A Two Dimensional Mesh Verification Algorithm

R. Bruce Simpson
Department of Computer Science
University of Waterloo
Waterloo, Ontario, Canada.

Abstract

A finite element mesh is commonly represented in a program by lists of data, i.e., vertex coordinates, element incidences, boundary data. In general, these lists describe a collection of triangles. Whether the triangles form proper mesh for a region or not, i.e. whether they 'tile' a region, is data dependent in a non obvious way. This paper specifies a set of conditions on the triangles (i.e. on the list data) which ensure that the triangles tile a region and which also can be verified by an algorithm which is referred to in the title and which is claimed to be of reasonable efficiency.

Basic List Representation of a Mesh

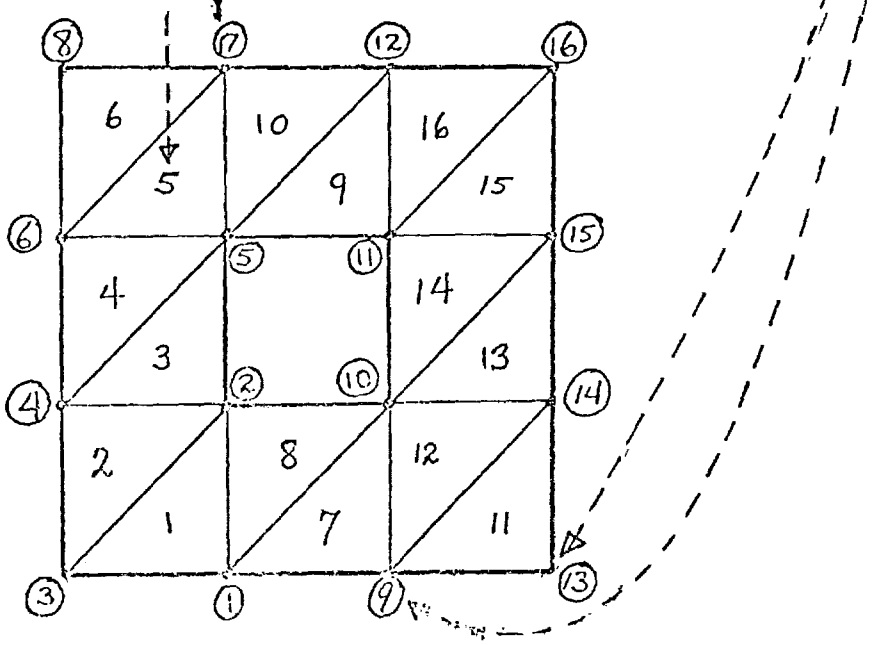
The mesh verification algorithm assumes that the collection of triangles is described by three lists as shown in the following small example.

Vertex Index	Coordinates x	Coordinates y
1	2.00	1.00
2	2.00	2.00
3	1.00	1.00
4	1.00	2.00
5	2.00	3.00
6	1.00	3.00
7	2.00	4.00
8	1.00	4.00
9	3.00	1.00
10	3.00	2.00
11	3.00	3.00
12	3.00	4.00
13	4.00	1.00
14	4.00	2.00
15	4.00	3.00
16	4.00	4.00

Element index	Vertex indices			
1	1	2	3	
2	2	4	3	
3	2	5	4	
4	5	6	4	
5	5	7	6	
6	7	8	6	
7	9	10	1	
8	10	2	1	
9	11	12	5	
10	12	7	5	
11	13	14	9	
12	14	10	9	
13	14	15	10	
14	15	11	10	
15	15	16	11	
16	16	12	11	

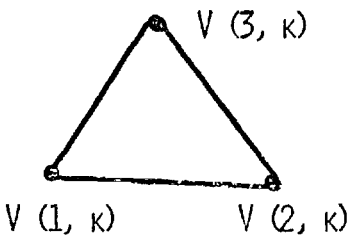
Boundary index	References		
1	1	3	
2	2	2	
3	3	1	
4	4	2	
5	6	1	
6	6	2	
7	7	3	
8	8	1	
9	9	3	
10	10	1	
11	11	1	
12	11	3	
13	13	1	
14	14	2	
15	15	1	
16	16	1	

indicates a boundary edge starts at →



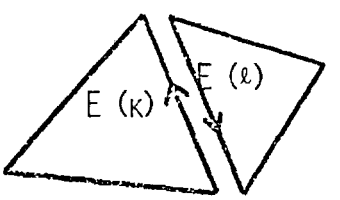
CONDITIONS

C 1) THE TRIANGLE VERTICES ARE SPECIFIED IN COUNTER CLOCKWISE ORDER



C 2) EITHER THE ITH EDGE OF E (K) IS THE ONLY EDGE JOINING ITS END POINTS (BOUNDARY ELEMENT)

OR THERE IS EXACTLY ONE ELEMENT, E (ℓ) HAVING THE SAME EDGE. IN THIS LATTER CASE, THE DIRECTIONS OF THIS LINE SEGMENT AS EDGES OF E (K) AND E (ℓ) MUST BE DIFFERENT.



C 3) NO BOUNDARY EDGE INTERSECTS MORE THAN ONE ELEMENT, EXCEPT AT ITS END POINTS.

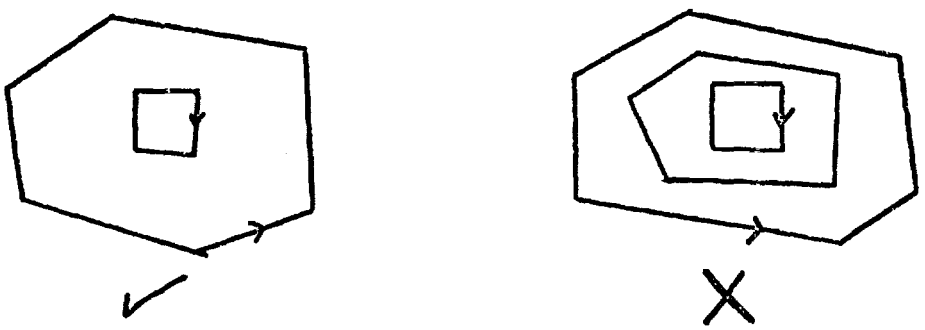
C 4) A VERTEX CAN HAVE AT MOST ONE BOUNDARY EDGE DIRECTED AWAY FROM IT.

IMPLICATIONS

1) MESH BOUNDARY EDGES FORM A SET OF DISJOINT, ORIENTED, SIMPLE CLOSED CURVES

$$C_1,\ C_2,\ \ldots,\ C_K \equiv \text{MESH BOUNDARY CURVES}$$

2) EACH CURVE OF BOUNDED INTERIOR DEFINES A CONNECTED REGION. THE BOUNDARY OF THIS REGION IS COMPOSED OF MESH BOUNDARY CURVES



(ASSUME 1 CURVE OF BOUNDED INTERIOR - C_1)

DEFINE $R \equiv \bigcap_{I=1}^{K}$ (INTERIOR OF C_I)

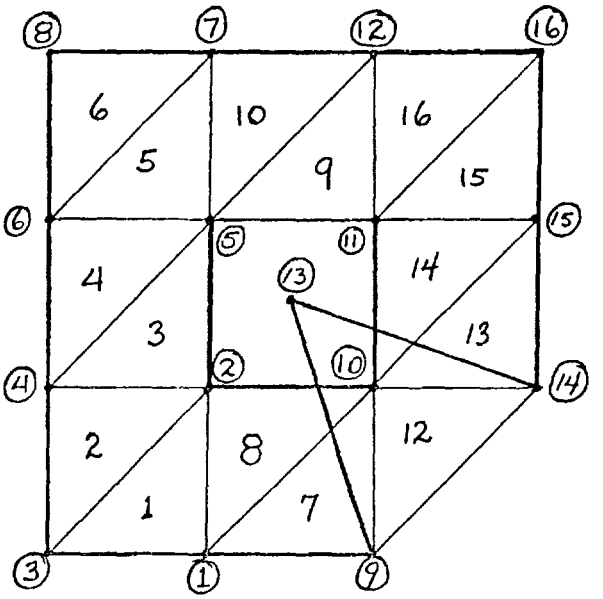
(CONNECTIVITY K)

3) $R = \bigcup_{J=1}^{N_E} E(J)$

4) IF $P \in R$, P IS NOT AN ELEMENT EDGE

$\Rightarrow$ P LIES IN EXACTLY ONE ELEMENT.

Small Example Invalid Mesh on Hollow Square



Coordinates of vertex 13 changed to (2.5, 2.5)

Section of Mesh Verification Algorithm Detailed Error Report

```
 MESH VERIFICATION ERROR

   INTERSECTING BOUNDARY EDGES -
   EDGE FROM VERTEX 13 AT ( 2.50, 2.50)   TO VERTEX  9 AT ( 3.00, 1.00)
   EDGE FROM VERTEX  2 AT ( 2.00, 2.00)   TO VERTEX 10 AT ( 3.00, 2.00)

 MESH VERIFICATION ERROR

   INTERSECTING BOUNDARY EDGES -
   EDGE FROM VERTEX 14 AT ( 4.00, 2.00)   TO VERTEX 13 AT ( 2.50, 2.50)
   EDGE FROM VERTEX 10 AT ( 3.00, 2.00)   TO VERTEX 11 AT ( 3.00, 3.00)

  FROM BDSCAN, NO. OF BOUNDARY CURVES=    2

 MESH VERIFICATION ERROR

   ELEMENT  11 APPEARS TO HAVE VERTICES LISTED IN WRONG ORDER
          X=    3.000000E 00   Y=    1.000000E 00
          X=    2.500000E 00   Y=    2.500000E 00
          X=    4.000000E 00   Y=    2.000000E 00
          DET =   -2.000000E 00

MESH CHECK ENCOUNTERED      3 ERRORS
```