JOURNAL OF COMPUTATIONAL PHYSICS 62, 180–198 (1986)

A General Collapsing Technique for Three-Dimensional Algebraic Grid Generation*

GUILLERMO MARSHALL[†]

Aldridge Laboratory of Applied Geophysics, Columbia University, New York, New York 10027

PETER EISEMAN

Department of Applied Physics and Nuclear Engineering, Columbia University, New York, New York 10027

AND

JOHN T. KUO

Aldridge Laboratory of Applied Geophysics, Columbia University, New York, New York 10027

Received May 2, 1984; revised October 16, 1984

We present a general collapsing technique for the generation of a regular lexicographically ordered grid within a three-dimensional object by using multilinear algebraic coordinate transformations. The method is applied to the grid generation of a topologically complex region consisting of various three-dimensional objects. The local lexicographically ordered object grid is transformed, with an appropriate assembling procedure, into a global lexicographically ordered region grid. The method presented possesses simplicity and at the same time a sufficient degree of generality, a considerable amount of grid control, and a desirable degree of global grid uniformity to make it competitive. © 1986 Academic Press, Inc.

INTRODUCTION

Automatic grid generation is an important element in the numerical solution of partial differential equations with geometric complexity. Because of their inherent simplicity and control properties algebraic coordinate transformations have attracted the attention of many researchers who have used them for multidimensional

* This work was supported through Project MIDAS at the Aldridge Laboratory of Applied Geophysics, Columbia University.

[†] Present address: Centro de Cálculo Científico, Comisión Nacional de Energía Atómica, 1429 Buenos Aires, Argentina.

0021-9991/86 $3.00

grid generation (see, for instance, Cook [1, 2], Gordon and Hall [3], and Eiseman [4], among others).

Here we introduce the method of collapsed algebraic transformations, which generalizes the algebraic grid generation. We construct an automatic grid generator for a topologically complex three-dimensional region. This generator is intended for use in finite element analysis of realistic geological configurations.

In a grid generation the desired properties to be obtained are a considerable amount of grid control and a desirable degree of global grid uniformity. In addition to these properties the method should have a sufficient degree of generality for dealing with various three-dimensional objects while keeping the simplicity of regular lexicographically[1] ordered grids.

The outline of our discussion, which is based on a previous report by Marshall, Eiseman, and Kuo [5], is: first we introduce the method of algebraic coordinate generation for single geometrical objects such as the straight line, the quadrilateral, and the hexahedron, and we discuss some examples showing the assembling procedure and the topological description of the lexicographically ordered grid generated; second we introduce the collapsing technique for single geometrical objects such as the triangle, pentahedra, and tetrahedra with examples; finally we present an application for the discretization of a topologically complex three-dimensional region describing realistic geological configurations.

Algebraic Coordinate Transformations

It is convenient for computational purposes to describe geometrical objects in parametric form. For instance if the cartesian coordinates x, y, and z are expressed as linear functions of a parameter t, i.e., $x=x(t)$, $y=y(t)$, and $z=z(t)$, the equations of the straight line passing through the points $\mathbf{P}_1(x, y, z)$ and $\mathbf{P}_2(x, y, z)$ is given by

$$\mathbf{P}(t)=\mathbf{P}_1(x_1, y_1, z_1)+t[\mathbf{P}_2(x_2, y_2, z_2)-\mathbf{P}_1(x_1, y_1, z_1)], \tag{1}$$

where bold capitals indicate vectors and where the parameter t lies between zero and one. In parametric form, the cartesian components are given by

$$x(t)=x_1+t(x_2-x_1) \tag{2}$$

$$y(t)=y_1+t(y_2-y_1) \tag{3}$$

$$z(t)=z_1+t(z_2-z_1). \tag{4}$$

Now given the points $\mathbf{P}_1(x, y, z)$ and $\mathbf{P}_2(x, y, z)$, a discrete and uniformly dis-

[1] The most natural and common way to assign memory space for a table is to store the table following the "lexicographic order" of its indexes; for a rectangular matrix $a(i, j)$ of order n it means: $a(1, 1)$,..., $a(1, n)$, $a(2, 1)$,..., $a(2, n)$,..., $a(n, 1)$,..., $a(n, n)$.

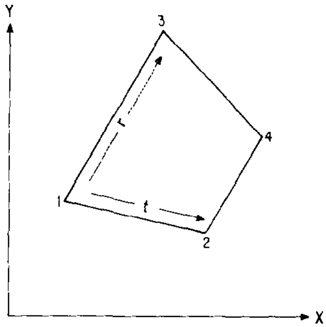

FIG. 1. A quadrilateral region in the plane.

tributed set of points $\mathbf{P}_i(x_i, y_i, z_i)$ is to be generated on the straight line passing through P_1 and P_2. This construction is carried out with the discrete forms of Eqs. (2), (3), and (4), which are

$$x(k) = x_1 + t_k(x_2 - x_1) \tag{5}$$

$$y(k) = y_1 + t_k(y_2 - y_1) \tag{6}$$

$$z(k) = z_1 + t_k(z_2 - z_1), \tag{7}$$

where $t_k = (k-1)/(n_t - 1)$ as k varies from 1 to a total number of points n_t,

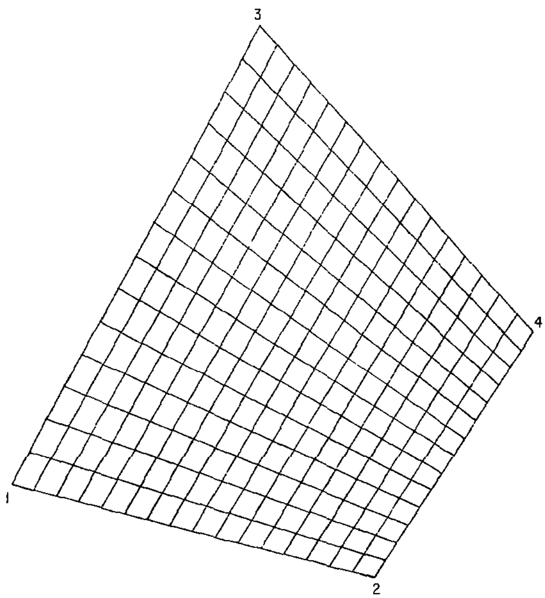

FIG. 2. Algebraic generation for the quadrilateral of Fig. 1.

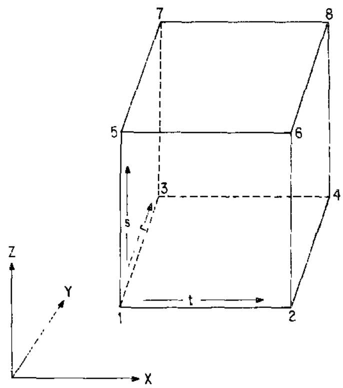

FIG. 3. A hexahedron region.

including P_1 and P_2. The construction just described is the simplest example of algebraic coordinate generation between two points.

For planar regions determined by four points, for example, a quadrilateral, the simplest construction is for a grid inside and on the boundaries of a quadrilateral determined by joining the points P_i $(i=1, 4)$ of coordinates (x_i, y_i) with straight lines, as illustrated in Fig. 1. In the same manner we use the parametric form of the straight line passing through the points P_1 and P_2, and P_3 and P_4, respectively. Using vector rather than component notation, they are given by

$$\mathbf{L}_{12}(t) = \mathbf{P}_1 + t(\mathbf{P}_2 - \mathbf{P}_1) \tag{8}$$

$$\mathbf{L}_{34}(t) = \mathbf{P}_3 + t(\mathbf{P}_4 - \mathbf{P}_3) \tag{9}$$

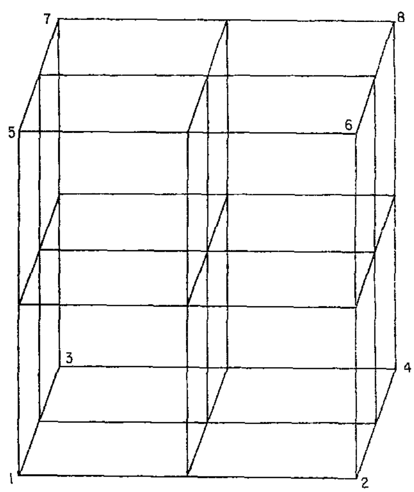

FIG. 4. Algebraic generation for the hexahedron of Fig. 3.

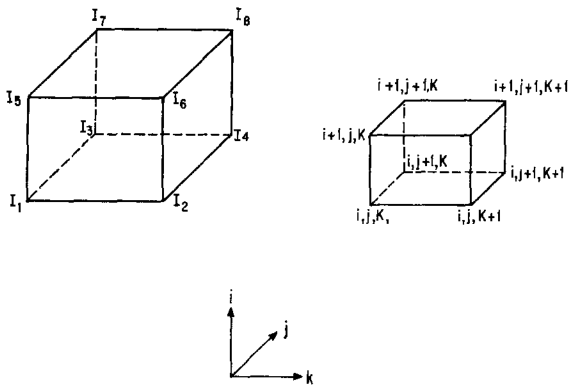

FIG. 5. Local element coordinates (left) and global region coordinates (right).

employing the same parametrization for both lines. Then with a parameter r varying from zero to one, the quadrilateral patch is analytically given in the form

$$\mathbf{L}_{1234}(r, t) = \mathbf{L}_{12}(t) + r[\mathbf{L}_{34}(t) - \mathbf{L}_{12}(t)]. \tag{10}$$

Equation (10) is known as the bilinear patch of the four points. Any point inside or on the quadrilateral is uniquely defined by the value of the parameters r and t. The two-dimensional grid generation inside and on the boundaries of the quadrilateral is constructed by the discrete form of Eq. (10),

$$\mathbf{L}_{1234}(r_j, t_k) = \mathbf{L}_{12}(t_k) + r_j[\mathbf{L}_{34}(t_k) - \mathbf{L}_{12}(t_k)], \tag{11}$$

where also $r_j = (j-1)/(n_r - 1)$ as j varies from one to n_r. With Eq. (11), a two-dimensional quadrilateral grid is displayed in Fig. 2, where $n_t = 17$ and $n_r = 15$.

In continuation from a quadrilateral to a hexahedron region, this is defined by

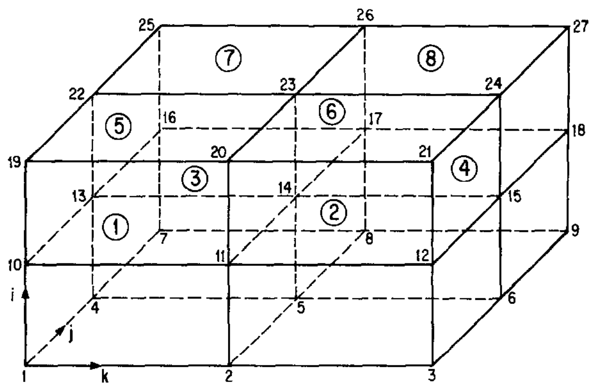

FIG. 6. Lexicographically ordered algebraic grid generation.

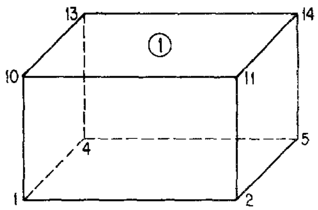

FIG. 7. Global node numbering for element 1.

eight points: $P_1, P_2, \ldots, P_8$, as illustrated in Fig. 3. With Eq. (10) and with a similar expression for the opposite face,

$$\mathbf{L}_{5678}(r, t) = \mathbf{L}_{56}(t) + r[\mathbf{L}_{78}(t) - \mathbf{L}_{56}(t)], \tag{12}$$

the unit cube in (s, r, t) is mapped into the general hexahedron by

$$\mathbf{L}\,\text{cube}(s, r, t) = \mathbf{L}_{1234}(r, t) + s[\mathbf{L}_{5678}(r, t) - \mathbf{L}_{1234}(r, t)]. \tag{13}$$

The grid is given by the discrete form of Eq. (13),

$$\mathbf{L}\,\text{cube}(s_i, r_j, t_k) = \mathbf{L}_{1234}(r_j, t_k) + s_i[\mathbf{L}_{5678}(r_j, t_k) - \mathbf{L}_{1234}(r_j, t_k)], \tag{14}$$

where we use $s_i = (i-1)/(n_s - 1)$ for $i = 1, 2, \ldots, n_s$ along with similar definitions for r_j and t_k. Shown in Fig. 4 is a perspective view of a three-dimensional grid for the hexahedron of Fig. 3 obtained with Eq. (14). For visual simplicity the same number of nodes have been used in the three directions $(n_t = n_s = n_r = 3)$.

For finite element analysis, it is necessary to perform a topological description of the grid generated. This permits the recovery from the algebraic coordinate generation of the subset of coordinates belonging to the nodes of a single cell. In finite element analysis this cell is called an element and the relation between an element and its nodes, nodal–element connectivity. In the implementation of the computational algorithm, the topological description of the grid is accomplished through the construction of a two-dimensional array. This array contains the global node ordering; its first subindex indicates the element number, and the second the

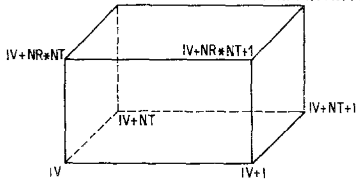

FIG. 8. Example of node–element connectivity.

local node number. Clearly this array relates global and local ordering, which is done following lexicographic ordering as illustrated in Fig. 5. The algebraic coordinate generation produces a lexicographically ordered grid automatically. This property of the method notably simplifies the element numbering.

An example illustrates the matter. Consider a cubic region in which an algebraic grid generation has been performed as shown in Fig. 6. The global node ordering is done in lexicographic order (left to right in the k direction, onwards in the j direction, and from bottom to top in the i direction). The numbers encircled indicate element numbers. The total number of nodes is $n_t * n_s * n_r = 27$ and the total number of elements $(n_t - 1) * (n_s - 1) * (n_r - 1) = 8$. For instance, the element 1 illustrated in Fig. 7 has the global node ordering 1, 2, 4, 5, 10, 11, 13, and 14, which corresponds to the local ordering 1, 2, 3, 4, 5, 6, 7, and 8. The face having the nodes 10, 11, 13, and 14 is the top face and that with nodes 1, 2, 4, and 5 is the bottom face. It is easily seen that, if we call the bottom left node of the bottom face IV, for a generic element e (see Fig. 8), the remaining nodes have the generic values shown in the figure. In lexicographic order the first node encountered is node IV, and its position is given by $\mathrm{IV} = k + n_t * (j-1) + n_t * n_r * (i-1)$. Therefore, once a global element number e is associated to node IV, the remaining seven nodes belonging to that element can be easily obtained from node IV as indicated in Fig. 8.

COLLAPSING TECHNIQUES

We introduce the collapsing technique, which generalizes the application of the algebraic coordinate generation. This is a reduction procedure by which a geometrical object is transformed into a different shape. For instance a quadrilateral can be reduced to a triangle if one of its vertices is collapsed into the midpoint of the diagonal determined by the adjacent vertices. This is illustrated in Fig. 9, where the triangle $P_1P_2P_3$ has been obtained from the quadrilateral $P_1P_2P_3P_4$ of Fig. 1. The point P_4 is called a pseudopoint because it serves for the definition of the pseudoquadrilateral; its coordinates are given by

$$\mathbf{P}_4 = 0.5(\mathbf{P}_2 + \mathbf{P}_3). \tag{15}$$

We can now define the equations of the straight line joining the points P_1P_2 and P_3P_4, respectively, by

$$\mathbf{L}_{12}(t) = \mathbf{P}_1 + t(\mathbf{P}_2 - \mathbf{P}_1) \tag{16}$$

$$\mathbf{L}_{34}(t) = \mathbf{P}_3 + t(\mathbf{P}_4 - \mathbf{P}_3). \tag{17}$$

Then, with a parameter r varying from 0 to 1, the pseudoquadrilateral or triangle patch is analytically given in the form

$$\mathbf{L}_{1234}(r, t) = \mathbf{L}_{12}(t) + r[\mathbf{L}_{34}(t) - \mathbf{L}_{12}(t)]. \tag{18}$$

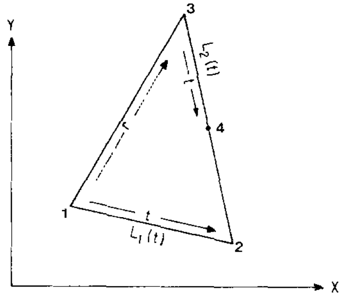

FIG. 9. The triangular region.

The grid is given by the discrete form

$$\mathbf{L}_{1234}(r_j, t_k) = \mathbf{L}_{12}(t_k) + r_j[\mathbf{L}_{34}(t_k) - \mathbf{L}_{12}(t_k)], \tag{19}$$

where r_j and t_k have been previously defined. Figure 10 shows a computer generated two-dimensional grid for the triangle of Fig. 9 ($n_t = n_r = 5$).

It is clear from the previous construction that the pseudopoint is a singular point. If the pseudopoint is not counted as a regular point, a nonuniform coordinate distribution on the pseudopoint side is obtained. However, if it is counted as a regular point, a uniform distribution is obtained provided that the distribution on the remaining sides is equal. Sometimes it might be necessary to use unequal dis-

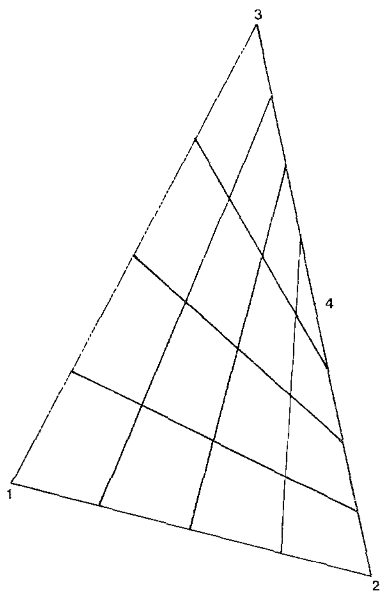

FIG. 10. A collapsed algebraic grid generation for the triangle of Fig. 9.

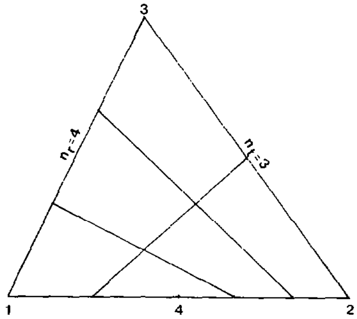

FIG. 11. Nonuniform distribution on the fake point side.

tribution of points in two of the sides of a triangle and therefore, uniform distribution in the pseudopoint side is lost. Consider the triangle of Fig. 11, which has been discretized with $n_r = 4$ and $n_t = 3$, and with the fake point at the midpoint of side P_1P_2. This results in a nonuniform distribution along the side P_1P_2 as shown in the figure. For obtaining uniform distribution the fake point has to be located in such a way that all the cells are equal in size. Thus, noticing that the total number of cells in the side P_1P_2 should be $(n_t - 1) + (n_r - 1)$, the cell size must be

$$\text{delta} = 1/(n_t - 1 + n_r - 1), \tag{20}$$

where we assumed the distance from P_1 to P_2 to be unity. Thus the pseudopoint P_4 can be obtained with the expression

$$\mathbf{P}_4 = [1 - (n_t - 1)\,\text{delta}]\,\mathbf{P}_1 + (n_t - 1)\,\text{delta}\,\mathbf{P}_2. \tag{21}$$

In Fig. 12, we carry out the above construction for the example of Fig. 11. For $n_r = 4$ and $n_t = 3$, delta $= \frac{1}{5}$. Supposing side P_1P_2 horizontal we obtain $x_4 = \frac{3}{5}x_1 + \frac{2}{5}x_2$. Let $x_1 = 0$ and $x_2 = 1$ (for example); then $x_4 = \frac{2}{5}$ as indicated in Fig. 12.

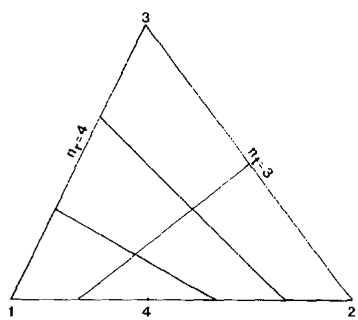

FIG. 12. Uniform distribution on the fake point side.

In continuation from a triangle to a pentahedron with triangular basis, this object can be easily constructed from the hexahedron by collapsing one of the edges into the plane joining the two adjacent edges. Once this simple procedure has been carried out, the grid generation for the hexahedron carries over for the pentahedron. For example, if in the hexahedron of Fig. 3, the edge P_4P_8 is collapsed into the plane passing through the edges P_3P_7 and P_2P_6, respectively, such that

$$\mathbf{P}_8 = 0.5(\mathbf{P}_6 + \mathbf{P}_7) \tag{22}$$

$$\mathbf{P}_4 = 0.5(\mathbf{P}_2 + \mathbf{P}_3), \tag{23}$$

a pentahedron is obtained. Figure 13 shows a perspective view of a computer generated three-dimensional grid using the technique just described $(n_t = n_s = n_r = 3)$. We note that the collapsing procedure produces a pseudoline (P_4P_8) which now is visible in the plotting.

Now we consider the pyramid grid generation obtained from the hexahedron of Fig. 3, for instance, by the following sequence of collapsing procedure (irrelevant of their order): (a) collapsing of the edge P_4P_8 into the plane joining the two adjacent edges; and (b) collapsing of the point P_5 into P_1, such that Eqs. (22) and (23) are satisfied together with

$$\mathbf{P}_5 = \mathbf{P}_1. \tag{24}$$

In Fig. 14 we present a perspective view of a computer generated three-dimensional grid of a pyramid using these techniques $(n_t = n_s = n_r = 3)$.

One alternative for obtaining a tetrahedron starts from the cube of Fig. 3 and follows the sequence of collapsing procedures: (a) collapsing of the point P_6 into

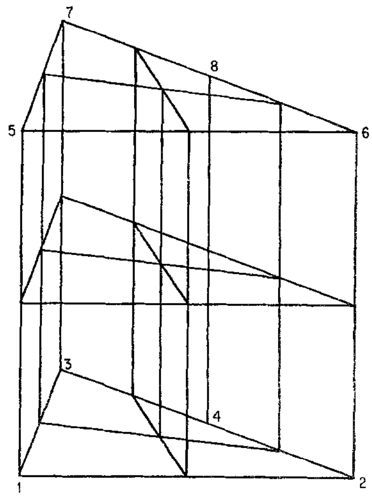

FIG. 13. A collapsed grid algebraic generation for a pentahedron with triangular basis.

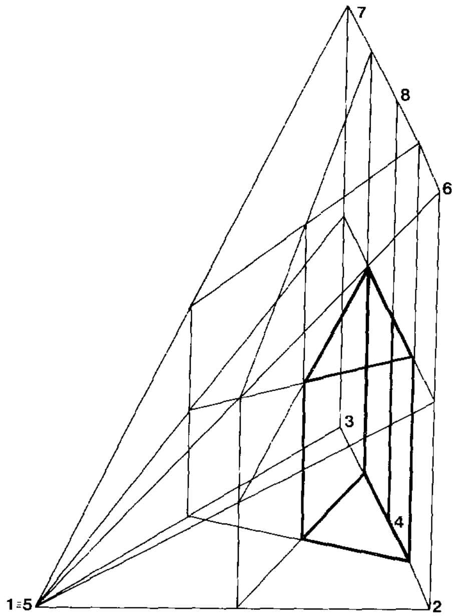

FIG. 14. A collapsed algebraic grid generation for a pyramid showing in thick lines one of the elements.

P_2; (b) collapsing of the edge P_4P_8 into the plane passing through the edge P_3P_7 and the point P_2; and (c) collapsing of the point P_5 into P_1, such that Eqs. (22), (23), and (24) are satisfied together with

$$\mathbf{P}_6 = \mathbf{P}_2. \tag{25}$$

In Fig. 15a we present a computer generated three-dimensional grid of a tetrahedron with $n_t = n_s = n_r = 3$.

A second alternative for obtaining a tetrahedron starting from the cube of Fig. 3 is to collapse, for instance, the points P_2, P_5, and P_8 into the lines determined by P_1P_4, P_1P_7, and P_4P_7, respectively, as shown in Fig. 15b. The collapsing is done according to the formulae

$$\mathbf{P}_2 = \mathbf{P}_1 + \alpha(\mathbf{P}_4 - \mathbf{P}_1) \tag{26}$$

$$\mathbf{P}_5 = \mathbf{P}_1 + \beta(\mathbf{P}_7 - \mathbf{P}_1) \tag{27}$$

$$\mathbf{P}_8 = \mathbf{P}_4 + \gamma(\mathbf{P}_7 - \mathbf{P}_4), \tag{28}$$

where $0 < \alpha,\ \beta,\ \gamma < 1$.

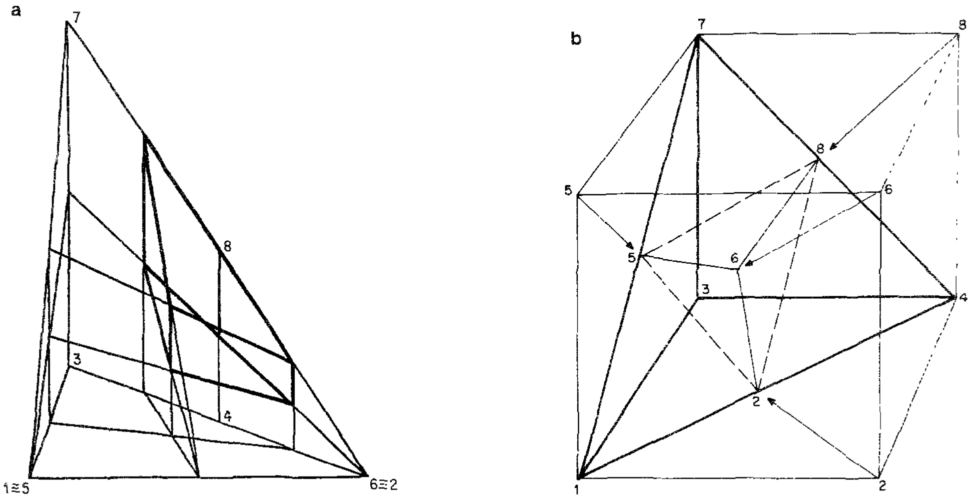

FIG. 15(a) A collapsed algebraic grid generation for a tetrahedron showing in thick lines one of the elements. (b) Second alternative for obtaining a tetrahedron from the cube of Fig. 3.

Next the point $\mathbf{P}_6$ is collapsed into the barycenter of the dotted triangle formed by the pseudonodes $\mathbf{P}_2$, $\mathbf{P}_5$, and $\mathbf{P}_8$ according to

$$\mathbf{P}_6 = \tfrac{1}{3}(\mathbf{P}_2 + \mathbf{P}_5 + \mathbf{P}_8). \tag{29}$$

For the particular case in which $\alpha = \beta = \gamma = \frac{1}{2}$, $\mathbf{P}_6$ becomes the barycenter of the triangle formed by $\mathbf{P}_1$, $\mathbf{P}_4$, and $\mathbf{P}_7$, i.e.,

$$\mathbf{P}_6 = \tfrac{1}{3}(\mathbf{P}_1 + \mathbf{P}_4 + \mathbf{P}_7). \tag{30}$$

In the event that there is unequal distribution of mesh points in any of the three parametric coordinates, and in order to obtain uniform distribution in the triangle $\mathbf{P}_1\mathbf{P}_4\mathbf{P}_7$, the location of the points on which to collapse the hexahedron can be determinated following the same technique as in the case of the triangle in the plane. Uniform distribution is then the main advantage of this second alternative in relation to the former.

The collapsing technique discussed above permits a grid generation with regularly lexicographic ordering. Hence, the procedure for the topological description of the grid, which was discussed in the previous section, can be readily extended to the objects studied in this section.

Application to a Topologically Complex Region

We present an application of the method of collapsed algebraic transformations to the discretization of a three-dimensional topologically complex region, i.e., a

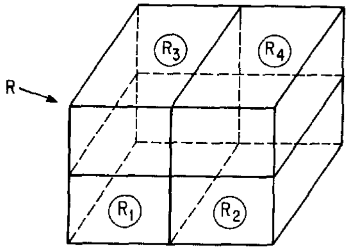

FIG. 16. A complex region.

region composed of an arbitrary number of three-dimensional objects. The final objective is the development of a method for an automatic grid generation in finite element analysis of realistic geological configurations. Suppose that a region R of a three-dimensional space is given by the unit cube and that this region is composed of four objects (four smaller cubes), each one having different physical properties (see Fig. 16). The four objects fill the space of the region R completely. The boundaries of each object are given as data. The problem consists in discretizing the four objects with a three-dimensional grid in such a way that the number of nodal points at the common interface between objects is matched. Once the grid generation for each object has been obtained independently, using the techniques of previous sections, a global assembling of all the object grids must be performed, after which the nodal-element connectivity and the element–object connectivity can be determined.

Before commencing the description of the procedure employed we enumerate the restrictions of the method proposed: (a) all the objects must be hexahedra, and (b) all the objects must have common edges.

Any object that is not a hexahedron (and has fewer than six faces) can be transformed into a hexahedron (or pseudohexahedron) by the manual addition of the appropriate geometrical components. Any object not having common edges with its neighbors can be brought into it by proper manual subdivision. Examples: (i) The region represented in Fig. 17 contains objects that fail to satisfy condition (b). This is true because the edge AB common to R_1 and R_2 and belonging to the bottom

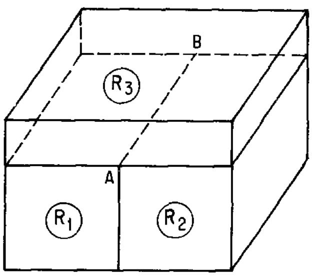

FIG. 17. A complex region not satisfying condition (b).

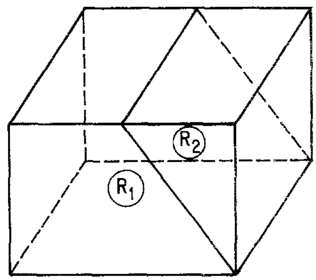

FIG. 18. A complex region not satisfying condition (a).

face of R_3 transforms the object R_3 into a figure with seven faces (the maximum number of faces recognized by the implementation of the method is six). Condition (b) can easily be satisfied if the object R_3 is in turn subdivided in two by the addition of a vertical plane containing the edge AB (for instance). (ii) The region represented in Fig. 18 contains objects that fail to satisfy condition (a). This is true because R_2 has only five faces and therefore is not a hexahedron. To fix this problem we add an extra edge (pseudoedge) parallel to the common edge between the two objects, thereby creating the sixth pseudoface needed in R_2 (note: we cannot add a pseudoface to R_2 in the common plane between R_1 and R_2 since this will fix condition (a) in R_2 but make R_1 fails to satisfy condition (b).

The foregoing considerations show that a manual preprocessing of the data is necessary before entering into the grid generation procedure. By now it is clear that the manual preprocessing is nothing else than a generalization of the collapsing technique previously discussed.

The new feature of this section is the appearance of a region composed of an arbitrary number of objects. For computational purposes it is assumed that these objects are ordered in lexicographic order. Therefore, once the grid generation is performed on a given object, the nodes of that object are ordered in a local (for that particular object) lexicographic order, which means that at the common interface between two objects the same node has different numbering. For finite element calculations it is necessary to have a global system numbering for the region in which every node is uniquely defined. To satisfy this requirement an appropriate assembling procedure is performed in which the local object lexicographically ordered grid is transformed into a global region lexicographically ordered grid. This is done through matrix manipulations assuming that the relative position of the objects inside the region is known; the three-dimensional array containing the coordinates of the nodes of a particular object grid is loaded into a global array in such a way that its relative position inside the region is preserved. Once this assembling procedure is done for all the objects we can easily recover the node–element connectivity using the same technique as for the case of a single object.

In order to recover the element–object connectivity a flag is assigned to each nodal point at the object level. The local object three-dimensional arrays containing

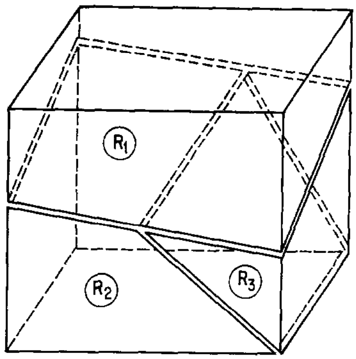

FIG. 19. Primitive region.

these flags are treated in the same fashion as the nodal points; i.e., they are assembled into a global region array from which the element–object connectivity is readily obtained.

We illustrate the method of collapsed algebraic transformations with some examples. Suppose that the region to be discretized is composed of three objects as depicted in Fig. 19. Here again we need to perform a manual preprocessing of the primitive objects in order to satisfy the restrictions imposed by the present implementation of the method. This is done in Fig. 20, where the dotted lines show the new geometrical components introduced: a subdivision of the object R_1 in two and the transformation of R_3 into a hexahedron by the addition of the pseudoedge 2–11 (note that the object node data ordering is arbitrary). In Figs. 21 and 22 we present the computed results obtained with the present method. Figure 21 shows a perspective view of the primitive region (with the preprocessing included) and Fig. 22 shows a perspective view of the grid generated ($n_s = n_t = n_r = 3$).

Finally, suppose that the primitive region shown in Fig. 23 has to be discretized.

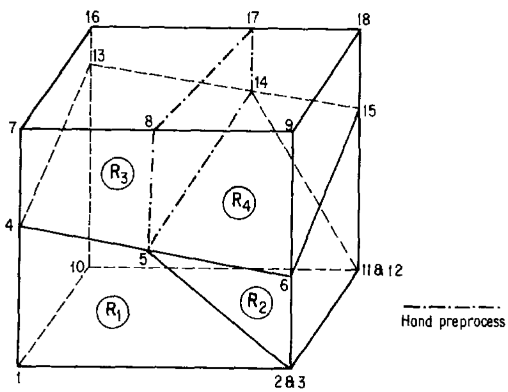

FIG. 20. Manual preprocessing of the region of Fig. 19.

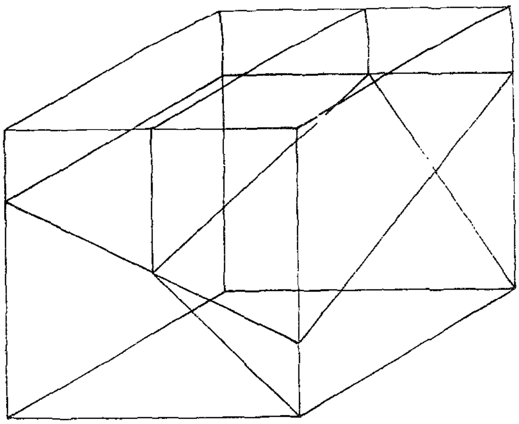

FIG. 21. Computed perspective view of the region of Fig. 20.

This figure pretends to represent a realistic geological configuration. The preprocessing of this problem consists, for instance, in a subdivision of R_2 by the addition of a new object (extension of R_3 into R_2) and the addition of a pseudoedge at the bottom of R_2. The result is shown in Fig. 24 (note again that the object node data ordering is arbitrary and takes advantage of the previous figure). In Figs. 25 and 26 we present the computed results. Figure 25 depicts a perspective view of the primitive region (preprocessing included) and Fig. 26 shows a perspective view of the grid generated ($n_t = n_s = n_r = 3$).

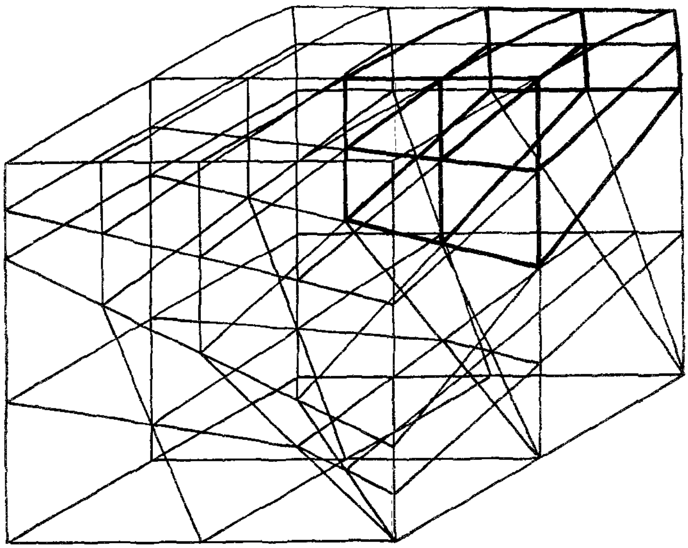

FIG. 22. A collapsed algebraic grid generation for the region of Fig. 21 showing in thick lines the four elements into which the back half of the top right corner object is subdivided.

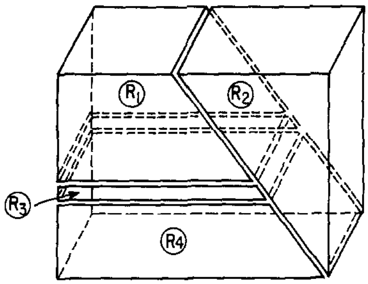

FIG. 23. An example of a geophysical region configuration.

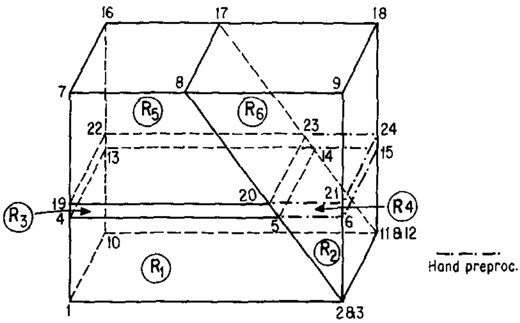

FIG. 24. Manual preprocessing of the region of Fig. 23.

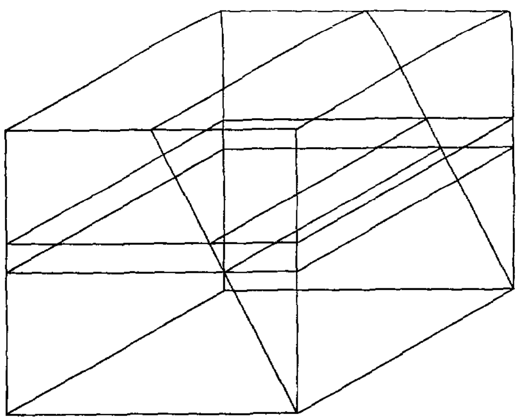

FIG. 25. Computed perspective view of the region of Fig. 24.

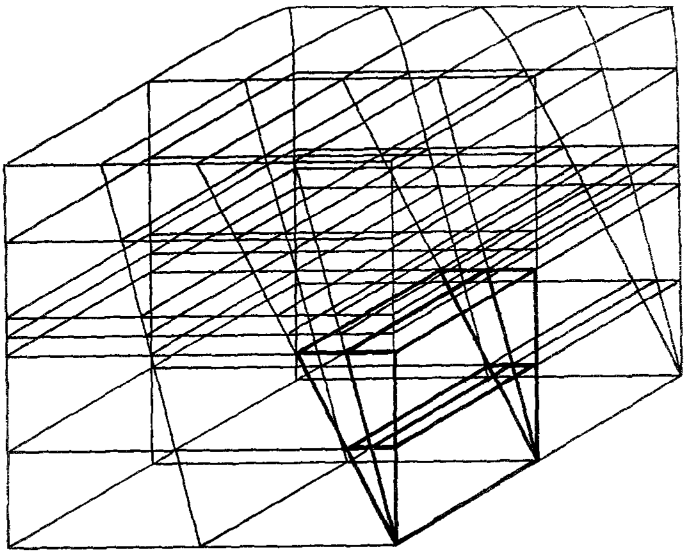

FIG. 26. A collapsed algebraic grid generation for the region of Fig. 25 showing in thick lines the four elements into which the front half of the bottom right corner object is subdivided.

From the description of the method and from the computed results it is clear that the grid uniformity is a direct consequence of the size uniformity of the objects. Therefore, to achieve this property it is necessary, when the objects are quite different in size, to perform an educated manual partition of the larger objects. Obviously, partition can also be used to work in the opposite direction, that is, to induce a grid densification on a particular object or in a part of it. This partition should not be confused with the manual preprocessing, a term that here is associated with the collapsing technique.

We would like to make a final comment on the collapsed algebraic transformations method and its use in finite element analysis. In the majority of standard finite element programs it is necessary to know beforehand the type of element and basis function to be used. This information is not directly provided by the collapsed algebraic transformations method. Clearly, in the element generated by this method, only data on its eight coordinate nodes are given, regardless of the geometric object represented. To overcome this difficulty one alternative is to perform a scanning of every element of the mesh generated, but this may be extremely time consuming. Another alternative is to construct general basis functions which do not require that information (this has been implemented in Marshall *et al.* [6]; see also Irons [7] for more details of this topic).

CONCLUSIONS

General collapsing techniques have been developed to generate regular lexicographically ordered grids within irregular objects by using algebraically defined transformations in both two and three dimensions. In two dimensions the collapsed objects are just triangles; in three dimensions, they comprise pyramids, tetrahedra, and prisms. In either dimension a uniform distribution of mesh cells is obtained by using the chosen number of mesh points in each direction as basic data to determine the locations on which to collapse the general pseudoquadrilateral or pseudohexahedron into the particular object.

While the basic technique is presented in terms of simple multilinear coordinate transformations, the results can be extended to more general algebraic transformations. Moreover, the established pattern in two and three dimensions can also be continued into even higher dimensions.

On application, the various objects are assembled to form a global discretization of a topologically complex region. The local lexicographically ordered object grid must then be appropriately matched at junctures with other object grids. A corresponding mesh numbering scheme was then suggested in order to increase the efficiency for applications.

The utilization of collapsed algebraic transformations allows a sufficient degree of generality to deal with various three-dimensional objects while maintaining a considerable amount of grid control. An educated manual partition of the objects permits a desirable degree of grid uniformity for the region. The numerical examples presented evidence of the simplicity, robustness, and efficiency of the method advocated.

ACKNOWLEDGMENTS

The authors wish to thank the referees of this paper for helpful comments which led to an improvement of its presentation and for bringing to their attention the work of B. M. Irons [7].

REFERENCES

1. W. A. COOK, *Int. J. Numer. Methods Eng.* **8** (1974), 27.
2. W. A. COOK, "Ingen: A general Purpose Mesh Generator for Finite Element Codes," Los Alamos Scientific Laboratory Report LA 7135 MS, March 1978.
3. W. J. GORDON AND C. A. HALL, *Int. J. Numer. Methods Eng.* **7** (1973), 461.
4. P. EISEMAN, *J. Comput. Phys.* **33** (1979), 118.
5. G. MARSHALL, P. EISEMAN, AND J. T. KUO, "Three-Dimensional Algebraic Mesh Generation," Project MIDAS Report, Aldridge Laboratory of Applied Geophysics, Columbia University, April 1983.
6. G. MARSHALL, J. VAN KAN, AND V. VISCONTI, General basis functions for the finite element method with collapsed algebraic mesh generation, presented at the Second National Meeting of Finite Element Users, Bariloche, July 2–6, 1984. [In Spanish]
7. B. M. IRONS, *Int. J. Numer. Methods Eng.* **8** (1974), 203.