FLAG: A FREE-LAGRANGE METHOD FOR NUMERICALLY SIMULATING HYDRODYNAMIC FLOWS IN TWO DIMENSIONS

W. P. Crowley

Lawrence Radiation Laboratory, University of California
Livermore, California

INTRODUCTION

Traditionally, time-dependent hydrodynamic motions are numerically simulated in two ways. The Lagrangian formulation is used for relatively smooth flows, while the Eulerian description must be used for flows involving violent motions. We consider here a quasi-Lagrangian approach which eliminates the mesh constraints traditionally associated with the Lagrangian method.

Mesh scrambling in traditional Lagrangian calculations happens because mesh connections are permanent entities. If logically connected fluid particles move substantial distances apart during a calculation, the resulting mesh distortions reduce the integration time step and decrease the accuracy of the solution.

The basic advantage in the Free-Lagrange (FLAG) method is that mesh points are not tied together for the duration of the calculation. As each cycle begins the mesh is optimized by an algorithm which links "nearest" neighbors together. Thus points are free to drift apart without the usual attendant mesh scrambling.

The FLAG code consists of three major parts: (1) Mesh-maker, (2) mesh optimization, (3) equations of motion. Each part will be described separately.

The FLAG concept applies to both compressible and incompressible fluid motions. We consider an incompressible, inviscid form of the equations of motion with streamfunction and vorticity as dependent variables. Incompressible, inviscid calculations give qualitative verification of a vorticity cascade toward higher wavenumbers and an energy cascade into lower wavenumbers currently postulated for two-dimensional hydrodynamic motions.

MESH-MAKER

The problem faced by the mesh-maker is interconnecting a random two-dimensional distribution of points to form a single-valued mesh. Single-valued means; if point i is connected to point j, then j is connected to i. A simple way to do this is to construct a mesh with triangular mesh elements.

The mesh-maker proceeds in two stages: (1) the points are connected to form concentric convex rings (this essentially orders the points); (2) the points on adjacent rings are interconnected to form triangular mesh elements. After step (2), each point is connected to several others and all non-boundary points are surrounded by triangles.

Convex rings are constructed as follows: let S be the set of all points, and let \( \hat{x} \) be the point of minimum \( y \), i.e., for all \( y_i \in S, y_i \geq \hat{y} \). Take \( X_1 = \hat{x} \) as the first member of the first (outer) ring \( R_1 \). (The nth point on the ith ring is denoted \( X_{1}^{n} \).) To find the second point on \( R_1 \), construct the horizontal vector \( V_0^1 \) originating at \( X_1 \) and terminating at \( X_1 + (1,0) \). Next consider all other vectors \( V_{i1} \) originating at \( X_1 \) and terminating at \( X_i \in S \). Choose for the second member of \( R_1 \) (viz. \( X_2^1 \)) the one minimizing the angle between \( V_0^1 \) and \( V_{i1} \). Remove point 2 from S and place it in \( R_1 \); define the vector \( V_{21} = X_2^1 - X_1^1 \). Choose the third member of \( R_1 \) by

*Work performed under the auspices of the U. S. Atomic Energy Commission.*
minimizing the angle between $V^{21}$ and the vectors originating at $X^2_1$ and terminating at all remaining $X^1_1 \in S$. Continue until the point chosen is $\hat{X}$ then remove $X^1_1$ from $S$. The outer ring is now closed. The points along this first ring are the boundary points for the region. The members of the second ring $R_2$ are chosen similarly from the set $S - R_1$; those of the third ring are chosen from $S - R_1 - R_2$ and so on. Finally all points originally in $S$ belong to some ring $R_m$; they are ordered since we know both which ring they are on and the spatial relation of one ring to another. Further, the ordering is counterclockwise on each ring, and each point has two neighbors.

The second stage of the mesh-maker interconnects the rings. We specifically consider connecting the two outer rings; the others are connected similarly. To start, connect $X^1_1$ to $X^2_1$; this gives three logical sides of a quadrilateral (Fig. 1). To form a triangle choose one of the diagonals $X^2_2 X^1_1$ or $X^1_1 X^2_2$. If both diagonals are interior to the quadrilateral, choose the shorter; otherwise choose the interior one.

![Figure 1](image1)
![Figure 2](image2)
![Figure 3](image3)

Suppose it is $X^2_1 X^1_2$. Move along $R_1$ and $R_2$ counterclockwise (ignoring $X^1_1$ for the moment) and consider the quadrilateral with vertices $X^2_1, X^2_2, X^1_2, X^2_2$ (Fig. 2). Again choose the shortest interior diagonal, and calculate through $R_1$ and $R_2$. Finally the end points are connected; the region between $R_1$ and $R_2$ is covered by a mesh of triangles. Use the same procedure to connect the members of $R_2$ and $R_3$ starting with $X^1_2 X^2_3$. Finally the region enclosed by $R_1$ is covered by triangles and the mesh is complete.

Each point now has a set of points as neighbors and the only remaining job is logically ordering the points in a manner consistent with their physical ordering.

At this point we abandon all information about the ring structure and go from global to local knowledge, retaining at each point only information about the neighbors of that point. In Fig. 3 for example, we see that point 3 has as neighbors only points 1, 2, 6, 7, and 9. The "order" routine orders these points counterclockwise. It is used both in the final stage of the mesh-maker and in mesh-optimization (described below).

The order routine proceeds as follows. Let a subscript $i$ label a point and a subscript $k$ label its neighbors in some given order. The components of the vector $W_k$ are given by $W_k: (x_k - x_i, y_k - y_i)$ and we associate an angle $\theta_k$ with each vector $\cos \theta_k = \frac{W_k \cdot W_1}{||W_k|| ||W_1||}, k > 1$, namely the angle between $W_k$ and $W_1$. To eliminate using a square root, we define two functions