COMMUNICATIONS IN NUMERICAL METHODS IN ENGINEERING
*Commun. Numer. Meth. Engng* 2001; **17**:903–910 (DOI: 10.1002/cnm.460)

# Graph-based data structures for skeleton-based refinement algorithms

J. P. Suárez[1,*,†], G. F. Carey[2,‡] and A. Plaza[1]

[1] *University of Las Palmas de Gran Canaria, Canary Islands, Spain*
[2] *University of Texas at Austin, Austin, TX, U.S.A.*

## SUMMARY

In this paper, we discuss a class of adaptive refinement algorithms for generating unstructured meshes in two and three dimensions. We focus on skeleton-based refinement (SBR) algorithms as proposed by Plaza and Carey (*Appl. Numer. Math.* 2000; **32**:195) and provide an extension that involves the introduction of the graph of the skeleton for meshes consisting of simplex cells. By the use of data structures derived from the graph of the skeleton, we reformulate the SBR scheme and devise a more natural and consistent approach for this class of adaptive refinement algorithms. As an illustrative case, we discuss in detail the graphs for 2D refinement of triangulations and for 3D we propose a corresponding new face-based data structure for tetrahedra. Experiments using the 2D algorithm and exploring the properties of the associated graph are provided. Copyright © 2001 John Wiley & Sons, Ltd.

KEY WORDS: mesh refinement; skeleton; data structures; edge bisection

## 1. INTRODUCTION

The skeleton view of a mesh provides a hierarchical set of simplex components defining the mesh in a recursive manner as the dimension is decreased. Thus, a mesh of tetrahedra can be defined using a skeleton made up of the surface triangles in the tessellation, these triangles can be viewed as composed of a skeleton of edges, which in turn are composed of a skeleton of nodes. In Reference [1] Plaza and Carey first developed algorithms for grid refinement based on the skeleton in two and three dimensions.

In 2D, the longest edge bisection of a triangle $t$ is the partition of the triangle by the midpoint of its longest edge and the opposite vertex. The longest edge neighbour of $t$ is

[*] Correspondence to: J. P. Suárez, Dto de Cartografía y Expresión Gráfica en la Ingeniería, University of Las Palmas de Gran Canaria, Canary Islands, Spain
[†] E-mail: josepa@cicei.ulpgc.es
[‡] E-mail: carey@cfdlab.ae.utexas.edu

Contract/grant sponsor: Gobierno de Canarias; contract/grant number: PI-1999/146
Contract/grant sponsor: DOE ASCI; contract/grant number: B347883
Contract/grant sponsor: DOD; contract/grant number: NRC-CR-97-0002

*Received 9 April 2001*
*Accepted 7 May 2001*

the neighbouring triangle $t^*$ which shares with $t$ the longest edge of $t$. The longest edge propagation path (LEPP) of a triangle $t$ as defined by Rivara in Reference [2] is an ordered list of all the adjacent triangles $\{t_0 = t, t_1, \ldots, t_n\}$ such that $t_i$ is the neighbour triangle of $t_{i-1}$ adjacent to the longest edge of $t_{i-1}$. The 4T longest edge (4T-LE) partition bisects the triangle in four triangles where the triangle is first subdivided by its longest edge, and then the two resulting triangles are bisected by joining the new midpoint of the longest edge to the midpoint of the remaining two edges of the original triangle.

In 3D, for any tetrahedron $t$ with unique longest edge, the 8-tetrahedra longest edge (8T-LE) partition of $t$ is defined as follows: (1) LE-bisection of $t$ producing tetrahedra $t_1$, $t_2$; (2) Bisection of $t_i$ by the midpoint of the unique edge of $t_i$ which is also the longest edge of a common face of $t_i$ with the original tetrahedron $t$, producing tetrahedra $t_{ij}$, for $i, j = 1, 2$; and (3) Bisection of each $t_{ij}$ by the midpoint of the unique edge corresponding to an edge of the original tetrahedron.

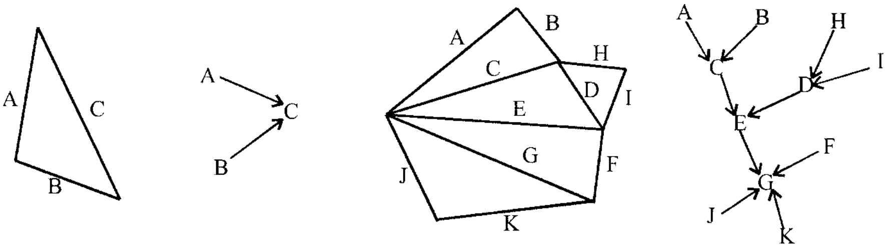
We use standard notations and terminology in graph theory as in References [3, 4]. We denote $l_k \sim (p_i, p_j)$ to represent the link $l_k$ associated with nodes $p_i$, $p_j$. A planar graph is a graph which can be drawn in the plane such that links intercept, in a geometric sense, only at their nodes. The depth-first traversal of a graph implies complete traversal of the graph as follows: starting with an initial node, a successor is visited, then the successor of this successor, and so on, until no further successor is found. Then the next successor of the initial node is visited, applying the same process again.

The outline of this paper is as follows: Section 2 develops the skeleton-based graphs of an $n$-dimensional triangulation for $n = 2, 3$. Section 3 provides the new versions of the 2D and 3D skeleton-based refinement (SBR) algorithms using the skeleton graph approach. Finally, results from some experiments in 2D are given to illustrate the use of the algorithm and data structure.

## 2. THE SKELETON-BASED GRAPH OF A TRIANGULATION

*Definition 1* (*$k$-skeleton graph $G^k(P, L)$*)
Let $\tau$ be an $n$-simplicial mesh with the corresponding hierarchy of $k$-skeleton sets, $k < n$. The $k$-skeleton undirected graph $G^k(P, L)$ of $\tau$ is built as follows: Let $P = \{p_0, p_1, \ldots, p_r\}$ so that $p_i$ is a $k$-face of $\tau$, and $L = \{l_0, l_1, \ldots, l_s\}$ so that $l_j$ represents the topological adjacency relationship $\mathbf{R}$ between two different nodes of $P$, that is between two $k$-faces of $\tau$.

Thus, for two different nodes $p_n, p_m$ of $P$, $p_n \mathbf{R} p_m$ iff the $k$-face represented by $p_n$ shares a common $(k-1)$-face with the $k$-face represented by $p_m$. For brevity we denote by $G^k(\tau)$ the $k$-skeleton graph. For each link $l_i \in L$ and nodes $p_n, p_m \in P$ such that $l_i \sim (p_m, p_n)$ we can assign a unique integer label that identifies the $(k+1)$-face or $n$-simplex in which the $k$-skeleton members represented by $(p_m, p_n)$ are contained. Thus, one can recognize the upward relation of the $k$-skeleton members such as, edge-face, face-tetrahedron. Furthermore, if it is relevant to know a relation such as edge-tetrahedron, the label will be a unique integer that refers to a given tetrahedron in the mesh. This feature is appropriate for the data structures of refinement/coarsening algorithms because it maintains upward links in the hierarchical representation of the skeleton.



(a) (b)

Figure 1. 1-skeleton oriented graph in 2D.

### 2.1. *The 1-skeleton graph for longest edge bisection algorithms in 2D*

Algorithms for 2D refinement based on the longest edge bisection of a triangle may use this property to develop directed graphs for the data structure. For example, in the present work, we classify the edges of each triangle into two types: the longest edge named type 1, and the remaining two edges, named type 2, and this idea can be expressed in terms of the 1-skeleton graph introduced previously by considering the relation $\mathbf{R}=$ '*length of edge x in the triangle is less than length of edge y*'. Because $\mathbf{R}$ is an order relation between edges, it is possible to add an orientation to the links in the graph that represent this ordering. Hence, the links in Figure 1 indicate the dependency of the edges when refining to enforce conformity.

*Proposition 1*
Let $G^1$ be the 1-skeleton directed graph of a 2D conforming triangulation $\tau$. Then the graph does not contain cycles [5].

*Proposition 2*
Updating the 1-skeleton directed graph data structure by applying 4T-LE bisection affects only up to three graph nodes in each refinement step.

From the computational point of view, Proposition 2 ensures the locality of the 1-skeleton directed graph data structure.

*Proposition 3*
Let $\tau$ be a 2D conforming triangulation with $N$ triangles and $t$ be an arbitrary triangle of $\tau$. Let $e$ be the longest edge of $t$. Then the LEPP of $t$ can be obtained from a depth-first traversal of the 1-skeleton directed graph of $\tau$ starting from the node $e$ of the graph, with a maximum cost of order $O(N-1)$.

Numerical experiments carried out in this report reveal that asymptotically, the average cost per triangle of traversing the LEPP, approaches a small constant.

### 2.2. *The 1- and 2-skeleton graph for longest edge bisection algorithms in 3D*

We first consider the 1-skeleton graph of a tetrahedron. As in 2D, the same criterion for the relationship between edges is applied. In this case, we can distinguish three different types of edges: The longest edge of a tetrahedron is denoted type 1; The longest edge of

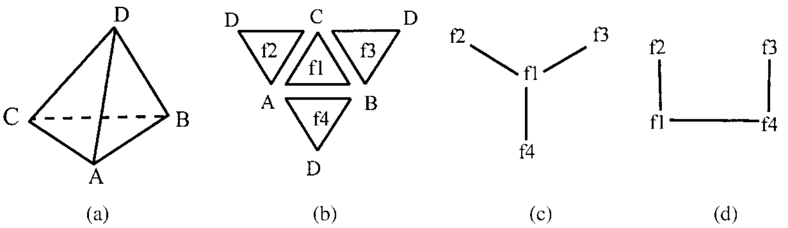


Figure 2. The 2-skeleton graph in 3D.

each of the other two faces not sharing this type 1 edge is type 2, and the remaining edges are type 3.

*Remark 1*
By small (hypothetical) perturbation of the co-ordinates of the nodes one can ensure that there is exactly one longest edge. The longest edge of a tetrahedron is frequently called the reference edge.

*Remark 2*
Some comments concerning the 1-skeleton directed graph for a tetrahedron are warranted:

1. the 1-skeleton directed graph has exactly 4 nodes and 8 links;
2. the degree of the node representing the longest edge is 4;
3. the 1-skeleton directed graph is a planar and disconnected graph and does not contain cycles;
4. each path is of length 1.

The 1-skeleton directed graph of a tetrahedron is an edge-based representation that can be used to model the data structure of any longest edge bisection scheme for tetrahedra (see Section 2.1). From the computational point of view, one can design the refinement algorithm to construct the graph locally for each tetrahedron dynamically. This is the approach we use here for memory efficiency. Alternatively, one could construct the graph and store it for every tetrahedron in the mesh and then update the graph as refinement occurs. This implies an extra data storage of $8n$, $n$ being the number of tetrahedra in the mesh.

When applying the 8T-LE partition, each edge is shared globally by $\frac{36}{7}$ tetrahedra on average [6]. Therefore, in the 1-skeleton graph a node could have on average up to $(\frac{36}{7}) \times 4$ links. This would imply an unacceptable storage requirement, and for this reason it is not efficient to maintain the 1-skeleton graph globally. Instead we provide it locally for each tetrahedron.

*Remark 3*
The representation in Figures 2(a) and 2(b) is unique. In this case edge $AB$ is the longest edge. This representation is called the standard position. We call the faces $f1$ and $f4$ reference faces.

We have two alternatives to build the 2-skeleton graph for a single tetrahedron. The first one is the graph in Figure 2(c). In this case the node with degree 3 is one of the two reference faces. The second possibility is shown in Figure 2(d). Then each node with degree 2 is intended to represent a reference face. Both configurations could be used for our purposes.

By constructing such a graph, we derive a face-based data structure that suits the need of the 3D-SBR algorithm as explained in Section 3.

*Proposition 4*
Let $G^2(\tau)$ be the 2-skeleton undirected graph of a 3D triangulation $\tau$, then the graph is planar and connected.

## 3. THE 2D–3D SKELETON-BASED REFINEMENT ALGORITHMS

The partitions used for refining the triangulation in our algorithms are the 4T-LE approach for the 2D case and the 8T-LE approach for 3D. With the skeleton graph, the algorithms admit a more natural and consistent description (see Figure 3).

The 2D-SBR algorithm as presented by Plaza and Carey [1] works in two sequential stages as follows: (i) identify and bisect the edges specified by the overall refinement indicators (steps 1 and 2 in Figure 3) and (ii) subdivide the individual triangles to define the new triangulation (step 3). Concerning computational time, it should be noted that the 2D-SBR algorithm is of linear complexity with respect to the number of nodes. Numerical experiments carried out in this report reveal that this cost approaches asymptotically a small constant as such a mesh is refined.

In the case of the 3D-SBR algorithm, the four major steps are: (1) subdivision of edges that form the 1-skeleton of the input tetrahedron; (2) subdivision of edges due to the propagation of the conformity to the neighbour tetrahedra; (3) having bisected the edges implied in the refinement, subdivide the corresponding faces according to the 4T-LE partition for 2D; (4) perform the appropriate interior subdivision of the tetrahedron taking part in the

```
Algorithm 2D-SBR($\tau$, $t_0$)
/* Input: $\tau$ mesh, $t_0$ triangle
/* Output: $\tau$ mesh
1. L=1-skeleton($t_0$)
For each edge $e_i \in L$ do
        Subdivision($e_i$)
End
2.For each edge $e_i \in L$ do
        $S_i$=LEPP($e_i$,$G^1(\tau)$)
        For each triangle $t_i \in S_i$ do
                Let $e_i$ be the longest edge of $t_i$
                Subdivision($e_i$)
        End
End
3.For each triangle $t_j \in \tau$ to be subdivided do
        Subdivision($t_j$)
End
```

```
Algorithm 3D-SBR($\tau$,$t_0$)
/* Input: $\tau$ mesh, $t_0$ tetrahedron
/* Output: $\tau$ mesh
1. L=1-skeleton($t_0$)
For each edge $e_j \in L$ do
        Subdivision($e_j$)
End
2.While $L \neq \emptyset$ do
    Let $e_k$ be an element $\in L$
    For each tetrahedron $t_i \in$ hull($e_k$) do
            For each non-conforming face $f_i \in G^2(t_i)$ do
                    Let $e_p$ be the longest edge of $f_i$
                    Subdivision($e_p$)
                    $L=L \cup e_p$
            End
    End
    $L=L-e_k$
End
3.For each face $f_i \in G^2(\tau)$ to be subdivided do
        Subdivision($f_i$)
End
4.For each tetrahedron $t_i \in \tau$ to be subdivided do
        Subdivision($t_i$)
End
```

Figure 3. The 2D-SBR and 3D-SBR algorithms.

refinement, following the 8T-LE partition strategy. It should be pointed out here that the procedure *subdivision* in Figure 3 subdivides the successive skeletons in reverse order: steps 1 and 2 perform the subdivision of the edges, step 3 performs the subdivision of the faces and step 4 performs the subdivision of tetrahedra. The algorithm is of linear complexity in the number of nodes, as stated in Reference [1]. The points where the 2-skeleton graph $G^2(\tau)$ is used are clearly seen in the previous algorithm. First, the inner loop of step 2 accesses the non-conforming faces in the mesh and in step 3 the subdivision of the 2-skeleton is performed. In both steps, the 2-skeleton graph provides access to the faces taking part in the refinement in a local and efficient way. In step 2, the procedure hull($e$) provides the set of simplices $S\in\tau$ such that $e\in S$. That is, hull($e$) is comprised of all the simplices that share the same edge $e$.

## 4. EXPERIMENTS ON THE SKELETON-BASED ALGORITHM (2D CASE)

We consider a particular domain corresponding to the Gran Canaria Island (Spain) and local refinement using 4T longest edge bisection on three contiguous subdomains. We focus on the following two points: (1) a demonstration of the skeleton based algorithm as described in the preceding sections of this paper; (2) some statistical measures of the LEPP subgraph determined at each refinement step. In the first study, we consider the Gran Canaria domain $S$ with mesh refinement on disjoint subregions $S_1$, $S_2$ and $S_3$ with $S=S_1\cup S_2\cup S_3$, for innermost region $S_3$, intermediate region $S_2$ and outermost annular region $S_1$. The goal is to successively refine the three subdomains according to the three elevation regions so defined. The sequences of meshes are obtained by refinement using the LEPP approach and the 2D-SBR algorithm as described earlier. These results effectively demonstrate the use of the graph-based data structure for SBR.

In the second part of the study, we briefly report on the behaviour of the LEPP statistics for the SBR approach. As noted in Proposition 3 earlier, the LEPP is computed directly from the subgraph in our graph based data structure. We report results using two metrics. The first metric is the number of additional triangles refined when refining $t$. The second metric is the maximum length of LEPPs of the neighbours of $t$ without counting triangle $t$. We will refer to these metrics as $M1$ and $M2$, respectively (Figures 4 and 5).

In Figure 5, the mean and standard deviation of $M1$ and $M2$ are given. The results show that $M1$, on average tends to 5. This means that the 1-skeleton graph as presented in

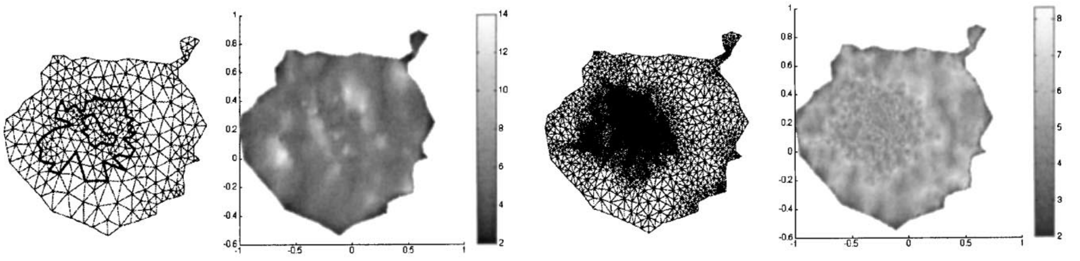


Figure 4. Gran Canaria meshes corresponding to refinement levels 1 and 5, and associated spatial distribution of metric $M1$.

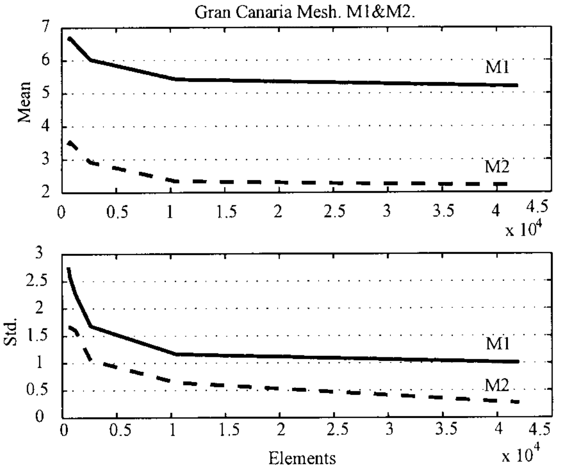


Figure 5. Statistics for Gran Canaria mesh.

Section 2.1 is on average traversed for 5 nodes per element. These results as well as those in the previous experiments show that the cost of performing the local refinement per element with the 2D-SBR algorithm approaches a fixed acceptable value as the number of refinement elements increases. It also suggests that a demanding refinement criterion can be used (since there will be moderate additional neighbour subdivision).

This result for the LEPP approach using these metrics is of considerable interest since the utility of this type of longest edge bisection scheme and their efficient use of memory has previously been questioned. Further details will be presented in more depth in a following study.

## 5. CONCLUSIONS

We have reviewed several refinement ideas in both 2D and 3D and extended our previous work on skeleton-based refinement algorithms. For the 2D-SBR algorithm, an improved and very natural edge-data structure based on graphs has been introduced and studied. Several properties of the supporting data structure are described, showing this to be efficient in storage and time cost. The extension of the graph based data structure for the 3D case it is also presented, proposing a corresponding face-based data structure. We discuss in detail the graphs for 2D refinement of triangulations and for 3D. Finally, numerical experiments to illustrate the ideas presented in the paper are presented using the 2D-SBR algorithm.

ACKNOWLEDGEMENTS

This research has been supported in part by Gobierno de Canarias, grant Number PI1999/146, by DOE ASCI Contract No. B347883, and by DOD Contract No. NRC-CR-97-0002.

## REFERENCES

1. Plaza A, Carey GF. Local refinement of simplical grids based on the skeleton. *Applied Numerical Mathematics* 2000; **32**:195–218.
2. Rivara MC. Mesh refinement based on the generalized bisection of simplices. *SIAM Journal on Numerical Analysis* 1984; **2**:604–613.
3. Gross JL, Tucker TW. *Topological Graph Theory*. Wiley: New York, 1987.
4. Haray F. *Graph Theory*. Addison-Wesley: Reading, MA, 1972.
5. Suárez JP, Carey GF, Plaza A, Padrón MA. Graph based data structures for skeleton based refinement algorithms. *TICAM Report Series 10* 2001.
6. Plaza A. Asymptotic behaviour of the average of the adjacencies of the topological entities in some simplex partitions. *SANDIA Report Sand 99-2288*, 1999; 233–240.