AUTOMATED FINITE ELEMENT GRID BREAK-UP METHOD - A
VERIFICATION OF THE SIX NODE AVERAGING APPROACH

H. J. Barten
Florida Research and Development Center
Pratt & Whitney Aircraft
Division of United Technologies Corporation

SUMMARY

The verification of the nodal averaging method for generating automated
finite triangular element grids has been demonstrated. This was accomplished
with a six node averaging program (SNAP) which was placed on an IBM 2250
vector graphics scope terminal. The advantage of this method is that it is un-
necessary to program time consuming geometric division and transition algo-
rithms.

INTRODUCTION

The most popular method for the automatic break-up of regions into plane
finite elements seems to be variations of the "sides-and-parts" method as dis-
cussed in references 1 through 4. In this method the structure is divided into
four-sided parts with the sides divided into straight line segments connected at
nodes. Figures 1, 2, and 3 show such parts in the form of a square which has
a one-to-one topological correspondence to a general four-sided figure. If the
number of nodes on pairs of opposite sides are equal, lines or curves drawn
through corresponding side nodes will generate internal mesh points at their
intersections, as shown in figure 1.

For further refinement a division algorithm can be used to specify variable
spacing in both directions. If the nodes on one pair of sides are equal in number,
while those on the other pair of sides are unequal, this algorithm can be used to
generate a transition pattern as shown in figure 2. In this instance the algorithm
is that the number of nodes on a row is one less or one more than the number on
an adjacent row. In both of these cases the topological pattern is easily predicta-
ble and is amenable to automatic mesh generation. If the numbers of nodes on both
pairs of sides are not equal, the pattern is generally not too apparent, although
exceptions do occur, as shown in figures 3 and 4. The pattern in figure 3 is ob-
tained by drawing the full lines and completing the triangulation by connecting the
mesh points at their intersections, as shown by the dashed lines. Here there is
a transition from four to five nodes on one pair of opposite sides and a transition
from three to four nodes on the other pair. This pattern could have been achieved
by using the transition algorithm in two directions and a little imagination. Fig-
ure 4 shows the pattern for the same configuration which was obtained by the
method which is described herein. This pattern was obtained without specifying
division or transition algorithms and depends only upon the positions of the side
nodes. Note that it has the same topological pattern as that shown in figure 3;
that the nodes in both figures can be made coincident by shifting without de-
stroying their connectivity.

A more general case in which the division or transition algorithms are
more difficult to formulate is shown in figure 5. This pattern was obtained
again by the method which is described herein and again required the definition
of the position of the side nodes only. Here, the division and transition algo-
rithms are automatic. By inspection it can be seen that the number of nodes
along any interior segmented curve is one more or less than the number on ad-

jacent curves, and hence the transition algorithm is satisfied. The transition
pattern for this configuration could also have been predicted with some imagina-
tion and hind-sight and computerized. However, as the number of side nodes
increases, the transition patterns become more difficult to formulate for com-
puter programming.

NODE AVERAGING APPROACH

This technique was programmed because it makes it unnecessary to use
transition and division algorithms or to input internal paths or subsystems. The
only rigid condition that is stipulated when using node averaging is that each in-
ternal mesh point always be the common vertex of six triangular elements. This
is stipulated in order to simplify the computer formulation and to provide uniform
nodal patterns at each internal mesh point. If this requirement is met, the top-
ological structure of the break-up is a series of nested polygons each of which
has a multiple of six nodes or edges whose number differs by six from the num-
ber in adjacent polygons. The reason that this pattern generates the desired
transition algorithm is quite obvious if one examines the sector ABCDEFG in
figure 6. The number of nodes across the sector increases in increments of 1
as the boundaries of the nested polygons are crossed. Thus, there is one node
at A, two nodes along BE, three nodes along CF and finally, four nodes along
DG. This algorithm exists in each of the six sectors. The triangulation is
achieved by connecting nodes in adjacent polygons as shown in figure 6. Since
the external contour of the region is the extreme polygon, the number of known
contour nodes or edges also must always be specified as a multiple of six.

We now treat the coordinates of the interior nodes as unknowns and stipu-
late that they are the average of the coordinates of the six adjacent nodes. This
is a rough application of the finite difference method for solving Laplace's
equation for a region with prescribed boundary conditions. This algorithm yields
two sets of simultaneous equations in which the coordinates of the internal nodes
are the unknowns. If the number of nodes is too large for simultaneous equation
solutions, the coordinates can be obtained by assuming initial values and iterating
until convergence is attained.

The program which has been developed uses the simultaneous equation
method. If a hole is present in the region, the number of nodes on the hole con-
tour must also be a multiple of six. The contour of the hole then becomes the
first nested polygon while the external contour becomes the last polygon or vice
versa. Since the number of nodes on each polygon follows an arithmetic series,
the total number of nodes and elements can be expressed in terms of the number

of internal and external contour nodes. If \( N_e \) is the number of external nodes and \( N_i \) is the number of internal nodes, these relations are:

\[
\text{Total number of nodes} = \frac{N_e + N_i}{2} + \frac{N_e^2 - N_i^2}{12}
\]

\[
\text{Total number of elements} = \frac{N_e^2 - N_i^2}{6}
\]

If no hole is present, the relations are:

\[
\text{Total number of nodes} = 1 + \frac{N_e}{2} + \frac{N_e^2}{12}
\]

\[
\text{Total number of elements} = \frac{N_e}{6}
\]

Figures 4 and 5 show the patterns obtained by solving the set of linear simultaneous equations using the node averaging algorithm. Attempts to use this algorithm on regions with concave and/or slender contours were generally unsuccessful because mesh points had a tendency to be near a contour or outside the region. Such regions can be handled by dividing them into subregions and piecing together independent solutions. This technique, however, negates the advantage of a single break-up. Other methods of handling concave boundaries are discussed in the references.

This automatic break-up algorithm has been programmed into a finite element structures program and a compatible finite element break-up scope program. This requires the input of contour nodes only as discussed previously. A test case was input and run on the scope program and a plot of the sequence of presentations is shown in figures 7 through 11. The first display (figure 7) shows the contour nodes generated by inputting corner nodes, hole location, and hole diameter while using an automatic boundary break-up option, which is also incorporated in the program. The second display (figure 8) is the generated automatic break-up. The third display (figure 9) is a refined break-up which is obtained by triangulating the mid-points of the edges of each element in the previous coarse break-up. The fourth display (figure 10) shows the coarse break-up as modified by moving nodes with the light-pen. The fifth display (figure 11) is the fine break-up obtained by triangulating the mid-points of the edges in the previous adjusted break-up. The light-pen can also be used to present magnified local regions, sub-systems, and boundary and perimeter nodes.

CONCLUDING REMARKS

A nodal pattern for triangular mesh generation, using nodal averaging techniques, has been developed. This pattern has intrinsic properties which make it unnecessary to program internal transition algorithms. An automated mesh generation program using this technique has been developed for use on the
IBM 2250 vector graphics scope terminal. The use of this program on this terminal allows the operator to modify geometry in order to improve the break-up prior to being input into an associated finite element structures program.

REFERENCES


Figure 1. Conventional simple break-up.

Figure 2. Conventional break-up with single transition.
Figure 3. Double transition break-up.

Figure 4. SNAP break-up.
Figure 5. Complex SNAP break-up.

Figure 6. Topology of SNAP break-up.
Figure 7. Scope display of boundary.

Figure 8. Scope display of coarse SNAP break-up.
Figure 9. Scope display of fine SNAP break-up.
Figure 10. Scope display of adjusted coarse SNAP break-up.
Figure 11. Scope display of adjusted fine SNAP break-up.