Note

Efficient Volume Computation for Three-Dimensional Hexahedral Cells*

Currently, algorithms for computing the volume of hexahedral cells with "ruled" surfaces require a minimum of 122 FLOPs (floating point operations) per cell. A new algorithm is described which reduces the operation count to 57 FLOPs per cell. © 1988 Academic Press, Inc.

INTRODUCTION

Many three-dimensional codes are based on meshes whose cells are topologically equivalent to hexahedrons or "bricks." The cells are defined by eight vertices. The twelve cell edges are straight line segments. However, the six cell faces are not plane in general, and there are many choices possible for specifying the surface.

A popular choice [1–3] is based on a trilinear mapping between physical $\{x, y, z\}$ space and logical $\{\xi, \eta, \zeta\}$ space, as illustrated in Fig. 1. Using the vertex numbering convention shown in the figure, this mapping is expressed as

$$\begin{aligned}\mathbf{r} = (1-\zeta)[\xi\{\mathbf{r}_1(1-\eta)+\mathbf{r}_2\eta\}+(1-\xi)\{\mathbf{r}_3\eta+\mathbf{r}_4(1-\eta)\}]\\ +\zeta[\xi\{\mathbf{r}_5(1-\eta)+\mathbf{r}_6\eta\}+(1-\xi)\{\mathbf{r}_7\eta+\mathbf{r}_8(1-\eta)\}],\end{aligned} \tag{1}$$

where $\mathbf{r}_i = \{x_i, y_i, z_i\}$, are the physical coordinates of vertex i, and $\{\xi, \eta, \zeta\}$ are logical space coordinates such that $0 \leqslant \xi, \eta, \zeta \leqslant 1$. For a typical cell face, such as that given by $\zeta = 0$, this reduces to

$$\mathbf{r} = \xi\{\mathbf{r}_1(1-\eta)+\mathbf{r}_2\eta\}+(1-\xi)\{\mathbf{r}_3\eta+\mathbf{r}_4(1-\eta)\}. \tag{2}$$

This is a bilinear mapping of the surface to the ξ, η plane, which amounts to defining a unique "ruled" surface among the four defining vertex points. In finite element terminology, this is the geometry obtained from the use of a linear isoparametric conforming element.

The computation of the volume of a cell is fundamental for any code employing such a mesh. Further, some hydrodynamics codes compute the volume swept out

* The U.S. Government's right to retain a nonexclusive royalty-free license in and to the copyright covering this paper, for governmental purposes is acknowledged. This work was performed under the auspices of the U.S. Department of Energy.

0021-9991/88 $3.00

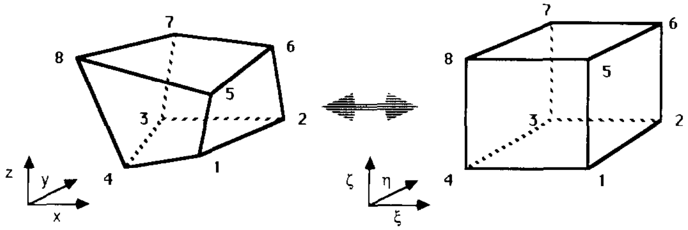

FIG. 1. Relationship of a computational cell in physical $\{x, y, z\}$ space to the unit cube in logical $\{\xi, \eta, \zeta\}$ space.

by a cell face during a time step. Such volumes are topologically equivalent to the cell volume, and they may be computed using the same algorithm.

The standard computation of the volume involves the straightforward evaluation of the integral

$$V=\int_0^1\int_0^1\int_0^1 \frac{\partial(x, y, z)}{\partial(\xi, \eta, \zeta)}\, d\xi\, d\eta\, d\zeta \tag{3}$$

for each cell. The operation count for this type of algorithm is a minimum of 122 FLOPs per cell [3–5].

The New Algorithm

We make use of the identity $\nabla \cdot \mathbf{r} = 3$ and the divergence theorem to write the cell volume as

$$V=\frac{1}{3}\iint \mathbf{r}\cdot\mathbf{n}\, dA, \tag{4}$$

where $\mathbf{n}$ is the unit outward normal vector, and the integration takes place over the surface of the cell. We have thus converted the computation of cell volume to a summation of contributions from cell faces. Since each cell face is shared by two cells, the primary saving of the new algorithm comes from the double use of each cell face contribution. This step is general and is useful for the computation of the volume of arbitrary cells.

Let us consider the bottom face ($\zeta = 0$) of the cell in Fig. 1 as typical. The unit normal will point in the negative ζ-direction; therefore

$$\mathbf{n}\, dA=\left[\frac{\partial \mathbf{r}}{\partial \eta}\times\frac{\partial \mathbf{r}}{\partial \xi}\right] d\xi\, d\eta. \tag{5}$$

Substituting (2) and (5) into (4), and evaluating the resulting surface integral, we obtain the fundamental expression for the contribution of this face as

$$S_{1234} = \tfrac{1}{12}\,[\mathbf{r}_1 \cdot (\mathbf{r}_4 \times \mathbf{r}_3) + \mathbf{r}_2 \cdot (\mathbf{r}_1 \times \mathbf{r}_4) + \mathbf{r}_3 \cdot (\mathbf{r}_2 \times \mathbf{r}_1) + \mathbf{r}_4 \cdot (\mathbf{r}_3 \times \mathbf{r}_2)]. \tag{6}$$

Notice that this is cyclically symmetric with respect to the four vertex points. It is not in the most useful form for computation, however. Equation (6) can be rewritten in a great many alternative forms. The most useful are

$$\begin{aligned} S_{1234} &= \tfrac{1}{12}(\mathbf{r}_2 + \mathbf{r}_3) \cdot [(\mathbf{r}_1 + \mathbf{r}_2) \times (\mathbf{r}_3 + \mathbf{r}_4)], \\ &= \tfrac{1}{12}(\mathbf{r}_1 + \mathbf{r}_4) \cdot [(\mathbf{r}_3 + \mathbf{r}_4) \times (\mathbf{r}_1 + \mathbf{r}_2)], \\ &= \tfrac{1}{12}(\mathbf{r}_1 + \mathbf{r}_2) \cdot [(\mathbf{r}_1 + \mathbf{r}_4) \times (\mathbf{r}_2 + \mathbf{r}_3)], \\ &= \tfrac{1}{12}(\mathbf{r}_3 + \mathbf{r}_4) \cdot [(\mathbf{r}_2 + \mathbf{r}_3) \times (\mathbf{r}_1 + \mathbf{r}_4)], \end{aligned} \tag{7}$$

or any of the cyclic permutations obtained by use of the identity

$$\mathbf{a} \cdot (\mathbf{b} \times \mathbf{c}) = \mathbf{c} \cdot (\mathbf{a} \times \mathbf{b}) = \mathbf{b} \cdot (\mathbf{c} \times \mathbf{a}).$$

Examination of Eqs. (7) should make it clear how to form the corresponding expressions for any face, with due regard for the appropriate sense of the surface normal.

The computation of the cell volume may be organized into three steps:

(a) compute and store the vector sums of the two vertices of each cell edge (9 FLOPS per cell),

(b) compute and store the vector dot-cross product of Eq. (7) for each cell face (42 FLOPs per cell),

(c) sum the contributions of the six faces for each cell and divide by 12 (6 FLOPs per cell).

The resulting operation count is 57 FLOPs per cell. The above algorithm assumes the availability of sufficient temporary working storage.

The situation is somewhat different for computing the volumes swept out by cell faces since these volumes do not form a three-dimensional mesh. They do, however, form two-dimensional meshes in the planes associated with integer values of the logical coordinates. Therefore, approximately two-thirds of the savings of the new algorithm are retained. Further, since one face of such a volume is the same as the original cell face, and if the contribution of this face had been saved, then essentially all the saving of the new algorithm may be obtained.

Acknowledgment

I thank John Baumgardner for putting this algorithm into practice in his code and for making valuable suggestions.

References

1. W. E. Pracht, *J. Comput. Phys.* **17**, 132 (1975).
2. J. U. Brackbill, *Methods Comput. Phys.* **16**, 1 (1976).
3. W. E. Pracht and J. U. Brackbill, Los Alamos Scientific Laboratory Report LA-6342, 1976 (unpublished).
4. A. A. Amsden and H. M. Ruppel, Los Alamos National Laboratory Report LA-8905, 1981 (unpublished).
5. A. A. Amsden, J. D. Ramshaw, P. J. O'Rourke, and J. K. Dukowicz, Los Alamos National Laboratory Report LA-10245-MS, 1985 (unpublished).

Received: January 30, 1987

John K. Dukowicz

Theoretical Division,
Group T-3 Los Alamos National Laboratory,
Los Alamos, New Mexico 87545