GENERAL FINITE ELEMENT METHODS AND GRAPHICAL DISPLAY OF CFD SOLUTIONS

A. N. Stokes[1] and M. C. Thompson[2]

[1]CSIRO Division of Mathematics and Statistics,
Private Bag 10, Clayton, Victoria 3168 AUSTRALIA

[2]CSIRO Division of Building, Construction and Engineering,
P.O. Box 26, Highett, Victoria 3190 AUSTRALIA

INTRODUCTION

In our CFD work, we have developed a finite element based suite of programs which can rapidly be deployed to solve problems in two dimensions on quite complicated geometrical arrangements. The method is based on the adoption of a standard convention for describing unstructured grids. The associated data structures are useful also for generating computer graphical presentations of the solutions.

UNSTRUCTURED GRIDS

Automatically generated grids for computational work are often generated by using a mapping between the complex domain of interest and a simple domain such as a unit square. This has advantages in efficiency and predictability but often requires some ingenuity by the user, some fudging at difficult boundaries, and often fails badly in multiply connected regions. We have adopted the more versatile alternative of an unstructured mesh of triangles, created by repeated subdivision. This has the disadvantage of being more difficult to program, but once this has been done, a mesh can be generated easily with only a small amount of user input (which essentially consists of a listing of the boundary lines). This type of mesh generation procedure allows control of local mesh density.

The difficulties associated with using unstructured meshes for problems like particle tracking, dispersion calculations, and time-dependent calculations with moving boundaries, is the subject of this talk. The essence is that an efficient algorithm is needed that can locate a point in an element of an unstructured mesh.

ELEMENT SEARCHING AND GRAPHICAL DISPLAY

In dealing with time-varying flows, or even steady flows, the ability to produce something like a smoke or bubble visualisation is very useful. Even on a stationary grid, this requires calculation of a velocity at any point in the domain, and the tracking of particles from one element to the next. Although this sounds fairly straightforward, there are a few difficulties:
(i) the handling of collisions with obstacles tends to introduce bias, leading to a perceived tendency to repel particles, and
(ii) inevitable errors, particularly where flow direction is changing rapidly, lead occasionally to contradictory directions at element interfaces.
With care these problems can be overcome.

When the geometry is time-dependent so that the boundaries are not stationary, the more difficult problem of a time-varying mesh must be overcome. We do not attempt to distort meshes to follow the moving boundaries, but rather produce a new mesh at each timestep. Consequently, there is no obvious mapping from the elements of one mesh to that of another. Each time that a new mesh is produced, any particles that are being tracked must be located within it.

Searching a mesh for the element that contains a point prescribed by geometrical coordinates is nontrivial, and may be very time-consuming. Fortunately there is some structure that can be imposed on a mesh by the passage of a moving front. This is sometimes used to actually perform the matrix inversion; we have found that equivalent results can be obtained by using the front to number the nodes, and then using a conventional band matrix solver. But the structure provided by the front then comes in handy for relating elements from different meshes. An association vector between these elements is set up, and from the passage of the front, adjacency information is also recorded. In this way, if a particle being tracked is not in the associated element of the new mesh, the element which does contain it can be found after a brief search.

Modern graphics terminals make it possible to represent continuous variables in many attractive ways. A field defined on a finite element mesh can be easily represented as a colour-coded image using Gouraud shading algorithms provided on advanced graphics computers. A continuous representation of time-varying dispersion is then possible.

MODELLING DISPERSION PROCESSES

Numerical instability can be a problem with modelling convection diffusion processes, especially with rapidly varying velocity fields. We have developed very stable mesh-based algorithms to convect and disperse material within the mesh topology in such a way that negative concentrations are rendered impossible.

CONCLUSION

As more computing power becomes available, and software accumulates, the generality of unstructured mesh methods makes them attractive in comparison with structured alternatives, not only for solving problems in CFD and partial differential equations generally, but also for graphical postprocessing.