A numerical scheme for the generation of streamlines in three dimensions

Beale, S. B.

Publisher's version / la version de l'éditeur:
Proceedings CFD93, pp. 289-300, 1993

A Numerical Scheme for the Generation of Streamlines in Three Dimensions

by

S.B. Beale
National Research Council of Canada
Montreal Road, Ottawa, Ontario K0E 1X0

Reproduced from Proceedings
CFD93, Conference of the CFD Society of Canada
Montreal, June 14-15, 1993

A Numerical Scheme for the Generation of Streamlines in Three Dimensions

by

S.B. Beale

National Research Council of Canada

Montreal Road, Ottawa, Ontario K0E 1X0

Summary

This paper describes a proposed method to generate streamlines in three dimensions. The method is based upon the numerical solution for two mutually orthogonal scalar stream functions, with the velocity field for the flow presumed known. The stream functions represent stream-surfaces which are tangent to the vector field. Streamlines are obtained as the intersection of these two families of surfaces. The mathematical foundation for the methodology is described together with a discretization procedure. An illustrative example is then presented. The advantages and disadvantages of the proposed method are analysed, together with a discussion of problems encountered, with a view to future work on the subject.

1 Introduction

Computational Fluid Dynamics (CFD) may conveniently be divided into three distinct activities:

- Pre-processing work, such as grid generation.
- Solution of transport equations, i.e. the flow main solver.
- Post-processing: Graphics, animation, data manipulation.

The solution of the partial differential equations governing the flow of a fluid, typically involves the use of an iterative procedure based on e.g. the finite-volume [**1**], or some other method. With increased computer storage and speed, iterative procedures have also also been used in grid generation [**2**] in recent years.

This paper is concerned with the third activity above, namely CFD graphics: In spite of much progress being made in terms of quality and speed of available graphics software, the CFD practioner still has relatively few tools available for data analysis; vector fields, scalar lines and surfaces, and particle-tracking utilities being the most common. With the possible exception of the latter, these are not particularly computationally intensive. There is, however, no reason not to use iterative procedures in the tertiary activity of CFD graphics: Flow visualisations are assuming increasing importance and additional tools must be made available to the engineer in the future.

This paper describes a pilot project used to generate streamlines in three dimensions. Streamlines differ from particle traces in that that they are based on an Eulerian field-approach, rather than a Lagrangian basis. The whole-field of stored data is readily available to the graphics utility, having been obtained previously. The theoretical basis for the methodology developed in this paper is not new [**3, 4**], however the application of these concepts do not appear to have been exploited to-date. This work differs from previous material [**5, 6**] in that the goal is to use stream-functions to display the results of previously obtained flow-field simulations in a meaningful manner, rather than to actually solve the flow-field problem itself. It is thus presumed that the velocity field itself is known, *a priori*.

A stream function, ψ, is any function such that,

$$\rho\vec{u} \cdot \vec{\nabla}\psi = 0 \tag{1}$$

This function has the physical significance that constant ψ-values are streamlines in 2D, and stream-surfaces in 3D, i.e. they are tangent to the velocity field.

1.1 Linear algebraic equations

The methodology used is based on the finite-difference or finite-volume scheme [**1**], whereby partial differential equations are transformed to linear algebraic equations having the general form,

$$\begin{aligned} &a_W(\psi_W - \psi_P) + a_E(\psi_E - \psi_P) + a_S(\psi_S - \psi_P) + \\ &a_N(\psi_N - \psi_P) + a_L(\psi_L - \psi_P) + a_H(\psi_H - \psi_P) + S = 0 \end{aligned} \tag{2}$$

where ψ_P is the nodal value of ψ at some cell P. W, E, S, N, L, H refer to the west, east, south, north, low and high neighbours of P and S is a linearized source term. The linking coefficients a_E, a_W, a_S, a_N, a_L, a_H ensure that the influence of the neighbour-cell values upon the value of ψ at P is accounted for. In this paper it is presumed that the momentum equations are solved using a staggered scheme, whereby lower case subscripts, u_e, v_n, w_h etc. indicate that the stored velocity values are located at the east, north and high interfaces rather than at cell nodes.

The choice of methodology used here is subjective: The problem is readily amenable to alternative solution methodologies such as finite-element analysis, or any other appropriate scheme capable of solving Poisson equations. As a preamble to the general three-dimensional case, the simpler example of 2D streamlines is considered below.

1.2 Streamlines in 2D

Equation 1 is a scalar transport equation with convection only present. As such it is not cast in a suitable manner[1]. For 2D incompressible planar flow, the problem may be rectified by defining a stream function according to,

$$u = \frac{\partial \psi}{\partial y}, \quad v = -\frac{\partial \psi}{\partial x} \tag{3}$$

so that Eq. 1 is identically true. Equation 3 may also be re-cast in the form,

$$\nabla^2 \psi = -\omega \tag{4}$$

where ω is the two-dimensional fluid vorticity. Equation 4 is a diffusion-source (Poisson) equation, and is in a form suitable for solution.

Two methods of generating stream functions are apparent:

(a) Integration

(b) Iterative methods

Integration (a) is the method employed almost universally in current software. It has the obvious advantage of being non-iterative. However there are some 2D situations where method (b) yields better results; for example the use of integration to generate streamlines in problems involving multiply-connected geometry can result in systematic discontinuities [**7**].

Equations 3 and 4 are forms for solution using finite-difference and finite-volume methods, respectively. Integration may be considered as a restricted form of case (b), where only one (or two) linking *a*-coefficients [Eq. 2] are active. For this very reason, iterative procedures will always yield more precise results, although when simple grids are employed the advantage does not offset the additional computing cost, and 'integration' is to be preferred. It will become apparent that in three dimensions, no matter how many coefficients are active, the process is always iterative.

1. Upwind boundary conditions alone will affect the solution of Eq. 1, so that recirculation zones will not be correctly predicted, unless they happen to be cut by the boundary where ψ is prescribed.

2 STREAM FUNCTIONS IN THREE-DIMENSIONS

2.1 Definitions

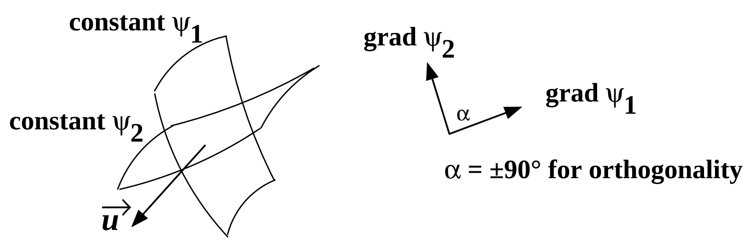

Fig. 1 Orthogonal stream-functions

For three-dimensional flow [**3, 4**] it is possible to define two stream functions, ψ_1 and ψ_2, which satisfy Eq. 1 such that,

$$\rho\vec{u} \equiv \lambda\vec{\nabla}\psi_1 \times \vec{\nabla}\psi_2 \tag{5}$$

The reader's attention is confined to the case of incompressible flow (without loss of generality choose $\lambda = \rho$). Surfaces of constant ψ_1 and ψ_2 represent space-surfaces which are tangent to the velocity field everywhere, Fig. 1. Thus grad ψ_1 and grad ψ_2 are vectors which are always perpendicular to $\vec{u}$. If grad ψ_1 and grad ψ_2 are also chosen to be mutually orthogonal so that,

$$\vec{\nabla}\psi_1 \cdot \vec{\nabla}\psi_2 = 0 \tag{6}$$

It follows that,

$$\vec{\nabla}\psi_1 = \frac{\vec{\nabla}\psi_2 \times \vec{u}}{\vec{\nabla}\psi_2 \cdot \vec{\nabla}\psi_2} \tag{7}$$

$$\vec{\nabla}\psi_2 = -\frac{\vec{\nabla}\psi_1 \times \vec{u}}{\vec{\nabla}\psi_1 \cdot \vec{\nabla}\psi_1} \tag{8}$$

Equations 7 and 8 are of the required form. The 2D case [Eq. 3] is recovered, by choosing grad $\psi_2 = \vec{k}$.

2.2 Discretization

Suppose ψ_2 is temporarily known. From Eq. 7:

$$\left.\frac{\partial\psi_1}{\partial x}\right|_w - \left.\frac{\partial\psi_1}{\partial x}\right|_e + \left.\frac{\partial\psi_1}{\partial y}\right|_s - \left.\frac{\partial\psi_1}{\partial y}\right|_n + \left.\frac{\partial\psi_1}{\partial z}\right|_l - \left.\frac{\partial\psi_1}{\partial z}\right|_h - \tag{9}$$

$$\left[\frac{1}{|\nabla\psi_2|^2}\left(\frac{w\partial\psi_2}{\partial y} - \frac{v\partial\psi_2}{\partial z}\right)_w - \frac{1}{|\nabla\psi_2|^2}\left(\frac{w\partial\psi_2}{\partial y} - \frac{v\partial\psi_2}{\partial z}\right)_e + \frac{1}{|\nabla\psi_2|^2}\left(\frac{u\partial\psi_2}{\partial z} - \frac{w\partial\psi_2}{\partial x}\right)_s\right.$$

$$\left. - \frac{1}{|\nabla\psi_2|^2}\left(\frac{u\partial\psi_2}{\partial z} - \frac{w\partial\psi_2}{\partial x}\right)_n + \frac{1}{|\nabla\psi_2|^2}\left(\frac{v\partial\psi_2}{\partial x} - \frac{u\partial\psi_2}{\partial y}\right)_l - \frac{1}{|\nabla\psi_2|^2}\left(\frac{v\partial\psi_2}{\partial x} - \frac{u\partial\psi_2}{\partial y}\right)_h\right] = 0$$

It is thus possible to write a set of a linear algebraic equations having the form Eq. 2 with,

$$a_e = \frac{A_e}{|\overrightarrow{P-E}|} \tag{10}$$

and similar terms for a_W, a_N etc. (The presence of the the area A_e in Eq. 10 serves only to write the linking terms in the familiar form of diffusion coefficients, similar terms occuring in the source-term below).

The source-term was computed from the six terms in parenthesis in Eq. 9, as follows: A nodal velocity vector, $\vec{u}_p$ is constructed at P based on the mean of the velocity-pairs u_w and u_e etc. Grad ψ_2 is similarly constructed[1]. The product of these two cell-centred vectors is formed and the following quantity obtained,

$$\vec{S}_P = \begin{pmatrix} S_{xP} \\ S_{yP} \\ S_{zP} \end{pmatrix} = \frac{\vec{u}_P \times \vec{\nabla}\psi_{2P}}{|\psi_{2P}|^2} \tag{11}$$

1. Grad ψ_1 and grad ψ_2 were in fact biased, being based only on the values at P, E, N, and H.

Defining S_E by,

$$S_e = \tfrac{1}{2}A_e\left(S_{xP} + S_{xE}\right) \tag{12}$$

with similar definitions for S_n, and S_h. The required source term in Eq. 2 may be written as,

$$S = S_e - S_w + S_n - S_s + S_h - S_l \tag{13}$$

(NB: S_w, S_s, and S_l are just the values of S_e, S_n, and S_h at the west, south and low neighbour-cells of P)

The resultant ψ_1 may then be used to solve for ψ_2 in a similar fashion after a number of sweeps. The coupled pair of linear algebraic equations are thus solved iteratively.

3 ILLUSTRATIVE EXAMPLE

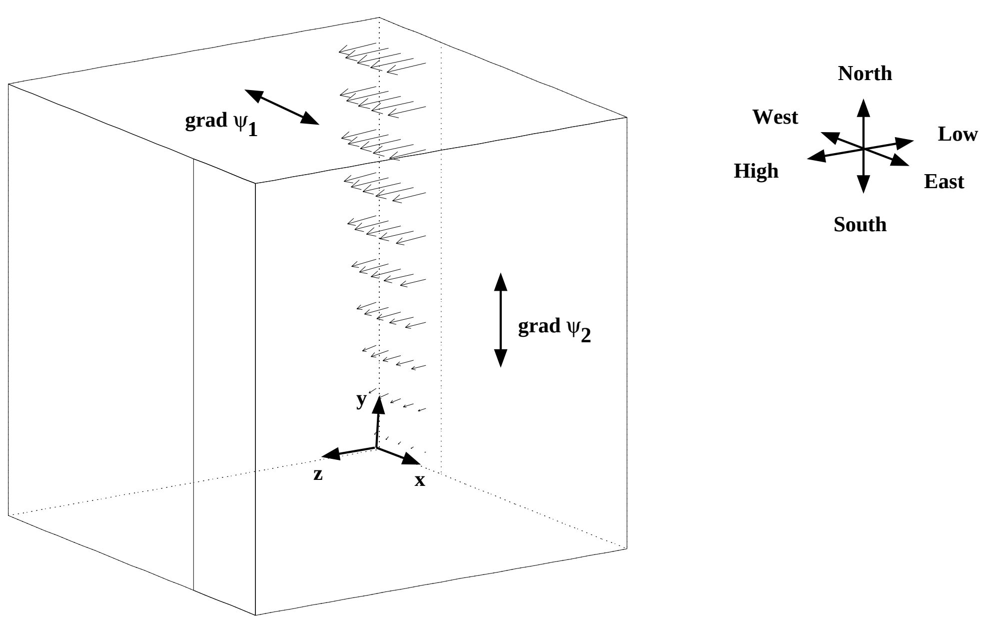

Fig. 2 Boundary conditions.

A rectangular duct is considered with the main flow in the z-direction. Solid walls, in the form of baffles, extend across three quarters of the inlet ($z = 0$) and exit ($z = z_{max}$) which are positioned diagonally opposite as illustrated in Fig. 2. The inlet velocity was specified as a parabolic-profile in the vertical y direction (similar to that occurring in a fully-developed duct flow) and uniform in the horizontal x direction. At the outlet the pressure was prescribed. The horizontal plane $y = 0$, and the two vertical planes $x = 0$, $x = x_{max}$, are solid walls, while the upper horizontal plane $y = y_{max}$ is treated as a symmetry plane.

On the two vertical walls (i.e. the east $x = 0$ and west $x = x_{max}$ planes) ψ_1 was fixed to constant values. On the low and high walls ψ_1 was fixed to these same values. In the horizontal (south and north) planes ψ_2 was set constant. The boundary ψ-values were chosen to satisfy the overall condition $\Delta\psi_1\Delta\psi_2 = Q$, where Q is the volumetric discharge ($\rho = 1$). Linear relaxation was applied to the source terms, S, in the finite-volume equations, while inertial relaxation was applied to the flux terms.

4 RESULTS AND DISCUSSION

Figure 3 shows velocity vectors on the plane of symmetry, $y = y_{max}$. Figure 4 shows contours of ψ_1. It can be seen that the main flow is in a diagonal direction with a large vortex forming behind the upstream baffle, and a smaller one downstream in the corner away from the exit zone. Figure 5 shows three-dimensional stream-lines in the interior of the flow field. These were obtained by parsing the data to see if each set of 8 nodal values span a given pair of values of ψ_1 and ψ_2, and if so drawing an element of the space curve (corresponding to the locus of intersection of the two iso-surfaces) through that volume in space [**7**].

Comparison of Figs. 3 and 4 shows that contours of constant ψ_1 on the symmetry plane to be tangent to the velocity field, as indicated by the vector plots. It can be seen that either streamlines enter and exit the domain of interest, or form simple closed loops. The stream-line results are less ambiguous than equivalent results generated using particle-tracking utilities (not shown). In the latter case, vortices are indicated by endless spirals and the user may receive false impressions of the flow due numerically-induced drift (for example in near-wall zones, where velocities are small). The reader who questions whether or not streamlines might assume such complex forms should appreciate that they are constrained by the fact that they correspond to the locus of two intersecting space-surfaces, and hence will, in general, assume simple forms. When using stream functions, the entire data set is available at the time of plotting and it is extremely simple to generate families of contours based on increments of ψ_1 and ψ_2 or zones of space in an orderly fashion

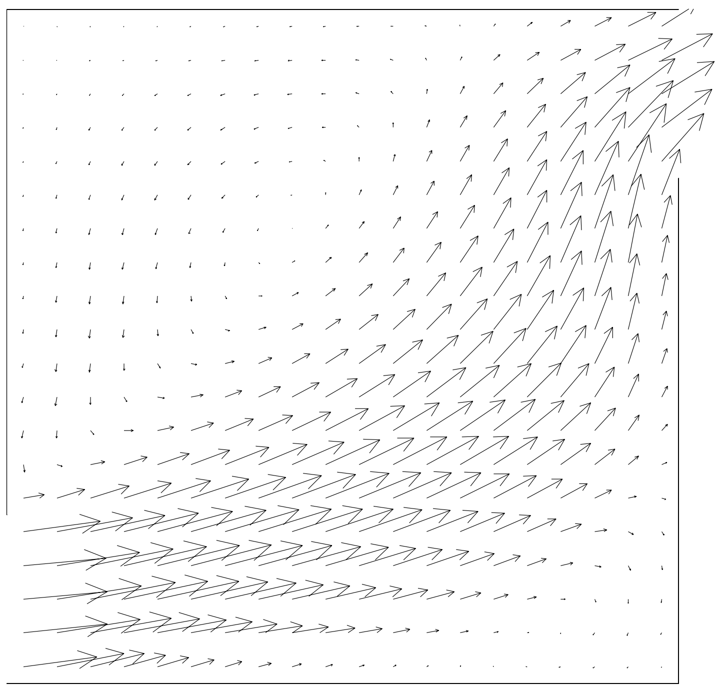

Fig. 3 Velocity vectors in the plane of symmetry ($y = y_{max}$).

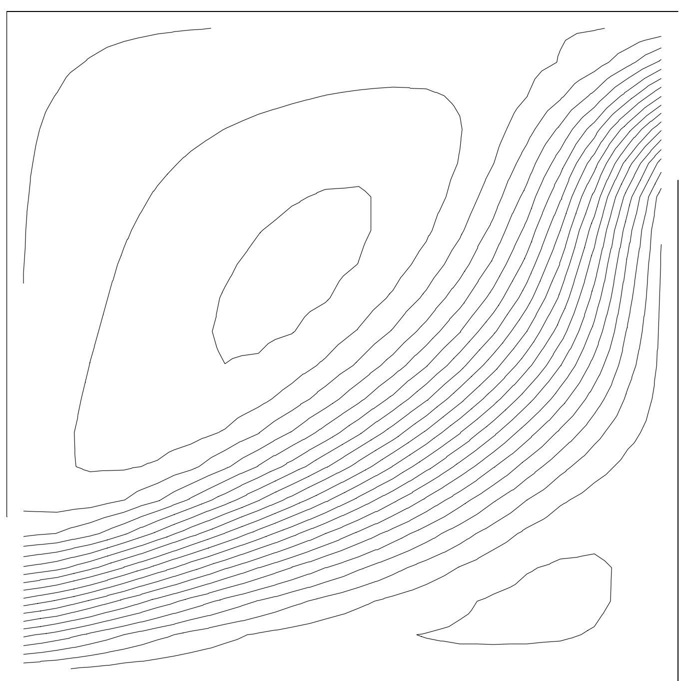

Fig. 4 Lines of constant ψ_1 in the plane of symmetry ($y = y_{max}$).

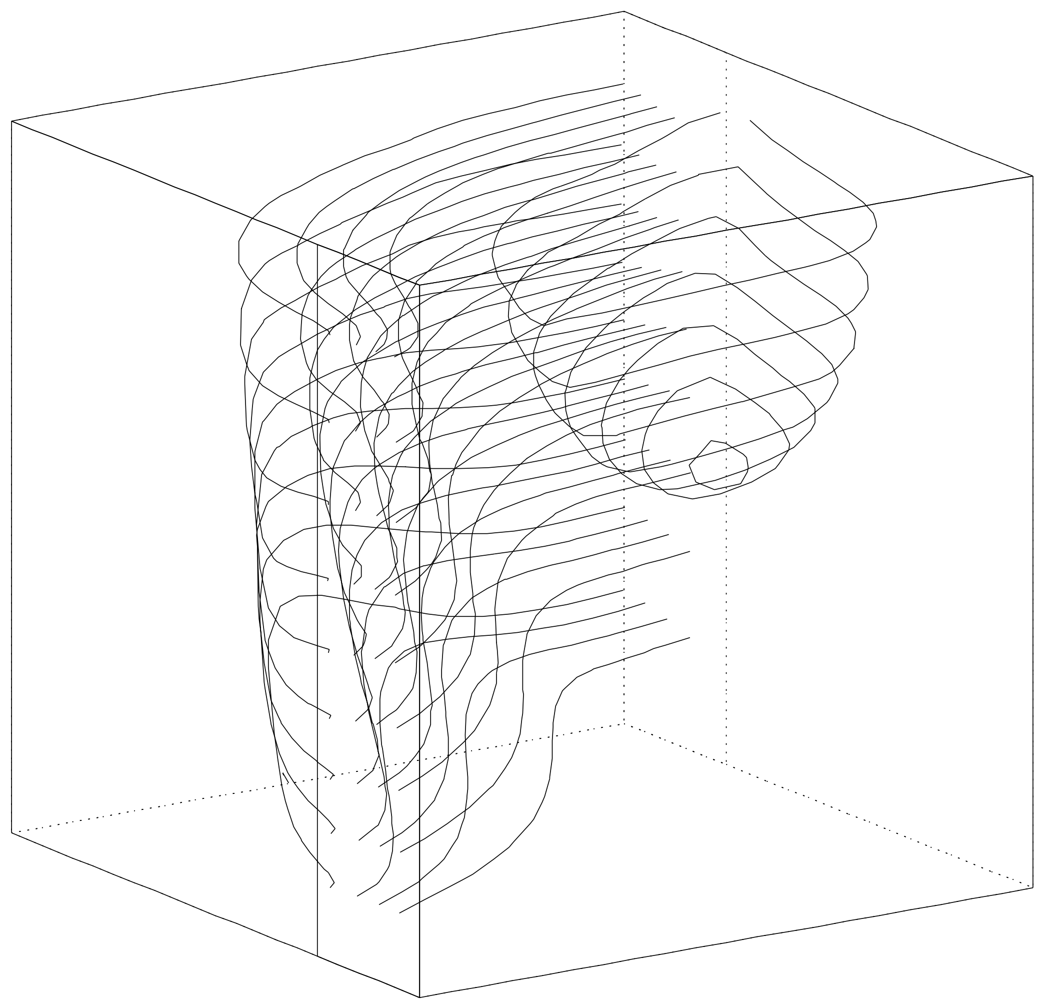

Fig. 5 Streamlines in three dimensions.

Some other advantages of the stream function approach are,

- The entire field of data is readily available for graphics with only a minimal need for further computing
- Line density is proportional to volumetric discharge (or mass flux)
- Good continuity, ie. no drift, as a result of all six neighbour linkages being active
- Streamlines may either be stationary or move with the flow
- Unsteady 3D flows may be visualised
- Compressible flows may be considered
- The method may be also used to display vortex-lines and surfaces (by replacing $\vec{u}$ by $\vec{\omega}$ in Eqs 7 and 8)

Some of the disadvantages when compared to particle-tracking are:

- Much more CPU intensive
- Experience required to formulate problem
- Possible to overprescribe boundary conditions
- Convergence is not guaranteed

Of these, the latter problem was found to be quite serious and the method described in this paper cannot, as yet, be considered reliable. In spite of early success with simple flow regimes, stability concerns were always a problem. It was found to be very difficult to obtain converged results (including for the example above). Difficulties were almost always encountered when simulating complex flows particularly when non-uniform grids were concentrated (so that the cells were short and flat) notably in regions where velocities are small. This may be due to fluctuations in the gradients of ψ_1 and ψ_2 resulting in stability problems via the coupled governing Equations 7 and 8. (Any attempt to de-couple them leads back to equations of the form Eq. 1).

Numerical generation of stream-surfaces is in some senses conceptually similar to grid generation. Two families of surfaces are required (the third family being pre-defined as the set of surfaces which are normal to the velocity field). Prior to solving the problem, the user must conceive of and identify appropriate boundaries for the stream surfaces. In the problem above, a set of rectilinear stream surfaces was selected. Many choices of orthogonal stream function are possible: for example, for a duct, it is also possible to define a set of radial-polar surfaces (resembling an O-grid). Other basic patterns may also be identified. Of course stream-surfaces differ from conventional grids in that they can form multiple closed sheets and so forth. The preferred choice for the stream-function corners will depend to some extent on the characteristics of the flow at hand, and how the user wishes to display the results. Care is required to avoid overprescribing boundary conditions (particularly when specifying pairs of Dirichlet boundary conditions). It is also possible to accidentally prescribe incompatible patterns of stream-surfaces.

Future work should focus on the stability problems mentioned above, and also address alternate discretization procedures and improved solvers: The solver used here was a Jacobi point-by-point procedure. Slab-wise, and whole-field procedures were also considered, however these were based on existing routines where the outer loop of the solver proceeded in the direction of the main flow. Consideration should also be given to flow-solvers which proceed across the flow, in the overall directions of grad ψ_1 and grad ψ_2: By de-activating all but one linking coefficient in two mutually perpendicular directions from the finite-volume equations, and marching across the flow from known stream-surfaces, it is possible that a substantial reduction in computation time could be effected. Note however that the procedure is still iterative; the equations are coupled via the source-terms (which themselves include various neighbour values). Note that it is possible to add a fraction of Eq. 1 to Eqs. 7 and 8, and employ a hybrid or higher-order scheme. This however was not found to offer any improvement in convergence, at least for the problem at hand. Other formulations are also possible, the best being that which renders as precise a solution as possible in as rapid a time as possible.

The formulation given above is correct for orthogonal stream functions when Cartesian co-ordinates or orthogonal body-fitted co-ordinates (BFC) grids are employed. To obtain results using orthogonal stream functions in non-orthogonal BFC grids, it is further necessary to compute the tangential and normal resolutes or components of $\vec{u}$ grad ψ_1 and grad ψ_2, and the direction cosines between the various directions. This adds additional complexity, but certainly is not beyond the realm of feasibility. The use of arbitrary non-orthogonal stream-surfaces in non-orthogonal BFC grids is an interesting subject, worthy of future research.

It is believed that the procedure described above could form the basis for a method for generating streamlines in the future. More work is required in this regard in order to increase the probability and speed of convergence, and the flexibility of the solution methodology, but it is maintained that the methodology holds promise. In conclusion: While there is more computational work associated with the stream function approach, the end result is elegant and utilitarian. With further progress in algorithm development, such a tool could well find widespread application.

Acknowledegements

The author would like to thank Prof. D.B. Spalding and Dr. D.R. Glynn of Concentration Heat and Momentum Ltd. for supplying the source code to, and giving advice on the use of the graphics procedure used in this project. The support of the National Research Council of Canada is gratefully acknowledged.

Nomenclature

Symbol

a	Linking coefficients in finite-volume equations
A	Area of cell interfaces
Q	Volumetric discharge
S	Source term in finite-volume equations
$\vec{u}$	Velocity
u, v, w	Speed in x, y, and z directions
ρ	Fluid density
λ	Constant
ψ, ψ_1, ψ_2	Stream functions
ω	Vorticity

Symbol

P	Nodal value at C
W, E, S, N, L, H	Nodal values at West, East, South, North, Low, and High neighbour cells of P
w, e, s, n, l, h	Value at west, east, south, north, low, and high interfaces

References

1. PATANKAR, S.V., and SPALDING, D.B. - A calculation procedure for heat, mass, and momentum transfer in three-dimensional parabolic flows. International Journal of Heat and Mass Transfer, 15, 1787-1806, 1972.
2. THOMPSON, J.F., WARSI, Z.U.A. and MASTIN C.W. Numerical Grid Generation, Foundations and Applications, North-Holland, New York, 19??
3. YIH, C.S. - Stream Functions in Threee-dimensional Flows. La Houille Blanche, 12, 3, 1957.
4. Advanced Mechanics of Fluids - Editor H. Rouse, Wiley, New York, 1958
5. WU, C.H. General Theory of Three-dimensional Flows in Subsonic and Supersonic Turbomachines. NACA TN 2604, 1952.
6. SHERIF, A., and HAFEZ, M. Computation of Three-dimensional Transonic Flows using Two Stream Functions. International Journal for Numerical Methods in Fluids, Vol. 8, pp. 17-29, 1988.
7. BEALE, S.B. - Fluid Flow and Heat Transfer in Tube Banks. Ph.D. Thesis, Imperial College, University of London, 1993.