COMPUTING STREAM FUNCTIONS ON ARBITRARY MESHES

David A.J. Knight and Gordon D. Mallinson
Department of Mechanical Engineering
The University of Auckland
Auckland
New Zealand

ABSTRACT

A streamline is defined as a line that is everywhere tangential to the local fluid velocity. In two dimensions, a streamline can be defined as an iso-line of a scalar stream function, and in three dimensions, as the intersection of the iso-surfaces of two stream functions. This paper describes algorithms for computing the stream functions from solenoidal vector field data arranged on an arbitrary mesh.

INTRODUCTION

Streamlines

One of the most fundamental methods of visualising CFD data is the construction of streamlines, which are integral solutions to the equation:

$$\frac{dx}{u} = \frac{dy}{v} = \frac{dz}{w}. \tag{1}$$

Usually the streamline is computed by numerical integration of the ordinary differential equation

$$\frac{d\boldsymbol{x}}{dt} = \boldsymbol{u}(\boldsymbol{x}) \tag{2}$$

where $\boldsymbol{u}$ is the velocity at a position $\boldsymbol{x}$. An analysis of particle path integration algorithms can be found in [Darmofal and Haimes, 1995]. It has been shown [Mallinson, 1988], that, in order to construct accurate and realistic streamlines, mass conservation should be upheld:

$$\nabla.\rho\boldsymbol{u} = 0. \tag{3}$$

Application of Gauss' divergence theorem results in the condition that the total flux through a cell is zero:

$$\oint \rho\boldsymbol{u}.\boldsymbol{n} = 0. \tag{4}$$

We assume that this condition holds for all mass conservation cells in the mesh, which may or may not correspond to the computational cells, depending on the discretisation. Typically, the computational mesh consists of quadrilateral or triangular cells in two dimensions, or hexahedral or tetrahedral cells in three dimensions. However, the algorithms presented in this paper work for conservation cells which can be arbitrarily shaped polygons or polyhedra, provided that they are convex. The particular motivation for this work comes from recent developments in unstructured finite volume techniques which use the Voronoi diagram as the conservation cells [Were and Mallinson, 1995]. In this case, values of mass-conservative flux can be computed through the edges of the Voronoi digram from the velocity and pressure held at the nodes of the computational cells (tetrahedra). These velocities are not mass-conservative.

If mass conservative flux is not available, and computing the flux from nodal velocities does not provide a solenoidal flux field, then it is assumed that the velocities represent some underlying solenoidal (and therefore physical) field and the field is smoothed until it is solenoidal. However, most flow solution methods will produce solenoidal flux fields.

STREAM FUNCTIONS IN TWO DIMENSIONS

The Single Stream Function

If we take the two-dimensional form of the divergence equation (3), it follows that $\rho u\delta y - \rho v\delta x$ is an exact differential, equal to $\delta\psi$ for example. Then

$$\rho u = \frac{\partial\psi}{\partial y} \text{ and } \rho v = -\frac{\partial\psi}{\partial x} \tag{5}$$

so that ψ is constant along the streamline. This is an important result, as it means that streamlines can be constructed by computing the iso-lines of the stream function ψ.

The stream function also relates to the mass flow between two points in the flow field:

$$\dot{m}_{12} = \int_1^2 (\rho\boldsymbol{u}.\boldsymbol{n})dA \tag{6}$$

$$= \int_1^2 d\psi = (\psi_2 - \psi_1) \qquad (7)$$

as shown in Fig.1.

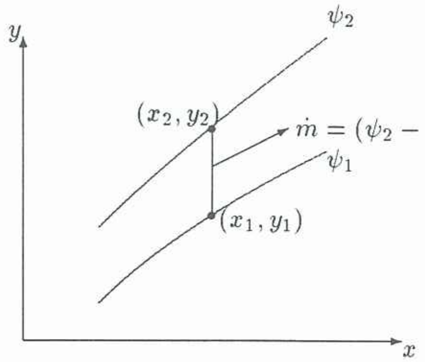

Figure 1: THE DISCHARGE BETWEEN TWO POINTS EXPRESSED AS A DIFFERENCE OF STREAM FUNCTION

Computing the Stream Function

If the stream function is not available from the output of the flow solution, it can be be calculated from the values of flux across the edges of the conservation cells. This is done using a recursive algorithm which starts by arbitrarily setting the stream function at one node to zero and traversing the edge lists. Providing the field is solenoidal, the value of stream function at a node is independent of the path Γ taken to reach that node:

$$\oint_{\delta\Gamma} = \oint_{\delta\Gamma} \rho u.dx - \rho v.dy \qquad (8)$$

$$= \int_{\Gamma} \nabla(\psi, 0) dx dy = 0 \quad \forall\Gamma \qquad (9)$$

from the definition of the stream function [Woodgate, 1992].

The algorithm is highly efficient and accurate as only one addition is performed for each edge traversed. In order to compute streamlines, the value of stream function is interpolated from the nodes using a multiquadric function defined by:

$$\psi(\boldsymbol{x}) = \sum_{i=1,n} \alpha_i \sqrt{(\boldsymbol{x} - \boldsymbol{x}_i)^2 + r^2} \qquad (10)$$

where the multiquadric constant, r, is related to the distance between data points [Carlson and Foley, 1991]. The streamline is computed by finding the contour of the stream function. The coefficients α_i are found from the interpolation conditions $\psi(\boldsymbol{x}_i) = \psi_i$. Fig.2 shows contours of stream function in flow around a rotating cylinder. The stream function is defined on the Voronoi diagram of the triangular grid, which is used as the conservation mesh [Were and Mallinson, 1995].

STREAM FUNCTIONS IN THREE DIMENSIONS

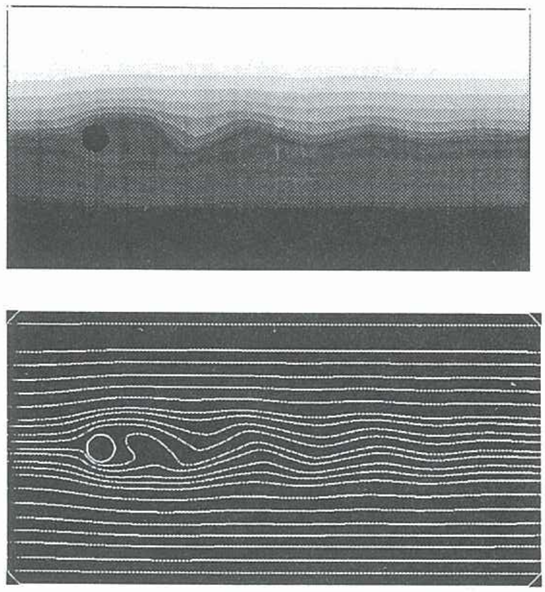

Figure 2: CONTOURS OF STREAM FUNCTION OF FLOW AROUND A ROTATING CYLINDER AND ASSOCIATED STREAMLINES

Dual Stream Functions

The problem becomes considerably more complex in three dimensions. If two independent *stream surfaces* can be computed, which are integral solutions to (1), then their intersection will be a streamline. The surfaces can be represented by functions $f = f(\boldsymbol{x})$ and $g = g(\boldsymbol{x})$. For a steady, compressible flow, the momentum is related to these functions by [Yih, 1957]:

$$\rho \boldsymbol{u} = \nabla f \times \nabla g \qquad (11)$$

from which we can see that the divergence vanishes:

$$\begin{aligned} \nabla.(\nabla f \times \nabla g) &= \nabla g.(\nabla \times \nabla f) \\ &- \nabla f.(\nabla \times \nabla g) = 0 \end{aligned}$$

The dual stream functions are also related to the mass flux through an area:

$$\dot{m} = \oint \rho \boldsymbol{u}.\boldsymbol{n} = (f_2 - f_1)(g_2 - g_1) \qquad (12)$$

which is equivalent to (6).

Solution Methods

Computing the dual stream functions has been considered for hexahedral meshes [Kenwright and Mallinson, 1992]. A different approach which relies on the stream functions being orthogonal, ie $\nabla f.\nabla g = 0$, has also been considered [Beale, 1993].

For a general conservation cell, the values of both stream functions at all of the nodes must be computed. The easiest way of doing this is to construct an fg diagram by plotting the values of the stream functions against each other at the nodes of the cell. An example fg diagram for a tetrahedral cell is shown in Fig.3. Note that both stream functions along a streamline are constant and streamlines are therefore reduced to points on the fg diagram.

There are two approaches to computing the stream functions: a global method and a local method. Both algorithms for computing stream functions on tetrahedral meshes are outlined below; the hexahedral case can be found in [Kenwright, 1992].

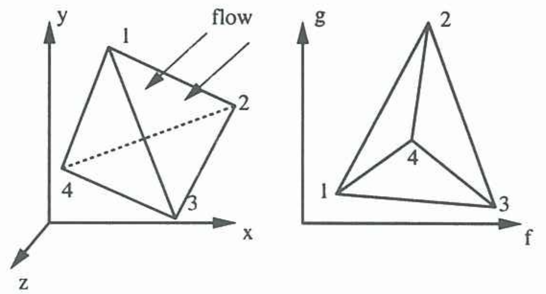

Figure 3: THE TRANSLATION FROM CARTESIAN TO fg SPACE FOR A GENERAL TETRAHEDRON.

Whole Field Solution Method Given the values of the dual stream functions at three nodes of a tetrahedron, the values at the fourth can be computed easily from the mass flux data. This is because the fourth node can be seen as a barycentric combination of the other nodes; the barycentric coordinates being computed from the relative fluxes through the faces of the tetrahedron:

$$f_4 = -\frac{\dot{m}_1}{\dot{m}_4} f_1 - \frac{\dot{m}_2}{\dot{m}_4} f_2 - \frac{\dot{m}_3}{\dot{m}_4} f_3 \qquad (13)$$

$$g_4 = -\frac{\dot{m}_1}{\dot{m}_4} g_1 - \frac{\dot{m}_2}{\dot{m}_4} g_2 - \frac{\dot{m}_3}{\dot{m}_4} g_3 \qquad (14)$$

provided $\dot{m}_4 \neq 0$. If the flux through the face corresponding to the unknown node is zero, a solution for both stream functions cannot be found, and the tetrahedron is skipped. It is usually possible to find the unknown stream functions from the fg diagrams of neighbouring tetrahedra. We can then travel through the mesh in a recursive fashion, computing the dual stream functions as we go.

This approach fails should either of the stream functions become multi-valued, as in areas of recirculating or spiralling flow. The stream surfaces can be visualised by constructing iso-surfaces, and streamlines obtained by calculating the intersection of iso-surfaces of both stream functions, as seen in Fig.4.

Local Solution Method If a whole field solution cannot be found, then a streamline can be computed by tracking through the mesh, computing the dual

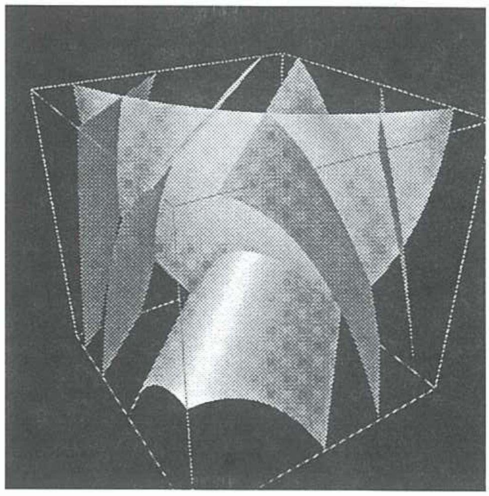

Figure 4: A FAMILY OF DUAL STREAM FUNCTION SURFACES. THE INTERSECTION OF THE STREAM SURFACES HERE FORM STREAMLINES.

stream functions for cells as and when they are needed. The algorithm proceeds as follows:

1. For a given start point, find the cell that contains that point.
2. Construct the fg diagram for that cell.
3. From the fg diagram, find the entry and exit faces for the streamlines.
4. Go the the neighbouring cell and repeat.

The fg diagrams can be constructed using one of the four *normalised fg* diagrams shown in Fig.5, depending on the number of inflow, outflow, and no-flow faces in the tetrahedron. Case (a) is used when there are two inflow and two outflow faces; case (b) when there are three inflow and one outflow faces (or vice versa); case (c) when there is one no-flow face, and case (d) when there are two no-flow faces. Finding the inlet and exit faces can be done by computing the barycentric coordinates of the streamline with respect to the nodes of each face, which involves inverting a three by three matrix for each face tested. Only faces known to be outflow faces need be tested. This compares favourably with a numerical integration scheme which requires the inversion of a four by four matrix for each cell the streamline visits. The algorithm terminates when the streamline reaches a boundary or reaches a face already visited. This prevents the streamline circulating forever in a re-circulation zone.

The streamlines are rendered by connecting the inlet and outlet points of each tetrahedron with a straight line. A smoother streamline can be created by then passing an interpolating spline through the points of the streamline. Streamlines computed using this technique can be seen in Fig.6. The flow is through a ventricular assist device [Were and

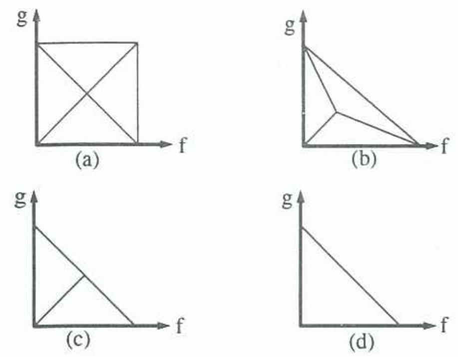

Figure 5: NORMALISED FG DIAGRAMS.

Mallinson, 1995]. In this case the complex and vortical nature of the flow prevents a whole field solution being obtained. If a numerical integration scheme is used, however, not all streamlines from the starting point reach the outlet, due to discrepancies in mass conservation.

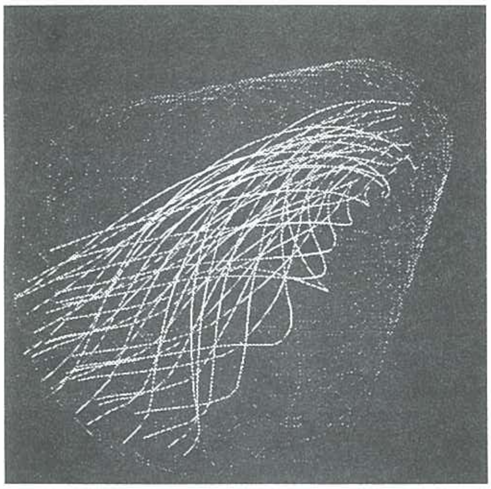

Figure 6: STREAMLINES IN FLOW THROUGH A ROTATING PIPE.

General Cells The corresponding algorithms for hexahedral meshes are more complicated as there are more unknowns, and extra equations involving mass flow gradients must be used. Another drawback in the hexahedral case is the possibility of concave and non-simple (self-intersecting) areas on the *fg* diagram. Extending the method to general polyhedra (such as Voronoi cells) is theoretically possible but would be very complicated in practice, as many more equations would have to be found. The best approach is then to decompose the conservation polyhedra into tetrahedra. This can be done using a Delauney algorithm, taking care to match faces on adjacent cells. The mass flux through internal faces can be calculated from mass conservation.

CONCLUSION

This paper demonstrates how stream functions can be computed from two and three dimensional solenoidal vector field data. The global solution algorithm provides a means of visualising the whole vector field, and constructing stream surfaces and streamlines provided a solution can be found. However, currently solutions can only be obtained for simple irrotational flows. Work is in progress in extending the global stream function algorithm to more complex flows.

The local method allows the construction of streamlines in almost all flow problems, and is faster than standard numerical integration techniques. Both methods have the advantage of being inherently mass conservative.

REFERENCES

Beale, S.B., 1993. "A numerical scheme for the generation of streamlines in three dimensions", In *Proceedings of CFD93*, pages 289–300. The CFD Society of Canada.

Carlson, R.E. and Foley, T.A., 1991. "The parameter r^2 in multiquadric interpolation", *Computers Math. Applic.*, 21(9):29–42.

Darmofal, D.L. and Haimes, R., 1995. "An analysis of 3D particle path integration algorithms". AIAA Paper 95-1713.

Greywall, M.S, 1993. "Streamwise computation of three-dimensional flows using two stream functions", *Journal of Fluids Engineering*, 115:233–238.

Kenwright, D., 1993. "Dual stream function methods for generating three-dimensional stream lines", PhD Thesis, The University of Auckland.

Mallinson, G.D, 1988. "The calculation of the lines of a three dimensional vector field", in G. de Vahl Davies and C. Fletcher, editors, *Computational Fluid Mechanics*, pages 525–534. North Holland.

Mallinson, G.D. and Kenwright, D.N., 1992. "Application of dual stream functions to the visualisation of three dimensional fluid motion", in M.R. Davis and G.J. Walker, editors, *Proceedings of the Eleventh Australasian Fluid Mechanics Conference*, pages 483–486.

Were, C.J. and Mallinson, G.D, 1995. "The free ALE method for flow in deforming geometries". These proceedings.

Woodgate, M.A., 1992. "Gradient recovery and superconvergence phenomena in linear finite elements", Oxford University Computing Laboratory Technical Report 92/13, Oxford.

Yih, C.S., 1957. "Stream functions in three dimensional flows". *La Houlle Blanche*, 3:445–450.