COMMUNICATIONS IN NUMERICAL METHODS IN ENGINEERING, Vol. 12, 295–302 (1996)

THE FRACTAL BEHAVIOUR OF TRIANGULAR REFINED/DEREFINED MESHES

A. PLAZA DE LA HOZ
Department of Mathematics, University of Las Palmas de Gran Canaria, Campus Universitario de Tafira, 35017-Las Palmas de Gran Canaria, Spain

SUMMARY

In the paper the author presents a novel point of view for the refinement and derefinement algorithms of triangular nested meshes using fractal concepts and iterated function systems (IFS). The fractal behaviour can be understood in the sense that these meshes feature a remarkable amplifying invariance under changes of magnification. Here we compare the meshes obtained by the combination of these algorithms with those presented by Bova and Carey (1992). Although both of the meshes are very similar, the current algorithms automatically build and manage sequences of nested irregular discretizations of the domain. The author illustrates here how the application of IFS families is equivalent to the use of an adaptive strategy that combines the refinement procedure with the derefinement one.

KEY WORDS mesh generation; adaptivity; iterated factal systems

1. INTRODUCTION

When studying finite element methods it is widely accepted that numerical grid generation and the ability to automatically and adaptively control the discretizations in the numerical solution of partial differential equations is critical to the reliable application of numerical analysis techniques, particularly when applying adaptive finite-element methods and multigrid algorithms. It should be understood that a good discretization of the domain in the solution of differential equations is as important as the numerical formulation of the problem.[1] However, we usually require efficient algorithms to achieve a good discretization. For example, this is of particular importance in the consideration of time-dependent problems where the manipulation of refined areas is required.

These algorithms[2–6] are based on the work of M. C. Rivara.[7–9] In fact they can be seen as other versions of her algorithms. These algorithms build and manage automatically sequences of nested irregular discretizations of the domain. This automatic control is very important to apply practical multigrid procedures.

In this paper the latest versions of the refinement and derefinement algorithms are introduced. They can be found in detail in References 2–5. Afterwards, and based on Reference 10, the approach taken there is summarized and particularized to the current algorithms. Finally, several meshes obtained by combining refinements and derefinements are shown.

CCC 1069–8299/96/050295–08

Received 12 July 1995
Revised 2 November 1995

2. THE ALGORITHMS

2.1. *A scheme of the refinement algorithm*

Let $T^n = \{\tau_1 < \tau_2 < \cdots < \tau_n\}$ be a sequence of nested triangular grids, where τ_1 represents the initial mesh and τ_n the finest mesh in the sequence. To refine the sequence, or the level n, means to obtain a new sequence:

$$T^{n+1} = \{\tau_1 < \tau_2 < \cdots < \tau_n < \tau_{n+1}\}$$

The most recent version[5] of the refinement algorithm works in a similar way to the one due to Ferragut:[6]

INPUT: Sequence $T^n = \{\tau_1 < \tau_2 < \cdots < \tau_n\}$
For each $t \in \tau_n$, do:
1. The refinement condition is evaluated.
2. If t must be refined, then:
   2(a) Its edges are marked. Conformity must be assured.
   2(b) For each edge of t, F, say, do:
      While conformity must be assured:
         Assure local conformity of the neighbouring element of suitable edge F.
      End while.
      End for.
   End if.

End for.
OUTPUT: Sequence $T^{n+1} = \{\tau 121 < \mathrm{t}_2 < \cdots < \mathrm{t}_n < t_{n+1}\}$.

It is worth noting here that the (local) conformity can be ensured at the same time as the elements of the level τ_n are taken for the evaluation of the refinement condition. However, this fact does not imply that this version of the refinement algorithm is better than the previous one in the sense of the number of operations. Both algorithms show a similar computational behaviour.

2.2. *A scheme of the derefinement algorithm*

The geometrical problem of the derefinement algorithms consists of eliminating some nodes of the sequence in such a way that the nestedness of the sequence is assured. This problem of the derefinement algorithm can be summarized as follows.

Let $T = \{\tau_1 < \tau_2 < \cdots < \tau_n\}$ be a sequence of nested triangular grids, where τ_1 represents the initial mesh and τ_n the finest mesh in the sequence. Our goal is to obtain another sequence after derefining T, T' say. $T' = \{\tau_1 < \tau'_2 < \cdots < \tau'_m\}$, where $m \leq n$.

The derefinement algorithm can be briefly described in this form:

INPUT: Sequence $T = \{\tau_1 < \tau_2 < \cdots < \tau_n\}$
Loop in levels of T; for $j = n$ to 2, do:
1. For each *proper node* of τ_j the derefinement condition is evaluated and the nodes and edges suitable to be eliminated are pointed out. Conformity is assured.
2(a) If some *proper node* of τ_j is eliminated, then:
   2(a)1 If some *proper node* of τ_j stays, then:
      New nodal connections are defined for the new level j. Genealogy vectors are modified.

In the other case,

2 (a)2 The current level j is deleted in the data structure. Genealogy vectors are modified.

End if.

In the other case,

2 (b) Level τ_j is not modified.

End if.

3. The changes in the mesh are inherited by the following meshes.
4. A new sequence of nested meshes T^j is obtained.

OUTPUT: Sequence $T' = T^2 = \{\tau_1 < \tau'_2 < \cdots < \tau'_m\}$

In this version the conformity is assured at the same time as the nodes of the level τ_j are taken for the evaluation of the derefinement condition. Further details can be found in Reference 5.

The order of operations required by the derefinement algorithm has been estimated by two parameters: the number of nodes in the mesh, *NN*, and the number of levels of mesh in the sequence, *n*, and if we can ensure that the number of levels is bounded we have a linear complexity for the algorithm: O(*NN*).

3. AN ITERATED FUNCTION SYSTEM

We can summarize here Bova and Carey's idea[10] in the design of iterated function systems having the desired attractor on a reference triangular domain. We show here that the refinement/derefinement combination yields *fractal* meshes because its application is thereby equivalent to the Bova and Carey approach. We have chosen the two-dimensional reference element shown in Figure 1(a). The reference triangular (x, h) co-ordinate system may be transformed into any closed, straight-sided physical triangle in the (x, y) co-ordinate system using a bilinear map.[10] The refined triangle obtained by means of the 4-T refinement procedure of Rivara is shown in Figure 1(b).

In order to discretize two-dimensional domains, Bova and Carey in a first stage generate sequences of nodes whose limits approach isolated points, straight-line segments or areas. This set of points in the domain is obtained by applying an appropriate IFS. Towards this end, they construct IFSs whose attractors are points, lines and areas. The second stage consists of an application of a Delaunay triangulation algorithm to achieve the triangulation of the domain.

In our approach for the discretization of two-dimensional domains, points and the related triangular subdomains are constructed at the same time. That is, we do not need to apply the

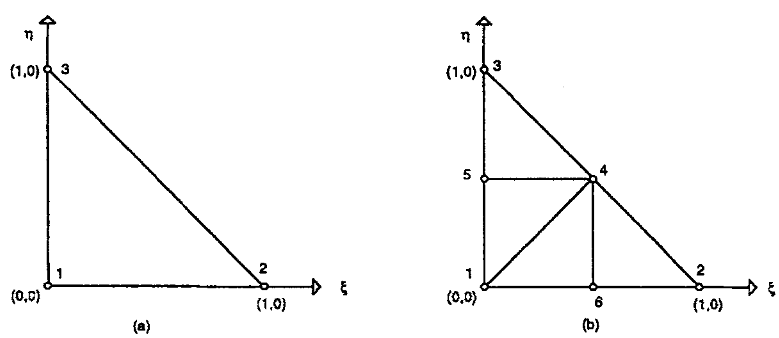

Figure 1: (a) Reference triangular element; (b) refined triangle

Delaunay triangulation algorithm. We can understand this process as the application of a similar IFS to those presented in Reference 10.

3.1. Attractors and iterated function systems

If an IFS consists of a set of n contractive mappings, $\{w_i, i = 1, \ldots, n\}$, then its attractor, A, is, by definition, the union of its images under each of the mappings:[10,11]

$$A = \bigcup_{i=1}^{n} w_i(A) \tag{1}$$

That is, if a point is on the attractor, all its images under the IFS will also lie on the attractor. In other words, A is the fixed point belonging to the corresponding space of non-empty compacts subsets $\mathcal{H}(X)$, with the Hausdorff metric $h(d)$ for the mapping defined in $\mathcal{H}(X)$ by

$$W(B) = \bigcup_{i=1}^{n} w_i(B) \tag{2}$$

for all $B \in \mathcal{H}(X)$, where $X = R^2$ and d is the Euclidean metric in the plane. From this point of view, the attractor associated to an IFS is only determined by the set of mappings of the IFS, and not by the particular initial point of $\mathcal{H}(X)$, that is the initial compact subset since, under the completeness

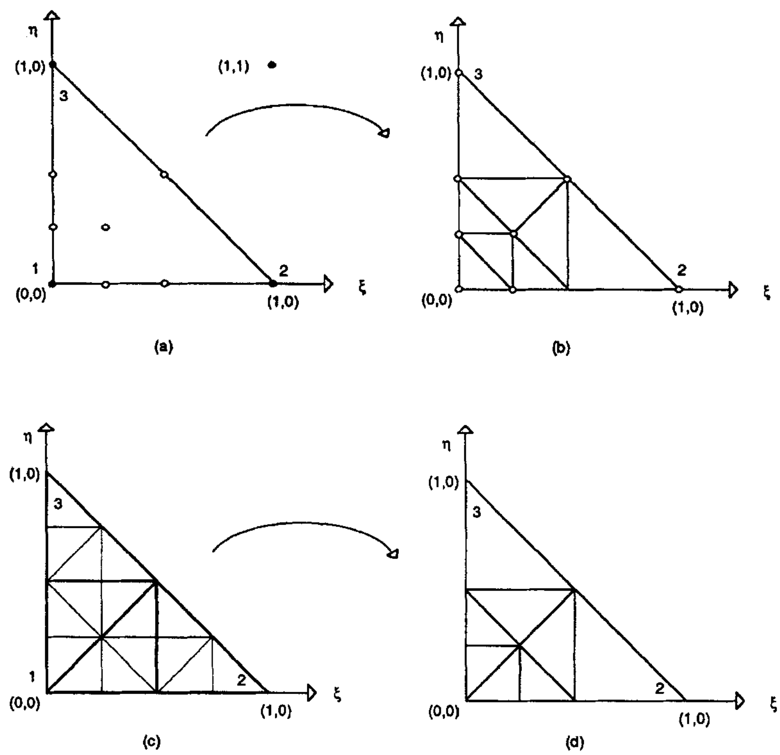

Figure 2. Refinement of reference triangle towards vertex 1: (a) set of points by application of an IFS (refinement); (b) triangulation; (c) global refinement; (d) derefinement

of the metric space (X, d), the completeness of the metric space $(\mathcal{H}(X), h(d))$ is also assured, and the mapping defined by (2) is then a contractive mapping from $(\mathcal{H}(X), h(d))$ into itself.

3.2. Adapted meshes and iterated function systems

We show here that the adapted meshes for a singularity on a point, a side or a triangle are equivalent to the application of some IFSs but now with the refined triangle of Figure 1(b) as initial entrance, instead of a set of points.

To refine the reference triangle of Figure 1(a) towards vertex 1, let us consider Figure 2. This shows the triangulation obtained by Bova and Carey (a, b) and the analogous discretization obtained by the application of two global refinements followed by a derefinement procedure considering a singularity in node 1 (c, d).

Bova and Carey pointed out that it is desirable to bisect each of the three sides of the subtriangles formed during their refinement, because if only the three nodes of the reference triangle are used as starting iterates, the hypotenuse will not be bisected, and poorly shaped triangles will be formed during their triangulation. Hence the IFS and the associated set of starting iterates, $\{(\xi, \eta\}_0$ for refining the reference triangle towards vertex 1 are:

$$w(x) = \begin{pmatrix} 1/2 & 0 \\ 0 & 1/2 \end{pmatrix}\begin{pmatrix} \xi \\ \eta \end{pmatrix}$$
$$\{(\xi, \eta)\}_0 = \{(0,0), (1,0), (1,1), (0,1)\} \tag{3}$$

In our case, to obtaining the mesh of Figure 2(d) we can apply the same IFS above, but now with the initial input given in Figure 1(b).

To refine the reference triangle towards the side which connects nodes 1 and 2, an IFS of the form $\{w_1, w_2\}$ is given in Reference 10 as:

$$w_1(x) = \begin{pmatrix} 1/2 & 0 \\ 0 & 1/2 \end{pmatrix}\begin{pmatrix} \xi \\ \eta \end{pmatrix}$$
$$w_2(x) = \begin{pmatrix} 1/2 & 0 \\ 0 & 1/2 \end{pmatrix}\begin{pmatrix} \xi \\ \eta \end{pmatrix} + \begin{pmatrix} 1/2 \\ 0 \end{pmatrix} \tag{4}$$

and with associated set of starting iterates being simply the three nodes of the original triangle. They note that, if the refinement of the triangle is towards its hypotenuse, there are holes in the discretization and only by an IFS similar to (4) would some slender triangles result. These problems are avoided in our case by using the refinement/derefinement combination, since the features of the arising meshes depend only on the features of the initial mesh; see Figure 3 to compare the triangulation obtained by Bova and Carey (a) with ours (b). This mesh can be understood also as the application to Figure 1(b) of the similarities:

$$w_1(x) = \begin{pmatrix} 1/2 & 0 \\ 0 & 1/2 \end{pmatrix}\begin{pmatrix} \xi \\ \eta \end{pmatrix} \qquad w_2(x) = \begin{pmatrix} 1/2 & 0 \\ 0 & 1/2 \end{pmatrix}\begin{pmatrix} \xi \\ \eta \end{pmatrix} + \begin{pmatrix} 1/2 \\ 0 \end{pmatrix}$$
$$w_2(x) = \begin{pmatrix} -1/2 & 0 \\ 0 & 1/2 \end{pmatrix}\begin{pmatrix} \xi \\ \eta \end{pmatrix} + \begin{pmatrix} 1/2 \\ 0 \end{pmatrix} \tag{5}$$

Finally Bova and Carey consider an IFS whose attractor is the original reference triangle. Since a simple scaling about each of the three vertices of the reference triangle will not suffice,

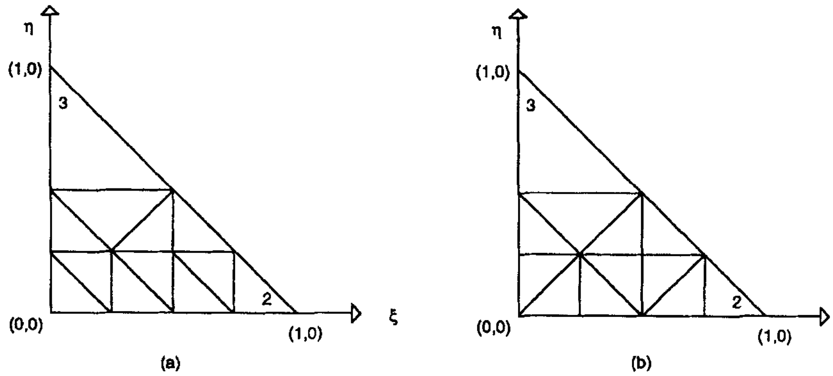

Figure 3. Refinement of the reference triangle to side 1–2

they consider a process of refinement similar to the application of the 2-T refinement procedure of Rivara twice. This process involves two similarities $\{w_1, w_2\}$:

$$w_1(x) = \begin{pmatrix} -1/2 & -1/2 \\ -1/2 & 1/2 \end{pmatrix}\begin{pmatrix} \xi \\ \eta \end{pmatrix} + \begin{pmatrix} 1/2 \\ 1/2 \end{pmatrix}$$
$$w_2(x) = \begin{pmatrix} 1/2 & -1/2 \\ -1/2 & -1/2 \end{pmatrix}\begin{pmatrix} \xi \\ \eta \end{pmatrix} + \begin{pmatrix} 1/2 \\ 1/2 \end{pmatrix} \tag{6}$$

and with associated set of starting iterates being simply the three nodes of the original triangle again. If we apply these two similarities twice we obtain the refined triangle of Figure 1(b); see the bold triangles in Figure 2(c). We can apply the IFS $\{w_i,\ i = 1, \ldots, 4\}$, where w_2 and w_3 are the same as in the previous case,

$$w_1(x) = \begin{pmatrix} 1/2 & 0 \\ 0 & 1/2 \end{pmatrix}\begin{pmatrix} \xi \\ \eta \end{pmatrix} + \begin{pmatrix} 0 \\ 1/2 \end{pmatrix}$$
$$w_4(x) = \begin{pmatrix} 1/2 & 0 \\ 0 & -1/2 \end{pmatrix}\begin{pmatrix} \xi \\ \eta \end{pmatrix} + \begin{pmatrix} 0 \\ 1/2 \end{pmatrix} \tag{7}$$

to obtain the mesh of Figure 2(c), which is the global 4-T refinement of the original input (Figure 1(b)).

4. APPLICATIONS

Here we present some applications to show the behaviour of the combination of refinement and derefinement algorithms. All the examples presented in this section were performed on an HP 710 workstation with the f77 compiler optimization option on.

Figure 4(a) shows the fractal behaviour when a local refinement towards the inner squares and the bold edge inside the domain is performed by combining global refinements and derefinements. This example is similar to one proposed by Bova and Carey using the same initial mesh. The mesh of Figure 4(a) contains 1575 nodes and shows a better fractality than theirs, because it occurs on the sides, not only on the nodes of the mesh. The mesh of Figure 4(b) shows the triangulation obtained in a similar domain, but now with inner circular holes, and contains 2151 nodes.

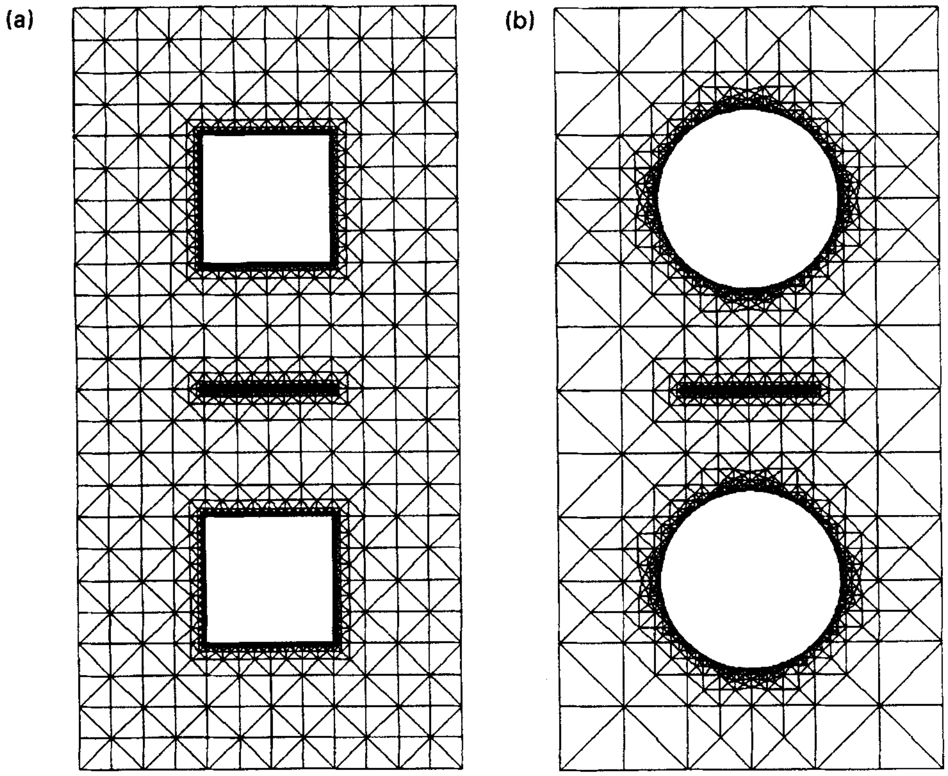

Figure 4. Examples of adapted meshes: (a) adapted mesh with square holes; (b) adapted mesh with circular holes

5. CONCLUSIONS

Although the concepts of iterated function systems to derive a set of intersection-based refinement rules are not in common use in the area of finite element methods, they are equivalent to the use of a combination of refinements and derefinements.

Furthermore, the IFSs proposed by Bova and Carey work almost directly in the case of triangular nested sequences obtained by the algorithms of M. C. Rivara or some other versions like those summarized here. These algorithms not only generate a set of nodes but the whole triangulation, therefore avoiding the use of the Delaunay triangulation algorithm as a post-process which otherwise would be required to obtain the triangulation.

This point of view can be generalized to more dimensions, but until now there are not algorithms analogous to Rivara's. There are already some papers[9,12] in this direction of research, but it is not yet clear whether they are simple generalizations of those. Determining an appropriate IFS and related initial input set in order to solve this generalization would be a worthwhile research effort.

ACKNOWLEDGEMENT

This work was supported in part by Fundación Universitaria de Las Palmas, Grant #2 : sponsor : UNELCO.

REFERENCES

1. J. Z. Zhu and O. C. Zienkiewicz, 'Adaptive techniques in the finite element method', *Commun. appl. numer. methods*, **4**, 197–204 (1988).
2. A. Plaza, 'Derefinement algorithms of nested bidimensional grids' (in Spanish), Doctoral thesis, University of Las Palmas de Gran Canaria, 1993.
3. A. Plaza, L. Ferragut and R. Montenegro, 'Derefinment algorithms of nested meshes', in J. van Leeuwen (Ed.), *Algorithms, Software, Architecture*, IFIP, Elsevier Science Publishers B.V. (North-Holland), Amsterdam, 1992, pp. 409–415.
4. L. Ferragut, R. Montenegro and A. Plaza, 'Efficient refinement/derefinement algorithm of nested meshes to solve evolution problems', *Commun. numer. methods Eng.*, **10**, 403–412 (1994).
5. A. Plaza, R. Montenegro and L. Ferragut, 'An improved derefinement algorithm of nested meshes', in M. Papadrakakis (Ed.), *Advances in Post and Preprocessing for Finite Element Technology*, Civil-Comp Ltd., Edinburgh, Scotland, 1994, pp. 175–180.
6. L. Ferragut, 'A solution to the programming problem of self-adaptive finite element methods' (in Spanish), *An. Ing. Mec.*, **5**, 201–206 (1987).
7. M. C. Rivara, 'A grid generator based on 4-triangles conforming mesh refinement algorithms', *Int. j. numer methods eng.*, **24**, 1343–1354 (1987).
8. M. C. Rivara, 'Selective refinement/derefinement algorithms for sequences nested triangulations', *Int. j. numer. methods eng.*, **28**, 2889–2906 (1989).
9. M. C. Rivara, 'Local modification of meshes for adaptive and/or multigrid finite-element methods', *J. Comput. Appl. Math.*, **36**, 79–89 (1991).
10. S. W. Bova and G. F. Carey, 'Mesh generation/refinement using fractal concepts and iterated function systems', *Int. j. numer. methods Eng.*, **33**, 287–305 (1992).
11. M. F. Barnsley, *Fractals Everywhere*, Academic Press, San Diego, 1993.
12. E. Bänsch, 'Local mesh refinement in 2 and 3 dimensions', *Impact Comput. Sci. Eng.*, **3**, 181–191 (1991).