing Dirichlet tessellations', *Comput. J.*, **24**, 162–166 (1981).
11. D. F. Watson, 'Computing the n-dimensional Delaunay tessellation with application to Voronoi polytopes', *Comput. J.*, **24**, 167–172 (1981).
12. J. Peraire, 'A finite element method for convection dominated flows', Ph.D. Thesis, University of Wales, 1986.