University of Strathclyde
Department of Mechanical Engineering
Division of Thermo-Fluid and Environmental Engineering

VORTEX SHEDDING FLOWMETER PULSATING FLOW CFD STUDIES

by

THOMAS J SCANLON

A thesis submitted for the degree of Doctor of Philosophy in Mechanical Engineering.

December 1992

SUMMARY

The computational analysis of vortex shedding flow is presented, using the commercially available computational fluid dynamics (CFD) software package PHOENICS. In this analysis it is shown how the use of the conventional PHOENICS default first-order hybrid-upwind convective differencing scheme provides an excellent example of the effects of multidimensional false diffusion. These effects are substantially reduced with the introduction of an alternative scheme, SUCCA (**S**kew **U**pwind **C**orner **C**onvection **A**lgorithm), for the modelling of convective transport in 2D and 3D analyses; resulting in the promotion of continuous vortex shedding for the 2D model. The mechanism of pulsating flow influence on the vortex shedding process has also been simulated. The results show that a complex transient phenomenon such as vortex shedding can be analysed using the PHOENICS code but only with the implementation of an alternative convection algorithm. The results also demonstrate the SUCCA scheme's ability to accurately represent convective transport and hence substantially reduce the effects of multidimensional false diffusion in numerical flow analyses.

ACKNOWLEDGEMENT

I would like to thank Dr. Simon M Fraser and Dr. Christopher Carey for their supervision and encouragement throughout this work.

I am very grateful to Mr. William Dempster for some helpful discussions during this period of work.

Special thanks are due to my family and friends for their continuous support.

I would finally like to express my appreciation to the Manufacturing Technology Division of the Department of Trade and Industry and the National Engineering Laboratory, East Kilbride for providing the funding for this project.

TABLE OF CONTENTS

APPENDICES:

NOMENCLATURE

a	discretised convection / diffusion coefficient
A	bluff body length, area
b	linearised source / sink term
B	bluff body width
C	modified convection coefficent
C1	species concentration
D	pipe diameter
f	vortex shedding frequency
H	channel width
k,KE	turbulence kinetic energy
$\dot{m}$	cell face mass flux
O	order of accuracy
P,p	pressure term
P,N,S,E,W, SW,SE,NW, NE	main and surrounding grid point locations
PIL	PHOENICS Input Language
Pe	Peclet number
r	radial component direction
Re	Reynolds number
Str	Strouhal number
S	volumetric source / sink term

T,t	time
U,u,U1	u-velocity component
V,v,V1	v-velocity component
W,w,W1	w-velocity component
X,x	x-direction
Y,y	y-direction
Z,z	z-direction
δ	distance, Kronecker delta
ε	fluid strain
ϵ ,EP	dissipation rate of turbulence kinetic energy
$\sum$	summation term
σ	normal stress, combined stress
ϕ	computed dependent variable
μ	fluid molecular viscosity
ρ	fluid density
θ	skew flow angle, circumferential component direction
τ	shear stress term
ν	fluid kinematic viscosity

subscripts

i,j,k	Cartesian tensor notation components where $i = 1,2,3$ $j = 1,2,3$ $k = 1,2,3$
l	laminar conditions

nb	neighbour
o	inlet conditions
P,N,S,E,W, SW,SE,NW, NE	main and surrounding grid point locations
p	inlet pulsating flow conditions
r	radial component direction
rms	root mean square
s,w	south and west cell faces
sw	typical modified convection term location for upwind cell
t	turbulent conditions
∞	outlet condition
ϕ	computed dependent variable
θ	circumferential component direction
vs	vortex shedding under steady flow conditions
vp	vortex shedding under pulsating flow conditions

superscripts

$\sim$	instantaneous quantity
$-$	average quantity
$'$	fluctuating quantity

CHAPTER 1 INTRODUCTION

1.1 The Vortex Shedding Flowmeter

The separating flow around bluff (or non-streamlined) bodies is renowned for its production of an unsteady wake. At appropriate Reynolds numbers the wake may be seen to take the form of the well known Karman(1) vortex street in which the cyclically shed vortices are convected downstream in a regular array. Such unsteady flows are of practical interest to engineers. Particular examples may range from the vortex induced vibrations of structures to the vortex shedding flowmeter where, in the latter case, the frequency of the shed vortices should be directly proportional to the bulk mean velocity in a well designed meter.

Over the last 20 years the vortex shedding flowmeter has been developed into a valuable metering instrument, being used for many industrial flow process monitoring and control applications(2,3) involving all types of liquids, gases and steam. The many advantages of this type of meter include good accuracy and long-term repeatability, a wide operating range and a linear frequency output. This linear frequency output is found to be a function of the dimensionless Strouhal number:

$$Str = \frac{f_{vs} b}{\bar{U}} \qquad 1.1$$

The principle of the meter is such that having measured the transient vortex shedding frequency f_{vs} , the calibrated universal Strouhal number for the meter is used with the bluff body width b to obtain the mean bulk velocity $\bar{U}$.

The meter itself has no moving parts but its operation does involve a hydrodynamic oscillation. It is known that the vortex shedding process in steady flow remains completely stable, even in the presence of high turbulence levels such as those found in high Reynolds number pipe flows (see Chapter 2). In periodic pulsating flow, however, from a source such as a reciprocating or centrifugal positive displacement machine it has been shown(4,5) that there exists an interference with the vortex shedding process.

The extent of this interference is found to be a function of the pulsation frequency and pulsation amplitude, with the maximum interference occurring with the vortex shedding process synchronizing itself at half the pulsation frequency; a phenomenon known as **locking-in**. This interference manifests itself as percentage change in the calibrated Strouhal number for the meter with changes of ± 40% occurring at lock-in, even though the pulsation amplitude may only be of a comparable level to that of normal pipe turbulence.

1.2 Computational Fluid Dynamics

The development of computational fluid dynamics (CFD) software for the numerical prediction of complex fluid flow behaviour has taken place over the last 30 years, in parallel with advances in digital computer technology. The development of such CFD codes has given the engineer a valuable tool which will further promote the understanding of fluid flow behaviour.

Computational techniques involved in solving fluid flow problems fall into two main categories, these being:

1) finite difference techniques

2) finite element techniques

The method of finite differences, within a finite volume environment, is described in detail in Chapter 3. Among the advantages of the use of this technique are, firstly, the maintenance of the conservation principle embodied within the flow equations and , secondly, the use of stable, efficient and accurate solution methods.(36)(38)

In the finite element method the flow equations are approximated using parameters relating to the geometry of the fluid element and the mathematical functions chosen to represent the velocity and pressure terms at locations on the edges and within the finite element. A penalty function method(57)may be adopted to give a relationship between the pressure field and the continuity equation. Such a method provides the basis for a stable and efficient numerical solution of the flow field since it promotes banded coefficient matrices.

The principal advantage of finite elements is the geometrical flexibility in the problem description which the technique allows. The development of more stable and efficient solution algorithms(57)(58)has seen an increase in the use of finite element CFD codes and for the vortex shedding problem itself Van de Vosse *et al*(58)have predicted this unsteady problem in laminar flow.

In the context of this work, the commercially available CFD software package PHOENICS(6), a finite volume / finite difference type code, has been used in an attempt to ascertain the extent of the capabilities of such a code in relation to the simulation of a complex transient phenomenon such as vortex shedding. Emphasis has been placed upon analyzing the vortex shedding process for both laminar and turbulent flow regimes, turbulent pulsating flow conditions with specific regard to the locking-in process and three-dimensional turbulent flow.

CHAPTER 2 LITERATURE REVIEW

The literature available in relation to the subject of flow over bluff bodies is extensive. The following review represents an assessment of the more relevant and most commonly referenced articles on the matter.

The unsteady wake produced by flow over a bluff symmetrical obstacle for a Reynolds number above a certain critical value was first observed by Strouhal(7), who noted the subsequent downstream formation of a vortex street. Mallock(8) and Benard(9) both described the initial formation of twin symmetrical standing vortices downstream of the obstacle at the beginning of the motion, followed by an elongation of the vortex pair to an asymmetrical position. The vortices were then observed to move away from the body, being discharged alternately from the two sides. This eddying motion was observed to possess a definite frequency for each Reynolds number.

Mallock and Benard observed that downstream, the vortices assumed what appeared to be a regular pattern, which in most cases was evident at a distance four or five diameters behind the solid body. The vortices arranged themselves in a double row, in which each vortex was positioned opposite the mid-point of the interval between the two vortices in the opposite row. In suitable circumstances the trail of vortices was seen to persist for a considerable distance downstream of the solid body.

Karman and Rubach(10) considered the system far downstream of the bluff body and found that the shed vortices do not in fact arrange themselves exactly on two parallel rows with a definite spacing ratio but the trail tends to widen out downstream, and the spacing ratio changes. It has been suggested(11) that the regularity shown in many photographs is often due, in part, to the effect of channel walls, which have a lateral compressing effect on the flow.

The first detailed experimental examination of the structure of the free vortex layers in the bluff body wake was made by Fage and Johansen(12), who, for two-dimensional motion examined the layers from bodies of several shapes. The authors found that the velocity distribution across the vortex layer in its initial stages showed maxima at the outside edges and minima at the inside edge with the maximum velocity in every case greater than the undisturbed free stream velocity. Measurements of the rate of discharge of vorticity and pressure distribution across the layer were also taken.

Schiller and Linke(13) considered a similar flow geometry, extending the analysis further downstream of the bluff blockage. The authors found that for certain Reynolds number values (based on the diameter of the bluff cylinder) the width of the vortex layer grew according to $x^{1/2}$ where x is some downstream distance. Since this law of growth is characteristic of the laminar boundary layer, it was assumed that the vortex layers are laminar so long as this law holds. Schiller and Linke also found that, as expected, the critical distance for the transition to turbulence approaches the cylinder as the Reynolds number increases.

At higher Reynolds numbers the diffusion of the vortices takes place so rapidly that the formation of the double vortex row is lost in the turbulent wake. However vortices continue to be shed with unfailing regularity until an approximate Reynolds number (based on bluff body width) of 5×10^5. Beyond this upper limit the flow is completely turbulent(14). Humphreys(54)and Morkovin(55)both noted this complete loss of periodicity in the wake for Reynolds numbers between the critical value of 1×10^5 and 3.5×10^6 when the re-attachment of the two vortex layers was observed. However, Roshko(56), who performed experiments for Reynolds numbers up to the value 1×10^7, observed the recovery of periodicity in the near wake at this post-critical Reynolds number.

On the frequency of vortex shedding Rayleigh(15) suggested the following formula for low Reynolds number flow over a circular cylinder:

$$\frac{f_{vs}d}{\bar{U}} = 0.195\left(1 - \frac{20.1}{Re}\right) \qquad 2.1$$

based on the result of his analysis of the original observations of Strouhal(7). The dimensionless ratio $f_{vs}d/\bar{U}$ is commonly referred to as the Strouhal number after the Austrian physicist. Karman and Rubach(10), Fage and Johansen(12) and Tyler(16) all made experimental studies of Strouhal number variation for various bluff body shapes, while Rosenhead and Schwabe(17) considered the effects of channel walls on the stability and characteristics of the vortex street.

The theoretical analysis of flow past bluff bodies was first considered by Kirchoff(18) and Karman(1). In the free streamline theory developed by Kirchoff, the free vortex layers which originate from the bluff body are idealised by surfaces (streamlines) of velocity discontinuity. These free streamlines divide the flow into a wake and an outer potential field. Kirchoff's theory, however, assumes the velocity of the vortex layer to be that of the undisturbed free stream velocity. This underestimation of the separation velocity, as experimentally verified by Fage and Johansen(12), thus predicts values of drag lower than those observed.

Karman(1) approached the problem by considering the periodic vortex shedding nature of the flow itself. However, the theory was incomplete in that it could not, by itself, relate the vortex street dimensions and velocities to the obstacle dimension and free stream velocity. Additional empirical data was required from experimental analyses.

Roshko(19) proposed a modified Kirchoff analysis where the separation velocity is allowed to assume some arbitrary value as a function of the free stream velocity. The same author(20), by allowing for some annihilation of vorticity in the free vortex layers (following Fage and Johansen), combined the free streamline theory of Kirchoff with Karman's theory of the vortex street to relate the drag to only one experimental measurement, that of the Strouhal number. In this paper, Roshko also defined a universal Strouhal number based upon the free shear wake width and the separation velocity. Roshko(20)stated that vortex shedding frequency is related to the wake width, with the following inverse relationship applying:

$$f_{vs} \propto \frac{1}{d^*} \qquad 2.2$$

where d^* is the wake width between the free shear layers prior to vortex creation. With this general relationship it follows that bluffer bodies, which promote a larger wake width, will have lower vortex shedding frequencies and hence lower Strouhal numbers. However, this relationship is only appropriate at higher Reynolds numbers (> 10000) for which the Strouhal number is distinctly different for different shaped cylinders, though approximately constant for each individual one.

Gerrard(21) proposed that there should in fact be two characteristic lengths influencing the mechanics of the vortex shedding process; these being the axial length of the formation region immediately downstream of the bluff body and the free shear wake width. Gerrard(22) also noted the apparent transition to three-dimensionality in the wake of a two-dimensional bluff body in which stable, transitional and fully turbulent flows were considered.

The existence of coherent large scale structures behind bodies at high Reynolds numbers has been well established(23). Cantwell(24) studied the near wake of a two-dimensional cylinder at a Reynolds number of 140000 to reveal coherent structures having an appearance resembling the classical periodic vortex pattern of Karman(1).

The study of Perry and Watmuff(25) revealed that coherent large scale eddies exist in the wakes behind three-dimensional blunt bodies at high Reynolds numbers and that these structures retain their identity for long streamwise distances. Also, the geometry of the phase-averaged vector field displays the same general features as in the unsteady laminar flow wake results of Perry *et al* (26).

Computational analyses in this field have mainly concentrated upon unsteady laminar and turbulent flow past circular cylinders(27)(28)(29). Some such simulations(29) analyzing the two-dimensional unsteady form of the Navier-Stokes equations have utilised the discrete vortex method, employing a transport equation for vorticity. This Lagrangian approach helps to circumvent the problems of artificial viscosity and stability often associated with Eulerian analyses(30)(31). However, problems involved in the modelling of energy dissipation and vortex diffusion in the wake have to be addressed. Others such as Braza *et al* (27) have used a control volume / finite difference approximation for the two-dimensional governing equations using a log-polar coordinate system to overcome the problem associated with modelling the circular geometry of the bluff cylinder. It was observed in this analysis that due to, what was considered to be, a relatively low order of accuracy (O,2) present in the numerical schemes employed that a very fine grid was required for moderate (300) and high (1000) Reynolds number calculations.

The most notable computational study involving vortex shedding from rectangular cylinders is that of Davis *et al* (32)(33). In these analyses the two-dimensional unsteady laminar form of the Navier-Stokes equations are modelled using a direct finite difference version of the conservative form of the governing equations. Convective transport was modelled using the third-order accurate QUICKEST scheme of Leonard(34).

CHAPTER 3 THE PHOENICS CFD CODE

3.1 General Description

3.1.1 The Structure of PHOENICS

PHOENICS (Parabolic Hyperbolic Or Elliptic Numerical Integration Code Series) is a commercial software tool for analysis of transport phenomena governed by conservation equations. For the numerical analysis of fluid flow in general, the description of 'computational fluid dynamics' (CFD) is often associated with such codes.

The two essential components within the PHOENICS code are a pre-processor called SATELLITE and a processor called EARTH.

The pre-processor SATELLITE can be described as an interpreter; from instructions provided by the user it creates a data file containing instructions which EARTH can understand and obey. Normally, SATELLITE will receive its instructions through a file called Q1, provided by the user, which contains all of the geometry, boundary conditions and solution control parameters necessary for the problem description.

The processor EARTH contains the main flow-simulating software. EARTH reads the data file provided by SATELLITE and executes the corresponding computations; it then creates an output file called RESULT, which the user can read, and a file of results called PHIDA. The PHIDA file is a graphics file which can be visualised by using the post-processor PHOTON.

Both SATELLITE and EARTH possess space for additional FORTRAN coding which is input by the user. In EARTH this coding framework is called GROUND and the main function of GROUND is to provide a variety of data-setting and feature-adding alternatives that are not contained within

EARTH. As the use of GROUND necessitates the addition of user-defined FORTRAN coding sequences, re-compilation and re-linking sequences must be performed prior to execution.

3.1.2 Equations Solved

The conservation equations solved by PHOENICS take the form of partial differential equations describing, for example, the conservation of:

- mass, momentum, energy, chemical reactions
- in one-, two- and three-dimensional geometries
- for steady and unsteady problems

The conservation equations have a general form:

$$\frac{\partial}{\partial t}(\rho\phi) + \frac{\partial}{\partial x_j}(\rho u_j \phi) = \frac{\partial}{\partial x_j}\left(\Gamma_\phi \frac{\partial \phi}{\partial x_j}\right) + S_\phi \qquad 3.1$$

as derived in Appendix A. The four terms in this **general differential equation** are referred to as the **transient, convection, diffusion** and **source** terms respectively. The general differential equation may take various different mathematical forms, for example, the momentum equation can be described as a second-order, non-linear partial differential equation.

3.2 Numerical Solution of Fluid Flow Within PHOENICS

3.2.1 Numerical Approximation

The partial differential equations describing the fluid flow are translated into a set of (solvable) linear algebraic equations. This is achieved by considering the behaviour of the continuous equations at discrete intervals of time and space. This process is known as **discretisation**. In PHOENICS the **finite-volume method** is used for discretisation.

3.2.2 The Finite-Volume Method

The strategy behind the finite-volume method can be described in the following manner. The geometrical space occupied by the problem is subdivided into a number of cells or volumes as shown in Figures 3.1 to 3.4. Each cell has six faces and has at its centre a grid point (node) **P**. The neighbouring nodes are denoted as North, South, East, West, High and Low. For rectangular Cartesian or cylindrical-polar grids, the lines connecting grid points are aligned with the coordinate axes.

At the point **P**, we denote the value of the dependent variable as ϕ_P . ϕ_P is related to the values of ϕ at neighbouring grid points, and in unsteady problems to the value of ϕ at an earlier time interval. A linear algebraic equation expressing this relationship may be written as:

$$a_P\phi_P = a_N\phi_N + a_S\phi_S + a_W\phi_W + a_E\phi_E + a_H\phi_H + a_L\phi_L + a_T\phi_T + b \qquad \ldots 3.2$$

where the ***a***'s are coefficients and ***b*** is necessary to take account of source terms. The coefficients and source term in equation 3.2 are formulated by integration of the appropriate differential equation over each control volume and time interval as described in Patankar(35) and as illustrated for steady flow in Section 3.3. For discretisation in time, implicit temporal differencing(35) is used.

Equations of the type 3.2 are known as **finite-volume equations** and equation 3.2 describes the processes affecting the value of ϕ in cell **P** in relation to its neighbour cells, transient effects and source term b.

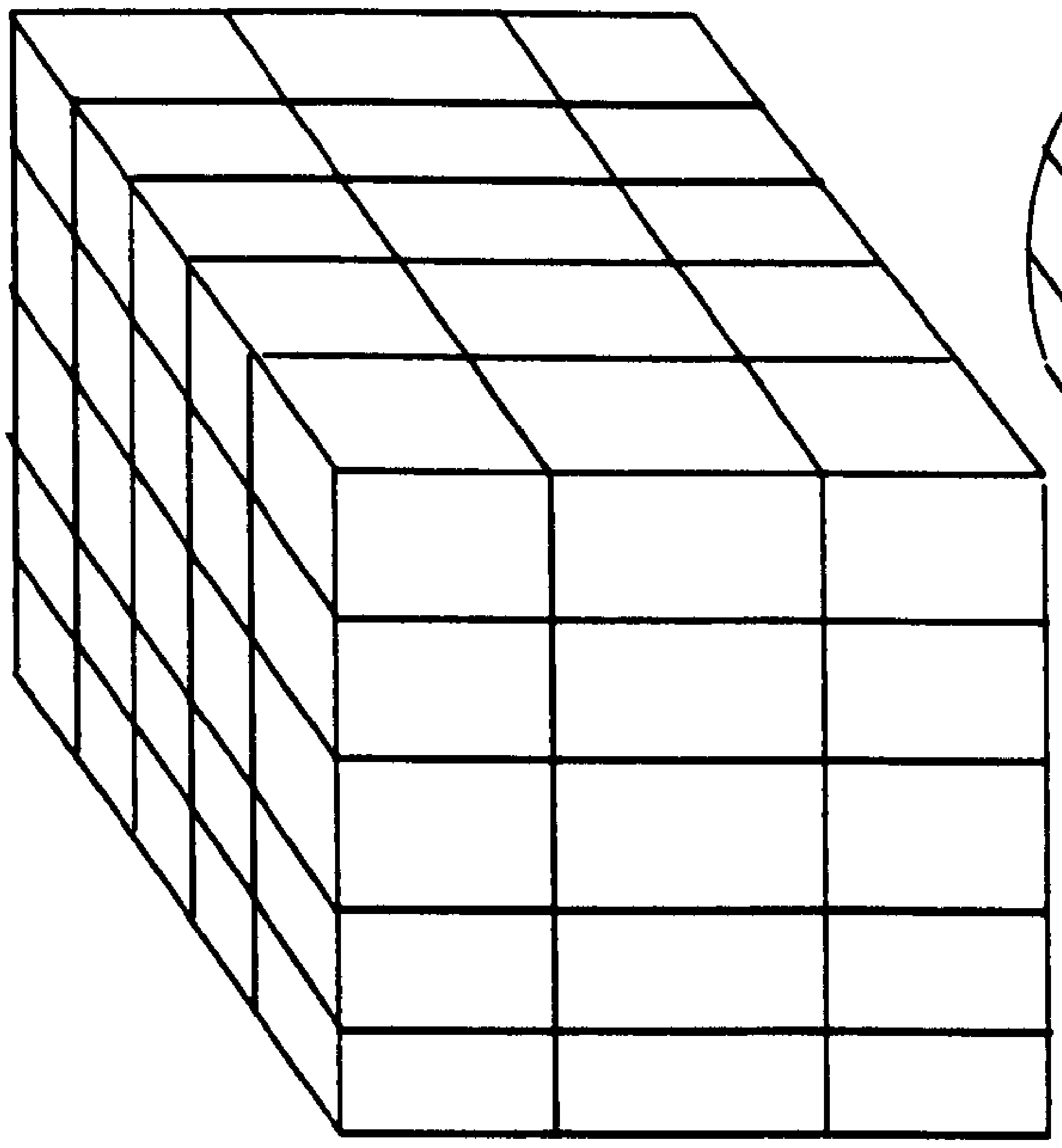


Fig.3.1 *Rectangular Cartesian grid.*

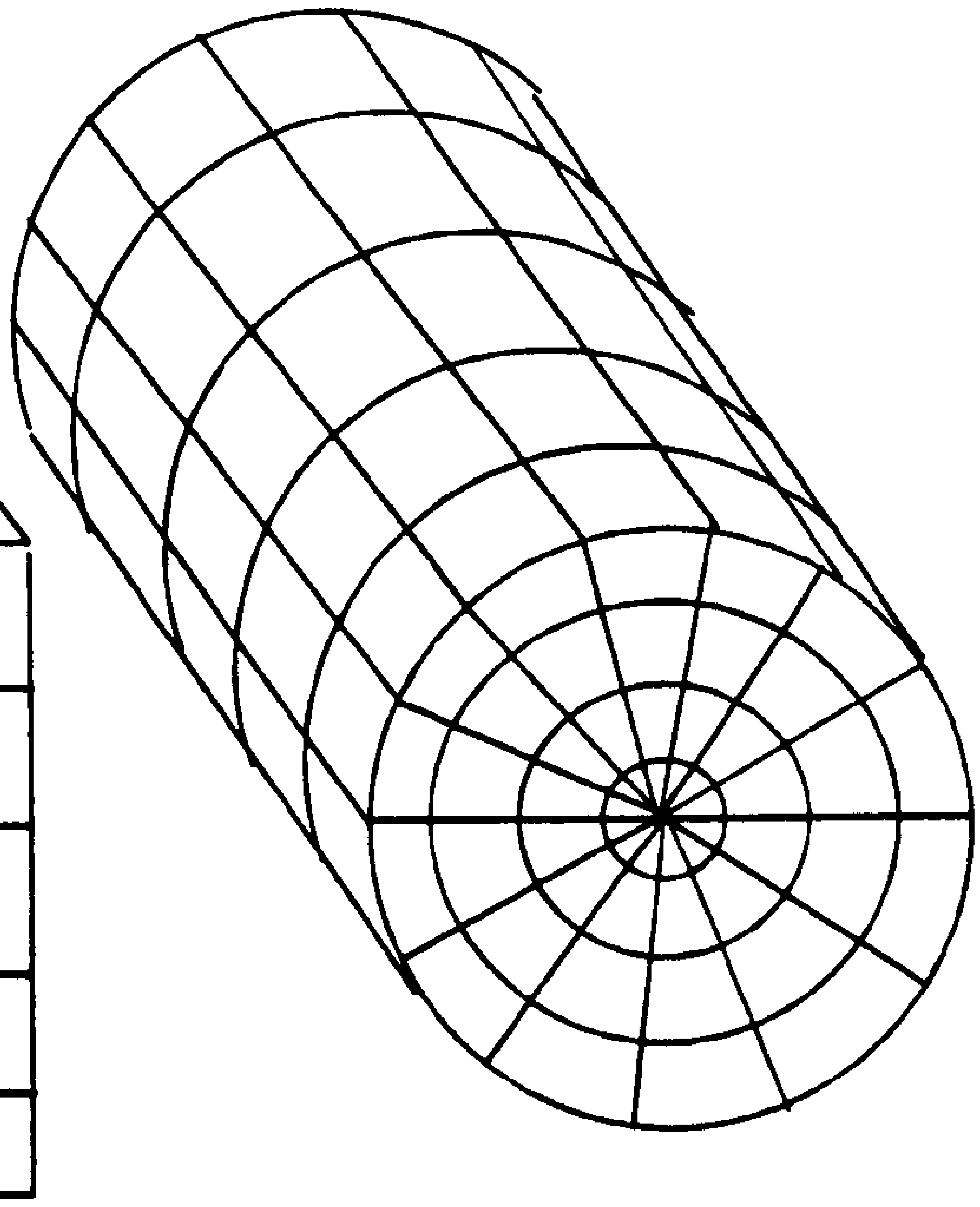


Fig.3.2 *Polar-cylindrical grid.*

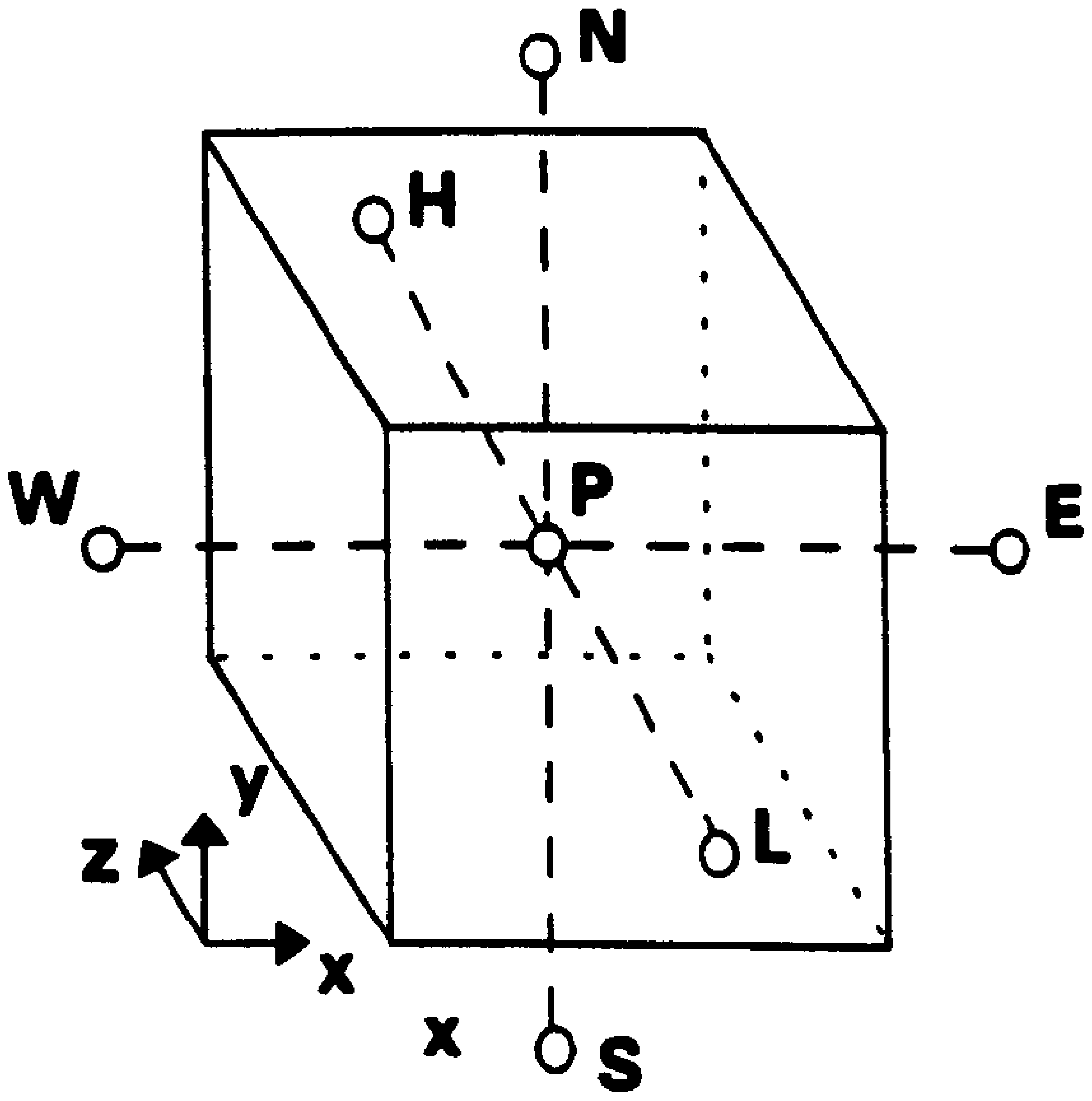


Fig.3.3 *Rectangular cell molecule.*

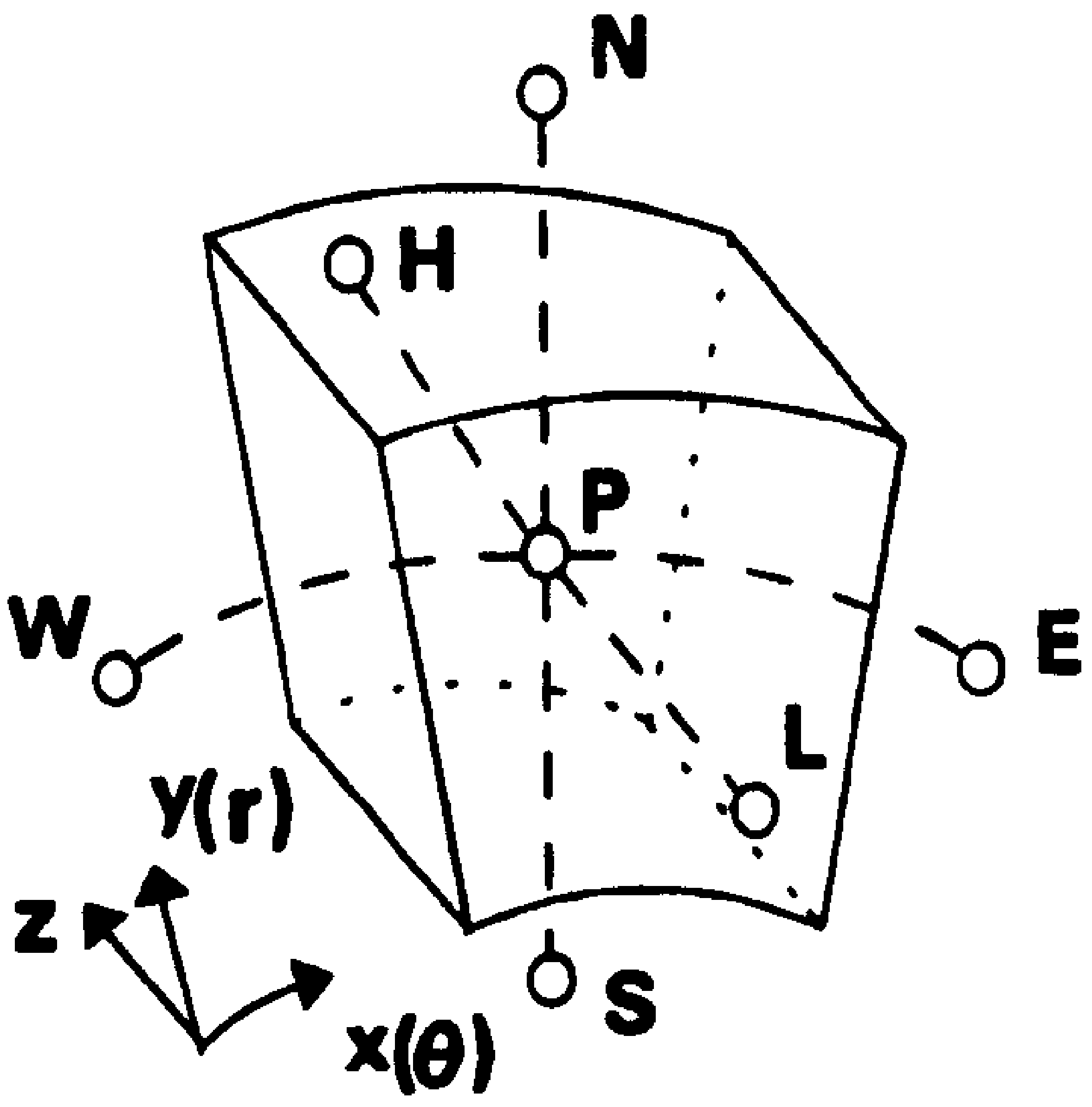


Fig.3.4 *Cylindrical cell molecule.*

3.2.3 Staggered Grids

To avoid numerical problems associated with schemes involving downwind grid points for the representation of the pressure and convection terms (35)(36)(37) PHOENICS uses a **staggered grid** for the solution of the hydrodynamic (momentum) equations as shown in Figure 3.5. The grid cell structure specified by the user is the grid used for the solution of **scalar** quantities; for example, the values of pressure and species concentration may be found at the central points of each cell. The **velocity** components are found for points at the centre of the faces of these grid cells.

3.2.4 Source Terms and Boundary Conditions

Boundary conditions within PHOENICS are implemented by the inclusion of additional **source** and/or **sink** terms in the finite volume equations for computational cells at the domain boundaries. Since the form of the source term is required to be linear in nature for compatibility with the algebraic equation solution, PHOENICS adopts the following relationship for source terms:

$$S_\phi = f \times CO \times (VAL - \phi) \qquad \mathbf{3.2a}$$

where ϕ is the solution dependent variable, f is a multiplying factor, ***CO*** is the source coefficient and ***VAL*** is the required value of dependent variable in each boundary cell. The use of this relationship is shown in the following example which considers the zero-slip boundary condition for the x-direction momentum equation in laminar flow. In Figure 3.6, the exact expression for the wall shear stress is $\tau_x = -\mu (\partial u/\partial y)$. For the cell shown in the Figure, the velocity gradient at the wall is approximated

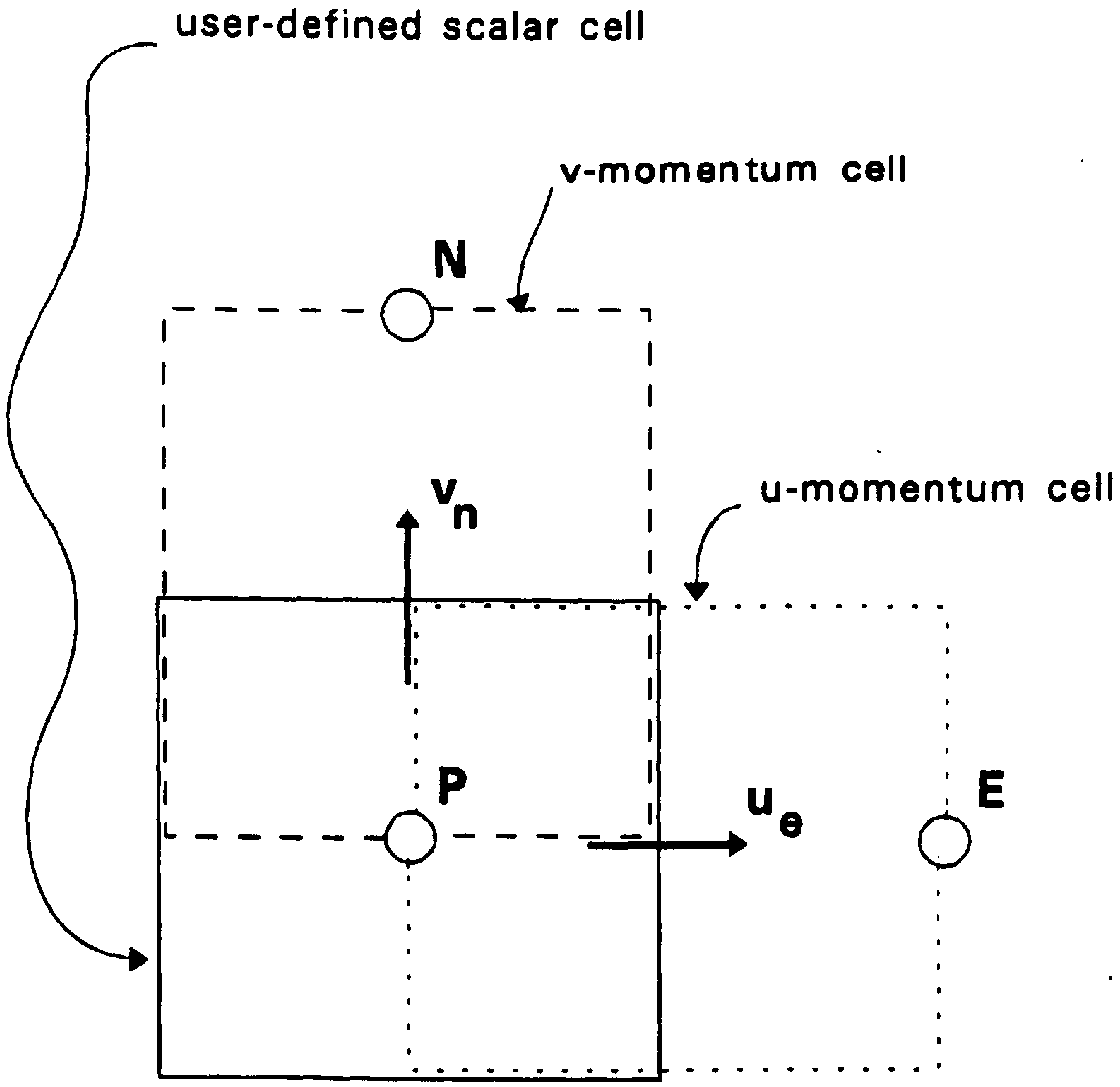


Figure 3.5 *Staggered momentum cells.*

by $\partial u/\partial y = u_B/(\delta y)_b$, since the fluid velocity at the wall is zero. The total shear force acting on the boundary control volume is:

$$\Delta x \Delta z \frac{\mu}{(\Delta y/2)}(0 - u_B)$$

hence the user may input a source *CO*efficient of 1.0 and a *VAL*ue of 0.0. The multiplication factor *f* is seen to be $\Delta x \Delta z \mu / (\Delta y/2)$ and the overall effect is to produce a sink of momentum in the finite volume equations at the boundary cells.

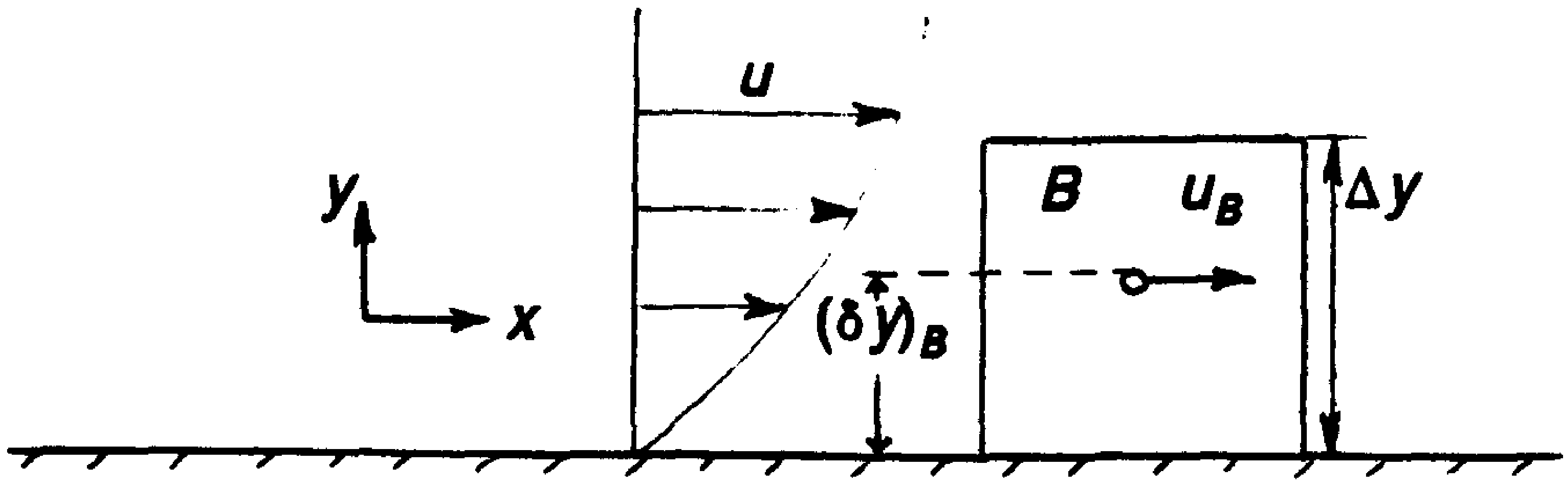


Figure 3.6 Boundary cell in laminar flow.

Here, it is assumed that the boundary condition is linear, however it is possible to input non-linear sources through special GROUND coding for the CO and VAL terms as described in Chapter 4.

3.3 Modelling of the Convection Terms Within PHOENICS

The default scheme within PHOENICS for the discretisation of convective-diffusive transport is the **hybrid** scheme(35). The hybrid scheme possesses the benefits of **central difference** approximation accuracy coupled with the stability of the **upwind** scheme(35). The following example describes the hybrid-upwind formulation. Consider the 1-D convection-diffusion problem, with no sources, as related to Figure 3.7:

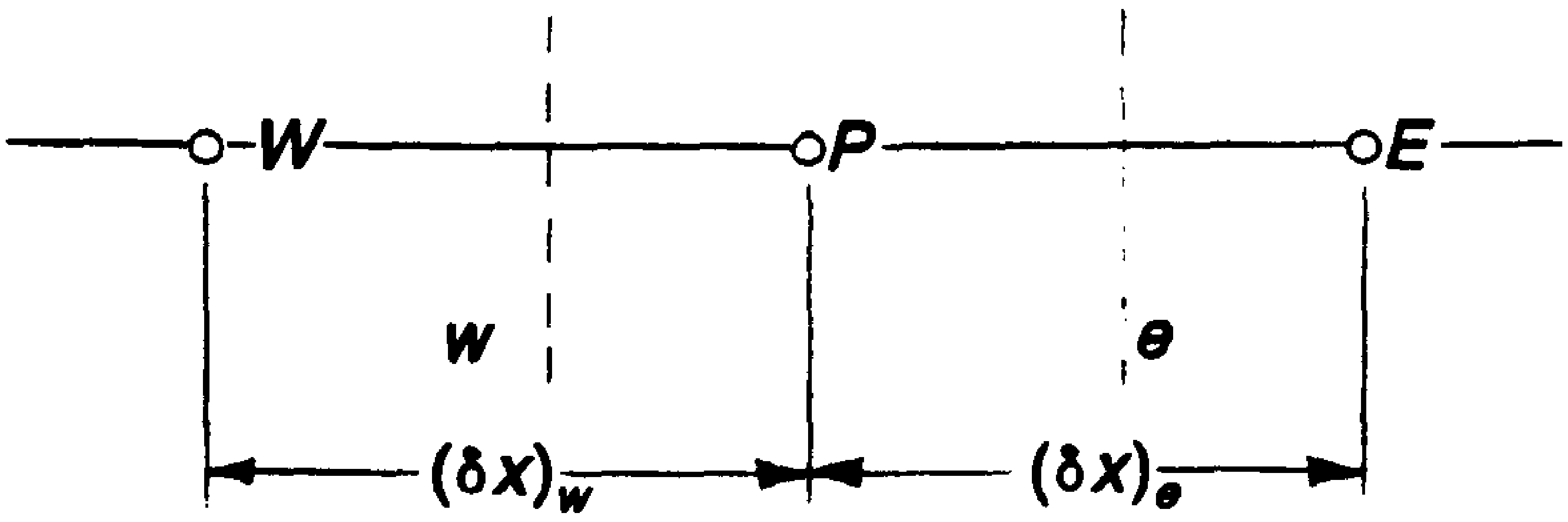


Figure 3.7 Grid cell layout for 1-D convection-diffusion problem.

The equation describing the transport process may be written as:

$$\frac{d}{dx}(\rho u \phi) = \frac{d}{dx}\left(\Gamma_\phi \frac{d\phi}{dx}\right) \qquad 3.3$$

integrating over the control volume for the solution cell P, i.e.:

$$\int_w^e \frac{d}{dx}(\rho u \phi)\, dx = \int_w^e \frac{d}{dx}\left(\Gamma_\phi \frac{d\phi}{dx}\right) dx \qquad 3.4$$

we obtain

$$(\rho u \phi)_e - (\rho u \phi)_w = \left(\Gamma_\phi \frac{d\phi}{dx}\right)_e - \left(\Gamma_\phi \frac{d\phi}{dx}\right)_w \qquad 3.5$$

Now, assuming a piecewise linear ϕ -profile for the diffusion terms we may represent the gradient values at the cell interfaces in the following manner:

$$(\rho u \phi)_e - (\rho u \phi)_w = \frac{\Gamma_e(\phi_E - \phi_P)}{(\delta x)_e} - \frac{\Gamma_w(\phi_P - \phi_W)}{(\delta x)_w} \qquad 3.6$$

For simplicity we let the mass flux terms $(\rho u)_e = F_e$ and $(\rho u)_w = F_w$, and the diffusive flux terms $\Gamma_e/(\delta x)_e = D_e$ and $\Gamma_w/(\delta x)_w = D_w$, hence equation 3.6 becomes:

$$F_e\phi_e - F_w\phi_w = D_e(\phi_E - \phi_P) - D_w(\phi_P - \phi_W) \qquad 3.7$$

Having discretised the diffusion terms in relation to the solution grid points we must now do the same for the convection terms. The overall discretisation equation for the solution cell P can be written as:

$$a_P\phi_P = a_E\phi_E + a_W\phi_W \qquad 3.8$$

and for the **hybrid** scheme we may represent the convective and diffusive transport of ϕ in the following manner:

$$a_E = D_e\left[\!\left[-Pe, 1 - \frac{Pe}{2}, 0\right]\!\right]$$

$$or \qquad 3.9$$

$$a_E = \left[\!\left[-F_e, D_e - \frac{F_e}{2}, 0\right]\!\right]$$

and

$$a_W = D_w\left[\!\left[Pe, 1 + \frac{Pe}{2}, 0\right]\!\right]$$

$$or \qquad 3.10$$

$$a_W = \left[\!\left[F_w, D_w + \frac{F_w}{2}, 0\right]\!\right]$$

where the symbol $[\![\ ]\!]$ stands for the largest of the quantities contained within it and Pe is the Peclet number (or cell Reynolds number) defined as the ratio of the strengths of convection to diffusion i.e. $Pe = F_e / D_e$. It can be seen that for cell Peclet numbers lying in the range $-2 \leq Pe \leq 2$, i.e. where the strengths of convection and diffusion are assumed to have approximately equal magnitude, the discretisation equation will take the form of a **central difference** approximation involving both upwind and downwind neighbour cells. Outwith this range convective transport is assumed to dominate diffusive transport and the **hybrid** scheme reduces to the **upwind** formulation where:

$$\begin{aligned} &a_E = [\![-F_e, 0]\!] \quad , \quad a_W = [\![F_w, 0]\!] \\ &\textit{and} \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad 3.11 \\ &a_P = a_E + a_W \end{aligned}$$

with diffusion terms negated.

Thus for convection-dominated flow regions where the strength of convection exceeds that of diffusion, the hybrid scheme defaults to the upwind scheme with diffusion terms being neglected.

3.4 Solution of the Flow Field

3.4.1 Non-Linearity

The conservation equations solved by PHOENICS may be non-linear in nature. An example of this non-linearity is the convection terms in the momentum equations. When the convection terms are discretised, the coefficients of the finite-volume equations will themselves be functions of the solution dependent variable. As such an iterative solution of the algebraic equations is required with regular updating of the finite-volume coefficients as the solution progresses.

3.4.2 The SIMPLE Algorithm

The difficulty in the solution of the momentum equations to yield the velocity components is the unknown pressure field; there is no obvious pressure equation. However, it is possible to determine the pressure field **indirectly** using the **continuity equation**. The basic principle is that when the correct pressure field is substituted into the momentum equations, the resulting velocity field satisfies the continuity equation. A 'guess and correct' procedure is adopted where the guessed pressure is continually adjusted in the momentum solution. The pressure modification continues until the continuity equation is satisfied.

This pressure correction process, together with the solution procedure, is known as the **SIMPLE** algorithm(35) (Semi-Implicit Method for Pressure Linked Equations). PHOENICS uses a slight variant of the SIMPLE algorithm known as SIMPLEST(38).

3.5 Turbulence Modelling Within PHOENICS

3.5.1 The Eddy-Viscosity Assumption

The aim of any turbulence model is to attempt to relate the Reynolds stresses of the time-averaged turbulent Navier-Stokes equations in terms of mean-flow quantities. The time-averaged turbulent compressible Navier-Stokes equations for may be written in Cartesian tensor notation:

$$\frac{\partial \overline{u_i}}{\partial t}+\overline{u_j}\frac{\partial \overline{u_i}}{\partial x_j}=-\frac{1}{\rho}\frac{\partial \overline{p}}{\partial x_j}\delta_{ij}+\nu_l\frac{\partial}{\partial x_j}\left(\left(\frac{\partial \overline{u_i}}{\partial x_j}+\frac{\partial \overline{u_j}}{\partial x_i}\right)-\frac{2}{3}\left(\frac{\partial \overline{u_k}}{\partial x_k}\right)\delta_{ij}\right)-\frac{\partial}{\partial x_j}\overline{u_i' u_j'} \qquad 3.12$$

as derived in Appendix B. In the context of this work an **eddy-viscosity** approach is adopted where, in analogy with the molecular viscous stress, the Reynolds (turbulence) stresses are modelled in the following manner(39):

$$-\overline{u_i'u_j'} = \nu_t\left(\frac{\partial \bar{u}_i}{\partial x_j} + \frac{\partial \bar{u}_j}{\partial x_i}\right) - \frac{2}{3}\delta_{ij}k \qquad 3.13$$

where k is the turbulent kinetic energy ($= \frac{1}{2}\overline{u_i'u_i'}$). Substituting equation 3.13 into the momentum equation 3.12 gives:

$$\begin{aligned}\frac{\partial \bar{u}_i}{\partial t} + \bar{u}_j\frac{\partial \bar{u}_i}{\partial x_j} = -\frac{1}{\rho}\frac{\partial \bar{p}}{\partial x_j}\delta_{ij} + \nu_l\frac{\partial}{\partial x_j}\left(\left(\frac{\partial \bar{u}_i}{\partial x_j} + \frac{\partial \bar{u}_j}{\partial x_i}\right) - \frac{2}{3}\left(\frac{\partial \bar{u}_k}{\partial x_k}\right)\delta_{ij}\right)& \\ + \frac{\partial}{\partial x_j}\left(\nu_t\left(\frac{\partial \bar{u}_i}{\partial x_j} + \frac{\partial \bar{u}_j}{\partial x_i}\right) - \frac{2}{3}\delta_{ij}k\right)& \qquad 3.14\end{aligned}$$

re-arranging equation 3.14 gives:

$$\frac{\partial \bar{u}_i}{\partial t} + \bar{u}_j\frac{\partial \bar{u}_i}{\partial x_j} = -\frac{1}{\rho}\frac{\partial \bar{p}}{\partial x_j}\delta_{ij} + \frac{\partial}{\partial x_j}\left[(\nu_l + \nu_t)\left(\frac{\partial \bar{u}_i}{\partial x_j} + \frac{\partial \bar{u}_j}{\partial x_i}\right) - \frac{2}{3}\delta_{ij}k - \frac{2}{3}\nu_l\left(\frac{\partial \bar{u}_k}{\partial x_k}\right)\delta_{ij}\right] \quad 3.15$$

assuming the normal turbulent stress terms $-\frac{2}{3}\delta_{ij}k$ may be included in the pre-decomposed pressure term[40] we may write:

$$\frac{\partial \bar{u}_i}{\partial t} + \bar{u}_j\frac{\partial \bar{u}_i}{\partial x_j} = -\frac{1}{\rho}\frac{\partial \bar{p}}{\partial x_j}\delta_{ij} + \frac{\partial}{\partial x_j}\left[(\nu_l + \nu_t)\left(\frac{\partial \bar{u}_i}{\partial x_j} + \frac{\partial \bar{u}_j}{\partial x_i}\right) - \frac{2}{3}\nu_l\left(\frac{\partial \bar{u}_k}{\partial x_k}\right)\delta_{ij}\right] \qquad 3.16$$

Re-arranging equation 3.16 we can write:

$$\frac{\partial \bar{u}_i}{\partial t} + \bar{u}_j\frac{\partial \bar{u}_i}{\partial x_j} = -\frac{1}{\rho}\frac{\partial \bar{p}}{\partial x_j}\delta_{ij} + \frac{\partial}{\partial x_j}\left[\nu_t\left(\frac{\partial \bar{u}_i}{\partial x_j} + \frac{\partial \bar{u}_j}{\partial x_i}\right) + \nu_l\left(\frac{\partial \bar{u}_i}{\partial x_j}\right) - \frac{1}{3}\nu_l\left(\frac{\partial \bar{u}_k}{\partial x_k}\right)\delta_{ij}\right] \qquad 3.17$$

and for incompressible flow the term $\partial\bar{u}_k/\partial x_k = 0$, thus equation 3.17 may be written as:

$$\frac{\partial \overline{u_i}}{\partial t} + \overline{u_j}\frac{\partial \overline{u_i}}{\partial x_j} = -\frac{1}{\rho}\frac{\partial \overline{p}}{\partial x_j}\delta_{ij} + \frac{\partial}{\partial x_j}\left[\nu_t\left(\frac{\partial \overline{u_i}}{\partial x_j} + \frac{\partial \overline{u_j}}{\partial x_i}\right) + \nu_l\left(\frac{\partial \overline{u_i}}{\partial x_j}\right)\right] \qquad 3.18$$

Equation 3.18 represents the modelled form of the incompressible turbulent momentum equations.

The **eddy-viscosity** ν_t itself can be dimensionally related to the local turbulent velocity and turbulent length scale$^{(41)(42)}$ by:

$$\nu_t = C_\mu^{\prime} L\sqrt{k} \qquad 3.19$$

where $C_\mu^{\prime}$ is a turbulent diffusion coefficient. From dimensional considerations$^{(40)}$ the dissipation of turbulent kinetic energy may be written as:

$$\epsilon \propto \frac{u^3}{L} \qquad 3.20$$

where u and L are local reference velocity and length scales of the large scale turbulent eddies, hence:

$$\epsilon = C_D \frac{k^{\frac{3}{2}}}{L} \qquad 3.21$$

again, taking $u \equiv \sqrt{k}$ and substituting equation 3.21 into equation 3.19 we eliminate L:

$$\nu_t = C_\mu^{\prime} C_D \frac{k^2}{\epsilon}$$

$$\rightarrow \quad \nu_t = C_\mu \frac{k^2}{\epsilon} \qquad 3.22$$

3.5.2 The k-ϵ Turbulence Model

The modelled form of the transport equations for the turbulent velocity scale($\propto k$) and the turbulent length scale($\propto \epsilon$) take the following form[43]:

k-transport

$$\frac{\partial k}{\partial t} + \overline{u}_j \frac{\partial k}{\partial x_j} = \frac{\partial}{\partial x_j}\left(\frac{\nu_{eff}}{\sigma_k}\frac{\partial k}{\partial x_j}\right) + P_k - \epsilon \tag{3.23}$$

ϵ-equation

$$\frac{\partial \epsilon}{\partial t} + \overline{u}_j \frac{\partial \epsilon}{\partial x_j} = \frac{\partial}{\partial x_j}\left(\frac{\nu_{eff}}{\sigma_\epsilon}\frac{\partial \epsilon}{\partial x_j}\right) + C_1 \frac{\epsilon}{k} P_k - C_2 \frac{\epsilon^2}{k} \tag{3.24}$$

where the **effective viscosity** term is defined as $\nu_{eff} = \nu_l + \nu_t$. The term representing the production of turbulent kinetic energy may be written as[39]:

$$P_k = -\overline{u_i' u_j'}\frac{\partial \overline{u}_i}{\partial x_j}$$

Substituting for the modelled Reynolds stress term (equation 3.13) gives:

$$P_k = \nu_t \left(\frac{\partial \overline{u}_i}{\partial x_j} + \frac{\partial \overline{u}_j}{\partial x_i}\right)\frac{\partial \overline{u}_i}{\partial x_j} \tag{3.25}$$

The empirically derived turbulence constants have the following values:

C_μ	C_1	C_2	σ_k	σ_ϵ
0.09	1.44	1.92	1.0	1.3

3.5.3 Boundary Conditions in Turbulent Flow

Boundary conditions for the momentum, k and ϵ equations at **solid boundaries** are provided using the universal log-law wall function:

$$\frac{u}{u_*} = \frac{1}{\kappa}\ln\left(\frac{u_* y}{\nu}\right) + C \qquad 3.26$$

where C and κ are constants, y is the distance from the wall and u_* is the shear velocity ($=\sqrt{\tau_0/\rho}$, where τ_0 is the wall shear stress). The use of this universal velocity profile provides a computationally-inexpensive link from the boundary cell node, through the turbulent boundary layer, to the solid surface. In the momentum equations the wall shear stress is calculated, based on a local skin friction coefficient, to provide a sink of momentum at the boundary cell. The resulting skin friction coefficient and node velocity are used to estimate the turbulence kinetic energy and turbulence dissipation rate within the boundary cell.

At **free boundaries**, realistic values of k and ϵ (and hence ν_t) may be estimated from equations relating the mean inlet velocity and a suitable length scale.

CHAPTER 4 LAMINAR VORTEX SHEDDING

4.1 Introduction

The computational analysis of laminar vortex shedding from a rectangular cylinder was undertaken. Such an analysis was executed as it was considered to represent a stepping stone to the eventual desired PHOENICS simulation of unsteady turbulent flow within the modelled vortex shedding flowmeter geometry. The Q1 file for this study is presented in Appendix C.

4.2 Geometrical Configuration

The geometrical configuration for the laminar vortex shedding problem is shown in Figure 4.1. The geometry is identical to that examined by Davis *et al*(33) where a rectangular bluff body is positioned in the centre of a two-dimensional plane channel whose containing walls lie a distance **H** apart. The bluff body itself has a length **A** and a width **B** with an aspect ratio **A/B** of 1.0. The top and bottom edges of the bluff body lie parallel with the containing walls. A blockage ratio is defined as the ratio **B/H** and has a value of 1/4.

A non-uniform computational mesh of 75X by 40Y was employed such that a direct comparison could be made with the results of Davis *et al*. The computational mesh in the region of vortex shedding is shown in Figure 4.2. In the study of Davis *et al* a similar non-uniform grid of 76X by 42Y was used.

4.3 Numerical Model

4.3.1 Governing Equations

The unsteady incompressible fluid flow was modelled by partial differential equations describing the conservation of mass and the conservation of momentum in two rectangular Cartesian coordinate directions:

conservation of momentum:

$$\frac{\partial \bar{u}_i}{\partial t} + \bar{u}_j \frac{\partial \bar{u}_i}{\partial x_j} = -\frac{1}{\rho} \frac{\partial \bar{p}}{\partial x_i} \delta_{ij} + \frac{\partial}{\partial x_j} \left[\nu_l \left(\frac{\partial \bar{u}_i}{\partial x_j} \right) \right] \qquad 4.1$$

conservation of mass:

$$\frac{\partial}{\partial x_j} (\rho \bar{u}_j) = 0 \qquad 4.2a$$

4.3.2 Boundary Conditions

At the free boundary on the upwind faces of the computational domain a fully developed laminar velocity profile was implemented which has a maximum velocity U_0 at the channel centre-line. A similar velocity profile was applied at the downwind free boundary, together with a uniform pressure prescription. At solid surfaces the no-slip condition was employed for parallel velocity components.

For the flow, the Reynolds number is defined as $Re = U_0 B / \nu$. Following Davis *et al* a non-dimensional analysis is considered where all lengths are normalised with respect to B, all velocities with respect to U_0 and time with respect to B / U_0 . The dimensionless Strouhal number which relates the vortex shedding frequency f to a characteristic dimension and velocity is defined as $Str = fB/U_0$.

4.3.3 Convergence Criteria

Convergence was assumed within each particular time step when progressive single cell values of pressure and velocity in the region of vortex shedding showed little change per iteration as the calculation progressed. An examination of the sum of the residual errors for each of the equation sets solved also gave an indication of the degree of convergence.

4.4 Laminar Vortex Shedding Analysis

A transient calculation was performed for a Reynolds number of 500 with the PHOENICS default hybrid scheme in operation. Such a Reynolds number was known to be beyond the critical Reynolds number at which the vortex shedding process would occur(33). On commencement of the impulsively started calculation the growth and development of a symmetrical vortex pair in the early stages of the transience was evident. The growth of this vortex pair was observed to be directly proportional to time to the power 2/3 which is in good agreement with experimental analysis(44). As the flow calculation continued a stage was reached where the growth of the twin vortices had ceased. This flow pattern is shown in Figure 4.3.

At this stage in the physical analysis of such a flow a position of unstable equilibrium has been reached. As the flow proceeds not every small fluctuation within the flow can be damped and this leads to an

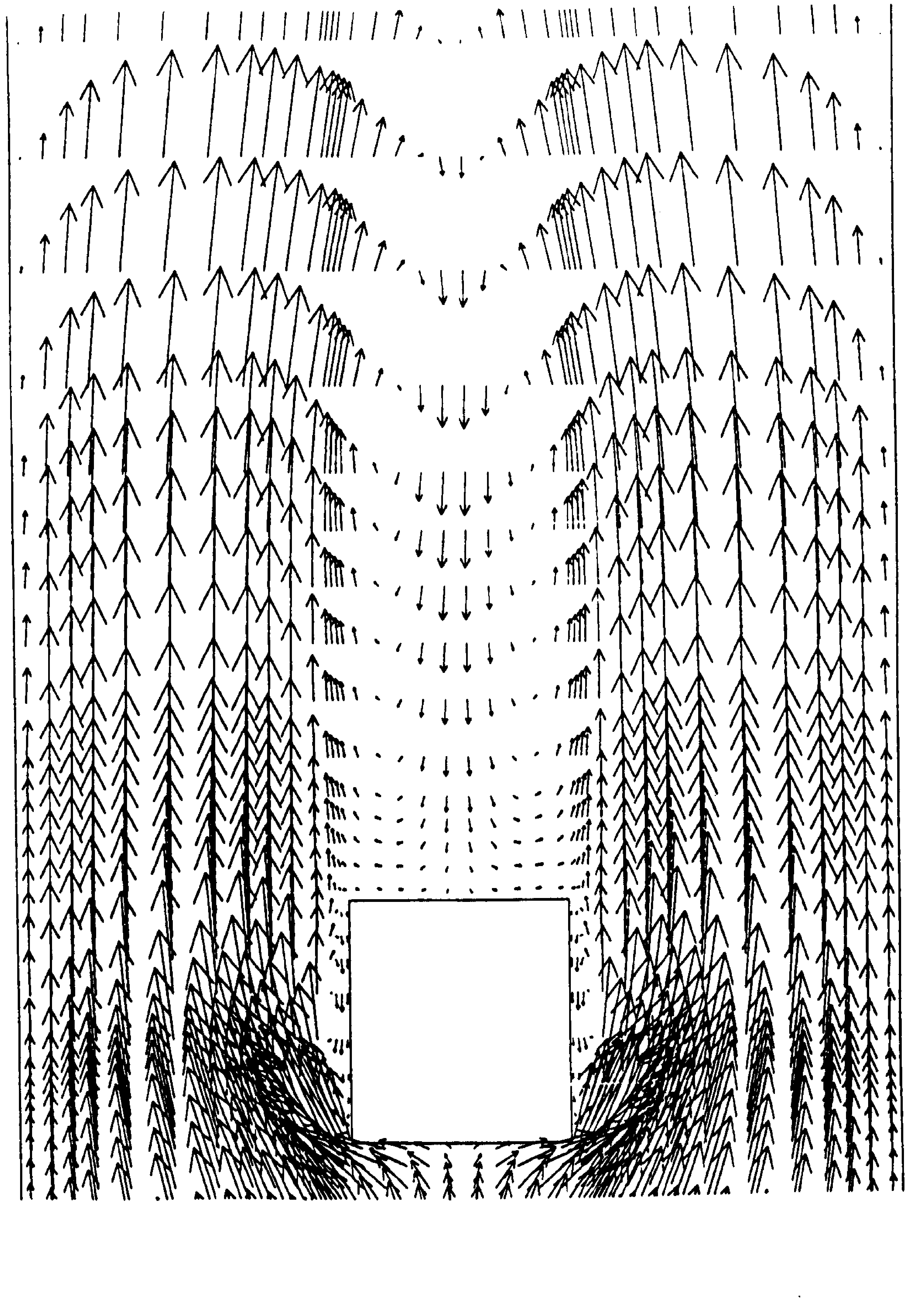


Figure 4.3 Symmetrical vortex pair in wake of bluff body.

asymmetrical eddy pattern. Contrary to expectations, however, the numerical calculation with the hybrid scheme produced only a twin symmetrical vortex pattern of stable, steady size as highlighted in Figure 4.3.

In a real flow the sources of the destabilising small fluctuations may be explained by:

a) non-uniform inlet conditions
b) irregularity of the boundary conditions (e.g. surface roughness)
c) oscillations in the running conditions of the experiment (e.g. sound waves or structural vibrations)

Obviously such destabilising parameters are absent in any numerical study which prescribes symmetric flow geometry, initial and boundary conditions. However, numerical perturbating factors due to round-off or truncation errors are present which may amplify instability and provoke vortex shedding. The work of Braza *et al*(27) shows, however, that such a calculation will always eventually achieve a symmetric flow pattern as the calculation progresses. Anderson *et al*(45) have also considered a similar problem where numerical errors have been allowed to effectively destabilise the Navier-Stokes equation and produce a resultant flow field which resembles its true physical counterpart. However, it was observed that if the time resolution of the calculation is improved then convergence to the initial time-dependent symmetric solution was achieved. Thus, it is apparent that the mathematical solution of the Navier-Stokes equation with symmetric initial conditions should remain symmetric as the flow evolves, even up to higher Reynolds numbers in the region of 1000(27).

The above analyses indicate that an explicit perturbation must be provided in order to compute accurately flows of physical interest, such as vortex shedding. However, it is important to verify whether the unsteady flow pattern produced is dependent upon the nature of the numerically introduced disturbance. Braza *et al*(27) have reported that the numerical perturbations introduced into a flow which is below the critical Reynolds number are effectively damped. When the same conditions are applied to a flow which is now above the critical Reynolds number a continuous periodic wake is formed which is found to be similar to that reported from experimental studies. This appears to indicate that any reasonable attempt at modelling the natural perturbations which promote instability in a real flow will create continuous vortex shedding. The vortex shedding process will occur, in spite of the fact that the nature of the numerical and physical perturbations may be fundamentally different. Such a phenomenon suggests that the periodic character of the flow appearing beyond a critical value of the Reynolds number is an intrinsic property of the Navier-Stokes equation and is independent of the nature of the perturbations. Also, it may be said that such numerical disturbances are responsible solely for changing the flow regime from steady to periodic and that the numerical disturbances need only be applied temporarily even though in the physical process such disturbances are random but always present.

The method employed to initiate the vortex shedding process was to impose a circulation around the bluff body in a steady-flow calculation. This was achieved by removing the no-slip condition around the surfaces of the bluff body, then by applying a velocity to the single cell layer surrounding the body a circulation is achieved. The circulation velocity was approximately equal to that of the incoming maximum velocity. Next, a steady-flow is imposed through the channel and the resulting steady asymmetric flow field is used as the initial conditions for a transient solution of flow past the bluff body without added circulation. This steady asymmetric flow field is shown in Figure 4.4.

The method was a success, in that the calculations showed vortices being shed from the body at a frequency in approximate agreement with expectations, as seen in Figure 4.5. However, the rate of dissipation of the vortices downstream of the body was clearly much too rapid, and after a number of cycles of shedding, the size and strength of the vortices had decayed to very low levels. This process of decay can be represented as a single cell axial velocity transient in the region of vortex shedding, as shown in Figure 4.6. The velocity transient is seen to decay over a period of approximately seven shedding cycles, eventually returning to the steady stable twin vortex wake of Figure 4.3.

Two separate possible causes of this premature vortex decay were identified:

1) the imposed downstream pressure boundary condition of a uniform pressure across the flow channel
2) false diffusion (numerical dissipation)

The first possibility was examined in depth. It was considered that the existence of a fully developed velocity profile at the outlet of the computational domain is uncertain. Such a profile can only be assumed if

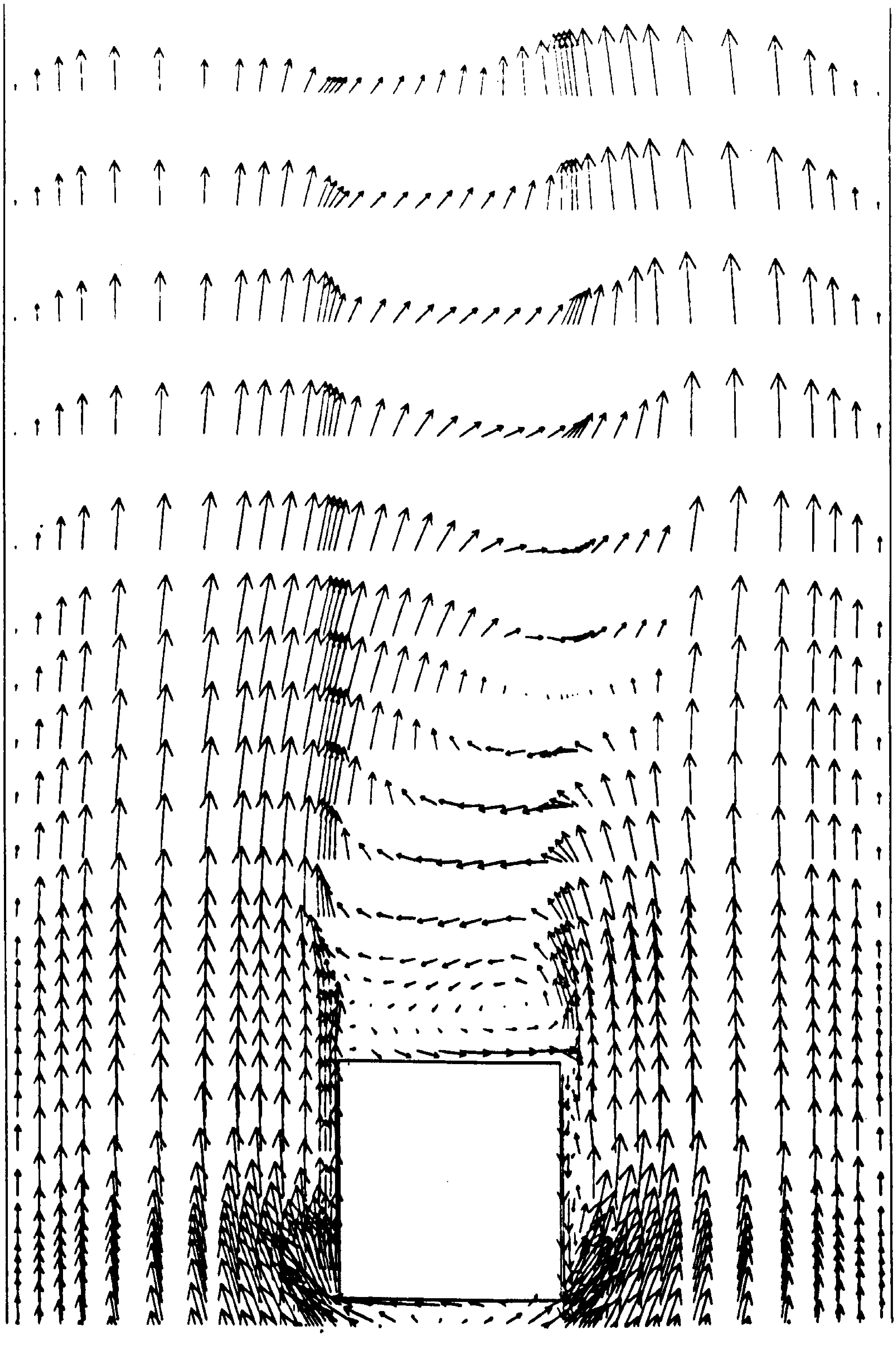


Figure 4.4 Steady asymmetric flow field used as initial conditions.

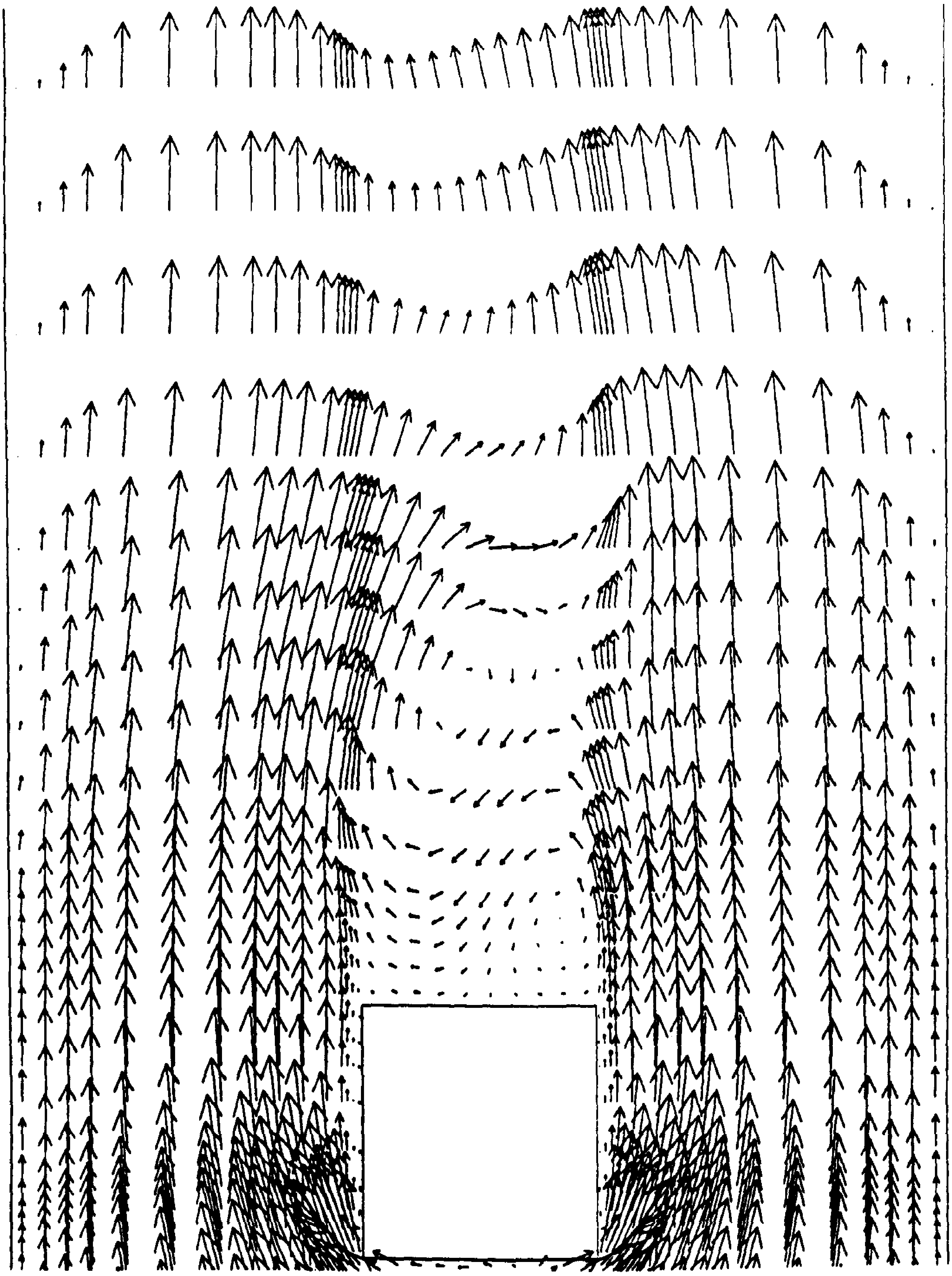


Figure 4.5 Vortex shedding triggered by circulation applied as initial condition.

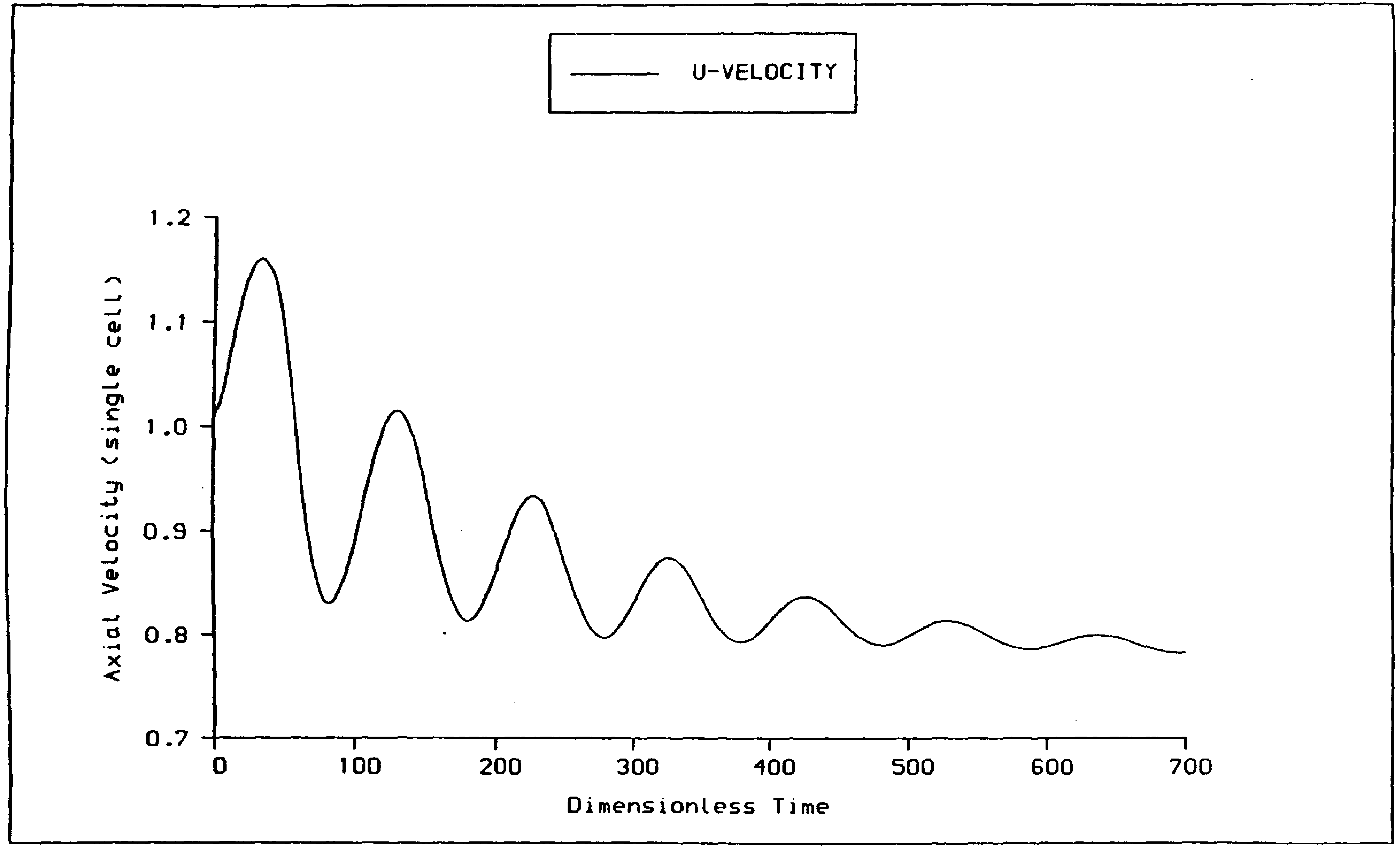


Figure 4.6 ***Axial velocity transient using HYBRID for a single cell located in the region of vortex shedding.***

the distance from the region of vortex shedding is such that the convected vortices will have decayed totally due to the action of viscous dissipation. Following Davis *et al* this problem is overcome by 'stretching' the calculation domain downstream of the bluff body using a finite-to-infinite mapping. This attempt to model infinity allows the prescription of a fully developed velocity profile at the outlet of the computational domain to be assumed with a reasonable degree of confidence. Such 'stretching' of the calculation domain thus allows a uniform pressure prescription on the downwind free boundary. This attempt to reduce the influence of the downstream boundary condition on the flow near to the bluff body was engaged, however, the elimination of the first possibility was concluded as these changes were observed to have little effect on the vortex decay.

The large amount of false diffusion apparently responsible for the rapid vortex decay is associated with the first order accurate upwind scheme used in PHOENICS for the estimation of the convection terms in the momentum conservation equations. Under this scheme, when the direction of flow is inclined at an angle to a computational cell (as must be the case in some regions for any numerical representation of a vortex), the assumed contributions of convection from neighbouring cells to the discretised conservation equation for the cell in question are of poor accuracy. The resulting errors manifest themselves as an apparent increase in the exchange coefficient of the conservation equation with a resultant increase in the diffusion of the conserved species normal to the streamlines. The magnitude of the false diffusion coefficient has been estimated(53) as:

$$\Gamma_{false} = \frac{\rho U \Delta x \Delta y \sin 2\theta}{4(\Delta y \sin^3\theta + \Delta x \cos^3\theta)} \qquad 4.2b$$

hence, the degree of false diffusion is seen to be a function of the oblique angle of the velocity vector from the x-direction and the cell dimensions. This effect has been described in detail by Patankar(35). It was thus concluded that an alternative to the upwind scheme was necessary. The development of this aspect of the computational software is described in the following section.

4.4.1 The SUCCA2D Scheme

4.4.1.1 Introduction

One of the main constraints within the PHOENICS code is that the grid cell cluster on which the discretised conservation equations are based comprises only five cells for a two-dimensional problem; a central grid cell **P** and four immediately adjacent neighbour cells **N,S,E** and **W**, as shown in Figure 4.7. The use of this simple cluster allows the use of simple and efficient numerical equation solvers. It presents no special difficulty if the conventional first- order upwind scheme is used for the discretisation of convective transport, since the conditions in the centre cell are assumed to be influenced only by those four neighbours.

Improvement of the accuracy of the numerical scheme necessitates that additional or nearby neighbouring grid cells are taken into account when forming the discretised conservation equation for a particular cell. The use of such schemes is possible in PHOENICS, but only by including the influence of cells other than the four immediate neighbours in the source term of the equation, rather than by increasing the number of terms. When this is done, care must be taken to ensure that the equations remain conservative.

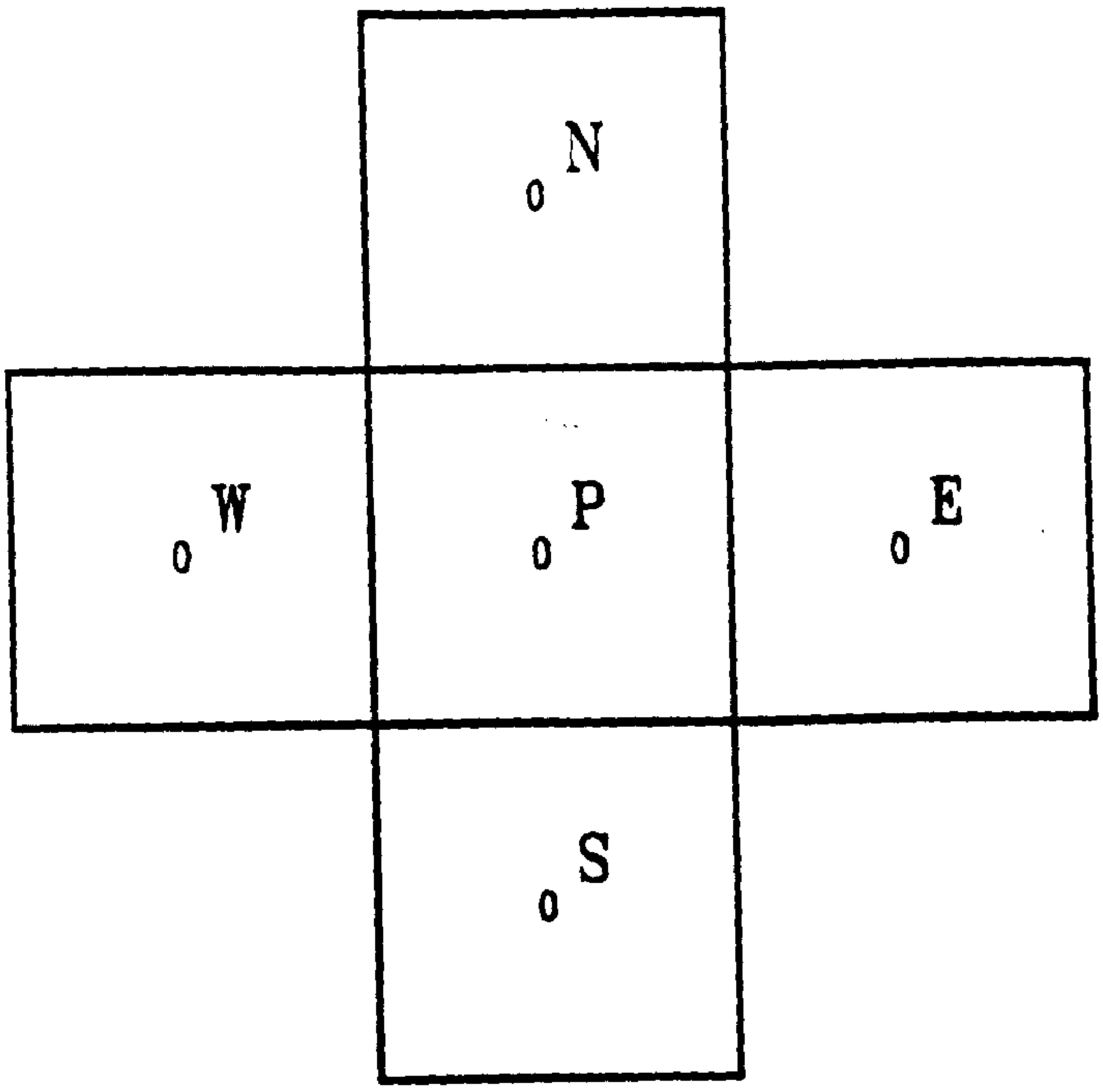


Figure 4.7 *PHOENICS grid cell cluster.*

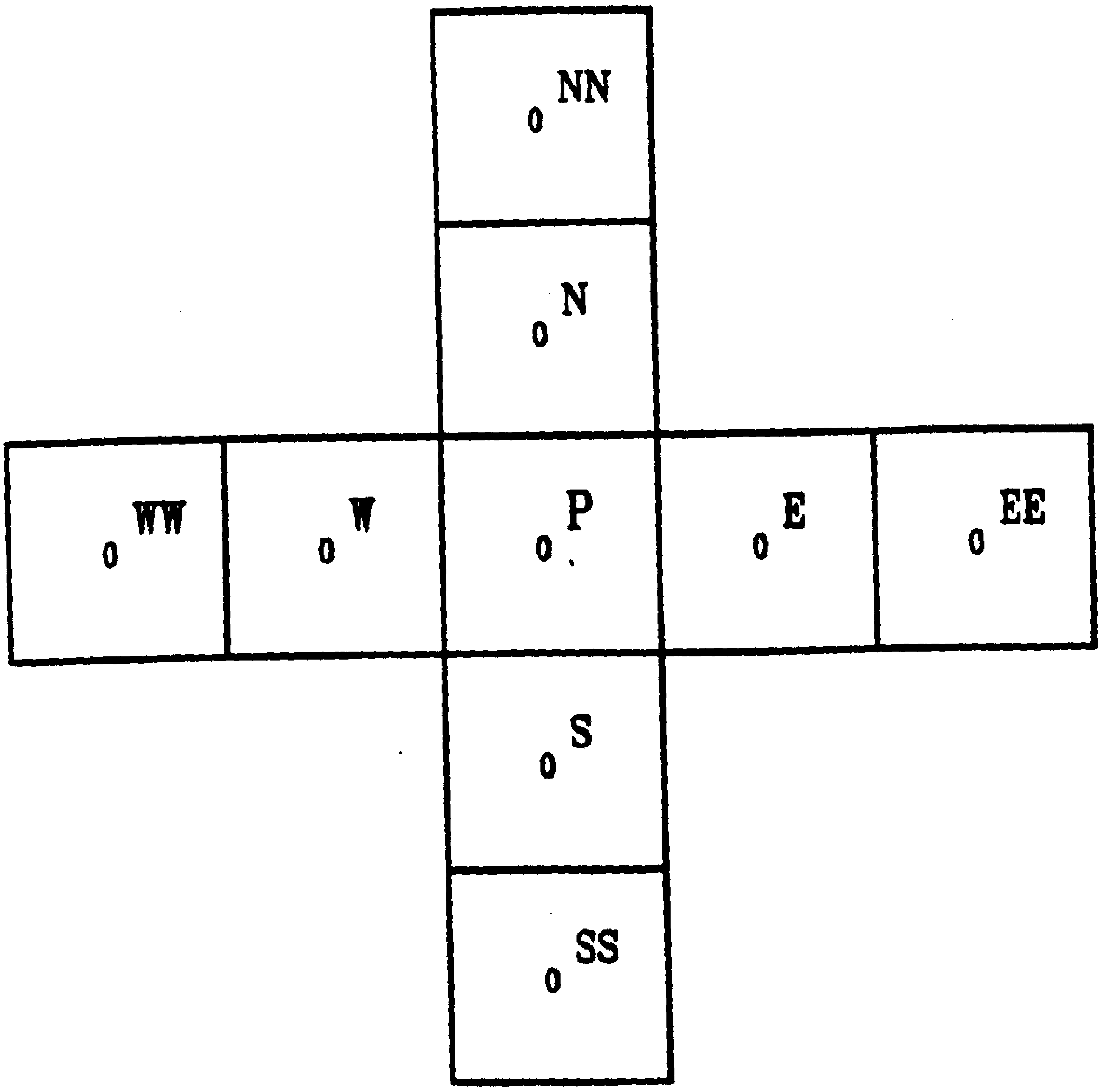


Figure 4.8 *QUICK grid cell cluster.*

Many alternative schemes have been proposed. For example, the QUICK scheme of Leonard(34) extends the cluster along the lines of grid cells to include four additional cells (Figure 4.8). Although this scheme improves the accuracy of gradient estimates along the lines of grid cells, it does not explicitly recognise the fact that flow variables are conserved in the streamwise direction; false diffusion errors can still result, and it also suffers from stability problems under certain circumstances(46).

The SKEW scheme devised by Raithby(47) and the CUPID scheme of Patel *et al*(46) include the corner cells (Figure 4.9). Raithby's scheme has now been demonstrated to be non-conservative and unstable(46). In the CUPID scheme the convection terms in the conservation equations are modelled in terms of the local direction of the streamlines. The developed scheme presented in this work is a modified version of the CUPID scheme and a comparison of these two flow-oriented schemes is discussed.

4.4.1.2 An Alternative Convection Scheme - SUCCA2D

Prior to analyzing the criteria required for the convection algorithm, the following basic rules should apply to the computational environment within which the alternative scheme will exist. These rules specify that the general numerical scheme should(35):

- ensure flux consistency at cell faces, i.e. the flux leaving through a specific cell face must be equivalent to that entering the adjacent cell through their common face.
- always produce positive coefficients to promote a stable, diagonally

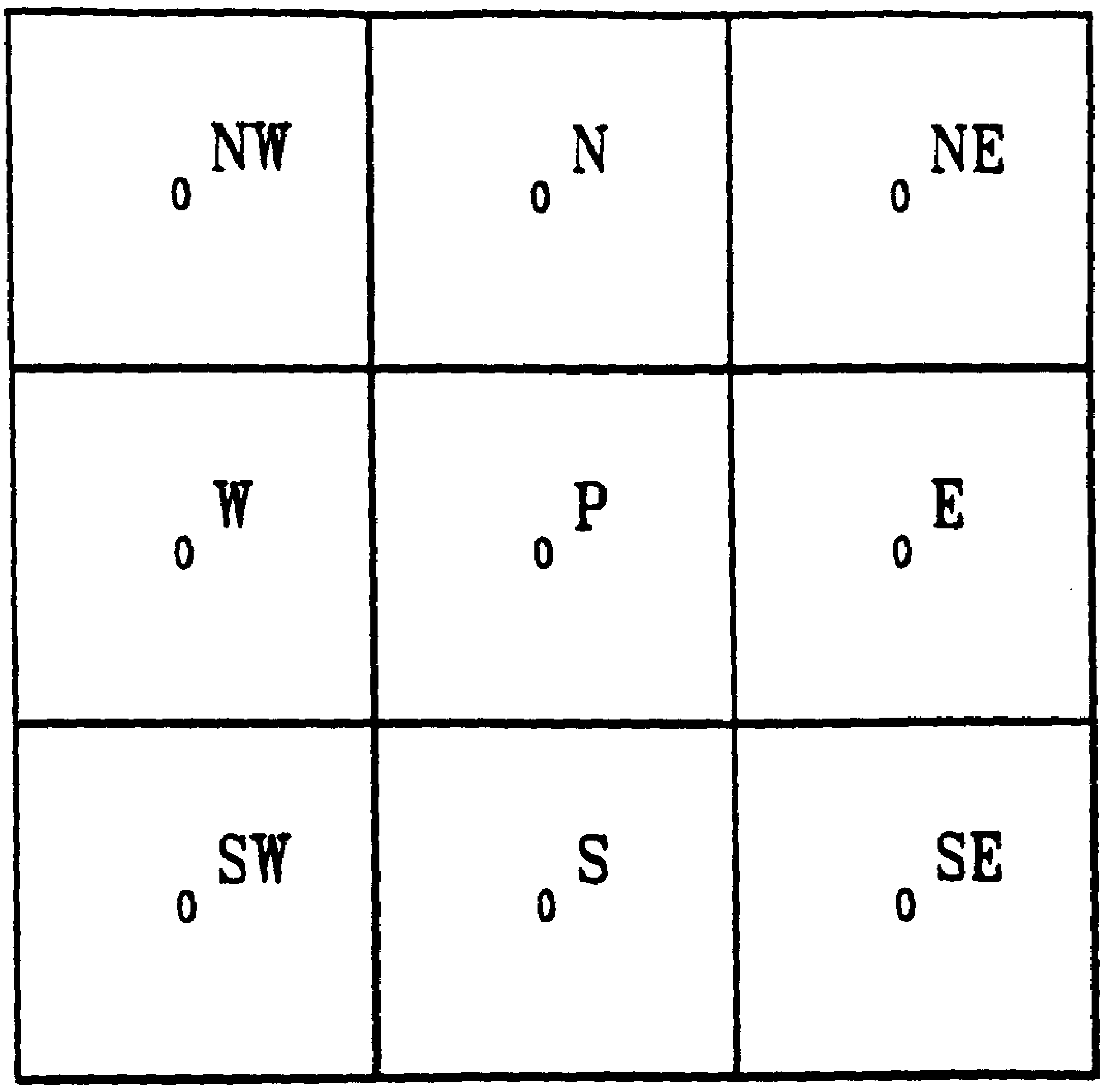


Figure 4.9 *SKEW/CUPID grid cell cluster.*

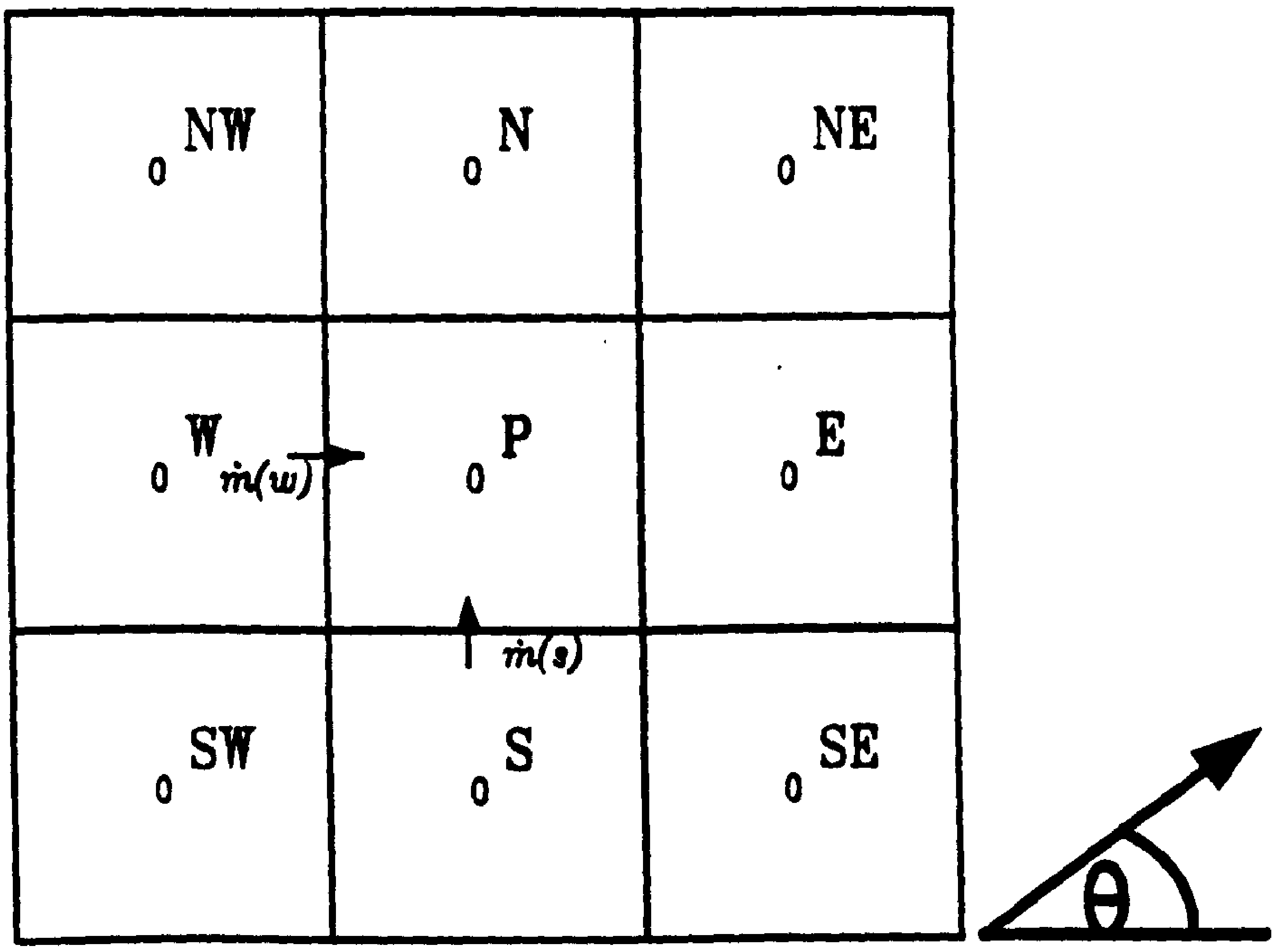


Figure 4.10 *SUCCA2D grid cell cluster.*

dominant coefficient matrix.

- ensure the conservation principle is maintained.
- treat sources in a manner such that the incidence of unbounded solutions is reduced. Broadly, this entails linearizing the source term with a negative slope.
- to be consistent with the differential transport equation the coefficients for the solution cell P must obey the relation $a_P = \sum a_{nb}$.
- obey sufficient criteria to promote a convergent solution, one such criterion being known as the Scarborough criterion:

$$\frac{\sum |a_{nb}|}{|a_P|} \quad \begin{array}{l} \leq 1 \text{ for all equations} \\ < 1 \text{ for at least one equation} \end{array}$$

note, this is a sufficient criterion and is not a pre-requisite to ensure convergence.

With the previously described limitations of earlier schemes in mind any alternative algorithm to reduce the effects of multi-dimensional false diffusion should satisfy the following criteria, i.e. the alternative scheme should:

- remain unconditionally conservative.
- be formulated in such a manner as to produce positive coefficients (thus reducing the risk of potential oscillatory behaviour, e.g. the SKEW scheme).
- be relatively easy to implement and computationally inexpensive to run.

With the satisfaction of the above constraints, the algorithm should be constructed, following the CUPID formulation, such that it will comply with the following two additional demands, namely:

a) for a zero skew flow angle the normal hybrid-upwind scheme should be defaulted for the convection terms

b) when the incoming cell mass fluxes are equal (flow angle $\theta = 45°$) **all** the contributions of the conserved species into the solution cell should come from the upwind corner cell.

The alternative scheme has been formally titled **SUCCA2D** (**Skew Upwind Corner Convection Algorithm 2D**) and is applied within the nine cell grid cluster shown in Figure 4.10. Considering the SW corner inflow for cell P the SUCCA2D algorithm may be written for the convective transport of the conserved species ϕ as:

$$C_P\phi_P = \left(\begin{array}{c} \left(\dot{m}_w - \dfrac{\dot{m}_s^2}{\dot{m}_w}\right)\phi_W \\ + \left(\dot{m}_s + \dfrac{\dot{m}_s^2}{\dot{m}_w}\right)\phi_{SW} \\ + 0.\phi_S \end{array} \right) \textit{for } 0° < \theta \leq 45° \qquad 4.3$$

i.e. $C_P\phi_P = C_w\phi_W + C_{sw}\phi_{SW}$

and:

$$C_P\phi_P = \left(\begin{array}{c} \left(\dot{m}_s - \dfrac{\dot{m}_w^2}{\dot{m}_s}\right)\phi_S \\ + \left(\dot{m}_w + \dfrac{\dot{m}_w^2}{\dot{m}_s}\right)\phi_{SW} \\ + 0.\phi_W \end{array} \right) \textit{for } 45° < \theta < 90° \qquad 4.4$$

i.e. $C_P\phi_P = C_s\phi_S + C_{sw}\phi_{SW}$

This formulation of the SUCCA2D algorithm satisfies all of the discriminating criteria outlined above and is discussed in section 4.4.1.4. The GROUND FORTRAN coding for the SUCCA2D scheme itself is presented in Appendix L.

4.4.1.3 The SUCCA2D Scheme Applied to the Momentum Equations

Previous flow-oriented schemes such as SKEW/CUPID have concentrated on analyzing the transport of scalar variables within the flow. The following example shows how the SUCCA2D scheme can be applied to the transport of momentum.

Consider the two-dimensional steady state convection-diffusion problem shown in Figure 4.11. The governing equations describing the fluid flow will be:

$$\frac{\partial}{\partial x}(\rho uu) + \frac{\partial}{\partial y}(\rho vu) = \frac{\partial}{\partial x}\left(\mu\frac{\partial u}{\partial x}\right) + \frac{\partial}{\partial y}\left(\mu\frac{\partial u}{\partial y}\right) - \frac{\partial P}{\partial x} \qquad 4.4a$$

$$\frac{\partial}{\partial x}(\rho uv) + \frac{\partial}{\partial y}(\rho vv) = \frac{\partial}{\partial x}\left(\mu\frac{\partial v}{\partial x}\right) + \frac{\partial}{\partial y}\left(\mu\frac{\partial v}{\partial y}\right) - \frac{\partial P}{\partial y} \qquad 4.4b$$

In Figure 4.11 the solution cell is that containing u_p and the local flow is from the SW corner cell with the local flow direction lying within the range $0° < \theta \leq 45°$. The mass fluxes through the west and south faces of the solution cell will be:

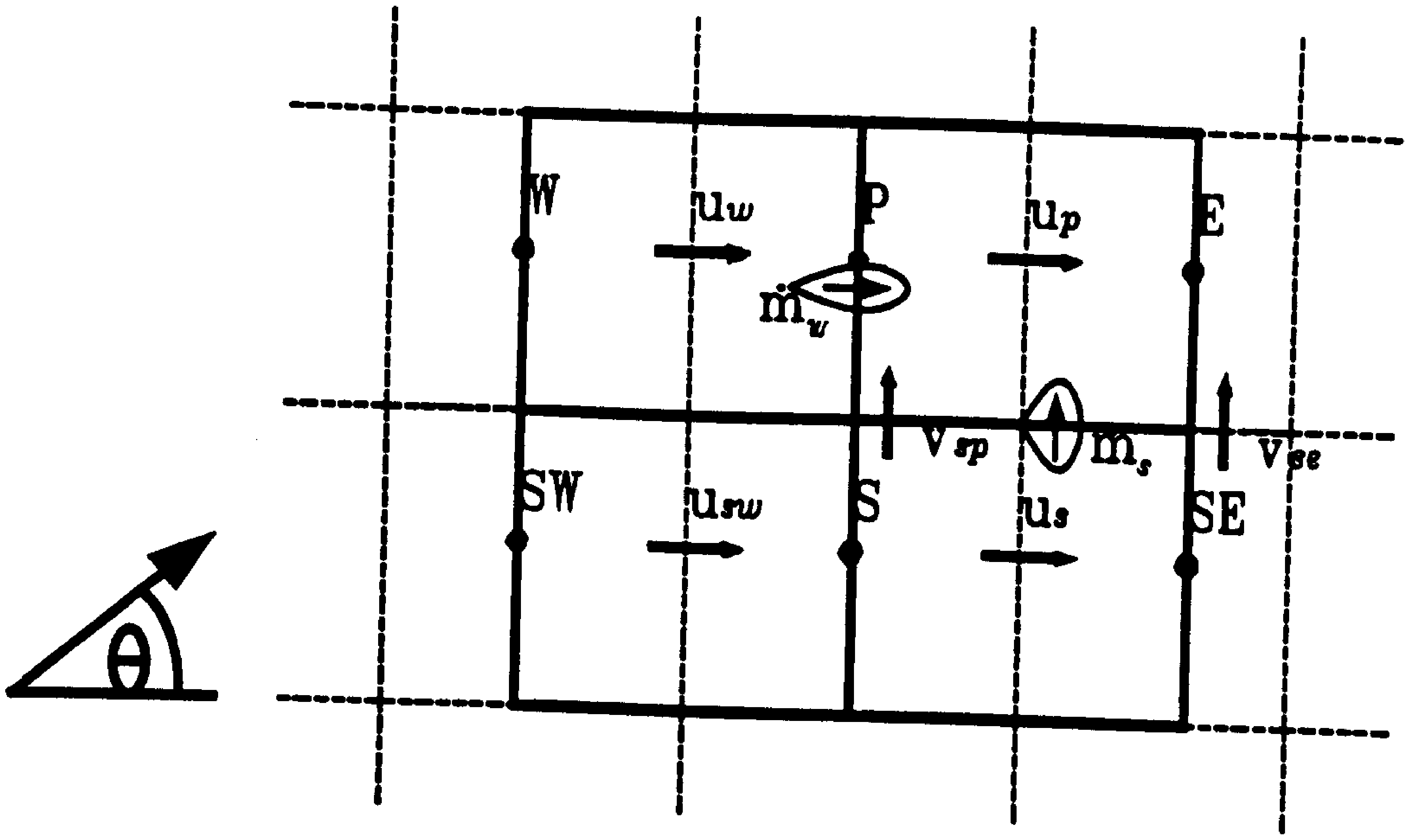


Figure 4.11 *Convection-diffusion problem showing staggered momentum cells.*

$$\dot{m}_w = \rho_P A_P (u_p + u_w)/2 \qquad 4.5$$

$$\dot{m}_s = (\dot{m}_{sp} + \dot{m}_{se})/2 \qquad 4.6$$

On the first iterative sweep through the calculation domain the PHOENICS default hybrid-upwind scheme is used. This is done to achieve initial values for the mass fluxes which are to be modified within the SUCCA2D scheme from the first sweep onwards.

With the default (hybrid) scheme in operation the x-direction momentum equation (equation 4.4a) in discretised form will be:

$$a_p u_p = a_s u_s + a_w u_w + a_n u_n + b + (P_P - P_E) A_p \qquad 4.7$$

where b = *source term*

P = *cell face pressure*

A = *cell face area*

and a_s = *convection plus diffusion terms*

a_w = *convection term only*

a_n = *diffusion term only*

It should also be noted that the downwind convection terms within a_n and a_e have been negated due to the upwinding nature of the hybrid scheme and that the only convection terms are contained within the upwind momentum coefficients a_s and a_w . As it stands, this discretisation process will promote the numerical diffusion of momentum normal to the streamlines.

With the SUCCA2D scheme now implemented on the second sweep the mass fluxes (convection coefficients) are modified according to equation 4.3 and the discretised momentum equation now takes the form:

$$a_p u_p = a_s u_s + a_w^{\prime} u_w + a_n u_n + b_s u_{sw} + b + (P_P - P_E) A_p \qquad 4.8$$

where a_s *and* a_n = *diffusion terms only* and the SUCCA2D scheme sets the following parameters:

$C_s = 0$ (*i.e. the convective part of coefficient* a_s)

$a_w^{\prime} = C_w$ = *modified convection coefficient* = $(\dot{m}_w - (\dot{m}_s)^2 / \dot{m}_w)$

b_s = *created convection coefficient* = $(\dot{m}_s + (\dot{m}_s)^2 / \dot{m}_w) = C_{sw}$

The term $b_s u_{sw}$ is added to the 'main' source term b and the series of these linear algebraic finite volume equations is solved within PHOENICS using the SIMPLEST algorithm.

Figure 4.12 summarises the solution sequence for the SUCCA2D scheme. The local flow in this example is from the SW corner cell with the local flow direction lying within the range $0° < \theta \leq 45°$. This summary flowchart will highlight the:

- negation of the convection coefficient C_s
- modification of the convection coefficient C_w
- creation of the convection coefficient b_s
- identification of the SW cell as the upwind corner cell
- implementation of the value of u in the SW corner cell for the solution of momentum

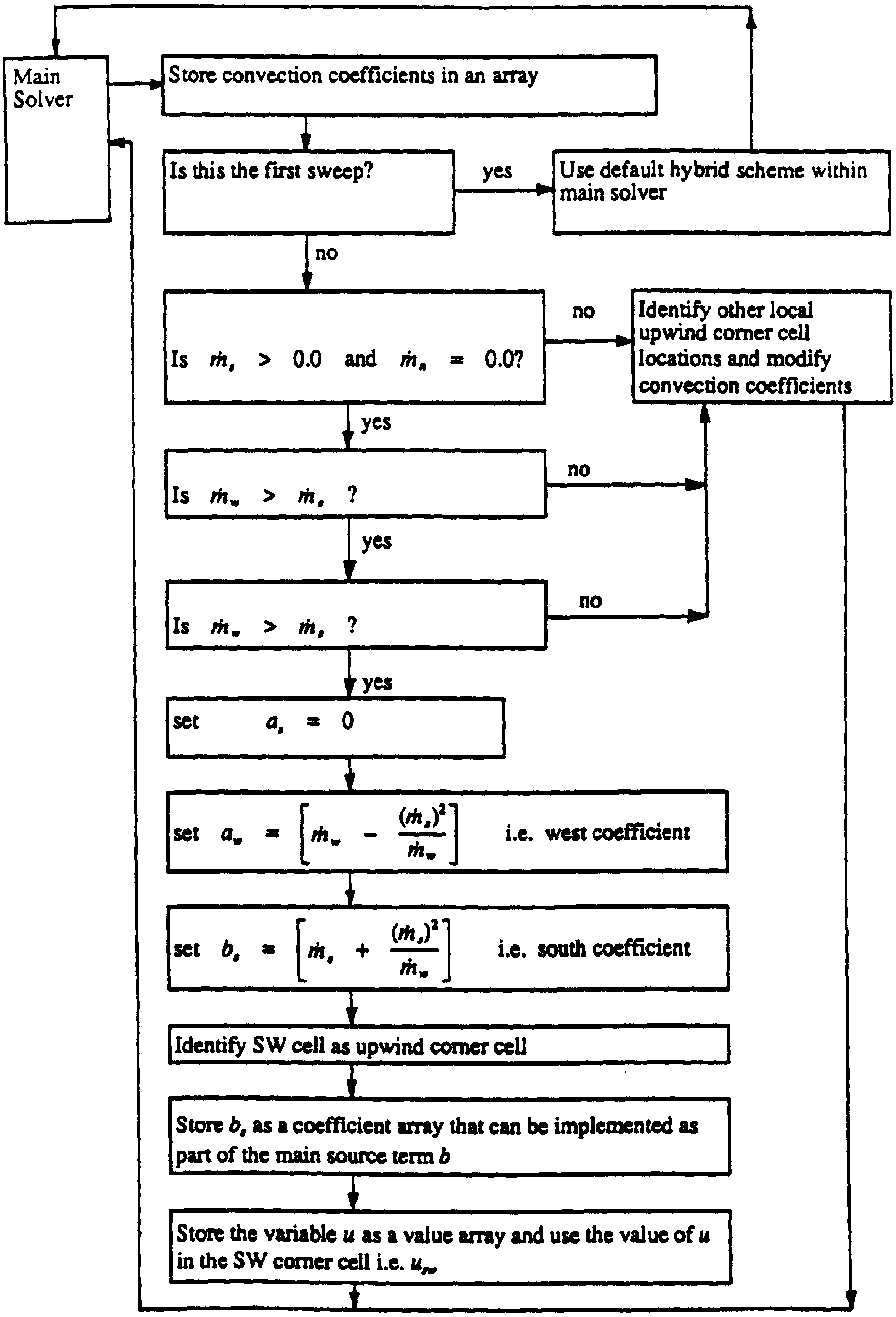


Figure 4.12 *The solution sequence for the SUCCA2D scheme.*

4.4.1.4 A Comparison Between SUCCA2D and CUPID

Adherence to Basic Rules

Just as the CUPID scheme has been formulated to adhere to the rules of any general numerical scheme, as outlined above, the SUCCA2D scheme too is found to comply, although in a slightly different manner. Since the SUCCA2D scheme introduces the influence of upstream variables through the main source term of the finite volume equation then the criterion $a_P = \sum a_{nb}$ is still satisfied but in the form $a_P = \sum a'_{nb} + b$ i.e. in the presence of the modified convection coefficient and the created source term.

This partial decoupling of the influence coefficients leads in turn to the fact that the Scarborough criterion will always be satisfied when the SUCCA scheme is applied.

Numerical Accuracy

In considering the numerical accuracy of both schemes it is perhaps beneficial to look at a graphical representation of the form of the coefficients as described by the upwind, CUPID and SUCCA2D schemes. Consider the scalar grid cell layout shown in Figure 4.13 in which convection is assumed to dominate diffusion and the grid geometry is such that $\Delta x = \Delta y = 1.0$. The flow angle $\theta = 67.5°$ is such that the tangent of the flow angle will represent the ratio of the incoming mass fluxes, i.e.

$$\tan\theta = \frac{\dot{m}_s}{\dot{m}_w} = \frac{2.414}{1.0}$$

For the upwind formulation we may write the finite volume equation for the

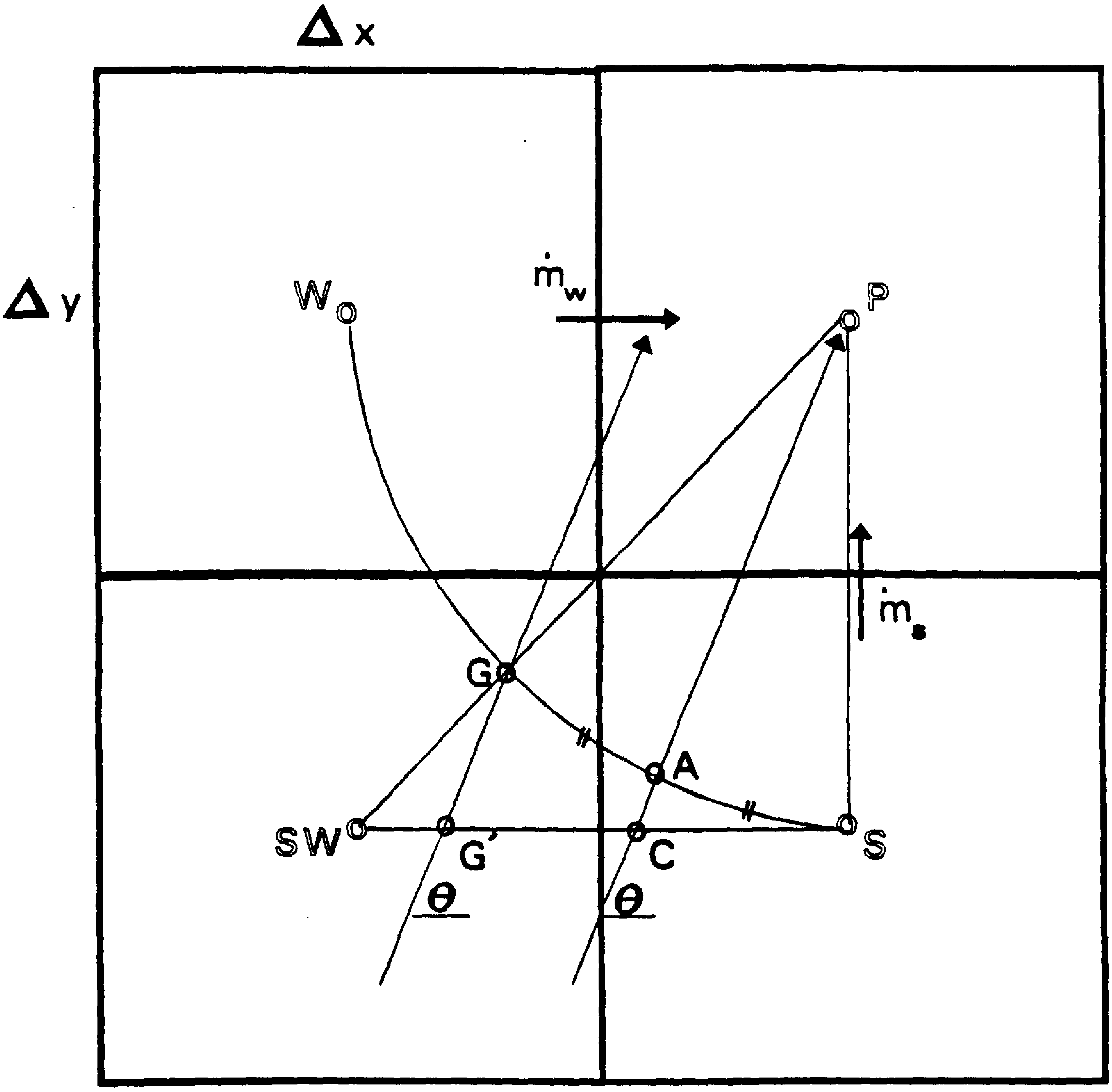


Figure 4.13 *Enhanced CUPID accuracy with SUCCA2D.*

conserved scalar as:

$$\begin{aligned} C_P\phi_P &= C_s\phi_S + C_w\phi_W \\ &= 2.414\,\phi_S + 1.0\,\phi_W \end{aligned}$$

Such an **upwind** assumption of local one-dimensionality will promote the crosswind false diffusion as discussed above.

For the **CUPID** scheme, i.e.

$$C_P\phi_P = \dot{m}_w(1+\tan\theta)\begin{pmatrix} \max\left(\frac{45-\theta}{45},0\right)\phi_W \\ +\max\left(\frac{\theta-45}{45},0\right)\phi_S \\ +\min\left(\frac{\theta}{45},\frac{90-\theta}{45}\right)\phi_{SW} \end{pmatrix} \qquad 4.9$$

we may represent the finite volume equation as:

$$\begin{aligned} C_P\phi_P &= (C_w+C_s)\left[\left(\frac{67.5-45}{45}\right)\phi_S + \left(\frac{90-67.5}{45}\right)\phi_{SW}\right] \\ &= 3.414(0.5\,\phi_S + 0.5\,\phi_{SW}) \\ &= 1.707\,\phi_S + 1.707\,\phi_{SW} \end{aligned}$$

It can be seen from the above equation and as shown in Figure 4.13 that the basis for the coefficient magnitudes are the arc lengths S-A and A-G. These arc lengths are shown to be equal for the flow angle $\theta = 67.5°$ as the velocity vector bisects the S-G arc. In this case, the magnitudes of the convection coefficients are thus equal.

In the SUCCA2D scheme the modified finite volume equation is:

$$C_P\phi_P = \left(2.414 - \frac{1}{2.414}\right)\phi_S + \left(1 + \frac{1}{2.414}\right)\phi_{SW}$$

$$= 2.0\,\phi_S + 1.414\,\phi_{SW}$$

In order to interpret the values of the SUCCA coefficients we must again look at Figure 4.13. It can be observed that the length S-C is equal to $1/\tan 67.5° = 0.414$ hence the length C-SW = 1 - 0.414 = 0.586. The magnitude of the SUCCA coefficients is seen to be directly proportional to the lengths S-C to C-SW i.e.

$$\text{length ratio} = \frac{S\text{-}C}{C\text{-}SW} = \frac{0.414}{0.586} = 0.707$$

$$\text{coefficient ratio} = \frac{C_{sw}}{C_s} = \frac{1.414}{2.0} = 0.707$$

Hence the form of the coefficients chosen for the SUCCA2D scheme is based upon a linear interpolation of lengths. In the above example these lengths are formed using the velocity vector for the solution cell P, extrapolated from the solution cell grid point, such that it intersects the horizontal line connecting the two immediate downwind cell grid points as shown in Figure 4.13.

For the SUCCA2D scheme to achieve an identical result to that obtained for the CUPID scheme in the above example, a necessary adjustment would have to be made. In order that the SUCCA scheme may obtain equality of coefficients on a linear scale which is based on the true physical distance between the two concerned upwind points would imply that ϕ_{sw} exists at $\phi_{G'}$ which is not the case. It may be observed that $\phi_{sw} = \phi_{G'}$ will only occur at $\theta = 45°$.

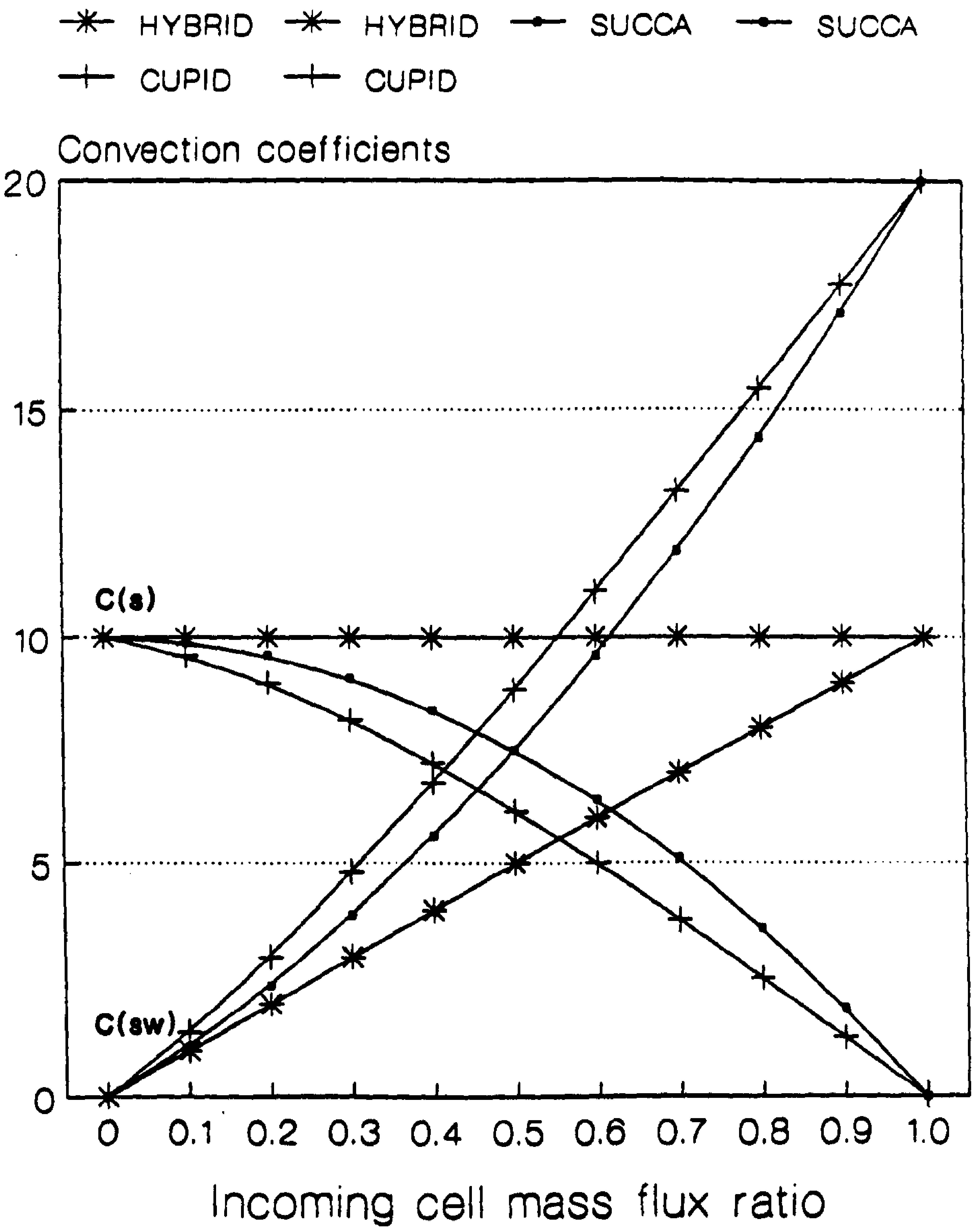
It would also seem reasonable, in a physically intuitive regard, that for the flow angle of $\theta = 67.5°$ one would anticipate, even under modified conditions, that a higher degree of influence would be appropriate from the southern cell containing ϕ_S , as the SUCCA scheme predicts. Such reasoning is also emphasised by analyzing Figure 4.14. It can be seen that for equality of coefficients C_s and C_{sw} using the SUCCA2D scheme the ratio of incoming mass fluxes is 0.5. This implies that the local velocity vector passes through the mid-point of the downstream horizontal line connecting the grid points and as such one would expect the value of ϕ_P to be the arithmetic average of the two downwind ϕ 's.

The SUCCA2D scheme may thus be considered to enhance the numerical accuracy of the CUPID formulation due to the fact that the value of ϕ in the upwind corner cell can be explicitly included within the SUCCA formulation.

Transition to Hybrid

One of the additional constraints within any alternative convection scheme was that when the incoming cell mass fluxes were equal, all of the contributions of ϕ into the solution cell should come from the upwind corner cell. Obviously, it would be desirable for the new scheme to approach this constraint within a gradual transition. This transition can be observed for both schemes in Figure 4.14 where the modified convection coefficients for both schemes are plotted against mass flux ratio (or flow angle) for a SW corner flow specification of $45° < \theta < 90°$. The modified convection coefficients may be written as:

Fig.4.14 Transition to HYBRID for SUCCA and CUPID



$$C_s = \left(\dot{m}_s - \frac{(\dot{m}_w)^2}{\dot{m}_s} \right)$$

and

$$C_{sw} = \left(\dot{m}_w + \frac{(\dot{m}_w)^2}{\dot{m}_s} \right)$$

It can be seen in Figure 4.14 that at shallow angles (when we wish to approach the normal hybrid-upwind discretisation scheme) the SUCCA2D scheme provides a smooth, tangential transition to hybrid. The more abrupt CUPID curve can be explained by the numerical accuracy error associated with the scheme's convection formulation as detailed above.

The SUCCA2D scheme was applied to two standard benchmark test problems. In the first analysis a backward facing step flow is considered in order to analyze the artificial diffusion of momentum. Secondly, the numerical diffusion of a scalar variable is considered.

4.4.1.5 Analysis of Backward Facing Step Flow

The SUCCA2D scheme was applied to a two-dimensional steady state laminar flow backward facing step problem. The analysis of this benchmark case was chosen as previous studies(37)(48) had encountered the effects of numerical diffusion. This diffusive phenomenon was seen to occur within the recirculation region produced by the flow, where the local flow direction is skewed to the computational cells. The error was observed to manifest itself as an artificial increase in the exchange coefficient (in this case, molecular kinematic viscosity) with the consequent increase in the diffusion of the conserved species (in this case, momentum) normal to the streamlines. The overall effect is to produce an underpredicted recirculation zone length. The Q1 file for this study is presented in Appendix D.

Geometry and Results

For this test case the geometry was modelled according to an experimental analysis by Armaly *et al*[49] as shown in Figure 4.15. The computational mesh in the step region is shown in Figure 4.16. For a Reynolds number (based on step height) of 200, the PHOENICS default hybrid scheme, in conjunction with grid refinement tests, produced a recirculation region whose length was underestimated by some 10 per cent in comparison with the experimental data of Armaly *et al*.

When the calculation was repeated using the same geometry and boundary conditions, but with the SUCCA2D scheme implemented for the transport of the U and V momentum components, the discrepancy between calculation and experiment was of the order of 2 per cent. Figure 4.17 shows contours of zero axial velocity, highlighting the re-attachment point lengths for both schemes. Axial velocity profiles are plotted at various locations downstream of the step in Figures 4.17a to 4.17c. These plots highlight the differences in the hybrid and SUCCA2D solutions.

As the application of the SUCCA2D scheme involves altering the overall solution sequence, an increase in CPU time of the order of 20 per cent was experienced, however, this increase was not considered to be excessive in comparison with the accuracy achieved.

4.4.1.6 Scalar Transport Analysis

The scalar transport analysis considered in this section represents

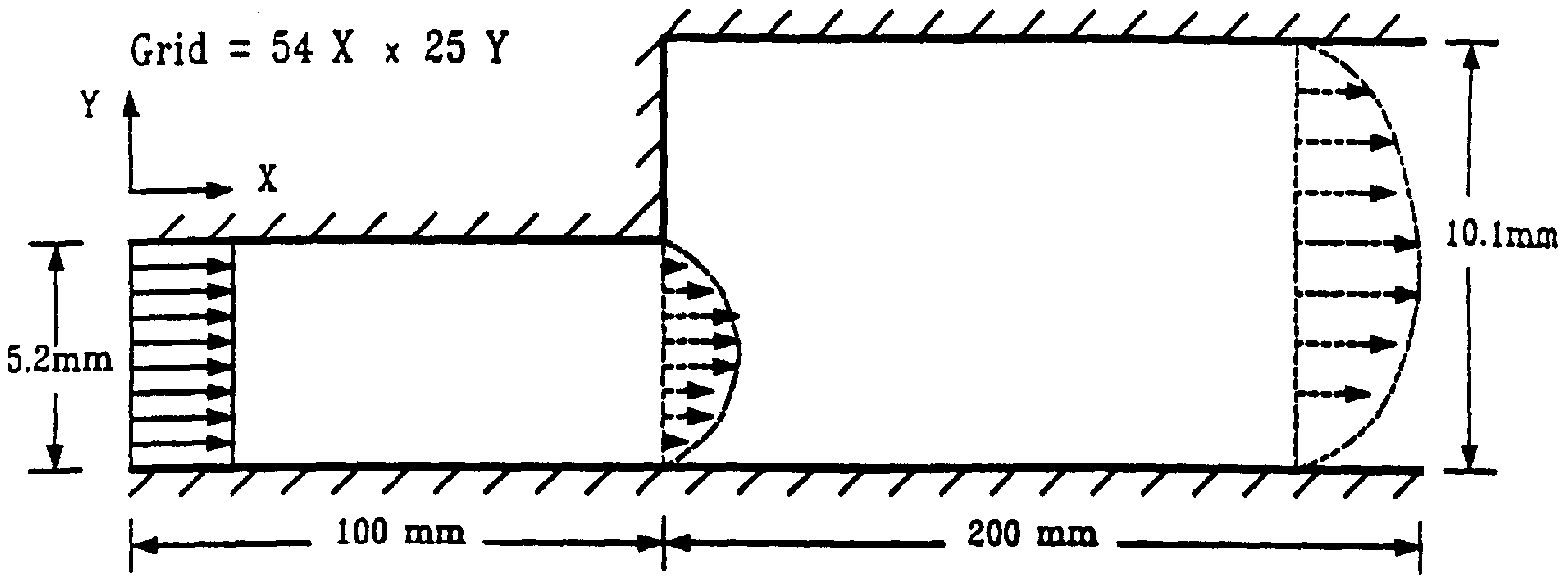


Figure 4.15 *Backward-facing step geometry.*

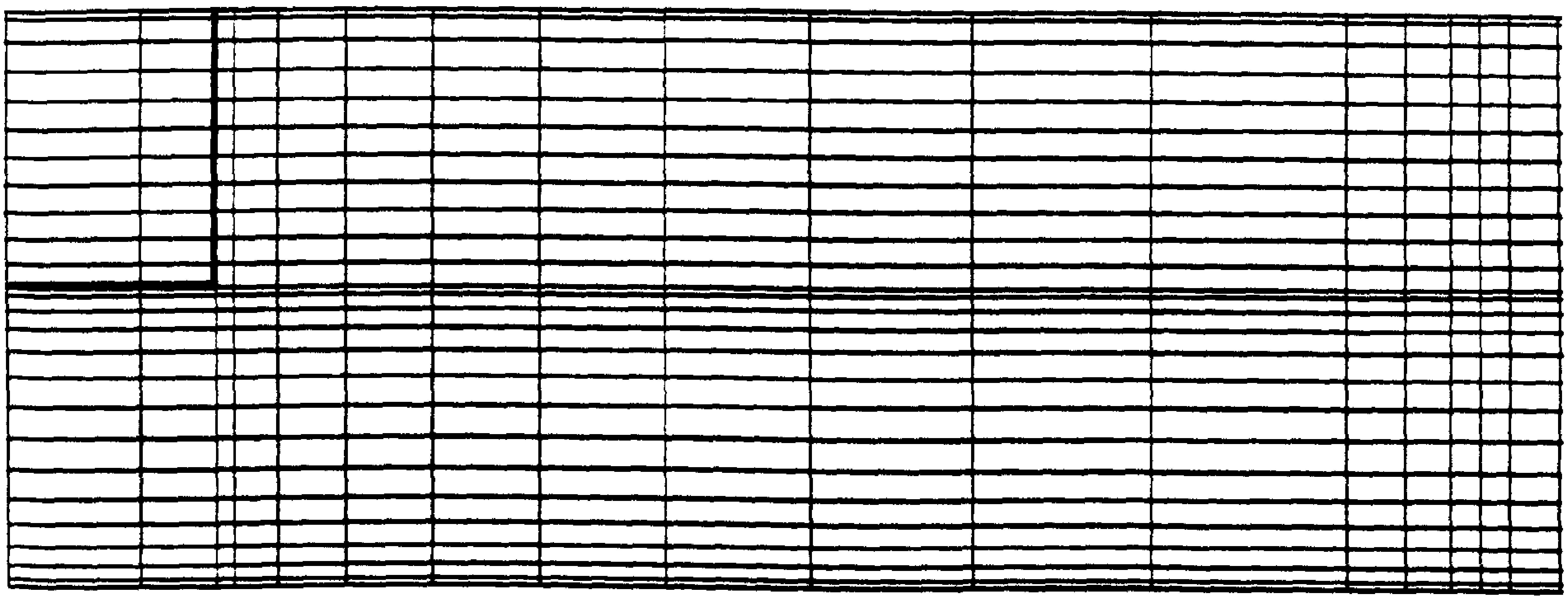


Figure 4.16 *Computational mesh in step region.*

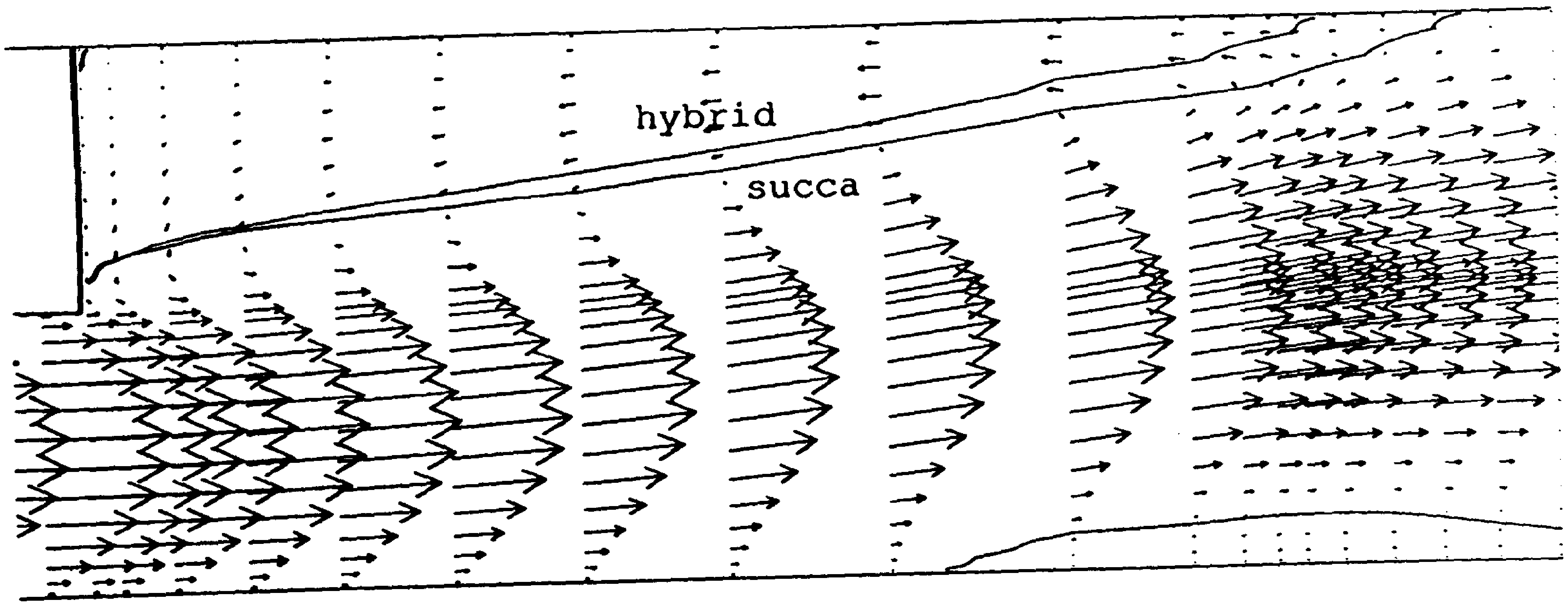


Figure 4.17 *A comparison of re-attachment lengths for Re = 200. Velocity vectors shown for SUCCA2D implementation.*

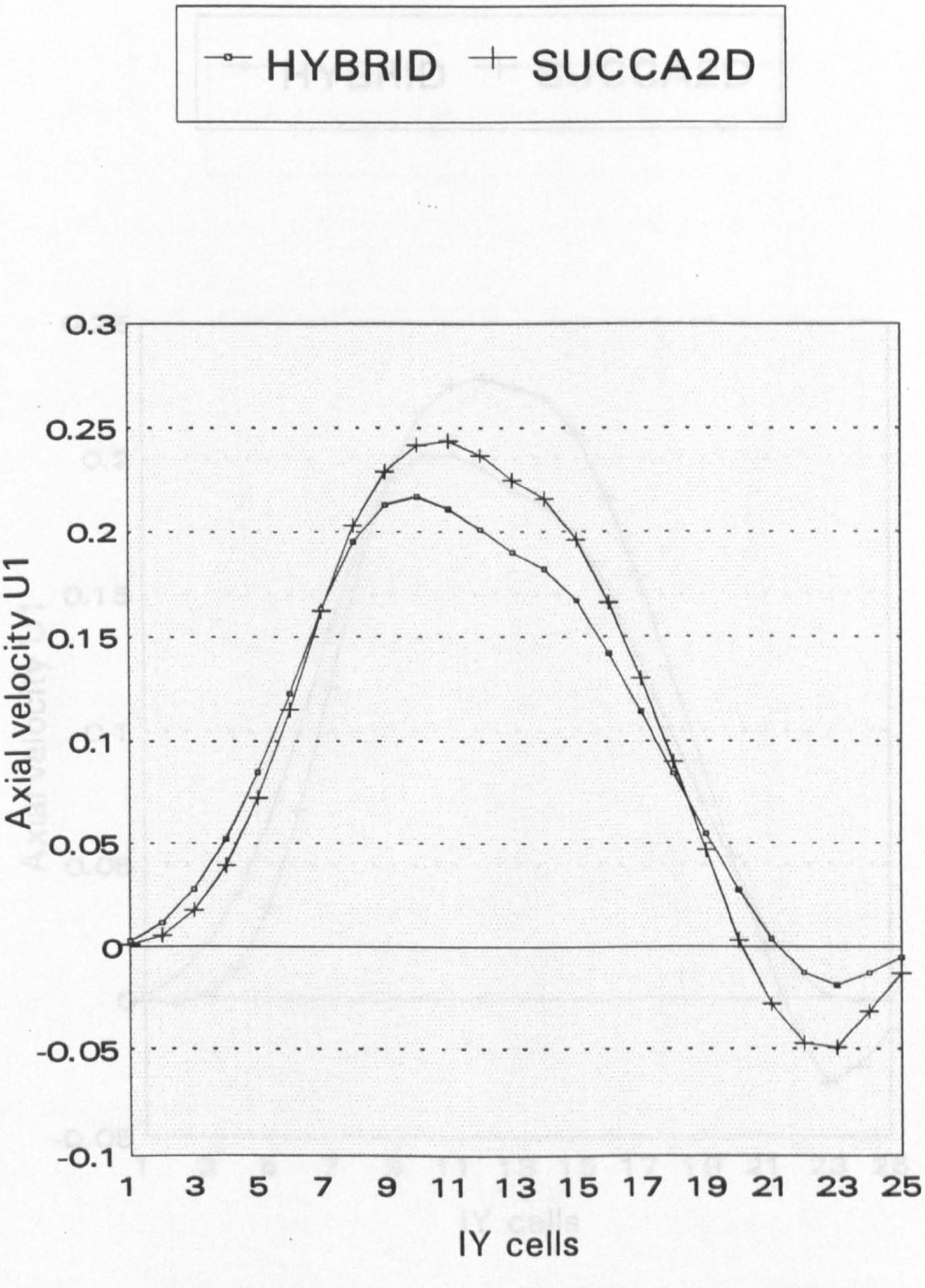


Figure 4.17a ***Axial velocity profiles for HYBRID and SUCCA2D at 3.33 step heights downstream of step.***

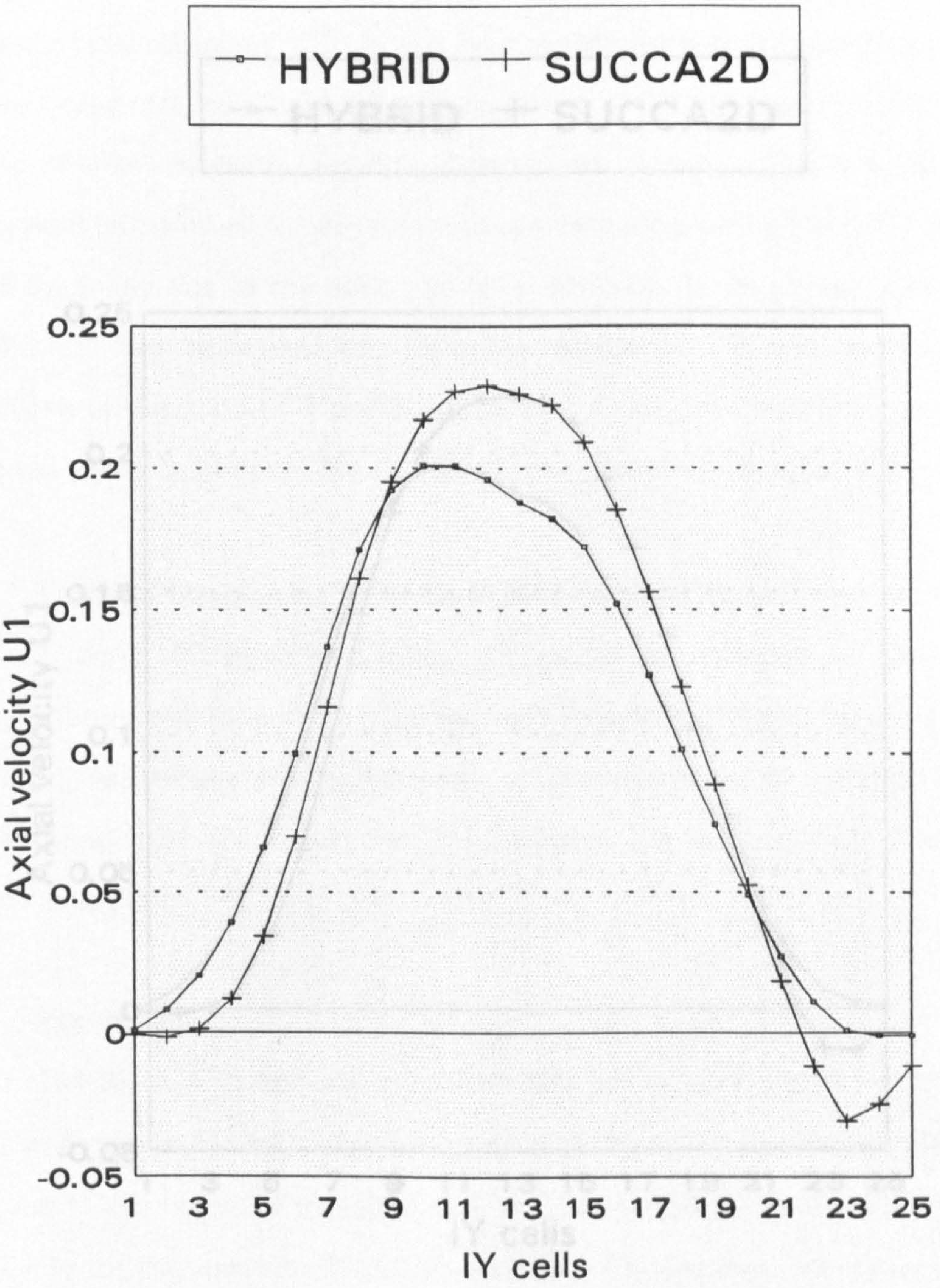


Figure 4.17b ***Axial velocity profiles for HYBRID and SUCCA2D at 4.27 step heights downstream of step.***

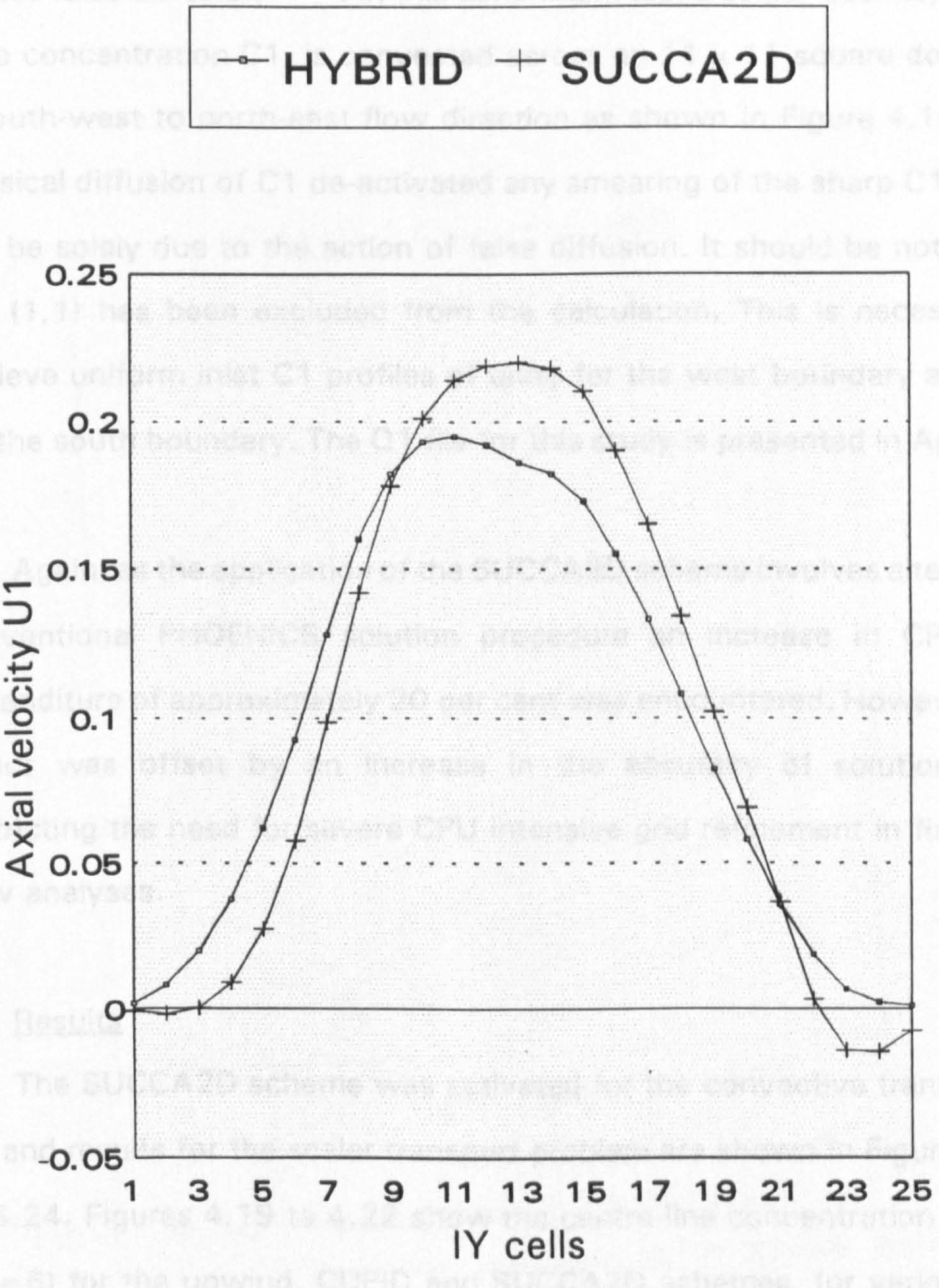


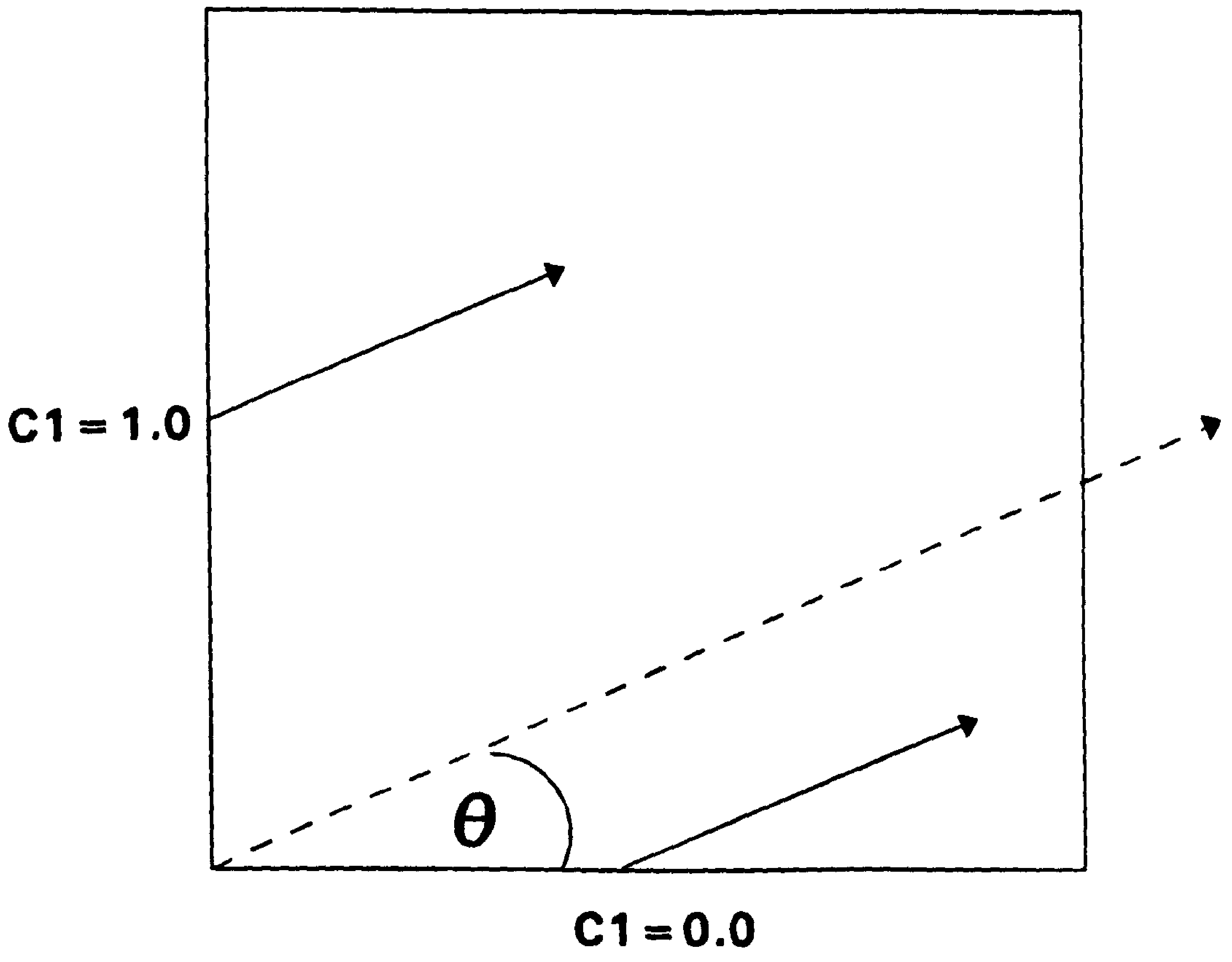
Figure 4.17c *Axial velocity profiles for HYBRID and SUCCA2D at 4.64 step heights downstream of step.*

a standard benchmark test to identify the ability of numerical schemes to reduce false diffusion(36)(50). In this benchmark test a scalar quantity, in this case concentration C1, is convected across an 11 x 11 square domain in a south-west to north-east flow direction as shown in Figure 4.18. With physical diffusion of C1 de-activated any smearing of the sharp C1 profile will be solely due to the action of false diffusion. It should be noted that cell (1,1) has been excluded from the calculation. This is necessary to achieve uniform inlet C1 profiles of unity for the west boundary and zero for the south boundary. The Q1 file for this study is presented in Appendix E.

Again, as the application of the SUCCA2D scheme involves altering the conventional PHOENICS solution procedure an increase in CPU time expenditure of approximately 20 per cent was encountered. However, this effect was offset by an increase in the accuracy of solution while restricting the need for severe CPU intensive grid refinement in fixed-grid flow analyses.

Results

The SUCCA2D scheme was activated for the convective transport of C1 and results for the scalar transport problem are shown in Figures 4.19 to 4.24. Figures 4.19 to 4.22 show the centre-line concentration profiles (IX = 6) for the upwind, CUPID and SUCCA2D schemes, for various flow angles. The colour plots of Figures 4.23 and 4.24 highlight the numerical diffusion of the conserved scalar for the upwind and SUCCA2D schemes, at a flow angle of 45 degrees. Although suppression of numerical diffusion is not complete for all flow angles, a significant reduction in the amount of



11 x 11 square grid

Figure 4.18 *Geometry for the analysis of pure convection of a scalar quantity.*

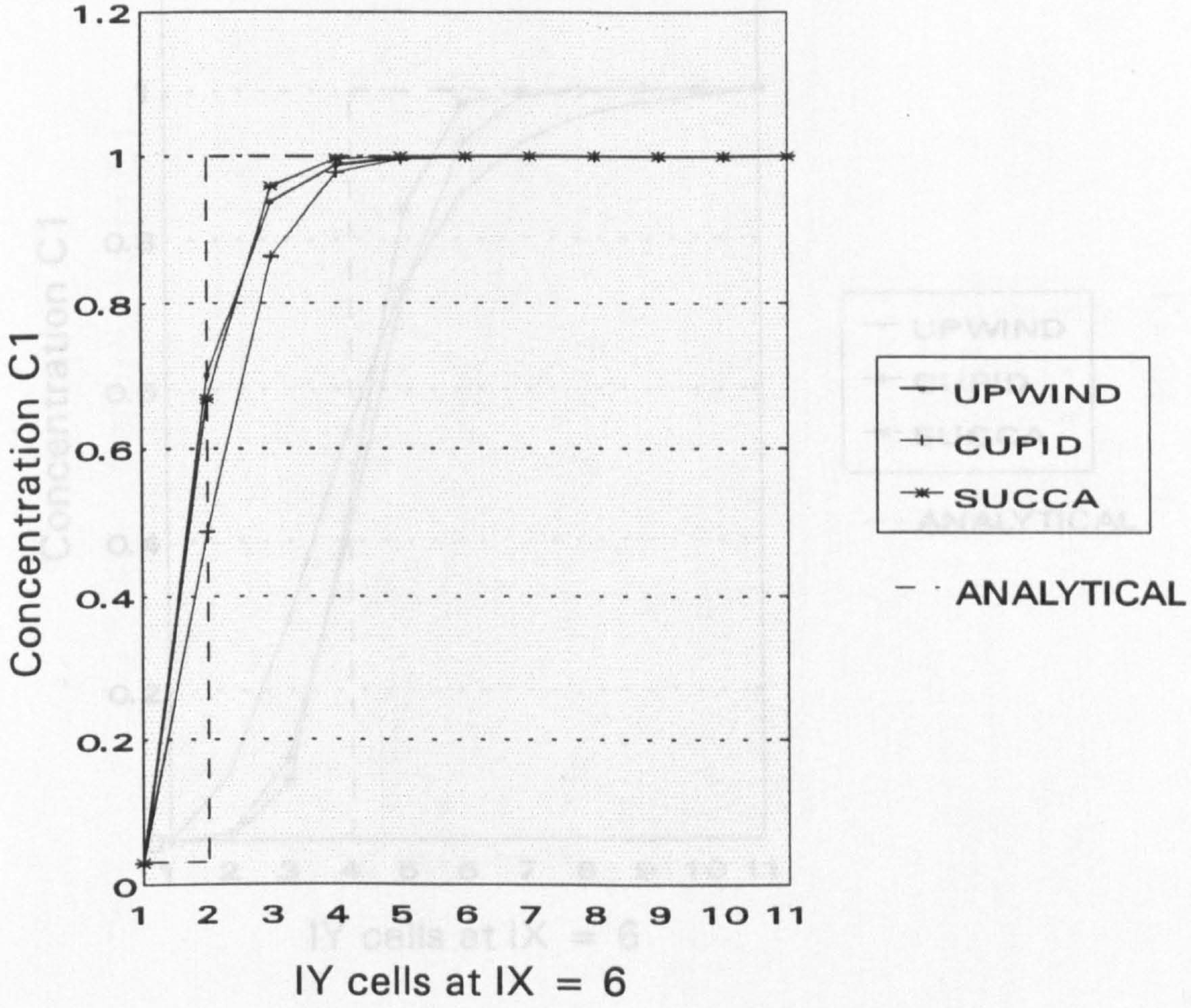


Figure 4.19 ***Profiles of scalar concentration for UPWIND, CUPID and SUCCA2D. Flow angle = 11.3.***

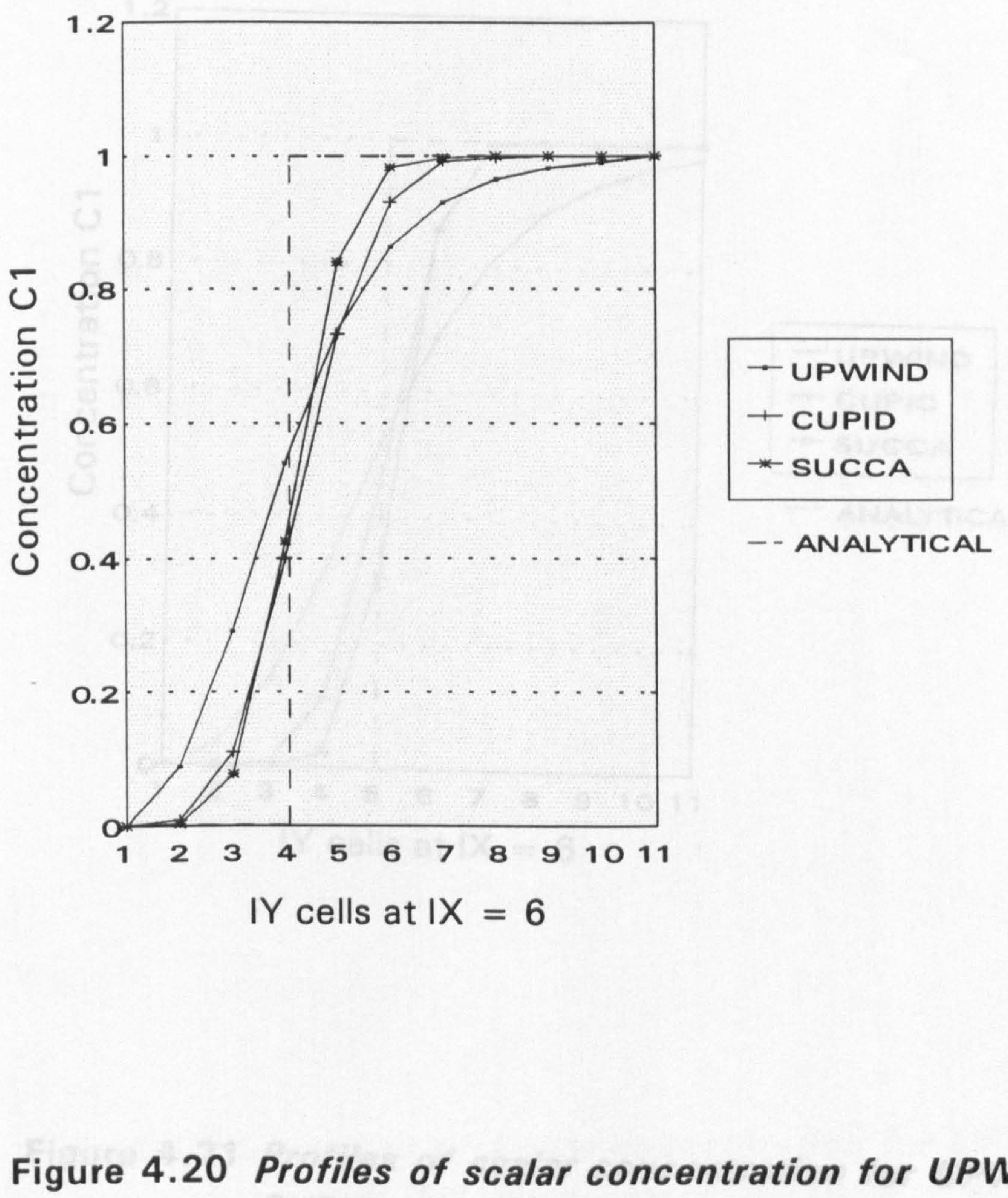


Figure 4.20 *Profiles of scalar concentration for UPWIND, CUPID and SUCCA2D. Flow angle = 31.0.*

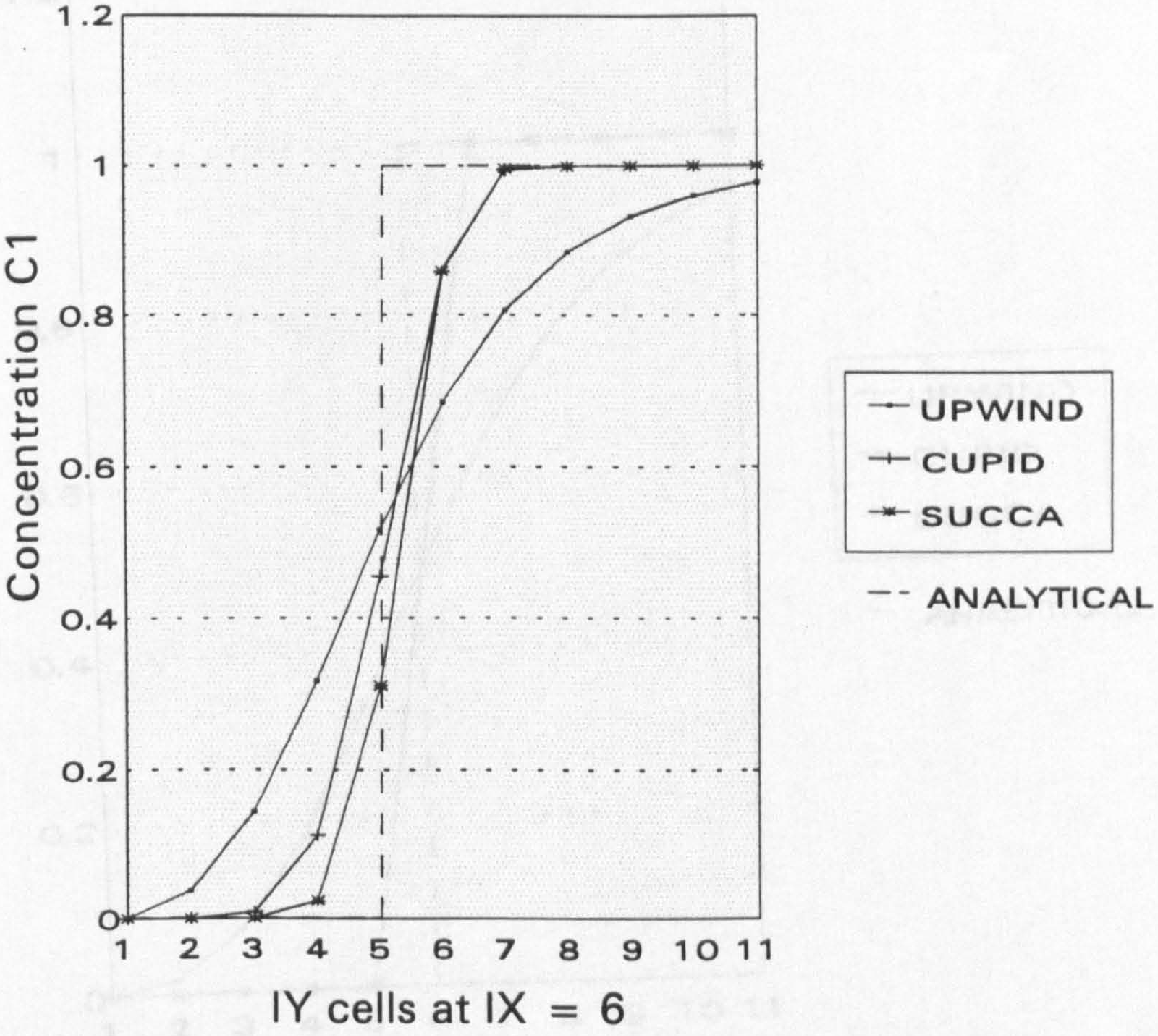


Figure 4.21 *Profiles of scalar concentration for UPWIND, CUPID and SUCCA2D. Flow angle = 38.0.*

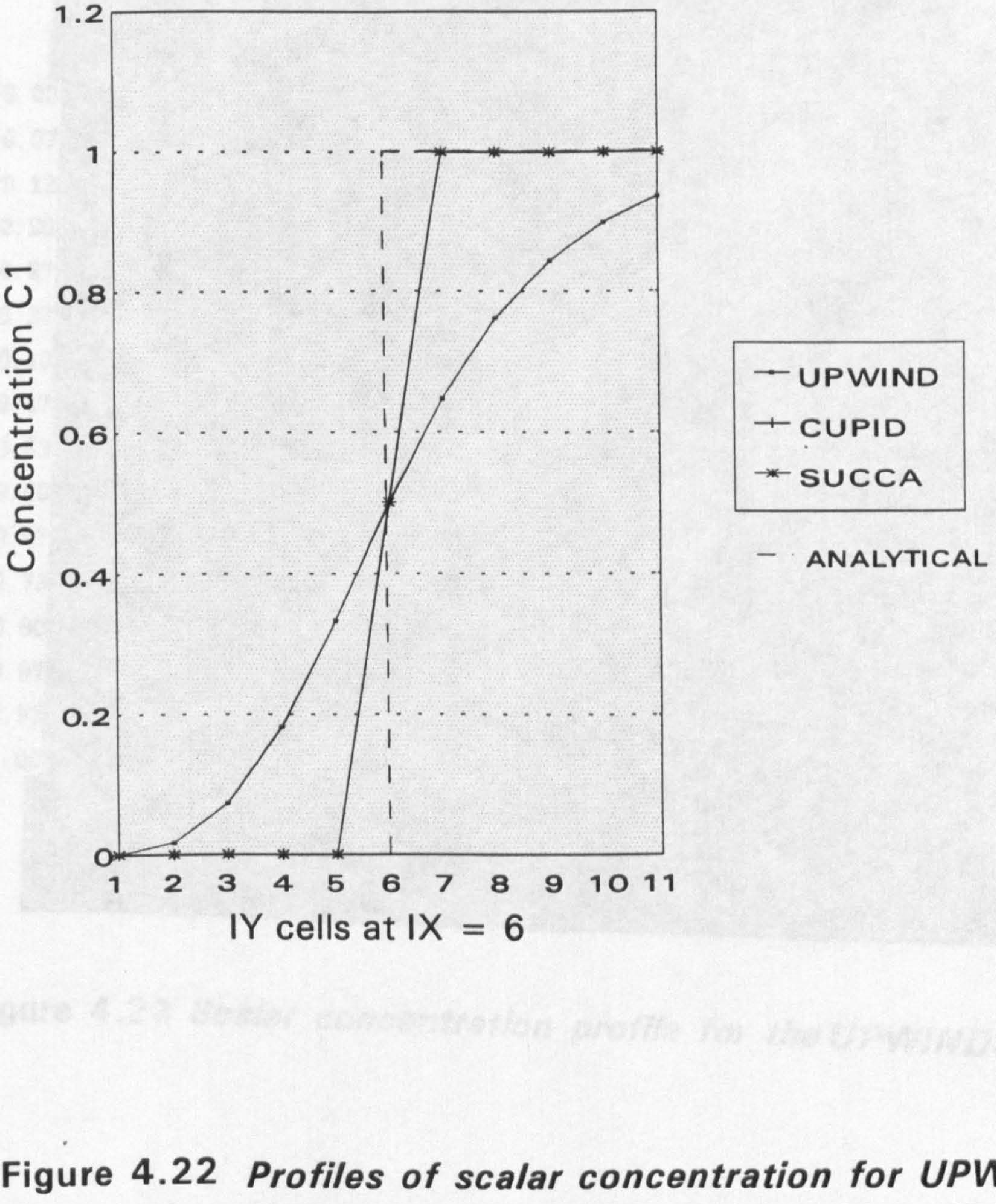


Figure 4.22 *Profiles of scalar concentration for UPWIND, CUPID and SUCCA2D. Flow angle = 45.0.*

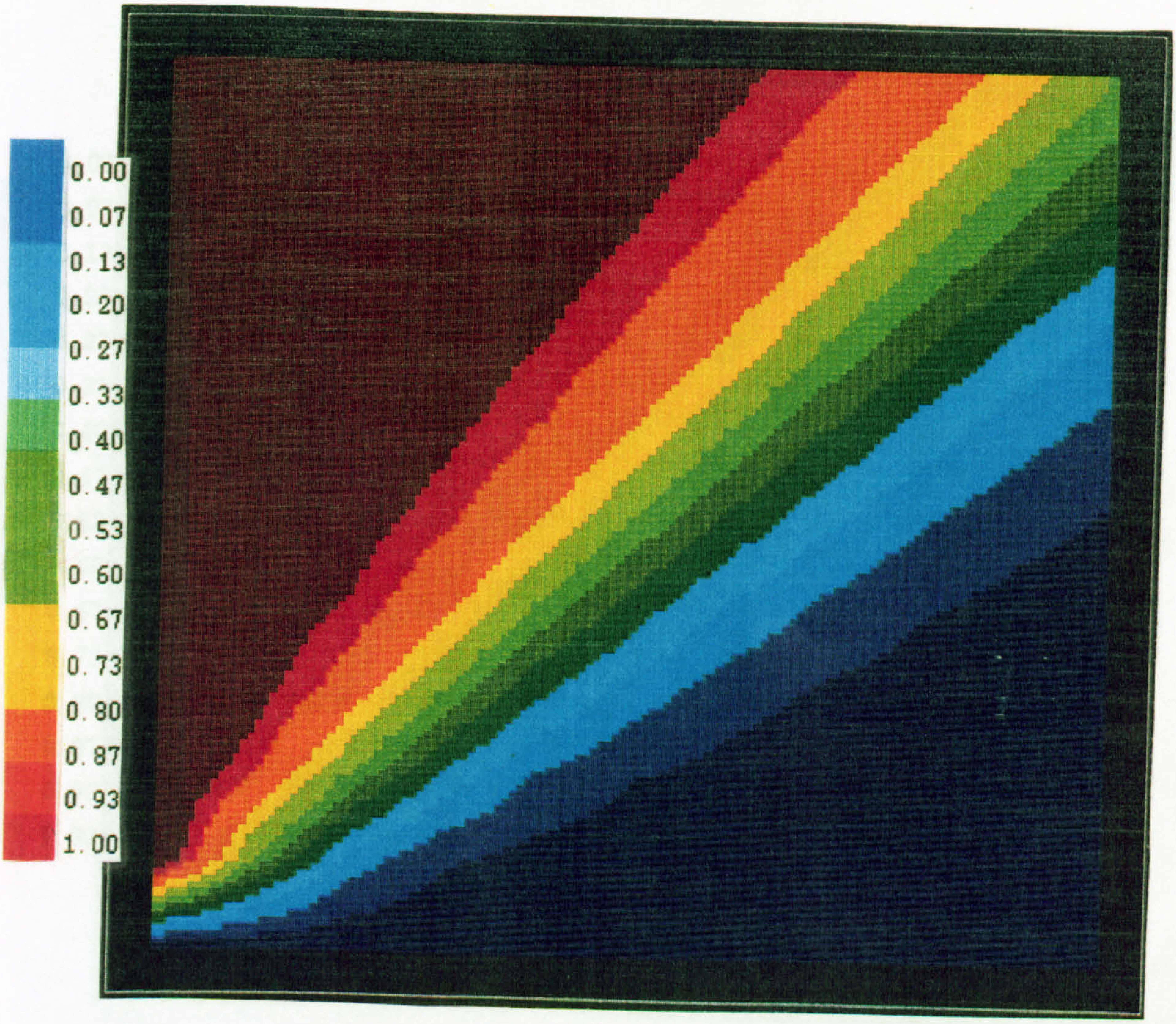


Figure 4.23 *Scalar concentration profile for the UPWIND scheme.*

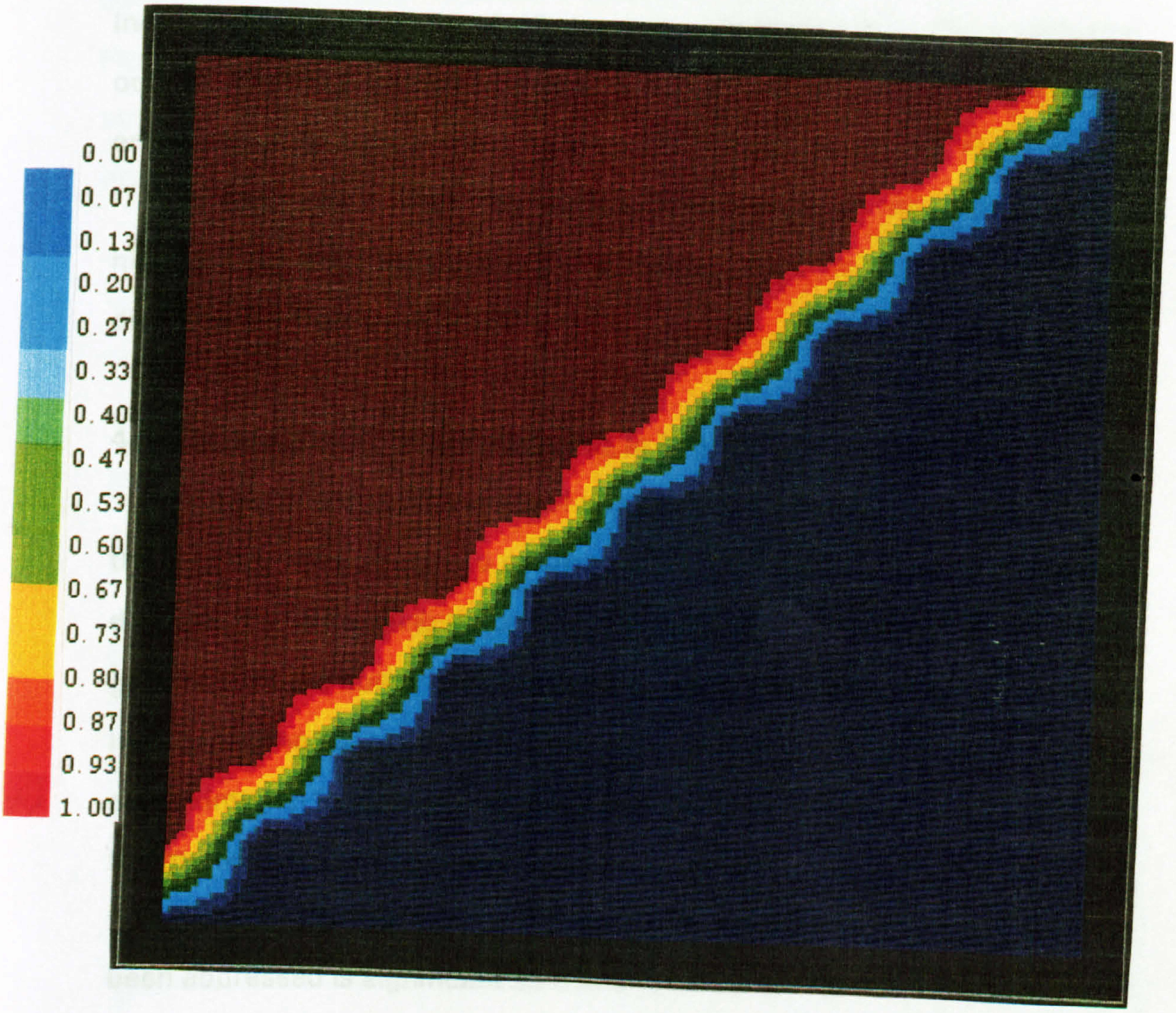


Figure 4.24 ***Scalar concentration profile for the SUCCA2D scheme.***

false diffusion is observed in comparison with the analytical solution. Indeed, for a flow angle of 45 degrees, where maximum false diffusion occurs, the discontinuity of scalar concentration is preserved across the entire computational domain for both CUPID and SUCCA2D schemes.

In comparison with the CUPID, the results at other flow angles highlight the enhanced numerical accuracy of the SUCCA2D scheme as discussed in Section 4.4.1.4.

4.5 The SUCCA2D Scheme Applied to Laminar Vortex Shedding

It is significant in the work of both Braza *et al*(27) and Davis *et al*(33) that the authors have sought to overcome the erroneous effects of numerical diffusion by employing discretisation schemes with orders of accuracy greater than the numerically dissipative first-order upwinding. As has been highlighted however, such higher order schemes will only give a better approximation to species gradient along the lines of grid cells. The fact that the multidimensional nature of numerical diffusion, caused by treating the flow across each control volume face as locally one-dimensional, has not been addressed is significant as this error is by far the most damaging part of numerical diffusion in two-dimensional flows(51).

Results

A transient analysis was carried out using the SUCCA2D scheme for the two-dimensional laminar vortex shedding problem using the same geometry and boundary conditions as outlined in Sections 4.1 and 4.2. The Reynolds number was 500. The SUCCA2D scheme was applied to the transport of the U and V momentum components, replacing the normal default hybrid scheme. Again, the steady asymmetric flow field shown in Figure 4.4 was used as the initial conditions for a transient run. With the scheme now in place the development of continuous vortex shedding ensued.

Time resolution trials were carried out to find an appropriate value of time step. The vortex shedding calculation was progressively extended in time until it could be concluded that the influence of numerical diffusion on vortex decay was minimal i.e. that a continuous vortex shedding signal had been predicted. A series of calculations, using different time steps, were executed and the results of this study are shown in Table 4.1.

Non-dimensional time step	CPU time per unit non-dimensional time increment (secs)	Multiplication factor in comparison with chosen time step (0.08)
0.08	17.4	1.0
0.04	230.8	13.0
0.01	710.9	41.0

Table 4.1 Time resolution trials for appropriate time step value.

It can be seen that as the time resolution is improved, the CPU time per unit non-dimensional time increment is increased significantly. It was also found that as the time resolution was improved, the mean amplitude of the velocity transient was maintained at a higher level. However, the difference in the mean amplitude of velocity was of the order of 20 per cent for the range of time steps studied and was not considered to be of such a magnitude as to merit the choice of the smaller time steps 0.04 or 0.01.

This conclusion was reached, particularly, as the vortex shedding frequency was insignificantly affected by changes in time resolution, with a frequency phase shift of the order of 6 per cent encountered for the range of time steps chosen.

With the above observations in mind a non-dimensional time step of 0.08 was chosen which produced results over a total time period of 56.0 time units i.e. 700 time steps at 0.08 units per step = 56.0. The CPU time for the calculation was 9 hours on a Sun sparc1 + computer.

Time resolution trials were executed for a second purpose. As previously highlighted, Anderson *et al*(45) had shown that as the time resolution of the periodic flow is improved, the results of the calculation tend to a symmetric pattern. However, the results for the SUCCA2D analysis show that when the time resolution is improved then, as opposed to stabilising the flow, the mean velocity amplitude of the vortex shedding transient was in fact increased. This may be explained by the fact that an explicit perturbation (by circulation) had been provided at the onset of the calculation, i.e. an attempt was made to model the natural perturbations which promote instability in the laminar Navier-Stokes flow. This method is preferred to the promotion of instability via numerical error with its apparent faults.

The development of continuous vortex shedding can be observed in Figure 4.25 as a sinusoidal single cell axial velocity transient in the region of vortex shedding. The frequency of the shed vortices corresponds to a

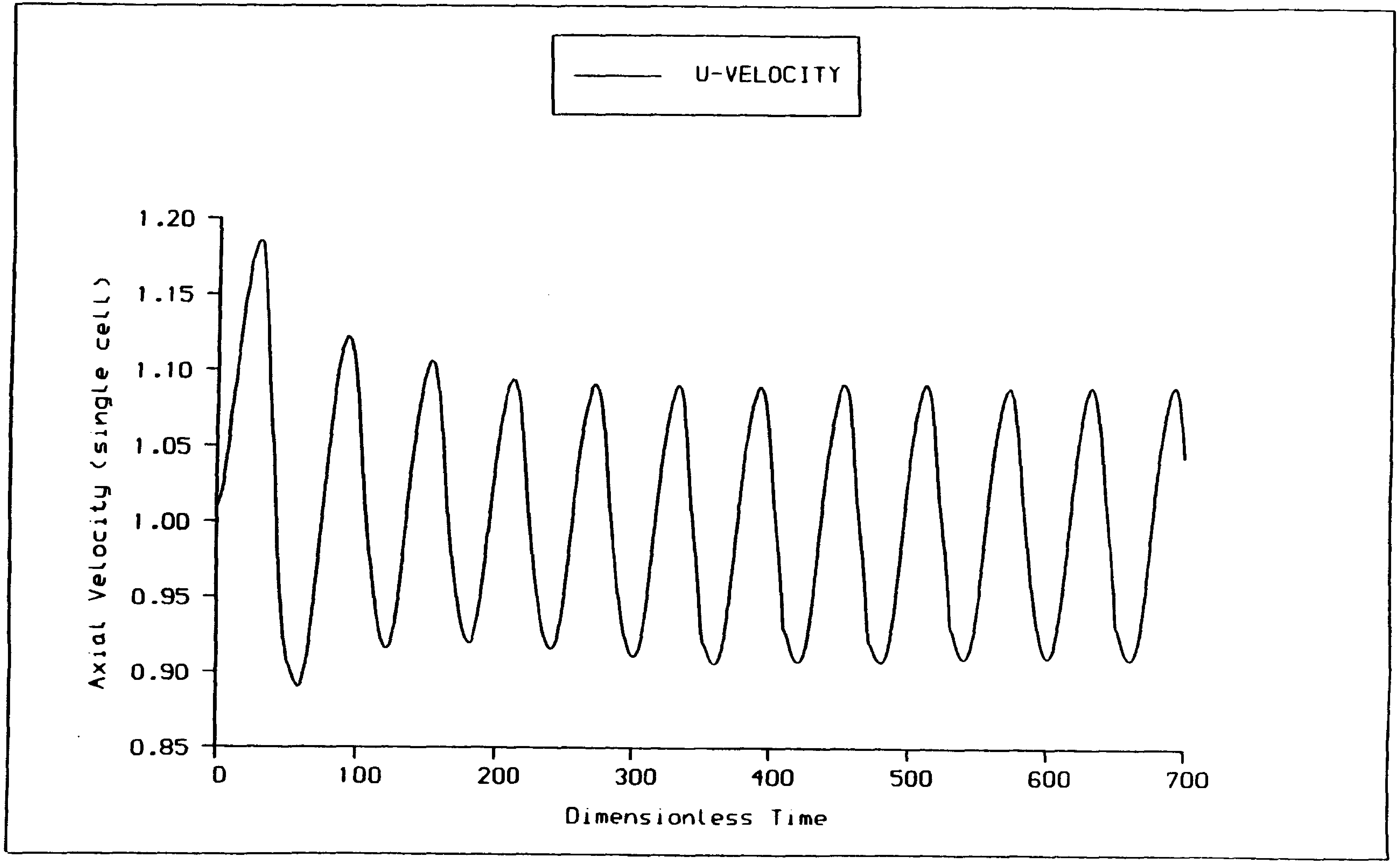


Figure 4.25 *Axial velocity transient using SUCCA2D for a single cell located in the region of vortex shedding.*

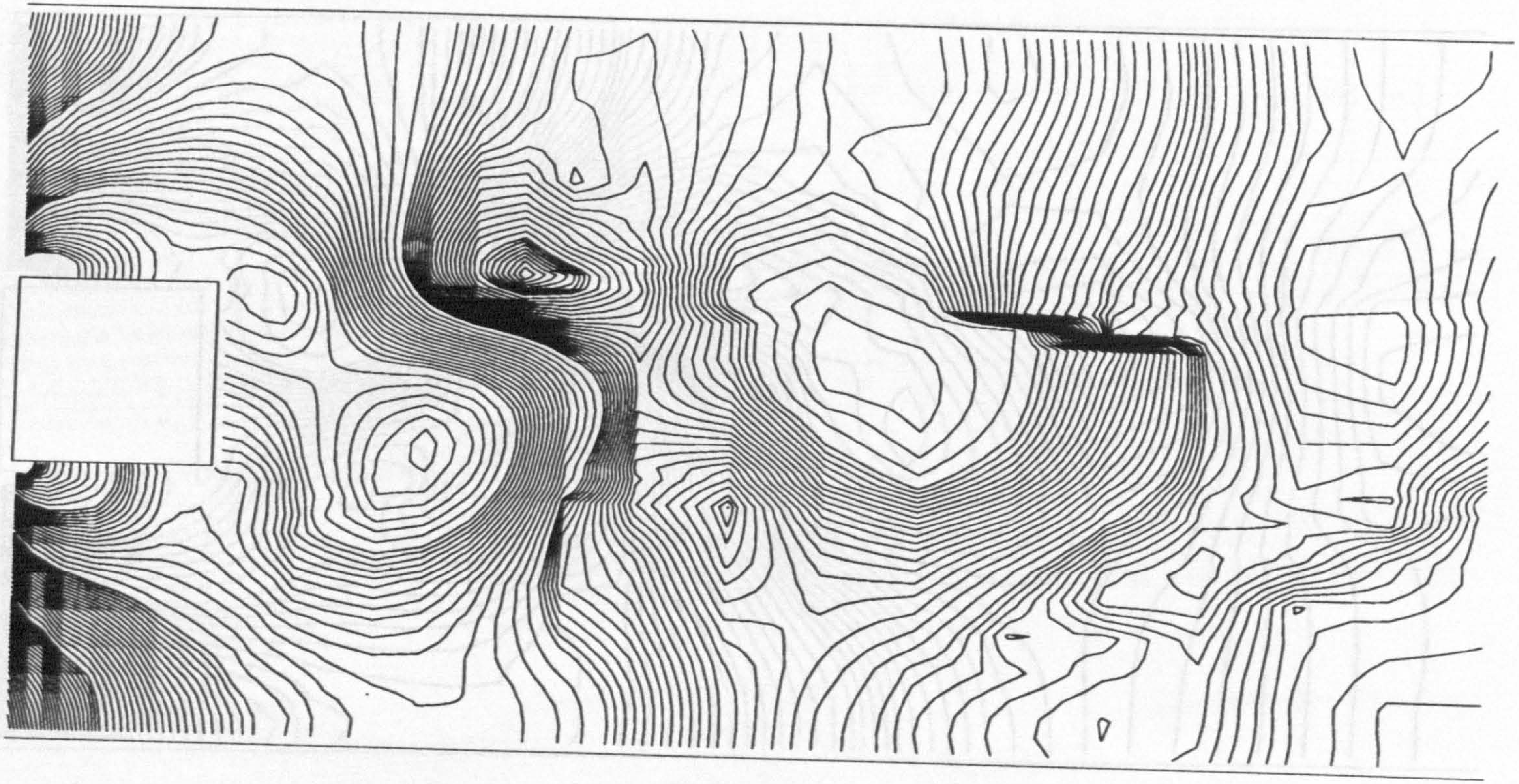


Figure 4.26 *Pressure contours showing vortex shedding development at a non-dimensional time $T = 300$.*

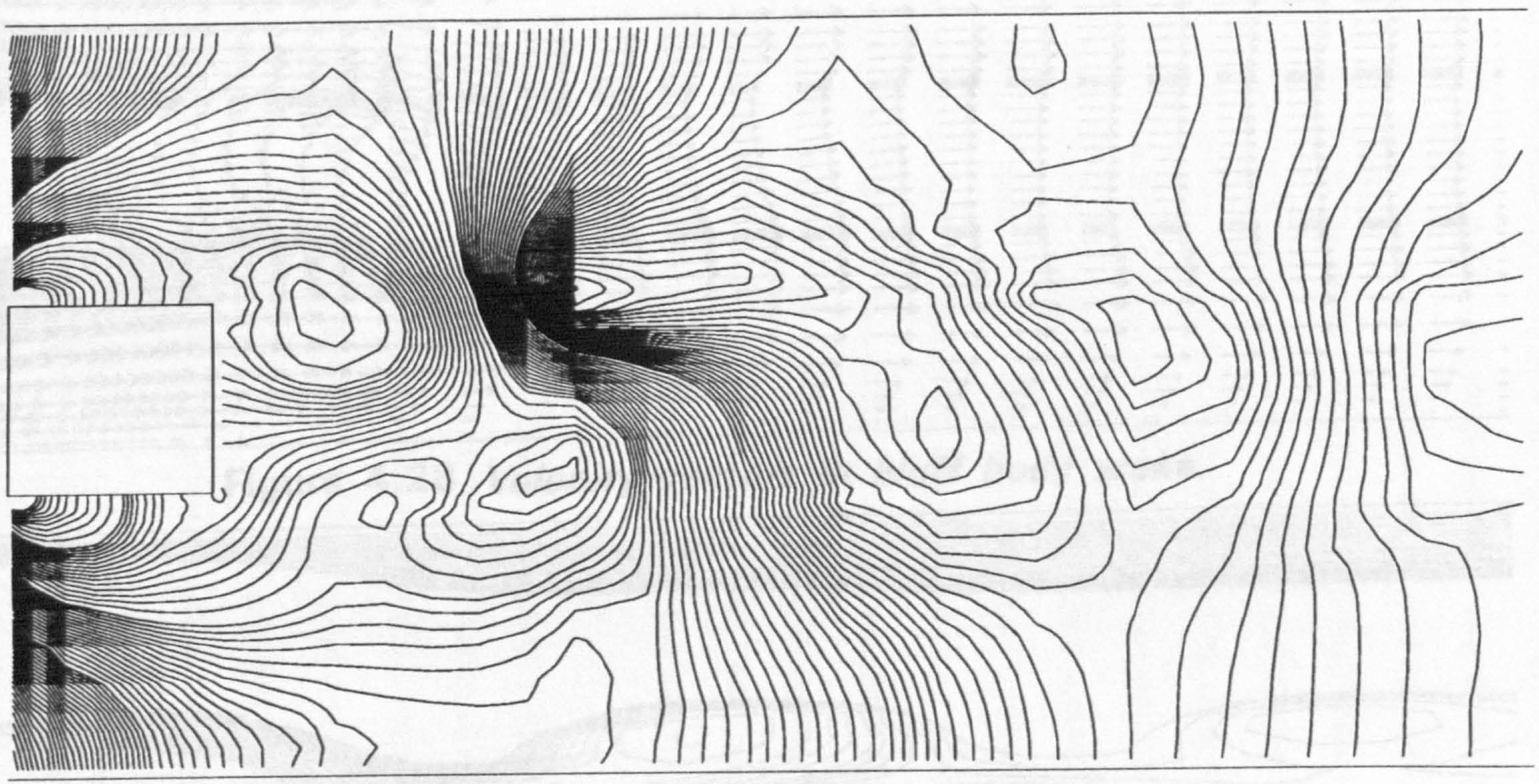


Figure 4.27 *Pressure contours showing vortex shedding development at a non-dimensional time T = 330.*

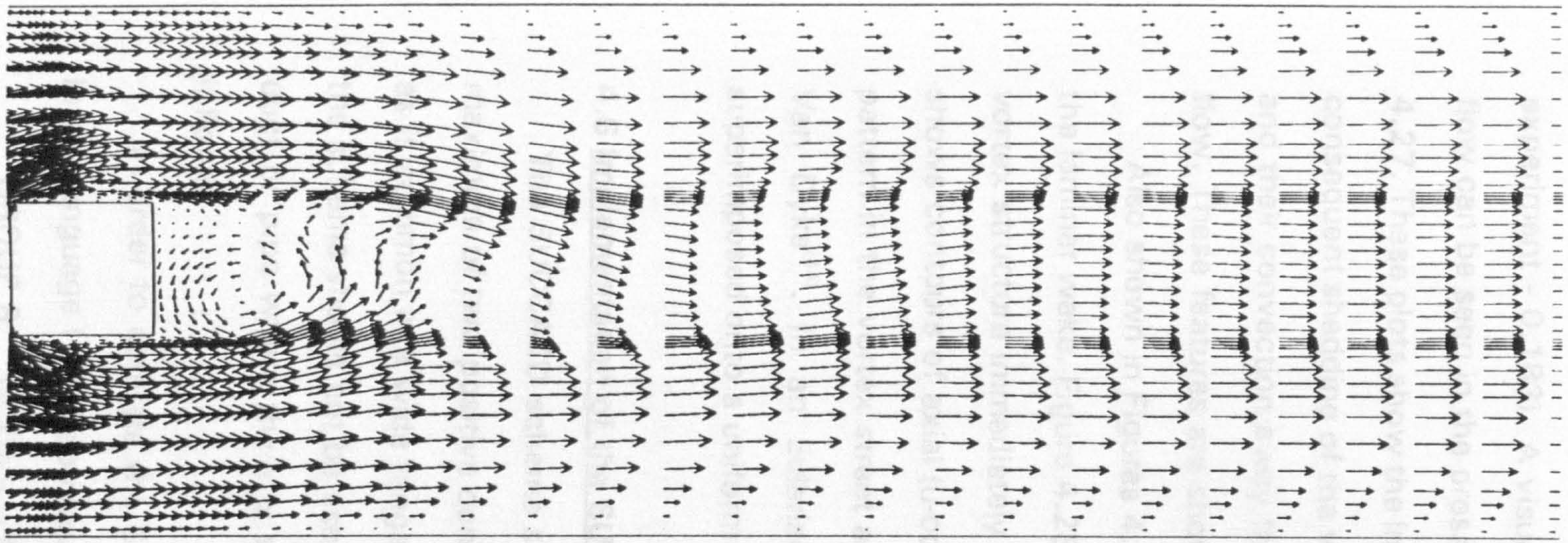


Figure 4.28 *Velocity vectors in bluff body wake.*

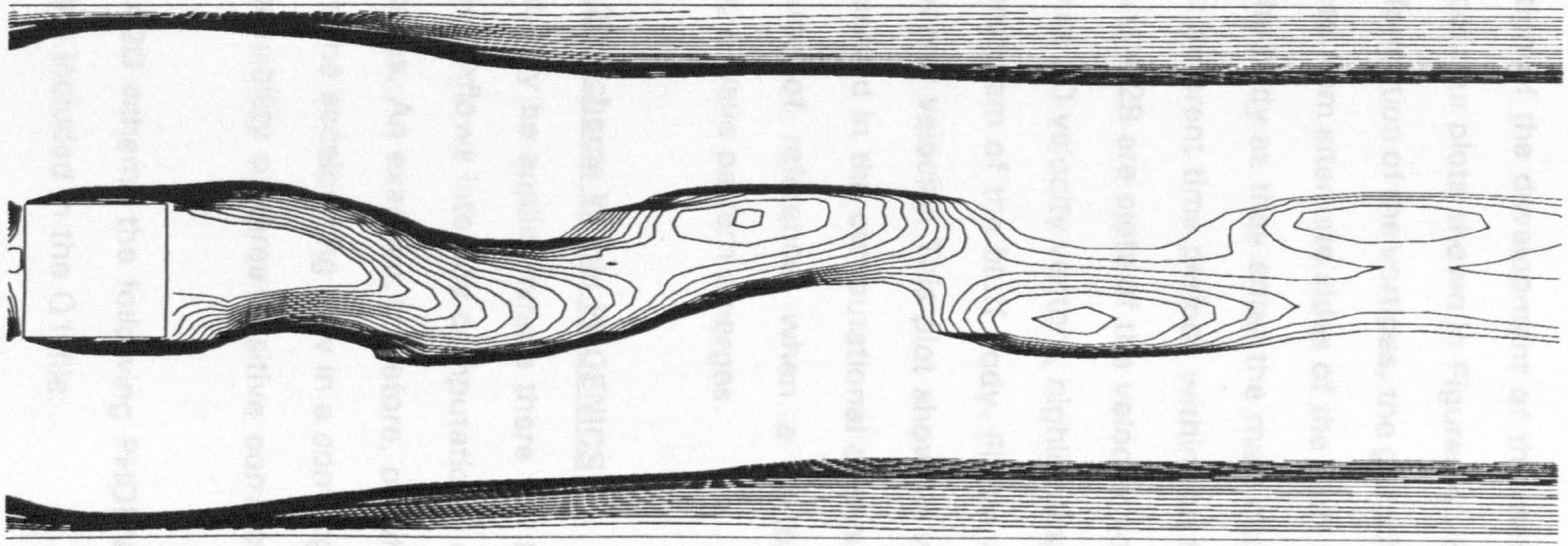


Figure 4.29 *Contours of axial velocity showing wave pattern in wake.*

Strouhal number of 0.196 which is in good agreement with the numerical and experimental analysis of Davis *et al*(33) (computation - 0.198, experiment - 0.183). A visualisation of the development of this periodic flow can be seen in the pressure contour plots shown in Figures 4.26 and 4.27. These plots show the initial formation of the vortices, the growth and consequent shedding of the vortices from alternate sides of the bluff body and their convection away from the body as they enter the main channel flow. These features are shown at different time periods within the flow.

Also shown in Figures 4.28 and 4.29 are plots of the velocity field in the laminar wake. Figure 4.28 shows 2D velocity vectors, highlighting the vortex structure immediately downstream of the bluff body. Figure 4.29 shows contours of axial (u-component) velocity. This plot shows a wave pattern in the vortex street as observed in the computational analysis of Van Dyke(44). In an Eulerian frame of reference, when a vortex is superimposed onto a uniform flow, a wave pattern emerges.

4.6 Implementation of the SUCCA2D Scheme Within PHOENICS

The SUCCA2D scheme should only be applied where there will be a maximum of two positive convective inflows into any computational cell, as is common in a wide range of cases. An example, therefore, of where the scheme should not be used is in the accelerating flow in a convergent duct or pipe where there is the possibility of three positive convective inflows.

In order to activate the SUCCA2D scheme the following PHOENICS Input Language (PIL) commands are included in the Q1 file:

```
GROUP 8.   UCONV=T
```

which implies that the user wishes to access and, in this case, modify the convection coefficients in Group 8, Section 8 of GROUND.

GROUP 13. PATCH(SUCCAU,CELL,1,NX,1,NY,1,1,1,LSTEP)

which is the normal method of implementing the scheme i.e. over the entire computational domain. The PATCH name in this case must be 'SUCCAU' as shown. It should be noted, however, that in the scalar transport example in Section 4.4.1.6 the SUCCA2D scheme was applied to the convective transport of C1 over the internal and outflow boundary cells i.e. (2,NX,2,NY,1,1,1,LSTEP). This was necessary due to the fact that at the inflow boundary cells there were no upwind corner cells to be included within the scheme. Such a condition will only occur where the inlet boundary flow is skewed (as in this case). Where the inlet boundary flow is non-skewed, as in most conventional analyses, the limits of the SUCCA2D application should be the entire computational domain.

To implement the scheme for the transport of momentum:

```
COVAL(SUCCAU,U1,GRND,GRND)
COVAL(SUCCAU,V1,GRND,GRND)
```

To implement the scheme for the scalar transport e.g.

```
COVAL(SUCCAU,KE,GRND,GRND)
COVAL(SUCCAU,EP,GRND,GRND)
COVAL(SUCCAU,C1,GRND,GRND)
```

Finally, in GROUP 19 the following LOGICAL statements are communicated by SATELLITE to GROUND:

To solve for U1 via SUCCA2D:

```
LG(1)=T
```

To solve for V1 via SUCCA2D:

LG(2)=T

To solve for scalars via SUCCA2D:

LG(3)=T

The above LOGICAL statements must be de-activated (LG(n)=F) if the scheme is not being used for that particular transported variable.

It should be noted that the scheme can be applied to any or all of these dependent variables. Also, the COefficient and VALue arguments have been formulated within the GRND sections of GROUND, however there is no reason why the coding cannot be placed within any other GRND section.

CHAPTER 5 TURBULENT VORTEX SHEDDING

5.1 Introduction

The existence of large, coherent structures in the turbulent wake behind a bluff body, which are periodic in nature, has been confirmed in the experimental analyses of Papailiou and Lykoudis(23) and Cantwell(24). In this chapter, the predictive capability of the PHOENICS code in capturing these turbulent vortices is described. The Q1 file for this study is presented in Appendix F.

5.2 Geometrical Configuration

The geometry considered for this analysis was based upon the experimental analysis of Mottram and Rawat(52) and is shown in Figure 5.1. In the computational calculation, a two-dimensional plane channel geometry is considered. The rectangular bluff body is located at the centre of the channel whose containing walls lie a distance D = 80mm apart. A blockage ratio of 0.27 is considered while the bluff body itself has an aspect ratio of 0.667.

A non-uniform computational mesh of 67X by 50Y was employed and the mesh outlay in the region of vortex shedding is shown in Figure 5.2.

5.3 Numerical Model

5.3.1 Governing Equations

The unsteady incompressible turbulent fluid flow was modelled by partial differential equations describing the conservation of mass, the conservation of momentum and a two-equation turbulence model in two

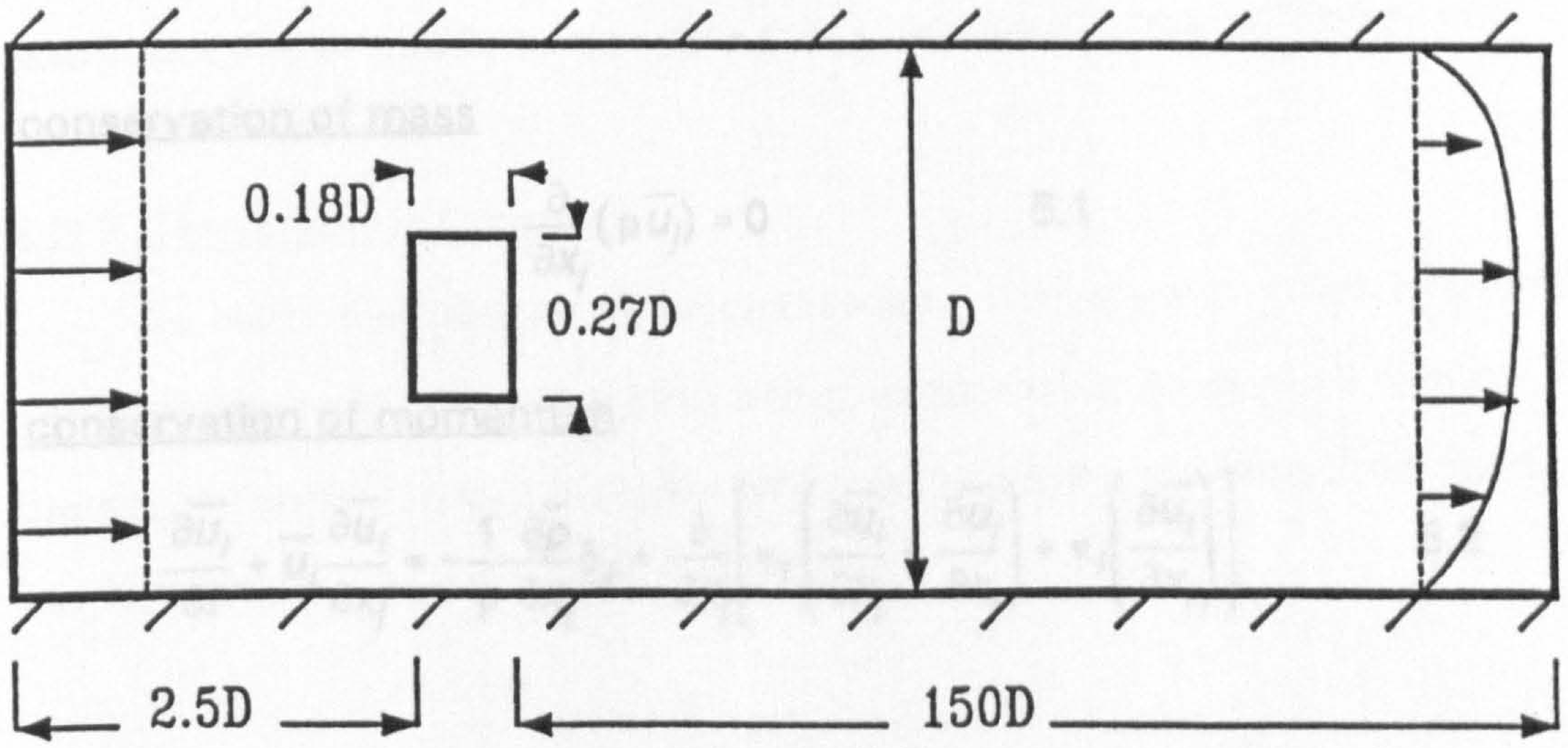


Figure 5.1 *Turbulent vortex shedding geometry.*

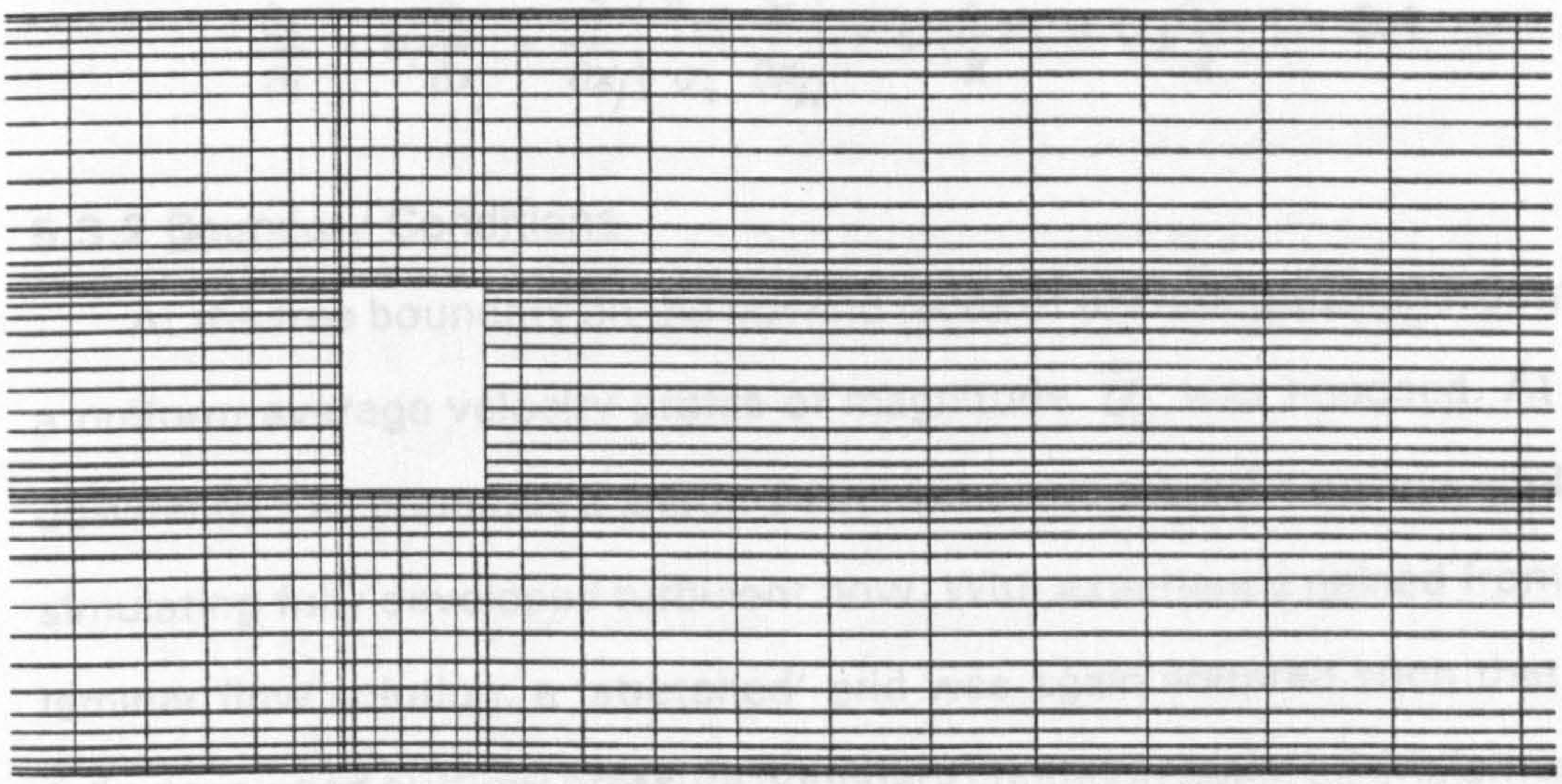


Figure 5.2 *Computational mesh in region of vortex shedding.*

rectangular Cartesian coordinate directions:

conservation of mass

$$\frac{\partial}{\partial x_j}(\rho \bar{u}_j) = 0 \qquad 5.1$$

conservation of momentum

$$\frac{\partial \bar{u}_i}{\partial t} + \bar{u}_j \frac{\partial \bar{u}_i}{\partial x_j} = -\frac{1}{\rho}\frac{\partial \bar{p}}{\partial x_i}\delta_{ij} + \frac{\partial}{\partial x_j}\left[\nu_t\left(\frac{\partial \bar{u}_i}{\partial x_j} + \frac{\partial \bar{u}_j}{\partial x_i}\right) + \nu_l\left(\frac{\partial \bar{u}_i}{\partial x_j}\right)\right] \qquad 5.2$$

conservation of turbulent kinetic energy *k*

$$\frac{\partial k}{\partial t} + \bar{u}_j \frac{\partial k}{\partial x_j} = \frac{\partial}{\partial x_j}\left(\frac{\nu_{eff}}{\sigma_k}\frac{\partial k}{\partial x_j}\right) + P_k - \epsilon \qquad 5.3$$

conservation of turbulent kinetic energy dissipation rate ϵ

$$\frac{\partial \epsilon}{\partial t} + \bar{u}_j \frac{\partial \epsilon}{\partial x_j} = \frac{\partial}{\partial x_j}\left(\frac{\nu_{eff}}{\sigma_\epsilon}\frac{\partial \epsilon}{\partial x_j}\right) + C_1\frac{\epsilon}{k}P_k - C_2\frac{\epsilon^2}{k} \qquad 5.4$$

5.3.2 Boundary Conditions

At the free boundary on the upwind faces of the computational domain a uniform average velocity profile of magnitude $\bar{U}_0$ was imposed. At the downwind free boundary a uniform mean pressure prescription was applied, simulating fully developed turbulent flow. With experience gained from the laminar flow solution, a 'stretched' grid was again adopted such that the fully developed outflow pressure boundary condition could be implemented with a reasonable degree of confidence. At solid surfaces the PHOENICS default log-law universal velocity profile was used to predict both momentum and turbulence quantities.

The Reynolds number of the flow is defined as $Re = \bar{U}_0 D / \nu_l$ and the Strouhal number is defined as $Str = fB/\bar{U}_0$ where B is the bluff body width.

5.3.3 Convergence Criteria

As with the laminar flow calculation, convergence was assumed within each particular time step when progressive single cell values of dependent variables in the region of vortex shedding showed little change per iteration as the calculation progressed.

5.4 Turbulent Vortex Shedding Analysis

As an explicit perturbation is required to trigger the vortex shedding flow the circulation method, employed in the laminar flow solution, was once again adopted. The no-slip condition was again removed for the momentum equations on the single cell layer surrounding the bluff body and a slip velocity applied. This slip velocity was approximately equal in magnitude to the incoming mean bulk velocity. A steady flow was then imposed through the channel and the resulting steady asymmetric flow field used as the initial conditions for a transient solution of flow past the bluff body without added circulation. This steady asymmetric flow pattern is shown in Figure 5.3.

A transient calculation was performed for a Reynolds number of 130000 in an attempt to obtain a similar Strouhal number found in the experimental analysis of Mottram and Rawat(52). It was found during the subsequent computational analysis that the prediction of the periodic nature of the flow was highly sensitive to the choice of time step. As the magnitude of the time step was gradually decreased (increased time

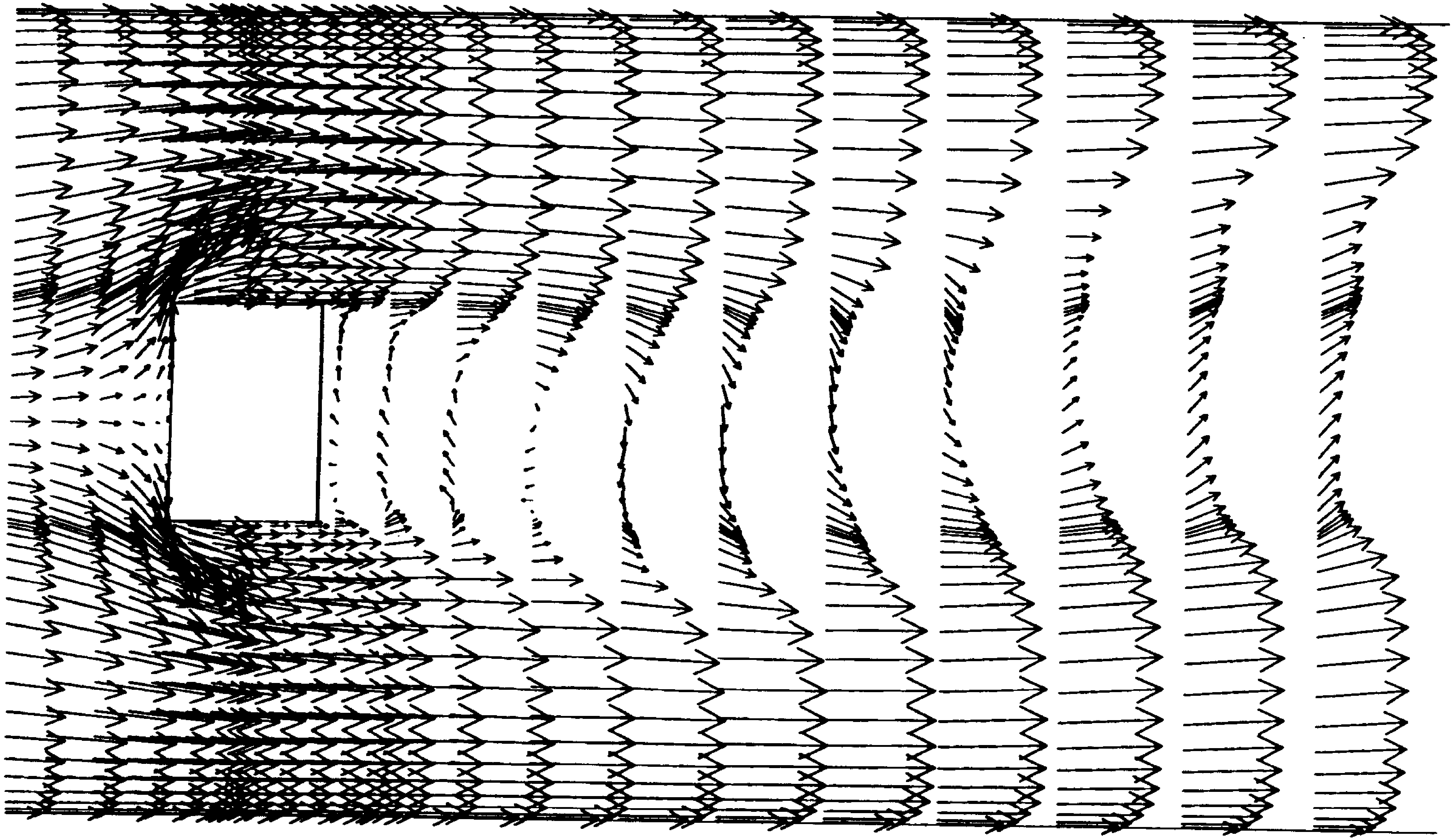


Figure 5.3 *Steady asymmetric flow field used as initial conditions.*

resolution) the adoption of turbulent vortex shedding became increasingly apparent. A time step of 4.0E-05 seconds was adopted which provided conclusive results over a working CPU timescale of approximately 2 days for a transient computation on a Sun sparc1 + computer.

Primarily, the computation was executed with the PHOENICS default hybrid scheme employed for the transport of all flow dependent variables. This resulted in the familiar decay of periodicity, encountered in the previous laminar vortex shedding analysis. The resulting computation yielded a steady, stable twin symmetrical vortex pattern downstream of the bluff body, as shown in Figure 5.4. The decay of flow periodicity in the wake is represented by a single cell axial mean velocity transient in the region of vortex shedding, as shown in Figure 5.5. The plotted transient is observed to decay very rapidly over the first three apparent shedding cycles, attaining the stable wake flow structure shown in Figure 5.4. Figures 5.6 and 5.7 show contours of the turbulence quantities k and ϵ for comparison with the SUCCA2D application. The plots show regions of high turbulence kinetic energy and its dissipation rate in areas of maximum mean shear, i.e. at the upstream corners of the bluff body. Both symmetrical plots highlight the loss of periodic asymmetry in the flow wake due to the excessive false diffusion of momentum.

The SUCCA2D scheme was implemented for both the turbulence parameters k and ϵ , and the two momentum components $\bar{U}$ and $\bar{V}$. With the scheme now in operation the development of continuous turbulent vortex shedding ensued. This flow phenomenon can be observed in Figure 5.8, where the average axial velocity is plotted against time for a single computational cell located at the edge of the turbulent wake. It was found

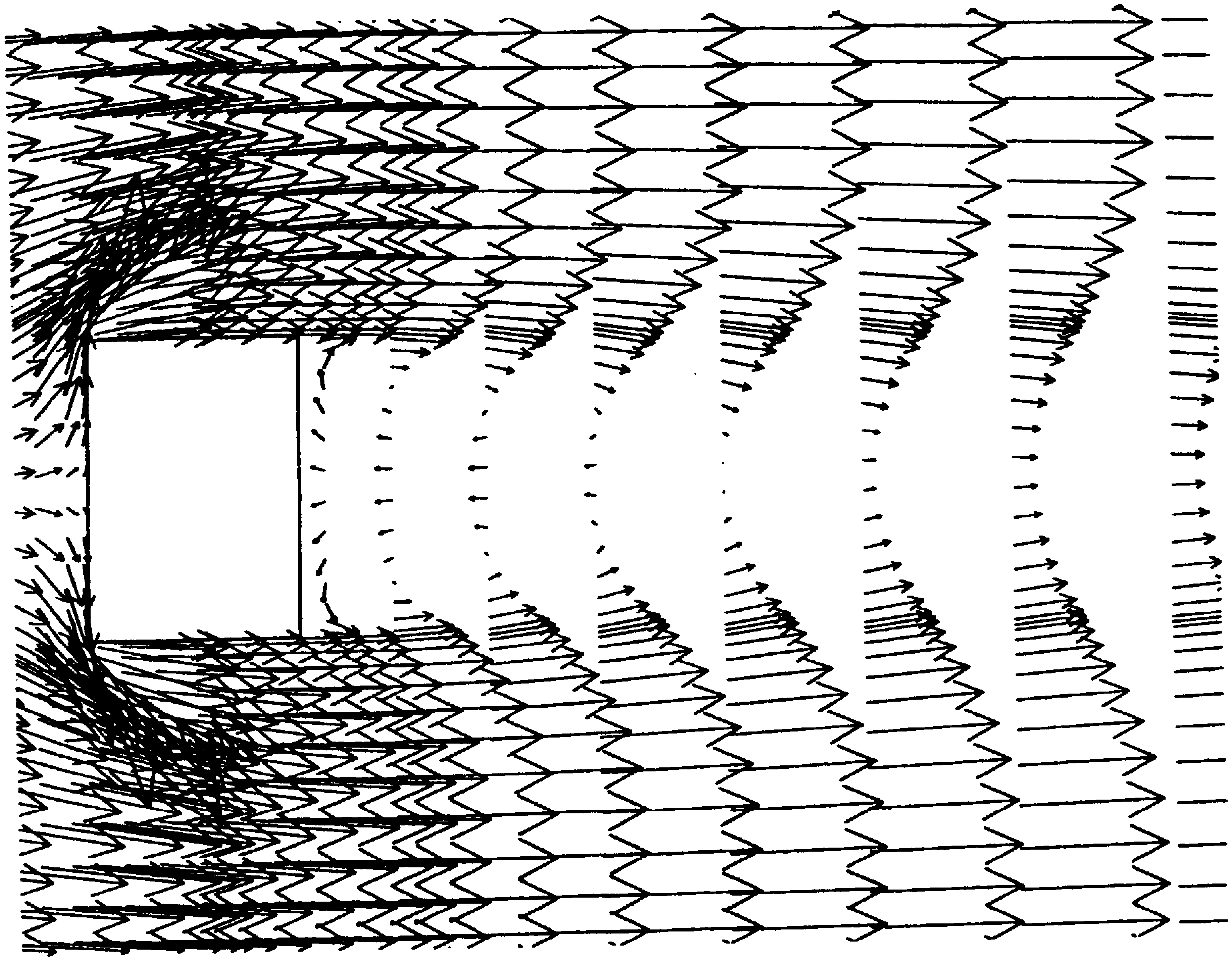


Figure 5.4 *Symmetrical vortex pair in wake of bluff body.*

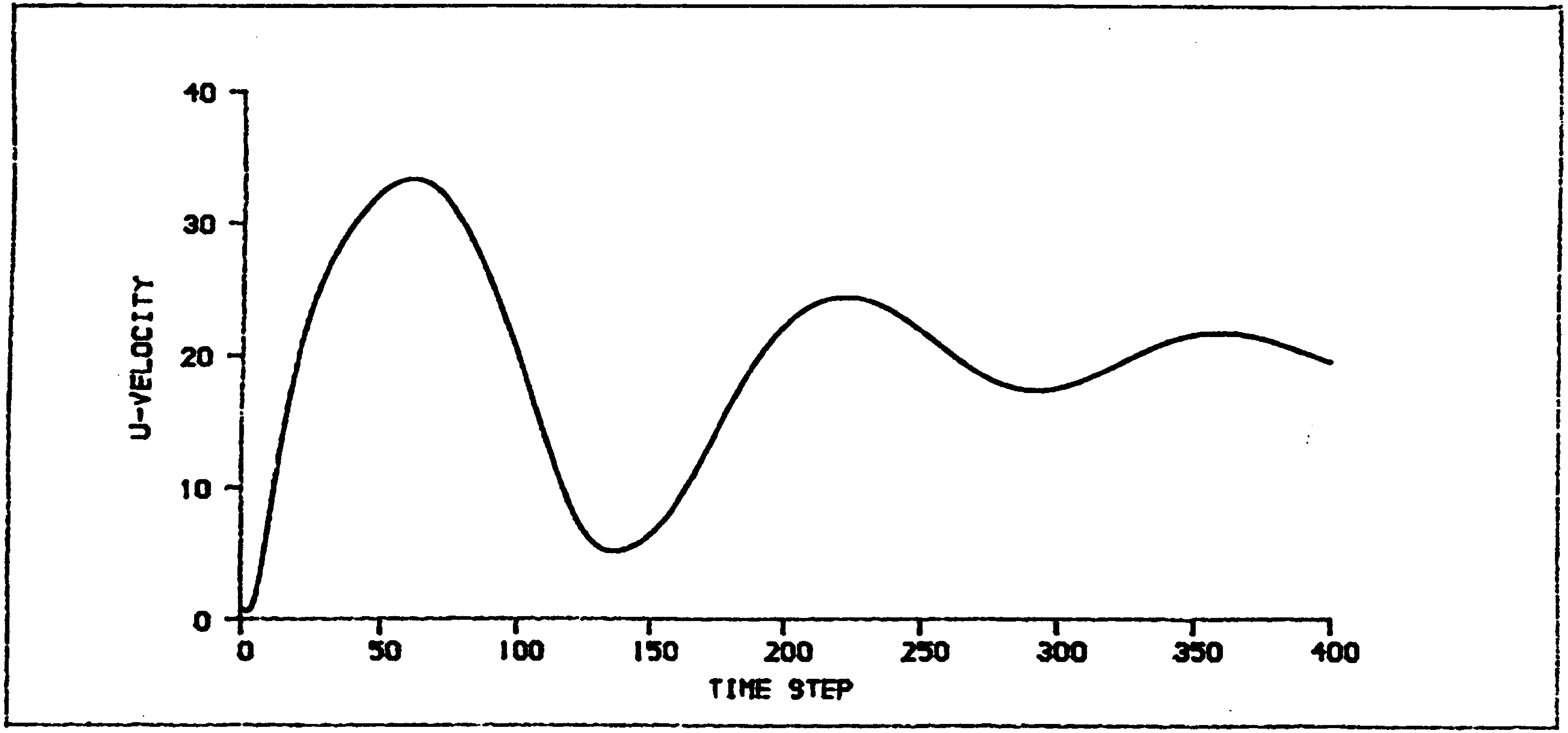


Figure 5.5 *Axial mean velocity transient using HYBRID for a single cell located in the region of vortex shedding.*

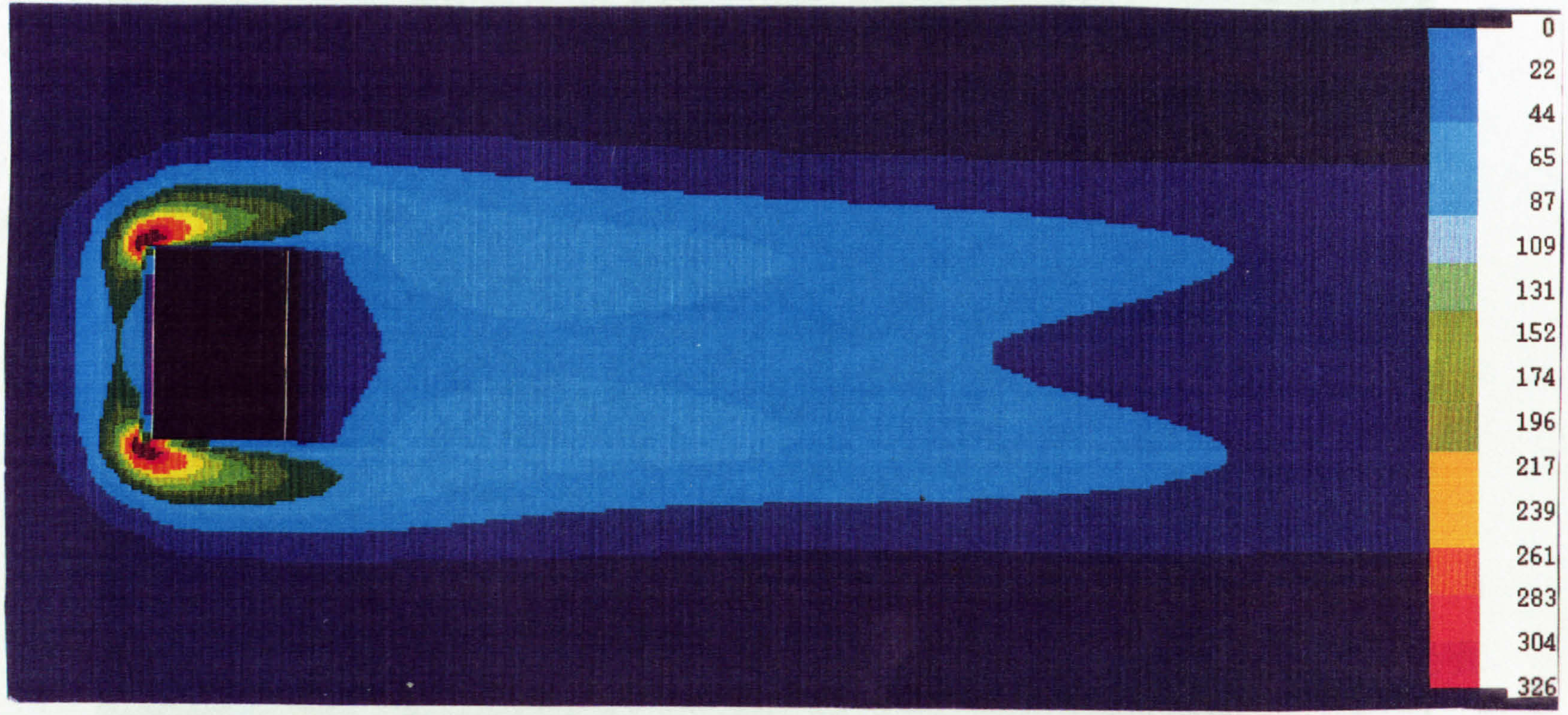


Figure 5.6 *Distribution of turbulence kinetic energy k with HYBRID implemented.*

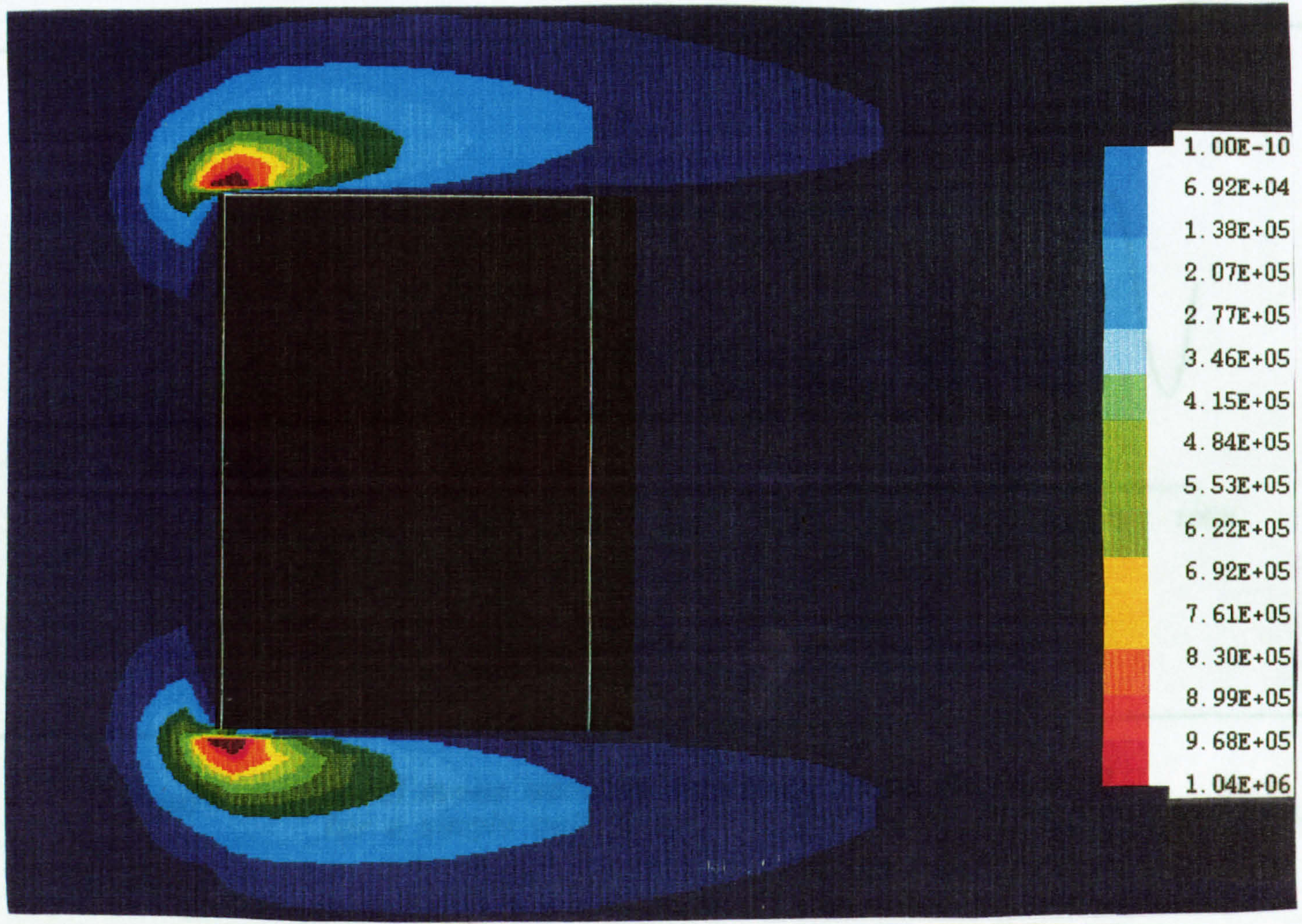


Figure 5.7 *Distribution of turbulence kinetic energy dissipation rate ∈ with HYBRID implemented.*

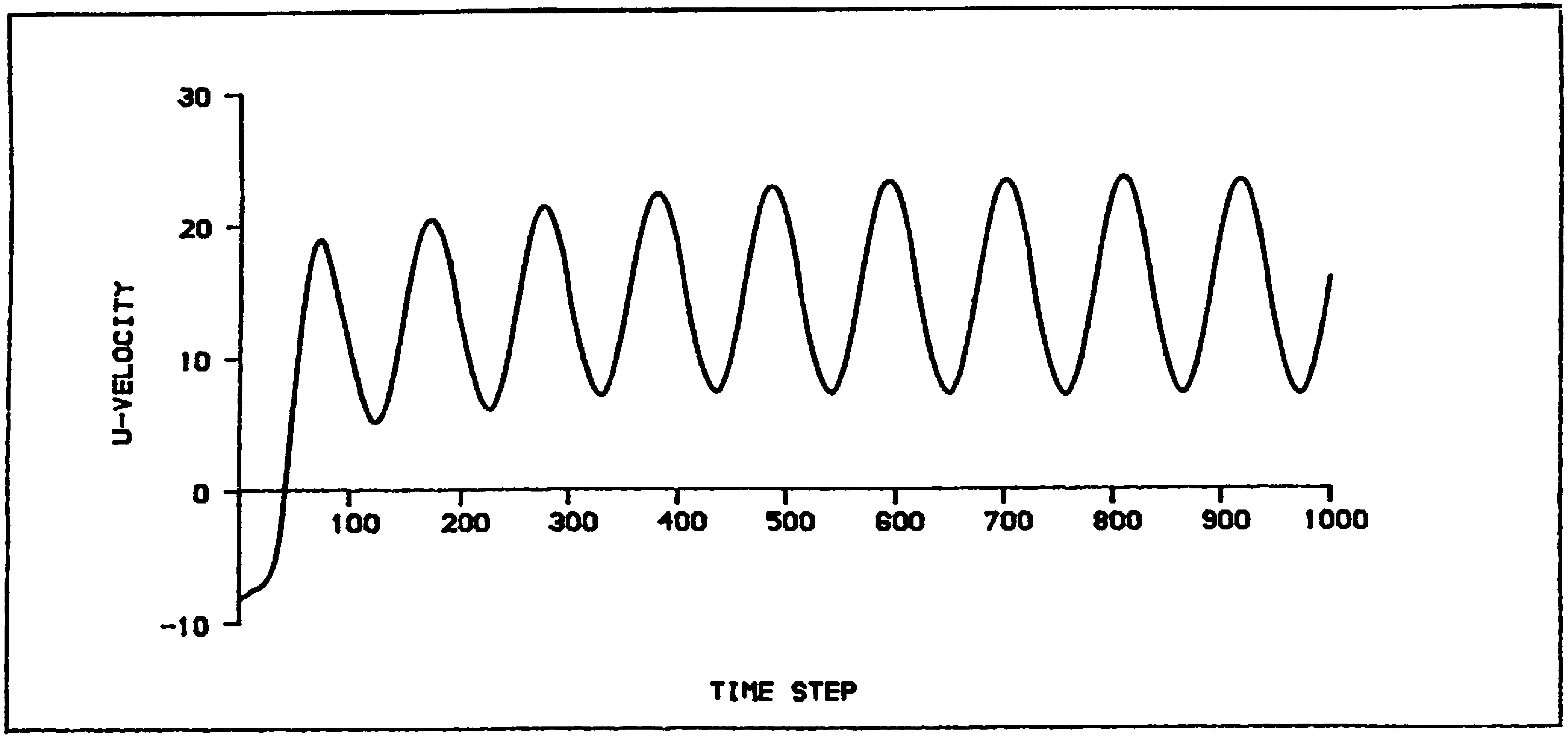


Figure 5.8 *Axial mean velocity transient using SUCCA2D for a single cell located in the region of vortex shedding.*

that the vortex shedding frequency, indicated by a Strouhal number of 0.211 was in reasonable agreement with the experimental results of Mottram and Rawat (Str ~ 0.24). This represents a difference of approximately 13 per cent when comparing the computational and experimental results.

A visualisation of the development of this periodic flow can be seen in the pressure contour plots of Figures 5.9 and 5.10. The initial formation, growth and consequent shedding of the vortices from alternate sides of the bluff body are shown for different time instances within the flow. The simulation shows the rapid vortex shedding formation followed by a stage where it appears there is a gradual increase in the distance between each vortex as each is convected downstream by the main flow. It is also observed that there is an apparent migration of the vortices towards both containing walls. In both Figures the lower view is a continuation of the upper plot.

Figure 5.11 shows the mean velocity vector field in the periodic wake. Together with Figure 5.12, showing contours of mean axial velocity in the turbulent wake, the wave pattern observed in the laminar flow computation is again repeated.

Contours of the turbulence quantities k and ϵ are shown in Figures 5.13 and 5.14 respectively. It is interesting to note that there is only a slight difference in the overall distribution of these variables, in comparison with the hybrid-upwind scheme (Figures 5.6 and 5.7) , even though the SUCCA2D scheme has been applied to improve the accuracy of their convective transport. This confirms the observation of Leschziner(51) that since the transport equations for k and ϵ are source dominated (i.e.

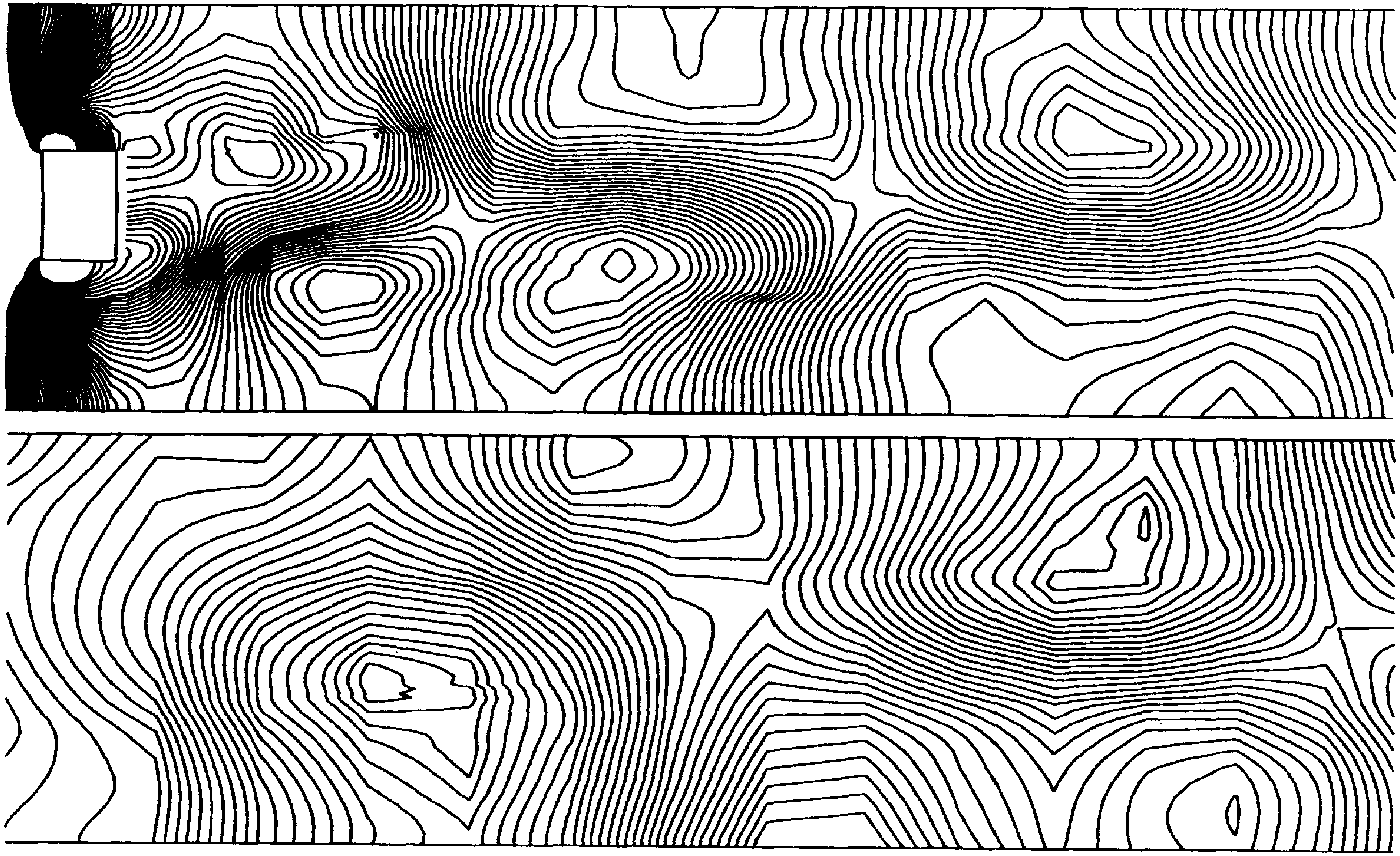


Figure 5.9 *Pressure contours showing vortex shedding development at time step 300.*

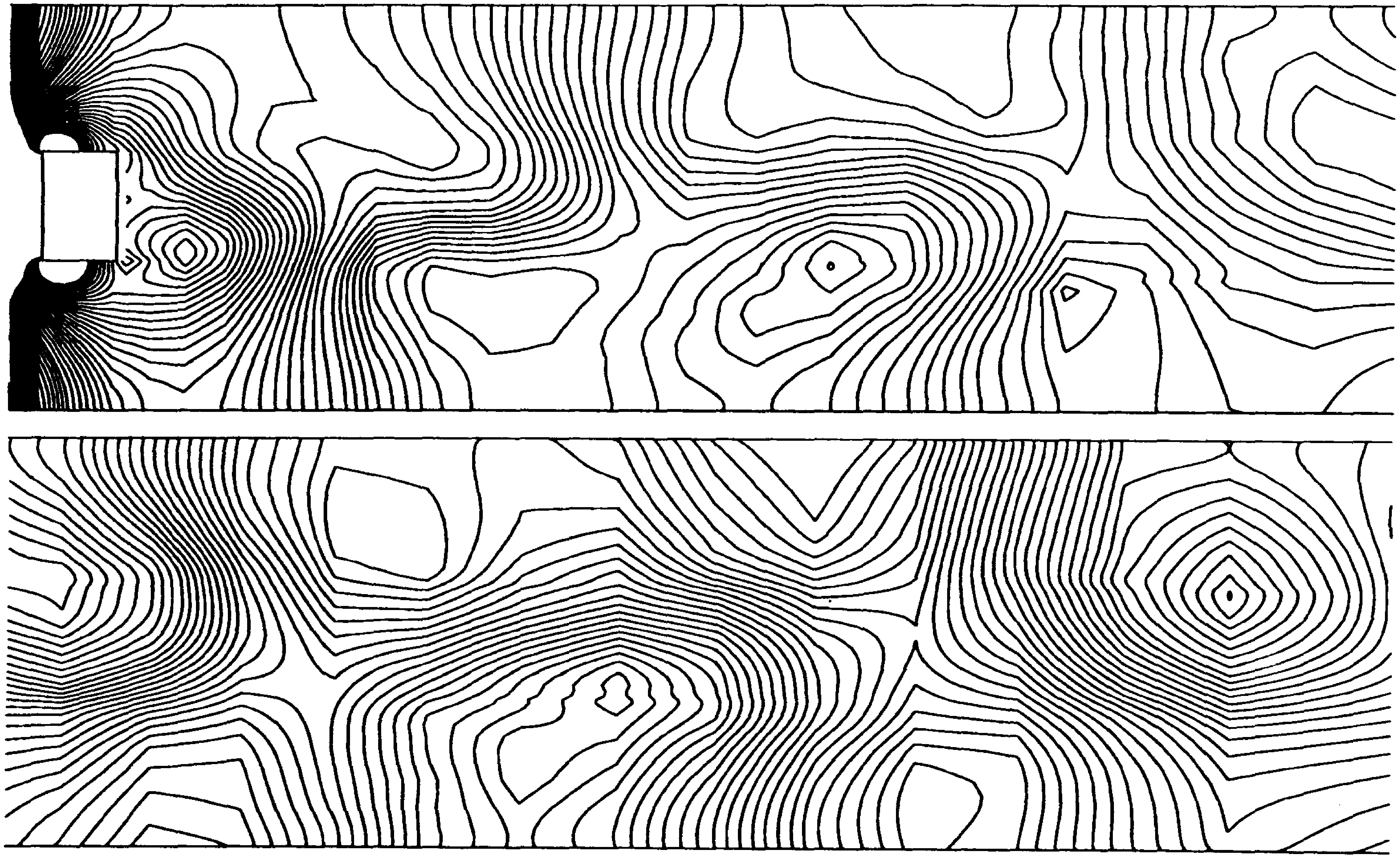


Figure 5.10 *Pressure contours showing vortex shedding development at time step 330.*

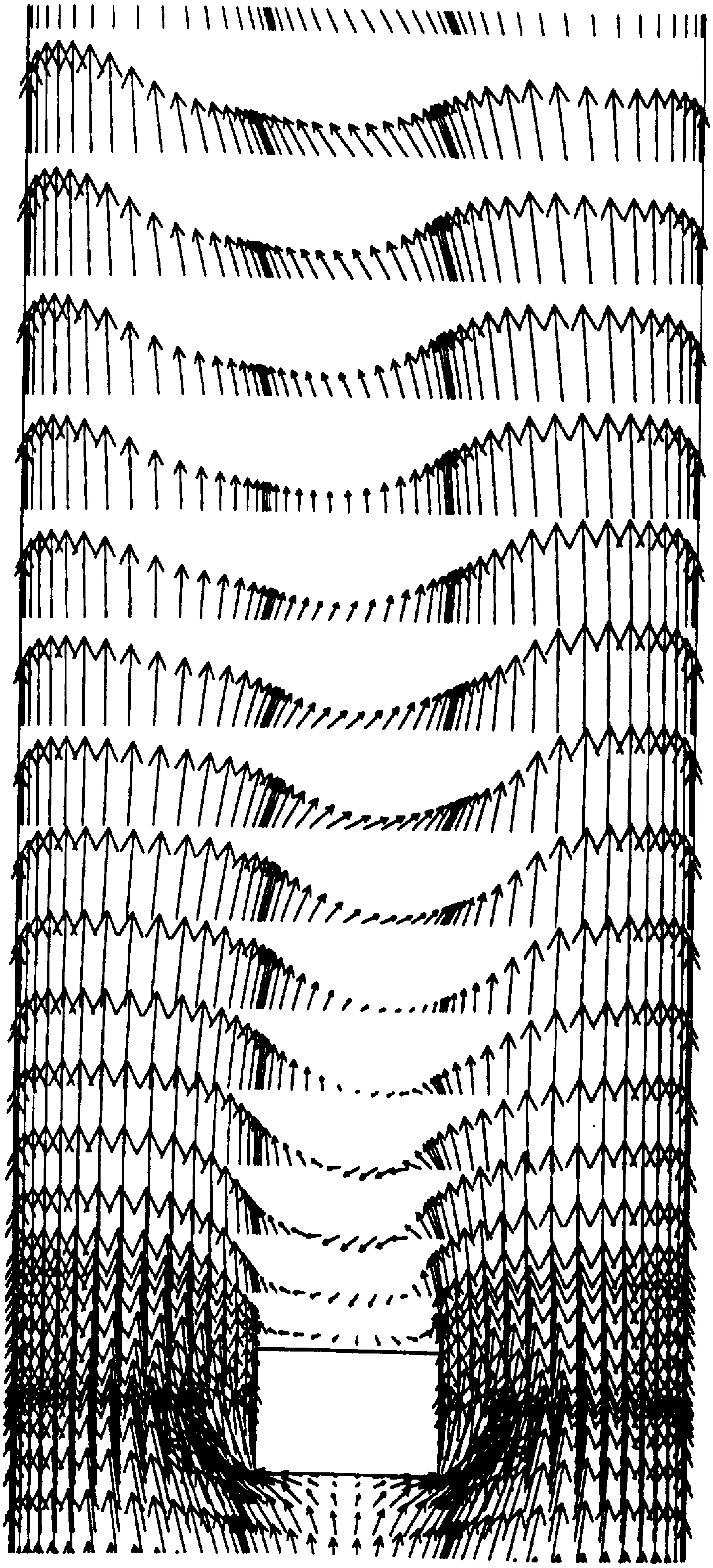


Figure 5.11 Velocity vectors in bluff body wake.

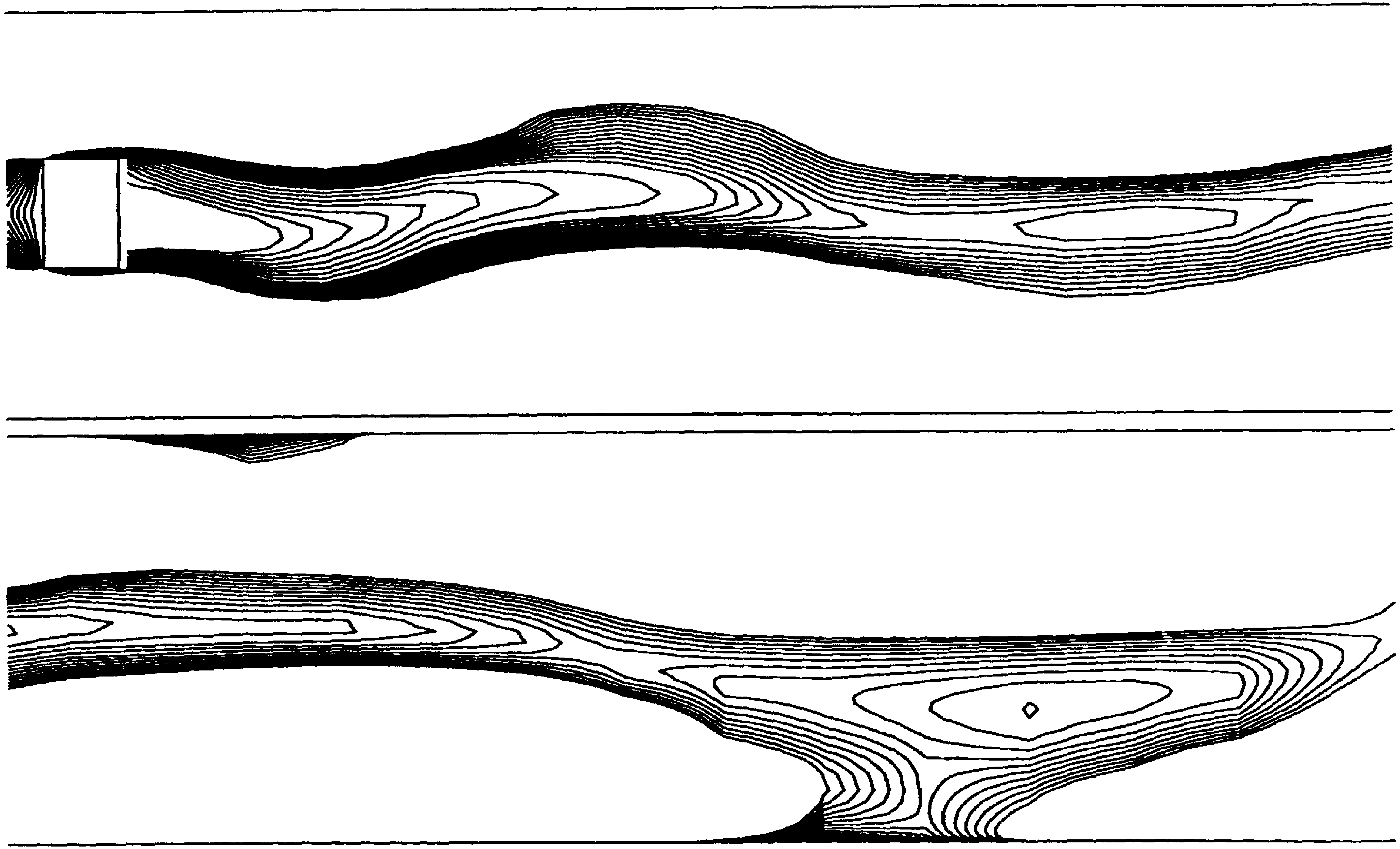


Figure 5.12 *Contours of mean axial velocity showing wave pattern in wake.*

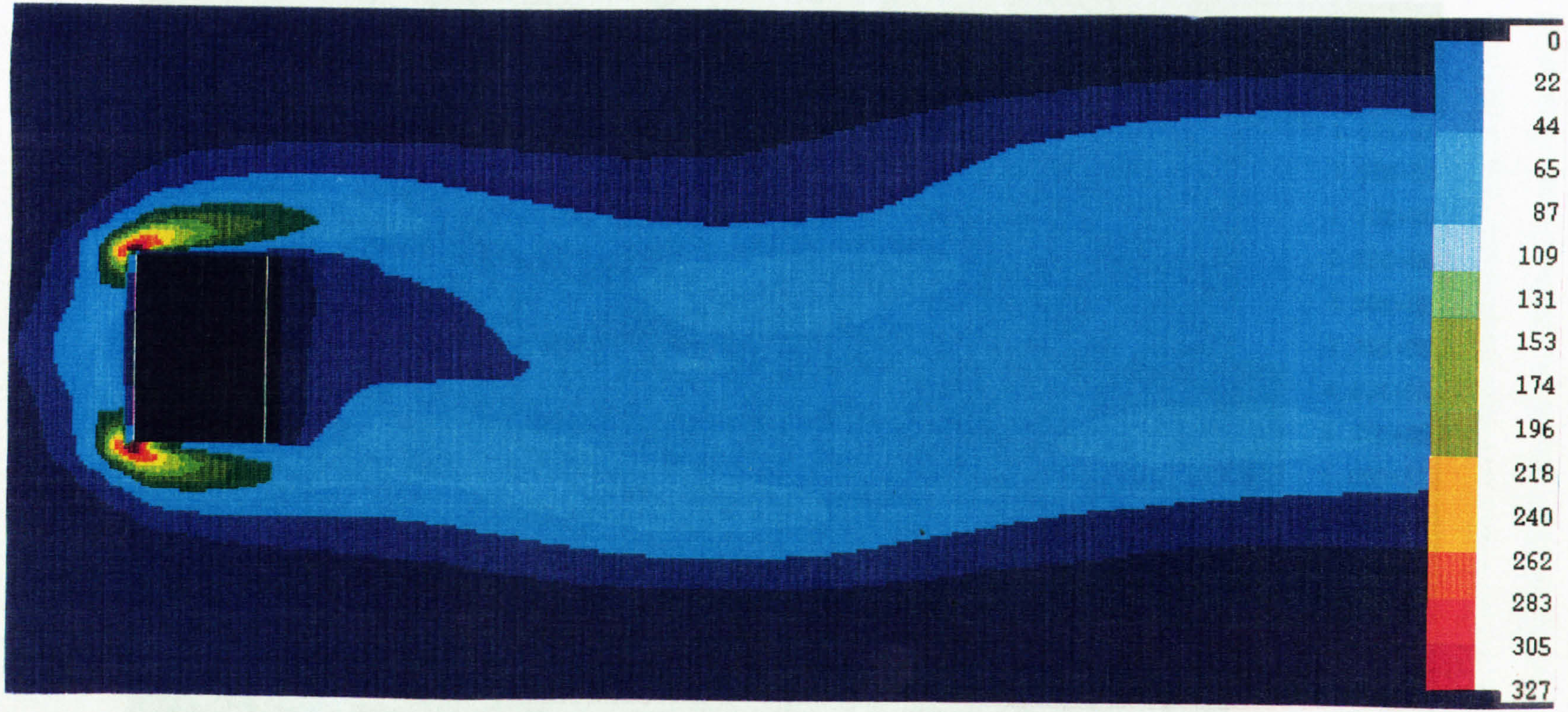


Figure 5.13 *Distribution of turbulence kinetic energy k with SUCCA2D implemented.*

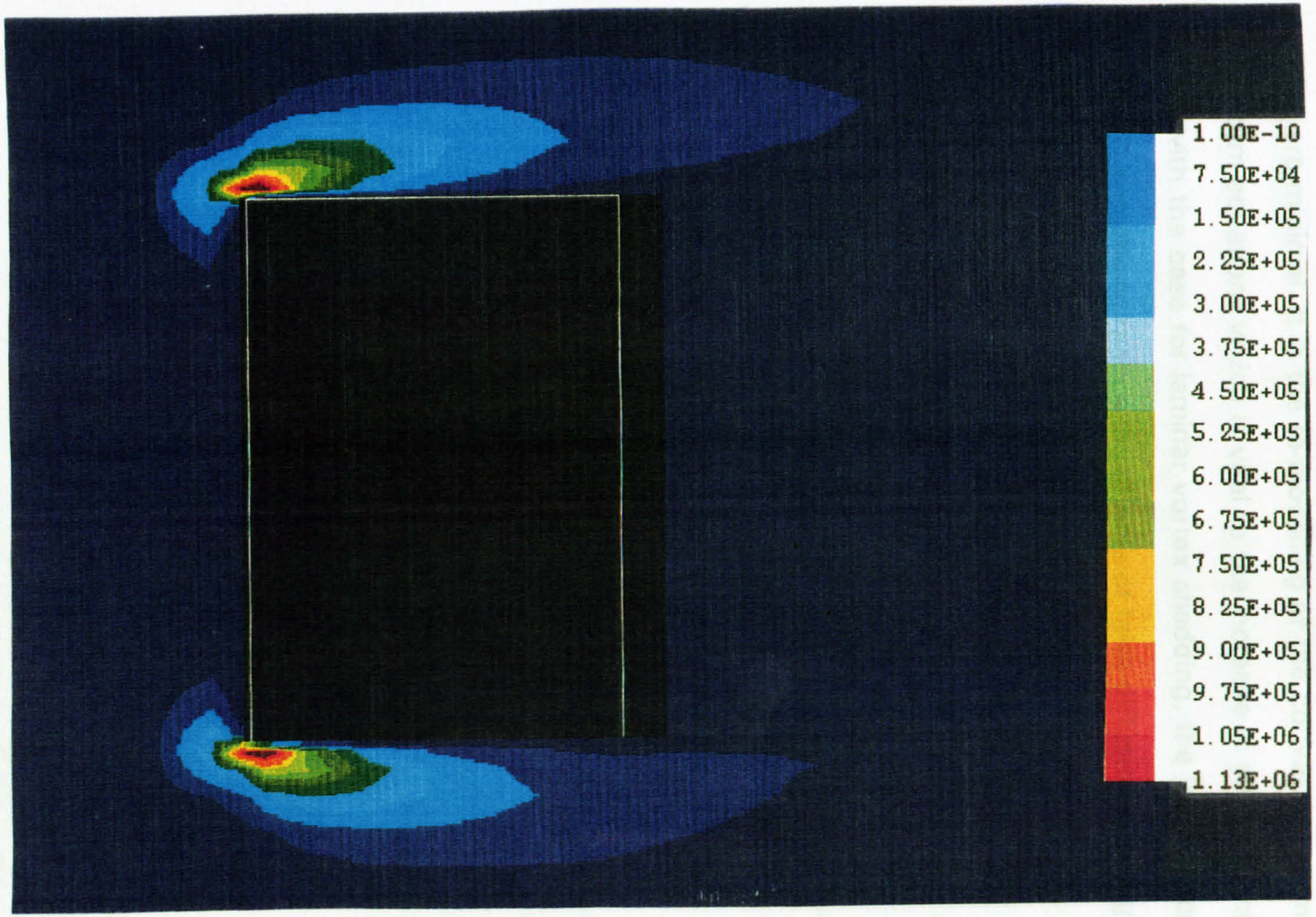


Figure 5.14 *Distribution of turbulence kinetic energy dissipation rate with SUCCA2D implemented.*

production and dissipation terms dominate) then the advantages of re-modelling the convection terms in these equations via any flow-oriented scheme are minimal. It is the re-modelling of the convection terms in the momentum equations which is vital to the increase in solution accuracy.

As with the case for laminar vortex shedding, the importance of the SUCCA2D scheme to reduce the effects of false diffusion was highlighted once more.

CHAPTER 6 THE VORTEX FLOWMETER IN PULSATING FLOW

6.1 Introduction

Pulsatile flow effects on the accuracy of various flow meter types have been documented in the experimental work of Mottram(4) and Mottram and Robati(5). For the vortex flowmeter it has been found that large errors in the calculated Strouhal number can occur under certain pulsating flow conditions. Flow pulsations of an amplitude comparable to that of normal pipe turbulence were found to be sufficient enough to produce significant errors. Also, the pulsation frequency was found to be a critical parameter. When the pulsation frequency lies within a certain range of the normal vortex shedding frequency, the shed vortices tend to synchronize themselves with the pulsating conditions. This phenomenon is known as 'locking-in'(4), as outlined in Chapter 1.

In this chapter, the PHOENICS code is used in an attempt to simulate the effects of pulsating flow on the vortex shedding process. The equations solved and convergence criteria are identical to those introduced in the two-dimensional plane turbulent vortex shedding study described in Chapter 5. The Q1 file for this study is presented in Appendix G.

6.2 Two-Dimensional Plane Pulsating Flow

The case of two-dimensional plane turbulent pulsating flow was considered. With the bluff body removed, the problem resorted to being one of a plane channel flow under pulsating conditions. This case was considered such that the simulation of pulsating flow could be initially isolated and analyzed prior to the study of vortex shedding under pulsating flow conditions. Although the modelled working fluid in this case is air, an incompressible flow regime was assumed. Following Mottram(4), cyclic variations in density are neglected on the basis that the most significant parameters in the pulsating vortex shedding analysis are the pulsation frequency and the mean velocity pulsation amplitude. The Reynolds number of the flow, based on the pipe diameter and mean bulk velocity, was 100000.

6.2.1 Geometrical Configuration

The geometry for this study was as described in the experimental study of Mottram(4). This geometry is identical to that outlined for the two-dimensional plane turbulent vortex shedding analysis described in Chapter 5, except that the inlet length to the bluff body was increased to 56.0 diameters in accordance with the geometry of the pulsating experimental analysis.

6.2.2 Boundary Conditions

Special consideration was given to the upstream free boundary at the inlet of the computational domain. On the upstream faces of the computational cells at this location a sinusoidal input of mass and axial momentum was applied which had the following form:

$$\bar{U}_p = \bar{U}_0 + A_0 \sin(\omega \Delta t) \qquad 6.1$$

where $\bar{U}_p$ is the mean axial pulsating velocity at any instant in time

$\bar{U}_0$ is the mean axial velocity under steady flow conditions

A_0 is the mean axial pulsating velocity amplitude

ω is the pulsation frequency (rad/sec)

Δt is the time step (secs)

Such an input formulation describes the cyclic temporal variation of mass and axial momentum, based on a sinusoidal profile, around their mean quantities. The mean quantities of mass and momentum are themselves based upon the bulk mean velocity $\bar{U}_0$. The basis for the sinusoidal variation was an attempt to model the periodic pulsations that occur in real flow situations from reciprocating or centrifugal positive displacement machines operating upstream of the vortex flowmeter. Equation 6.1 has been formulated in such a manner that the user has input control over the pulsation amplitude A_0 and the pulsation frequency ω . The additional GROUND coding for the inlet pulsatile conditions is given in Appendix H.

The convective transport of all flow dependent variables was again modelled using the SUCCA2D scheme as detailed in Chapters 4 and 5.

A time increment of 4.E-05 secs was utilised in an identical manner to the turbulent vortex shedding analysis of Chapter 5. All other boundary conditions were as outlined in Chapter 5 and a non-uniform computational mesh of 104X x 50Y was employed. The time scale of CPU expenditure was approximately 3 days on a Sun sparc1 + computer.

6.2.3 Pulsating Channel Flow Results

Figure 6.1 shows the pulsation profile of axial momentum U1 about its mean value. This transient plot is for a single inlet free boundary cell located on the channel centre-line. In this case the inlet pulsation amplitude was such that $\bar{U}_{rms}/\bar{U}_o$ =0.194 and the inlet pulsation frequency f_p = $2.0\, f_{vs}$. Having confirmed the operation of the GROUND

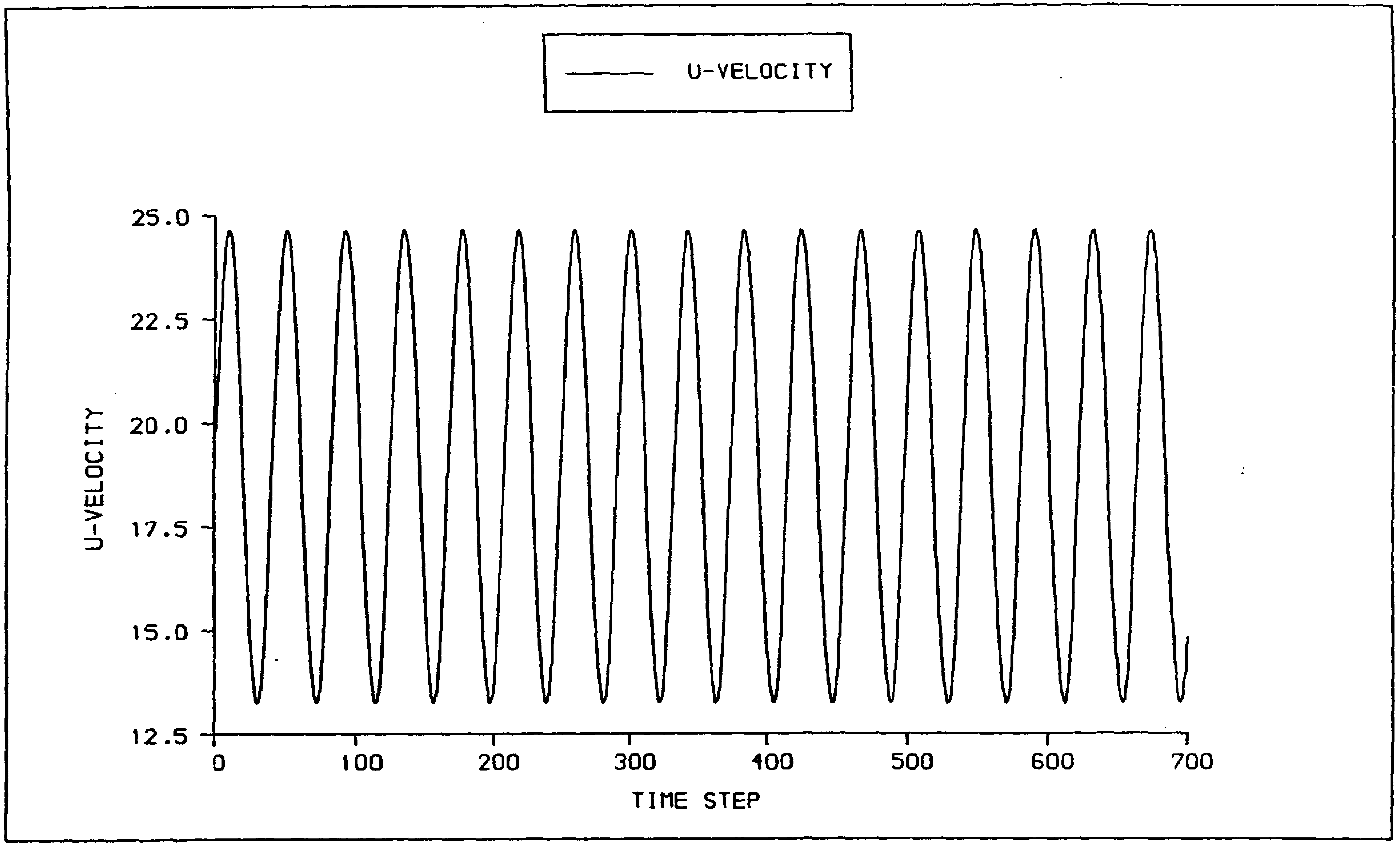


Figure 6.1 ***Axial velocity transient for pulsating channel flow.*** $f_p/f_{vs} = 2.0 : \bar{U}_{rms}/\bar{U} = 0.194$.

coding for pulsating channel flow, the same inlet boundary conditions were then used for the analysis of turbulent vortex shedding in pulsating flow with the bluff body now in place.

6.3 Turbulent Vortex Shedding in Pulsating Flow

With the pulsating inlet boundary conditions now implemented, the case of two-dimensional plane channel turbulent vortex shedding in pulsating flow was considered.

An inlet pulsation amplitude of $\bar{U}_{rms}/\bar{U}_o$ =0.194 was chosen and four frequency ratios f_p/f_{vs} of values 1.5,2.0,2.5 and 3.1 were analyzed in an attempt to model the experimental results of Mottram(4). The results of Mottram's work for the above case are shown in Figure 6.2. This Figure shows the percentage change in Strouhal number with frequency ratio f_p/f_{vs} and highlights two locking curves, one where the vortex shedding frequency under pulsating flow f_{vp} is equal to the inlet pulsation frequency f_p and the other where $f_{vp} = 1/2\, f_p$. The computational analysis was undertaken in an attempt to predict the second locking curve $f_{vp} = 1/2\, f_p$.

The variations in mean axial velocity with time for the pulsating vortex flow are shown in Figures 6.3 to 6.6 for various frequency ratios. These plots show the axial velocity transients for a single computational cell located in the region of vortex shedding and highlight the mechanism of pulsating flow influence on the formerly sinusoidal vortex shedding transient of Figure 5.8.

A frequency spectrum analysis of the four velocity transients using Fast Fourier Transform software was carried out and the results are shown

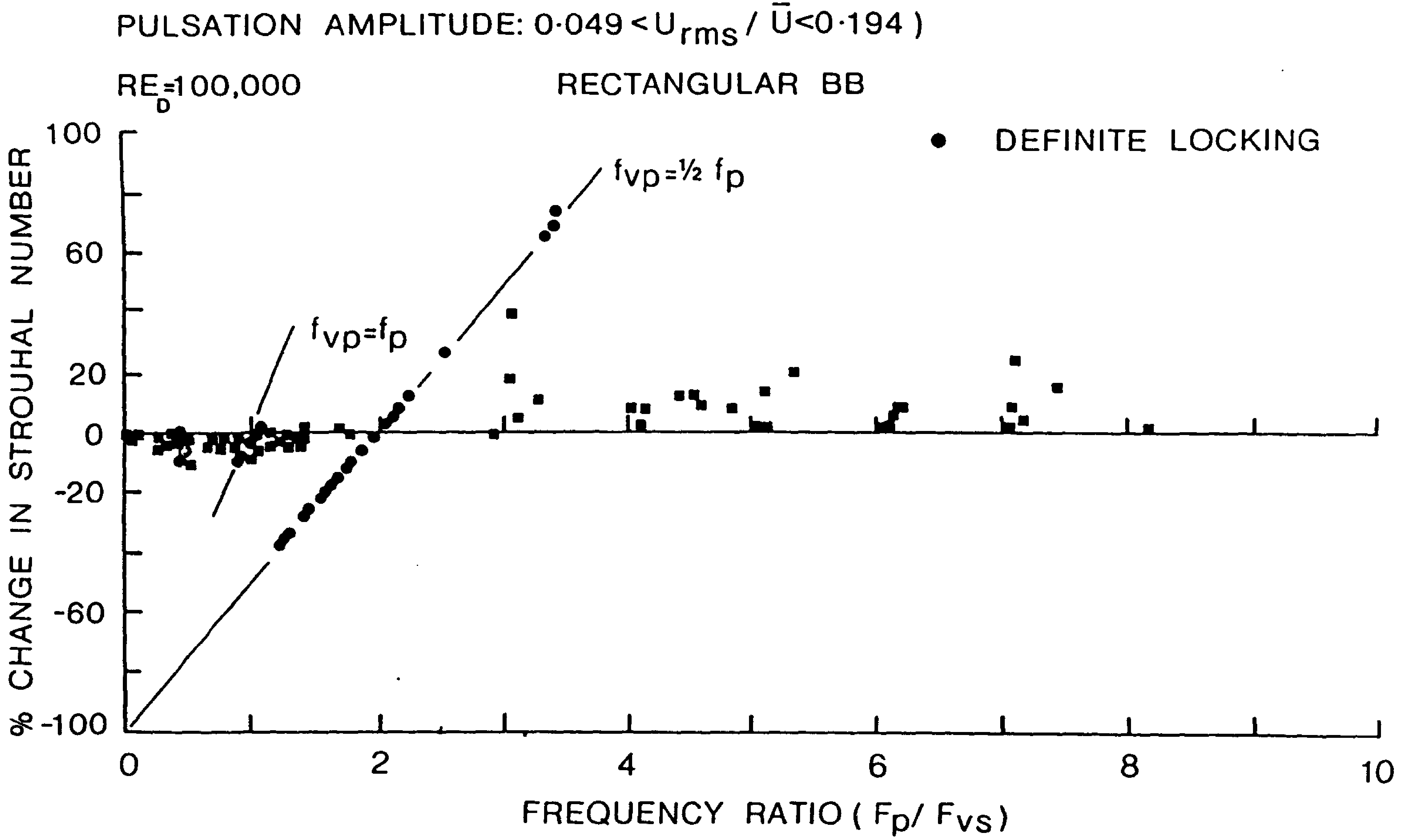


FIG. 6.2% CHANGE IN STROUHAL NUMBER VS FREQUENCY RATIO (F_p/ F_{vs})

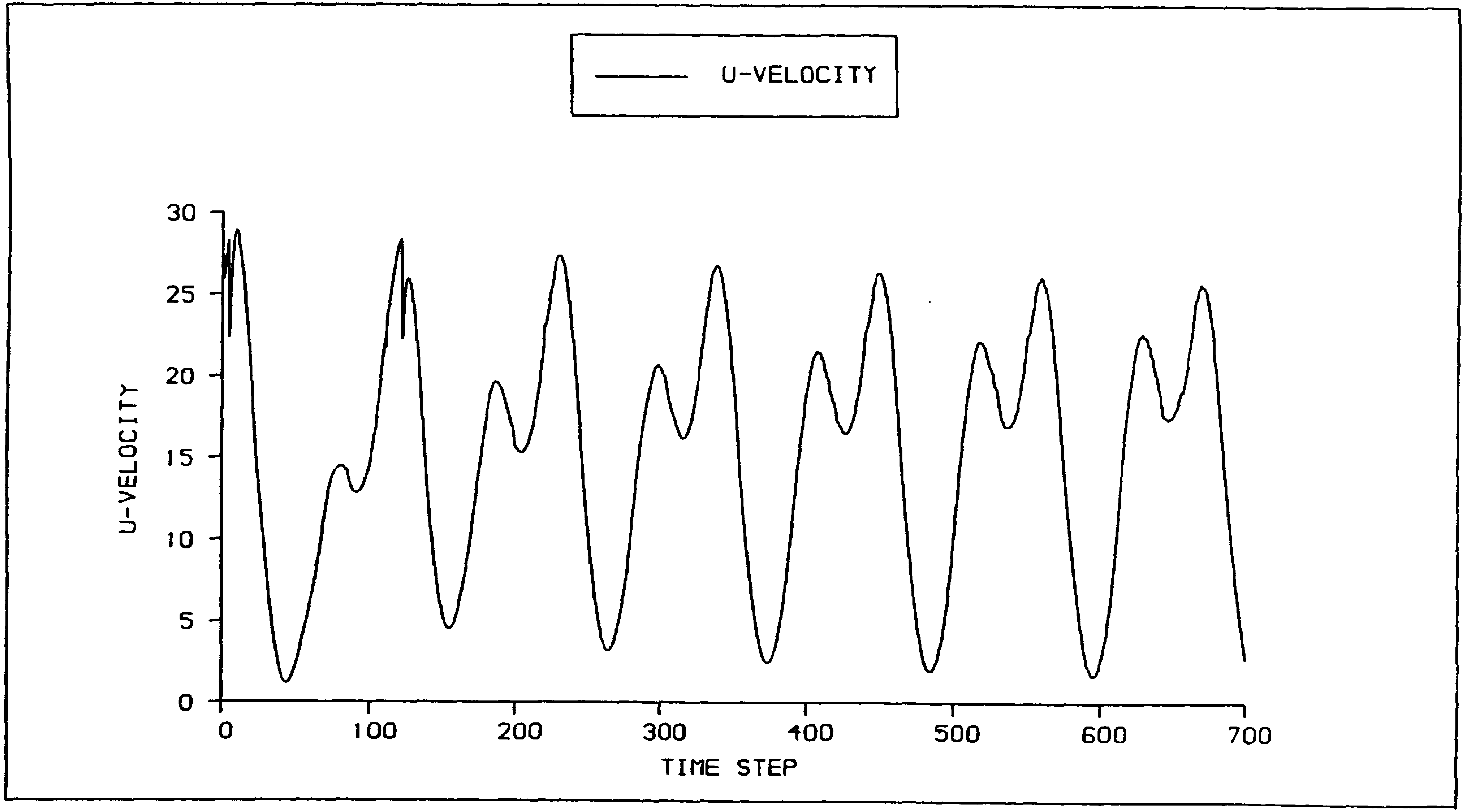


Figure 6.3 *Vortex shedding transient in pulsating flow.* $f_p/f_{vs} = 1.5 : \bar{U}_{rms}/\bar{U} = 0.194$.

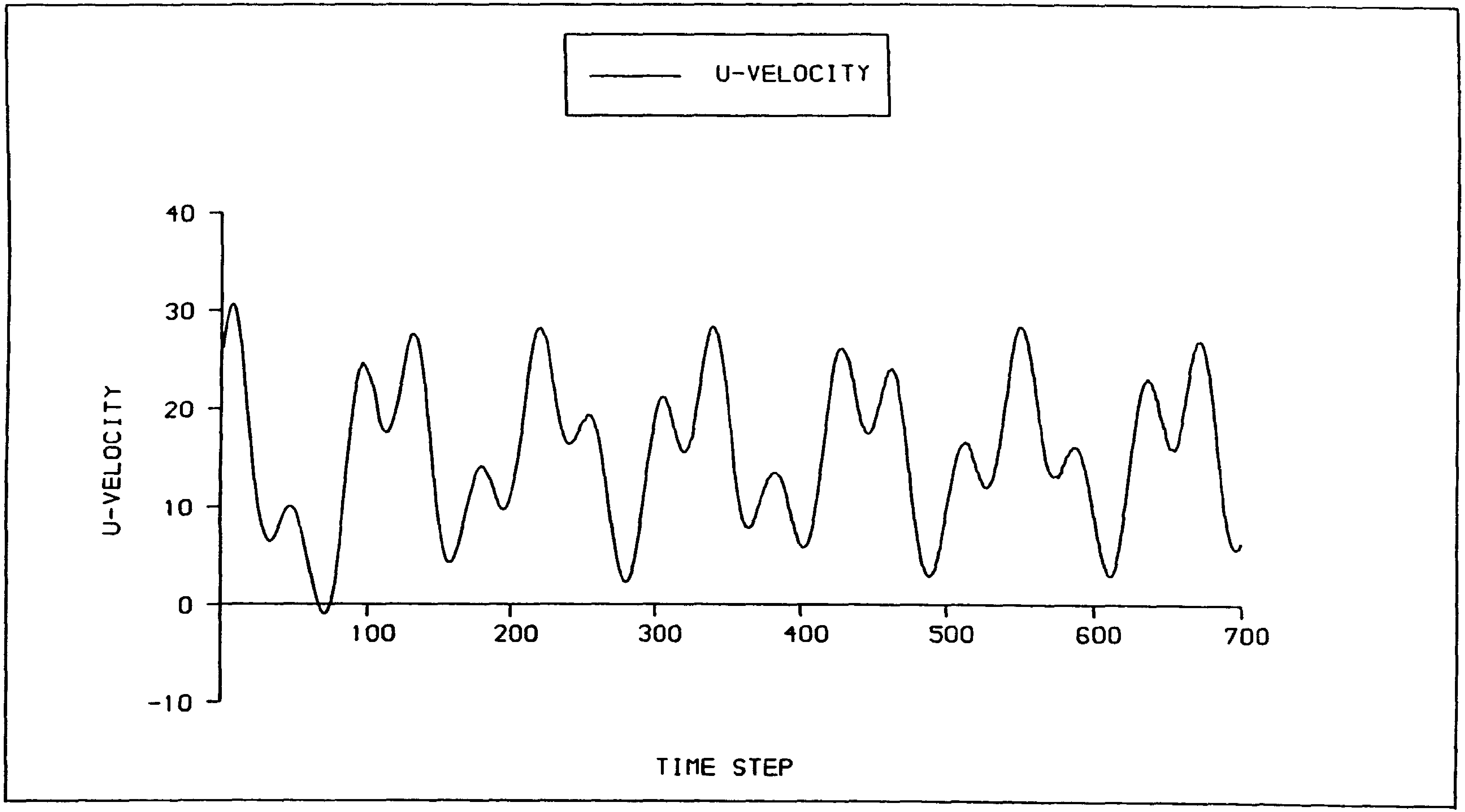


Figure 6.4 ***Vortex shedding transient in pulsating flow.***
$f_p/f_{vs} = 2.0 : \bar{U}_{rms}/\bar{U} = 0.194$.

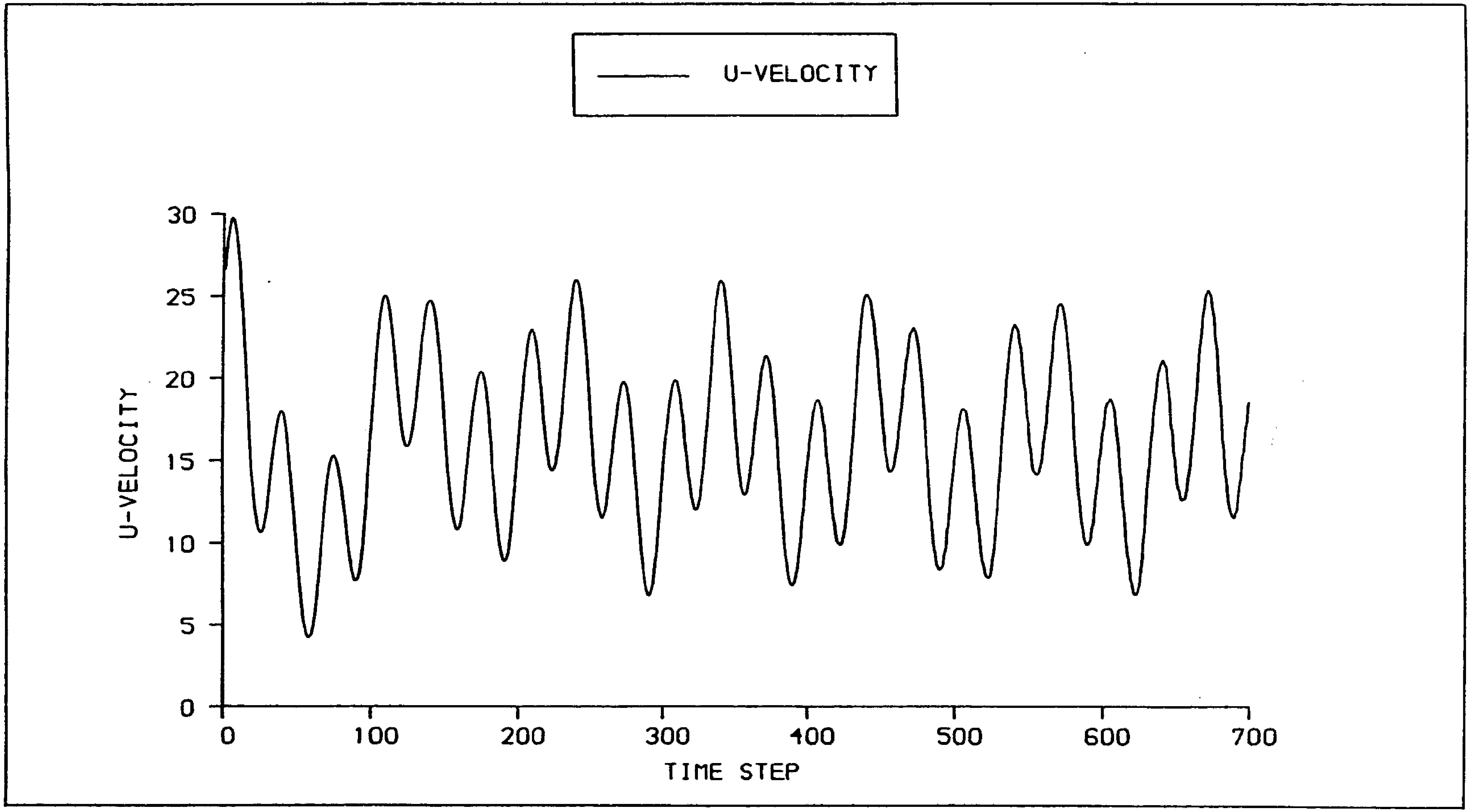


Figure 6.5 *Vortex shedding transient in pulsating flow.*
$f_p/f_{vs} = 2.5 : \bar{U}_{rms}/\bar{U} = 0.194$.

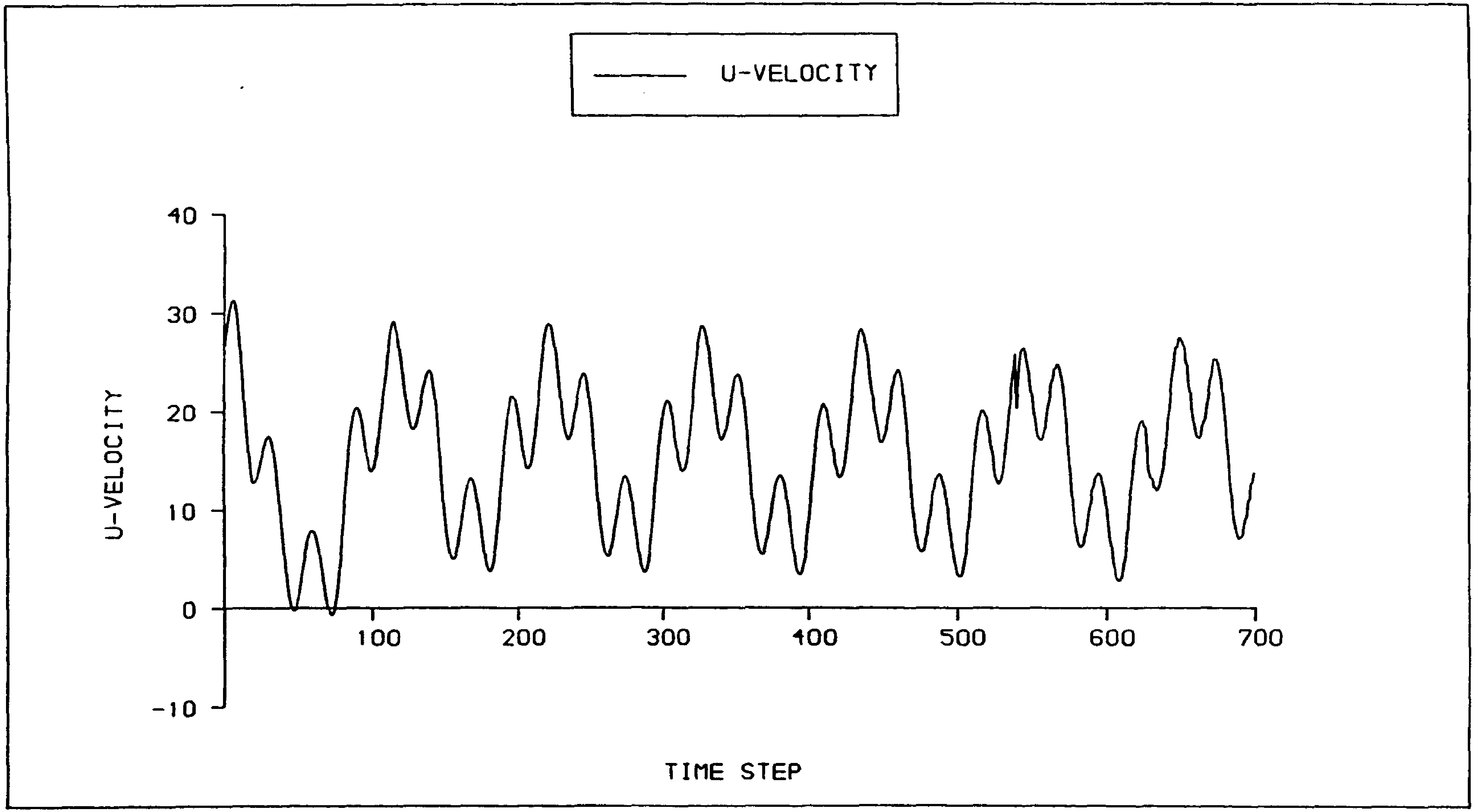


Figure 6.6 ***Vortex shedding transient in pulsating flow.***
$f_p/f_{vs} = 3.1 : \bar{U}_{rms}/\bar{U} = 0.194$.

in Table 6.1.

f_{vs} (Hz)	f_p (Hz)	$\frac{f_p}{f_{vs}}$	Exp. locking frequency (Hz) $f_{vp} = 1/2\, f_p$	Comp. lock frequency (Hz) f_{vp}	% ch. in Str. Comp.	% ch. in Str. Exp.
301	**452**	**1.5**	**226**	**221**	**-27**	**-28**
301	**602**	**2.0**	**301**	**585**	**+94**	**0**
301	**753**	**2.5**	**377**	**757**	**+151**	**+22**
301	**933**	**3.1**	off curve	**903**	**+199**	**+50**

Table 6.1 Comparison of experimental and computational results for turbulent vortex shedding in pulsating flow.

The above results show that the correct locking frequency is predicted for the lowest frequency ratio f_p/f_{vs} = 1.5. At higher frequency ratios, however, the computed vortex frequencies are seen to deviate substantially from the experimental locking curve $f_{vp} = 1/2\, f_p$ although the trend of negative to increasing positive change in Strouhal number is maintained. The results also show a tendency for the computed vortex frequencies, at higher values of f_p/f_{vs} to follow the initial locking curve $f_{vp} = f_p$.

From the above results it appears that it will only be possible to predict the correct locking frequency at lower values of the frequency ratio f_p/f_{vs} i.e. f_p/f_{vs} ~1.5.

CHAPTER 7 THREE-DIMENSIONAL TURBULENT VORTEX SHEDDING

7.1 Introduction

The turbulent vortex shedding work contained in the preceding chapters has focused on a two-dimensional flow analysis. Although predictions of global flow parameters such as the Strouhal number have been reasonably well predicted, a loss of accuracy will have occurred due to the two-dimensional geometry constriction. The experimental work of Mottram and Rawat(52), on which the PHOENICS simulation is based, consists of a pipe flow with a linear bluff body blockage and is thus three-dimensional in nature. This implies that the prediction of the complex three-dimensional turbulent structures in the bluff body wake, as described by Perry and Watmuff(25), cannot be accurately captured using a two-dimensional model. In order to account for the additional dimension a three-dimensional model was constructed and is described in Section 7.5. The Q1 file for this study is presented in Appendix J.

The multidimensional false diffusion errors that occur in two-dimensional flows are further enhanced with the extra space dimension. Modification of the original SUCCA2D scheme into three-dimensions did possess some inherent computational difficulties. There were now some twenty additional possible upwind cell locations (as opposed to four for the 2D grid cell cluster) and the numerical task involved in accounting for these cells, although not insurmountable, had obvious book-keeping difficulties. The development of this and other aspects of the computational software to reduce the effects of three-dimensional false diffusion is described in the following section.

7.2 The SUCCA3D Scheme

The derived form of the convection coefficients for the SUCCA3D scheme is exactly analogous to the SUCCA2D form. Consider the solution molecule for the computational cell P, as shown in Figure 7.1. Two cases are then considered, namely:

1) $\dot{m}_w \geq \dot{m}_s \geq \dot{m}_l \geq 0$
2) $\dot{m}_w \geq \dot{m}_l \geq \dot{m}_s \geq 0$

If $\dot{m}_w \geq \dot{m}_s \geq \dot{m}_l \geq 0$, the streamline through P passes through the triangle W.SW.SWL and the scheme should be of the form:

$$C_p \phi_P = C_w \phi_W + C_{sw} \phi_{SW} + C_{swl} \phi_{SWL} \qquad 7.1$$

If, on the other hand, $\dot{m}_w \geq \dot{m}_l \geq \dot{m}_s \geq 0$ then the streamline through P passes through W.WL.SWL and the scheme is of the form:

$$C_p \phi_P = C_w \phi_W + C_{wl} \phi_{WL} + C_{swl} \phi_{SWL} \qquad 7.2$$

This approach has the following merits:

(i) ϕ_W is always retained, ensuring that there is at least some coupling of the finite volume equations for nearby cells.

(ii) only two other ϕ 's are ever needed, so only two extra source terms must be introduced.

(iii) the coefficients can be made 'correct' for the special cases when the streamline through P actually passes through SW, WL or SWL.

The adopted form of the convection coefficients is as follows, for the case $\dot{m}_w \geq \dot{m}_l \geq \dot{m}_s \geq 0$:

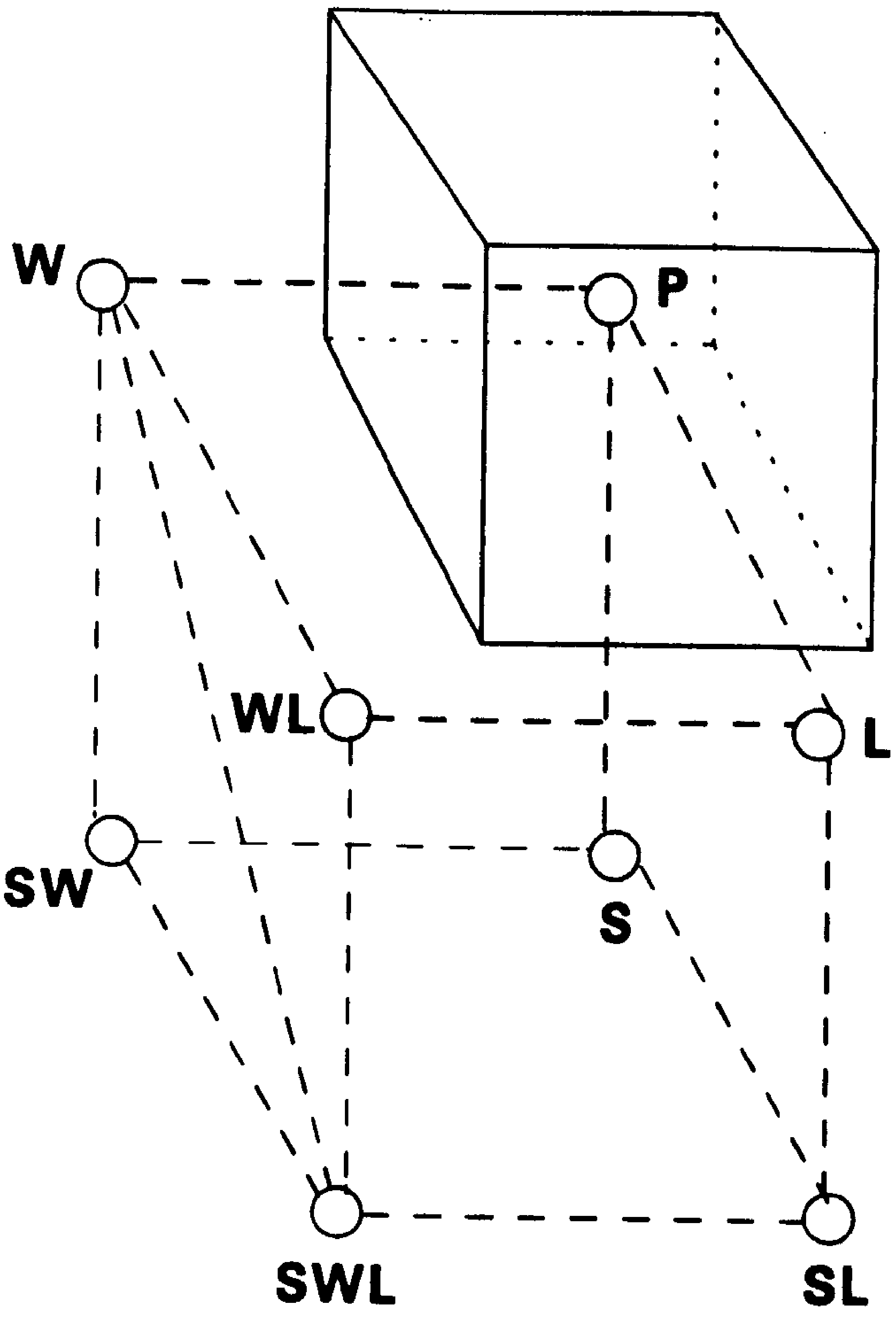


Figure 7.1 *Solution molecule for the SUCCA3D scheme.*

$$C_w = \left(1 - \frac{\dot{m}_l}{\dot{m}_w}\right)(\dot{m}_l + \dot{m}_w + \dot{m}_s) \qquad 7.3$$

$$C_{wl} = \left(\frac{\dot{m}_l}{\dot{m}_w} - \frac{\dot{m}_s}{\dot{m}_w}\right)(\dot{m}_l + \dot{m}_w + \dot{m}_s) \qquad 7.4$$

$$C_{swl} = \frac{\dot{m}_s}{\dot{m}_w}(\dot{m}_l + \dot{m}_w + \dot{m}_s) \qquad 7.5$$

Such a formulation allows for the situation where part of the flow may indeed be two-dimensional, e.g. $\dot{m}_s = 0$, and in this case the coefficients correct themselves by reducing to the SUCCA2D form.

The SUCCA3D scheme has also been constructed such that it will observe all of the discriminating criteria required for a new convection algorithm, as outlined in Section 4.4.1.2 of Chapter 4 for the 2D algorithm.

Another feature of the of the SUCCA3D algorithm is that it involves one **face** neighbour (e.g. W), one **edge** neighbour (e.g. SW) and one **corner** neighbour (e.g. SWL). Therefore, provision is made within the PHOENICS special GROUND coding for two source contributions; one from the solution molecule **edges** and one from the **corners**. The implementation of the SUCCA3D scheme is discussed in Section 7.6 and the GROUND FORTRAN coding for the SUCCA3D scheme itself is presented in Appendix M.

7.3 3D Convection of a Scalar Quantity

7.3.1 3D Cavity Cross-Flow

Testing of the SUCCA3D scheme was executed on a problem concerning the pure convection of a scalar variable, in this case species concentration C1, across a three-dimensional cavity of 11X x 11Y x 11Z grid cells. Such a problem is directly analogous to the 2D scalar convection problem described in Chapter 4. Inlet concentrations of 1.0, 0.0 and 0.0 were prescribed and physical diffusion of C1 was again de-activated.The SUCCA3D scheme was applied to the convective transport of C1 over the computational region containing the internal and outflow boundary cells. This was necessary due to the fact that at inflow boundaries there are no upwind corner or edge cells to be included within the scheme. Such a condition will only occur where the inlet boundary flow is skewed (as in this case). Where the inlet boundary flow is non-skewed, as in most conventional analyses, the limits of either the SUCCA2D or SUCCA3D application should be the entire computational domain. Figure 7.2 shows the geometry and boundary conditions for the scalar convection analysis and the Q1 file is presented in Appendix K.

The results of the flow analysis can be seen in Figures 7.3 and 7.4, showing outflow concentration profiles. In Figure 7.3, the PHOENICS default hybrid scheme has been used for the discretisation of scalar convective transport. With physical diffusion of C1 absent, the convection scheme changes from hybrid to upwind and the excessive multi-dimensional false diffusion errors associated with the use of the upwind scheme are highlighted in the large degree of C1 smearing observed. With the SUCCA3D scheme implemented for C1 convective transport, Figure 7.4 shows a significant reduction in the degree of false diffusion of C1. Both plots show results for the inflow velocity conditions of U1 = V1 = W1 = 1.0, simulating a cross-diagonal flow, which lies in the region of maximum false diffusion occurrence.

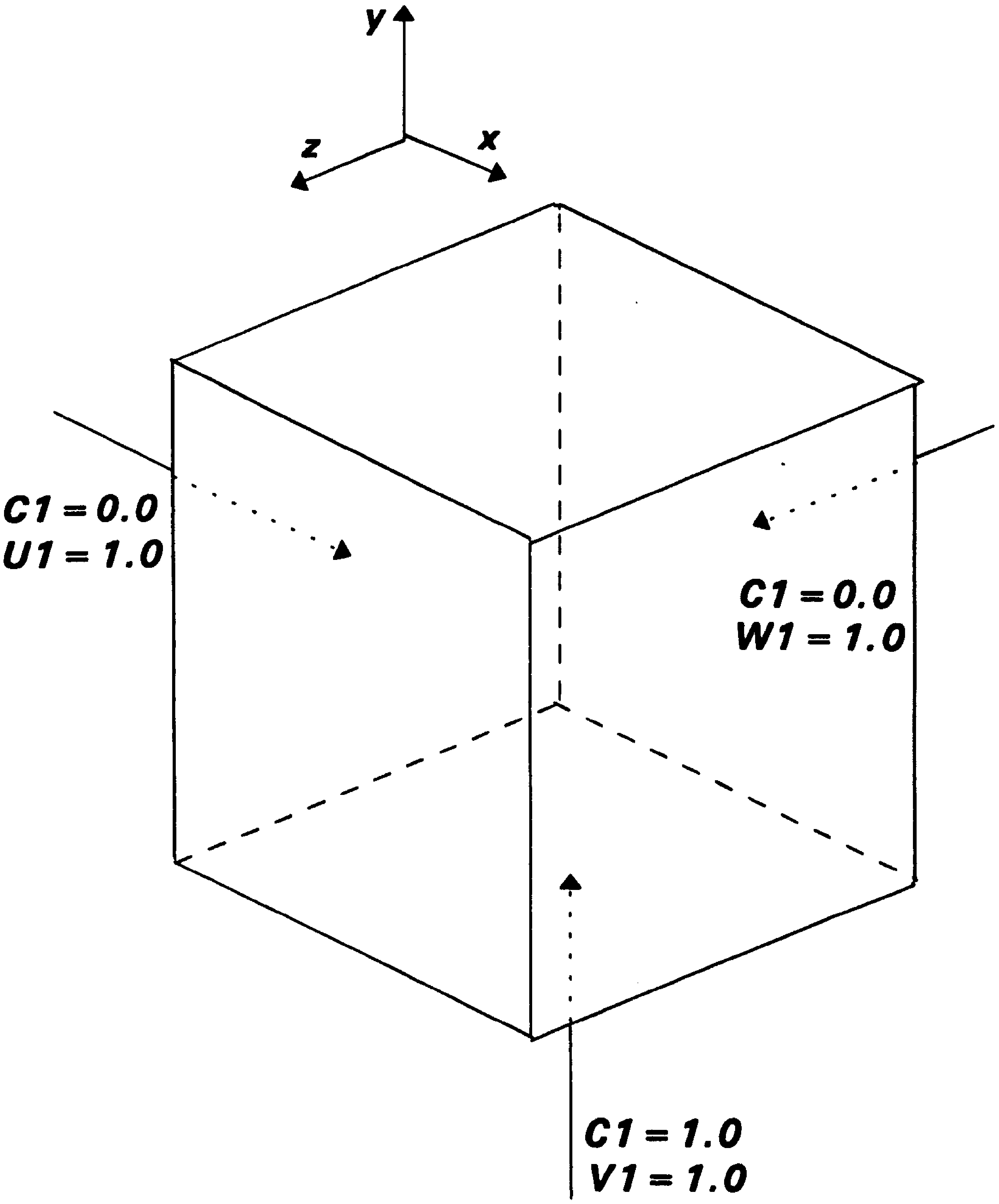


Figure 7.2 *Boundary conditions for 3D scalar convection analysis.*

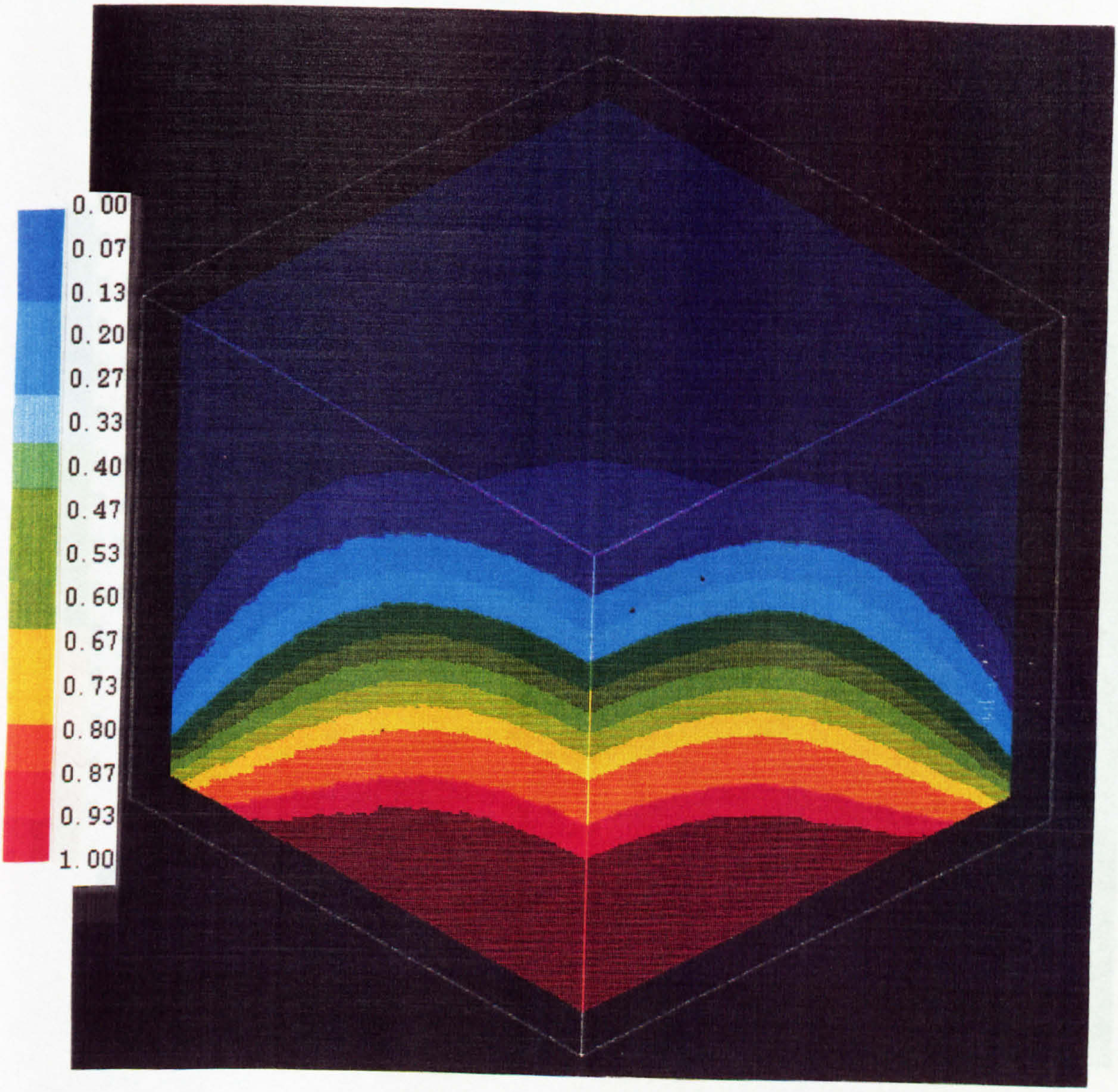


Figure 7.3 *Outflow concentration profile using UPWIND.*

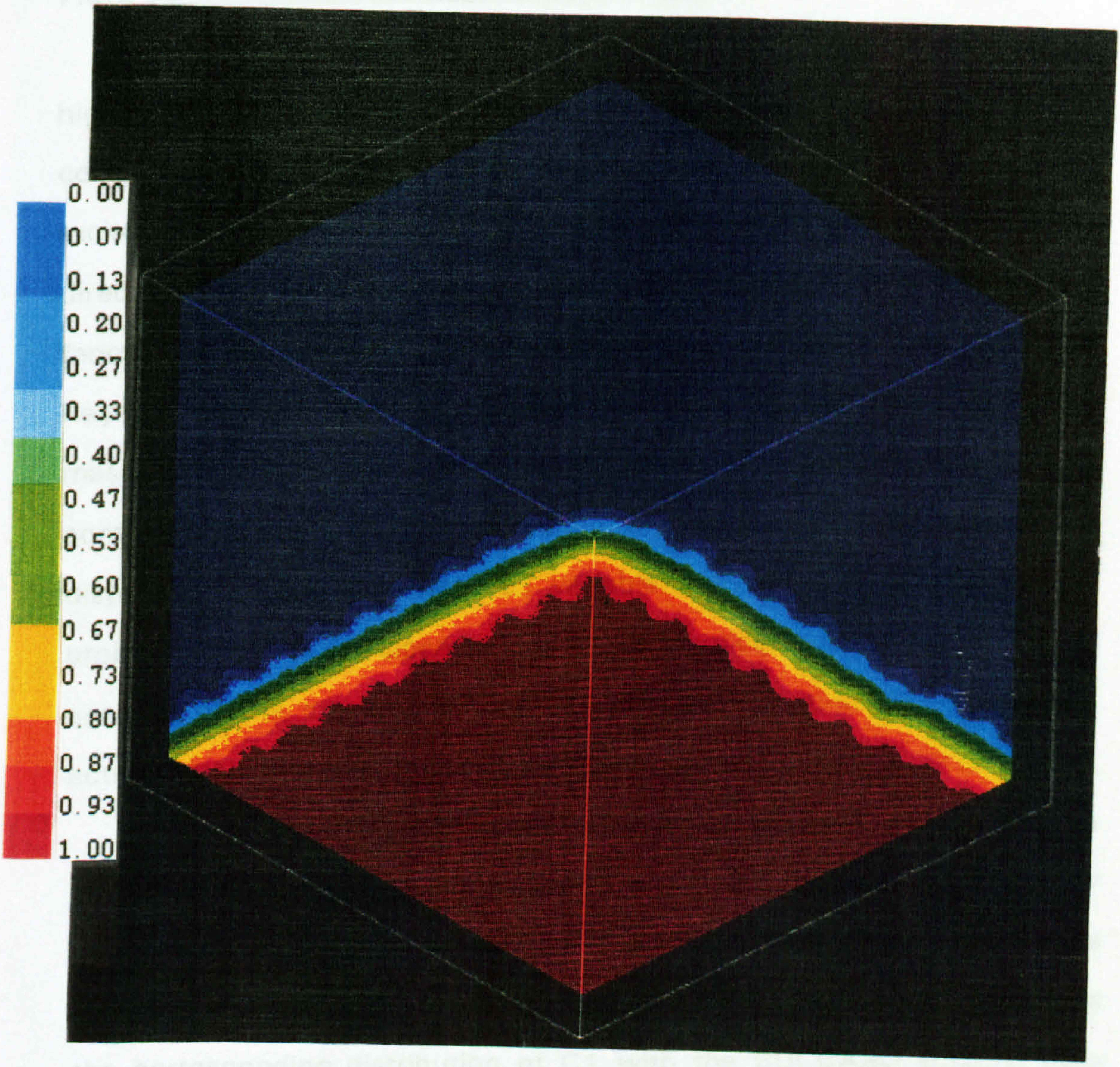


Figure 7.4 ***Outflow concentration profile using SUCCA3D.***

7.3.2 3D Convection of a Scalar Over a Fence

This three-dimensional test problem was undertaken with the aim of highlighting the potential of the SUCCA3D scheme when applied to the convective transport of a scalar variable. The flow itself contains a 3D recirculation region and was thus considered to provide a good test for the directionality book-keeping of the SUCCA3D code. A laminar flow over a fence type blockage was considered as shown in Figure 7.4a, with the Reynolds number based on the inlet height being 350. An injection of mass, momentum and species concentration C1 through two central computational cells located upstream of the fence was considered and the distribution of the introduced C1 variable analyzed. The Q1 file for the problem is given in Appendix K1. In an identical manner to the scalar analysis of the previous section the physical diffusion of C1 was de-activated. No frictional boundary conditions were applied at solid obstacles.

Figure 7.4b shows the distribution of C1 with the upwind scheme employed for the convective transport of C1 and the hybrid scheme for momentum. A right angle section is shown which bisects the fence blockage with half of the fence highlighted in the Figure. Figure 7.4c shows the corresponding distribution of C1 with the SUCCA3D scheme now applied to the C1 convective transport. Additional contour plots showing the C1 dispersion through out the vertical X4 and horizontal Y1 planes are highlighted in Figures 7.4d to 7.4g. These plots are shown for both upwind and SUCCA3D applications.

In comparing the results for both the upwind and SUCCA3D schemes there appears to be an apparent reduction in the degree of smearing in the distribution of C1 when the SUCCA3D scheme is used.

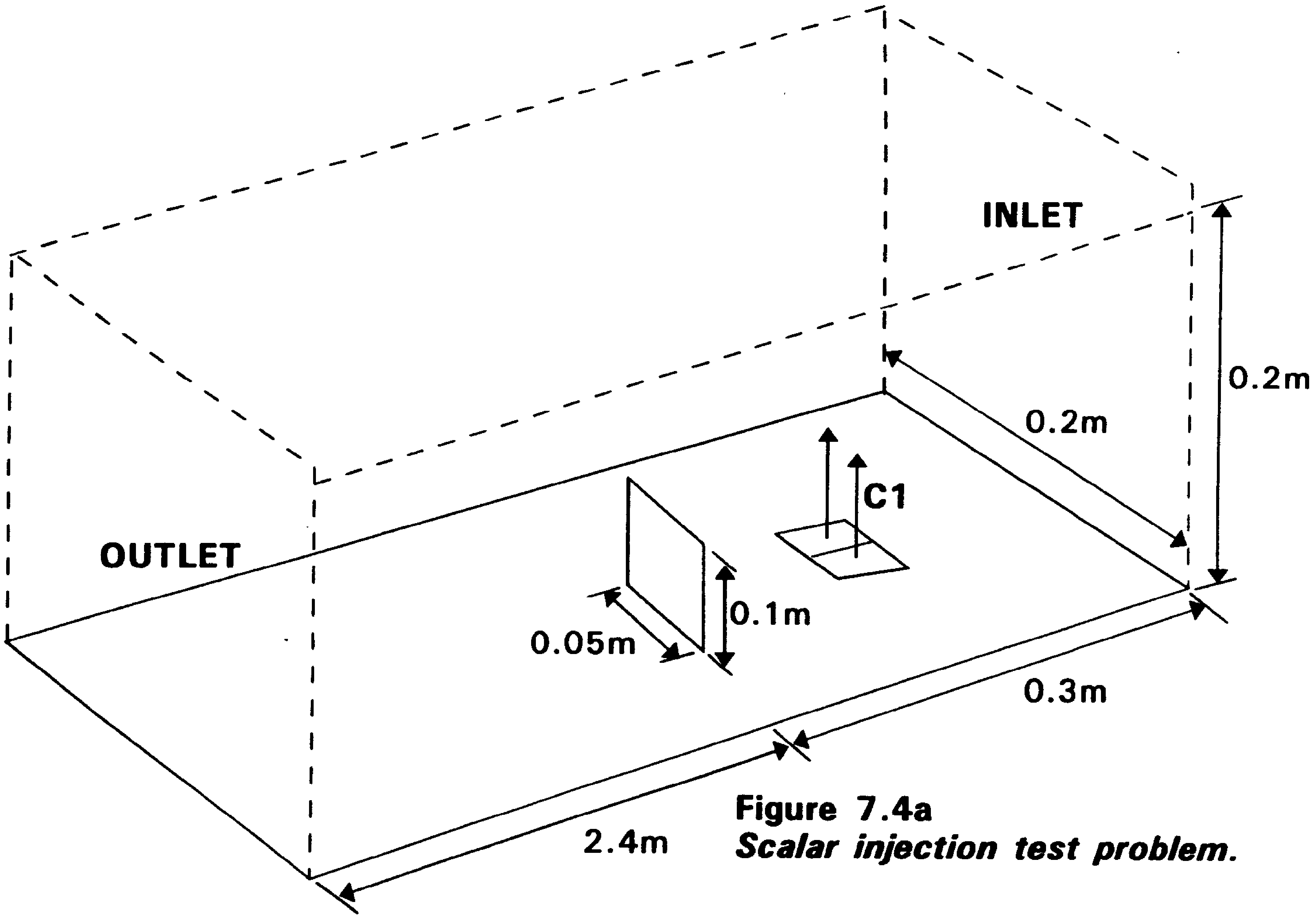


Figure 7.4a
Scalar injection test problem.

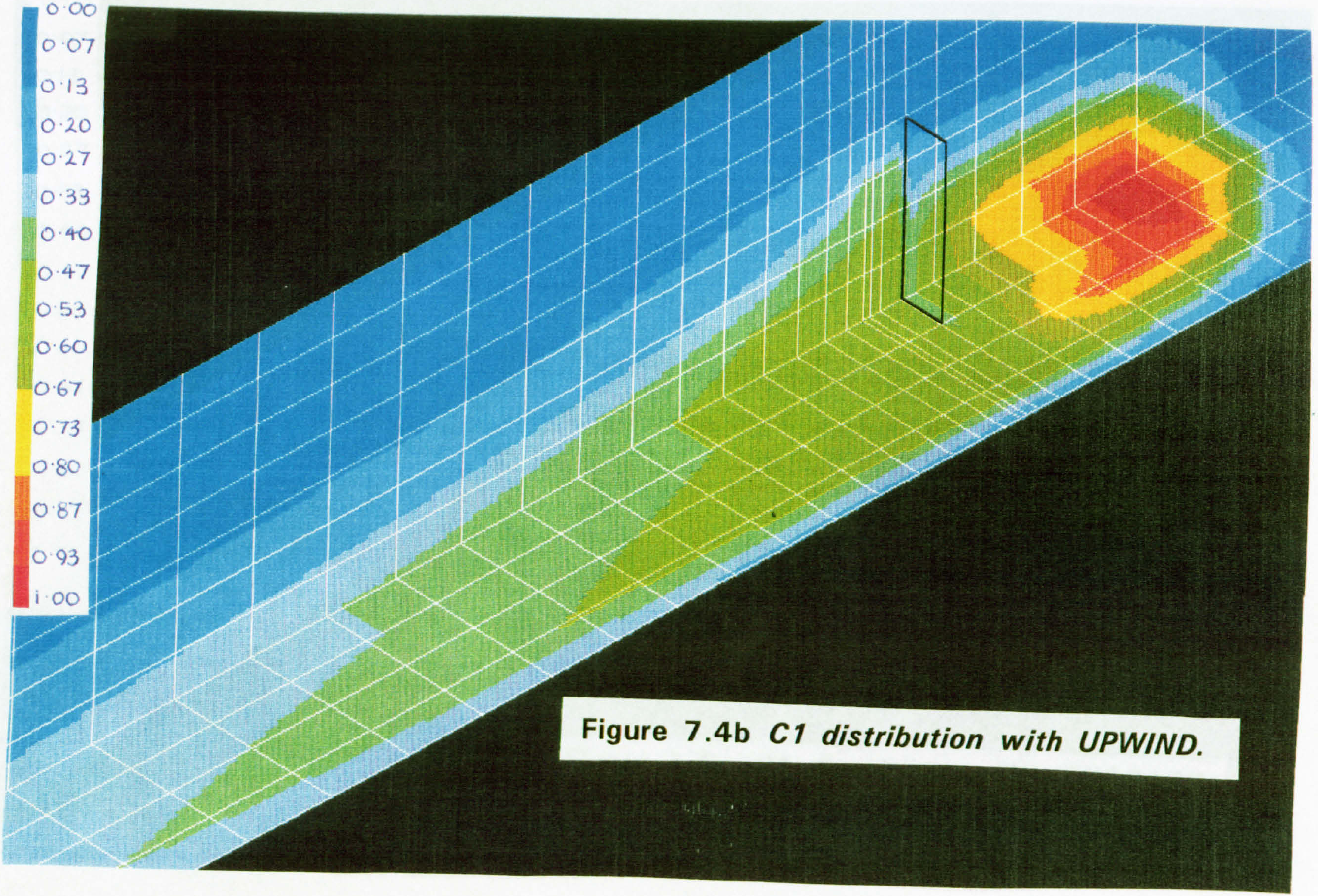


Figure 7.4b *C1 distribution with UPWIND.*

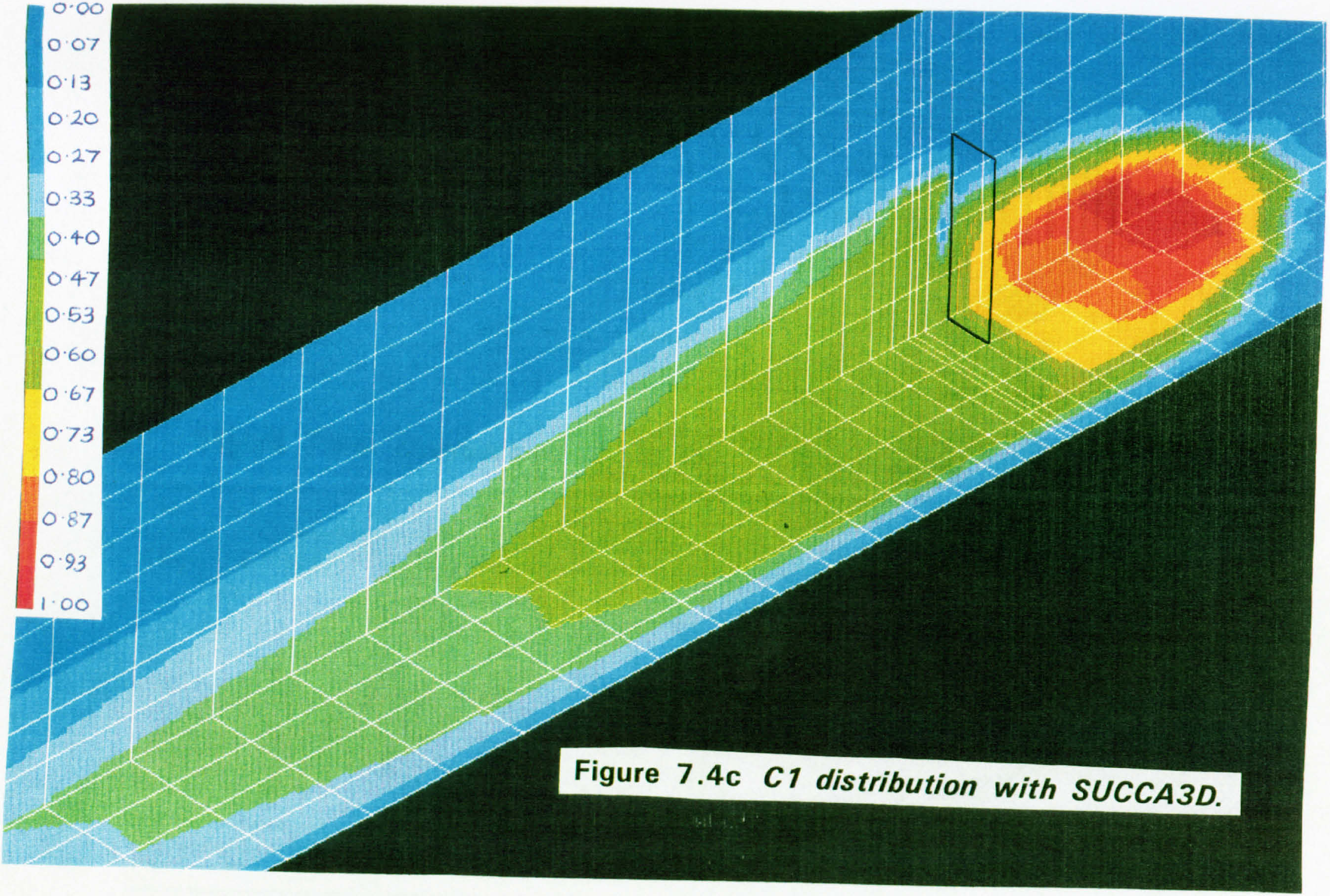


Figure 7.4c *C1 distribution with SUCCA3D.*

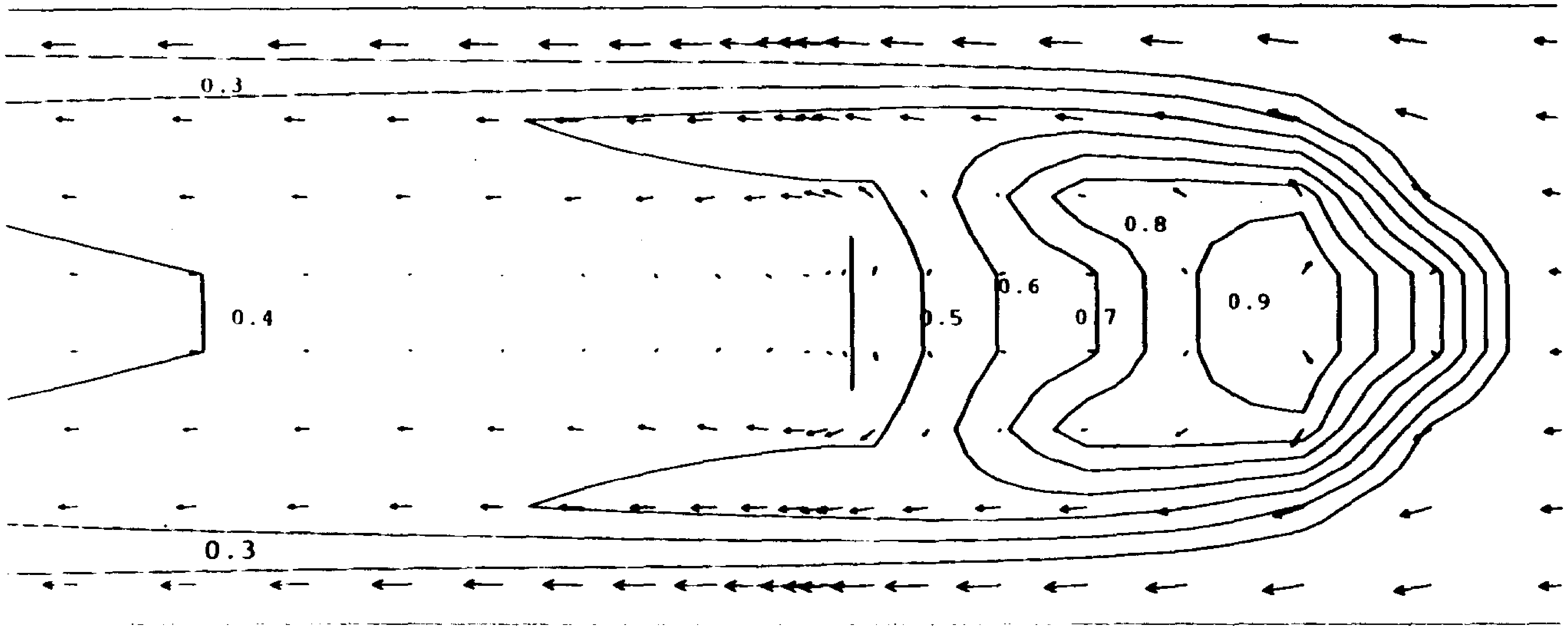


Figure 7.4d *Contours of C1 in the Y1 plane using UPWIND.*

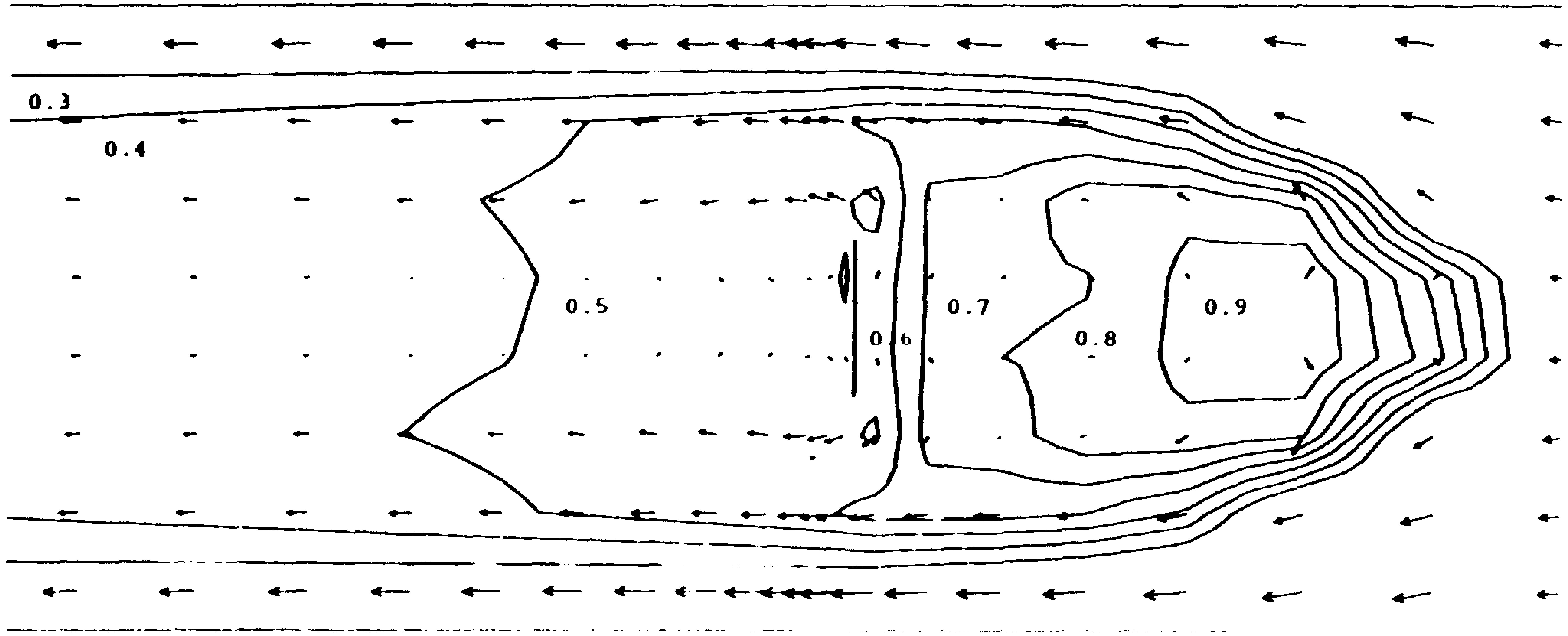


Figure 7.4e *Contours of C1 in the Y1 plane using SUCCA3D.*

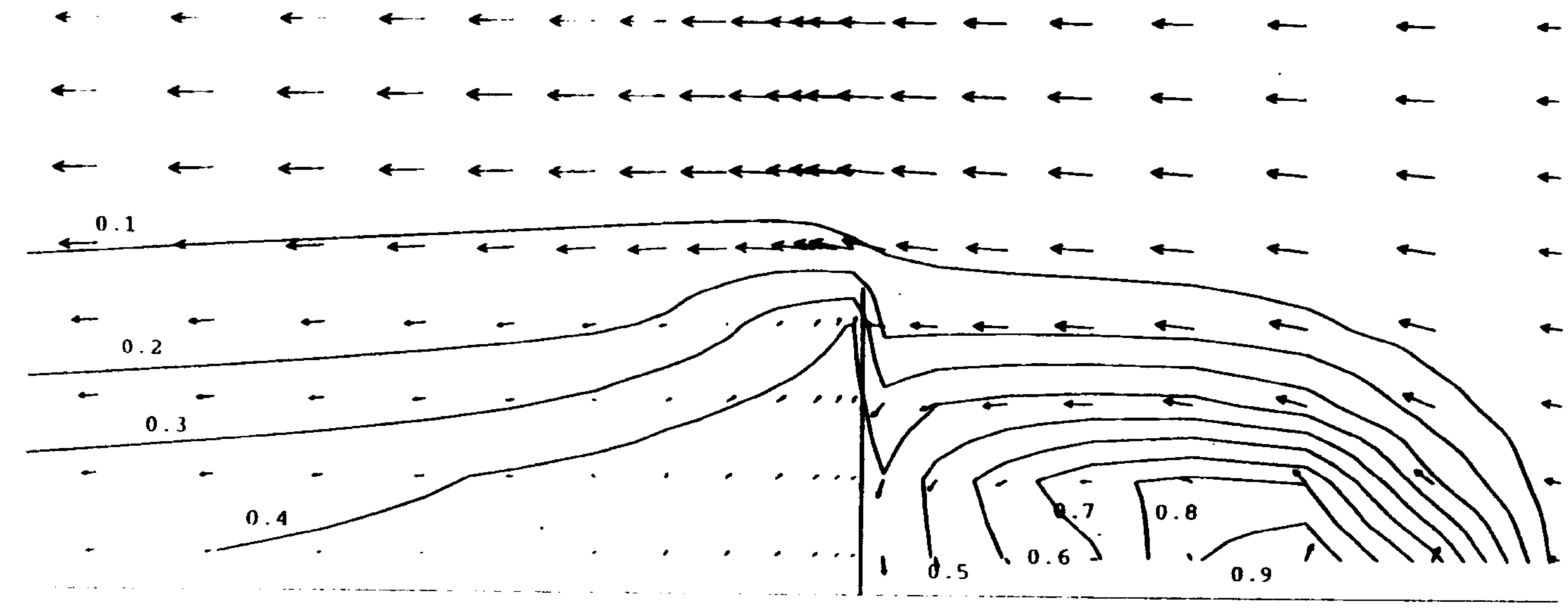


Figure 7.4f *Contours of C1 in the X4 plane using UPWIND.*

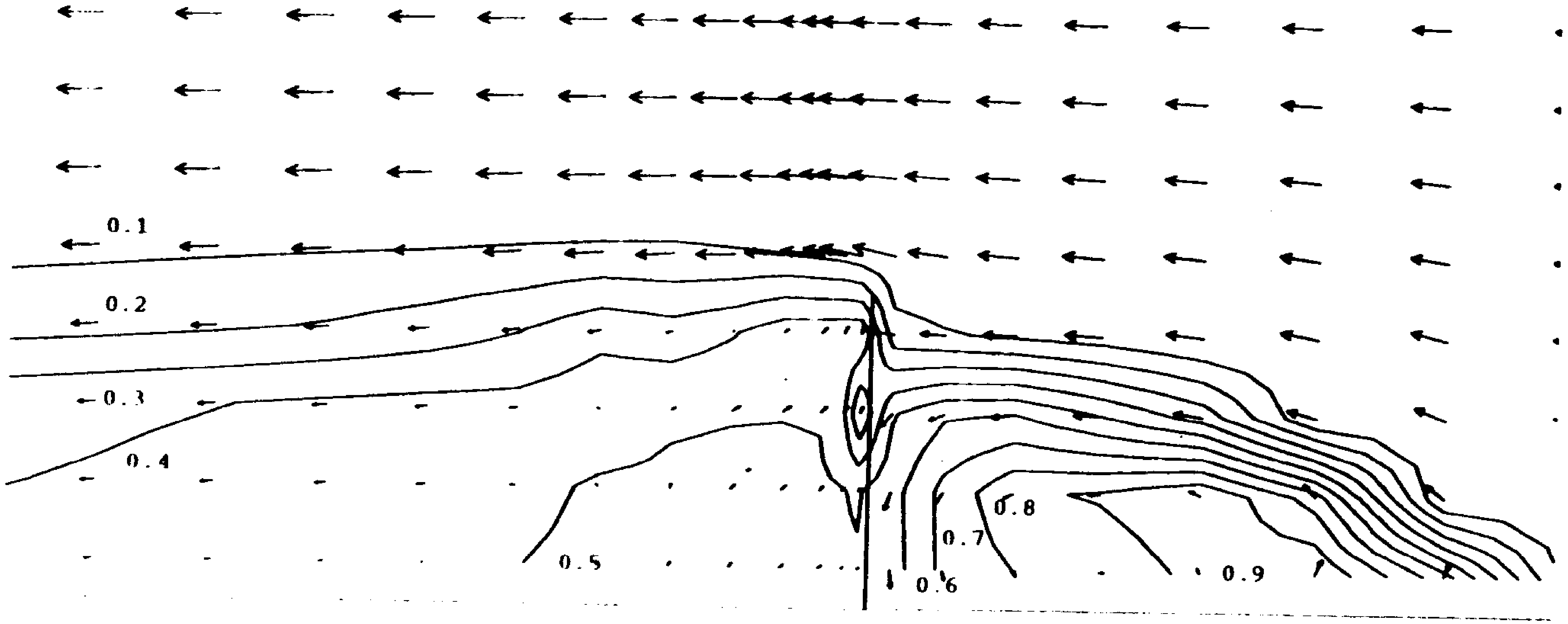


Figure 7.4g *Contours of C1 in the X4 plane using SUCCA3D.*

This qualitative analysis serves to highlight the potential of the SUCCA3D scheme in a three-dimensional scalar transport analysis. However, further testing and some experimental validation will be necessary. An increase in overall CPU expenditure of approximately 20% was encountered when using the SUCCA3D scheme; similar to that experienced with 2D flows.

7.4 The SUCCA3D Scheme Applied to the Transport of Momentum

A similar computational analysis to that executed for the 3D convection of a scalar over a fence was considered. The upstream injection of mass, momentum and C1 was removed and the SUCCA3D scheme was now implemented for the convective transport of momentum. Identical geometry and boundary conditions to that of the scalar analysis were employed while the Reynolds number was decreased to a value of 35. The Q1 file for this analysis is presented in Appendix K2.

Figure 7.4h shows the distribution of axial (W1) velocity with the hybrid scheme employed for the convective transport of momentum in the three space directions. A right angle section again is shown which bisects the fence blockage. Figure 7.4i shows the corresponding W1 distribution using the SUCCA3D scheme applied to U1,V1 and W1 transport. Additional contour plots showing the W1 distribution through out the vertical X4 and horizontal Y1 planes are highlighted in Figures 7.4j to 7.4m. These plots are shown for both hybrid and SUCCA3D applications. In comparing the results for both hybrid and SUCCA3D there appears to be a significantly different distribution of axial velocity. Experimental verification will again be required.

For Reynolds numbers greater than 35 it was found that no

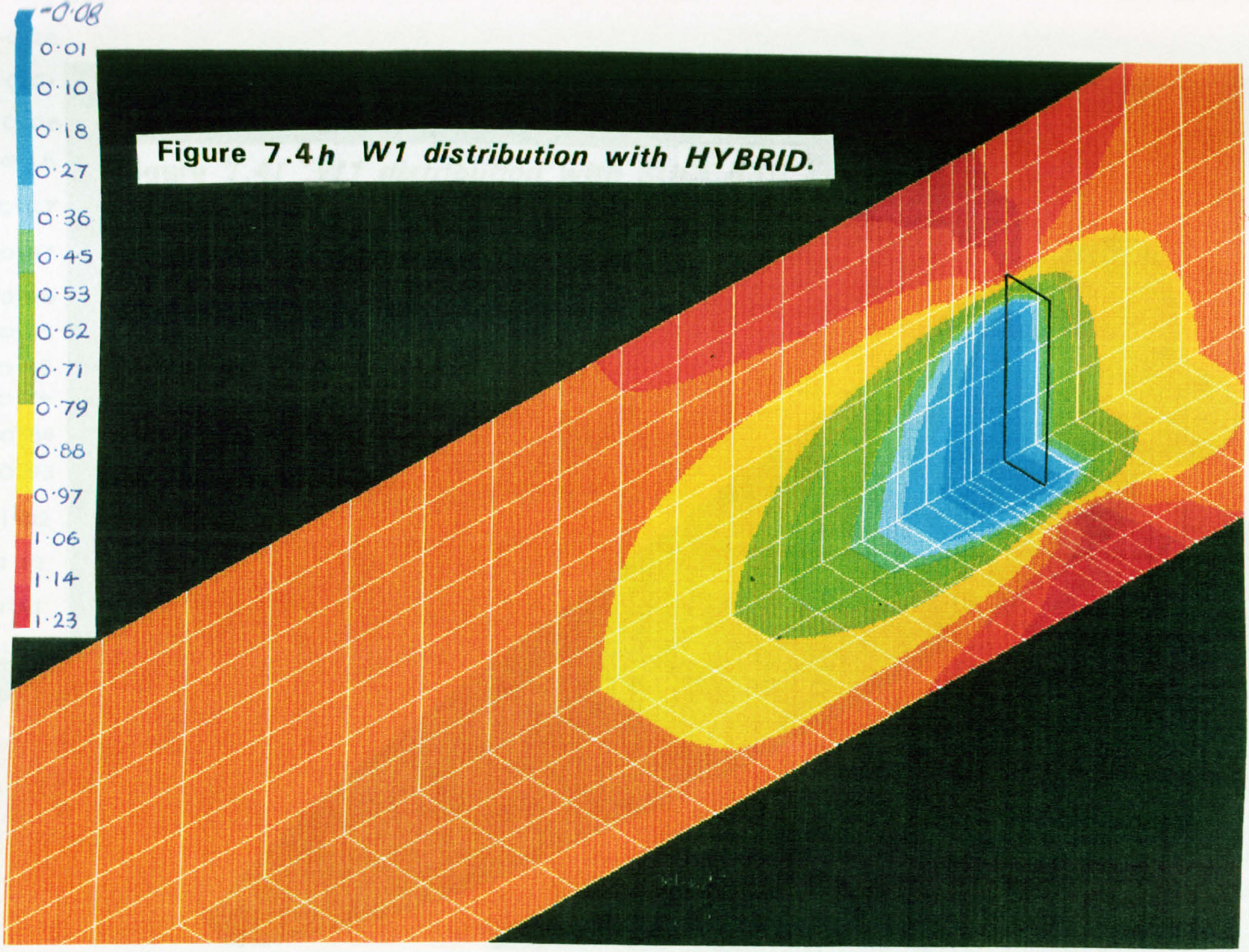


Figure 7.4h W1 distribution with HYBRID.

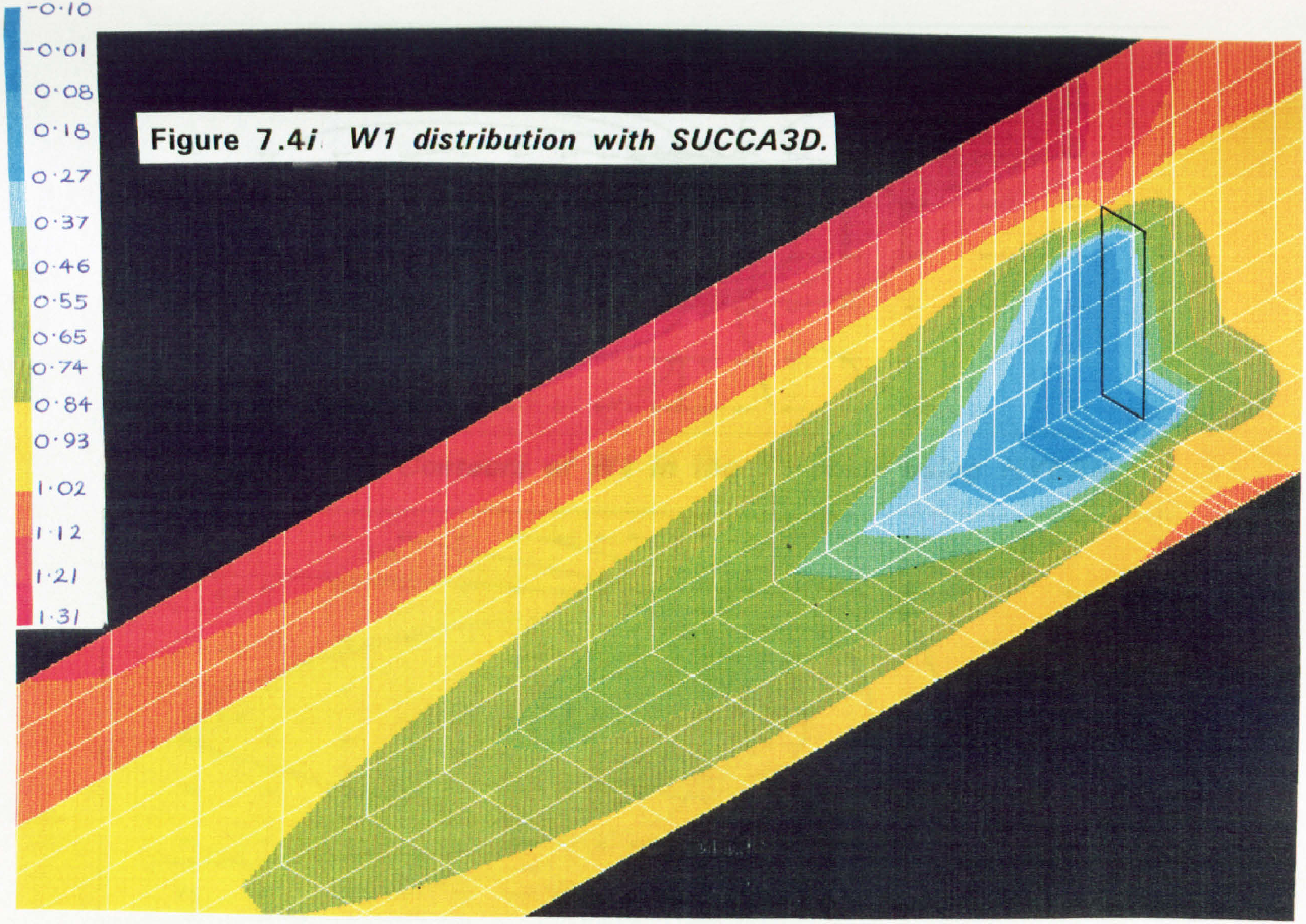


Figure 7.4i W1 distribution with SUCCA3D.

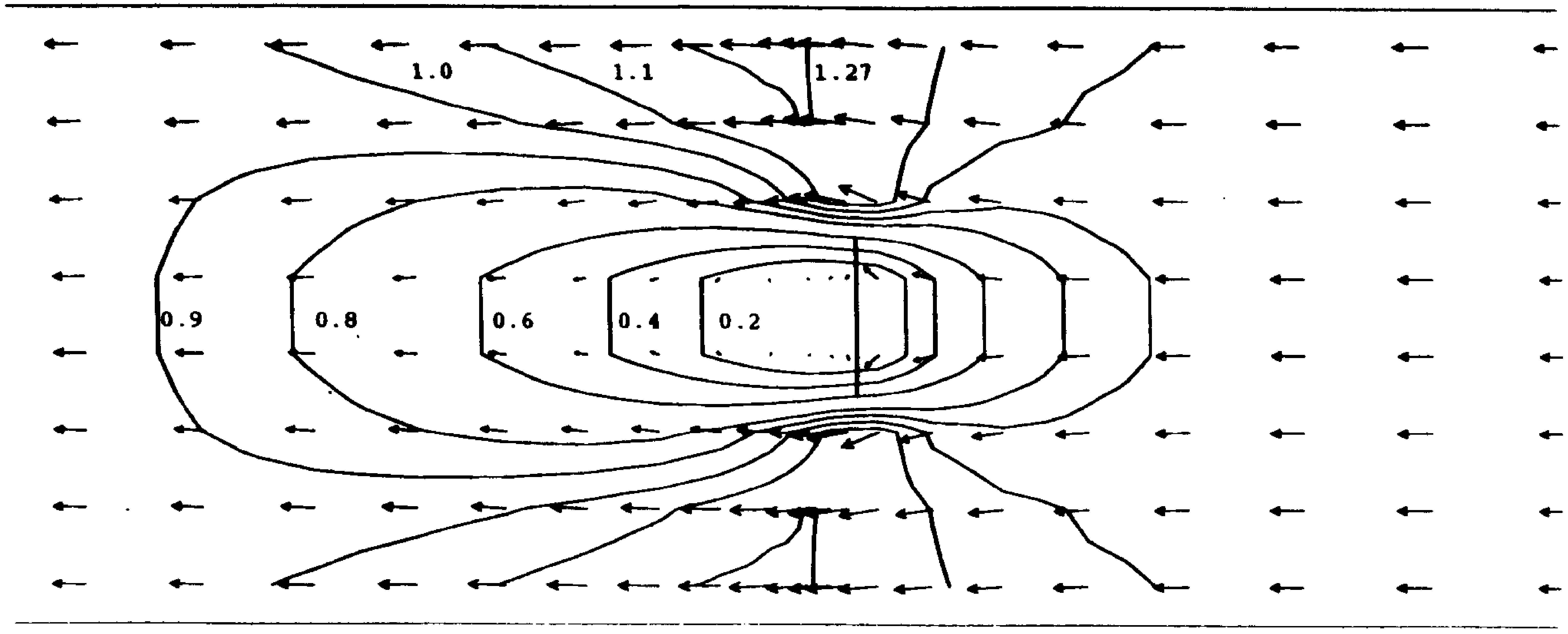


Figure 7.4j *Contours of W1 in the Y1 plane using HYBRID.*

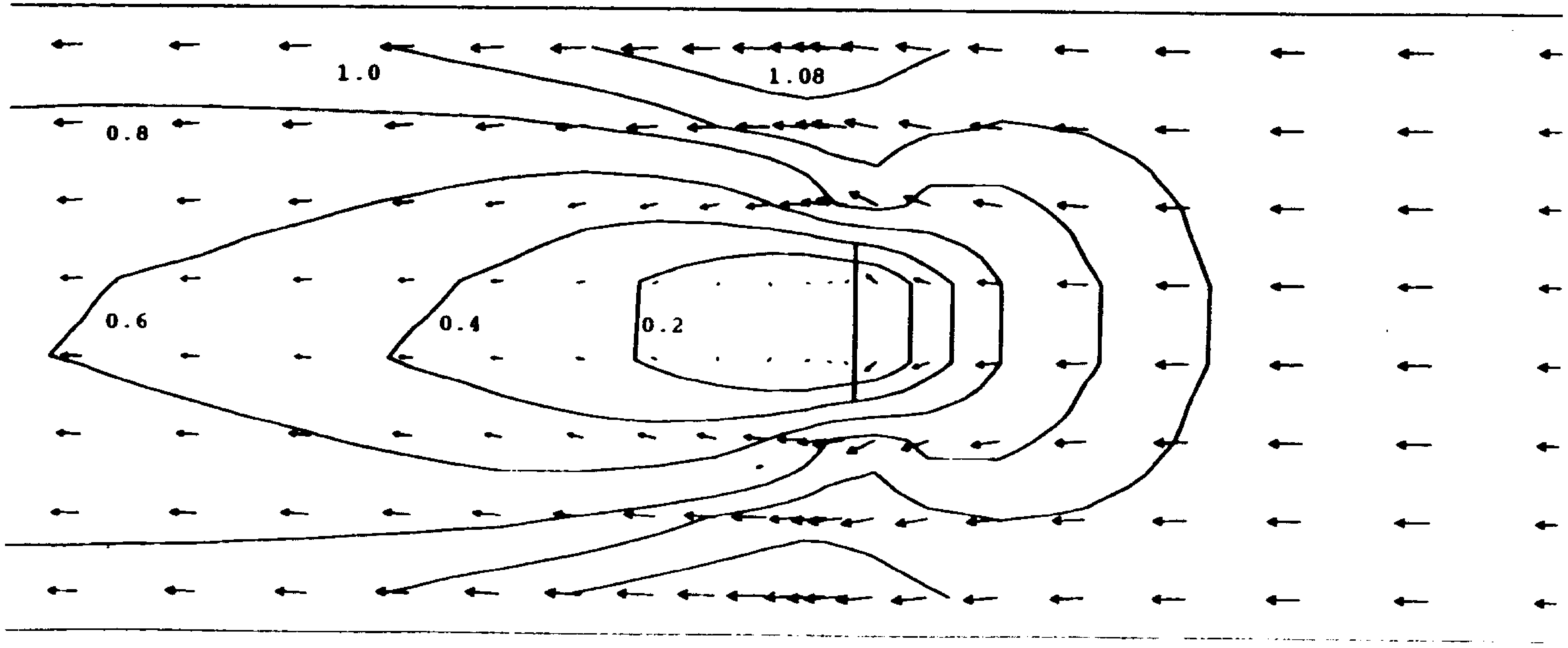


Figure 7.4k *Contours of W1 in the Y1 plane using SUCCA3D.*

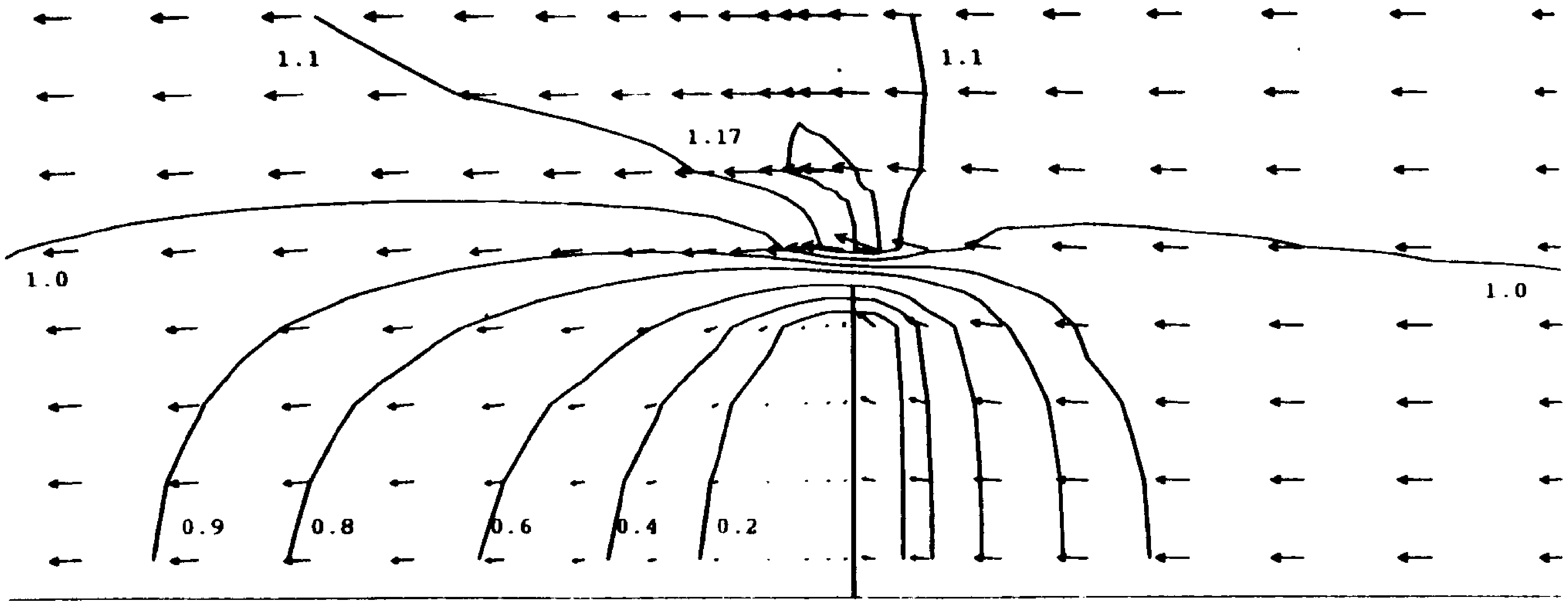


Figure 7.4l Contours of W1 in the X4 plane using HYBRID.

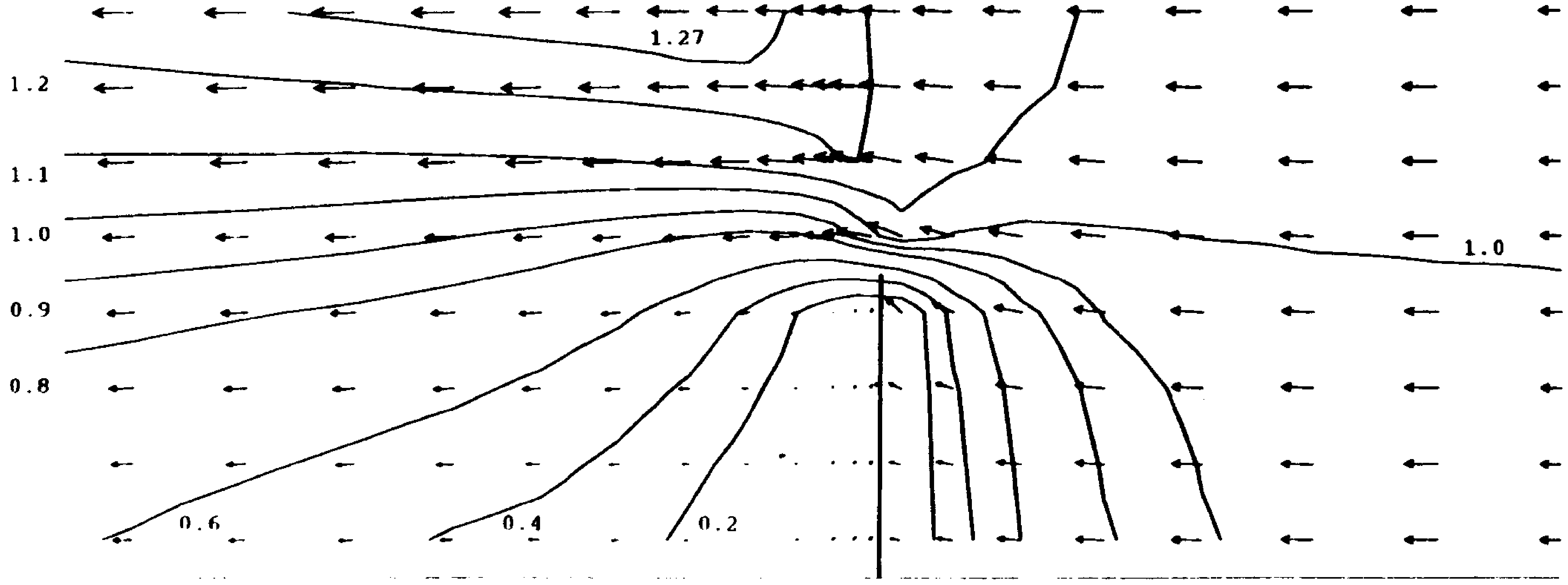


Figure 7.4m Contours of W1 in the X4 plane using SUCCA3D.

converged solution could be obtained. Instability in the convergence path could not be smoothed out using the various existing under-relaxation techniques within PHOENICS and the result was a physically unrealisitic solution. The reasons for the lack of convergence at higher Reynolds numbers remain unclear, however the SUCCA3D scheme does show some promise in the area of momentum transport and further testing may ease this problem.

7.5 Turbulent Vortex Shedding in 3D Flow

The simulation of three-dimensional turbulent vortex shedding was undertaken. The main aims of this analysis were as listed below:

a) to identify whether the three-dimensional false diffusion effects would have the same stabilising influence on the vortex shedding process as that encountered in the 2D analysis.

b) to construct a three-dimensional model that would allow future users to consider the 3D vortex shedding flow in conjunction with further development of the SUCCA3D scheme to account for high Reynolds number flows and alternative coordinate systems.

7.5.1 Geometrical Configuration and Associated Input Parameters

The geometry considered for this analysis was based upon the experimental work of Mottram and Rawat(52) as detailed in Section 5.2 of Chapter 5. In the computational analysis, a three-dimensional polar-cylindrical geometry is considered to account for the pipe flow and the linear bluff body blockage. The modelling of the linear blockage required special consideration. In order to successfully model the linear obstruction partially blocked computational cells at the block edges were used via the POROSITY function within PHOENICS. For each partially blocked cell a POROSITY factor is deduced which is a multiplier for the mass and momentum equations. The calculation of each POROSITY factor necessitated the construction of a special GROUND coding sequence which would identify whether a cell was to be fully blocked, fully open or partially blocked with a subsequent POROSITY calculation. This coding sequence was based upon the mathematics relating the intersection of a straight line (the bluff body) and a circle (the polar-cylindrical grid) and is contained within Group 11 Section 11 of Appendix M. Using this GROUND coding it was thus able to simulate a linear blockage within a non-Cartesian coordinate system. The bluff body obstruction is shown in Figure 7.5 which highlights the partially blocked cells; with all cells radially inward of the partially blocked cells being fully blocked and all others fully open.

All other dimensions were as detailed in Chapter 5 and a non-uniform computational mesh of 64X x 18Y x 57Z was employed. The governing equations for the unsteady incompressible turbulent flow were the conservation of mass and momentum in three polar-cylindrical coordinate directions, as described in Appendix A. The transport equations for the turbulence quantities k and ϵ were also solved. The boundary conditions and convergence criteria were as detailed in Chapter 5 and, following time-step trials on the 2D flow, the time step was increased to 8.0E-5 secs to help counter the significant increase in CPU expenditure (approximately 9 days on a Sun sparc1+ computer) for the 3D flow calculation while

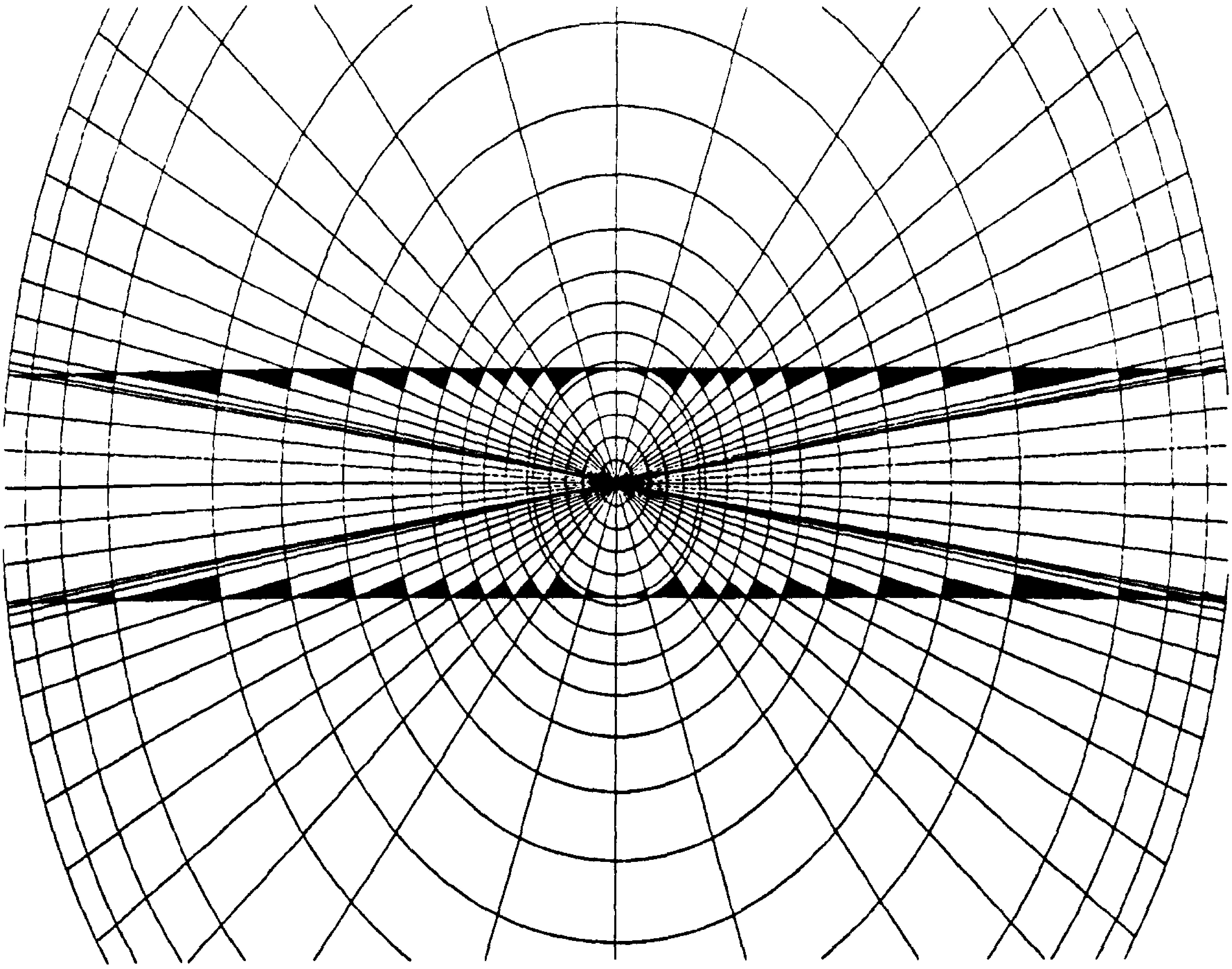


Figure 7.5 ***Simulation of a linear blockage in a non-Cartesian coordinate system.***

maintaining overall solution accuracy. The Reynolds number, based on pipe diameter, was again 130000 and the Strouhal number being as defined in Chapter 5.

7.5.2 3D Flow Analysis

As with both the 2D laminar and turbulent vortex shedding flow analyses an explicit perturbation, in the form of some flow disturbance, is required at the onset of the transient calculation to provoke and maintain flow periodicity in the wake. In the 3D calculation this disturbance took the form of an imposed cross-flow on the upstream and downstream plane faces of the bluff body. The no-slip condition was removed for a series of cells on the bluff body in the circumferential coordinate direction and a radial slip velocity applied whose magnitude was of the order of the incoming mean bulk velocity. These asymmetry inducing conditions are shown in Figure 7.6. A steady flow was then imposed through the pipe and the resulting steady asymmetric flow field used as the initial conditions for a transient solution of the flow past the bluff body with the radial slip conditions removed. This steady asymmetric flow pattern is shown in the velocity vector and axial velocity contour plots of Figures 7.7 and 7.8. The flow pattern is shown for two perpendicular diametrical planes bisecting the bluff body.

A transient calculation was performed for the Reynolds number of 130000 in an attempt to obtain a similar Strouhal number found in the experimental analysis of Mottram and Rawat(52). Primarily, this computation was executed with the PHOENICS default hybrid scheme employed for the convective transport of all flow dependent variables. This resulted in the

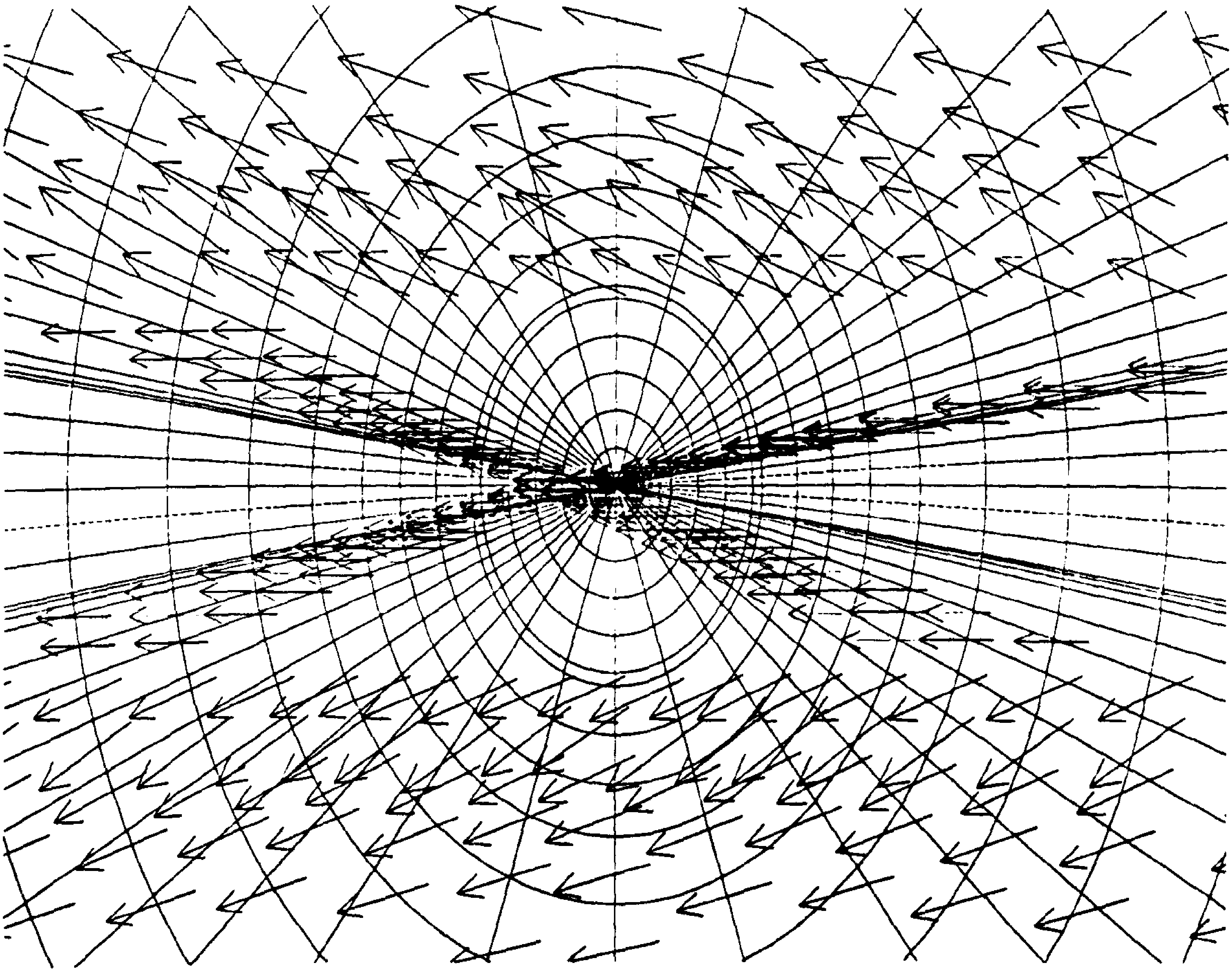


Figure 7.6 *Velocity vectors of cross-flow to promote asymmetry in the flow field.*

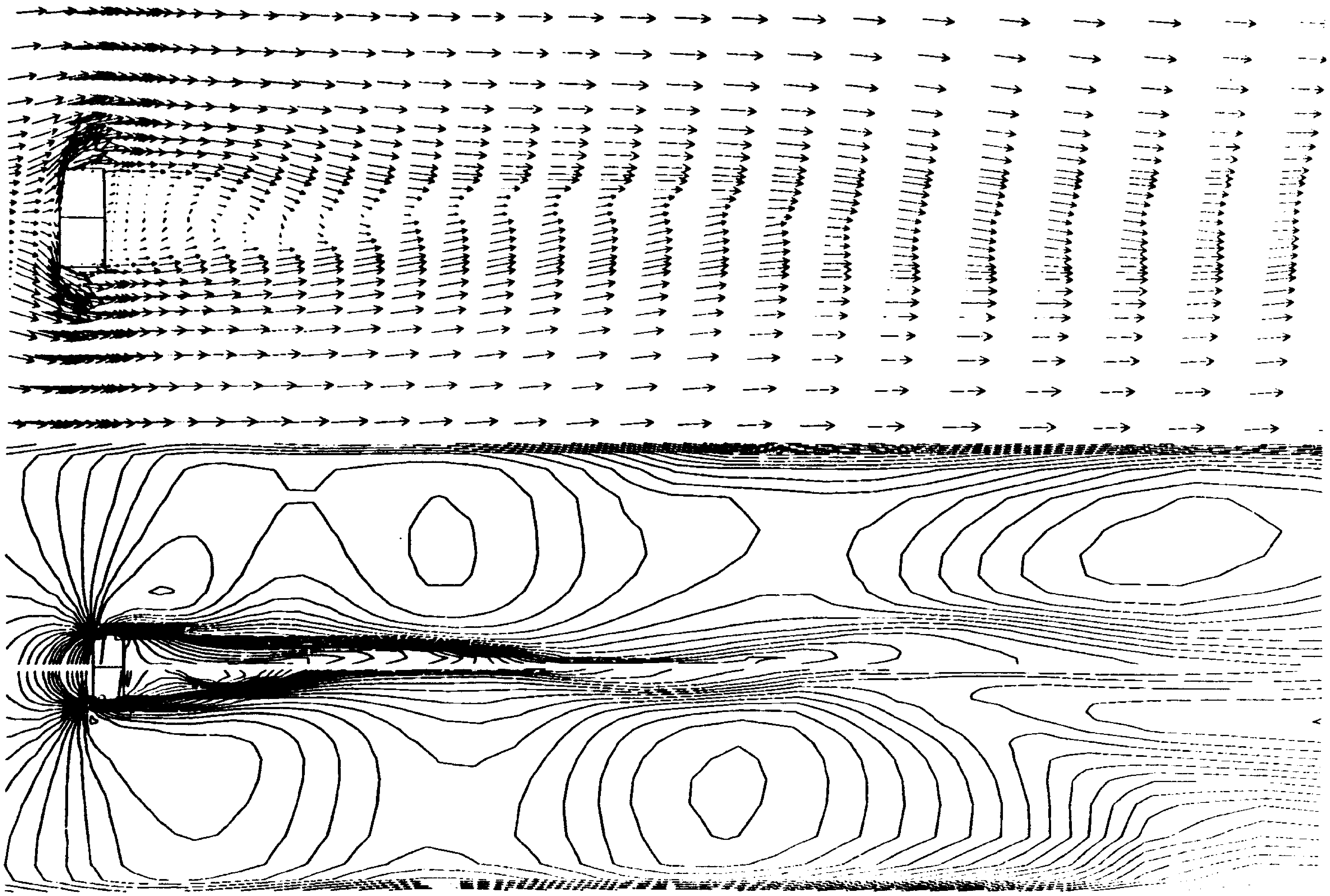


Figure 7.7 *Velocity vectors and axial velocity contours for steady asymmetric flow for a vertical plane bisecting the bluff body.*

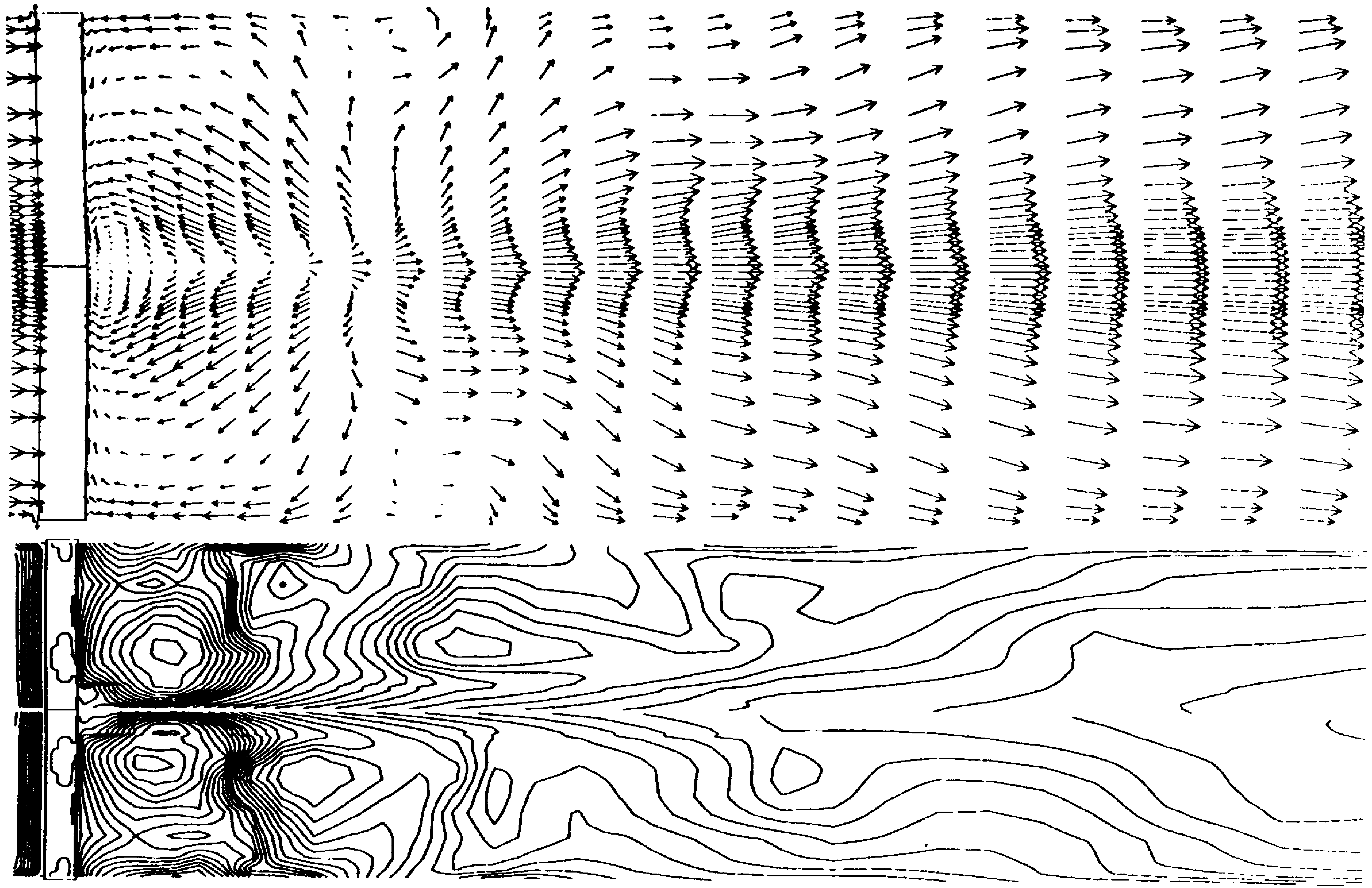


Figure 7.8 *Velocity vectors and axial velocity contours for steady asymmetric flow for a horizontal plane bisecting the bluff body.*

already familiar decay of wake periodicity, encountered in the previous 2D laminar and turbulent studies, due to the excessive false diffusion of momentum associated with the hybrid formulation. The resulting calculation yielded a steady, stable symmetric wake flow pattern in both diametric planes as shown in Figures 7.9 and 7.10. These plots show the velocity vectors and contours of axial (W-component) velocity in the wake immediately downstream of the bluff body. The decay of vortex shedding is represented by transients for the mean velocity components U1, V1 and W1 for a single cell located in the region of vortex shedding, as shown in Figures 7.11 to 7.13. These plots show the transition from the initially disturbed flow pattern to the stable wake structure shown in Figures 7.9 and 7.10.

Within the context of this 3D computation it may be said that the use of the hybrid scheme for the discretisation of the convective transport of momentum will produce a stabilised wake. The excessive false diffusion of momentum in 3D is seen to be equally as dissipative in comparison with the 2D turbulent vortex shedding calculation.

7.6 Implementation of the SUCCA3D Scheme Within PHOENICS

The SUCCA3D scheme should only be applied where there will be a maximum of three positive convective inflows into any computational cell, as is common in a wide range of cases.

In order to activate the SUCCA3D scheme the following PIL commands are introduced into the Q1 file:

```
GROUP 8.   UCONV=T
```

which implies that the user wishes to access and, in this case, modify the

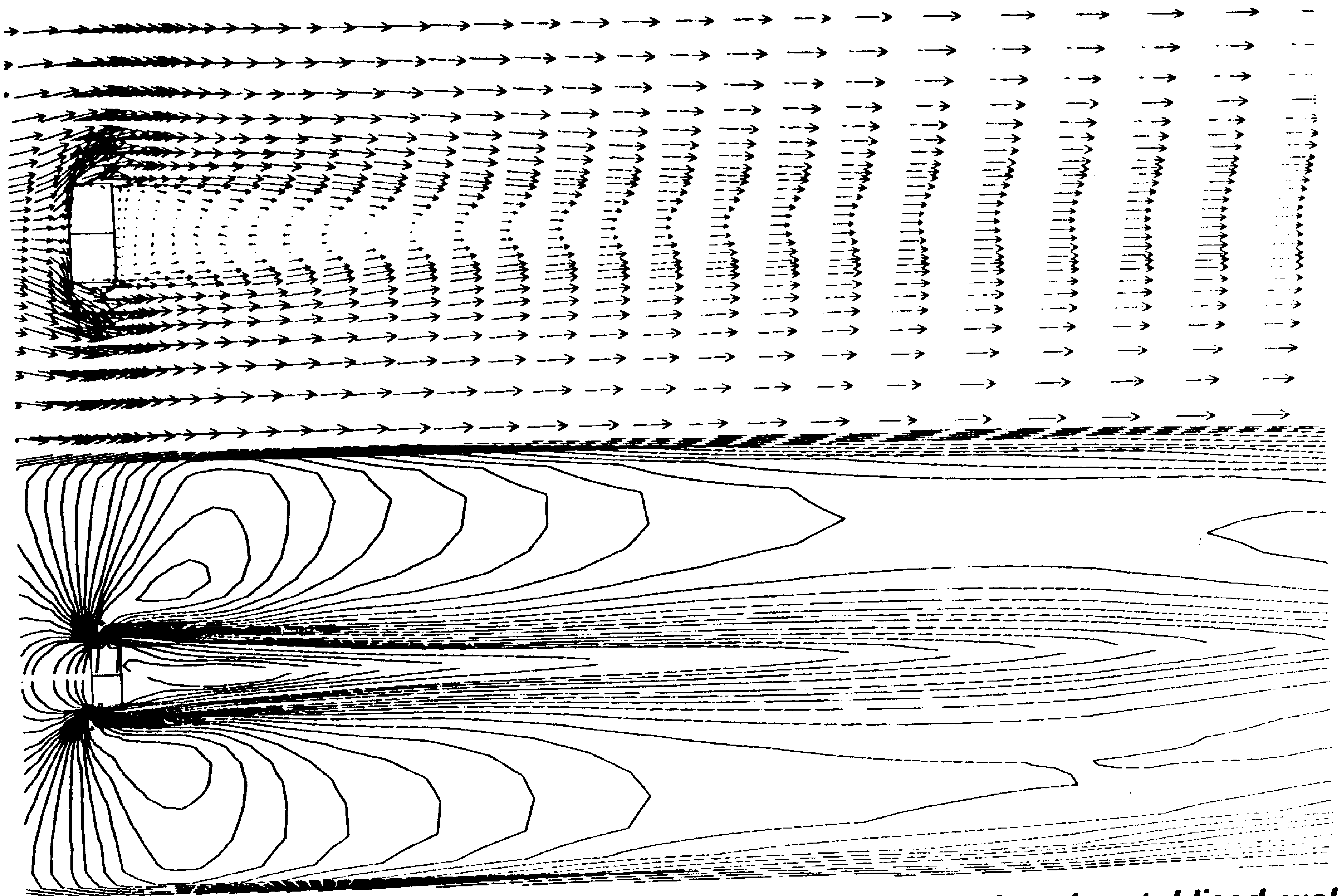


Figure 7.9 *Velocity vectors and axial velocity contours showing the stablised wake flow pattern using HYBRID for a vertical plane bisecting the bluff body.*

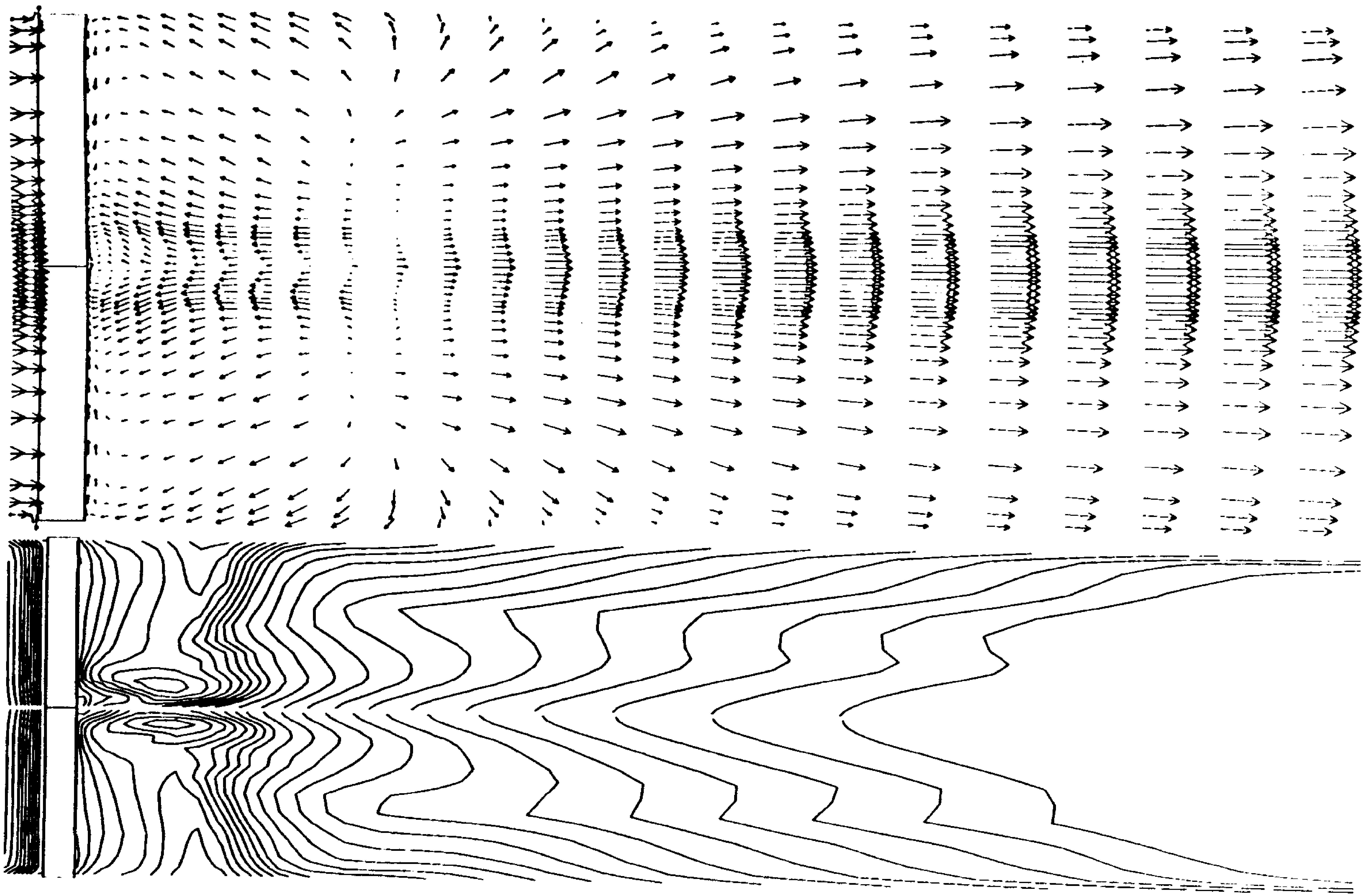


Figure 7.10 *Velocity vectors and axial velocity contours showing the stablised wake flow pattern using HYBRID for a horizontal plane bisecting the bluff body.*

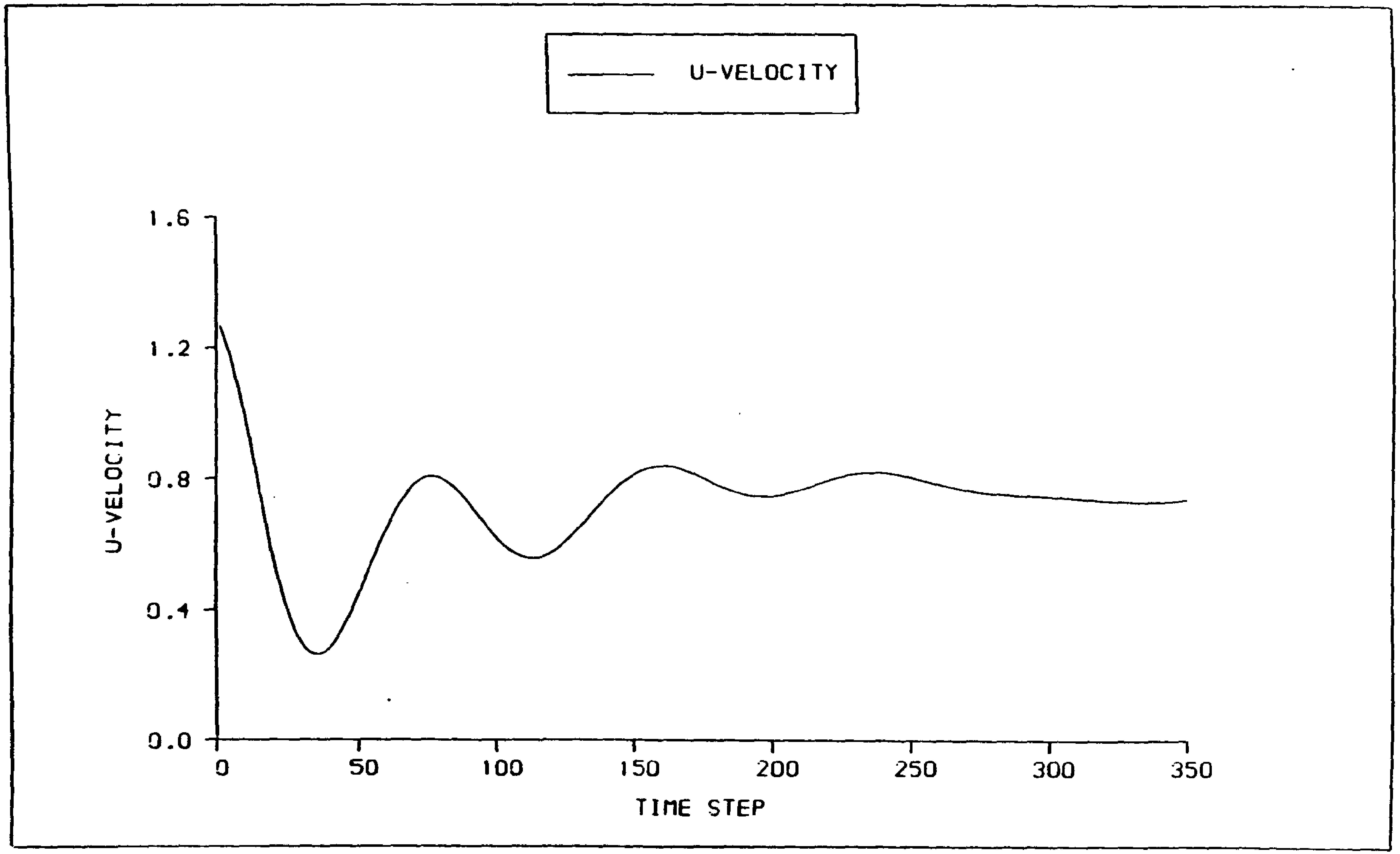


Figure 7.11 *Circumferential U-velocity transient using HYBRID for a single cell located in the region of vortex shedding.*

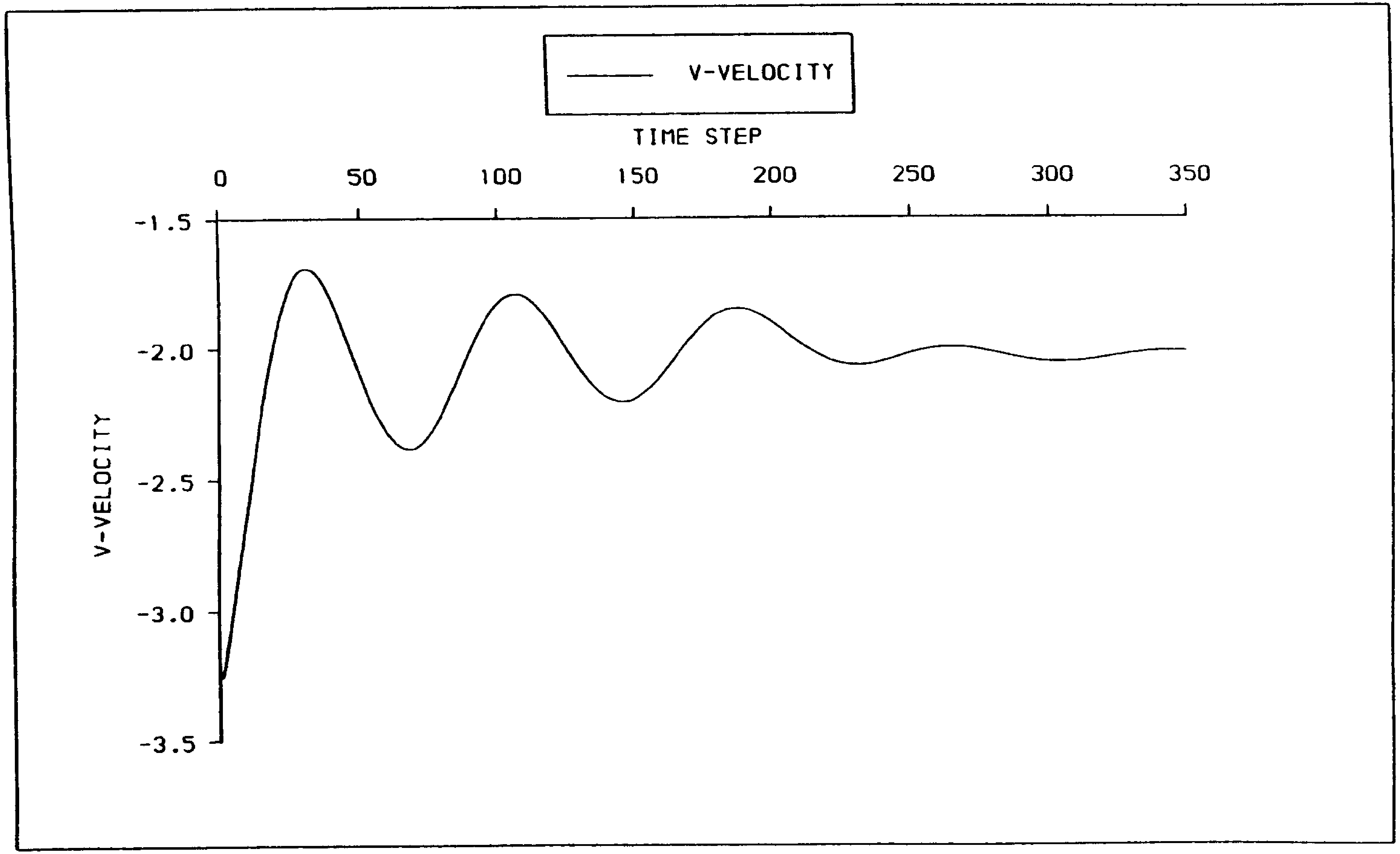


Figure 7.12 ***Radial V-velocity transient using HYBRID for a single cell located in the region of vortex shedding.***

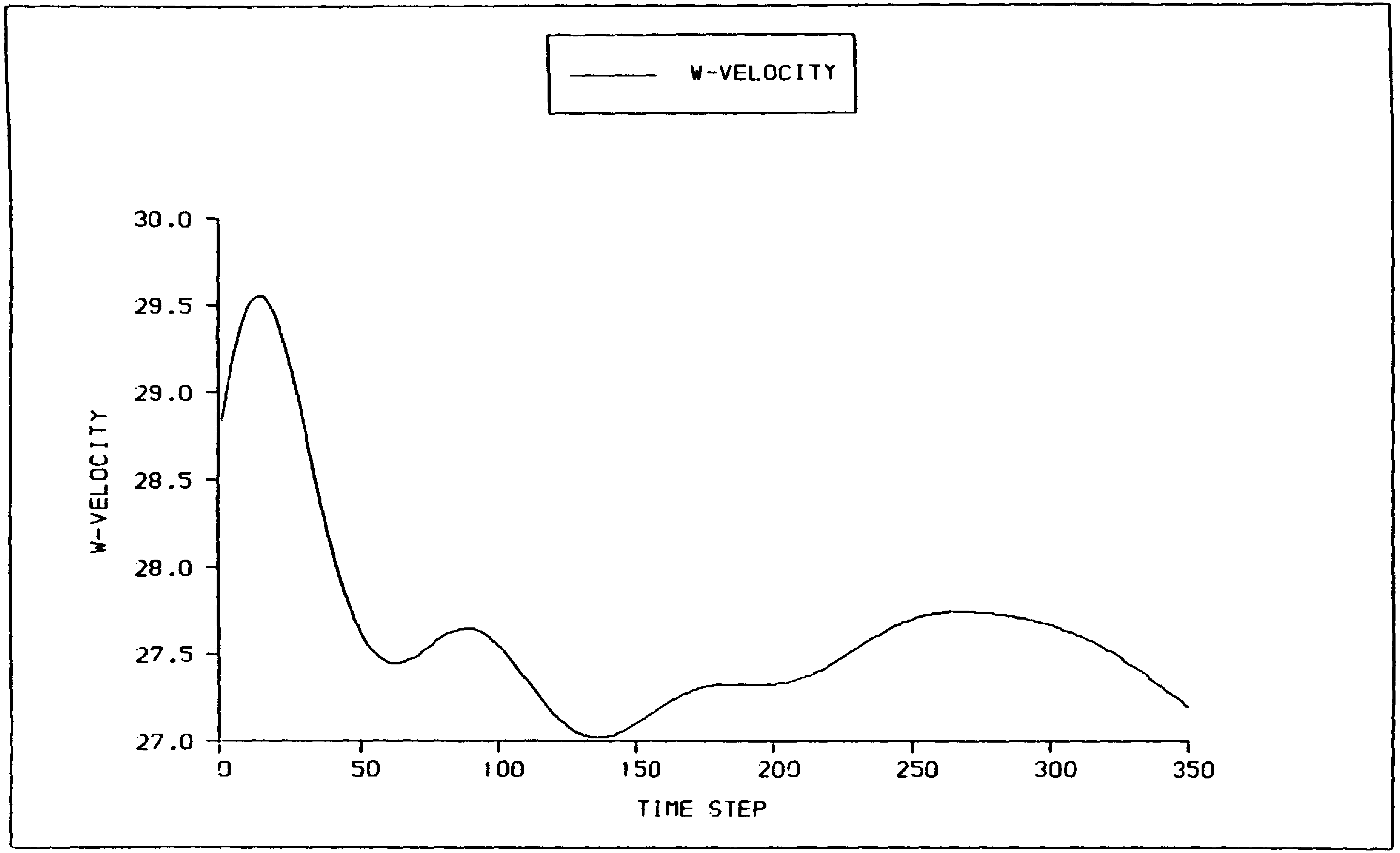


Figure 7.13 ***Axial W-velocity transient using HYBRID for a single cell located in the region of vortex shedding.***

convection coefficients in Group 8, Section 8 of GROUND.

In GROUP 13 when solving for **scalar variables**:

```
PATCH(SUCCAC,CELL,1,NX,1,NY,1,NZ,1,LSTEP)
PATCH(SUCCAE,CELL,1,NX,1,NY,1,NZ,1,LSTEP)
```

which is the normal method of implementing the scheme for scalar variables. It should be noted, however, that in the scalar transport example of Section 7.3.1 the SUCCA3D scheme was applied over the internal and outflow boundary cells i.e. (2,NX,2,NY,2,NZ,1,LSTEP). This was necessary due to the fact that at inflow boundaries there are no upwind corner cells to be included within the scheme. Such a condition will only occur where the inlet boundary flow is skewed (as in this case). Where the inlet boundary flow is non-skewed, as in the example of 3D flow over a fence, the limits of the SUCCA3D application for scalar transport should be the entire computational domain.

When solving for **momentum variables**:

```
PATCH(SUCCAC,CELL,1,NX,1,NY,2,NZ,1,LSTEP)
PATCH(SUCCAE,CELL,1,NX,1,NY,2,NZ,1,LSTEP)
```

The initial Z-slab is excluded from the SUCCA3D application in order to account for the staggered momentum cells in this direction.

To implement the scheme for scalar transport e.g. -

```
COVAL(SUCCAC,C1,GRND,GRND)
COVAL(SUCCAE,C1,GRND1,GRND1)
```

which describes the convective contribution from both corner (SUCCAC) and edge (SUCCAE) cells of the solution molecule shown in Figure 7.1.

To implement the scheme for the transport of momentum e.g. -

```
COVAL(SUCCAC,W1,GRND,GRND)
COVAL(SUCCAE,W1,GRND1,GRND1)
```

which again describes both corner and edge cell contributions.

Finally, in GROUP 19 the following LOGICAL statements are communicated by SATELLITE to GROUND:

To solve for U1 via SUCCA3D:

```
LG(1)=T
```

To solve for V1 via SUCCA3D:

```
LG(2)=T
```

To solve for W1 via SUCCA3D:

```
LG(3)=T
```

To solve for scalars via SUCCA3D:

```
LG(4)=T
```

The above LOGICAL statements must be de-activated (LG(n)=F) if the scheme is not being used for that particular transported variable.

It should be noted that the scheme can be applied to any or all of these dependent variables. Also the COefficient and VALue arguments have been formulated within the GRND and GRND1 sections of GROUND, however there is no reason why the coding cannot be placed within any other GRND section.

The compilation and linking procedures for the SUCCA3D coding are contained within APPENDIX K3.

CHAPTER 8 CONCLUSIONS AND RECOMMENDATIONS FOR FUTURE WORK

The commercially available PHOENICS CFD software package has been used for a transient analysis of vortex shedding flow in both laminar and turbulent regimes.

Two different numerical schemes were employed for the representation of convective transport and, in order to promote instability within the flow, an explicit perturbation was provided at the onset of the transient calculation.

In the first analysis, using the hybrid-upwind convection scheme, premature vortex decay occurred for both 2D and 3D flow analyses; resulting in a steady symmetric flow pattern. The source of this decay was identified as the excessive false diffusion associated with the hybrid-upwind scheme.

Two alternative modified first-order schemes, SUCCA2D and SUCCA3D, were formulated and implemented in an attempt to take into account local flow direction, thus directly reducing the errors associated with multidimensional false diffusion. Both schemes were tested on various discriminating flow examples.

For the 2D case this reduction in the degree of false diffusion helped promote continuous vortex shedding with a frequency in reasonable agreement with experimental analysis. The reduction of false diffusion of a scalar quantity in a 2D benchmark test case was also successfully computed. In the same scalar test the SUCCA scheme was shown to improve the numerical accuracy of the calculation in comparison with the similar flow oriented scheme CUPID.

For 3D flow problems it was observed that a converged solution was obtainable only at low Reynolds numbers when applying the SUCCA3D scheme to the transport of momentum. For a low Reynolds number flow over a fence type blockage the SUCCA3D scheme showed some promise in the reduction of false diffusion of momentum. Further testing of the SUCCA3D code in relation to the convective transport of momentum at higher Reynolds numbers is required. However, the implementation of the scheme for the 3D transport of a scalar quantity did remain stable at higher Reynolds numbers and the converged solution highlighted an apparent reduction in the degree of 3D false diffusion of the transported scalar in comparison with the upwind scheme.

With the SUCCA2D scheme in place, the mechanism of pulsating flow influence on the turbulent vortex shedding process has been simulated, although with limited success in comparison with experimental data.

A modest increase in CPU expenditure of approximately 20 per cent was encountered for both SUCCA2D and SUCCA3D schemes.

Both schemes can easily be included as part of the overall solution process and the convective transport of both momentum and scalar variables has been incorporated.

The results indicate that a complex transient phenomenon such as vortex shedding can be analyzed in two-dimensions using a commercially available CFD software tool such as PHOENICS. However, the necessity for user-defined code modification was highlighted during this process. The results of this code modification, by implementing a simple flow-oriented numerical algorithm, SUCCA, in two and three dimensions, indicates that multidimensional false diffusion errors may be substantially reduced in convection-dominated fixed-grid CFD flow analyses.

Recommendations for further work include the following:

a) testing of both SUCCA2D and SUCCA3D schemes on numerical problems where the incidence of multidimensional false diffusion is known or suspected to occur.

b) further development of the SUCCA3D coding in relation to the transport of momentum such that a stable, converged solution may be found at higher Reynolds numbers.

c) formulation of the SUCCA2D and SUCCA3D schemes for general application in different coordinate systems other than Cartesian, e.g. body-fitted and polar-cylindrical coordinates, such that a fully 3D vortex shedding calculation may be undertaken.

d) studying the influence of alternative bluff body shapes on the vortex shedding process which may involve the use of a body fitted coordinate grid system in conjunction with SUCCA.

e) investigate the influence spatial grid refinement on global vortex shedding parameters such as the Strouhal number.

f) investigating the influence of alternative turbulence models, e.g. a Reynolds stress model, on the vortex shedding process.

APPENDIX A

DERIVATION OF THE CONSERVATION LAWS DESCRIBING FLUID FLOW

APPENDIX A

DERIVATION OF THE CONSERVATION LAWS DESCRIBING FLUID FLOW

A.1 The Conservation Principle

We wish to determine the rate of change of a quantity, ϕ , within the finite control volume (CV) shown in Figure A.1.1. We approach the mathematics of the problem by considering a Eulerian frame of reference where the CV is fixed in space:

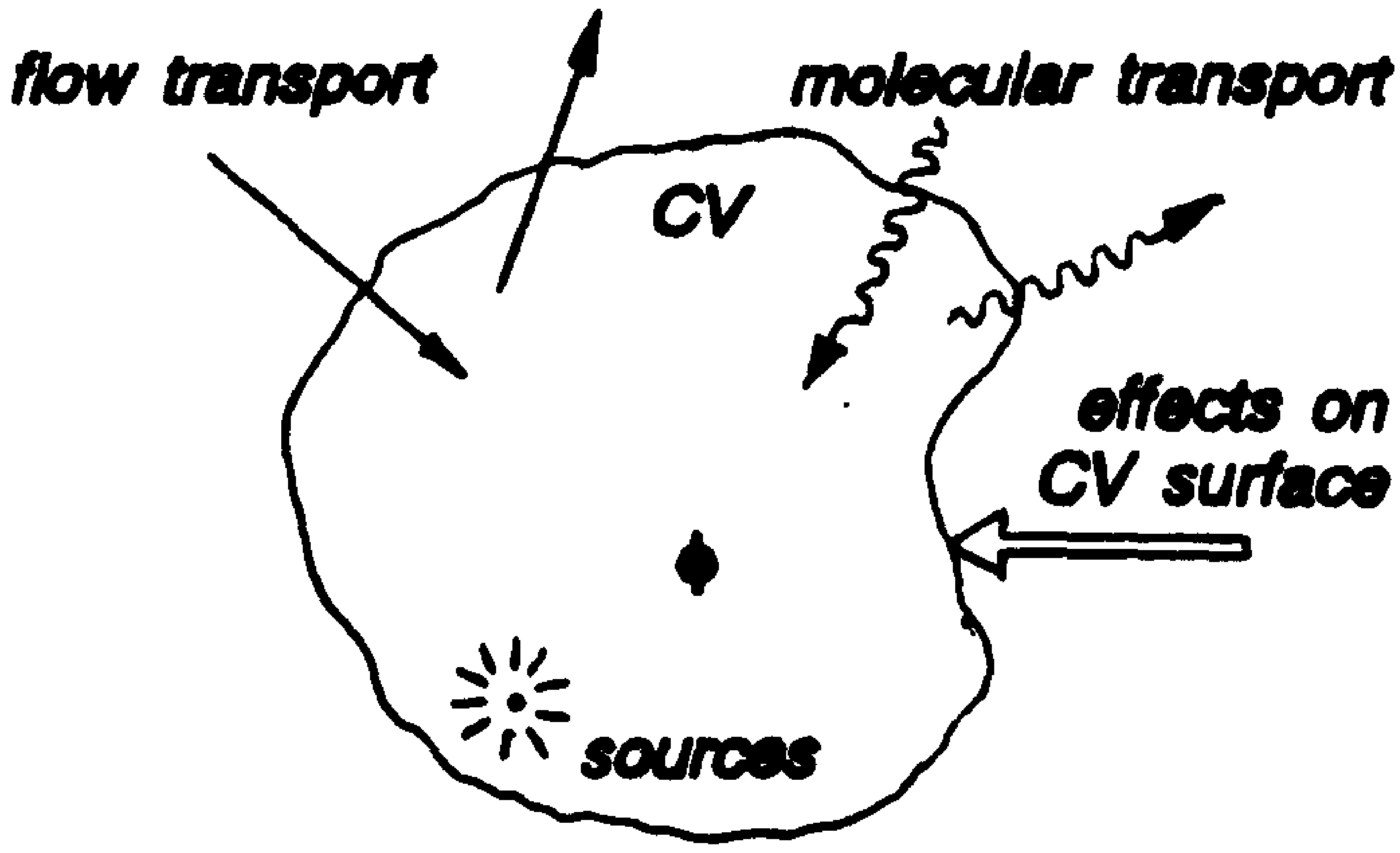


Figure A.1.1 Processes affecting the value of phi in a CV

The above physical processes determine the value of the conserved species, ϕ , that will exist in the CV. The net rate of change of transported variable may be written as:

(RATE OF CHANGE OF ϕ) =
(NET FLOW TRANSPORT) +
(NET MOLECULAR TRANSPORT) +
(SOURCES / SINKS) +
(EFFECTS ON CV SURFACE)

These physical processes can be represented mathematically in the form of partial differential equations. The conservation equation for the transported species, ϕ , may then be written in the form of a **general transport equation**, in Cartesian tensor notation:

$$\frac{\partial(\rho\phi)}{\partial t}+\frac{\partial(\rho u_j\phi)}{\partial x_j}=\frac{\partial}{\partial x_j}\left(\Gamma_\phi\frac{\partial\phi}{\partial x_j}\right)+S_\phi \qquad A.1.1$$

where the value of ϕ depends on the solution dependent variable.

A.2 Conservation of Mass

We wish to determine the rate of change of mass within the finite control volume shown in Figure A.2.1 Here, a two-dimensional Cartesian CV is considered:

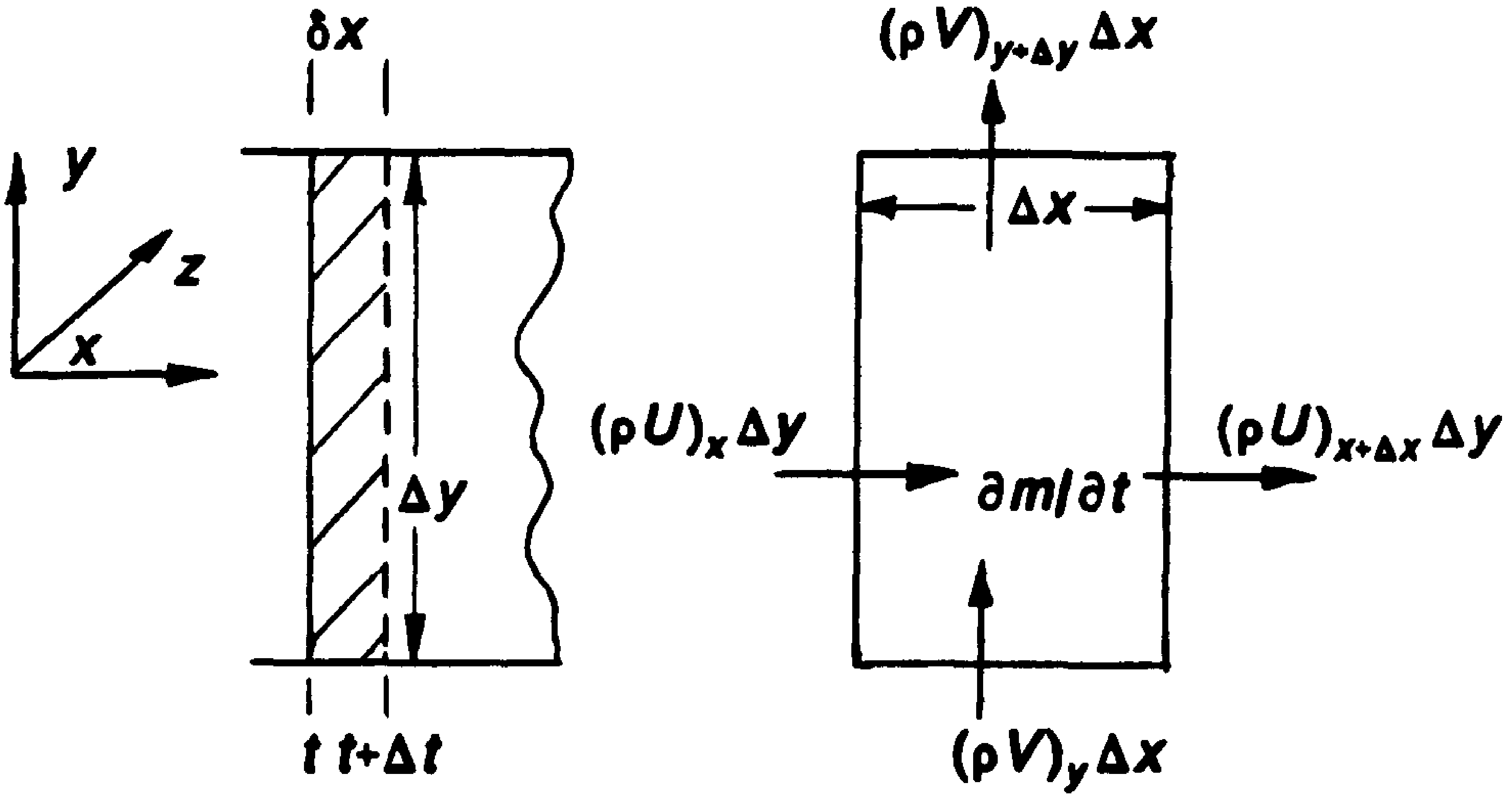


Figure A.2.1 Control volume for mass conservation

The principle of mass conservation maintains that:

[***mass accumulation rate***] = [***rate of mass in***] - [***rate of mass out***]

where the mass m contained within the CV is $\rho\Delta x\Delta y.1$ and the term $\frac{\rho\delta x\Delta y}{\Delta t}=\rho U\Delta y$ ·

thus the following balance equation may be written:

$$\frac{\partial \rho}{\partial t}\Delta x\Delta y = [(\rho U)_x - (\rho U)_{x+\Delta x}]\Delta y + [(\rho V)_y - (\rho V)_{y+\Delta y}]\Delta x$$

division by $\Delta x\Delta y$ yields the difference equation:

$$\frac{\partial \rho}{\partial t} = -\frac{(\rho U)_{x+\Delta x}-(\rho U)_x}{\Delta x} - \frac{(\rho V)_{y+\Delta y}-(\rho V)_y}{\Delta y}$$

now, taking $\lim \Delta x\Delta y \to 0$ gives the differential equation for mass conservation (continuity equation) for 2D flows:

$$\frac{\partial \rho}{\partial t} + \frac{\partial(\rho U)}{\partial x} + \frac{\partial(\rho V)}{\partial y} = 0$$

For 3D flows:

$$\frac{\partial \rho}{\partial t} + \frac{\partial(\rho U)}{\partial x} + \frac{\partial(\rho V)}{\partial y} + \frac{\partial(\rho W)}{\partial z} = 0$$

In tensor notation the continuity equation may be written as:

$$\frac{\partial \rho}{\partial t} + \frac{\partial(\rho U_i)}{\partial x_i} = 0 \qquad \text{A.2.1}$$

In a cylindrical coordinate system the continuity equation becomes:

$$\frac{\partial \rho}{\partial t} + \frac{1}{r}\frac{\partial r\rho V_r}{\partial r} + \frac{1}{r}\frac{\partial \rho V_\theta}{\partial \theta} + \frac{\partial \rho V_z}{\partial z} = 0$$

A.3 Conservation of Momentum

Since momentum is itself a vector quantity, we require 3 equations for 3 space directions x,y,z. In an analogous manner with the conservation of mass derivation in section A.2 we wish to determine the rate of change of momentum within the finite control volume shown in Figure A.3.1 Once again, a two dimensional CV is considered:

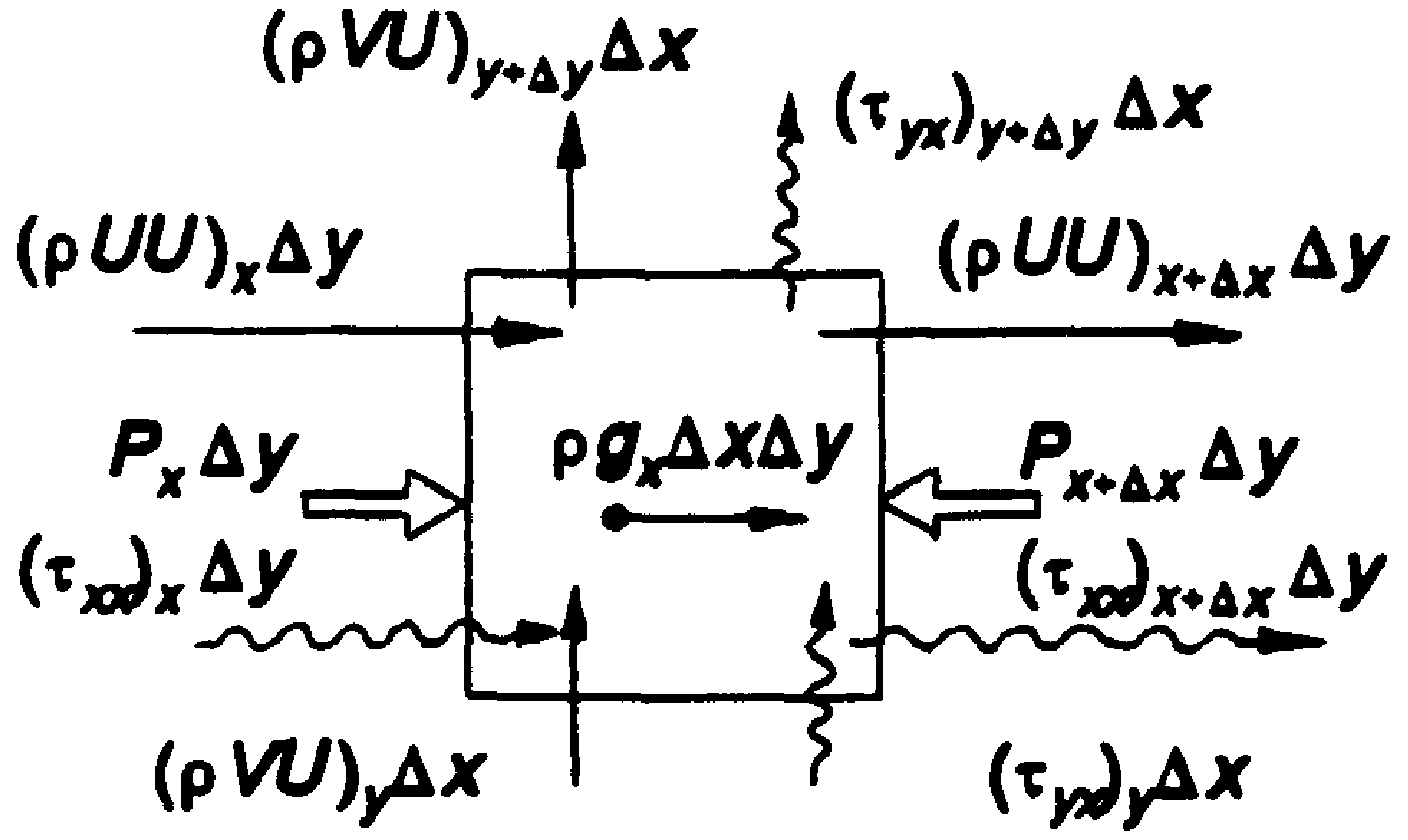


Figure A.3.1 Control volume for momentum conservation

The principle of momentum conservation implies that:

[***net accumulation rate of momentum***] = [***net flow transport (convection)***] +
[***net molecular transport (diffusion)***] +
[***sum of forces on control volume***]

The accumulation rate of x-momentum is represented by:

$$\frac{\partial(\rho U)}{\partial t}\Delta x\Delta y$$

Flow transport (convection) of momentum may be written as:

$$[(\rho UU)_x-(\rho UU)_{x+\Delta x}]\Delta y+[(\rho VU)_y-(\rho VU)_{y+\Delta y}]\Delta x$$

Molecular transport takes place through the diffusion of momentum via the fluid viscous stresses:

$$[(\tau_{xx})_x-(\tau_{xx})_{x+\Delta x}]\Delta y+[(\tau_{yx})_y-(\tau_{yx})_{y+\Delta y}]\Delta x$$

where the subscript notation on the viscous stress terms, for example yx, indicates quantities acting on the face perpendicular to the y axis and in the x direction. Taking moments of the forces about an arbitrary axis and setting the result equal to zero for a fluid CV in equilibrium leads directly to the result that

$$\tau_{xy}=\tau_{yx}$$

$$\tau_{xz}=\tau_{zx}$$

$$\tau_{yz}=\tau_{zy}$$

must hold in general.

The net pressure force is given by:

$$(P_x-P_{x+\Delta x})\Delta y$$

and volumetric forces, for example, gravitation are:

$$\rho g_x \Delta x \Delta y$$

Dividing by $\Delta x \Delta y$ and taking $\lim \Delta x \Delta y \to 0$ transforms the difference equation into a differential equation i.e. the 2D x-momentum equation, which may be written as:

$$\frac{\partial(\rho U)}{\partial t}+\frac{\partial(\rho U^2)}{\partial x}+\frac{\partial(\rho VU)}{\partial y}=-\frac{\partial P}{\partial x}-\frac{\partial\tau_{xx}}{\partial x}-\frac{\partial\tau_{yx}}{\partial y}+\rho g_x \qquad A.3.1$$

Equation A.3.1 is a direct representation of the principles embodied within Newton's Second Law - that the sum of the forces on the CV is equal to the rate of change of momentum within the CV.

A.3.1 Stress-Strain Relation

Stokes' Law, valid for Newtonian fluids, is adopted to relate fluid viscous stresses to fluid deformation rates (i.e. normal and transverse velocity gradients). Unlike a solid, a fluid cannot sustain a shear force. When forces are engaged the fluid will change shape to adopt a stressless condition and it is found that the rate at which the fluid changes shape (deforms) is related to the magnitude of viscous stress within the fluid.

Considering a hydrostatic analysis of the fluid CV we note that the negative of the pressure term P contained in equation A.3.1 is related to to the average of the normal hydrostatic stresses. This relationship holds whether there is motion (and therefore viscous normal stresses) or not,i.e.

$$-P = \frac{\sigma_x + \sigma_y + \sigma_z}{3} \qquad A3.2$$

where σ is the hydrostatic fluid stress.

A.3.1.1 Viscous Shear Stress

Consider the shear component in one direction only as shown in Figure A.3.2.

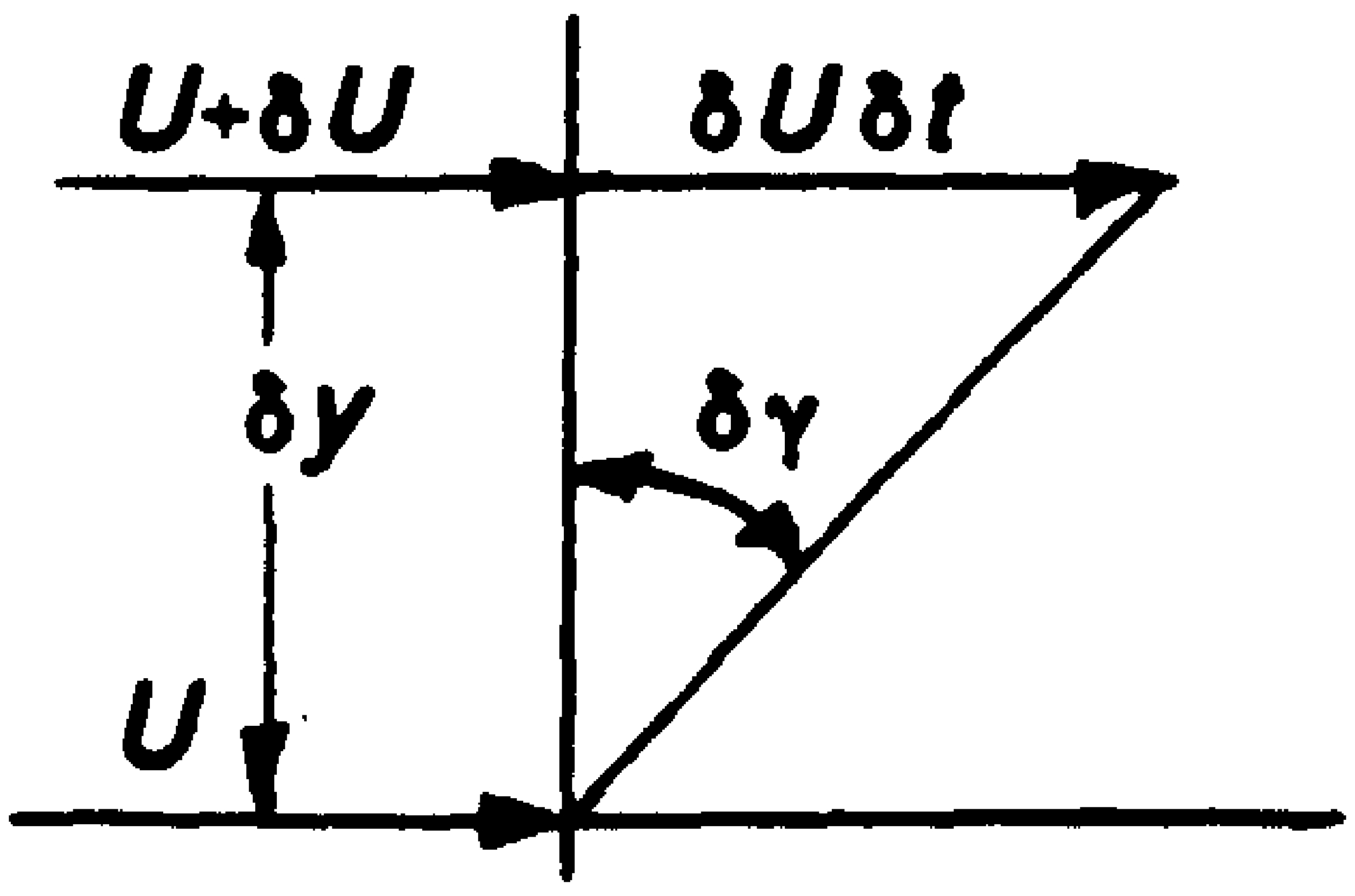


Figure A.3.2 Shear component derivation

For laminar flow the shear stress is defined as:

$$\tau = \mu \frac{dU}{dy} \qquad A.3.3$$

We can then write, for a small deformation angle,

$$\tan \delta\gamma \approx \delta\gamma = \frac{\delta U \delta t}{\delta y}$$

$$or \quad \frac{\delta U}{\delta y} = \frac{\delta \gamma}{\delta t}$$

thus in the limit

$$\delta y \rightarrow 0, \quad \tau = \mu \frac{\partial \gamma}{\partial t} \qquad A.3.4$$

Thus the shear stress is proportional to the time rate of change of deformation. When distortion takes place in two dimensions, we may write the <u>total</u> angle of deformation as $\gamma = \gamma_1 + \gamma_2$, thus:

$$\tau = \mu \frac{\delta}{\delta t}(\gamma_1 + \gamma_2) = \mu\left(\frac{\delta U}{\delta y} + \frac{\delta V}{\delta x}\right)$$

and in the limit $\delta x, \delta y \rightarrow 0$ we may write for the shear stress term in equation A.3.1:

$$\tau_{yx} = \mu\left(\frac{\partial U}{\partial y} + \frac{\partial V}{\partial x}\right)$$

A.3.1.2 Viscous Normal Stress

Considering the stresses within a cube we observe that the shear stresses give rise to **principal stresses** in the principal planes which are at an angle of 45 degrees to the co-ordinate axes. As shown in Figure A.3.3 the original cube

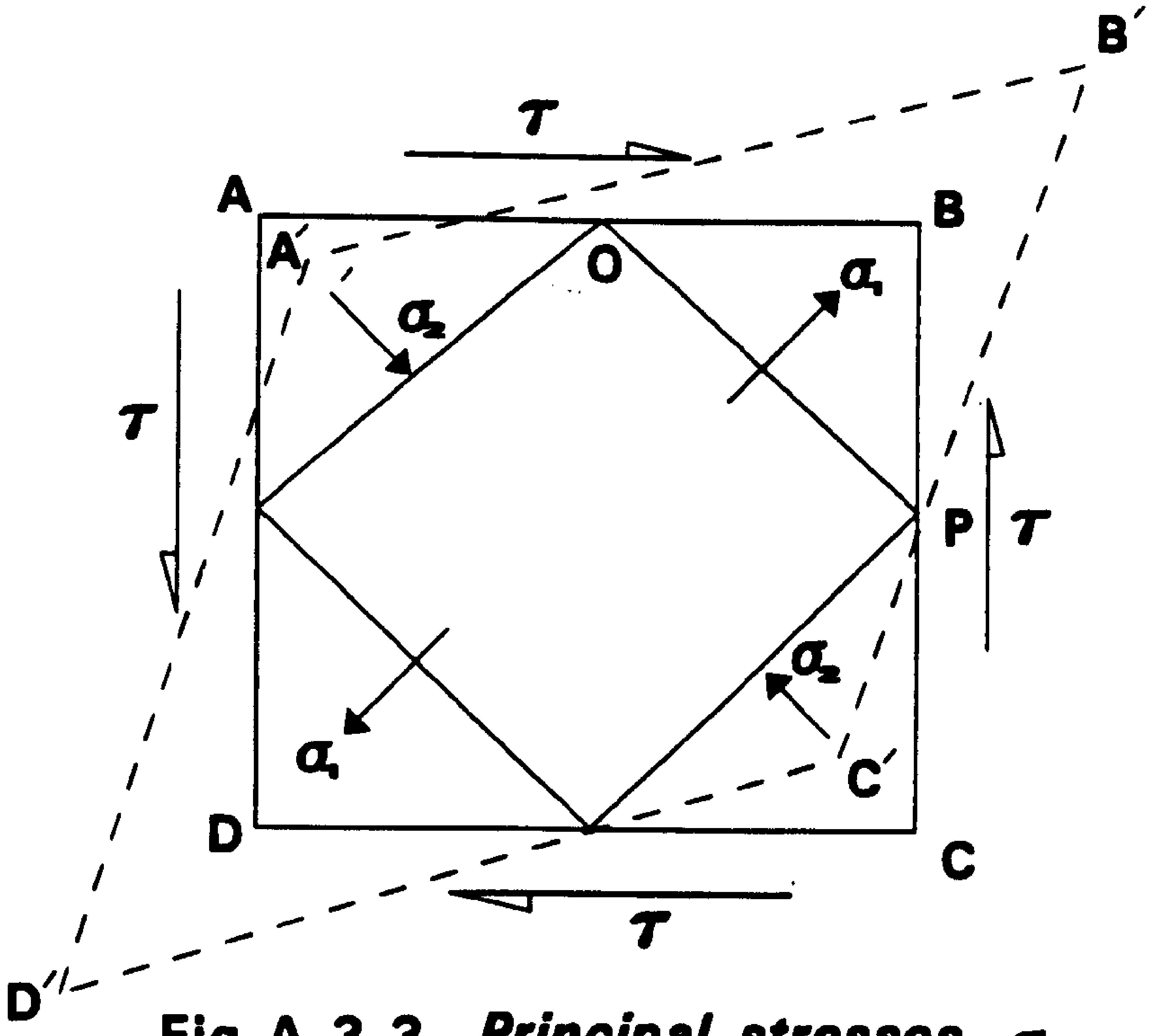


Fig.A.3.3 *Principal stresses σ*

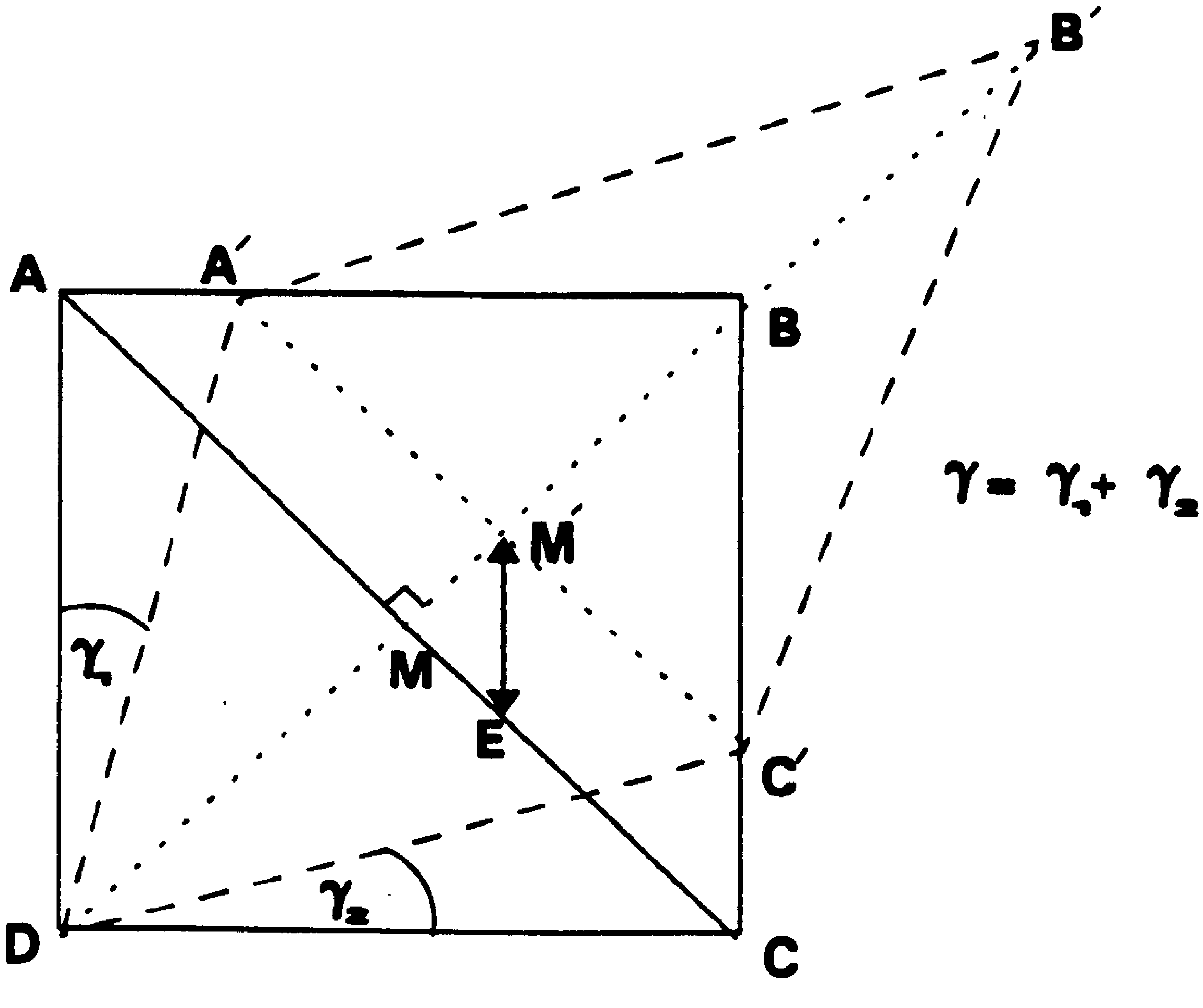
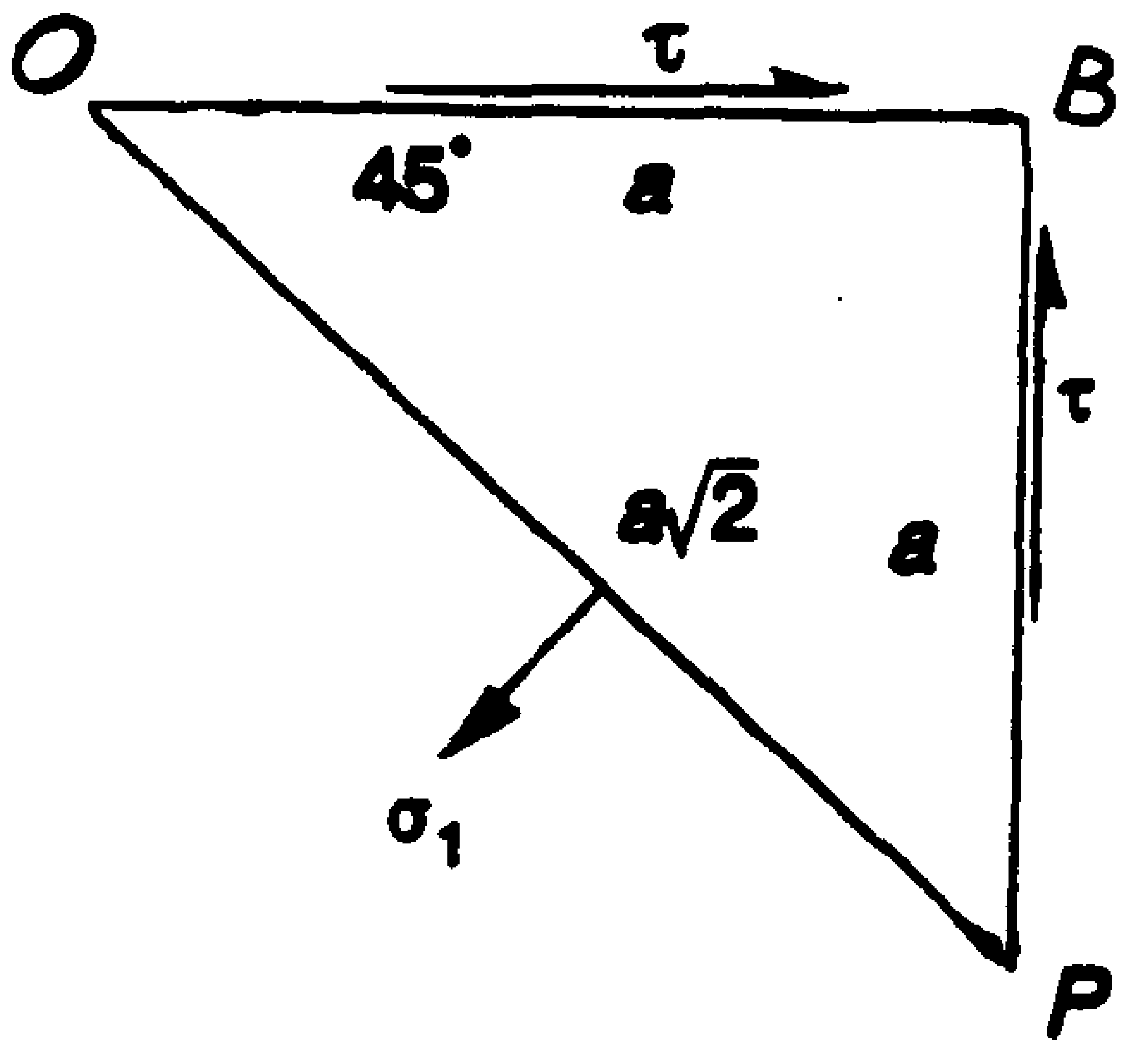


Fig.A.3.4 *Calculation of strain.*

ABCD is deformed under the action of the stresses to A′B′C′D′. Consider the top right hand corner OBP of unit depth:



The equilibrium condition along the diagonal is given by:

$$2a\tau \sin 45^\circ = a\sqrt{2}\,\sigma_1$$

thus

$$\sigma_1 = \tau \qquad \textit{+ve tension}$$

similarly

$$\sigma_2 = -\tau \qquad \textit{-ve compression}$$

thus

$$\sigma_1 - \sigma_2 = 2\tau \qquad A3.5$$

To find the change of length of the diagonal DB consider the cube as shown in Figure A.3.4. We note that:

$$AA' = CC' = EM' = DA\tan\left(\frac{\gamma}{2}\right) = DA\left(\frac{\gamma}{2}\right) \quad \textit{for small angles}$$

Now,

$$MM' = \left(\frac{EM'}{\sqrt{2}}\right) = \frac{(DA\,\gamma/2)}{\sqrt{2}}$$

$$\textit{and} \quad DM = \frac{DA}{\sqrt{2}}$$

Thus the strain of DB may be written as:

$$\varepsilon_1 = \frac{BB'}{DB} = \frac{MM'}{DM} = \frac{(DA\,\gamma/2)/\sqrt{2}}{DA/\sqrt{2}} = \frac{\gamma}{2}$$

Similarly,

$$\varepsilon_2 = -\frac{\gamma}{2}$$

$$\therefore \; \varepsilon_1 - \varepsilon_2 = \gamma \qquad A.3.6$$

Now using equation A.3.4 and substituting from equations A.3.5 and A.3.6 we may write:

$$\sigma_1 - \sigma_2 = 2\tau = 2\mu\frac{\partial}{\partial t}(\varepsilon_1 - \varepsilon_2) = 2\mu\left(\frac{\partial \varepsilon_1}{\partial t} - \frac{\partial \varepsilon_2}{\partial t}\right)$$

If direction 1 is taken along the x-axis and 2 along the y-axis then:

$$\sigma_x - \sigma_y = 2\mu\left(\frac{\partial \varepsilon_x}{\partial t} - \frac{\partial \varepsilon_y}{\partial t}\right) \qquad A.3.7$$

As shown in Figure A.3.5 the strain in the x-direction is given by:

$$\delta\varepsilon_x = \frac{\partial U}{\partial x}\delta t$$

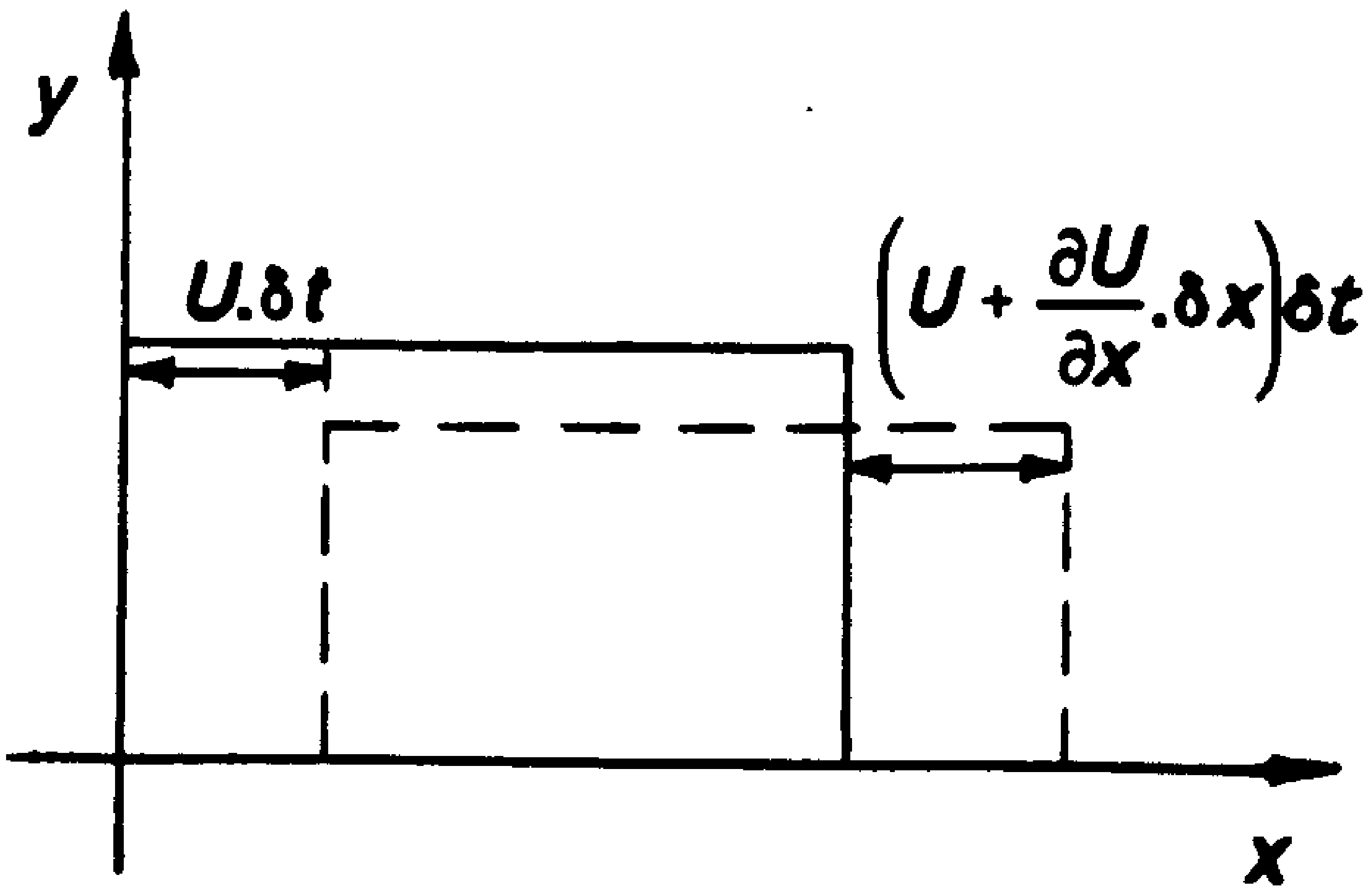


Fig.A3.5 Strain in x-direction

$$\text{strain in x-direction } \delta\varepsilon_x = \frac{\left(U + \frac{\partial U}{\partial x}.\delta x\right)\delta t - U.\delta t}{\delta x}$$

$$= \frac{\partial U}{\partial x}.\delta t$$

$$\Rightarrow \quad \frac{\delta\varepsilon_x}{\delta t} = \frac{\partial U}{\partial x}$$

thus

$$\frac{\delta\varepsilon_x}{\delta t} = \frac{\partial U}{\partial x} = \frac{\partial\varepsilon_x}{\partial t} \quad \text{as } t \to 0$$

$$\text{similarly} \quad \frac{\partial\varepsilon_y}{\partial t} = \frac{\partial V}{\partial y} \qquad \text{A3.8}$$

Substituting equation A.3.7 in equation A.3.8 and hence:

$$\sigma_x = \tau_{xx} = 2\,\mu\left(\frac{\partial U}{\partial x}\right)$$

$$\sigma_y = \tau_{yy} = 2\,\mu\left(\frac{\partial V}{\partial y}\right)$$

$$\sigma_z = \tau_{zz} = 2\,\mu\left(\frac{\partial W}{\partial z}\right)$$

A.3.2 The Navier-Stokes Equations

If we assume that the hydrostatic stress is as given in equation A.3.2 we can thus represent the viscous normal stress term in equation A.3.1 as:

$$\tau_{xx} = P + \frac{2}{3}\mu\left(\frac{\partial U}{\partial x} + \frac{\partial V}{\partial y} + \frac{\partial W}{\partial z}\right) - 2\mu\frac{\partial U}{\partial x} \qquad A.3.9$$

and the viscous shear stress term as:

$$\tau_{yx} = -\mu\left(\frac{\partial U}{\partial y} + \frac{\partial V}{\partial x}\right) \qquad A.3.10$$

We may write equation A.3.1 in 3D non-conservative form:

$$\frac{\partial(U)}{\partial t} + U\frac{\partial(U)}{\partial x} + V\frac{\partial(U)}{\partial y} + W\frac{\partial(U)}{\partial z} = -\frac{1}{\rho}\frac{\partial P}{\partial x} - \frac{\partial \tau_{xx}}{\partial x} - \frac{\partial \tau_{yx}}{\partial y} - \frac{\partial \tau_{zx}}{\partial z} + \rho g_x \qquad A.3.11$$

substituting for all viscous normal and shear stresses and re-arranging we may write:

$$\frac{\partial(U)}{\partial t} + U\frac{\partial(U)}{\partial x} + V\frac{\partial(U)}{\partial y} + W\frac{\partial u}{\partial z} = -\frac{1}{\rho}\frac{\partial P}{\partial x} + \nu\left(\frac{\partial^2 U}{\partial x^2} + \frac{\partial^2 U}{\partial y^2} + \frac{\partial^2 U}{\partial z^2}\right)$$
$$+ \frac{\nu}{3}\frac{\partial}{\partial x}\left(\frac{\partial U}{\partial x} + \frac{\partial V}{\partial y} + \frac{\partial W}{\partial z}\right) + \rho g_x \qquad A.3.12$$

Equation A.3.12 represents the x-direction **Navier-Stokes** equation. The full set of Navier-Stokes equations for unsteady compressible viscous flow may then be written in Cartesian tensor notation as:

$$\frac{\partial U_i}{\partial t}+U_j\frac{\partial U_i}{\partial x_j}=\frac{1}{\rho}\frac{\partial \sigma_{ij}}{\partial x_j}+\rho g_i$$

***where*:** **A.3.13**

$$\sigma_{ij}=-P\delta_{ij}+\mu\left(\frac{\partial U_i}{\partial x_j}+\frac{\partial U_j}{\partial x_i}\right)-\frac{2}{3}\mu\frac{\partial U_k}{\partial x_k}\delta_{ij}$$

Note:

y-momentum (x_2) equation is generated by setting $i=2$;
z-momentum (x_3) equation is generated by setting $i=3$
j is a running index

For an incompressible fluid the overlined term in equation A.3.12 is zero.

In cylindrical coordinates the Navier-Stokes equations may be written as:

r-component

$$\frac{\partial v_r}{\partial t}+v_r\frac{\partial v_r}{\partial r}+\frac{v_\theta}{r}\frac{\partial v_r}{\partial \theta}-\frac{v_\theta^2}{r}+v_z\frac{\partial v_r}{\partial z}$$

$$=-\frac{1}{\rho}\frac{\partial p}{\partial r}+\nu\left[\frac{\partial}{\partial r}\left(\frac{1}{r}\frac{\partial}{\partial r}(rv_r)\right)+\frac{1}{r^2}\frac{\partial^2 v_r}{\partial \theta^2}-\frac{2}{r^2}\frac{\partial v_\theta}{\partial \theta}+\frac{\partial^2 v_r}{\partial z^2}\right]+\rho g_r$$

θ-component

$$\frac{\partial v_\theta}{\partial t}+v_r\frac{\partial v_\theta}{\partial r}+\frac{v_\theta}{r}\frac{\partial v_\theta}{\partial \theta}-\frac{v_r v_\theta}{r}+v_z\frac{\partial v_\theta}{\partial z}$$

$$=-\frac{1}{\rho r}\frac{\partial p}{\partial \theta}+\nu\left[\frac{\partial}{\partial r}\left(\frac{1}{r}\frac{\partial}{\partial r}(rv_\theta)\right)+\frac{1}{r^2}\frac{\partial^2 v_\theta}{\partial \theta^2}-\frac{2}{r^2}\frac{\partial v_r}{\partial \theta}+\frac{\partial^2 v_\theta}{\partial z^2}\right]+\rho g_\theta$$

z-component

$$\frac{\partial v_z}{\partial t}+v_r\frac{\partial v_z}{\partial r}+\frac{v_\theta}{r}\frac{\partial v_z}{\partial \theta}+v_z\frac{\partial v_z}{\partial z}$$

$$=-\frac{1}{\rho}\frac{\partial p}{\partial z}+\nu\left[\frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial v_z}{\partial r}\right)+\frac{1}{r^2}\frac{\partial^2 v_z}{\partial \theta^2}+\frac{\partial^2 v_z}{\partial z^2}\right]+\rho g_z$$

APPENDIX B

DERIVATION OF THE TURBULENT MOMENTUM EQUATIONS

APPENDIX B

DERIVATION OF THE TURBULENT MOMENTUM EQUATIONS

We consider the instantaneous Navier-Stokes (momentum) equations B.1 and the mass continuity equation B.2 in Cartesian tensor notation. In equation B.2 it is assumed that the accumulation rate of mass is zero:

$$\frac{\partial \tilde{u}_i}{\partial t}+\tilde{u}_j\frac{\partial \tilde{u}_i}{\partial x_j}=\frac{1}{\rho}\frac{\partial \tilde{\sigma}_{ij}}{\partial x_j} \qquad B.1$$

$$\frac{\partial(\rho \tilde{u}_i)}{\partial x_i}=0 \qquad B.2$$

where:

$\tilde{u}_i$= instantaneous velocity at (x_i,t)

$\tilde{\sigma}_{ij}$= fluid stress tensor

$$\tilde{\sigma}_{ij}=-\tilde{p}\delta_{ij}+2\mu\tilde{s}_{ij}-\frac{2}{3}\mu\frac{\partial \tilde{u}_k}{\partial x_k}\delta_{ij} \qquad B.3$$

δ_{ij}= Kronecker delta which equals

1.0 if $i=j$ and zero else.

$\tilde{s}_{ij}$= rate of strain of fluid

$$=\frac{1}{2}\left(\frac{\partial \tilde{u}_i}{\partial x_j}+\frac{\partial \tilde{u}_j}{\partial x_i}\right) \qquad B.4$$

Now, substituting for the above values gives:

$$\frac{\partial \tilde{u}_i}{\partial t}+\tilde{u}_j\frac{\partial \tilde{u}_i}{\partial x_j}=-\frac{1}{\rho}\frac{\partial \tilde{p}}{\partial x_i}\delta_{ij}+\nu\frac{\partial}{\partial x_j}\left[\left(\frac{\partial \tilde{u}_i}{\partial x_j}+\frac{\partial \tilde{u}_j}{\partial x_i}\right)-\frac{2}{3}\left(\frac{\partial \tilde{u}_k}{\partial x_k}\right)\delta_{ij}\right] \qquad B.5$$

$$\rightarrow \quad \frac{\partial \tilde{U}_i}{\partial t}+\tilde{U}_j\frac{\partial \tilde{U}_i}{\partial x_j}=-\frac{1}{\rho}\frac{\partial \tilde{p}}{\partial x_j}\delta_{ij}+\nu\left(\frac{\partial^2 \tilde{U}_i}{\partial x_j \partial x_j}+\frac{\partial}{\partial x_j}\left(\frac{\partial \tilde{U}_j}{\partial x_i}\right)-\frac{2}{3}\frac{\partial}{\partial x_j}\left(\frac{\partial \tilde{U}_k}{\partial x_k}\right)\delta_{ij}\right)$$

$$\rightarrow \quad \frac{\partial \tilde{U}_i}{\partial t}+\tilde{U}_j\frac{\partial \tilde{U}_i}{\partial x_j}=-\frac{1}{\rho}\frac{\partial \tilde{p}}{\partial x_j}\delta_{ij}+\nu\left(\frac{\partial^2 \tilde{U}_i}{\partial x_j \partial x_j}+\frac{1}{3}\frac{\partial}{\partial x_j}\left(\frac{\partial \tilde{U}_k}{\partial x_k}\right)\delta_{ij}\right)$$

assuming incompressible flow we let:

$$\frac{\partial \tilde{U}_k}{\partial x_k}=0$$

so the instantaneous momentum equation in incompressible flow becomes:

$$\frac{\partial \tilde{U}_i}{\partial t}+\tilde{U}_j\frac{\partial \tilde{U}_i}{\partial x_j}=-\frac{1}{\rho}\frac{\partial \tilde{p}}{\partial x_j}\delta_{ij}+\nu\left(\frac{\partial^2 \tilde{U}_i}{\partial x_j \partial x_j}\right) \qquad B.6$$

If we now apply the Reynolds decomposition for average and fluctuating quantities,we let:

instantaneous $\tilde{\phi}$ = *average* $\bar{\phi}$ + *fluctuating* ϕ'

for example:

$$\tilde{U}_i=\bar{u}_i+u_i'$$

where the average value is interpreted as a time average:

$$\bar{u}_i=\frac{1}{\Delta T}\int_{t}^{t+\Delta T}\tilde{U}_i\,dt \qquad B.7$$

such that the mean velocity may be a function of time and where the averaging time period is small in comparison with the macroscopic fluid motion time scale, yet large in comparison with typical turbulent fluctuating component time scales.

The mean value of the fluctuating component is itself zero by definition:

$$\overline{u_i'} = \frac{1}{\Delta T}\int_{t_*}^{t_*+\Delta T} (\tilde{u}_i - \bar{u}_i)\,dt \qquad B.8$$

The instantaneous pressure term is also decomposed into an average and fluctuating component:

$$\tilde{p} = \bar{p} + p'$$

Continuity

Applying the Reynolds decomposition to the continuity equation we obtain:

$$\frac{\partial \tilde{u}_i}{\partial x_i} = \frac{\partial}{\partial x_i}(\bar{u}_i + u_i') = \frac{\partial \bar{u}_i}{\partial x_i} + \frac{\partial u_i'}{\partial x_i} = 0$$

If the time average of all terms in this equation is taken we find that:

$$\frac{\partial \bar{\bar{u}}_i}{\partial x_i} = 0$$

hence the mean flow is incompressible.

Subtracting these two equations we find that the turbulent velocity fluctuations are also incompressible:

$$\frac{\partial u_i'}{\partial x_i} = 0$$

The following rules are applied to the temporal averaging process:

time average of (mean value).(fluctuating value) is zero: e.g.

$$\overline{\bar{u}\phi'} = o$$

time average of linear terms: e.g.

$$\overline{\frac{\partial \tilde{\phi}}{\partial x}} = \overline{\frac{\partial(\bar{\phi}+\phi')}{\partial x}} = \overline{\frac{\partial \bar{\phi}}{\partial x}} + \overline{\frac{\partial \phi'}{\partial x}} = \frac{\partial \bar{\phi}}{\partial x}$$

time average of non-linear terms: e.g.

$$\overline{\frac{\partial(\tilde{u}\bar{\phi})}{\partial x}}=\overline{\frac{\partial[(\bar{u}+u')(\phi+\phi')]}{\partial x}}=\overline{\frac{\partial(\bar{u}\phi+\bar{u}\phi'+u'\phi+u'\phi')}{\partial x}}=\frac{\partial(\bar{u}\bar{\phi})}{\partial x}+\frac{\partial\,\overline{u'\phi'}}{\partial x}$$

where the value ϕ is the solution dependent variable.

The term $\overline{u'\phi'}$ is non-zero if the two signals are correlated.

Momentum

Applying the Reynolds decomposition to the instantaneous momentum equation B.6 and time averaging all terms we obtain:

$$\frac{\partial\bar{u}_i}{\partial t}+\bar{u}_j\frac{\partial\bar{u}_i}{\partial x_j}+\overline{u_j'\frac{\partial u_i'}{\partial x_j}}=-\frac{1}{\rho}\frac{\partial\bar{p}}{\partial x_j}\delta_{i}+\nu\left(\frac{\partial^2\bar{u}_i}{\partial x_j\partial x_j}\right) \qquad \textit{B.9}$$

Now, using the continuity equation we can write the convection terms in equation B.9 in conservative form: e.g.

$$\overline{u_j'\frac{\partial u_i'}{\partial x_j}}=\frac{\partial}{\partial x_j}\overline{u_i'u_j'}$$

The term $\overline{u_i'u_j'}$ is known as the '**Reynolds stress**'. It represents the transfer of momentum by the turbulent motion and can be perceived as an agency that produces stresses in the mean flow. This being the case, the mean flow will do work against the Reynolds stresses.

Equation B.9 can be re-arranged so that all stress terms are grouped together:

$$\frac{\partial\bar{u}_i}{\partial t}+\frac{\partial}{\partial x_j}\bar{u}_i\bar{u}_j=\frac{1}{\rho}\frac{\partial}{\partial x_j}\bar{p}\delta_{i}+\frac{\partial}{\partial x_j}\left(\nu_i\left(\frac{\partial\bar{u}_i}{\partial x_j}\right)-\overline{u_i'u_j'}\right) \qquad \textit{B.10}$$

Equation B.10 represents the Reynolds-decomposed unsteady turbulent momentum equation for incompressible flow.

APPENDIX C

Q1 FILE FOR TWO-DIMENSIONAL LAMINAR VORTEX SHEDDING

```
TALK=T;RUN( 1, 1);VDU=GRAPHICS
    GROUP 1. Run title and other preliminaries
TEXT(BLUFF BODY CHANNEL FLOW)
REAL(REYNO,UIN,A,B,H,DELT,WAS)
B=1.
UIN=1.
WAS=0.
A=1.*B
H=4.*B
REYNO=500.
    GROUP 2. Transience; time-step specification
   STEADY=T
STEADY=F
GRDPWR(T,20,1.6,1.)
    GROUP 3. X-direction grid specification
NX=75
XULAST=B
XFRAC(1)=-5.
XFRAC(2)=1.
XFRAC(3)=3.
XFRAC(4)=0.5
XFRAC(5)=6.
XFRAC(6)=0.25
XFRAC(7)=4.
XFRAC(8)=0.1
XFRAC(9)=4.
XFRAC(10)=0.05
XFRAC(11)=2.
XFRAC(12)=0.1
XFRAC(13)=2.
XFRAC(14)=0.2
XFRAC(15)=8.
XFRAC(16)=0.1
XFRAC(17)=6.
XFRAC(18)=0.25
XFRAC(19)=25.
XFRAC(20)=0.5
XFRAC(21)=1.
XFRAC(22)=2.
XFRAC(23)=1.
XFRAC(24)=3.
XFRAC(25)=1.
XFRAC(26)=3.5
XFRAC(27)=1.
XFRAC(28)=4.5
XFRAC(29)=1.
XFRAC(30)=7.
XFRAC(31)=1.
XFRAC(32)=9.
XFRAC(33)=1.
XFRAC(34)=15.
XFRAC(35)=1.
XFRAC(36)=30.
XFRAC(37)=1.
XFRAC(38)=70.
XFRAC(39)=1.
XFRAC(40)=400.
    GROUP 4. Y-direction grid specification
NY=40
YVLAST=B
YFRAC(1)=-4
YFRAC(2)=0.1
YFRAC(3)=3.
YFRAC(4)=0.2
YFRAC(5)=4.
YFRAC(6)=0.1
```

```
YFRAC(7)=4.
YFRAC(8)=0.025
YFRAC(9)=10.
YFRAC(10)=0.1
YFRAC(11)=4.
YFRAC(12)=0.025
YFRAC(13)=4.
YFRAC(14)=0.1
YFRAC(15)=3.
YFRAC(16)=0.2
YFRAC(17)=4
YFRAC(18)=0.1
    GROUP 5. Z-direction grid specification
    GROUP 6. Body-fitted coordinates or grid distortion
    GROUP 7. Variables stored, solved & named
SOLUTN(P1,Y,Y,N,N,N,N)
SOLUTN(U1,Y,Y,N,N,N,N)
SOLUTN(V1,Y,Y,N,N,N,N)
UCONV=T
    GROUP 8. Terms (in differential equations) & devices
    GROUP 9. Properties of the medium (or media)
RHO1=1.2
ENUL=(UIN*B)/REYNO
    GROUP 10. Inter-phase-transfer processes and properties
    GROUP 11. Initialization of variable or porosity fields
    The following commands between the stars
    represent the initial parabolic velocity
    profile field used in the steady asymmetric
    calculation. GRND9 is the parabolic velocity profile.
    ************************************
    PATCH(INITU1,INIVAL,1,NX,1,NY,1,1,1,1)
    INIT(INITU1,U1,0.,GRND9)
    FIINIT(U1)=1.0
    ************************************
CONPOR(0.0,CELL,20,29,16,25,1,1)
    GROUP 12. Convection and diffusion adjustments
    GROUP 13. Boundary conditions and special sources

PATCH(SUCCAU,CELL,1,NX,1,NY,1,1,1,LSTEP)
COVAL(SUCCAU,U1,GRND,GRND)
COVAL(SUCCAU,V1,GRND,GRND)

PATCH(WALL1,NWALL,1,NX,NY,NY,1,1,1,LSTEP)
COVAL(WALL1,U1,1.0,0.0)
PATCH(WALL2,SWALL,1,NX,1,1,1,1,1,LSTEP)
COVAL(WALL2,U1,1.0,0.0)
PATCH(WALL3,SWALL,20,29,26,26,1,1,1,LSTEP)
COVAL(WALL3,U1,1.0,WAS)
PATCH(WALL4,NWALL,20,29,15,15,1,1,1,LSTEP)
COVAL(WALL4,U1,1.0,-WAS)
PATCH(WALL5,EWALL,19,19,16,25,1,1,1,LSTEP)
COVAL(WALL5,V1,1.0,WAS)
PATCH(WALL6,WWALL,30,30,16,25,1,1,1,LSTEP)
COVAL(WALL6,V1,1.0,-WAS)

PATCH(INLET,WEST,1,1,1,NY,1,1,1,LSTEP)
COVAL(INLET,P1,FIXFLU,GRND9)
COVAL(INLET,U1,ONLYMS,GRND9)

PATCH(OUTLET,EAST,NX,NX,1,NY,1,1,1,LSTEP)
COVAL(OUTLET,P1,10000.,0.)
COVAL(OUTLET,U1,ONLYMS,GRND9)

    GROUP 14. Downstream pressure for PARAB=.TRUE.
    GROUP 15. Termination of sweeps
LSWEEP=30
```

```
RESREF(P1)=0.03
RESREF(U1)=0.4
RESREF(V1)=0.4
    GROUP 16. Termination of iterations
    GROUP 17. Under-relaxation devices
RELAX(P1,LINRLX,0.5)
RELAX(U1,FALSDT,0.03)
RELAX(V1,FALSDT,0.03)
    GROUP 18. Limits on variables or increments to them
    GROUP 19. Data communicated by satellite to GROUND
    solve for U via SUCCA ?
LG(1)=T
    solve for V via SUCCA ?
LG(2)=T
    solve for SCALAR via SUCCA ?
LG(3)=F
    CALL MAKE(YG2D) for lam. parab. profile ?
LG(4)=T
RG(1)=4.*UIN/H
RG(2)=-RG(1)/H
RG(3)=RHO1
    GROUP 20. Preliminary print-out
    GROUP 21. Print-out of variables
    INIFLD=T
OUTPUT(P1,N,Y,N,Y,Y,Y)
OUTPUT(U1,Y,Y,N,Y,Y,Y)
OUTPUT(V1,N,Y,N,Y,Y,Y)
    GROUP 22. Spot-value print-out
NPLT=1
IXMON=40
IYMON=15
    GROUP 23. Field print-out and plot control
PATCH(P1-2,PROFIL,37,37,27,27,1,1,1,LSTEP)
PLOT(P1-2,P1,0.0,0.0)
PATCH(P1-3,PROFIL,42,42,NY-1,NY-1,1,1,1,LSTEP)
PLOT(P1-3,P1,0.0,0.0)
PATCH(U1-1,PROFIL,42,42,31,31,1,1,1,LSTEP)
PLOT(U1-1,U1,0.0,0.0)
PATCH(U1-5,PROFIL,34,34,31,31,1,1,1,LSTEP)
PLOT(U1-5,U1,0.0,0.0)
PATCH(U1-6,PROFIL,42,42,13,13,1,1,1,LSTEP)
PLOT(U1-6,U1,0.0,0.0)
PATCH(U1-7,PROFIL,34,34,13,13,1,1,1,LSTEP)
PLOT(U1-7,U1,0.0,0.0)
PATCH(V1-1,PROFIL,42,42,31,31,1,1,1,LSTEP)
PLOT(V1-1,V1,0.0,0.0)
PATCH(V1-6,PROFIL,42,42,13,13,1,1,1,LSTEP)
PLOT(V1-6,V1,0.0,0.0)
IPROF=3
    GROUP 24. Dumps for restarts
    RESTRT(ALL)
STOP
```

APPENDIX D

Q1 FILE FOR TWO-DIMENSIONAL LAMINAR FLOW BACKWARD FACING STEP PROBLEM

```
TALK=T;RUN(1,1);VDU=VGACURSR
    GROUP 1. Run title and other preliminaries
TEXT(BACKWARD FACING STEP)
REAL(REYNOS,UMAX,UAV,HINLET,CHRDIM)
    GROUP 2. Transience; time-step specification
    GROUP 3. X-direction grid specification
NX=54
XULAST=1.E-3
XFRAC(1)=1.3
XFRAC(2)=3.6
XFRAC(3)=6.5
XFRAC(4)=10.0
XFRAC(5)=14.0
XFRAC(6)=18.0
XFRAC(7)=22.6
XFRAC(8)=27.6
XFRAC(9)=32.8
XFRAC(10)=38.3
XFRAC(11)=44.0
XFRAC(12)=50.0
XFRAC(13)=56.0
XFRAC(14)=61.7
XFRAC(15)=67.2
XFRAC(16)=72.4
XFRAC(17)=77.4
XFRAC(18)=82.0
XFRAC(19)=86.0
XFRAC(20)=90.0
XFRAC(21)=93.5
XFRAC(22)=96.4
XFRAC(23)=98.7
XFRAC(24)=100.0
XFRAC(25)=100.3
XFRAC(26)=101.05
XFRAC(27)=102.2
XFRAC(28)=103.67
XFRAC(29)=105.5
XFRAC(30)=107.64
XFRAC(31)=110.1
XFRAC(32)=113.0
XFRAC(33)=116.3
XFRAC(34)=119.23
XFRAC(35)=119.93
XFRAC(36)=120.53
XFRAC(37)=120.93
XFRAC(38)=121.33
XFRAC(39)=121.93
XFRAC(40)=122.73
XFRAC(41)=123.53
XFRAC(42)=124.73
XFRAC(43)=127.0
XFRAC(44)=131.0
XFRAC(45)=136.0
XFRAC(46)=143.0
XFRAC(47)=153.0
XFRAC(48)=168.0
XFRAC(49)=184.0
XFRAC(50)=200.0
XFRAC(51)=220.0
XFRAC(52)=240.0
XFRAC(53)=265.0
XFRAC(54)=300.0
    GROUP 4. Y-direction grid specification
NY=25
YVLAST=1.E-3
YFRAC(1)=0.13
```

```
YFRAC(2)=0.38
YFRAC(3)=0.71
YFRAC(4)=1.10
YFRAC(5)=1.55
YFRAC(6)=2.05
YFRAC(7)=2.6
YFRAC(8)=3.15
YFRAC(9)=3.65
YFRAC(10)=4.10
YFRAC(11)=4.49
YFRAC(12)=4.82
YFRAC(13)=5.07
YFRAC(14)=5.2
YFRAC(15)=5.6
YFRAC(16)=6.05
YFRAC(17)=6.52
YFRAC(18)=7.01
YFRAC(19)=7.5
YFRAC(20)=8.01
YFRAC(21)=8.52
YFRAC(22)=9.04
YFRAC(23)=9.57
YFRAC(24)=9.83
YFRAC(25)=10.1
    GROUP 5. Z-direction grid specification
    GROUP 6. Body-fitted coordinates or grid distortion
    GROUP 7. Variables stored, solved & named
SOLVE(P1,U1,V1)
    GROUP 8. Terms (in differential equations) & devices
UCONV=T
    GROUP 9. Properties of the medium (or media)
RHO1=1.2
REYNOS=200.
ENUL=1.E-5
HINLET=5.2E-3
CHRDIM=2.0*HINLET
UMAX=(3.0*REYNOS*ENUL)/(2.0*CHRDIM)
UAV=UMAX/1.5
    GROUP 10. Inter-phase-transfer processes and properties
    GROUP 11. Initialization of variable or porosity fields
CONPOR(0.0,CELL,1,24,15,NY,1,1)
    GROUP 12. Convection and diffusion adjustments
    GROUP 13. Boundary conditions and special sources

PATCH(INLET,WEST,1,1,1,14,1,1,1,1)
COVAL(INLET,P1,FIXFLU,RHO1*UAV)
COVAL(INLET,U1,ONLYMS,UAV)

PATCH(SUCCAU,CELL,1,NX,1,NY,1,1,1,1)
COVAL(SUCCAU,U1,GRND,GRND)
COVAL(SUCCAU,V1,GRND,GRND)

PATCH(EOUTLE,EAST,NX,NX,1,NY,1,1,1,1)
COVAL(EOUTLE,P1,5000.,0.0)

PATCH(WALL1,NWALL,1,24,14,14,1,1,1,1)
COVAL(WALL1,U1,1.0,0.0)

PATCH(WALL2,WWALL,25,25,15,NY,1,1,1,1)
COVAL(WALL2,V1,1.0,0.0)

PATCH(WALL3,NWALL,25,NX,NY,NY,1,1,1,1)
COVAL(WALL3,U1,1.0,0.0)

PATCH(WALL4,SWALL,1,NX,1,1,1,1,1,1)
COVAL(WALL4,U1,1.0,0.0)
```

```
    GROUP 14. Downstream pressure for PARAB=.TRUE.
    GROUP 15. Termination of sweeps
LSWEEP=100
RESREF(P1)=1.E-6
RESREF(U1)=1.E-6
RESREF(V1)=1.E-6
    GROUP 16. Termination of iterations
LITER(P1)=-1
    GROUP 17. Under-relaxation devices
   RELAX(P1,LINRLX,0.6)
   RELAX(U1,FALSDT,3.0)
   RELAX(V1,FALSDT,3.0)

RELAX(P1,LINRLX,0.3)
RELAX(U1,FALSDT,0.003)
RELAX(V1,FALSDT,0.003)

    GROUP 18. Limits on variables or increments to them
    GROUP 19. Data communicated by satellite to GROUND
    solve for U1 via succa2d?
LG(1)=T
    solve for V1 via succa2d?
LG(2)=T
    solve for scalars via succa2d?
LG(3)=F
    GROUP 20. Preliminary print-out
    GROUP 21. Print-out of variables
    GROUP 22. Spot-value print-out
IXMON=29
IYMON=18
NPLT=1
    GROUP 23. Field print-out and plot control
    GROUP 24. Dumps for restarts
    tstswp=-1
STOP
```

APPENDIX E

Q1 FILE FOR TWO-DIMENSIONAL SCALAR CONVECTION TEST PROBLEM

```
TALK=T;RUN(1,1);VDU=GRAPHICS
    GROUP 1. Run title and other preliminaries
TEXT(SUCCA TEST FLOW)
REAL(REYNOS,UIN,VIN,CELLDIM)
REYNOS=200
CELLDIM=1.
    ***************************************************
    vary VIN keeping UIN=1.0 and compare C1 profile with
    hybrid across the vertical line of cells at IX=6.
    ***************************************************
UIN=1.0
VIN=1.0
    GROUP 2. Transience; time-step specification
    GROUP 3. X-direction grid specification
GRDPWR(X,11,1.0,1.)
    GROUP 4. Y-direction grid specification
GRDPWR(Y,11,1.0,1.)
    GROUP 5. Z-direction grid specification
    GROUP 6. Body-fitted coordinates or grid distortion
    GROUP 7. Variables stored, solved & named
SOLVE(P1,U1,V1,C1)
    GROUP 8. Terms (in differential equations) & devices
UCONV=T
TERMS(C1,N,Y,N,N,Y,N)
    GROUP 9. Properties of the medium (or media)
RHO1=1.0
ENUL=(CELLDIM*UIN)/REYNOS
    GROUP 10. Inter-phase-transfer processes and properties
    GROUP 11. Initialization of variable or porosity fields
    GROUP 12. Convection and diffusion adjustments
    GROUP 13. Boundary conditions and special sources

PATCH(BOTTOM,SOUTH,2,NX,1,1,1,1,1,1)
COVAL(BOTTOM,P1,FIXFLU,RHO1*VIN)
COVAL(BOTTOM,U1,ONLYMS,UIN)
COVAL(BOTTOM,V1,ONLYMS,VIN)
COVAL(BOTTOM,C1,ONLYMS,1.E-8)

PATCH(SIDE1,WEST,1,1,2,NY,1,1,1,1)
COVAL(SIDE1,P1,FIXFLU,RHO1*UIN)
COVAL(SIDE1,U1,ONLYMS,UIN)
COVAL(SIDE1,V1,ONLYMS,VIN)
COVAL(SIDE1,C1,ONLYMS,1.0)

PATCH(SUCCAU,CELL,2,NX,2,NY,1,1,1,1)
COVAL(SUCCAU,C1,GRND,GRND)

PATCH(EOUTLE,EAST,NX,NX,1,NY,1,1,1,1)
COVAL(EOUTLE,P1,10000.,0.0)

PATCH(NOUTLE,NORTH,1,NX,NY,NY,1,1,1,1)
COVAL(NOUTLE,P1,10000.,0.0)

    GROUP 14. Downstream pressure for PARAB=.TRUE.
    GROUP 15. Termination of sweeps
LSWEEP=80
RESREF(P1)=1.E-8
RESREF(U1)=1.E-8
RESREF(V1)=1.E-8
RESREF(C1)=1.E-8
    GROUP 16. Termination of iterations
    GROUP 17. Under-relaxation devices
RELAX(P1,LINRLX,0.5)
RELAX(U1,FALSDT,0.3)
RELAX(V1,FALSDT,0.3)
RELAX(C1,FALSDT,0.3)
```

```
    GROUP 18. Limits on variables or increments to them
    GROUP 19. Data communicated by satellite to GROUND
    solve for U1 via succa2d?
LG(1)=F
    solve for V1 via succa2d?
LG(2)=F
    solve for scalars via succa2d?
LG(3)=T
    GROUP 20. Preliminary print-out
    GROUP 21. Print-out of variables
    GROUP 22. Spot-value print-out
IXMON=2
IYMON=3
NPLT=1
    GROUP 23. Field print-out and plot control
    GROUP 24. Dumps for restarts
STOP
```

APPENDIX F

Q1 FILE FOR TWO-DIMENSIONAL TURBULENT VORTEX SHEDDING

```
TALK=T;RUN(1,1);VDU=GRAPHICS
    GROUP 1. Run title and other preliminaries
TEXT(2-D PLANE TURBULENT VORTEX SHEDDING)
REAL(REYNOS,UIN,DIA,A,B,WAS)
REAL(KEIN,EPIN,FIXVAP,KEJET,EPJET)
DIA=8.E-2
B=0.27*DIA
A=(2/3)*B
    GROUP 2. Transience; time-step specification
STEADY=F
GRDPWR(T,400,0.04,1.0)
    GROUP 3. X-direction grid specification
SUBGRD(X,1,5,1.5*DIA,1.0)
SUBGRD(X,6,27,1.0*DIA,-2.2)
SUBGRD(X,28,-34,A,1.5)
SUBGRD(X,35,67,6.0*DIA,1.4)
SUBGRD(X,68,68,2*B,1.0)
SUBGRD(X,69,69,3*B,1.0)
SUBGRD(X,70,70,3.5*B,1.0)
SUBGRD(X,71,71,4.5*B,1.0)
SUBGRD(X,72,72,7*B,1.0)
SUBGRD(X,73,73,9*B,1.0)
SUBGRD(X,74,74,15*B,1.0)
SUBGRD(X,75,75,30*B,1.0)
SUBGRD(X,76,76,70*B,1.0)
SUBGRD(X,77,77,400*B,1.0)
    GROUP 4. Y-direction grid specification
SUBGRD(Y,1,-18,0.365*DIA,2.0)
SUBGRD(Y,19,-32,B,1.5)
SUBGRD(Y,33,-50,0.365*DIA,2.0)
    GROUP 5. Z-direction grid specification
    GROUP 6. Body-fitted coordinates or grid distortion
    GROUP 7. Variables stored, solved & named
SOLVE(P1,U1,V1,KE,EP)
UCONV=T
    GROUP 8. Terms (in differential equations) & devices
    GROUP 9. Properties of the medium (or media)
RHO1=1.207
ENUL=1.5E-5
ENUT=GRND3
REYNOS=1.3E5
EL1=GRND4
UIN=(ENUL*REYNOS)/DIA
WAS=0.
   WAS=15*UIN
    GROUP 10. Inter-phase-transfer processes and properties
    GROUP 11. Initialization of variable or porosity fields
KEIN=0.01*UIN**2
EPIN=((0.09**0.75)*KEIN**1.5)/(0.1*DIA)
   FIINIT(KE)=KEIN
   FIINIT(EP)=EPIN
CONPOR(0.0,CELL,28,34,19,32,1,1)
    GROUP 12. Convection and diffusion adjustments
    GROUP 13. Boundary conditions and special sources

PATCH(INLET,WEST,1,1,1,NY,1,1,1,LSTEP)
COVAL(INLET,P1,FIXFLU,RHO1*UIN)
COVAL(INLET,U1,ONLYMS,UIN)
COVAL(INLET,KE,ONLYMS,KEIN)
COVAL(INLET,EP,ONLYMS,EPIN)

PATCH(SUCCAU,CELL,1,NX,1,NY,1,1,1,LSTEP)
COVAL(SUCCAU,U1,GRND,GRND)
COVAL(SUCCAU,V1,GRND,GRND)
COVAL(SUCCAU,KE,GRND,GRND)
COVAL(SUCCAU,EP,GRND,GRND)
```

```
PATCH(KESOURCE,PHASEM,1,NX,1,NY,1,1,1,LSTEP)
COVAL(KESOURCE,KE,GRND4,GRND4)
COVAL(KESOURCE,EP,GRND4,GRND4)

PATCH(TURBWL1,NWALL,1,NX,NY,NY,1,1,1,LSTEP)
COVAL(TURBWL1,U1,GRND2,0.0)
COVAL(TURBWL1,KE,GRND2,GRND2)
COVAL(TURBWL1,EP,GRND2,GRND2)

PATCH(TURBWL2,SWALL,28,34,33,33,1,1,1,LSTEP)
COVAL(TURBWL2,U1,GRND2,WAS)
COVAL(TURBWL2,KE,GRND2,GRND2)
COVAL(TURBWL2,EP,GRND2,GRND2)

PATCH(TURBWL3,WWALL,35,35,19,32,1,1,1,LSTEP)
COVAL(TURBWL3,V1,GRND2,-WAS)
COVAL(TURBWL3,KE,GRND2,GRND2)
COVAL(TURBWL3,EP,GRND2,GRND2)

PATCH(TURBWL4,NWALL,28,34,18,18,1,1,1,LSTEP)
COVAL(TURBWL4,U1,GRND2,-WAS)
COVAL(TURBWL4,KE,GRND2,GRND2)
COVAL(TURBWL4,EP,GRND2,GRND2)

PATCH(TURBWL5,EWALL,27,27,19,32,1,1,1,LSTEP)
COVAL(TURBWL5,V1,GRND2,WAS)
COVAL(TURBWL5,KE,GRND2,GRND2)
COVAL(TURBWL5,EP,GRND2,GRND2)

PATCH(TURBWL6,SWALL,1,NX,1,1,1,1,1,LSTEP)
COVAL(TURBWL6,U1,GRND2,0.0)
COVAL(TURBWL6,KE,GRND2,GRND2)
COVAL(TURBWL6,EP,GRND2,GRND2)

    *** Replace FIXVAL (too tight a restriction) ***

FIXVAP=UIN*RHO1*1000.

PATCH(OUTLET,EAST,NX,NX,1,NY,1,1,1,LSTEP)
COVAL(OUTLET,P1,FIXVAP,0.0)

    GROUP 14. Downstream pressure for PARAB=.TRUE.
    GROUP 15. Termination of sweeps
LSWEEP=15
RESREF(P1)=DIA*UIN/1000.
RESREF(U1)=RHO1*DIA*UIN**2/1000.
RESREF(V1)=RHO1*DIA*UIN**2/1000.
RESREF(KE)=RHO1*DIA*UIN*KEIN/1000.
RESREF(EP)=RHO1*DIA*UIN*EPIN/1000.
    GROUP 16. Termination of iterations
    GROUP 17. Under-relaxation devices
KELIN=1
RELAX(P1,LINRLX,0.4)
RELAX(U1,FALSDT,0.05)
RELAX(V1,FALSDT,0.05)
RELAX(KE,LINRLX,0.3)
RELAX(EP,LINRLX,0.3)
   Alternatively
   RELAX(P1,LINRLX,0.4)
   RELAX(U1,FALSDT,0.05)
   RELAX(V1,FALSDT,0.05)
   RELAX(KE,FALSDT,0.001)
   RELAX(EP,FALSDT,0.001)
    GROUP 18. Limits on variables or increments to them
    GROUP 19. Data communicatee by satellite to GROUND
```

```
GENK=T
LG(1)=T
LG(2)=T
LG(3)=T
    GROUP 20. Preliminary print-out
ECHO=F
    GROUP 21. Print-out of variables
    GROUP 22. Spot-value print-out
IXMON=35
IYMON=34
NPLT=1
    GROUP 23. Field print-out and plot control
PATCH(PRESS1,PROFIL,44,44,36,36,1,1,1,LSTEP)
PLOT(PRESS1,P1,0.0,0.0)
PATCH(AXVEL1,PROFIL,44,44,36,36,1,1,1,LSTEP)
PLOT(AXVEL1,U1,0.0,0.0)
PATCH(RAVEL1,PROFIL,44,44,36,36,1,1,1,LSTEP)
PLOT(RAVEL1,V1,0.0,0.0)
PATCH(PRESS2,PROFIL,40,40,18,18,1,1,1,LSTEP)
PLOT(PRESS2,P1,0.0,0.0)
PATCH(AXVEL2,PROFIL,40,40,18,18,1,1,1,LSTEP)
PLOT(AXVEL2,U1,0.0,0.0)
PATCH(RAVEL2,PROFIL,40,40,18,18,1,1,1,LSTEP)
PLOT(RAVEL2,V1,0.0,0.0)
PATCH(PRES2,PROFIL,57,57,40,40,1,1,1;LSTEP)
PLOT(PRES2,P1,0.0,0.0)
PATCH(AXVL2,PROFIL,57,57,40,40,1,1,1,LSTEP)
PLOT(AXVL2,U1,0.0,0.0)
PATCH(RAVE2,PROFIL,57,57,40,40,1,1,1,LSTEP)
PLOT(RAVE2,V1,0.0,0.0)
IPROF=3
    GROUP 24. Dumps for restarts
STOP
```

APPENDIX G

Q1 FILE FOR TWO-DIMENSIONAL TURBULENT VORTEX SHEDDING IN PULSATING FLOW

```
TALK=T;RUN(1,1);VDU=GRAPHICS
    GROUP 1. Run title and other preliminaries
TEXT(2-D PLANE PULSATING FLOW TURBULENT V/S)
   ***********************************************
   REMEMBER TO REMOVE ALL FIINIT(n's) ON ANY RESTRT!

   USE STEADY ANALYSIS FOR STASS..................
   i.e. COVAL(TURBWL,U1,1.0,WAS)
   this just takes the local wall cell vel=WAS
   as opp. to using GRND2 wall function + WAS.

   USE SHELL TOOL ON TRANSIENT....................

   IN GROUND PUT (120,60) FOR TURBLUFF.GRND
   ***********************************************
REAL(REYNOS,UAV,HT,A,B,WAS)
REAL(KEIN,EPIN,FIXVAP)
REAL(FREQRS,FREQVS,FREQPL,AMPL,ARMS,PI)
    FREQRS= stdy v/s frequency    (rad/sec)
    FREQVS= stdy v/s frequency    (Hz) (from prev. calc.)
    FREQPL= Pulsating flow freq   (rad/sec)
    AMPL  = U-Pulsat Amplitude    (m/sec)
    ARMS  = U(rms) Puls Amplitude (m/sec)
PI=3.1415927
HT=8.E-2
B=0.27*HT
A=(2.0/3.0)*B
FREQVS=301.2
   i.e. the vortex shedding frequency in steady flow
FREQRS=2.0*PI*FREQVS
FREQPL=2.0*FREQRS
   i.e. set the ratio (FP/FVS)=2.0
    GROUP 2. Transience; time-step specification
STEADY=F
GRDPWR(T,700,0.028,1.0)
    GROUP 3. X-direction grid specification
SUBGRD(X,1,-10,56.0*HT,2.8)
SUBGRD(X,11,15,1.5*HT,1.0)
SUBGRD(X,16,32,1.0*HT,-2.8)
SUBGRD(X,33,-42,A,1.3)
SUBGRD(X,43,63,1.0*HT,1.7)
SUBGRD(X,64,79,2.0*HT,1.0)
SUBGRD(X,80,90,3.0*HT,1.2)
SUBGRD(X,91,94,2.0*HT,1.1)
SUBGRD(X,95,95,2*B,1.0)
SUBGRD(X,96,96,3*B,1.0)
SUBGRD(X,97,97,3.5*B,1.0)
SUBGRD(X,98,98,4.5*B,1.0)
SUBGRD(X,99,99,7*B,1.0)
SUBGRD(X,100,100,9*B,1.0)
SUBGRD(X,101,101,15*B,1.0)
SUBGRD(X,102,102,30*B,1.0)
SUBGRD(X,103,103,70*B,1.0)
SUBGRD(X,104,104,400*B,1.0)
    GROUP 4. Y-direction grid specification
SUBGRD(Y,1,-18,0.365*HT,2.0)
SUBGRD(Y,19,-32,B,1.5)
SUBGRD(Y,33,-50,0.365*HT,2.0)
    GROUP 5. Z-direction grid specification
    GROUP 6. Body-fitted coordinates or grid distortion
    GROUP 7. Variables stored, solved & named
SOLUTN(P1,Y,Y,Y,N,N,N)
SOLVE(U1,V1,KE,EP)
    GROUP 8. Terms (in differential equations) & devices
UCONV=T
    GROUP 9. Properties of the medium (or media)
```

```
RHO1=1.207
ENUL=1.50E-5
ENUT=GRND3
REYNOS=1.0E5
EL1=GRND4
UAV=(ENUL*REYNOS)/HT
    WAS=5.0*UAV
ARMS=0.194*UAV
    i.e. ratio U(rms)/UAV= 0.194
AMPL=(ARMS*PI)/2.0
    i.e.U(rms)=(2/PI)*(AMPL/2)...area under sin wave is
    (intgrl)sinU dt between 0 and PI.This equals 2.0 hence
    height,x,=2/PI i.e. this faction of AMPL/2.i.e.
    Urms=(x/1.0)*(AMPL/2.0) where 1.0 is ht of sin curve.
    We must then/ by a further 2 such that me get (+/-) ampl=AMPL.
     GROUP 10. Inter-phase-transfer processes and properties
     GROUP 11. Initialization of variable or porosity fields
KEIN=0.01*UAV**2
EPIN=((0.09**0.75)*KEIN**1.5)/(0.1*HT)
    FIINIT(KE)=KEIN
    FIINIT(EP)=EPIN
    FIINIT(U1)=UAV
CONPOR(0.0,CELL,33,42,19,32,1,1)
     GROUP 12. Convection and diffusion adjustments
     GROUP 13. Boundary conditions and special sources

PATCH(INLET,WEST,1,1,1,NY,1,1,1,LSTEP)
COVAL(INLET,P1,FIXFLU,GRND9)
COVAL(INLET,U1,ONLYMS,GRND9)
COVAL(INLET,KE,ONLYMS,KEIN)
COVAL(INLET,EP,ONLYMS,EPIN)

PATCH(SUCCAU,CELL,1,NX,1,NY,1,1,1,LSTEP)
COVAL(SUCCAU,U1,GRND,GRND)
COVAL(SUCCAU,V1,GRND,GRND)
COVAL(SUCCAU,KE,GRND,GRND)
COVAL(SUCCAU,EP,GRND,GRND)

PATCH(KESOURCE,PHASEM,1,NX,1,NY,1,1,1,LSTEP)
COVAL(KESOURCE,KE,GRND4,GRND4)
COVAL(KESOURCE,EP,GRND4,GRND4)

PATCH(1TURBWL,NWALL,1,NX,NY,NY,1,1,1,LSTEP)
COVAL(1TURBWL,U1,GRND2,0.0)
COVAL(1TURBWL,KE,GRND2,GRND2)
COVAL(1TURBWL,EP,GRND2,GRND2)

PATCH(2TURBWL,SWALL,33,42,33,33,1,1,1,LSTEP)
    COVAL(2TURBWL,U1,1.0,WAS)
COVAL(2TURBWL,U1,GRND2,0.0)
COVAL(2TURBWL,KE,GRND2,GRND2)
COVAL(2TURBWL,EP,GRND2,GRND2)

PATCH(3TURBWL,WWALL,43,43,19,32,1,1,1,LSTEP)
    COVAL(3TURBWL,V1,1.0,-WAS)
COVAL(3TURBWL,V1,GRND2,0.0)
COVAL(3TURBWL,KE,GRND2,GRND2)
COVAL(3TURBWL,EP,GRND2,GRND2)

PATCH(4TURBWL,NWALL,33,42,18,18,1,1,1,LSTEP)
    COVAL(4TURBWL,U1,1.0,-WAS)
COVAL(4TURBWL,U1,GRND2,0.0)
COVAL(4TURBWL,KE,GRND2,GRND2)
COVAL(4TURBWL,EP,GRND2,GRND2)

PATCH(5TURBWL,EWALL,32,32,19,32,1,1,1,LSTEP)
```

```
   COVAL(5TURBWL,V1,1.0,WAS)
COVAL(5TURBWL,V1,GRND2,0.0)
COVAL(5TURBWL,KE,GRND2,GRND2)
COVAL(5TURBWL,EP,GRND2,GRND2)

PATCH(6TURBWL,SWALL,1,NX,1,1,1,1,1,LSTEP)
COVAL(6TURBWL,U1,GRND2,0.0)
COVAL(6TURBWL,KE,GRND2,GRND2)
COVAL(6TURBWL,EP,GRND2,GRND2)

    *** Replace FIXVAL (too tight a restriction) ***

FIXVAP=UAV*RHO1*1000.

PATCH(OUTLET,EAST,NX,NX,1,NY,1,1,1,LSTEP)
COVAL(OUTLET,P1,FIXVAP,0.0)

    GROUP 14. Downstream pressure for PARAB=.TRUE.
    GROUP 15. Termination of sweeps
LSWEEP=30
RESREF(P1)=HT*UAV/1000.
RESREF(U1)=RHO1*HT*UAV**2/1000.
RESREF(V1)=RHO1*HT*UAV**2/1000.
RESREF(KE)=RHO1*HT*UAV*KEIN/1000.
RESREF(EP)=RHO1*HT*UAV*EPIN/1000.
    GROUP 16. Termination of iterations
    GROUP 17. Under-relaxation devices
KELIN=1
   Alternative relaxation parameters
   RELAX(P1,LINRLX,0.3)
   RELAX(U1,LINRLX,0.2)
   RELAX(V1,LINRLX,0.2)
   RELAX(KE,LINRLX,0.1)
   RELAX(EP,LINRLX,0.1)
RELAX(P1,LINRLX,0.4)
RELAX(U1,FALSDT,0.05)
RELAX(V1,FALSDT,0.05)
RELAX(KE,LINRLX,0.3)
RELAX(EP,LINRLX,0.3)
    GROUP 18. Limits on variables or increments to them
    GROUP 19. Data communicated by satellite to GROUND
GENK=T
LG(1)=T
LG(2)=T
LG(3)=T
RG(1)=UAV
RG(2)=FREQPL
RG(3)=AMPL
RG(4)=RHO1
    GROUP 20. Preliminary print-out
    GROUP 21. Print-out of variables
OUTPUT(P1,Y,N,N,Y,Y,Y)
    GROUP 22. Spot-value print-out
IXMON=35
IYMON=34
NPLT=1
    GROUP 23. Field print-out and plot control
PATCH(PULSE3,PROFIL,1,1,25,25,1,1,1,LSTEP)
PLOT(PULSE3,P1,0.0,0.0)
PATCH(PULSE,PROFIL,NX,NX,25,25,1,1,1,LSTEP)
PLOT(PULSE,P1,0.0,0.0)
PATCH(PULSE2,PROFIL,1,1,25,25,1,1,1,LSTEP)
PLOT(PULSE2,U1,0.0,0.0)
PATCH(PULSE1,PROFIL,NX-1,NX-1,25,25,1,1,1,LSTEP)
PLOT(PULSE1,U1,0.0,0.0)
PATCH(PRESS1,PROFIL,50,50,46,46,1,1,1,LSTEP)
```

```
PLOT(PRESS1,P1,0.0,0.0)
PATCH(AXVEL1,PROFIL,50,50,46,46,1,1,1,LSTEP)
PLOT(AXVEL1,U1,0.0,0.0)
PATCH(RAVEL1,PROFIL,50,50,46,46,1,1,1,LSTEP)
PLOT(RAVEL1,V1,0.0,0.0)
PATCH(PRESS2,PROFIL,65,65,42,42,1,1,1,LSTEP)
PLOT(PRESS2,P1,0.0,0.0)
PATCH(AXVEL2,PROFIL,65,65,42,42,1,1,1,LSTEP)
PLOT(AXVEL2,U1,0.0,0.0)
PATCH(RAVEL2,PROFIL,65,65,42,42,1,1,1,LSTEP)
PLOT(RAVEL2,V1,0.0,0.0)
PATCH(PRES2,PROFIL,57,57,15,15,1,1,1,LSTEP)
PLOT(PRES2,P1,0.0,0.0)
PATCH(AXVL2,PROFIL,57,57,15,15,1,1,1,LSTEP)
PLOT(AXVL2,U1,0.0,0.0)
PATCH(RAVE2,PROFIL,57,57,15,15,1,1,1,LSTEP)
PLOT(RAVE2,V1,0.0,0.0)
IPROF=3
    GROUP 24. Dumps for restarts
STOP
```

APPENDIX H

ADDITIONAL GROUND CODING FOR PULSATING INLET FLOW BOUNDARY CONDITIONS

```
C------------------- SECTION 16 ------------------- value = GRND4
 1315 CONTINUE
      RETURN
C------------------- SECTION 17 ------------------- value = GRND5
 1316 CONTINUE
      RETURN
C------------------- SECTION 18 ------------------- value = GRND6
 1317 CONTINUE
      RETURN
C------------------- SECTION 19 ------------------- value = GRND7
 1318 CONTINUE
      RETURN
C------------------- SECTION 20 ------------------- value = GRND8
 1319 CONTINUE
      RETURN
C------------------- SECTION 21 ------------------- value = GRND9
 1320 CONTINUE
C
      GTIM=RG(1)+(RG(3)*SIN(RG(2)*ISTEP*DT))
C  i.e. UAV(instantaneous)= UAV + Asin(wt)...
C     ...simple harmonic motion on UAV.
C
      CALL ONLYIF(U1,U1,'ALL')
      CALL FN1(VAL,GTIM)
C  i.e. set U(inlet)= GTIM
C
      CALL ONLYIF(P1,P1,'ALL')
      CALL FN1(VAL,GTIM)
      CALL FN25(VAL,RG(4))
C  i.e. set mass flow(inlet)= RHO1*GTIM (per m**2)
C
      RETURN
C------------------- SECTION 22 ------------------- value = GRND10
 1321 CONTINUE
      RETURN
C*********************************************************************
C
C--- GROUP 14. Downstream pressure for PARAB=.TRUE.
C
   14 CONTINUE
      RETURN
C*********************************************************************
```

APPENDIX J

Q1 FILE FOR THREE-DIMENSIONAL TURBULENT VORTEX SHEDDING IN PIPE FLOW

```
TALK=T;RUN(1,1);VDU=X11-TERM
    GROUP 1. Run title and other preliminaries
TEXT(3D VORTEX SHEDDING)
REAL(REYNOS,PI,WIN,RI,DIA,A,B,WAS)
REAL(KEIN,EPIN,FIXVAP)
PI=3.1415927
DIA=80.E-3
RI=(0.27*DIA)/2.0
B=0.27*DIA
A=(2/3)*B
    GROUP 2. Transience; time-step specification
STEADY=F
GRDPWR(T,350,0.028,1.0)
    GROUP 3. X-direction grid specification
CARTES=F
NX=64;XULAST=1.0
XFRAC(1)=0.3588457
XFRAC(2)=0.7176915
XFRAC(3)=0.8899374
XFRAC(4)=1.0047681
XFRAC(5)=1.1052449
XFRAC(6)=1.1913679
XFRAC(7)=1.263137
XFRAC(8)=1.3205524
XFRAC(9)=1.3636139
XFRAC(10)=1.4066753
XFRAC(11)=1.4210292
XFRAC(12)=1.43
XFRAC(13)=1.435383
   **********************
XFRAC(14)=1.4805207
XFRAC(15)=1.5256584
XFRAC(16)=0.5*PI
XFRAC(17)=1.6159338
XFRAC(18)=1.6610715
XFRAC(19)=1.7062092
   **********************
XFRAC(20)=1.7115922
XFRAC(21)=1.720563
XFRAC(22)=1.7349169
XFRAC(23)=1.7779783
XFRAC(24)=1.8210398
XFRAC(25)=1.8784552
XFRAC(26)=1.9502243
XFRAC(27)=2.0363473
XFRAC(28)=2.1368241
XFRAC(29)=2.2516548
XFRAC(30)=2.4239007
XFRAC(31)=2.7827465
XFRAC(32)=PI
   *********************
XFRAC(33)=3.5004384
XFRAC(34)=3.8592842
XFRAC(35)=4.0315301
XFRAC(36)=4.1463608
XFRAC(37)=4.2468376
XFRAC(38)=4.3329606
XFRAC(39)=4.4047297
XFRAC(40)=4.4621451
XFRAC(41)=4.5052066
XFRAC(42)=4.548268
XFRAC(43)=4.5626219
XFRAC(44)=4.5715926
XFRAC(45)=4.5769756
   **********************
XFRAC(46)=4.6221135
```

```
XFRAC(47)=4.6672513
XFRAC(48)=1.5*PI
XFRAC(49)=4.7575268
XFRAC(50)=4.8026646
XFRAC(51)=4.8478023
    **********************
XFRAC(52)=4.8531853
XFRAC(53)=4.8621561
XFRAC(54)=4.87651
XFRAC(55)=4.9195714
XFRAC(56)=4.9626329
XFRAC(57)=5.0200483
XFRAC(58)=5.0918174
XFRAC(59)=5.1779404
XFRAC(60)=5.2784172
XFRAC(61)=5.3932479
XFRAC(62)=5.5654938
XFRAC(63)=5.9243396
XFRAC(64)=2.0*PI
    GROUP 4. Y-direction grid specification
NY=18;YVLAST=1.E-02
YFRAC(1)=0.216
YFRAC(2)=0.432
YFRAC(3)=0.648
YFRAC(4)=0.864
YFRAC(5)=1.08
YFRAC(6)=1.1534731
YFRAC(7)=1.4336428
YFRAC(8)=1.7157551
YFRAC(9)=2.0138595
YFRAC(10)=2.4057988
YFRAC(11)=2.9158507
YFRAC(12)=3.5663757
YFRAC(13)=4.3611687
YFRAC(14)=5.2502841
YFRAC(15)=6.6101588
YFRAC(16)=7.2382398
YFRAC(17)=7.6960713
YFRAC(18)=8.0
    GROUP 5. Z-direction grid specification
SUBGRD(Z,1,3,1.0*DIA,1.0)
SUBGRD(Z,4,11,1.0*DIA,-2.3)
SUBGRD(Z,12,-17,A,1.7)
SUBGRD(Z,18,37,3.0*DIA,1.7)
SUBGRD(Z,38,47,3.0*DIA,1.0)
SUBGRD(Z,48,48,2*B,1.0)
SUBGRD(Z,49,49,3*B,1.0)
SUBGRD(Z,50,50,3.5*B,1.0)
SUBGRD(Z,51,51,4.5*B,1.0)
SUBGRD(Z,52,52,7*B,1.0)
SUBGRD(Z,53,53,9*B,1.0)
SUBGRD(Z,54,54,15*B,1.0)
SUBGRD(Z,55,55,30*B,1.0)
SUBGRD(Z,56,56,70*B,1.0)
SUBGRD(Z,57,57,400*B,1.0)
    GROUP 6. Body-fitted coordinates or grid distortion
    GROUP 7. Variables stored, solved & named
   SOLVE(P1,U1,V1,W1)
SOLUTN(P1,Y,Y,Y,N,N,N)
SOLUTN(W1,Y,Y,N,N,N,N)
SOLUTN(U1,Y,Y,N,N,N,N)
SOLUTN(V1,Y,Y,N,N,N,N)
SOLUTN(KE,Y,Y,N,N,N,N)
SOLUTN(EP,Y,Y,N,N,N,N)
STORE(EPOR,NPOR,HPOR,VPOR)
    GROUP 8. Terms (in differential equations) & devices
```

```
    GROUP 9. Properties of the medium (or media)
RHO1=1.207
REYNOS=1.3E5
ENUL=1.5E-5
ENUT=GRND3
EL1=GRND4
WIN=(REYNOS*ENUL)/DIA
   WAS=2.0*WIN
    GROUP 10. Inter-phase-transfer processes and properties
    GROUP 11. Initialization of variable or porosity fields
   ***************************************
   The following FIINIT's are for STEADY
   asymmetric calcs ONLY (see WAS)
   ***************************************
   FIINIT(W1)=WIN
KEIN=0.01*WIN**2
EPIN=((0.09**0.75)*KEIN**1.5)/(0.1*DIA)
   FIINIT(KE)=KEIN
   FIINIT(EP)=EPIN
   ***************************************
INIADD=F
FIINIT(EPOR)=1.0
FIINIT(VPOR)=1.0
FIINIT(HPOR)=1.0
FIINIT(NPOR)=1.0
PATCH(BLUPOR,INIVAL,1,NX,1,NY,12,17,1,LSTEP)
INIT(BLUPOR,HPOR,0.0,GRND)
INIT(BLUPOR,EPOR,0.0,GRND)
INIT(BLUPOR,NPOR,0.0,GRND)
INIT(BLUPOR,VPOR,0.0,GRND)
CONPOR(0.0,CELL,-1,-1,1,5,12,12)
CONPOR(0.0,CELL,-2,-2,1,6,12,12)
CONPOR(0.0,CELL,-3,-3,1,7,12,12)
CONPOR(0.0,CELL,-4,-4,1,8,12,12)
CONPOR(0.0,CELL,-5,-5,1,9,12,12)
CONPOR(0.0,CELL,-6,-6,1,10,12,12)
CONPOR(0.0,CELL,-7,-7,1,11,12,12)
CONPOR(0.0,CELL,-8,-8,1,12,12,12)
CONPOR(0.0,CELL,-9,-9,1,13,12,12)
CONPOR(0.0,CELL,-10,-10,1,14,12,12)
CONPOR(0.0,CELL,-11,-11,1,15,12,12)
   indent the following starred CONPOR's
   when considering WAS .GT. 0.0
   otherwise activate turbulent wall
   function on blockage plane faces.
   ***********************************
CONPOR(0.0,CELL,-12,-12,1,16,12,12)
CONPOR(0.0,CELL,-13,-13,1,17,12,12)
CONPOR(0.0,CELL,-14,-14,1,18,12,12)
   ***********************************
CONPOR(0.0,CELL,-15,-15,1,18,12,12)
CONPOR(0.0,CELL,-16,-16,1,18,12,12)
CONPOR(0.0,CELL,-17,-17,1,18,12,12)
CONPOR(0.0,CELL,-18,-18,1,18,12,12)
CONPOR(0.0,CELL,-19,-19,1,18,12,12)
CONPOR(0.0,CELL,-20,-20,1,17,12,12)
CONPOR(0.0,CELL,-21,-21,1,16,12,12)
CONPOR(0.0,CELL,-22,-22,1,15,12,12)
CONPOR(0.0,CELL,-23,-23,1,14,12,12)
   ***********************************
CONPOR(0.0,CELL,-24,-24,1,13,12,12)
CONPOR(0.0,CELL,-25,-25,1,12,12,12)
CONPOR(0.0,CELL,-26,-26,1,11,12,12)
   ***********************************
CONPOR(0.0,CELL,-27,-27,1,10,12,12)
```

```
CONPOR(0.0,CELL,-28,-28,1,9,12,12)
CONPOR(0.0,CELL,-29,-29,1,8,12,12)
CONPOR(0.0,CELL,-30,-30,1,7,12,12)
CONPOR(0.0,CELL,-31,-31,1,6,12,12)
CONPOR(0.0,CELL,-32,-32,1,5,12,12)
CONPOR(0.0,CELL,-33,-33,1,5,12,12)
CONPOR(0.0,CELL,-34,-34,1,6,12,12)
CONPOR(0.0,CELL,-35,-35,1,7,12,12)
CONPOR(0.0,CELL,-36,-36,1,8,12,12)
CONPOR(0.0,CELL,-37,-37,1,9,12,12)
CONPOR(0.0,CELL,-38,-38,1,10,12,12)
CONPOR(0.0,CELL,-39,-39,1,11,12,12)
   ***********************************
CONPOR(0.0,CELL,-40,-40,1,12,12,12)
CONPOR(0.0,CELL,-41,-41,1,13,12,12)
CONPOR(0.0,CELL,-42,-42,1,14,12,12)
   ***********************************
CONPOR(0.0,CELL,-43,-43,1,15,12,12)
CONPOR(0.0,CELL,-44,-44,1,16,12,12)
CONPOR(0.0,CELL,-45,-45,1,17,12,12)
CONPOR(0.0,CELL,-46,-46,1,18,12,12)
CONPOR(0.0,CELL,-47,-47,1,18,12,12)
CONPOR(0.0,CELL,-48,-48,1,18,12,12)
CONPOR(0.0,CELL,-49,-49,1,18,12,12)
CONPOR(0.0,CELL,-50,-50,1,18,12,12)
CONPOR(0.0,CELL,-51,-51,1,18,12,12)
CONPOR(0.0,CELL,-52,-52,1,17,12,12)
CONPOR(0.0,CELL,-53,-53,1,16,12,12)
CONPOR(0.0,CELL,-54,-54,1,15,12,12)
CONPOR(0.0,CELL,-55,-55,1,14,12,12)
   ***********************************
CONPOR(0.0,CELL,-56,-56,1,13,12,12)
CONPOR(0.0,CELL,-57,-57,1,12,12,12)
CONPOR(0.0,CELL,-58,-58,1,11,12,12)
   ***********************************
CONPOR(0.0,CELL,-59,-59,1,10,12,12)
CONPOR(0.0,CELL,-60,-60,1,9,12,12)
CONPOR(0.0,CELL,-61,-61,1,8,12,12)
CONPOR(0.0,CELL,-62,-62,1,7,12,12)
CONPOR(0.0,CELL,-63,-63,1,6,12,12)
CONPOR(0.0,CELL,-64,-64,1,5,12,12)
CONPOR(0.0,CELL,-1,-1,1,5,17,17)
CONPOR(0.0,CELL,-2,-2,1,6,17,17)
CONPOR(0.0,CELL,-3,-3,1,7,17,17)
CONPOR(0.0,CELL,-4,-4,1,8,17,17)
CONPOR(0.0,CELL,-5,-5,1,9,17,17)
CONPOR(0.0,CELL,-6,-6,1,10,17,17)
CONPOR(0.0,CELL,-7,-7,1,11,17,17)
CONPOR(0.0,CELL,-8,-8,1,12,17,17)
CONPOR(0.0,CELL,-9,-9,1,13,17,17)
CONPOR(0.0,CELL,-10,-10,1,14,17,17)
CONPOR(0.0,CELL,-11,-11,1,15,17,17)
   ***********************************
CONPOR(0.0,CELL,-12,-12,1,16,17,17)
CONPOR(0.0,CELL,-13,-13,1,17,17,17)
CONPOR(0.0,CELL,-14,-14,1,18,17,17)
   ***********************************
CONPOR(0.0,CELL,-15,-15,1,18,17,17)
CONPOR(0.0,CELL,-16,-16,1,18,17,17)
CONPOR(0.0,CELL,-17,-17,1,18,17,17)
CONPOR(0.0,CELL,-18,-18,1,18,17,17)
CONPOR(0.0,CELL,-19,-19,1,18,17,17)
CONPOR(0.0,CELL,-20,-20,1,17,17,17)
CONPOR(0.0,CELL,-21,-21,1,16,17,17)
CONPOR(0.0,CELL,-22,-22,1,15,17,17)
CONPOR(0.0,CELL,-23,-23,1,14,17,17)
```

```
    ************************************
CONPOR(0.0,CELL,-24,-24,1,13,17,17)
CONPOR(0.0,CELL,-25,-25,1,12,17,17)
CONPOR(0.0,CELL,-26,-26,1,11,17,17)
    ************************************
CONPOR(0.0,CELL,-27,-27,1,10,17,17)
CONPOR(0.0,CELL,-28,-28,1,9,17,17)
CONPOR(0.0,CELL,-29,-29,1,8,17,17)
CONPOR(0.0,CELL,-30,-30,1,7,17,17)
CONPOR(0.0,CELL,-31,-31,1,6,17,17)
CONPOR(0.0,CELL,-32,-32,1,5,17,17)
CONPOR(0.0,CELL,-33,-33,1,5,17,17)
CONPOR(0.0,CELL,-34,-34,1,6,17,17)
CONPOR(0.0,CELL,-35,-35,1,7,17,17)
CONPOR(0.0,CELL,-36,-36,1,8,17,17)
CONPOR(0.0,CELL,-37,-37,1,9,17,17)
CONPOR(0.0,CELL,-38,-38,1,10,17,17)
CONPOR(0.0,CELL,-39,-39,1,11,17,17)
    ************************************
CONPOR(0.0,CELL,-40,-40,1,12,17,17)
CONPOR(0.0,CELL,-41,-41,1,13,17,17)
CONPOR(0.0,CELL,-42,-42,1,14,17,17)
    ************************************
CONPOR(0.0,CELL,-43,-43,1,15,17,17)
CONPOR(0.0,CELL,-44,-44,1,16,17,17)
CONPOR(0.0,CELL,-45,-45,1,17,17,17)
CONPOR(0.0,CELL,-46,-46,1,18,17,17)
CONPOR(0.0,CELL,-47,-47,1,18,17,17)
CONPOR(0.0,CELL,-48,-48,1,18,17,17)
CONPOR(0.0,CELL,-49,-49,1,18,17,17)
CONPOR(0.0,CELL,-50,-50,1,18,17,17)
CONPOR(0.0,CELL,-51,-51,1,18,17,17)
CONPOR(0.0,CELL,-52,-52,1,17,17,17)
CONPOR(0.0,CELL,-53,-53,1,16,17,17)
CONPOR(0.0,CELL,-54,-54,1,15,17,17)
CONPOR(0.0,CELL,-55,-55,1,14,17,17)
    ************************************
CONPOR(0.0,CELL,-56,-56,1,13,17,17)
CONPOR(0.0,CELL,-57,-57,1,12,17,17)
CONPOR(0.0,CELL,-58,-58,1,11,17,17)
    ************************************
CONPOR(0.0,CELL,-59,-59,1,10,17,17)
CONPOR(0.0,CELL,-60,-60,1,9,17,17)
CONPOR(0.0,CELL,-61,-61,1,8,17,17)
CONPOR(0.0,CELL,-62,-62,1,7,17,17)
CONPOR(0.0,CELL,-63,-63,1,6,17,17)
CONPOR(0.0,CELL,-64,-64,1,5,17,17)
    GROUP 12. Convection and diffusion adjustments
    GROUP 13. Boundary conditions and special sources
PATCH(INLET,LOW,1,NX,1,NY,1,1,1,LSTEP)
COVAL(INLET,P1,FIXFLU,RHO1*WIN)
COVAL(INLET,W1,ONLYMS,WIN)
COVAL(INLET,KE,ONLYMS,KEIN)
COVAL(INLET,EP,ONLYMS,EPIN)

PATCH(KESOURCE,PHASEM,1,NX,1,NY,1,1,1,LSTEP)
COVAL(KESOURCE,KE,GRND4,GRND4)
COVAL(KESOURCE,EP,GRND4,GRND4)

PATCH(TURBWL1,NWALL,1,NX,NY,NY,1,NZ,1,LSTEP)
COVAL(TURBWL1,W1,GRND2,0.0)
COVAL(TURBWL1,U1,GRND2,0.0)
COVAL(TURBWL1,KE,GRND2,GRND2)
COVAL(TURBWL1,EP,GRND2,GRND2)
```

```
*********************************************
Use the following ASYM's in steady asymmetric
calculations only.....
*********************************************

PATCH(ASYM1,HWALL,12,12,1,16,11,11,1,1)
COVAL(ASYM1,V1,GRND2,WAS)
COVAL(ASYM1,KE,GRND2,GRND2)
COVAL(ASYM1,EP,GRND2,GRND2)
PATCH(ASYM2,HWALL,13,13,1,17,11,11,1,1)
COVAL(ASYM2,V1,GRND2,WAS)
COVAL(ASYM2,KE,GRND2,GRND2)
COVAL(ASYM2,EP,GRND2,GRND2)
PATCH(ASYM3,HWALL,14,14,1,18,11,11,1,1)
COVAL(ASYM3,V1,GRND2,WAS)
COVAL(ASYM3,KE,GRND2,GRND2)
COVAL(ASYM3,EP,GRND2,GRND2)

PATCH(ASYM4,HWALL,24,24,1,13,11,11,1,1)
COVAL(ASYM4,V1,GRND2,-WAS)
COVAL(ASYM4,KE,GRND2,GRND2)
COVAL(ASYM4,EP,GRND2,GRND2)
PATCH(ASYM5,HWALL,25,25,1,12,11,11,1,1)
COVAL(ASYM5,V1,GRND2,-WAS)
COVAL(ASYM5,KE,GRND2,GRND2)
COVAL(ASYM5,EP,GRND2,GRND2)
PATCH(ASYM6,HWALL,26,26,1,11,11,11,1,1)
COVAL(ASYM6,V1,GRND2,-WAS)
COVAL(ASYM6,KE,GRND2,GRND2)
COVAL(ASYM6,EP,GRND2,GRND2)

PATCH(ASYM7,HWALL,40,40,1,12,11,11,1,1)
COVAL(ASYM7,V1,GRND2,WAS)
COVAL(ASYM7,KE,GRND2,GRND2)
COVAL(ASYM7,EP,GRND2,GRND2)
PATCH(ASYM8,HWALL,41,41,1,13,11,11,1,1)
COVAL(ASYM8,V1,GRND2,WAS)
COVAL(ASYM8,KE,GRND2,GRND2)
COVAL(ASYM8,EP,GRND2,GRND2)
PATCH(ASYM9,HWALL,42,42,1,14,11,11,1,1)
COVAL(ASYM9,V1,GRND2,WAS)
COVAL(ASYM9,KE,GRND2,GRND2)
COVAL(ASYM9,EP,GRND2,GRND2)

PATCH(A1SYM,HWALL,56,56,1,13,11,11,1,1)
COVAL(A1SYM,V1,GRND2,WAS)
COVAL(A1SYM,KE,GRND2,GRND2)
COVAL(A1SYM,EP,GRND2,GRND2)
PATCH(A2SYM,HWALL,57,57,1,12,11,11,1,1)
COVAL(A2SYM,V1,GRND2,WAS)
COVAL(A2SYM,KE,GRND2,GRND2)
COVAL(A2SYM,EP,GRND2,GRND2)
PATCH(A3SYM,HWALL,58,58,1,11,11,11,1,1)
```

```
     COVAL(A3SYM,V1,GRND2,WAS)
     COVAL(A3SYM,KE,GRND2,GRND2)
     COVAL(A3SYM,EP,GRND2,GRND2)

     PATCH(A4SYM,LWALL,12,12,1,16,18,18,1,1)
     COVAL(A4SYM,V1,GRND2,WAS)
     COVAL(A4SYM,KE,GRND2,GRND2)
     COVAL(A4SYM,EP,GRND2,GRND2)
     PATCH(A5SYM,LWALL,13,13,1,17,18,18,1,1)
     COVAL(A5SYM,V1,GRND2,WAS)
     COVAL(A5SYM,KE,GRND2,GRND2)
     COVAL(A5SYM,EP,GRND2,GRND2)
     PATCH(A6SYM,LWALL,14,14,1,18,18,18,1,1)
     COVAL(A6SYM,V1,GRND2,WAS)
     COVAL(A6SYM,KE,GRND2,GRND2)
     COVAL(A6SYM,EP,GRND2,GRND2)

     PATCH(A7SYM,LWALL,24,24,1,13,18,18,1,1)
     COVAL(A7SYM,V1,GRND2,-WAS)
     COVAL(A7SYM,KE,GRND2,GRND2)
     COVAL(A7SYM,EP,GRND2,GRND2)
     PATCH(A8SYM,LWALL,25,25,1,12,18,18,1,1)
     COVAL(A8SYM,V1,GRND2,-WAS)
     COVAL(A8SYM,KE,GRND2,GRND2)
     COVAL(A8SYM,EP,GRND2,GRND2)
     PATCH(A9SYM,LWALL,26,26,1,11,18,18,1,1)
     COVAL(A9SYM,V1,GRND2,-WAS)
     COVAL(A9SYM,KE,GRND2,GRND2)
     COVAL(A9SYM,EP,GRND2,GRND2)

     PATCH(ASY1,LWALL,40,40,1,12,18,18,1,1)
     COVAL(ASY1,V1,GRND2,WAS)
     COVAL(ASY1,KE,GRND2,GRND2)
     COVAL(ASY1,EP,GRND2,GRND2)
     PATCH(ASY2,LWALL,41,41,1,13,18,18,1,1)
     COVAL(ASY2,V1,GRND2,WAS)
     COVAL(ASY2,KE,GRND2,GRND2)
     COVAL(ASY2,EP,GRND2,GRND2)
     PATCH(ASY3,LWALL,42,42,1,14,18,18,1,1)
     COVAL(ASY3,V1,GRND2,WAS)
     COVAL(ASY3,KE,GRND2,GRND2)
     COVAL(ASY3,EP,GRND2,GRND2)

     PATCH(ASY4,LWALL,56,56,1,13,18,18,1,1)
     COVAL(ASY4,V1,GRND2,WAS)
     COVAL(ASY4,KE,GRND2,GRND2)
     COVAL(ASY4,EP,GRND2,GRND2)
     PATCH(ASY5,LWALL,57,57,1,12,18,18,1,1)
     COVAL(ASY5,V1,GRND2,WAS)
     COVAL(ASY5,KE,GRND2,GRND2)
     COVAL(ASY5,EP,GRND2,GRND2)
     PATCH(ASY6,LWALL,58,58,1,11,18,18,1,1)
     COVAL(ASY6,V1,GRND2,WAS)
     COVAL(ASY6,KE,GRND2,GRND2)
     COVAL(ASY6,EP,GRND2,GRND2)

      *** Replace FIXVAL (too tight a restriction) ***

FIXVAP=WIN*RHO1*1000.

PATCH(OUTLET,HIGH,1,NX,1,NY,NZ,NZ,1,LSTEP)
COVAL(OUTLET,P1,FIXVAP,0.0)

    GROUP 15. Termination of sweeps
LSWEEP=10
RESREF(P1)=((PI/4.0)*DIA**2.0)*WIN/1000.
```

```
RESREF(U1)=RHO1*((PI/4.0)*DIA**2.0)*WIN**2/1000.
RESREF(V1)=RHO1*((PI/4.0)*DIA**2.0)*WIN**2/1000.
RESREF(W1)=RHO1*((PI/4.0)*DIA**2.0)*WIN**2/1000.
RESREF(KE)=RHO1*((PI/4.0)*DIA**2.0)*WIN*KEIN/1000.
RESREF(EP)=RHO1*((PI/4.0)*DIA**2.0)*WIN*EPIN/1000.
    GROUP 16. Termination of iterations
    GROUP 17. Under-relaxation devices
KELIN=1
RELAX(P1,LINRLX,0.4)
RELAX(U1,FALSDT,0.05)
RELAX(V1,FALSDT,0.05)
RELAX(W1,FALSDT,0.05)
RELAX(KE,LINRLX,0.3)
RELAX(EP,LINRLX,0.3)
    GROUP 18. Limits on variables or increments to them
    GROUP 19. Data communicated by satellite to GROUND
GENK=T



RG(1)=RI
RG(2)=PI
RG(3)=DIA/2.0
    GROUP 20. Preliminary print-out
    GROUP 21. Print-out of variables
OUTPUT(P1,N,N,Y,Y,Y,Y)
OUTPUT(U1,N,N,Y,Y,Y,Y)
OUTPUT(V1,N,N,Y,Y,Y,Y)
OUTPUT(W1,N,N,Y,Y,Y,Y)
OUTPUT(KE,N,N,Y,Y,Y,Y)
OUTPUT(EP,N,N,Y,Y,Y,Y)
OUTPUT(NPOR,N,N,P,P,P,P)
OUTPUT(HPOR,N,N,P,P,P,P)
OUTPUT(VPOR,N,N,P,P,P,P)
OUTPUT(EPOR,N,N,P,P,P,P)
    GROUP 22. Spot-value print-out
IXMON=1
IYMON=8
IZMON=29
NPLT=1
    GROUP 23. Field print-out and plot control
PATCH(MIDLP,PROFIL,1,1,8,8,25,25,1,LSTEP)
PLOT(MIDLP,P1,0.0,0.0)
PATCH(MIDLW,PROFIL,1,1,8,8,25,25,1,LSTEP)
PLOT(MIDLW,W1,0.0,0.0)
PATCH(MIDLU,PROFIL,1,1,8,8,25,25,1,LSTEP)
PLOT(MIDLU,U1,0.0,0.0)
PATCH(MIDLV,PROFIL,1,1,8,8,25,25,1,LSTEP)
PLOT(MIDLV,V1,0.0,0.0)
   ****************************************
PATCH(BOTMP,PROFIL,33,33,8,8,25,25,1,LSTEP)
PLOT(BOTMP,P1,0.0,0.0)
PATCH(BOTMW,PROFIL,33,33,8,8,25,25,1,LSTEP)
PLOT(BOTMW,W1,0.0,0.0)
PATCH(BOTMU,PROFIL,33,33,8,8,25,25,1,LSTEP)
PLOT(BOTMU,U1,0.0,0.0)
PATCH(BOTMV,PROFIL,33,33,8,8,25,25,1,LSTEP)
PLOT(BOTMV,V1,0.0,0.0)
   *****************************************
PATCH(BOTLP,PROFIL,25,25,13,13,25,25,1,LSTEP)
PLOT(BOTLP,P1,0.0,0.0)
PATCH(BOTLW,PROFIL,25,25,13,13,25,25,1,LSTEP)
PLOT(BOTLW,W1,0.0,0.0)
PATCH(BOTLU,PROFIL,25,25,13,13,25,25,1,LSTEP)
PLOT(BOTLU,U1,0.0,0.0)
```

```
PATCH(BOTLV,PROFIL,25,25,13,13,25,25,1,LSTEP)
PLOT(BOTLV,V1,0.0,0.0)
    *************************************
PATCH(MIDL1P,PROFIL,1,1,8,8,30,30,1,LSTEP)
PLOT(MIDL1P,P1,0.0,0.0)
PATCH(MIDL1W,PROFIL,1,1,8,8,30,30,1,LSTEP)
PLOT(MIDL1W,W1,0.0,0.0)
PATCH(MIDL1U,PROFIL,1,1,8,8,30,30,1,LSTEP)
PLOT(MIDL1U,U1,0.0,0.0)
PATCH(MIDL1V,PROFIL,1,1,8,8,30,30,1,LSTEP)
PLOT(MIDL1V,V1,0.0,0.0)
    *************************************
PATCH(BOTM1P,PROFIL,33,33,8,8,30,30,1,LSTEP)
PLOT(BOTM1P,P1,0.0,0.0)
PATCH(BOTM1W,PROFIL,33,33,8,8,30,30,1,LSTEP)
PLOT(BOTM1W,W1,0.0,0.0)
PATCH(BOTM1U,PROFIL,33,33,8,8,30,30,1,LSTEP)
PLOT(BOTM1U,U1,0.0,0.0)
PATCH(BOTM1V,PROFIL,33,33,8,8,30,30,1,LSTEP)
PLOT(BOTM1V,V1,0.0,0.0)
    *************************************
PATCH(BOTL1P,PROFIL,25,25,13,13,30,30,1,LSTEP)
PLOT(BOTL1P,P1,0.0,0.0)
PATCH(BOTL1W,PROFIL,25,25,13,13,30,30,1,LSTEP)
PLOT(BOTL1W,W1,0.0,0.0)
PATCH(BOTL1U,PROFIL,25,25,13,13,30,30,1,LSTEP)
PLOT(BOTL1U,U1,0.0,0.0)
PATCH(BOTL1V,PROFIL,25,25,13,13,30,30,1,LSTEP)
PLOT(BOTL1V,V1,0.0,0.0)
IPROF=3
    GROUP 24. Dumps for restarts
    TSTSWP=-1
STOP
```

APPENDIX K

Q1 FILE FOR THREE-DIMENSIONAL SCALAR CONVECTION TEST PROBLEM

```
TALK=T;RUN(1,1);VDU=X11-TERM
    GROUP 1. Run title and other preliminaries
TEXT(SUCCA3D TEST FLOW)
REAL(REYNOS,CELLDIM,UIN,VIN,WIN)
REYNOS=200
CELLDIM=1.0
UIN=1.0
VIN=1.0
WIN=1.0
    GROUP 2. Transience; time-step specification
    GROUP 3. X-direction grid specification
GRDPWR(X,11,11.0,1.)
    GROUP 4. Y-direction grid specification
GRDPWR(Y,11,11.0,1.)
    GROUP 5. Z-direction grid specification
GRDPWR(Z,11,11.0,1.)
    GROUP 6. Body-fitted coordinates or grid distortion
    GROUP 7. Variables stored, solved & named
   SOLUTN(C1,Y,Y,Y,N,N,N)
SOLUTN(P1,Y,Y,Y,N,N,N)
SOLVE(U1,V1,W1,C1)
    GROUP 8. Terms (in differential equations) & devices
UCONV=T
TERMS(C1,N,Y,N,N,Y,N)
    GROUP 9. Properties of the medium (or media)
RHO1=1.0
ENUL=(CELLDIM*UIN)/REYNOS
    GROUP 10. Inter-phase-transfer processes and properties
    GROUP 11. Initialization of variable or porosity fields
    GROUP 12. Convection and diffusion adjustments
    GROUP 13. Boundary conditions and special sources

PATCH(BOTTOM,SOUTH,1,NX,1,1,1,NZ,1,1)
COVAL(BOTTOM,P1,FIXFLU,RHO1*VIN)
COVAL(BOTTOM,U1,ONLYMS,UIN)
COVAL(BOTTOM,V1,ONLYMS,VIN)
COVAL(BOTTOM,W1,ONLYMS,WIN)
COVAL(BOTTOM,C1,ONLYMS,1.0)

PATCH(SIDE1,WEST,1,1,1,NY,1,NZ,1,1)
COVAL(SIDE1,P1,FIXFLU,RHO1*UIN)
COVAL(SIDE1,U1,ONLYMS,UIN)
COVAL(SIDE1,V1,ONLYMS,VIN)
COVAL(SIDE1,W1,ONLYMS,WIN)
COVAL(SIDE1,C1,ONLYMS,0.0)

PATCH(SIDE2,LOW,1,NX,1,NY,1,1,1,1)
COVAL(SIDE2,P1,FIXFLU,RHO1*WIN)
COVAL(SIDE2,U1,ONLYMS,UIN)
COVAL(SIDE2,V1,ONLYMS,VIN)
COVAL(SIDE2,W1,ONLYMS,WIN)
COVAL(SIDE2,C1,ONLYMS,0.0)

   *******************************************************
   When studying the case where either U or V or W = 0.0
   it is necessary to alter the SUCCA3D application limits
   shown below. For example when W = 0.0 the domain limits
   would be 2,NX,2,NY,1,NZ. This is obvously due to the
   skewed inlet flow b.c.'s and is not required in most
   conventional non-skewed inlet CFD analyses.
   *******************************************************

PATCH(SUCCAC,CELL,2,NX,2,NY,2,NZ,1,LSTEP)
COVAL(SUCCAC,C1,GRND,GRND)

PATCH(SUCCAE,CELL,2,NX,2,NY,2,NZ,1,LSTEP)
```

```
COVAL(SUCCAE,C1,GRND1,GRND1)

PATCH(EOUTLE,EAST,NX,NX,1,NY,1,NZ,1,1)
COVAL(EOUTLE,P1,10000.,0.0)

PATCH(NOUTLE,NORTH,1,NX,NY,NY,1,NZ,1,1)
COVAL(NOUTLE,P1,10000.,0.0)

PATCH(HOUTLE,HIGH,1,NX,1,NY,NZ,NZ,1,1)
COVAL(HOUTLE,P1,10000.,0.0)

    GROUP 14. Downstream pressure for PARAB=.TRUE.
    GROUP 15. Termination of sweeps
LSWEEP=70
RESREF(P1)=1.E-4
RESREF(U1)=1.E-4
RESREF(V1)=1.E-4
RESREF(W1)=1.E-4
RESREF(C1)=1.E-4
    GROUP 16. Termination of iterations
    GROUP 17. Under-relaxation devices
RELAX(P1,LINRLX,0.4)
RELAX(U1,FALSDT,0.2)
RELAX(V1,FALSDT,0.2)
RELAX(W1,FALSDT,0.2)
RELAX(C1,FALSDT,0.2)
    GROUP 18. Limits on variables or increments to them
    GROUP 19. Data communicated by satellite to GROUND
    solve for U1 via succa3d?
LG(1)=F
    solve for V1 via succa3d?
LG(2)=F
    solve for W1 via succa3d?
LG(3)=F
    solve for Scalars via succa3d?
LG(4)=T
    GROUP 20. Preliminary print-out
    GROUP 21. Print-out of variables
    GROUP 22. Spot-value print-out
IXMON=6
IYMON=3
IZMON=7
NPLT=1
    GROUP 23. Field print-out and plot control
    GROUP 24. Dumps for restarts
TSTSWP=-1
STOP
```

APPENDIX K1

Q1 FILE FOR 3D CONVECTION OF A SCALAR OVER A FENCE TYPE BLOCKAGE

```
TALK=T;RUN(1,1);VDU=X11-TERM
    GROUP 1. Run title and other preliminaries
TEXT(3D FLOW OVER A FENCE)
REAL(REYNOS,WIN,CHRDIM)
WIN=1.0
CHRDIM=0.2
REYNOS=350.
    GROUP 2. Transience; time-step specification
    GROUP 3. X-direction grid specification
GRDPWR(X,8,0.2,1.)
    GROUP 4. Y-direction grid specification
GRDPWR(Y,8,0.2,1.)
    GROUP 5. Z-direction grid specification
NREGZ=4
IREGZ=1;GRDPWR(Z,10,0.3,-1.7)
IREGZ=2;GRDPWR(Z,20,0.7,1.7)
IREGZ=3;GRDPWR(Z,8,0.5,1.0)
IREGZ=4;GRDPWR(Z,4,0.9,1.7)
    GROUP 6. Body-fitted coordinates or grid distortion
    GROUP 7. Variables stored, solved & named
SOLVE(P1,U1,V1,W1,C1)
    GROUP 8. Terms (in differential equations) & devices
TERMS(C1,Y,Y,Y,N,N,N)
UCONV=T
    GROUP 9. Properties of the medium (or media)
RHO1=1.0
ENUL=(WIN*CHRDIM)/REYNOS
    GROUP 10. Inter-phase-transfer processes and properties
    GROUP 11. Initialization of variable or porosity fields
CONPOR(BLK1,0.0,HIGH,4,5,1,4,9,9)
    GROUP 12. Convection and diffusion adjustments
    GROUP 13. Boundary conditions and special sources

PATCH(INLET,LOW,1,NX,1,NY,1,1,1,LSTEP)
COVAL(INLET,P1,FIXFLU,RHO1*WIN)
COVAL(INLET,W1,ONLYMS,WIN)

PATCH(CINL,SOUTH,4,5,1,1,4,4,1,LSTEP)
COVAL(CINL,P1,FIXFLU,2.0*RHO1*WIN)
COVAL(CINL,C1,ONLYMS,1.0)

PATCH(SUCCAC,CELL,1,NX,1,NY,1,NZ,1,LSTEP)
COVAL(SUCCAC,C1,GRND,GRND)

PATCH(SUCCAE,CELL,1,NX,1,NY,1,NZ,1,LSTEP)
COVAL(SUCCAE,C1,GRND1,GRND1)

PATCH(OUTLET,HIGH,1,NX,1,NY,NZ,NZ,1,LSTEP)
COVAL(OUTLET,P1,10000.,0.0)

    GROUP 14. Downstream pressure for PARAB=.TRUE.
    GROUP 15. Termination of sweeps
LSWEEP=70
RESREF(P1)=1.E-6
RESREF(U1)=1.E-6
RESREF(V1)=1.E-6
RESREF(W1)=1.E-6
RESREF(C1)=1.E-6
    GROUP 16. Termination of iterations
    GROUP 17. Under-relaxation devices
RELAX(P1,LINRLX,0.3)
RELAX(U1,FALSDT,0.2)
RELAX(V1,FALSDT,0.2)
RELAX(W1,FALSDT,0.2)
RELAX(C1,FALSDT,0.1)
    GROUP 18. Limits on variables or increments to them
```

```
    GROUP 19. Data communicated by satellite to GROUND
LG(1)=F
LG(2)=F
LG(3)=F
LG(4)=T
    GROUP 20. Preliminary print-out
    GROUP 21. Print-out of variables
    GROUP 22. Spot-value print-out
IXMON=2
IYMON=3
IZMON=12
NPLT=1
    GROUP 23. Field print-out and plot control
    GROUP 24. Dumps for restarts
    RESTRT(ALL)
TSTSWP=-1
STOP
```

APPENDIX K2

Q1 FILE FOR 3D FLOW OVER A FENCE TYPE BLOCKAGE

```
TALK=T;RUN(1,1);VDU=X11-TERM
    GROUP 1. Run title and other preliminaries
TEXT(3D FLOW OVER A FENCE)
REAL(REYNOS,WIN,CHRDIM)
WIN=1.0
CHRDIM=0.2
REYNOS=35.
    GROUP 2. Transience; time-step specification
    GROUP 3. X-direction grid specification
GRDPWR(X,8,0.2,1.)
    GROUP 4. Y-direction grid specification
GRDPWR(Y,8,0.2,1.)
    GROUP 5. Z-direction grid specification
NREGZ=4
IREGZ=1;GRDPWR(Z,10,0.3,-1.7)
IREGZ=2;GRDPWR(Z,20,0.7,1.7)
IREGZ=3;GRDPWR(Z,8,0.5,1.0)
IREGZ=4;GRDPWR(Z,4,0.9,1.7)
    GROUP 6. Body-fitted coordinates or grid distortion
    GROUP 7. Variables stored, solved & named
SOLVE(P1,U1,V1,W1)
    GROUP 8. Terms (in differential equations) & devices
UCONV=T
    GROUP 9. Properties of the medium (or media)
RHO1=1.0
ENUL=(WIN*CHRDIM)/REYNOS
    GROUP 10. Inter-phase-transfer processes and properties
    GROUP 11. Initialization of variable or porosity fields
CONPOR(BLK1,0.0,HIGH,4,5,1,4,9,9)
    GROUP 12. Convection and diffusion adjustments
    GROUP 13. Boundary conditions and special sources

PATCH(INLET,LOW,1,NX,1,NY,1,1,1,1)
COVAL(INLET,P1,FIXFLU,RHO1*WIN)
COVAL(INLET,W1,ONLYMS,WIN)

PATCH(SUCCAC,CELL,1,NX,1,NY,2,NZ,1,LSTEP)
COVAL(SUCCAC,U1,GRND,GRND)
COVAL(SUCCAC,V1,GRND,GRND)
COVAL(SUCCAC,W1,GRND,GRND)

PATCH(SUCCAE,CELL,1,NX,1,NY,2,NZ,1,LSTEP)
COVAL(SUCCAE,U1,GRND1,GRND1)
COVAL(SUCCAE,V1,GRND1,GRND1)
COVAL(SUCCAE,W1,GRND1,GRND1)

PATCH(OUTLET,HIGH,1,NX,1,NY,NZ,NZ,1,1)
COVAL(OUTLET,P1,10000.,0.0)

    GROUP 14. Downstream pressure for PARAB=.TRUE.
    GROUP 15. Termination of sweeps
LSWEEP=70
RESREF(P1)=1.E-6
RESREF(U1)=1.E-6
RESREF(V1)=1.E-6
RESREF(W1)=1.E-6
    GROUP 16. Termination of iterations
    GROUP 17. Under-relaxation devices
RELAX(P1,LINRLX,0.5)
RELAX(U1,FALSDT,0.2)
RELAX(V1,FALSDT,0.2)
RELAX(W1,FALSDT,0.2)
    GROUP 18. Limits on variables or increments to them
    GROUP 19. Data communicated by satellite to GROUND
LG(1)=T
LG(2)=T
```

```
LG(3)=T
LG(4)=F
    GROUP 20. Preliminary print-out
    GROUP 21. Print-out of variables
    GROUP 22. Spot-value print-out
IXMON=2
IYMON=3
NPLT=1
    GROUP 23. Field print-out and plot control
    GROUP 24. Dumps for restarts
    RESTRT(ALL)
TSTSWP=-1
STOP
```

APPENDIX K3

COMPILATION AND LINKING PROCEDURE FOR SUCCA3D FORTRAN CODING

With regard to the compilation and linking of the SUCCA3D source code within the UNIX environment, an alternative method is employed in comparison with the direct compilation and linking of the SUCCA2D code. Since the SUCCA2D scheme has been formulated using a single piece of coding it is possible to compile and link this coding directly to create an executable file. However, the length of the SUCCA3D program necessitates that the coding be segmented into appropriate subroutines. In total there are six separate blocks of coding and each block must be compiled separately such that six object modules are obtained. Following this compilation procedure all of the object modules are linked sequentially using a single link command to produce a single executable file.

APPENDIX L

GROUND FORTRAN CODING FOR SUCCA2D SCHEME

```
C FILE NAME GROUND.FTN--------------------------------120691
C   THIS IS THE MAIN PROGRAM OF EARTH
C
C   (C) COPYRIGHT 1984, LAST REVISION 1991
C   CONCENTRATION HEAT AND MOMENTUM LTD. ALL RIGHTS RESERVED.
C   This subroutine and the remainder of the PHOENICS code are
C   proprietary software owned by Concentration Heat and Momentum
C   Limited, 40 High Street, Wimbledon, London SW19 5AU, England.
C
C
C     PROGRAM MAIN
C
C 1   The following  COMMON's, which appear identically in the
C     satellite MAIN program, allow up to 50 dependent variables to
C     be solved for (or their storage spaces to be occupied by
C     other variables, such as density). If a larger number is
C     required, the PARAMETER NUMPHI should be reset to the required
C     larger number. Numbers less than 50 are not permitted.
C
      PARAMETER (NUMPHI=50,   NM=NUMPHI,NM4=NM*4)
C
      COMMON/LGE4/L4(NM)
     1/LDB1/L5(NM)/IDA1/I1(NM)/IDA2/I2(NM)/IDA3/I3(NM)/IDA4/I4(NM)
     1/IDA5/I5(NM)/IDA6/I6(NM)/GI1/I7(NM)/GI2/I8(NM)/HDA1/IH1(NM)
     1/GH1/IH2(NM)/RDA1/R1(NM)/RDA2/R2(NM)/RDA3/R3(NM)/RDA4/R4(NM)
     1/RDA5/R5(NM)/RDA6/R6(NM)/RDA7/R7(NM)/RDA8/R8(NM)/RDA9/R9(NM)
     1/RDA10/R10(NM)/RDA11/R11(NM)
     1/GR1/R12(NM)/GR2/R13(NM)/GR3/R14(NM)/GR4/R15(NM)
     1/IPIP1/IP1(NM)/HPIP2/IHP2(NM)/RPIP1/RVAL(NM)/LPIP1/LVAL(NM)
     1/IFPL/IPL0(NM)/RFPL1/ORPRIN(NM)/RFPL2/ORMAX(NM)
     1/RFPL3/ORMIN(NM)/IDA7/ID7(NM)/IDA8/ID8(NM)
      LOGICAL L1,L2,L3,L4,L5,DBGFIL,LVAL
      CHARACTER*4 IH1,IH2,IHP2,NSDA
C
      COMMON/F01/I9(NM4)
      COMMON/DISC/DBGFIL
      COMMON/LUNITS/LUNIT(60)
C
      EXTERNAL WAYOUT
C
C 2   Set dimensions of data-for-GROUND arrays here. WARNING: the
C     corresponding arrays in the MAIN program of the satellite
C     (see SATLIT) must have the same dimensions.
      PARAMETER (NLG=20, NIG=20, NRG=100, NCG=10)
C
      COMMON/LGRND/LG(NLG)/IGRND/IG(NIG)/RGRND/RG(NRG)/CGRND/CG(NCG)
      LOGICAL LG
      CHARACTER*4 CG
      COMMON/GUGRND/GCUN(60,80),GCUS(60,80),GCUE(60,80),GCUW(60,80),
     1              GCVN(60,80),GCVS(60,80),GCVE(60,80),GCVW(60,80),
     2              GCSN(60,80),GCSS(60,80),GCSE(60,80),GCSW(60,80),
     3              GSPCO(60,80),GUPCO(60,80),GVPCO(60,80)
      COMMON/IUGRND/ISPVAL(60,80),IUPVAL(60,80),IVPVAL(60,80)
C
C 3   Set dimensions of data-for-GREX arrays here. WARNING: the
C     corresponding arrays in the MAIN program of the satellite
C     (see SATLIT) must have the same dimensions.
      PARAMETER(NLSG=20, NISG=20, NRSG=100,NCSG=10)
C
      COMMON/LSG/LSGD(NLSG)/ISG/ISGD(NISG)/RSG/RSGD(NRSG)/CSG/CSGD(NCSG)
      LOGICAL LSGD
      CHARACTER*4 CSGD
C
C 4   Set dimension of patch-name array here. WARNING: the array
C     NAMPAT in the MAIN program of the satellite must have the
C     same dimension.
```

```
      PARAMETER (NPNAM=200)
C
      COMMON/NPAT/NAMPAT(NPNAM)
      COMMON/LWFUN1/DOSKIN(NPNAM)
      COMMON/LWFUN2/DHCHKD(NPNAM)
      CHARACTER*8 NAMPAT
      LOGICAL DOSKIN,DHCHKD
C
C     CONFIG FILE name declaration.
      COMMON/CNFG/CNFIG
      CHARACTER CNFIG*48
C
C 5   The numbers in the next statement indicates how much computer
C     memory is to be set aside for storing the main and auxiliary
C     variables. The user may alter them if he wishes, to accord
C     with the number of grid nodes and dependent variables he is
C     concerned with.
      PARAMETER (NFDIM=600000)
C
      COMMON F(NFDIM)
C
C 6   The following three statements concern storage for the PATCH-wise
C     variables. If more than 30 PATCH-wise variables are required
C     NPVDM should be increased and the common block /LBPV/ in the
C     include file GRDLOC15 should be lengthened.
      PARAMETER (NPVDM=30)
      COMMON/INDPV/NPVMX,NIMAX,NITOT,LOPV(NPVDM)
C
      CALL SUB2(NPVMX,NPVDM,NIMAX,NPVDM)
C
      CALL CNFGZZ(2)
      CALL EARSET(1)
      CALL OPENFL(6)
C
      CALL MAIN1(NFDIM,NUMPHI,NLSG,NISG,NRSG,NCSG,NLG,NIG,NRG,NCG)
      CALL WAYOUT(0)
      STOP
      END
C*******************************************************************
C$DIR**GROSTA
      SUBROUTINE GROSTA
      INCLUDE '1p16/d_earth/SATEAR'
      INCLUDE '1p16/d_earth/GRDLOC'
      INCLUDE '1p16/d_earth/GRDEAR'
C
C.... This subroutine acts as a junction-box, directing control to
C     the GROUNDs selected by the SATELLITE settings of USEGRX,
C     NAMGRD & USEGRD.
C
C     Subroutine GREX3 contains options for fluid properties,
C     turbulence models, wall functions, chemical reaction etc.
C
      IF(USEGRX) CALL GREX3
C
C.... SPECGR, SPC1GR, SPC2GR and SPC3GR are names which the user may
C     give to "special GROUNDs" of his own.
C
      IF(NAMGRD.NE.'NONE') THEN
        IF(NAMGRD.EQ.'SPEC') THEN
          CALL SPECGR
        ELSE IF(NAMGRD.EQ.'SPC1') THEN
          CALL SPC1GR
        ELSE IF(NAMGRD.EQ.'SPC2') THEN
          CALL SPC2GR
        ELSE IF(NAMGRD.EQ.'SPC3') THEN
          CALL SPC3GR
```

```
C
        ELSE
          CALL WRITBL
          CALL WRITST
          CALL WRIT40('NAMGRD set but no CALL made, ie.        ')
          CALL WRIT1A('NAMGRD  ',NAMGRD)
          CALL WRIT40(' Permissible calls for this GROSTA are:-')
          CALL WRIT40('SPEC, SPC1, SPC2, SPC3,                 ')
          CALL WRIT40('Use upper case only!!                   ')
          CALL WRITST
          CALL WRITBL
          CALL WAYOUT(2)
        ENDIF
      ENDIF
C
C.... The subroutine GROUND attached to the bottom of this file is
C     an unallocated blank form into which the user can insert his
C     own FORTRAN sequences. The PIL parameter USEGRD governs entry
C     to it.
C
      IF(USEGRD) CALL GROUND
C
C.... The data "echo" is called at the preliminary print-out stage.
      IF(IGR.EQ.20) THEN
        IF(ECHO) THEN
          CALL DATPRN(Y,Y,Y,Y,  Y,Y,Y,Y,  Y,Y,Y,N,  Y,Y,Y,Y,
     1                Y,Y,Y,Y,  Y,Y,Y,Y)
        ELSE
          CALL DATPRN(Y,N,N,N,N,N,N,N,N,N,N,N,N,N,N,N,N,N,N,N,N,N,N,N)
        ENDIF
      ENDIF
      END
C***************************************************************
      SUBROUTINE GROUND
      INCLUDE 'lp16/d_earth/SATEAR'
      INCLUDE 'lp16/d_earth/GRDLOC'
      INCLUDE 'lp16/d_earth/GRDEAR'
      INCLUDE 'lp16/d_earth/GRDBFC'
CXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX USER SECTION STARTS:
C
C 1   Set dimensions of data-for-GROUND arrays here. WARNING: the
C     corresponding arrays in the MAIN program of the satellite
C     and EARTH must have the same dimensions.
      PARAMETER (NLG=20, NIG=20, NRG=100, NCG=10)
C
      COMMON/LGRND/LG(NLG)/IGRND/IG(NIG)/RGRND/RG(NRG)/CGRND/CG(NCG)
      LOGICAL LG
      CHARACTER*4 CG
      COMMON/GUGRND/GCUN(60,80),GCUS(60,80),GCUE(60,80),GCUW(60,80),
     1              GCVN(60,80),GCVS(60,80),GCVE(60,80),GCVW(60,80),
     2              GCSN(60,80),GCSS(60,80),GCSE(60,80),GCSW(60,80),
     3              GSPCO(60,80),GUPCO(60,80),GVPCO(60,80)
      COMMON/IUGRND/ISPVAL(60,80),IUPVAL(60,80),IVPVAL(60,80)
C
C 2   User dimensions own arrays here, for example:
C     DIMENSION GUH(10,10),GUC(10,10),GUX(10,10),GUZ(10)
      DIMENSION GCC(60,80),GTMP(60,80)
C
C 3   User places his data statements here, for example:
      DATA NXDIM,NYDIM/80,60/
C
C 4   Insert own coding below as desired, guided by GREX examples.
C     Note that the satellite-to-GREX special data in the labelled
C     COMMONs /RSG/, /ISG/, /LSG/ and /CSG/ can be included and
C     used below but the user must check GREX for any conflicting
C     uses. The same comment applies to the EARTH-spare working
```

```
C     arrays EASP1, EASP2,....EASP20. In addition to the EASPs,
C     there are 10 GRound-earth SPare arrays, GRSP1,...,GRSP10,
C     supplied solely for the user, which are not used by GREX. If
C     the call to GREX has been deactivated then all of the arrays
C     may be used without reservation.
C
      IXL=IABS(IXL)
      IF(IGR.EQ.13) GO TO 13
      IF(IGR.EQ.19) GO TO 19
      GO TO (1,2,3,4,5,6,25,8,9,10,11,12,13,14,25,25,25,25,19,20,25,
     125,23,24),IGR
   25 CONTINUE
      RETURN
C******************************************************************
C
C--- GROUP 1. Run title and other preliminaries
C
    1 GO TO (1001,1002),ISC
 1001 CONTINUE
C
C     User may here change message transmitted to the VDU screen or
C     batch-run log file.
      IF(IGR.EQ.1.AND.ISC.EQ.1) THEN
        CALL WRYT40('GROUND file is GROUND.FTN of:    120691 ')
      ENDIF
C
      RETURN
 1002 CONTINUE
      DO 1010 IX=1,NX
      DO 1010 IY=1,NY
       ISPVAL(IY,IX)=0
       IUPVAL(IY,IX)=0
       IVPVAL(IY,IX)=0
       GUPCO(IY,IX)=0.
       GVPCO(IY,IX)=0.
       GSPCO(IY,IX)=0.
 1010 CONTINUE
      RETURN
C******************************************************************
C
C--- GROUP 2. Transience; time-step specification
C
    2 CONTINUE
      RETURN
C******************************************************************
C
C--- GROUP 3. X-direction grid specification
C
    3 CONTINUE
      RETURN
C******************************************************************
C
C--- GROUP 4. Y-direction grid specification
C
    4 CONTINUE
      RETURN
C******************************************************************
C
C--- GROUP 5. Z-direction grid specification
C
    5 CONTINUE
      RETURN
C******************************************************************
C
C--- GROUP 6. Body-fitted coordinates or grid distortion
C
```

```
    6 CONTINUE
      RETURN
C*****************************************************************
C   * Make changes for this group only in group 19.
C--- GROUP 7. Variables stored, solved & named
C*****************************************************************
C
C--- GROUP 8. Terms (in differential equations) & devices
C
    8 GO TO (81,82,83,84,85,86,87,88,89,810,811,812,813,814,815)
     1,ISC
   81 CONTINUE
C   * ------------------ SECTION  1 --------------------------
C    For U1AD.LE.GRND--- phase 1 additional velocity. Index VELAD
      RETURN
   82 CONTINUE
C   * ------------------ SECTION  2 --------------------------
C    For U2AD.LE.GRND--- phase 2 additional velocity. Index VELAD
      RETURN
   83 CONTINUE
C   * ------------------ SECTION  3 --------------------------
C    For V1AD.LE.GRND--- phase 1 additional velocity. Index VELAD
      RETURN
   84 CONTINUE
C   * ------------------ SECTION  4 --------------------------
C    For V2AD.LE.GRND--- phase 2 additional velocity. Index VELAD
      RETURN
   85 CONTINUE
C   * ------------------ SECTION  5 --------------------------
C    For W1AD.LE.GRND--- phase 1 additional velocity. Index VELAD
      RETURN
   86 CONTINUE
C   * ------------------ SECTION  6 --------------------------
C    For W2AD.LE.GRND--- phase 2 additional velocity. Index VELAD
      RETURN
   87 CONTINUE
C   * ------------------ SECTION 7 ---- Volumetric source for gala
      RETURN
   88 CONTINUE
C   * ------------------ SECTION 8 ---- Convection fluxes
C      CALL WRIT1I('INDVAR  ',INDVAR)
C      IF(INDVAR.EQ.16) THEN
C       IF(NDIREC.EQ.1) CALL PRN('CONN',LD11)
C       IF(NDIREC.EQ.1) CALL PRN('CONS',LD12)
C       IF(NDIREC.EQ.3) CALL PRN('CONE',LD11)
C       IF(NDIREC.EQ.3) CALL PRN('CONW',LD12)
C      ENDIF
C
C      IF(INDVAR.EQ.3) THEN
C       IF(NDIREC.EQ.1) CALL PRN('CONN',LD11)
C       IF(NDIREC.EQ.1) CALL PRN('CONS',LD12)
C       IF(NDIREC.EQ.3) CALL PRN('CONE',LD11)
C       IF(NDIREC.EQ.3) CALL PRN('CONW',LD12)
C      ENDIF
C
C      IF(INDVAR.EQ.5) THEN
C       IF(NDIREC.EQ.1) CALL PRN('CONN',LD11)
C       IF(NDIREC.EQ.1) CALL PRN('CONS',LD12)
C       IF(NDIREC.EQ.3) CALL PRN('CONE',LD11)
C       IF(NDIREC.EQ.3) CALL PRN('CONW',LD12)
C      ENDIF
C
        IF(.NOT.(LG(1))) GO TO 88100
         IF(INDVAR.NE.3.AND.INDVAR.NE.4) GO TO 88100
C
C  CORRECT COEFFICIENTS FOR U EQUATION
```

```
C   ************************************
C
C   NORTH-SOUTH
C
        IF(NDIREC.NE.1) GO TO 88080
C
        LOLD11=LOF(LD11)
        LOLD12=LOF(LD12)
C
        DO 88052 IX=1,NX
        DO 88052 IY=1,NY
         ICELL=IY+NY*(IX-1)
         GCUN(IY,IX)=F(LOLD11+ICELL)
         GCUS(IY,IX)=F(LOLD12+ICELL)
88052   CONTINUE
        IF(ISWEEP.EQ.FSWEEP) GO TO 88080
C
C   NORTH COEFFICIENT
C
        DO 88060 IX=1,NX-1
        DO 88060 IY=1,NY
         IF(GCUN(IY,IX).GT.0.0.AND.GCUS(IY,IX).EQ.0.0) GO TO 88055
           GCC(IY,IX)=GCUN(IY,IX)
           GO TO 88060
88055    IF(GCUE(IY,IX).GT.GCUW(IY,IX)) GO TO 88057
           IF(GCUW(IY,IX).GT.GCUN(IY,IX)) GO TO 88056
             GCC(IY,IX)=GCUN(IY,IX)-(GCUW(IY,IX)**2)/GCUN(IY,IX)
             GO TO 88060
88056      GCC(IY,IX)=0.
           GUPCO(IY,IX)=GCUN(IY,IX)+(GCUN(IY,IX)**2)/GCUW(IY,IX)
           IUPVAL(IY,IX)=1
           GO TO 88060
88057    IF(GCUE(IY,IX).GT.GCUN(IY,IX)) GO TO 88058
             GCC(IY,IX)=GCUN(IY,IX)-(GCUE(IY,IX)**2)/GCUN(IY,IX)
             GO TO 88060
88058      GCC(IY,IX)=0.
           GUPCO(IY,IX)=GCUN(IY,IX)+(GCUN(IY,IX)**2)/GCUE(IY,IX)
           IUPVAL(IY,IX)=2
88060   CONTINUE
         LOLD11=LOF(LD11)
        DO 88062 IX=1,NX
        DO 88062 IY=1,NY
          LFLD11=LOLD11+IY+NY*(IX-1)
          F(LFLD11)=GCC(IY,IX)
88062   CONTINUE
C         CALL PRN('CCUN',LD11)
C
C    SOUTH COEFFICIENT
C
        DO 88070 IX=1,NX-1
        DO 88070 IY=1,NY
         IF(GCUS(IY,IX).GT.0.0.AND.GCUN(IY,IX).EQ.0.0) GO TO 88065
           GCC(IY,IX)=GCUS(IY,IX)
           GO TO 88070
88065    IF(GCUE(IY,IX).GT.GCUW(IY,IX)) GO TO 88067
           IF(GCUW(IY,IX).GT.GCUS(IY,IX)) GO TO 88066
             GCC(IY,IX)=GCUS(IY,IX)-(GCUW(IY,IX)**2)/GCUS(IY,IX)
             GO TO 88070
88066      GCC(IY,IX)=0.
           GUPCO(IY,IX)=GCUS(IY,IX)+(GCUS(IY,IX)**2)/GCUW(IY,IX)
           IUPVAL(IY,IX)=4
           GO TO 88070
88067    IF(GCUE(IY,IX).GT.GCUS(IY,IX)) GO TO 88068
             GCC(IY,IX)=GCUS(IY,IX)-(GCUE(IY,IX)**2)/GCUS(IY,IX)
             GO TO 88070
88068      GCC(IY,IX)=0.
```

```
          GUPCO(IY,IX)=GCUS(IY,IX)+(GCUS(IY,IX)**2)/GCUE(IY,IX)
          IUPVAL(IY,IX)=3
88070  CONTINUE
        LOLD12=LOF(LD12)
       DO 88072 IX=1,NX
       DO 88072 IY=1,NY
         LFLD12=LOLD12+IY+NY*(IX-1)
         F(LFLD12)=GCC(IY,IX)
88072  CONTINUE
C         CALL PRN('CCUS',LD12)
         RETURN
C
C     EAST-WEST
C
88080 IF(NDIREC.NE.3) RETURN
C
       LOLD11=LOF(LD11)
       LOLD12=LOF(LD12)
C
       DO 88082 IX=1,NX
       DO 88082 IY=1,NY
        ICELL=IY+NY*(IX-1)
        GCUE(IY,IX)=F(LOLD11+ICELL)
        GCUW(IY,IX)=F(LOLD12+ICELL)
88082  CONTINUE
         IF(ISWEEP.EQ.FSWEEP) RETURN
C
C    EAST COEFFICIENT
C
      DO 88090 IX=1,NX-1
      DO 88090 IY=1,NY
         IF(GCUE(IY,IX).GT.0.0.AND.GCUW(IY,IX).EQ.0.0) GO TO 88085
           GCC(IY,IX)=GCUE(IY,IX)
           GO TO 88090
88085   IF(GCUN(IY,IX).GT.GCUS(IY,IX)) GO TO 88087
          IF(GCUS(IY,IX).GT.GCUE(IY,IX)) GO TO 88086
             GCC(IY,IX)=GCUE(IY,IX)-(GCUS(IY,IX)**2)/GCUE(IY,IX)
             GO TO 88090
88086      GCC(IY,IX)=0.
           GUPCO(IY,IX)=GCUE(IY,IX)+(GCUE(IY,IX)**2)/GCUS(IY,IX)
           IUPVAL(IY,IX)=3
           GO TO 88090
88087   IF(GCUN(IY,IX).GT.GCUE(IY,IX)) GO TO 88088
             GCC(IY,IX)=GCUE(IY,IX)-(GCUN(IY,IX)**2)/GCUE(IY,IX)
             GO TO 88090
88088      GCC(IY,IX)=0.
           GUPCO(IY,IX)=GCUE(IY,IX)+(GCUE(IY,IX)**2)/GCUN(IY,IX)
           IUPVAL(IY,IX)=2
88090 CONTINUE
        LOLD11=LOF(LD11)
      DO 88092 IX=1,NX
      DO 88092 IY=1,NY
         LFLD11=LOLD11+IY+NY*(IX-1)
         F(LFLD11)=GCC(IY,IX)
88092 CONTINUE
C         CALL PRN('CCUE',LD11)
C
C    WEST COEFFICIENT
C
          DO 88098 IX=1,NX-1
          DO 88098 IY=1,NY
         IF(GCUW(IY,IX).GT.0.0.AND.GCUE(IY,IX).EQ.0.0) GO TO 88093
           GCC(IY,IX)=GCUW(IY,IX)
           GO TO 88098
88093    IF(GCUN(IY,IX).GT.GCUS(IY,IX)) GO TO 88095
           IF(GCUS(IY,IX).GT.GCUW(IY,IX)) GO TO 88094
```

```
             GCC(IY,IX)=GCUW(IY,IX)-(GCUS(IY,IX)**2)/GCUW(IY,IX)
             GO TO 88098
88094      GCC(IY,IX)=0.
           GUPCO(IY,IX)=GCUW(IY,IX)+(GCUW(IY,IX)**2)/GCUS(IY,IX)
           IUPVAL(IY,IX)=4
           GO TO 88098
88095      IF(GCUN(IY,IX).GT.GCUW(IY,IX)) GO TO 88096
             GCC(IY,IX)=GCUW(IY,IX)-(GCUN(IY,IX)**2)/GCUW(IY,IX)
             GO TO 88098
88096      GCC(IY,IX)=0.
           GUPCO(IY,IX)=GCUW(IY,IX)+(GCUW(IY,IX)**2)/GCUN(IY,IX)
           IUPVAL(IY,IX)=1
88098    CONTINUE
        LOLD12=LOF(LD12)
       DO 88099 IX=1,NX
       DO 88099 IY=1,NY
         LFLD12=LOLD12+IY+NY*(IX-1)
         F(LFLD12)=GCC(IY,IX)
88099  CONTINUE
C         CALL PRN('CCUW',LD12)
         RETURN
C
C
88100 IF(.NOT.(LG(2))) GO TO 88200
       IF(INDVAR.NE.5.AND.INDVAR.NE.6) GO TO 88200
C
C  CORRECT COEFFICIENTS FOR V EQUATION
C  ***********************************
C
C  NORTH-SOUTH
C
      IF(NDIREC.NE.1) GO TO 88180
C
       LOLD11=LOF(LD11)
       LOLD12=LOF(LD12)

       DO 88102 IX=1,NX
       DO 88102 IY=1,NY
        ICELL=IY+NY*(IX-1)
        GCVN(IY,IX)=F(LOLD11+ICELL)
        GCVS(IY,IX)=F(LOLD12+ICELL)
88102  CONTINUE
         IF(ISWEEP.EQ.FSWEEP) GO TO 88180
C
C  NORTH COEFFICIENT
C
       DO 88160 IX=1,NX
       DO 88160 IY=1,NY-1
        IF(GCVN(IY,IX).GT.0.0.AND.GCVS(IY,IX).EQ.0.0) GO TO 88155
          GCC(IY,IX)=GCVN(IY,IX)
          GO TO 88160
88155   IF(GCVE(IY,IX).GT.GCVW(IY,IX)) GO TO 88157
          IF(GCVW(IY,IX).GT.GCVN(IY,IX)) GO TO 88156
            GCC(IY,IX)=GCVN(IY,IX)-(GCVW(IY,IX)**2)/GCVN(IY,IX)
            GO TO 88160
88156     GCC(IY,IX)=0.
          GVPCO(IY,IX)=GCVN(IY,IX)+(GCVN(IY,IX)**2)/GCVW(IY,IX)
          IVPVAL(IY,IX)=1
          GO TO 88160
88157   IF(GCVE(IY,IX).GT.GCVN(IY,IX)) GO TO 88158
            GCC(IY,IX)=GCVN(IY,IX)-(GCVE(IY,IX)**2)/GCVN(IY,IX)
            GO TO 88160
88158     GCC(IY,IX)=0.
          GVPCO(IY,IX)=GCVN(IY,IX)+(GCVN(IY,IX)**2)/GCVE(IY,IX)
          IVPVAL(IY,IX)=2
88160   CONTINUE
```

```
        LOLD11=LOF(LD11)
       DO 88162 IX=1,NX
       DO 88162 IY=1,NY
         LFLD11=LOLD11+IY+NY*(IX-1)
         F(LFLD11)=GCC(IY,IX)
88162  CONTINUE
C        CALL PRN('CCVN',LD11)
C
C    SOUTH COEFFICIENT
C
        DO 88170 IX=1,NX
        DO 88170 IY=1,NY-1
         IF(GCVS(IY,IX).GT.0.0.AND.GCVN(IY,IX).EQ.0.0) GO TO 88165
           GCC(IY,IX)=GCVS(IY,IX)
           GO TO 88170
88165    IF(GCVE(IY,IX).GT.GCVW(IY,IX)) GO TO 88167
           IF(GCVW(IY,IX).GT.GCVS(IY,IX)) GO TO 88166
             GCC(IY,IX)=GCVS(IY,IX)-(GCVW(IY,IX)**2)/GCVS(IY,IX)
             GO TO 88170
88166      GCC(IY,IX)=0.
           GVPCO(IY,IX)=GCVS(IY,IX)+(GCVS(IY,IX)**2)/GCVW(IY,IX)
           IVPVAL(IY,IX)=4
           GO TO 88170
88167      IF(GCVE(IY,IX).GT.GCVS(IY,IX)) GO TO 88168
             GCC(IY,IX)=GCVS(IY,IX)-(GCVE(IY,IX)**2)/GCVS(IY,IX)
             GO TO 88170
88168      GCC(IY,IX)=0.
           GVPCO(IY,IX)=GCVS(IY,IX)+(GCVS(IY,IX)**2)/GCVE(IY,IX)
           IVPVAL(IY,IX)=3
88170    CONTINUE
        LOLD12=LOF(LD12)
       DO 88172 IX=1,NX
       DO 88172 IY=1,NY
         LFLD12=LOLD12+IY+NY*(IX-1)
         F(LFLD12)=GCC(IY,IX)
88172  CONTINUE
C          CALL PRN('CCVS',LD12)
C
      RETURN
C
C     EAST-WEST
C
88180 IF(NDIREC.NE.3) RETURN
C
       LOLD11=LOF(LD11)
       LOLD12=LOF(LD12)

       DO 88182 IX=1,NX
       DO 88182 IY=1,NY
        ICELL=IY+NY*(IX-1)
        GCVE(IY,IX)=F(LOLD11+ICELL)
        GCVW(IY,IX)=F(LOLD12+ICELL)
88182  CONTINUE
         IF(ISWEEP.EQ.FSWEEP) RETURN
C
C    EAST COEFFICIENT
C
          DO 88190 IX=1,NX
          DO 88190 IY=1,NY-1
         IF(GCVE(IY,IX).GT.0.0.AND.GCVW(IY,IX).EQ.0.0) GO TO 88185
           GCC(IY,IX)=GCVE(IY,IX)
           GO TO 88190
88185    IF(GCVN(IY,IX).GT.GCVS(IY,IX)) GO TO 88187
           IF(GCVS(IY,IX).GT.GCVE(IY,IX)) GO TO 88186
             GCC(IY,IX)=GCVE(IY,IX)-(GCVS(IY,IX)**2)/GCVE(IY,IX)
             GO TO 88190
```

```
88186        GCC(IY,IX)=0.
             GVPCO(IY,IX)=GCVE(IY,IX)+(GCVE(IY,IX)**2)/GCVS(IY,IX)
             IVPVAL(IY,IX)=3
             GO TO 88190
88187        IF(GCVN(IY,IX).GT.GCVE(IY,IX)) GO TO 88188
               GCC(IY,IX)=GCVE(IY,IX)-(GCVN(IY,IX)**2)/GCVE(IY,IX)
               GO TO 88190
88188        GCC(IY,IX)=0.
             GVPCO(IY,IX)=GCVE(IY,IX)+(GCVE(IY,IX)**2)/GCVN(IY,IX)
             IVPVAL(IY,IX)=2
88190      CONTINUE
          LOLD11=LOF(LD11)
        DO 88192 IX=1,NX
        DO 88192 IY=1,NY
           LFLD11=LOLD11+IY+NY*(IX-1)
           F(LFLD11)=GCC(IY,IX)
88192 CONTINUE
C           CALL PRN('CCVE',LD11)
C
C     WEST COEFFICIENT
C
            DO 88198 IX=1,NX
            DO 88198 IY=1,NY-1
           IF(GCVW(IY,IX).GT.0.0.AND.GCVE(IY,IX).EQ.0.0) GO TO 88193
             GCC(IY,IX)=GCVW(IY,IX)
             GO TO 88198
88193      IF(GCVN(IY,IX).GT.GCVS(IY,IX)) GO TO 88195
             IF(GCVS(IY,IX).GT.GCVW(IY,IX)) GO TO 88194
               GCC(IY,IX)=GCVW(IY,IX)-(GCVS(IY,IX)**2)/GCVW(IY,IX)
               GO TO 88198
88194        GCC(IY,IX)=0.
             GVPCO(IY,IX)=GCVW(IY,IX)+(GCVW(IY,IX)**2)/GCVS(IY,IX)
             IVPVAL(IY,IX)=4
             GO TO 88198
88195        IF(GCVN(IY,IX).GT.GCVW(IY,IX)) GO TO 88196
               GCC(IY,IX)=GCVW(IY,IX)-(GCVN(IY,IX)**2)/GCVW(IY,IX)
               GO TO 88198
88196        GCC(IY,IX)=0.
             GVPCO(IY,IX)=GCVW(IY,IX)+(GCVW(IY,IX)**2)/GCVN(IY,IX)
             IVPVAL(IY,IX)=1
88198      CONTINUE
          LOLD12=LOF(LD12)
         DO 88199 IX=1,NX
         DO 88199 IY=1,NY
           LFLD12=LOLD12+IY+NY*(IX-1)
           F(LFLD12)=GCC(IY,IX)
88199  CONTINUE
C           CALL PRN('CCVW',LD12)
           RETURN
C
C
88200 IF(INDVAR.NE.7.AND.INDVAR.NE.8) GO TO 88280
C
C  CORRECTION NOT IMPLEMENTED FOR W EQUATION
C
      CALL WRIT40('WARNING: SUCCA NOT IMPLEMENTED FOR W    ')
      RETURN
C
88280 IF(.NOT.(LG(3))) RETURN
      IF(INDVAR.GT.11) GO TO 88301
      RETURN
88301 CONTINUE
C
C  CORRECT COEFFICIENTS FOR SCALAR VARIABLES
C  ******************************************
C
```

```
C  NORTH-SOUTH
C
      IF(NDIREC.NE.1) GO TO 88330
C
       LOLD11=LOF(LD11)
       LOLD12=LOF(LD12)
C
       DO 88302 IX=1,NX
       DO 88302 IY=1,NY
        ICELL=IY+NY*(IX-1)
        GCSN(IY,IX)=F(LOLD11+ICELL)
        GCSS(IY,IX)=F(LOLD12+ICELL)
88302  CONTINUE
         IF(ISWEEP.EQ.FSWEEP) GO TO 88330
C
C   NORTH COEFFICIENT
C
         DO 88310 IX=1,NX
         DO 88310 IY=1,NY
          IF(GCSN(IY,IX).GT.0.0.AND.GCSS(IY,IX).EQ.0.0) GO TO 88305
            GCC(IY,IX)=GCSN(IY,IX)
            GO TO 88310
88305     IF(GCSE(IY,IX).GT.GCSW(IY,IX)) GO TO 88307
            IF(GCSW(IY,IX).GT.GCSN(IY,IX)) GO TO 88306
              GCC(IY,IX)=GCSN(IY,IX)-(GCSW(IY,IX)**2)/GCSN(IY,IX)
              GO TO 88310
88306       GCC(IY,IX)=0.
            GSPCO(IY,IX)=GCSN(IY,IX)+(GCSN(IY,IX)**2)/GCSW(IY,IX)
            ISPVAL(IY,IX)=1
            GO TO 88310
88307     IF(GCSE(IY,IX).GT.GCSN(IY,IX)) GO TO 88308
              GCC(IY,IX)=GCSN(IY,IX)-(GCSE(IY,IX)**2)/GCSN(IY,IX)
              GO TO 88310
88308       GCC(IY,IX)=0.
            GSPCO(IY,IX)=GCSN(IY,IX)+(GCSN(IY,IX)**2)/GCSE(IY,IX)
            ISPVAL(IY,IX)=2
88310    CONTINUE
        LOLD11=LOF(LD11)
       DO 88312 IX=1,NX
       DO 88312 IY=1,NY
         LFLD11=LOLD11+IY+NY*(IX-1)
         F(LFLD11)=GCC(IY,IX)
88312  CONTINUE
C         CALL PRN('CCSN',LD11)
C
C   SOUTH COEFFICIENT
C
         DO 88320 IX=1,NX
         DO 88320 IY=1,NY
          IF(GCSS(IY,IX).GT.0.0.AND.GCSN(IY,IX).EQ.0.0) GO TO 88315
            GCC(IY,IX)=GCSS(IY,IX)
            GO TO 88320
88315     IF(GCSE(IY,IX).GT.GCSW(IY,IX)) GO TO 88317
            IF(GCSW(IY,IX).GT.GCSS(IY,IX)) GO TO 88316
              GCC(IY,IX)=GCSS(IY,IX)-(GCSW(IY,IX)**2)/GCSS(IY,IX)
              GO TO 88320
88316       GCC(IY,IX)=0.
            GSPCO(IY,IX)=GCSS(IY,IX)+(GCSS(IY,IX)**2)/GCSW(IY,IX)
            ISPVAL(IY,IX)=4
            GO TO 88320
88317     IF(GCSE(IY,IX).GT.GCSS(IY,IX)) GO TO 88318
              GCC(IY,IX)=GCSS(IY,IX)-(GCSE(IY,IX)**2)/GCSS(IY,IX)
              GO TO 88320
88318       GCC(IY,IX)=0.
            GSPCO(IY,IX)=GCSS(IY,IX)+(GCSS(IY,IX)**2)/GCSE(IY,IX)
            ISPVAL(IY,IX)=3
```

```
88320      CONTINUE
          LOLD12=LOF(LD12)
         DO 88322 IX=1,NX
         DO 88322 IY=1,NY
           LFLD12=LOLD12+IY+NY*(IX-1)
           F(LFLD12)=GCC(IY,IX)
88322    CONTINUE
C           CALL PRN('CCSS',LD12)
           RETURN
C
C      EAST-WEST
C
88330 IF(NDIREC.NE.3) RETURN
C
         LOLD11=LOF(LD11)
         LOLD12=LOF(LD12)
C
         DO 88332 IX=1,NX
         DO 88332 IY=1,NY
          ICELL=IY+NY*(IX-1)
          GCSE(IY,IX)=F(LOLD11+ICELL)
          GCSW(IY,IX)=F(LOLD12+ICELL)
88332    CONTINUE
           IF(ISWEEP.EQ.FSWEEP) RETURN
C
C    EAST COEFFICIENT
C
           DO 88340 IX=1,NX
           DO 88340 IY=1,NY
            IF(GCSE(IY,IX).GT.0.0.AND.GCSW(IY,IX).EQ.0.0) GO TO 88335
              GCC(IY,IX)=GCSE(IY,IX)
              GO TO 88340
88335       IF(GCSN(IY,IX).GT.GCSS(IY,IX)) GO TO 88337
              IF(GCSS(IY,IX).GT.GCSE(IY,IX)) GO TO 88336
                GCC(IY,IX)=GCSE(IY,IX)-(GCSS(IY,IX)**2)/GCSE(IY,IX)
                GO TO 88340
88336         GCC(IY,IX)=0.
              GSPCO(IY,IX)=GCSE(IY,IX)+(GCSE(IY,IX)**2)/GCSS(IY,IX)
              ISPVAL(IY,IX)=3
              GO TO 88340
88337       IF(GCSN(IY,IX).GT.GCSE(IY,IX)) GO TO 88338
                GCC(IY,IX)=GCSE(IY,IX)-(GCSN(IY,IX)**2)/GCSE(IY,IX)
                GO TO 88340
88338         GCC(IY,IX)=0.
              GSPCO(IY,IX)=GCSE(IY,IX)+(GCSE(IY,IX)**2)/GCSN(IY,IX)
              ISPVAL(IY,IX)=2
88340      CONTINUE
         LOLD11=LOF(LD11)
       DO 88342 IX=1,NX
       DO 88342 IY=1,NY
          LFLD11=LOLD11+IY+NY*(IX-1)
          F(LFLD11)=GCC(IY,IX)
88342 CONTINUE
C          CALL PRN('CCSE',LD11)
C
C   WEST COEFFICIENT
C
           DO 88350 IX=1,NX
           DO 88350 IY=1,NY
            IF(GCSW(IY,IX).GT.0.0.AND.GCSE(IY,IX).EQ.0.0) GO TO 88345
              GCC(IY,IX)=GCSW(IY,IX)
              GO TO 88350
88345       IF(GCSN(IY,IX).GT.GCSS(IY,IX)) GO TO 88347
              IF(GCSS(IY,IX).GT.GCSW(IY,IX)) GO TO 88346
                GCC(IY,IX)=GCSW(IY,IX)-(GCSS(IY,IX)**2)/GCSW(IY,IX)
                GO TO 88350
```

```
88346       GCC(IY,IX)=0.
            GSPCO(IY,IX)=GCSW(IY,IX)+(GCSW(IY,IX)**2)/GCSS(IY,IX)
            ISPVAL(IY,IX)=4
            GO TO 88350
88347     IF(GCSN(IY,IX).GT.GCSW(IY,IX)) GO TO 88348
              GCC(IY,IX)=GCSW(IY,IX)-(GCSN(IY,IX)**2)/GCSW(IY,IX)
              GO TO 88350
88348       GCC(IY,IX)=0.
            GSPCO(IY,IX)=GCSW(IY,IX)+(GCSW(IY,IX)**2)/GCSN(IY,IX)
            ISPVAL(IY,IX)=1
88350     CONTINUE
        LOLD12=LOF(LD12)
       DO 88352 IX=1,NX
       DO 88352 IY=1,NY
         LFLD12=LOLD12+IY+NY*(IX-1)
         F(LFLD12)=GCC(IY,IX)
88352  CONTINUE
C         CALL PRN('CCSW',LD12)
      RETURN
   89 CONTINUE
C   * -------------------- SECTION 9 ---- Diffusion coefficients
C   * -----------GROUP 8  SECTION 9 --- DIFFUSION COEFFICIENTS
C--- Entered when UDIFF =.TRUE.; block-location indices are LAE
C    for east, LAW for west, LAN for north, LAS for
C    south, LD11 for high, and LD11 for low.
C    User should provide INDVAR and NDIREC IF's as above.
C
      RETURN
  810 CONTINUE
C   * ------------------- SECTION 10 --- Convection neighbours
      RETURN
  811 CONTINUE
C   * ------------------- SECTION 11 --- Diffusion neighbours
      RETURN
  812 CONTINUE
C   * ------------------- SECTION 12 --- Linearised sources
      RETURN
  813 CONTINUE
C   * ------------------- SECTION 13 --- Correction coefficients
      RETURN
  814 CONTINUE
C   * ------------------- SECTION 14 --- User's solver
      RETURN
  815 CONTINUE
C   * ------------------- SECTION 15 --- Change solution
      RETURN
C
C   * See the equivalent section in GREX for the indices to be
C     used in sections 7 - 15
C
C   * Make all other group-8 changes in GROUP 19.
C*********************************************************************
C
C--- GROUP 9. Properties of the medium (or media)
C
C   The sections in this group are arranged sequentially in their
C   order of calling from EARTH. Thus, as can be seen from below,
C   the temperature sections (10 and 11) precede the density
C   sections (1 and 3); so, density formulae can refer to
C   temperature stores already set.
C
    9 GO TO (91,92,93,94,95,96,97,98,99,900,901,902,903),ISC
C*********************************************************************
C   * ------------------- SECTION 10 ---------------------------
C    For TMP1.LE.GRND--------- phase-1 temperature Index TEMP1
  900 CONTINUE
```

```
      RETURN
C   * ------------------- SECTION 11 ---------------------------
C    For TMP2.LE.GRND--------- phase-2 temperature Index TEMP2
  901 CONTINUE
      RETURN
C   * ------------------- SECTION 12 ---------------------------
C    For EL1.LE.GRND--------- phase-1 length scale Index LEN1
  902 CONTINUE
      RETURN
C   * ------------------- SECTION 13 ---------------------------
C    For EL2.LE.GRND--------- phase-2 length scale Index LEN2
  903 CONTINUE
      RETURN
C   * ------------------- SECTION  1 ---------------------------
C    For RHO1.LE.GRND--- density for phase 1       Index DEN1
   91 CONTINUE
      RETURN
C   * ------------------- SECTION  2 ---------------------------
C    For DRH1DP.LE.GRND--- D(LN(DEN))/DP for phase 1
C                                                  Index D1DP
   92 CONTINUE
      RETURN
C   * ------------------- SECTION  3 ---------------------------
C    For RHO2.LE.GRND--- density for phase 2       Index DEN2
   93 CONTINUE
      RETURN
C   * ------------------- SECTION  4 ---------------------------
C    For DRH2DP.LE.GRND--- D(LN(DEN))/DP for phase 2
C                                                  Index D2DP
   94 CONTINUE
      RETURN
C   * ------------------- SECTION  5 ---------------------------
C    For ENUT.LE.GRND--- reference turbulent kinematic viscosity
C                                                  Index VIST
   95 CONTINUE
      RETURN
C   * ------------------- SECTION  6 ---------------------------
C    For ENUL.LE.GRND--- reference laminar kinematic viscosity
   96 CONTINUE
C                                                  Index VISL
      RETURN
C   * ------------------- SECTION  7 ---------------------------
C    For PRNDTL( ).LE.GRND--- laminar PRANDTL nos., or diffusivity
C                                                  Index LAMPR
   97 CONTINUE
      RETURN
C   * ------------------- SECTION  8 ---------------------------
C    For PHINT( ).LE.GRND--- interface value of first phase
C                                                  Index FII1
   98 CONTINUE
      RETURN
C   * ------------------- SECTION  9 ---------------------------
C    For PHINT( ).LE.GRND--- interface value of second phase
C                                                  Index FII2
   99 CONTINUE
      RETURN
C*********************************************************************
C
C--- GROUP 10. Inter-phase-transfer processes and properties
C
   10 GO TO (101,102,103,104),ISC
C   * ------------------- SECTION  1 ---------------------------
C    For CFIPS.LE.GRND--- inter-phase friction coeff.
C                                                  Index AUX(INTFRC)
  101 CONTINUE
      RETURN
```

```
C   * ------------------- SECTION  2 ---------------------------
C    For CMDOT.EQ.GRND- inter-phase mass transfer  Index AUX(INTMDT)
  102 CONTINUE
      RETURN
C   * ------------------- SECTION  3 ---------------------------
C    For CINT( ).EQ.GRND--- phase1-to-interface transfer coefficients
C                                                     Index COI1
  103 CONTINUE
      RETURN
C   * ------------------- SECTION  4 ---------------------------
C    For CINT( ).EQ.GRND--- phase2-to-interface transfer coefficients
C                                                     Index COI2
  104 CONTINUE
      RETURN
C**********************************************************************
C
C--- GROUP 11. Initialization of variable or porosity fields
C                                                     Index VAL
   11 CONTINUE
      RETURN
C**********************************************************************
C
C--- GROUP 12. Convection and diffusion adjustments
C
   12 CONTINUE
      RETURN
C**********************************************************************
C
C--- GROUP 13. Boundary conditions and special sources
C                                           Index for Coefficient - CO
C                                           Index for Value       - VAL
   13 GO TO (130,131,132,133,134,135,136,137,138,139,1310,
     11311,1312,1313,1314,1315,1316,1317,1318,1319,1320,1321),ISC
C------------------- SECTION  1 ------------- coefficient = GRND
  130 CONTINUE
C
C  SCALAR VARIABLE
C
C       CALL WRIT1I('INDVAR  ',INDVAR)
C
      IF(NPATCH.NE.'SUCCAU'.OR.INDVAR.LT.9) GO TO 13020
      CALL ONLYIF(12,50,'SUCCAU')
      CALL SETYX(CO,GSPCO,NYDIM,NXDIM)
C
13020 CONTINUE
C
C  U-VELOCITY VARIABLE
C
      IF(NPATCH.NE.'SUCCAU'.OR.INDVAR.NE.3) GO TO 13040
      CALL ONLYIF(3,3,'SUCCAU')
      CALL SETYX(CO,GUPCO,NYDIM,NXDIM)
C
13040 CONTINUE
C
C  V-VELOCITY VARIABLE
C
      IF(NPATCH.NE.'SUCCAU'.OR.INDVAR.NE.5) GO TO 13060
      CALL ONLYIF(5,5,'SUCCAU')
      CALL SETYX(CO,GVPCO,NYDIM,NXDIM)
13060 RETURN
C------------------- SECTION  2 ------------- coefficient = GRND1
  131 CONTINUE
      RETURN
C------------------- SECTION  3 ------------- coefficient = GRND2
  132 CONTINUE
      RETURN
```

```
C------------------- SECTION  4 ------------- coefficient = GRND3
  133 CONTINUE
      RETURN
C------------------- SECTION  5 ------------- coefficient = GRND4
  134 CONTINUE
      RETURN
C------------------- SECTION  6 ------------- coefficient = GRND5
  135 CONTINUE
      RETURN
C------------------- SECTION  7 ------------- coefficient = GRND6
  136 CONTINUE
      RETURN
C------------------- SECTION  8 ------------- coefficient = GRND7
  137 CONTINUE
      RETURN
C------------------- SECTION  9 ------------- coefficient = GRND8
  138 CONTINUE
      RETURN
C------------------- SECTION 10 ------------- coefficient = GRND9
  139 CONTINUE
      RETURN
C------------------- SECTION 11 ------------- coefficient = GRND10
 1310 CONTINUE
      RETURN
C------------------- SECTION 12 ------------------- value = GRND
 1311 CONTINUE
C
C  SOURCES FOR CONVECTION FROM UPWIND CORNER CELL
C  **********************************************
C
C  SCALAR VARIABLES
C
      IF(NPATCH.NE.'SUCCAU'.OR.INDVAR.LT.9) GO TO 13118
      CALL ONLYIF(12,50,'SUCCAU')
C
       LVAR=LOF(INDVAR)
       DO 13112 IX=1,NX
       DO 13112 IY=1,NY
       LOCATE=IY+NY*(IX-1)
       GTMP(IY,IX)=F(LVAR+LOCATE)
13112  CONTINUE
C
      DO 13115 IY=1,NY
      DO 13115 IX=1,NX
      IF(ISPVAL(IY,IX).EQ.0) GO TO 13115
      IF(ISPVAL(IY,IX).EQ.1) GCC(IY,IX)=GTMP(IY+1,IX-1)
      IF(ISPVAL(IY,IX).EQ.2) GCC(IY,IX)=GTMP(IY+1,IX+1)
      IF(ISPVAL(IY,IX).EQ.3) GCC(IY,IX)=GTMP(IY-1,IX+1)
      IF(ISPVAL(IY,IX).EQ.4) GCC(IY,IX)=GTMP(IY-1,IX-1)
13115 CONTINUE
      CALL SETYX(VAL,GCC,NYDIM,NXDIM)
      DO 13117 IX=1,NX
      DO 13117 IY=1,NY
        ISPVAL(IY,IX)=0
13117 CONTINUE
C      CALL PRN('VAHU',VAL)
13118 CONTINUE
C
C  U-VELOCITY VARIABLE
C
      IF(NPATCH.NE.'SUCCAU'.OR.INDVAR.NE.3) GO TO 13124
      CALL ONLYIF(3,3,'SUCCAU')
C
       LVAR=LOF(INDVAR)
       DO 13120 IX=1,NX
       DO 13120 IY=1,NY
```

```
        LOCATE=IY+NY*(IX-1)
        GTMP(IY,IX)=F(LVAR+LOCATE)
13120   CONTINUE
C
      DO 13121 IY=1,NY
      DO 13121 IX=1,NX-1
      IF(IUPVAL(IY,IX).EQ.0) GO TO 13121
      IF(IUPVAL(IY,IX).EQ.1) GCC(IY,IX)=GTMP(IY+1,IX-1)
      IF(IUPVAL(IY,IX).EQ.2) GCC(IY,IX)=GTMP(IY+1,IX+1)
      IF(IUPVAL(IY,IX).EQ.3) GCC(IY,IX)=GTMP(IY-1,IX+1)
      IF(IUPVAL(IY,IX).EQ.4) GCC(IY,IX)=GTMP(IY-1,IX-1)
13121 CONTINUE
      CALL SETYX(VAL,GCC,NYDIM,NXDIM)
      DO 13123 IX=1,NX-1
      DO 13123 IY=1,NY
        IUPVAL(IY,IX)=0
13123 CONTINUE
C       CALL PRN('VAUU',VAL)
13124 CONTINUE
C
C  V-VELOCITY VARIABLE
C
      IF(NPATCH.NE.'SUCCAU'.OR.INDVAR.NE.5) GO TO 13130
      CALL ONLYIF(5,5,'SUCCAU')
C
       LVAR=L0F(INDVAR)
       DO 13126 IX=1,NX
       DO 13126 IY=1,NY
       LOCATE=IY+NY*(IX-1)
       GTMP(IY,IX)=F(LVAR+LOCATE)
13126  CONTINUE
C
      DO 13127 IY=1,NY-1
      DO 13127 IX=1,NX
      IF(IVPVAL(IY,IX).EQ.0) GO TO 13127
      IF(IVPVAL(IY,IX).EQ.1) GCC(IY,IX)=GTMP(IY+1,IX-1)
      IF(IVPVAL(IY,IX).EQ.2) GCC(IY,IX)=GTMP(IY+1,IX+1)
      IF(IVPVAL(IY,IX).EQ.3) GCC(IY,IX)=GTMP(IY-1,IX+1)
      IF(IVPVAL(IY,IX).EQ.4) GCC(IY,IX)=GTMP(IY-1,IX-1)
13127 CONTINUE
      CALL SETYX(VAL,GCC,NYDIM,NXDIM)
      DO 13129 IX=1,NX
      DO 13129 IY=1,NY-1
        IVPVAL(IY,IX)=0
13129 CONTINUE
C       CALL PRN('VAVU',VAL)
C
13130 RETURN
C------------------- SECTION 13 ------------------- value = GRND1
 1312 CONTINUE
      RETURN
C------------------- SECTION 14 ------------------- value = GRND2
 1313 CONTINUE
      RETURN
C------------------- SECTION 15 ------------------- value = GRND3
 1314 CONTINUE
      RETURN
C------------------- SECTION 16 ------------------- value = GRND4
 1315 CONTINUE
      RETURN
C------------------- SECTION 17 ------------------- value = GRND5
 1316 CONTINUE
      RETURN
C------------------- SECTION 18 ------------------- value = GRND6
 1317 CONTINUE
      RETURN
```

```
C------------------- SECTION 19 ------------------- value = GRND7
 1318 CONTINUE
      RETURN
C------------------- SECTION 20 ------------------- value = GRND8
 1319 CONTINUE
      RETURN
C------------------- SECTION 21 ------------------- value = GRND9
 1320 CONTINUE
      RETURN
C------------------- SECTION 22 ------------------- value = GRND10
 1321 CONTINUE
      RETURN
C***************************************************************
C
C--- GROUP 14. Downstream pressure for PARAB=.TRUE.
C
   14 CONTINUE
      RETURN
C***************************************************************
C* Make changes for these groups only in GROUP 19.
C--- GROUP 15. Termination of sweeps
C--- GROUP 16. Termination of iterations
C--- GROUP 17. Under-relaxation devices
C--- GROUP 18. Limits on variables or increments to them
C***************************************************************
C
C--- GROUP 19. Special calls to GROUND from EARTH
C
   19 GO TO (191,192,193,194,195,196,197,198),ISC
C   * ------------------- SECTION 1 ---- Start of time step.
  191 CONTINUE
      RETURN
C   * ------------------- SECTION 2 ---- Start of sweep.
  192 CONTINUE
      RETURN
C   * ------------------- SECTION 3 ---- Start of iz slab.
  193 CONTINUE
      RETURN
C   * ------------------- SECTION 4 ---- Start of iteration.
  194 CONTINUE
      RETURN
C   * ------------------- SECTION 5 ---- Finish of iteration.
  195 CONTINUE
      RETURN
C   * ------------------- SECTION 6 ---- Finish of iz slab.
  196 CONTINUE
      RETURN
C   * ------------------- SECTION 7 ---- Finish of sweep.
  197 CONTINUE
      RETURN
C   * ------------------- SECTION 8 ---- Finish of time step.
  198 CONTINUE
      RETURN
C***************************************************************
C
C--- GROUP 20. Preliminary print-out
C
   20 CONTINUE
      RETURN
C***************************************************************
C* Make changes for these groups only in GROUP 19.
C--- GROUP 21. Print-out of variables
C--- GROUP 22. Spot-value print-out
C***************************************************************
C
C--- GROUP 23. Field print-out and plot control
```

```
   23 CONTINUE
      RETURN
C*******************************************************************
C
C--- GROUP 24. Dumps for restarts
C
   24 CONTINUE
      END
C*******************************************************************
      SUBROUTINE SPECGR
      CALL WRIT40('Dummy subroutine SPECGR  called.         ')
      CALL WAYOUT(2)
      END
C*******************************************************************
      SUBROUTINE SPC1GR
      CALL WRIT40('Dummy subroutine SPC1GR called.          ')
      CALL WAYOUT(2)
      END
C*******************************************************************
      SUBROUTINE SPC2GR
      CALL WRIT40('Dummy subroutine SPC2GR called.          ')
      CALL WAYOUT(2)
      END
C*******************************************************************
      SUBROUTINE SPC3GR
      CALL WRIT40('Dummy subroutine SPC3GR called.          ')
      CALL WAYOUT(2)
      END
C*******************************************************************
      SUBROUTINE QUIZ
C--- This subroutine is used by CHAM for de-bugging
      END
```

APPENDIX M

GROUND FORTRAN CODING FOR SUCCA3D SCHEME

```
C FILE NAME GROUND.FTN--------------------------------100392
C   THIS IS THE MAIN PROGRAM OF EARTH
C
C   (C) COPYRIGHT 1984, LAST REVISION 1992
C   CONCENTRATION HEAT AND MOMENTUM LTD. ALL RIGHTS RESERVED.
C   This subroutine and the remainder of the PHOENICS code are
C   proprietary software owned by Concentration Heat and Momentum
C   Limited, 40 High Street, Wimbledon, London SW19 5AU, England.
C
C
      PROGRAM MAIN
C
C 1   The following  COMMON's, which appear identically in the
C     satellite MAIN program, allow up to 50 dependent variables to
C     be solved for (or their storage spaces to be occupied by
C     other variables, such as density). If a larger number is
C     required, the PARAMETER NUMPHI should be reset to the required
C     larger number. Numbers less than 50 are not permitted.
C
      PARAMETER (NUMPHI=50,   NM=NUMPHI,NM4=NM*4)
C
      COMMON/LGE4/L4(NM)
     1/LDB1/L5(NM)/IDA1/I1(NM)/IDA2/I2(NM)/IDA3/I3(NM)/IDA4/I4(NM)
     1/IDA5/I5(NM)/IDA6/I6(NM)/GI1/I7(NM)/GI2/I8(NM)/HDA1/IH1(NM)
     1/GH1/IH2(NM)/RDA1/R1(NM)/RDA2/R2(NM)/RDA3/R3(NM)/RDA4/R4(NM)
     1/RDA5/R5(NM)/RDA6/R6(NM)/RDA7/R7(NM)/RDA8/R8(NM)/RDA9/R9(NM)
     1/RDA10/R10(NM)/RDA11/R11(NM)
     1/GR1/R12(NM)/GR2/R13(NM)/GR3/R14(NM)/GR4/R15(NM)
     1/IPIP1/IP1(NM)/HPIP2/IHP2(NM)/RPIP1/RVAL(NM)/LPIP1/LVAL(NM)
     1/IFPL/IPL0(NM)/RFPL1/ORPRIN(NM)/RFPL2/ORMAX(NM)
     1/RFPL3/ORMIN(NM)/IDA7/ID7(NM)/IDA8/ID8(NM)
      LOGICAL L4,L5,DBGFIL,LVAL
      CHARACTER*4 IH1,IH2,IHP2
C
      COMMON/F01/I9(NM4)
      COMMON/DISC/DBGFIL
      COMMON/LUNITS/LUNIT(60)
C
      EXTERNAL WAYOUT
C
C 2   Set dimensions of data-for-GROUND arrays here. WARNING: the
C     corresponding arrays in the MAIN program of the satellite
C     (see SATLIT) must have the same dimensions.
      PARAMETER (NLG=20, NIG=20, NRG=100, NCG=10)
C
      COMMON/LGRND/LG(NLG)/IGRND/IG(NIG)/RGRND/RG(NRG)/CGRND/CG(NCG)
      LOGICAL LG
      CHARACTER*4 CG
      COMMON/GUGRND/GCSN(60,80),GCSS(60,80),GCSE(60,80),GCSW(60,80),
     7              GCSL(60,80),GCSH(60,80),GSPCO(60,80),GSECO(60,80),
     8              GCUN(60,80),GCUS(60,80),GCUE(60,80),GCUW(60,80),
     7              GCUL(60,80),GCUH(60,80),GUPCO(60,80),GUECO(60,80),
     8              GCVN(60,80),GCVS(60,80),GCVE(60,80),GCVW(60,80),
     7              GCVL(60,80),GCVH(60,80),GVPCO(60,80),GVECO(60,80),
     8              GCWN(60,80),GCWS(60,80),GCWE(60,80),GCWW(60,80),
     7              GCWL(60,80),GCWH(60,80),GWPCO(60,80),GWECO(60,80)
      COMMON/IUGRND/ISPVAL(60,80),ISEVAL(60,80),IUPVAL(60,80),
     7              IUEVAL(60,80),IVEVAL(60,80),IVPVAL(60,80),
     8              IWEVAL(60,80),IWPVAL(60,80)
C
C
C 3   Set dimensions of data-for-GREX arrays here. WARNING: the
C     corresponding arrays in the MAIN program of the satellite
C     (see SATLIT) must have the same dimensions.
      PARAMETER(NLSG=20, NISG=20, NRSG=100,NCSG=10)
C
```

```
      COMMON/LSG/LSGD(NLSG)/ISG/ISGD(NISG)/RSG/RSGD(NRSG)/CSG/CSGD(NCSG)
      LOGICAL LSGD
      CHARACTER*4 CSGD
C
C 4   Set dimension of patch-name array here. WARNING: the array
C     NAMPAT in the MAIN program of the satellite must have the
C     same dimension.
      PARAMETER (NPNAM=900)
C
      COMMON/NPAT/NAMPAT(NPNAM)
      COMMON/LWFUN1/DOSKIN(NPNAM)
      COMMON/LWFUN2/DHCHKD(NPNAM)
      CHARACTER*8 NAMPAT
      LOGICAL DOSKIN,DHCHKD
C
C     CONFIG FILE name declaration.
      COMMON/CNFG/CNFIG
      CHARACTER CNFIG*48
C
C 5   The numbers in the next statement indicates how much computer
C     memory is to be set aside for storing the main and auxiliary
C     variables. The user may alter them if he wishes, to accord
C     with the number of grid nodes and dependent variables he is
C     concerned with.
      PARAMETER (NFDIM=3200000)
C
      COMMON F(NFDIM)
C
C 6   The following three statements concern storage for the PATCH-wise
C     variables. If more than 30 PATCH-wise variables are required
C     NPVDM should be increased and the common block /LBPV/ in the
C     include file GRDLOC15 should be lengthened.
      PARAMETER (NPVDM=30)
      COMMON/INDPV/NPVMX,NIMAX,NITOT,LOPV(NPVDM)
C
      CALL SUB2(NPVMX,NPVDM,NIMAX,NPVDM)
C
      CALL CNFGZZ(2)
      CALL EARSET(1)
      CALL OPENFL(6)
C
      CALL MAIN1(NFDIM,NUMPHI,NLSG,NISG,NRSG,NCSG,NLG,NIG,NRG,NCG)
      CALL WAYOUT(0)
      STOP
      END
C*****************************************************************
      SUBROUTINE GROSTA
      INCLUDE 'lp16/d_earth/SATEAR'
      INCLUDE 'lp16/d_earth/GRDLOC'
      INCLUDE 'lp16/d_earth/GRDEAR'
C
C.... This subroutine acts as a junction-box, directing control to
C     the GROUNDs selected by the SATELLITE settings of USEGRX,
C     NAMGRD & USEGRD.
C
      IF(USEGRX) CALL GREX3
C
C.... SPECGR, SPC1GR, SPC2GR and SPC3GR are names which the user may
C     give to "special GROUNDs" of his own.
C
      IF(NAMGRD.NE.'NONE') THEN
        IF(NAMGRD.EQ.'SPEC') THEN
          CALL SPECGR
        ELSEIF(NAMGRD.EQ.'SPC1') THEN
          CALL SPC1GR
        ELSEIF(NAMGRD.EQ.'SPC2') THEN
```

```
          CALL SPC2GR
        ELSEIF(NAMGRD.EQ.'SPC3') THEN
          CALL SPC3GR
        ELSEIF(NAMGRD.EQ.'GHOL') THEN
          CALL HOLGR
        ELSEIF(NAMGRD.EQ.'RSTM') THEN
          CALL RSTMGR
        ELSEIF(NAMGRD.EQ.'COAL') THEN
          CALL COALGR
        ELSEIF(NAMGRD.EQ.'RSTR') THEN
          CALL RSTRGR
        ELSEIF(NAMGRD.EQ.'SURF') THEN
          CALL SURFGR
        ELSEIF(NAMGRD.EQ.'QUIK') THEN
          CALL QUIKGR
        ELSEIF(NAMGRD.EQ.'STRA') THEN
          CALL STRAGR
        ELSEIF(NAMGRD.EQ.'RADI') THEN
          CALL RADIGR
        ELSEIF(NAMGRD.EQ.'ESTR') THEN
          CALL ESTRGR
        ELSEIF(NAMGRD.EQ.'LINK') THEN
          CALL LINKGR
        ELSE
          CALL WRITBL
          CALL WRITST
          CALL WRIT40('NAMGRD set but no CALL made, ie.        ')
          CALL WRIT1A('NAMGRD  ',NAMGRD)
          CALL WRIT40(' sections 9 and 10 added to Group 19   ')
          CALL WRIT40(' Permissible calls for this GROSTA are:-')
          CALL WRIT40(' SPEC, SPC1, SPC2, SPC3, GHOL, RSTM,    ')
          CALL WRIT40(' RSTR, SURF, QUIK, STRA, LINK           ')
          CALL WRIT40(' Use upper-case names only              ')
          CALL WRITST
          CALL WRITBL
          CALL WAYOUT(2)
        ENDIF
      ENDIF
C
C.... The subroutine GROUND attached to the bottom of this file is
C     an unallocated blank form into which the user can insert his
C     own FORTRAN sequences. The PIL parameter USEGRD governs entry
C     to it.
C
      IF(USEGRD) CALL GROUND
C
C.... The data "echo" is called at the preliminary print-out stage.
      IF(IGR.EQ.20) THEN
        IF(ECHO) THEN
          CALL DATPRN(Y,Y,Y,Y,  Y,Y,Y,Y,  Y,Y,Y,N,  Y,Y,Y,Y,
     1                Y,Y,Y,Y,  Y,Y,Y,Y)
        ELSE
          CALL DATPRN(Y,N,N,N,N,N,N,N,N,N,N,N,N,N,N,N,N,N,N,N,N,N,N,N)
        ENDIF
      ENDIF
      END
C****************************************************************
      SUBROUTINE SPECGR
      CALL WRIT40('Dummy subroutine SPECGR  called.        ')
      CALL WAYOUT(2)
      END
C****************************************************************
      SUBROUTINE SPC1GR
      CALL WRIT40('Dummy subroutine SPC1GR called.         ')
      CALL WAYOUT(2)
      END
```

```
C****************************************************************
      SUBROUTINE SPC2GR
      CALL WRIT40('Dummy subroutine SPC2GR called.         ')
      CALL WAYOUT(2)
      END
C****************************************************************
      SUBROUTINE SPC3GR
      CALL WRIT40('Dummy subroutine SPC3GR called.         ')
      CALL WAYOUT(2)
      END
C****************************************************************
      SUBROUTINE QUIZ
C--- This subroutine is used by CHAM for de-bugging
      END
```

```
C*******************************************************************
      SUBROUTINE GROUND
      INCLUDE 'lp16/d_earth/SATEAR'
      INCLUDE 'lp16/d_earth/GRDLOC'
      INCLUDE 'lp16/d_earth/GRDEAR'
      INCLUDE 'lp16/d_earth/GRDBFC'
CXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX USER SECTION STARTS:
C
C 1   Set dimensions of data-for-GROUND arrays here. WARNING: the
C     corresponding arrays in the MAIN program of the satellite
C     and EARTH must have the same dimensions.
      PARAMETER (NLG=20, NIG=20, NRG=100, NCG=10)
C
      COMMON/LGRND/LG(NLG)/IGRND/IG(NIG)/RGRND/RG(NRG)/CGRND/CG(NCG)
      LOGICAL LG
      CHARACTER*4 CG
      COMMON/GUGRND/GCSN(60,80),GCSS(60,80),GCSE(60,80),GCSW(60,80),
     7              GCSL(60,80),GCSH(60,80),GSPCO(60,80),GSECO(60,80),
     8              GCUN(60,80),GCUS(60,80),GCUE(60,80),GCUW(60,80),
     7              GCUL(60,80),GCUH(60,80),GUPCO(60,80),GUECO(60,80),
     8              GCVN(60,80),GCVS(60,80),GCVE(60,80),GCVW(60,80),
     7              GCVL(60,80),GCVH(60,80),GVPCO(60,80),GVECO(60,80),
     8              GCWN(60,80),GCWS(60,80),GCWE(60,80),GCWW(60,80),
     7              GCWL(60,80),GCWH(60,80),GWPCO(60,80),GWECO(60,80)
      COMMON/IUGRND/ISPVAL(60,80),ISEVAL(60,80),IUPVAL(60,80),
     7              IUEVAL(60,80),IVEVAL(60,80),IVPVAL(60,80),
     8              IWEVAL(60,80),IWPVAL(60,80)
C
C 2   User dimensions own arrays here, for example:
C     DIMENSION GUH(10,10),GUC(10,10),GUX(10,10),GUZ(10)
      DIMENSION GCC(60,80),GCMH(60,80),GCML(60,80),
     1          GCE(60,80),GEMP(60,80),GEMH(60,80),GEML(60,80),
     2          GAR2(40,80),GAR3(40,80),GBET(40,80),GBLK(40,80),
     3          GPOR(40,80),GAR1(40,80),GDIS(40,80),GRI(40,80)
C
C 3   User places his data statements here, for example:
      DATA NXDIM,NYDIM/80,60/
C
C 4   Insert own coding below as desired, guided by GREX examples.
C     Note that the satellite-to-GREX special data in the labelled
C     COMMONs /RSG/, /ISG/, /LSG/ and /CSG/ can be included and
C     used below but the user must check GREX for any conflicting
C     uses. The same comment applies to the EARTH-spare working
C     arrays EASP1, EASP2,....EASP20. In addition to the EASPs,
C     there are 10 GRound-earth SPare arrays, GRSP1,...,GRSP10,
C     supplied solely for the user, which are not used by GREX. If
C     the call to GREX has been deactivated then all of the arrays
C     may be used without reservation.
C
      IXL=IABS(IXL)
      IF(IGR.EQ.13) GO TO 13
      IF(IGR.EQ.19) GO TO 19
      GO TO (1,2,3,4,5,6,25,8,9,10,11,12,13,14,25,25,25,25,19,20,25,
     125,23,24),IGR
   25 CONTINUE
      RETURN
C*******************************************************************
C
C--- GROUP 1. Run title and other preliminaries
C
    1 GO TO (1001,1002),ISC
 1001 CONTINUE
      CALL MAKE(XU2D)
      CALL MAKE(YV2D)
      CALL MAKE(XG2D)
      CALL MAKE(YG2D)
```

```
      CALL MAKE(DXU2D)
      CALL MAKE(DYV2D)
      CALL MAKE(GRSP1)
      CALL MAKE(GRSP2)
      CALL MAKE(GRSP3)
      CALL MAKE(GRSP4)
C
C     User may here change message transmitted to the VDU screen or
C     batch-run log file.
      IF(IGR.EQ.1.AND.ISC.EQ.1) THEN
        CALL WRYT40('GROUND file is GROUND.FTN of:    120691 ')
      ENDIF
C
      RETURN
 1002 CONTINUE
      DO 1010 IX=1,NX
      DO 1010 IY=1,NY
       ISPVAL(IY,IX)=0
       ISEVAL(IY,IX)=0
       IUPVAL(IY,IX)=0
       IUEVAL(IY,IX)=0
       IVPVAL(IY,IX)=0
       IVEVAL(IY,IX)=0
       IWPVAL(IY,IX)=0
       IWEVAL(IY,IX)=0
       GSPCO(IY,IX)=0.
       GSECO(IY,IX)=0.
       GUPCO(IY,IX)=0.
       GUECO(IY,IX)=0.
       GVPCO(IY,IX)=0.
       GVECO(IY,IX)=0.
       GWPCO(IY,IX)=0.
       GWECO(IY,IX)=0.
 1010 CONTINUE
      RETURN
C*********************************************************************
C
C--- GROUP 2. Transience; time-step specification
C
    2 CONTINUE
      RETURN
C*********************************************************************
C
C--- GROUP 3. X-direction grid specification
C
    3 CONTINUE
      RETURN
C*********************************************************************
C
C--- GROUP 4. Y-direction grid specification
C
    4 CONTINUE
      RETURN
C*********************************************************************
C
C--- GROUP 5. Z-direction grid specification
C
    5 CONTINUE
      RETURN
C*********************************************************************
C
C--- GROUP 6. Body-fitted coordinates or grid distortion
C
    6 CONTINUE
      RETURN
C*********************************************************************
```

```
C   * Make changes for this group only in group 19.
C--- GROUP 7. Variables stored, solved & named
C*****************************************************************
C
C--- GROUP 8. Terms (in differential equations) & devices
C
    8 GO TO (81,82,83,84,85,86,87,88,89,810,811,812,813,814,815)
     1,ISC
   81 CONTINUE
C   * ------------------- SECTION  1 ---------------------------
C    For U1AD.LE.GRND--- phase 1 additional velocity. Index VELAD
      RETURN
   82 CONTINUE
C   * ------------------- SECTION  2 ---------------------------
C    For U2AD.LE.GRND--- phase 2 additional velocity. Index VELAD
      RETURN
   83 CONTINUE
C   * ------------------- SECTION  3 ---------------------------
C    For V1AD.LE.GRND--- phase 1 additional velocity. Index VELAD
      RETURN
   84 CONTINUE
C   * ------------------- SECTION  4 ---------------------------
C    For V2AD.LE.GRND--- phase 2 additional velocity. Index VELAD
      RETURN
   85 CONTINUE
C   * ------------------- SECTION  5 ---------------------------
C    For W1AD.LE.GRND--- phase 1 additional velocity. Index VELAD
      RETURN
   86 CONTINUE
C   * ------------------- SECTION  6 ---------------------------
C    For W2AD.LE.GRND--- phase 2 additional velocity. Index VELAD
      RETURN
   87 CONTINUE
C   * ------------------- SECTION 7 ---- Volumetric source for gala
      RETURN
   88 CONTINUE
C   * ------------------- SECTION 8 ---- Convection fluxes
      CALL GROUND4
      CALL GROUND3
      CALL GROUND2
      CALL GROUND1
      RETURN
   89 CONTINUE
C   * ------------------- SECTION 9 ---- Diffusion coefficients
C   * -----------GROUP 8  SECTION 9 --- DIFFUSION COEFFICIENTS
C--- Entered when UDIFF =.TRUE.; block-location indices are LAE
C    for east, LAW for west, LAN for north, LAS for
C    south, LD11 for high, and LD11 for low.
C    User should provide INDVAR and NDIREC IF's as above.
C
      RETURN
  810 CONTINUE
C   * ------------------- SECTION 10 --- Convection neighbours
      RETURN
  811 CONTINUE
C   * ------------------- SECTION 11 --- Diffusion neighbours
      RETURN
  812 CONTINUE
C   * ------------------- SECTION 12 --- Linearised sources
      RETURN
  813 CONTINUE
C   * ------------------- SECTION 13 --- Correction coefficients
      RETURN
  814 CONTINUE
C   * ------------------- SECTION 14 --- User's solver
      RETURN
```

```
  815 CONTINUE
C   * ------------------- SECTION 15 --- Change solution
      RETURN
C
C   * See the equivalent section in GREX for the indices to be
C     used in sections 7 - 15
C
C   * Make all other group-8 changes in GROUP 19.
C*****************************************************************
C
C--- GROUP 9. Properties of the medium (or media)
C
C   The sections in this group are arranged sequentially in their
C   order of calling from EARTH. Thus, as can be seen from below,
C   the temperature sections (10 and 11) precede the density
C   sections (1 and 3); so, density formulae can refer to
C   temperature stores already set.
C
    9 GO TO (91,92,93,94,95,96,97,98,99,900,901,902,903),ISC
C*****************************************************************
C   * ------------------- SECTION 10 ---------------------------
C    For TMP1.LE.GRND--------- phase-1 temperature Index TEMP1
  900 CONTINUE
      RETURN
C   * ------------------- SECTION 11 ---------------------------
C    For TMP2.LE.GRND--------- phase-2 temperature Index TEMP2
  901 CONTINUE
      RETURN
C   * ------------------- SECTION 12 ---------------------------
C    For EL1.LE.GRND--------- phase-1 length scale Index LEN1
  902 CONTINUE
      RETURN
C   * ------------------- SECTION 13 ---------------------------
C    For EL2.LE.GRND--------- phase-2 length scale Index LEN2
  903 CONTINUE
      RETURN
C   * ------------------- SECTION  1 ---------------------------
C    For RHO1.LE.GRND--- density for phase 1       Index DEN1
   91 CONTINUE
      RETURN
C   * ------------------- SECTION  2 ---------------------------
C    For DRH1DP.LE.GRND--- D(LN(DEN))/DP for phase 1
C                                                  Index D1DP
   92 CONTINUE
      RETURN
C   * ------------------- SECTION  3 ---------------------------
C    For RHO2.LE.GRND--- density for phase 2       Index DEN2
   93 CONTINUE
      RETURN
C   * ------------------- SECTION  4 ---------------------------
C    For DRH2DP.LE.GRND--- D(LN(DEN))/DP for phase 2
C                                                  Index D2DP
   94 CONTINUE
      RETURN
C   * ------------------- SECTION  5 ---------------------------
C    For ENUT.LE.GRND--- reference turbulent kinematic viscosity
C                                                  Index VIST
   95 CONTINUE
      RETURN
C   * ------------------- SECTION  6 ---------------------------
C    For ENUL.LE.GRND--- reference laminar kinematic viscosity
   96 CONTINUE
C                                                  Index VISL
      RETURN
C   * ------------------- SECTION  7 ---------------------------
C    For PRNDTL( ).LE.GRND--- laminar PRANDTL nos., or diffusivity
```

```
C                                                   Index LAMPR
   97 CONTINUE
      RETURN
C   * ------------------- SECTION  8 ---------------------------
C    For PHINT( ).LE.GRND--- interface value of first phase
C                                                   Index FII1
   98 CONTINUE
      RETURN
C   * ------------------- SECTION  9 ---------------------------
C    For PHINT( ).LE.GRND--- interface value of second phase
C                                                   Index FII2
   99 CONTINUE
      RETURN
C******************************************************************
C
C--- GROUP 10. Inter-phase-transfer processes and properties
C
   10 GO TO (101,102,103,104),ISC
C   * ------------------- SECTION  1 ---------------------------
C    For CFIPS.LE.GRND--- inter-phase friction coeff.
C                                                   Index AUX(INTFRC)
  101 CONTINUE
      RETURN
C   * ------------------- SECTION  2 ---------------------------
C    For CMDOT.EQ.GRND- inter-phase mass transfer  Index AUX(INTMDT)
  102 CONTINUE
      RETURN
C   * ------------------- SECTION  3 ---------------------------
C    For CINT( ).EQ.GRND--- phase1-to-interface transfer coefficients
C                                                   Index COI1
  103 CONTINUE
      RETURN
C   * ------------------- SECTION  4 ---------------------------
C    For CINT( ).EQ.GRND--- phase2-to-interface transfer coefficients
C                                                   Index COI2
  104 CONTINUE
      RETURN
C******************************************************************
C
C--- GROUP 11. Initialization of variable or porosity fields
C                                                   Index VAL
   11 CONTINUE
C
C  ****************************************************************
C      Calculation of porosity field for linear bluff blockage
C  ****************************************************************
C
      SETPOR=.TRUE.
      IF(NPATCH.NE.'BLUPOR') RETURN
      LOVAL=L0F(VAL)
      LOX=L0F(XU2D)
      LOY=L0F(YV2D)
      LOXG=L0F(XG2D)
      LOYG=L0F(YG2D)
      LODX=L0F(DXU2D)
      LODY=L0F(DYV2D)
C
      DO 11050 IX=1,NX
      DO 11050 IY=1,NY
       ICELL=IY+NY*(IX-1)
        IF(INDVAR.EQ.VPOR) F(LOVAL+ICELL)=1.0
        IF(INDVAR.EQ.EPOR) F(LOVAL+ICELL)=1.0
        IF(INDVAR.EQ.NPOR) F(LOVAL+ICELL)=1.0
        IF(INDVAR.EQ.HPOR) F(LOVAL+ICELL)=1.0
C       WRITE(6,*) 'THETA = ',F(LOX+ICELL)
C       WRITE(6,*) 'RADIAL = ',F(LOY+ICELL)
```

```
C
      IF(F(LOX+ICELL).GE.4.71239.AND.
     1 F(LOX+ICELL).LE.6.28319) GO TO 11035
C
      IF(F(LOX+ICELL).GE.3.14159.AND.
     2 F(LOX+ICELL).LT.4.71239) GO TO 11022
C
      IF(F(LOX+ICELL).GE.1.57080.AND.
     3 F(LOX+ICELL).LT.3.14159) GO TO 11009
C
      GRI(IY,IX)=F(LOYG+ICELL)*COS(F(LOXG+ICELL))
C      IF(ABS(RG(1)-GRI(IY,IX)).LT.0.00035) GO TO 11002
C
      IF(ABS(RG(1)-GRI(IY,IX)).GT.0.00035.AND.
     4 (RG(1)-GRI(IY,IX)).GT.0.0)  GO TO 11007
C
      IF(ABS(RG(1)-GRI(IY,IX)).GT.0.00035.AND.
     4 (RG(1)-GRI(IY,IX)).LT.0.0)  GO TO 11006
C
11002   GAR1(IY,IX)=(F(LOY+ICELL)**2.0/2.0)*
     1              (F(LODX+ICELL)-SIN(F(LODX+ICELL)))
C         WRITE(6,*) 'IZ =',IZ
C         WRITE(6,*) 'IX =',IX
C         WRITE(6,*) 'IY =',IY
C         WRITE(6,*) 'GAR1 =' ,GAR1(IY,IX)
        GBET(IY,IX)=0.5*(3.14159+F(LODX+ICELL))
        GDIS(IY,IX)=2.0*F(LOY+ICELL)*SIN(F(LODX+ICELL)/2.0)
        GAR2(IY,IX)=0.5*GDIS(IY,IX)*F(LODY+ICELL)*
     1              SIN(GBET(IY,IX))
        GBLK(IY,IX)=GAR2(IY,IX)-GAR1(IY,IX)
        GAR3(IY,IX)=0.5*F(LODX+ICELL)*(F(LODY+ICELL)**2.0+
     2              (2.0*F(LOY+ICELL)*F(LODY+ICELL)))
        GPOR(IY,IX)=GBLK(IY,IX)/GAR3(IY,IX)
       IF(INDVAR.EQ.HPOR) F(LOVAL+ICELL)=GPOR(IY,IX)
       IF(INDVAR.EQ.NPOR) F(LOVAL+ICELL)=GPOR(IY,IX)
       IF(INDVAR.EQ.VPOR) F(LOVAL+ICELL)=GPOR(IY,IX)
       GO TO 11008
C
11006  IF(INDVAR.EQ.HPOR) F(LOVAL+ICELL)=1.0
       IF(INDVAR.EQ.NPOR) F(LOVAL+ICELL)=1.0
       IF(INDVAR.EQ.EPOR) F(LOVAL+ICELL)=1.0
       IF(INDVAR.EQ.VPOR) F(LOVAL+ICELL)=1.0
       GO TO 11008
C
11007  IF(INDVAR.EQ.HPOR) F(LOVAL+ICELL)=0.
       IF(INDVAR.EQ.NPOR) F(LOVAL+ICELL)=0.
       IF(INDVAR.EQ.EPOR) F(LOVAL+ICELL)=0.
       IF(INDVAR.EQ.VPOR) F(LOVAL+ICELL)=0.
11008  CONTINUE
C
11009 GRI(IY,IX)=F(LOYG+ICELL)*ABS(COS(F(LOXG+ICELL)))
      IF(ABS(RG(1)-GRI(IY,IX)).LT.0.00035) GO TO 11015
C
      IF(ABS(RG(1)-GRI(IY,IX)).GT.0.00035.AND.
     4 (RG(1)-GRI(IY,IX)).GT.0.0)  GO TO 11020
C
      IF(ABS(RG(1)-GRI(IY,IX)).GT.0.00035.AND.
     4 (RG(1)-GRI(IY,IX)).LT.0.0)  GO TO 11019
C
11015  GAR1(IY,IX)=(F(LOY+ICELL)**2.0/2.0)*
     1              (F(LODX+ICELL)-SIN(F(LODX+ICELL)))
        GBET(IY,IX)=0.5*(3.14159+F(LODX+ICELL))
        GDIS(IY,IX)=2.0*F(LOY+ICELL)*SIN(F(LODX+ICELL)/2.0)
        GAR2(IY,IX)=0.5*GDIS(IY,IX)*F(LODY+ICELL)*
     1              SIN(GBET(IY,IX))
        GBLK(IY,IX)=GAR2(IY,IX)-GAR1(IY,IX)
```

```
         GAR3(IY,IX)=0.5*F(LODX+ICELL)*(F(LODY+ICELL)**2.0+
     2               (2.0*F(LOY+ICELL)*F(LODY+ICELL)))
         GPOR(IY,IX)=GBLK(IY,IX)/GAR3(IY,IX)
        IF(INDVAR.EQ.HPOR) F(LOVAL+ICELL)=GPOR(IY,IX)
        IF(INDVAR.EQ.NPOR) F(LOVAL+ICELL)=GPOR(IY,IX)
        IF(INDVAR.EQ.EPOR) F(LOVAL+ICELL)=GPOR(IY,IX)
        IF(INDVAR.EQ.VPOR) F(LOVAL+ICELL)=GPOR(IY,IX)
        GO TO 11021
C
11019   IF(INDVAR.EQ.HPOR) F(LOVAL+ICELL)=1.0
        IF(INDVAR.EQ.NPOR) F(LOVAL+ICELL)=1.0
        IF(INDVAR.EQ.EPOR) F(LOVAL+ICELL)=1.0
        IF(INDVAR.EQ.VPOR) F(LOVAL+ICELL)=1.0
        GO TO 11021
C
11020   IF(INDVAR.EQ.HPOR) F(LOVAL+ICELL)=0.
        IF(INDVAR.EQ.NPOR) F(LOVAL+ICELL)=0.
        IF(INDVAR.EQ.EPOR) F(LOVAL+ICELL)=0.
        IF(INDVAR.EQ.VPOR) F(LOVAL+ICELL)=0.
11021   CONTINUE
C
11022   GRI(IY,IX)=F(LOYG+ICELL)*ABS(COS(F(LOXG+ICELL)))
        IF(ABS(RG(1)-GRI(IY,IX)).LT.0.00035) GO TO 11023
C
       IF(ABS(RG(1)-GRI(IY,IX)).GT.0.00035.AND.
     4 (RG(1)-GRI(IY,IX)).GT.0.0)  GO TO 11033
C
       IF(ABS(RG(1)-GRI(IY,IX)).GT.0.00035.AND.
     4 (RG(1)-GRI(IY,IX)).LT.0.0)  GO TO 11032
C
11023   GAR1(IY,IX)=(F(LOY+ICELL)**2.0/2.0)*
     1              (F(LODX+ICELL)-SIN(F(LODX+ICELL)))
         GBET(IY,IX)=0.5*(3.14159+F(LODX+ICELL))
         GDIS(IY,IX)=2.0*F(LOY+ICELL)*SIN(F(LODX+ICELL)/2.0)
         GAR2(IY,IX)=0.5*GDIS(IY,IX)*F(LODY+ICELL)*
     1               SIN(GBET(IY,IX))
         GBLK(IY,IX)=GAR2(IY,IX)-GAR1(IY,IX)
         GAR3(IY,IX)=0.5*F(LODX+ICELL)*(F(LODY+ICELL)**2.0+
     2               (2.0*F(LOY+ICELL)*F(LODY+ICELL)))
         GPOR(IY,IX)=GBLK(IY,IX)/GAR3(IY,IX)
        IF(INDVAR.EQ.HPOR) F(LOVAL+ICELL)=GPOR(IY,IX)
        IF(INDVAR.EQ.NPOR) F(LOVAL+ICELL)=GPOR(IY,IX)
        IF(INDVAR.EQ.VPOR) F(LOVAL+ICELL)=GPOR(IY,IX)
        GO TO 11034
C
11032   IF(INDVAR.EQ.HPOR) F(LOVAL+ICELL)=1.0
        IF(INDVAR.EQ.NPOR) F(LOVAL+ICELL)=1.0
        IF(INDVAR.EQ.EPOR) F(LOVAL+ICELL)=1.0
        IF(INDVAR.EQ.VPOR) F(LOVAL+ICELL)=1.0
        GO TO 11034
C
11033   IF(INDVAR.EQ.HPOR) F(LOVAL+ICELL)=0.
        IF(INDVAR.EQ.NPOR) F(LOVAL+ICELL)=0.
        IF(INDVAR.EQ.EPOR) F(LOVAL+ICELL)=0.
        IF(INDVAR.EQ.VPOR) F(LOVAL+ICELL)=0.
11034   CONTINUE
C
11035   GRI(IY,IX)=F(LOYG+ICELL)*ABS(COS(F(LOXG+ICELL)))
        IF(ABS(RG(1)-GRI(IY,IX)).LT.0.00035) GO TO 11036
C
       IF(ABS(RG(1)-GRI(IY,IX)).GT.0.00035.AND.
     4 (RG(1)-GRI(IY,IX)).GT.0.0)  GO TO 11047
C
       IF(ABS(RG(1)-GRI(IY,IX)).GT.0.00035.AND.
     4 (RG(1)-GRI(IY,IX)).LT.0.0)  GO TO 11046
C
```

```
C
11036   GAR1(IY,IX)=(F(LOY+ICELL)**2.0/2.0)*
     1                 (F(LODX+ICELL)-SIN(F(LODX+ICELL)))
         GBET(IY,IX)=0.5*(3.14159+F(LODX+ICELL))
         GDIS(IY,IX)=2.0*F(LOY+ICELL)*SIN(F(LODX+ICELL)/2.0)
         GAR2(IY,IX)=0.5*GDIS(IY,IX)*F(LODY+ICELL)*
     1                 SIN(GBET(IY,IX))
         GBLK(IY,IX)=GAR2(IY,IX)-GAR1(IY,IX)
         GAR3(IY,IX)=0.5*F(LODX+ICELL)*(F(LODY+ICELL)**2.0+
     2                 (2.0*F(LOY+ICELL)*F(LODY+ICELL)))
         GPOR(IY,IX)=GBLK(IY,IX)/GAR3(IY,IX)
        IF(INDVAR.EQ.HPOR) F(LOVAL+ICELL)=GPOR(IY,IX)
        IF(INDVAR.EQ.NPOR) F(LOVAL+ICELL)=GPOR(IY,IX)
        IF(INDVAR.EQ.EPOR) F(LOVAL+ICELL)=GPOR(IY,IX)
        IF(INDVAR.EQ.VPOR) F(LOVAL+ICELL)=GPOR(IY,IX)
        GO TO 11048
C
11046   IF(INDVAR.EQ.HPOR) F(LOVAL+ICELL)=1.0
        IF(INDVAR.EQ.NPOR) F(LOVAL+ICELL)=1.0
        IF(INDVAR.EQ.EPOR) F(LOVAL+ICELL)=1.0
        IF(INDVAR.EQ.VPOR) F(LOVAL+ICELL)=1.0
        GO TO 11048
C
11047   IF(INDVAR.EQ.HPOR) F(LOVAL+ICELL)=0.
        IF(INDVAR.EQ.NPOR) F(LOVAL+ICELL)=0.
        IF(INDVAR.EQ.EPOR) F(LOVAL+ICELL)=0.
        IF(INDVAR.EQ.VPOR) F(LOVAL+ICELL)=0.
11048   CONTINUE
11050 CONTINUE
      RETURN
C*********************************************************************
C
C--- GROUP 12. Convection and diffusion adjustments
C
   12 CONTINUE
      RETURN
C*********************************************************************
C
C--- GROUP 13. Boundary conditions and special sources
C                                        Index for Coefficient - CO
C                                        Index for Value       - VAL
   13 GO TO (130,131,132,133,134,135,136,137,138,139,1310,
     11311,1312,1313,1314,1315,1316,1317,1318,1319,1320,1321),ISC
C------------------- SECTION  1 ------------- coefficient = GRND
  130 CONTINUE
C
C  Scalar Variables - CORNER source.
C
C      CALL WRIT1I('INDVAR  ',INDVAR)
      IF(NPATCH.NE.'SUCCAC'.OR.INDVAR.LT.9) GO TO 13020
      CALL ONLYIF(12,50,'SUCCAC')
      CALL SETYX(CO,GSPCO,NYDIM,NXDIM)
C      CALL PRN('COSC',CO)
C
13020 CONTINUE
C
C  U-Momentum Variable - CORNER source
C
C      CALL WRIT1I('INDVAR  ',INDVAR)
      IF(NPATCH.NE.'SUCCAC'.OR.INDVAR.NE.3) GO TO 13030
      CALL ONLYIF(3,3,'SUCCAC')
      CALL SETYX(CO,GUPCO,NYDIM,NXDIM)
C      CALL PRN('COUC',CO)
C
13030 CONTINUE
C
```

```
C  V-Momentum Variable - CORNER source
C
C       CALL WRIT1I('INDVAR  ',INDVAR)
      IF(NPATCH.NE.'SUCCAC'.OR.INDVAR.NE.5) GO TO 13040
      CALL ONLYIF(5,5,'SUCCAC')
      CALL SETYX(CO,GVPCO,NYDIM,NXDIM)
C       CALL PRN('COVC',CO)
C
13040 CONTINUE
C
C  W-Momentum Variable - CORNER source
C
C       CALL WRIT1I('INDVAR  ',INDVAR)
      IF(NPATCH.NE.'SUCCAC'.OR.INDVAR.NE.7) GO TO 13050
      CALL ONLYIF(7,7,'SUCCAC')
      CALL SETYX(CO,GWPCO,NYDIM,NXDIM)
C       CALL PRN('COWC',CO)
C
13050 RETURN
C
C------------------ SECTION  2 ------------ coefficient = GRND1
  131 CONTINUE
C
C  Scalar Variables - EDGE source.
C
C       CALL WRIT1I('INDVAR  ',INDVAR)
      IF(NPATCH.NE.'SUCCAE'.OR.INDVAR.LT.9) GO TO 13101
      CALL ONLYIF(12,50,'SUCCAE')
      CALL SETYX(CO,GSECO,NYDIM,NXDIM)
C       CALL PRN('COSE',CO)
C
13101 CONTINUE
C
C  U-Momentum Variable - EDGE source.
C
C       CALL WRIT1I('INDVAR  ',INDVAR)
      IF(NPATCH.NE.'SUCCAE'.OR.INDVAR.NE.3) GO TO 13102
      CALL ONLYIF(3,3,'SUCCAE')
      CALL SETYX(CO,GUECO,NYDIM,NXDIM)
C       CALL PRN('COUE',CO)
C
13102 CONTINUE
C
C  V-Momentum Variable - EDGE source.
C
C       CALL WRIT1I('INDVAR  ',INDVAR)
      IF(NPATCH.NE.'SUCCAE'.OR.INDVAR.NE.5) GO TO 13103
      CALL ONLYIF(5,5,'SUCCAE')
      CALL SETYX(CO,GVECO,NYDIM,NXDIM)
C       CALL PRN('COVE',CO)
C
13103 CONTINUE
C
C  W-Momentum Variable - EDGE source.
C
C       CALL WRIT1I('INDVAR  ',INDVAR)
      IF(NPATCH.NE.'SUCCAE'.OR.INDVAR.NE.7) GO TO 13104
      CALL ONLYIF(7,7,'SUCCAE')
      CALL SETYX(CO,GWECO,NYDIM,NXDIM)
C       CALL PRN('COWE',CO)
C
13104 RETURN
C
C------------------ SECTION  3 ------------ coefficient = GRND2
  132 CONTINUE
      RETURN
```

```
C------------------- SECTION  4 ------------- coefficient = GRND3
  133 CONTINUE
      RETURN
C------------------- SECTION  5 ------------- coefficient = GRND4
  134 CONTINUE
      RETURN
C------------------- SECTION  6 ------------- coefficient = GRND5
  135 CONTINUE
      RETURN
C------------------- SECTION  7 ------------- coefficient = GRND6
  136 CONTINUE
      RETURN
C------------------- SECTION  8 ------------- coefficient = GRND7
  137 CONTINUE
      RETURN
C------------------- SECTION  9 ------------- coefficient = GRND8
  138 CONTINUE
      RETURN
C------------------- SECTION 10 ------------- coefficient = GRND9
  139 CONTINUE
      RETURN
C------------------- SECTION 11 ------------- coefficient = GRND10
 1310 CONTINUE
      RETURN
C------------------- SECTION 12 ------------------- value = GRND
 1311 CONTINUE
C
C
C  Values of INDVAR for convection from upwind cell - CORNERS.
C  ***********************************************************
C
C  ****************
C  Scalar Variables
C  ****************
C
      IF(NPATCH.NE.'SUCCAC'.OR.INDVAR.LT.9) GO TO 13116
      CALL ONLYIF(12,50,'SUCCAC')
C
C       DO 13110 IX=1,NX
C       DO 13110 IY=1,NY
C       WRITE(6,*) 'ISWEEP= ',ISWEEP
C       WRITE(6,*) 'INDVAR= ',INDVAR
C       WRITE(6,*) 'IZ= ',IZ
C       WRITE(6,*) 'IX= ',IX
C       WRITE(6,*) 'IY= ',IY
C       WRITE(6,*) 'ISPVAL= ',ISPVAL(IY,IX)
C 13110 CONTINUE
C
      CALL GETYX(HIGH(INDVAR),GCMH,NYDIM,NXDIM)
      CALL GETYX(LOW(INDVAR),GCML,NYDIM,NXDIM)
C
      DO 13113 IX=1,NX
      DO 13113 IY=1,NY
      IF(ISPVAL(IY,IX).EQ.0) GO TO 13113
      IF(ISPVAL(IY,IX).GT.4) GO TO 13111
C ***********************************************************
      IF(ISPVAL(IY,IX).EQ.1) GCC(IY,IX)=GCMH(IY+1,IX-1)
      IF(ISPVAL(IY,IX).EQ.2) GCC(IY,IX)=GCMH(IY+1,IX+1)
      IF(ISPVAL(IY,IX).EQ.3) GCC(IY,IX)=GCMH(IY-1,IX-1)
      IF(ISPVAL(IY,IX).EQ.4) GCC(IY,IX)=GCMH(IY-1,IX+1)
      GO TO 13113
C ***********************************************************
13111 IF(ISPVAL(IY,IX).EQ.5) GCC(IY,IX)=GCML(IY+1,IX-1)
      IF(ISPVAL(IY,IX).EQ.6) GCC(IY,IX)=GCML(IY+1,IX+1)
      IF(ISPVAL(IY,IX).EQ.7) GCC(IY,IX)=GCML(IY-1,IX-1)
      IF(ISPVAL(IY,IX).EQ.8) GCC(IY,IX)=GCML(IY-1,IX+1)
```

```
13113 CONTINUE
      CALL SETYX(VAL,GCC,NYDIM,NXDIM)
      DO 13114 IX=1,NX
      DO 13114 IY=1,NY
        ISPVAL(IY,IX)=0
13114 CONTINUE
C       CALL PRN('VASC',VAL)
13116 CONTINUE
C
C
C
C  *******************
C  U-Momentum Variable
C  *******************
C
      IF(NPATCH.NE.'SUCCAC'.OR.INDVAR.NE.3) GO TO 13124
      CALL ONLYIF(3,3,'SUCCAC')
C
C       DO 13118 IX=1,NX-1
C       DO 13118 IY=1,NY
C       WRITE(6,*) 'ISWEEP= ',ISWEEP
C       WRITE(6,*) 'INDVAR= ',INDVAR
C       WRITE(6,*) 'IZ= ',IZ
C       WRITE(6,*) 'IX= ',IX
C       WRITE(6,*) 'IY= ',IY
C       WRITE(6,*) 'IUPVAL= ',IUPVAL(IY,IX)
C 13118 CONTINUE
C
      CALL GETYX(HIGH(INDVAR),GCMH,NYDIM,NXDIM)
      CALL GETYX(LOW(INDVAR),GCML,NYDIM,NXDIM)
C
      DO 13121 IX=1,NX-1
      DO 13121 IY=1,NY
      IF(IUPVAL(IY,IX).EQ.0) GO TO 13121
      IF(IUPVAL(IY,IX).GT.4) GO TO 13119
C *********************************************************
      IF(IUPVAL(IY,IX).EQ.1) GCC(IY,IX)=GCMH(IY+1,IX-1)
      IF(IUPVAL(IY,IX).EQ.2) GCC(IY,IX)=GCMH(IY+1,IX+1)
      IF(IUPVAL(IY,IX).EQ.3) GCC(IY,IX)=GCMH(IY-1,IX-1)
      IF(IUPVAL(IY,IX).EQ.4) GCC(IY,IX)=GCMH(IY-1,IX+1)
      GO TO 13121
C *********************************************************
13119 IF(IUPVAL(IY,IX).EQ.5) GCC(IY,IX)=GCML(IY+1,IX-1)
      IF(IUPVAL(IY,IX).EQ.6) GCC(IY,IX)=GCML(IY+1,IX+1)
      IF(IUPVAL(IY,IX).EQ.7) GCC(IY,IX)=GCML(IY-1,IX-1)
      IF(IUPVAL(IY,IX).EQ.8) GCC(IY,IX)=GCML(IY-1,IX+1)
13121 CONTINUE
      CALL SETYX(VAL,GCC,NYDIM,NXDIM)
      DO 13122 IX=1,NX-1
      DO 13122 IY=1,NY
        IUPVAL(IY,IX)=0
13122 CONTINUE
C       CALL PRN('VAUC',VAL)
13124 CONTINUE
C
C  *******************
C  V-Momentum Variable
C  *******************
C
      IF(NPATCH.NE.'SUCCAC'.OR.INDVAR.NE.5) GO TO 13134
      CALL ONLYIF(5,5,'SUCCAC')
C
C       DO 13126 IX=1,NX
C       DO 13126 IY=1,NY-1
C       WRITE(6,*) 'ISWEEP= ',ISWEEP
C       WRITE(6,*) 'INDVAR= ',INDVAR
```

```
C       WRITE(6,*) 'IZ= ',IZ
C       WRITE(6,*) 'IX= ',IX
C       WRITE(6,*) 'IY= ',IY
C       WRITE(6,*) 'IVPVAL= ',IVPVAL(IY,IX)
C 13126 CONTINUE
C
      CALL GETYX(HIGH(INDVAR),GCMH,NYDIM,NXDIM)
      CALL GETYX(LOW(INDVAR),GCML,NYDIM,NXDIM)
C
      DO 13131 IX=1,NX
      DO 13131 IY=1,NY-1
      IF(IVPVAL(IY,IX).EQ.0) GO TO 13131
      IF(IVPVAL(IY,IX).GT.4) GO TO 13129
C **********************************************************
      IF(IVPVAL(IY,IX).EQ.1) GCC(IY,IX)=GCMH(IY+1,IX-1)
      IF(IVPVAL(IY,IX).EQ.2) GCC(IY,IX)=GCMH(IY+1,IX+1)
      IF(IVPVAL(IY,IX).EQ.3) GCC(IY,IX)=GCMH(IY-1,IX-1)
      IF(IVPVAL(IY,IX).EQ.4) GCC(IY,IX)=GCMH(IY-1,IX+1)
      GO TO 13131
C **********************************************************
13129 IF(IVPVAL(IY,IX).EQ.5) GCC(IY,IX)=GCML(IY+1,IX-1)
      IF(IVPVAL(IY,IX).EQ.6) GCC(IY,IX)=GCML(IY+1,IX+1)
      IF(IVPVAL(IY,IX).EQ.7) GCC(IY,IX)=GCML(IY-1,IX-1)
      IF(IVPVAL(IY,IX).EQ.8) GCC(IY,IX)=GCML(IY-1,IX+1)
13131 CONTINUE
      CALL SETYX(VAL,GCC,NYDIM,NXDIM)
      DO 13132 IX=1,NX
      DO 13132 IY=1,NY-1
        IVPVAL(IY,IX)=0
13132 CONTINUE
C       CALL PRN('VAVC',VAL)
13134 CONTINUE
C
C
C   *******************
C   W-Momentum Variable
C   W-equation only staggered in Z-direction.
C   *******************
C
      IF(NPATCH.NE.'SUCCAC'.OR.INDVAR.NE.7) GO TO 13142
      CALL ONLYIF(7,7,'SUCCAC')
C
C       DO 13136 IX=1,NX
C       DO 13136 IY=1,NY
C       WRITE(6,*) 'ISWEEP= ',ISWEEP
C       WRITE(6,*) 'INDVAR= ',INDVAR
C       WRITE(6,*) 'IZ= ',IZ
C       WRITE(6,*) 'IX= ',IX
C       WRITE(6,*) 'IY= ',IY
C       WRITE(6,*) 'IWPVAL= ',IWPVAL(IY,IX)
C 13136 CONTINUE
C
      CALL GETYX(HIGH(INDVAR),GCMH,NYDIM,NXDIM)
      CALL GETYX(LOW(INDVAR),GCML,NYDIM,NXDIM)
C
      DO 13139 IX=1,NX
      DO 13139 IY=1,NY
      IF(IWPVAL(IY,IX).EQ.0) GO TO 13139
      IF(IWPVAL(IY,IX).GT.4) GO TO 13137
C **********************************************************
      IF(IWPVAL(IY,IX).EQ.1) GCC(IY,IX)=GCMH(IY+1,IX-1)
      IF(IWPVAL(IY,IX).EQ.2) GCC(IY,IX)=GCMH(IY+1,IX+1)
      IF(IWPVAL(IY,IX).EQ.3) GCC(IY,IX)=GCMH(IY-1,IX-1)
      IF(IWPVAL(IY,IX).EQ.4) GCC(IY,IX)=GCMH(IY-1,IX+1)
      GO TO 13139
C **********************************************************
```

```
13137 IF(IWPVAL(IY,IX).EQ.5) GCC(IY,IX)=GCML(IY+1,IX-1)
      IF(IWPVAL(IY,IX).EQ.6) GCC(IY,IX)=GCML(IY+1,IX+1)
      IF(IWPVAL(IY,IX).EQ.7) GCC(IY,IX)=GCML(IY-1,IX-1)
      IF(IWPVAL(IY,IX).EQ.8) GCC(IY,IX)=GCML(IY-1,IX+1)
13139 CONTINUE
      CALL SETYX(VAL,GCC,NYDIM,NXDIM)
      DO 13140 IX=1,NX
      DO 13140 IY=1,NY
        IWPVAL(IY,IX)=0
13140 CONTINUE
C       CALL PRN('VAWC',VAL)
13142 CONTINUE
      RETURN
C------------------- SECTION 13 ------------------- value = GRND1
 1312 CONTINUE
C
C  Values of INDVAR for convection from upwind cell - EDGES.
C  *************************************************************
C
C  ****************
C  Scalar Variables
C  ****************
C
      IF(NPATCH.NE.'SUCCAE'.OR.INDVAR.LT.9) GO TO 13156
      CALL ONLYIF(12,50,'SUCCAE')
C
C      DO 13130 IX=1,NX
C      DO 13130 IY=1,NY
C      WRITE(6,*) 'ISWEEP= ',ISWEEP
C      WRITE(6,*) 'INDVAR= ',INDVAR
C      WRITE(6,*) 'IZ= ',IZ
C      WRITE(6,*) 'IX= ',IX
C      WRITE(6,*) 'IY= ',IY
C      WRITE(6,*) 'ISEVAL= ',ISEVAL(IY,IX)
C 13130 CONTINUE
C
      CALL GETYX(INDVAR,GEMP,NYDIM,NXDIM)
      CALL GETYX(HIGH(INDVAR),GEMH,NYDIM,NXDIM)
      CALL GETYX(LOW(INDVAR),GEML,NYDIM,NXDIM)
C
      DO 13153 IX=1,NX
      DO 13153 IY=1,NY
      IF(ISEVAL(IY,IX).EQ.0) GO TO 13153
      IF(ISEVAL(IY,IX).GT.4) GO TO 13151
C ***********************************************************
      IF(ISEVAL(IY,IX).EQ.1) GCE(IY,IX)=GEMP(IY+1,IX-1)
      IF(ISEVAL(IY,IX).EQ.2) GCE(IY,IX)=GEMP(IY+1,IX+1)
      IF(ISEVAL(IY,IX).EQ.3) GCE(IY,IX)=GEMP(IY-1,IX-1)
      IF(ISEVAL(IY,IX).EQ.4) GCE(IY,IX)=GEMP(IY-1,IX+1)
      GO TO 13153
C ***********************************************************
13151 IF(ISEVAL(IY,IX).GT.8) GO TO 13152
      IF(ISEVAL(IY,IX).EQ.5) GCE(IY,IX)=GEMH(IY+1,IX)
      IF(ISEVAL(IY,IX).EQ.6) GCE(IY,IX)=GEMH(IY-1,IX)
      IF(ISEVAL(IY,IX).EQ.7) GCE(IY,IX)=GEMH(IY,IX+1)
      IF(ISEVAL(IY,IX).EQ.8) GCE(IY,IX)=GEMH(IY,IX-1)
      GO TO 13153
C ***********************************************************
13152 IF(ISEVAL(IY,IX).EQ.9)  GCE(IY,IX)=GEML(IY+1,IX)
      IF(ISEVAL(IY,IX).EQ.10) GCE(IY,IX)=GEML(IY-1,IX)
      IF(ISEVAL(IY,IX).EQ.11) GCE(IY,IX)=GEML(IY,IX+1)
      IF(ISEVAL(IY,IX).EQ.12) GCE(IY,IX)=GEML(IY,IX-1)
13153 CONTINUE
      CALL SETYX(VAL,GCE,NYDIM,NXDIM)
      DO 13154 IX=1,NX
      DO 13154 IY=1,NY
```

```
        ISEVAL(IY,IX)=0
13154 CONTINUE
C      CALL PRN('VASE',VAL)
13156 CONTINUE
C
C
C   *******************
C   U-Momentum Variable
C   *******************
      IF(NPATCH.NE.'SUCCAE'.OR.INDVAR.NE.3) GO TO 13164
      CALL ONLYIF(3,3,'SUCCAE')
C
C      DO 13158 IX=1,NX-1
C      DO 13158 IY=1,NY
C      WRITE(6,*) 'ISWEEP= ',ISWEEP
C      WRITE(6,*) 'INDVAR= ',INDVAR
C      WRITE(6,*) 'IZ= ',IZ
C      WRITE(6,*) 'IX= ',IX
C      WRITE(6,*) 'IY= ',IY
C      WRITE(6,*) 'IUEVAL= ',IUEVAL(IY,IX)
C 13158 CONTINUE
C
      CALL GETYX(INDVAR,GEMP,NYDIM,NXDIM)
      CALL GETYX(HIGH(INDVAR),GEMH,NYDIM,NXDIM)
      CALL GETYX(LOW(INDVAR),GEML,NYDIM,NXDIM)
C
      DO 13161 IX=1,NX-1
      DO 13161 IY=1,NY
      IF(IUEVAL(IY,IX).EQ.0) GO TO 13161
      IF(IUEVAL(IY,IX).GT.4) GO TO 13159
C ********************************************************
      IF(IUEVAL(IY,IX).EQ.1) GCE(IY,IX)=GEMP(IY+1,IX-1)
      IF(IUEVAL(IY,IX).EQ.2) GCE(IY,IX)=GEMP(IY+1,IX+1)
      IF(IUEVAL(IY,IX).EQ.3) GCE(IY,IX)=GEMP(IY-1,IX-1)
      IF(IUEVAL(IY,IX).EQ.4) GCE(IY,IX)=GEMP(IY-1,IX+1)
      GO TO 13161
C ********************************************************
13159 IF(IUEVAL(IY,IX).GT.8) GO TO 13160
      IF(IUEVAL(IY,IX).EQ.5) GCE(IY,IX)=GEMH(IY+1,IX)
      IF(IUEVAL(IY,IX).EQ.6) GCE(IY,IX)=GEMH(IY-1,IX)
      IF(IUEVAL(IY,IX).EQ.7) GCE(IY,IX)=GEMH(IY,IX+1)
      IF(IUEVAL(IY,IX).EQ.8) GCE(IY,IX)=GEMH(IY,IX-1)
      GO TO 13161
C ********************************************************
13160 IF(IUEVAL(IY,IX).EQ.9)  GCE(IY,IX)=GEML(IY+1,IX)
      IF(IUEVAL(IY,IX).EQ.10) GCE(IY,IX)=GEML(IY-1,IX)
      IF(IUEVAL(IY,IX).EQ.11) GCE(IY,IX)=GEML(IY,IX+1)
      IF(IUEVAL(IY,IX).EQ.12) GCE(IY,IX)=GEML(IY,IX-1)
13161 CONTINUE
      CALL SETYX(VAL,GCE,NYDIM,NXDIM)
      DO 13163 IX=1,NX-1
      DO 13163 IY=1,NY
        IUEVAL(IY,IX)=0
13163 CONTINUE
C      CALL PRN('VAUE',VAL)
13164 CONTINUE
C
C
C   *******************
C   V-Momentum Variable
C   *******************
C
      IF(NPATCH.NE.'SUCCAE'.OR.INDVAR.NE.5) GO TO 13174
      CALL ONLYIF(5,5,'SUCCAE')
C
C      DO 13166 IX=1,NX
```

```
C        DO 13166 IY=1,NY-1
C        WRITE(6,*) 'ISWEEP= ',ISWEEP
C        WRITE(6,*) 'INDVAR= ',INDVAR
C        WRITE(6,*) 'IZ= ',IZ
C        WRITE(6,*) 'IX= ',IX
C        WRITE(6,*) 'IY= ',IY
C        WRITE(6,*) 'IVEVAL= ',IVEVAL(IY,IX)
C 13166 CONTINUE
C
      CALL GETYX(INDVAR,GEMP,NYDIM,NXDIM)
      CALL GETYX(HIGH(INDVAR),GEMH,NYDIM,NXDIM)
      CALL GETYX(LOW(INDVAR),GEML,NYDIM,NXDIM)
C
      DO 13171 IX=1,NX
      DO 13171 IY=1,NY-1
      IF(IVEVAL(IY,IX).EQ.0) GO TO 13171
      IF(IVEVAL(IY,IX).GT.4) GO TO 13169
C ************************************************************
      IF(IVEVAL(IY,IX).EQ.1) GCE(IY,IX)=GEMP(IY+1,IX-1)
      IF(IVEVAL(IY,IX).EQ.2) GCE(IY,IX)=GEMP(IY+1,IX+1)
      IF(IVEVAL(IY,IX).EQ.3) GCE(IY,IX)=GEMP(IY-1,IX-1)
      IF(IVEVAL(IY,IX).EQ.4) GCE(IY,IX)=GEMP(IY-1,IX+1)
      GO TO 13171
C ************************************************************
13169 IF(IVEVAL(IY,IX).GT.8) GO TO 13170
      IF(IVEVAL(IY,IX).EQ.5) GCE(IY,IX)=GEMH(IY+1,IX)
      IF(IVEVAL(IY,IX).EQ.6) GCE(IY,IX)=GEMH(IY-1,IX)
      IF(IVEVAL(IY,IX).EQ.7) GCE(IY,IX)=GEMH(IY,IX+1)
      IF(IVEVAL(IY,IX).EQ.8) GCE(IY,IX)=GEMH(IY,IX-1)
      GO TO 13171
C ************************************************************
13170 IF(IVEVAL(IY,IX).EQ.9)  GCE(IY,IX)=GEML(IY+1,IX)
      IF(IVEVAL(IY,IX).EQ.10) GCE(IY,IX)=GEML(IY-1,IX)
      IF(IVEVAL(IY,IX).EQ.11) GCE(IY,IX)=GEML(IY,IX+1)
      IF(IVEVAL(IY,IX).EQ.12) GCE(IY,IX)=GEML(IY,IX-1)
13171 CONTINUE
      CALL SETYX(VAL,GCE,NYDIM,NXDIM)
      DO 13173 IX=1,NX
      DO 13173 IY=1,NY-1
        IVEVAL(IY,IX)=0
13173 CONTINUE
C       CALL PRN('VAVE',VAL)
13174 CONTINUE
C
C
C   ********************
C   W-Momentum Variable
C   ********************
      IF(NPATCH.NE.'SUCCAE'.OR.INDVAR.NE.7) GO TO 13184
      CALL ONLYIF(7,7,'SUCCAE')
C
C       DO 13176 IX=1,NX
C       DO 13176 IY=1,NY
C       WRITE(6,*) 'ISWEEP= ',ISWEEP
C       WRITE(6,*) 'INDVAR= ',INDVAR
C       WRITE(6,*) 'IZ= ',IZ
C       WRITE(6,*) 'IX= ',IX
C       WRITE(6,*) 'IY= ',IY
C       WRITE(6,*) 'IWEVAL= ',IWEVAL(IY,IX)
C 13176 CONTINUE
C
      CALL GETYX(INDVAR,GEMP,NYDIM,NXDIM)
      CALL GETYX(HIGH(INDVAR),GEMH,NYDIM,NXDIM)
      CALL GETYX(LOW(INDVAR),GEML,NYDIM,NXDIM)
C
      DO 13181 IX=1,NX
```

```
      DO 13181 IY=1,NY
      IF(IWEVAL(IY,IX).EQ.0) GO TO 13181
      IF(IWEVAL(IY,IX).GT.4) GO TO 13179
C ************************************************************
      IF(IWEVAL(IY,IX).EQ.1) GCE(IY,IX)=GEMP(IY+1,IX-1)
      IF(IWEVAL(IY,IX).EQ.2) GCE(IY,IX)=GEMP(IY+1,IX+1)
      IF(IWEVAL(IY,IX).EQ.3) GCE(IY,IX)=GEMP(IY-1,IX-1)
      IF(IWEVAL(IY,IX).EQ.4) GCE(IY,IX)=GEMP(IY-1,IX+1)
      GO TO 13181
C ************************************************************
13179 IF(IWEVAL(IY,IX).GT.8) GO TO 13180
      IF(IWEVAL(IY,IX).EQ.5) GCE(IY,IX)=GEMH(IY+1,IX)
      IF(IWEVAL(IY,IX).EQ.6) GCE(IY,IX)=GEMH(IY-1,IX)
      IF(IWEVAL(IY,IX).EQ.7) GCE(IY,IX)=GEMH(IY,IX+1)
      IF(IWEVAL(IY,IX).EQ.8) GCE(IY,IX)=GEMH(IY,IX-1)
      GO TO 13181
C ************************************************************
13180 IF(IWEVAL(IY,IX).EQ.9)  GCE(IY,IX)=GEML(IY+1,IX)
      IF(IWEVAL(IY,IX).EQ.10) GCE(IY,IX)=GEML(IY-1,IX)
      IF(IWEVAL(IY,IX).EQ.11) GCE(IY,IX)=GEML(IY,IX+1)
      IF(IWEVAL(IY,IX).EQ.12) GCE(IY,IX)=GEML(IY,IX-1)
13181 CONTINUE
      CALL SETYX(VAL,GCE,NYDIM,NXDIM)
      DO 13183 IX=1,NX
      DO 13183 IY=1,NY
        IWEVAL(IY,IX)=0
13183 CONTINUE
C       CALL PRN('VAWE',VAL)
13184 CONTINUE
      RETURN
C------------------- SECTION 14 ------------------- value = GRND2
 1313 CONTINUE
      RETURN
C------------------- SECTION 15 ------------------- value = GRND3
 1314 CONTINUE
      RETURN
C------------------- SECTION 16 ------------------- value = GRND4
 1315 CONTINUE
      RETURN
C------------------- SECTION 17 ------------------- value = GRND5
 1316 CONTINUE
      RETURN
C------------------- SECTION 18 ------------------- value = GRND6
 1317 CONTINUE
      RETURN
C------------------- SECTION 19 ------------------- value = GRND7
 1318 CONTINUE
      RETURN
C------------------- SECTION 20 ------------------- value = GRND8
 1319 CONTINUE
      RETURN
C------------------- SECTION 21 ------------------- value = GRND9
 1320 CONTINUE
      RETURN
C------------------- SECTION 22 ------------------- value = GRND10
 1321 CONTINUE
      RETURN
C******************************************************************
C
C--- GROUP 14. Downstream pressure for PARAB=.TRUE.
C
   14 CONTINUE
      RETURN
C******************************************************************
C* Make changes for these groups only in GROUP 19.
C--- GROUP 15. Termination of sweeps
```

```
C--- GROUP 16. Termination of iterations
C--- GROUP 17. Under-relaxation devices
C--- GROUP 18. Limits on variables or increments to them
C****************************************************************
C
C--- GROUP 19. Special calls to GROUND from EARTH
C
   19 GO TO (191,192,193,194,195,196,197,198),ISC
C   * ------------------- SECTION 1 ---- Start of time step.
  191 CONTINUE
      RETURN
C   * ------------------- SECTION 2 ---- Start of sweep.
  192 CONTINUE
      RETURN
C   * ------------------- SECTION 3 ---- Start of iz slab.
  193 CONTINUE
      RETURN
C   * ------------------- SECTION 4 ---- Start of iteration.
  194 CONTINUE
      RETURN
C   * ------------------- SECTION 5 ---- Finish of iteration.
  195 CONTINUE
      RETURN
C   * ------------------- SECTION 6 ---- Finish of iz slab.
  196 CONTINUE
      RETURN
C   * ------------------- SECTION 7 ---- Finish of sweep.
  197 CONTINUE
      RETURN
C   * ------------------- SECTION 8 ---- Finish of time step.
  198 CONTINUE
      RETURN
C****************************************************************
C
C--- GROUP 20. Preliminary print-out
C
   20 CONTINUE
      RETURN
C****************************************************************
C* Make changes for these groups only in GROUP 19.
C--- GROUP 21. Print-out of variables
C--- GROUP 22. Spot-value print-out
C****************************************************************
C
C--- GROUP 23. Field print-out and plot control
   23 CONTINUE
      RETURN
C****************************************************************
C
C--- GROUP 24. Dumps for restarts
C
   24 CONTINUE
      END
```

```
C***************************************************************
      SUBROUTINE GROUND1
      INCLUDE 'lp16/d_earth/SATEAR'
      INCLUDE 'lp16/d_earth/GRDLOC'
      INCLUDE 'lp16/d_earth/GRDEAR'
      INCLUDE 'lp16/d_earth/GRDBFC'
CXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX USER SECTION STARTS:
C
C 1   Set dimensions of data-for-GROUND arrays here. WARNING: the
C     corresponding arrays in the MAIN program of the satellite
C     and EARTH must have the same dimensions.
      PARAMETER (NLG=20, NIG=20, NRG=100, NCG=10)
C
      COMMON/LGRND/LG(NLG)/IGRND/IG(NIG)/RGRND/RG(NRG)/CGRND/CG(NCG)
      LOGICAL LG
      CHARACTER*4 CG
      COMMON/GUGRND/GCSN(60,80),GCSS(60,80),GCSE(60,80),GCSW(60,80),
     7              GCSL(60,80),GCSH(60,80),GSPCO(60,80),GSECO(60,80),
     8              GCUN(60,80),GCUS(60,80),GCUE(60,80),GCUW(60,80),
     7              GCUL(60,80),GCUH(60,80),GUPCO(60,80),GUECO(60,80),
     8              GCVN(60,80),GCVS(60,80),GCVE(60,80),GCVW(60,80),
     7              GCVL(60,80),GCVH(60,80),GVPCO(60,80),GVECO(60,80),
     8              GCWN(60,80),GCWS(60,80),GCWE(60,80),GCWW(60,80),
     7              GCWL(60,80),GCWH(60,80),GWPCO(60,80),GWECO(60,80)
      COMMON/IUGRND/ISPVAL(60,80),ISEVAL(60,80),IUPVAL(60,80),
     7              IUEVAL(60,80),IVEVAL(60,80),IVPVAL(60,80),
     8              IWEVAL(60,80),IWPVAL(60,80)
C
C 2   User dimensions own arrays here, for example:
C     DIMENSION GUH(10,10),GUC(10,10),GUX(10,10),GUZ(10)
      DIMENSION GCC(60,80),GCMH(60,80),GCML(60,80),
     1          GCE(60,80),GEMP(60,80),GEMH(60,80),GEML(60,80)
C
C 3   User places his data statements here, for example:
      DATA NXDIM,NYDIM/80,60/
C
C 4   Insert own coding below as desired, guided by GREX examples.
C     Note that the satellite-to-GREX special data in the labelled
C     COMMONs /RSG/, /ISG/, /LSG/ and /CSG/ can be included and
C     used below but the user must check GREX for any conflicting
C     uses. The same comment applies to the EARTH-spare working
C     arrays EASP1, EASP2,....EASP20. In addition to the EASPs,
C     there are 10 GRound-earth SPare arrays, GRSP1,...,GRSP10,
C     supplied solely for the user, which are not used by GREX. If
C     the call to GREX has been deactivated then all of the arrays
C     may be used without reservation.
C
      IF(.NOT.(LG(1))) RETURN
      IF(INDVAR.EQ.3) GO TO 88949
      RETURN
88949 CONTINUE
C
C  Corrected Coefficients For U-Equation
C  **************************************
C
C  North-South
C
      IF(NDIREC.NE.1) GO TO 89020
C  i.e. consider flow through North and South faces.
         CALL GETYX(LD11,GCUN,NYDIM,NXDIM)
         CALL GETYX(LD12,GCUS,NYDIM,NXDIM)
         IF(ISWEEP.EQ.FSWEEP) GO TO 89020
C
C
C  North Coefficient
      DO 88980 IX=1,NX-1
```

```
      DO 88980 IY=1,NY
       IF(GCUN(IY,IX).GT.0.0.AND.GCUS(IY,IX).EQ.0.0) GO TO 88955
C......i.e.consider scheme ONLY with a max. of 3 +'ve fluxes...else..
88950    GCC(IY,IX)=GCUN(IY,IX)
         GO TO 88980
88955  IF(GCUE(IY,IX).GT.GCUW(IY,IX)) GO TO 88967
        IF(GCUH(IY,IX).GT.GCUL(IY,IX)) GO TO 88961
C......i.e.m(N),m(W)and m(L) involved...
         IF(GCUL(IY,IX).GE.GCUN(IY,IX).AND.GCUL(IY,IX).GE.GCUW(IY,IX))
     1   GO TO 88959
C......i.e.m(L) dominant...
         IF(GCUW(IY,IX).GE.GCUN(IY,IX).AND.GCUW(IY,IX).GE.GCUL(IY,IX))
     2   GO TO 88957
C......i.e.m(W) dominant...otherwise m(N) is dominant...
          IF(GCUW(IY,IX).GE.GCUL(IY,IX).AND.GCUL(IY,IX).GE.0.0)
     3    GO TO 88956
C......i.e.flow specification is m(N)>m(W)>m(L)>=0.0...otherwise..
C..........flow specification is m(N)>m(L)>m(W)>=0.0...
          GCC(IY,IX)=(1.0-(GCUL(IY,IX)/GCUN(IY,IX)))*(GCUL(IY,IX)+
     4              GCUW(IY,IX)+GCUN(IY,IX))
          GO TO 88980
88956     GCC(IY,IX)=(1.0-(GCUW(IY,IX)/GCUN(IY,IX)))*(GCUL(IY,IX)+
     5              GCUW(IY,IX)+GCUN(IY,IX))
          GO TO 88980
88957     IF(GCUN(IY,IX).GE.GCUL(IY,IX).AND.GCUL(IY,IX).GE.0.0)
     6    GO TO 88958
          GCC(IY,IX)=0.
          GUPCO(IY,IX)=(GCUN(IY,IX)/GCUW(IY,IX))*(GCUL(IY,IX)+
     7                GCUW(IY,IX)+GCUN(IY,IX))
          IUPVAL(IY,IX)=5
C  i.e. Phi = @NWL - corner coeff.
          GO TO 88980
88958     GCC(IY,IX)=0.
          GUECO(IY,IX)=((GCUN(IY,IX)-GCUL(IY,IX))/GCUW(IY,IX))*
     8                (GCUL(IY,IX)+GCUW(IY,IX)+GCUN(IY,IX))
          IUEVAL(IY,IX)=1
C  i.e. Phi = @NW - edge coeff.
          GO TO 88980
88959     IF(GCUW(IY,IX).GE.GCUN(IY,IX).AND.GCUN(IY,IX).GE.0.0)
     9    GO TO 88960
          GCC(IY,IX)=0.
          GUECO(IY,IX)=((GCUN(IY,IX)-GCUW(IY,IX))/GCUL(IY,IX))*
     1                 (GCUL(IY,IX)+GCUW(IY,IX)+GCUN(IY,IX))
          IUEVAL(IY,IX)=9
C  i.e. Phi = @NL - edge coeff
          GO TO 88980
88960     GCC(IY,IX)=0.
          GUPCO(IY,IX)=(GCUN(IY,IX)/GCUL(IY,IX))*(GCUL(IY,IX)+
     2                GCUW(IY,IX)+GCUN(IY,IX))
          IUPVAL(IY,IX)=5
C  i.e. Phi = @NWL - corner coeff.
          GO TO 88980
C **********************************************************************
88961    IF(GCUH(IY,IX).GE.GCUN(IY,IX).AND.GCUH(IY,IX).GE.GCUW(IY,IX))
     1   GO TO 88965
         IF(GCUW(IY,IX).GE.GCUN(IY,IX).AND.GCUW(IY,IX).GE.GCUH(IY,IX))
     2   GO TO 88963
          IF(GCUW(IY,IX).GE.GCUH(IY,IX).AND.GCUH(IY,IX).GE.0.0)
     3    GO TO 88962
          GCC(IY,IX)=(1.0-(GCUH(IY,IX)/GCUN(IY,IX)))*(GCUH(IY,IX)+
     4              GCUW(IY,IX)+GCUN(IY,IX))
          GO TO 88980
88962     GCC(IY,IX)=(1.0-(GCUW(IY,IX)/GCUN(IY,IX)))*(GCUH(IY,IX)+
     5              GCUW(IY,IX)+GCUN(IY,IX))
          GO TO 88980
88963     IF(GCUN(IY,IX).GE.GCUH(IY,IX).AND.GCUH(IY,IX).GE.0.0)
```

```
     6    GO TO 88964
          GCC(IY,IX)=0.
          GUPCO(IY,IX)=(GCUN(IY,IX)/GCUW(IY,IX))*(GCUH(IY,IX)+
     7                GCUW(IY,IX)+GCUN(IY,IX))
          IUPVAL(IY,IX)=1
C  i.e. Phi = @NWH - corner coeff.
          GO TO 88980
88964     GCC(IY,IX)=0.
          GUECO(IY,IX)=((GCUN(IY,IX)-GCUH(IY,IX))/GCUW(IY,IX))*
     8                 (GCUH(IY,IX)+GCUW(IY,IX)+GCUN(IY,IX))
          IUEVAL(IY,IX)=1
C  i.e. Phi = @NW - edge coeff.
          GO TO 88980
88965     IF(GCUW(IY,IX).GE.GCUN(IY,IX).AND.GCUN(IY,IX).GE.0.0)
     9    GO TO 88966
          GCC(IY,IX)=0.
          GUECO(IY,IX)=((GCUN(IY,IX)-GCUW(IY,IX))/GCUH(IY,IX))*
     1                 (GCUH(IY,IX)+GCUW(IY,IX)+GCUN(IY,IX))
          IUEVAL(IY,IX)=5
C  i.e. Phi = @NH - edge coeff
          GO TO 88980
88966     GCC(IY,IX)=0.
          GUPCO(IY,IX)=(GCUN(IY,IX)/GCUH(IY,IX))*(GCUH(IY,IX)+
     2                GCUW(IY,IX)+GCUN(IY,IX))
          IUPVAL(IY,IX)=1
C  i.e. Phi = @NWH - corner coeff.
          GO TO 88980
C ***********************************************************************
88967   IF(GCUH(IY,IX).GT.GCUL(IY,IX)) GO TO 88973
         IF(GCUL(IY,IX).GE.GCUN(IY,IX).AND.GCUL(IY,IX).GE.GCUE(IY,IX))
     1   GO TO 88971
         IF(GCUE(IY,IX).GE.GCUN(IY,IX).AND.GCUE(IY,IX).GE.GCUL(IY,IX))
     2   GO TO 88969
          IF(GCUE(IY,IX).GE.GCUL(IY,IX).AND.GCUL(IY,IX).GE.0.0)
     3    GO TO 88968
          GCC(IY,IX)=(1.0-(GCUL(IY,IX)/GCUN(IY,IX)))*(GCUL(IY,IX)+
     4              GCUE(IY,IX)+GCUN(IY,IX))
          GO TO 88980
88968     GCC(IY,IX)=(1.0-(GCUE(IY,IX)/GCUN(IY,IX)))*(GCUL(IY,IX)+
     5              GCUE(IY,IX)+GCUN(IY,IX))
          GO TO 88980
88969     IF(GCUN(IY,IX).GE.GCUL(IY,IX).AND.GCUL(IY,IX).GE.0.0)
     6    GO TO 88970
          GCC(IY,IX)=0.
          GUPCO(IY,IX)=(GCUN(IY,IX)/GCUE(IY,IX))*(GCUL(IY,IX)+
     7                GCUE(IY,IX)+GCUN(IY,IX))
          IUPVAL(IY,IX)=6
C  i.e. Phi = @NEL - corner coeff.
          GO TO 88980
88970     GCC(IY,IX)=0.
          GUECO(IY,IX)=((GCUN(IY,IX)-GCUL(IY,IX))/GCUE(IY,IX))*
     8                 (GCUL(IY,IX)+GCUE(IY,IX)+GCUN(IY,IX))
          IUEVAL(IY,IX)=2
C  i.e. Phi = @NE - edge coeff.
          GO TO 88980
88971     IF(GCUE(IY,IX).GE.GCUN(IY,IX).AND.GCUN(IY,IX).GE.0.0)
     9    GO TO 88972
          GCC(IY,IX)=0.
          GUECO(IY,IX)=((GCUN(IY,IX)-GCUE(IY,IX))/GCUL(IY,IX))*
     1                 (GCUL(IY,IX)+GCUE(IY,IX)+GCUN(IY,IX))
          IUEVAL(IY,IX)=9
C  i.e. Phi = @NL - edge coeff
          GO TO 88980
88972     GCC(IY,IX)=0.
          GUPCO(IY,IX)=(GCUN(IY,IX)/GCUL(IY,IX))*(GCUL(IY,IX)+
     2                GCUE(IY,IX)+GCUN(IY,IX))
```

```
          IUPVAL(IY,IX)=6
C  i.e. Phi = @NEL - corner coeff.
          GO TO 88980
C  ***********************************************************************
88973    IF(GCUH(IY,IX).GE.GCUN(IY,IX).AND.GCUH(IY,IX).GE.GCUE(IY,IX))
     1   GO TO 88977
         IF(GCUE(IY,IX).GE.GCUN(IY,IX).AND.GCUE(IY,IX).GE.GCUH(IY,IX))
     2   GO TO 88975
          IF(GCUE(IY,IX).GE.GCUH(IY,IX).AND.GCUH(IY,IX).GE.0.0)
     3    GO TO 88974
          GCC(IY,IX)=(1.0-(GCUH(IY,IX)/GCUN(IY,IX)))*(GCUH(IY,IX)+
     4              GCUE(IY,IX)+GCUN(IY,IX))
          GO TO 88980
88974     GCC(IY,IX)=(1.0-(GCUE(IY,IX)/GCUN(IY,IX)))*(GCUH(IY,IX)+
     5              GCUE(IY,IX)+GCUN(IY,IX))
          GO TO 88980
88975     IF(GCUN(IY,IX).GE.GCUH(IY,IX).AND.GCUH(IY,IX).GE.0.0)
     6    GO TO 88976
          GCC(IY,IX)=0.
          GUPCO(IY,IX)=(GCUN(IY,IX)/GCUE(IY,IX))*(GCUH(IY,IX)+
     7                GCUE(IY,IX)+GCUN(IY,IX))
          IUPVAL(IY,IX)=2
C  i.e. Phi = @NEH - corner coeff.
          GO TO 88980
88976     GCC(IY,IX)=0.
          GUECO(IY,IX)=((GCUN(IY,IX)-GCUH(IY,IX))/GCUE(IY,IX))*
     8                 (GCUH(IY,IX)+GCUE(IY,IX)+GCUN(IY,IX))
          IUEVAL(IY,IX)=2
C  i.e. Phi = @NE - edge coeff.
          GO TO 88980
88977     IF(GCUE(IY,IX).GE.GCUN(IY,IX).AND.GCUN(IY,IX).GE.0.0)
     9    GO TO 88978
          GCC(IY,IX)=0.
          GUECO(IY,IX)=((GCUN(IY,IX)-GCUE(IY,IX))/GCUH(IY,IX))*
     1                 (GCUH(IY,IX)+GCUE(IY,IX)+GCUN(IY,IX))
          IUEVAL(IY,IX)=5
C  i.e. Phi = @NH - edge coeff
          GO TO 88980
88978     GCC(IY,IX)=0.
          GUPCO(IY,IX)=(GCUN(IY,IX)/GCUH(IY,IX))*(GCUH(IY,IX)+
     2                GCUE(IY,IX)+GCUN(IY,IX))
          IUPVAL(IY,IX)=2
C  i.e. Phi = @NEH - corner coeff.
88980 CONTINUE
      CALL SETYX(LD11,GCC,NYDIM,NXDIM)
C      CALL PRN('CCSN',LD11)
C
C   South Coefficient
C
      DO 89010 IX=1,NX-1
      DO 89010 IY=1,NY
       IF(GCUS(IY,IX).GT.0.0.AND.GCUN(IY,IX).EQ.0.0) GO TO 88985
88982    GCC(IY,IX)=GCUS(IY,IX)
         GO TO 89010
88985  IF(GCUE(IY,IX).GT.GCUW(IY,IX)) GO TO 88997
        IF(GCUH(IY,IX).GT.GCUL(IY,IX)) GO TO 88991
         IF(GCUL(IY,IX).GE.GCUS(IY,IX).AND.GCUL(IY,IX).GE.GCUW(IY,IX))
     1   GO TO 88989
         IF(GCUW(IY,IX).GE.GCUS(IY,IX).AND.GCUW(IY,IX).GE.GCUL(IY,IX))
     2   GO TO 88987
          IF(GCUW(IY,IX).GE.GCUL(IY,IX).AND.GCUL(IY,IX).GE.0.0)
     3    GO TO 88986
          GCC(IY,IX)=(1.0-(GCUL(IY,IX)/GCUS(IY,IX)))*(GCUL(IY,IX)+
     4              GCUW(IY,IX)+GCUS(IY,IX))
          GO TO 89010
88986     GCC(IY,IX)=(1.0-(GCUW(IY,IX)/GCUS(IY,IX)))*(GCUL(IY,IX)+
```

```
     5              GCUW(IY,IX)+GCUS(IY,IX))
      GO TO 89010
88987 IF(GCUS(IY,IX).GE.GCUL(IY,IX).AND.GCUL(IY,IX).GE.0.0)
     6 GO TO 88988
      GCC(IY,IX)=0.
      GUPCO(IY,IX)=(GCUS(IY,IX)/GCUW(IY,IX))*(GCUL(IY,IX)+
     7              GCUW(IY,IX)+GCUS(IY,IX))
      IUPVAL(IY,IX)=7
C  i.e. Phi = @SWL - corner coeff.
      GO TO 89010
88988 GCC(IY,IX)=0.
      GUECO(IY,IX)=((GCUS(IY,IX)-GCUL(IY,IX))/GCUW(IY,IX))*
     8               (GCUL(IY,IX)+GCUW(IY,IX)+GCUS(IY,IX))
      IUEVAL(IY,IX)=3
C  i.e. Phi = @SW - edge coeff.
      GO TO 89010
88989 IF(GCUW(IY,IX).GE.GCUS(IY,IX).AND.GCUS(IY,IX).GE.0.0)
     9 GO TO 88990
      GCC(IY,IX)=0.
      GUECO(IY,IX)=((GCUS(IY,IX)-GCUW(IY,IX))/GCUL(IY,IX))*
     1               (GCUL(IY,IX)+GCUW(IY,IX)+GCUS(IY,IX))
      IUEVAL(IY,IX)=10
C  i.e. Phi = @SL - edge coeff
      GO TO 89010
88990 GCC(IY,IX)=0.
      GUPCO(IY,IX)=(GCUS(IY,IX)/GCUL(IY,IX))*(GCUL(IY,IX)+
     2              GCUW(IY,IX)+GCUS(IY,IX))
      IUPVAL(IY,IX)=7
C  i.e. Phi = @SWL - corner coeff.
      GO TO 89010
C ***********************************************************************
88991 IF(GCUH(IY,IX).GE.GCUS(IY,IX).AND.GCUH(IY,IX).GE.GCUW(IY,IX))
     1 GO TO 88995
      IF(GCUW(IY,IX).GE.GCUS(IY,IX).AND.GCUW(IY,IX).GE.GCUH(IY,IX))
     2 GO TO 88993
      IF(GCUW(IY,IX).GE.GCUH(IY,IX).AND.GCUH(IY,IX).GE.0.0)
     3 GO TO 88992
      GCC(IY,IX)=(1.0-(GCUH(IY,IX)/GCUS(IY,IX)))*(GCUH(IY,IX)+
     4            GCUW(IY,IX)+GCUS(IY,IX))
      GO TO 89010
88992 GCC(IY,IX)=(1.0-(GCUW(IY,IX)/GCUS(IY,IX)))*(GCUH(IY,IX)+
     5            GCUW(IY,IX)+GCUS(IY,IX))
      GO TO 89010
88993 IF(GCUS(IY,IX).GE.GCUH(IY,IX).AND.GCUH(IY,IX).GE.0.0)
     6 GO TO 88994
      GCC(IY,IX)=0.
      GUPCO(IY,IX)=(GCUS(IY,IX)/GCUW(IY,IX))*(GCUH(IY,IX)+
     7              GCUW(IY,IX)+GCUS(IY,IX))
      IUPVAL(IY,IX)=3
C  i.e. Phi = @SWH - corner coeff.
      GO TO 89010
88994 GCC(IY,IX)=0.
      GUECO(IY,IX)=((GCUS(IY,IX)-GCUH(IY,IX))/GCUW(IY,IX))*
     8               (GCUH(IY,IX)+GCUW(IY,IX)+GCUS(IY,IX))
      IUEVAL(IY,IX)=3
C  i.e. Phi = @SW - edge coeff.
      GO TO 89010
88995 IF(GCUW(IY,IX).GE.GCUS(IY,IX).AND.GCUS(IY,IX).GE.0.0)
     9 GO TO 88996
      GCC(IY,IX)=0.
      GUECO(IY,IX)=((GCUS(IY,IX)-GCUW(IY,IX))/GCUH(IY,IX))*
     1               (GCUH(IY,IX)+GCUW(IY,IX)+GCUS(IY,IX))
      IUEVAL(IY,IX)=6
C  i.e. Phi = @SH - edge coeff
      GO TO 89010
88996 GCC(IY,IX)=0.
```

```
           GUPCO(IY,IX)=(GCUS(IY,IX)/GCUH(IY,IX))*(GCUH(IY,IX)+
     2                GCUW(IY,IX)+GCUS(IY,IX))
           IUPVAL(IY,IX)=3
C  i.e. Phi = @SWH - corner coeff.
           GO TO 89010
C ***********************************************************************
88997   IF(GCUH(IY,IX).GT.GCUL(IY,IX)) GO TO 89003
          IF(GCUL(IY,IX).GE.GCUS(IY,IX).AND.GCUL(IY,IX).GE.GCUE(IY,IX))
     1    GO TO 89001
          IF(GCUE(IY,IX).GE.GCUS(IY,IX).AND.GCUE(IY,IX).GE.GCUL(IY,IX))
     2    GO TO 88999
           IF(GCUE(IY,IX).GE.GCUL(IY,IX).AND.GCUL(IY,IX).GE.0.0)
     3     GO TO 88998
           GCC(IY,IX)=(1.0-(GCUL(IY,IX)/GCUS(IY,IX)))*(GCUL(IY,IX)+
     4              GCUE(IY,IX)+GCUS(IY,IX))
           GO TO 89010
88998      GCC(IY,IX)=(1.0-(GCUE(IY,IX)/GCUS(IY,IX)))*(GCUL(IY,IX)+
     5              GCUE(IY,IX)+GCUS(IY,IX))
           GO TO 89010
88999      IF(GCUS(IY,IX).GE.GCUL(IY,IX).AND.GCUL(IY,IX).GE.0.0)
     6     GO TO 89000
           GCC(IY,IX)=0.
           GUPCO(IY,IX)=(GCUS(IY,IX)/GCUE(IY,IX))*(GCUL(IY,IX)+
     7                GCUE(IY,IX)+GCUS(IY,IX))
           IUPVAL(IY,IX)=8
C  i.e. Phi = @SEL - corner coeff.
           GO TO 89010
89000      GCC(IY,IX)=0.
           GUECO(IY,IX)=((GCUS(IY,IX)-GCUL(IY,IX))/GCUE(IY,IX))*
     8                 (GCUL(IY,IX)+GCUE(IY,IX)+GCUS(IY,IX))
           IUEVAL(IY,IX)=4
C  i.e. Phi = @SE - edge coeff.
           GO TO 89010
89001      IF(GCUE(IY,IX).GE.GCUS(IY,IX).AND.GCUS(IY,IX).GE.0.0)
     9     GO TO 89002
           GCC(IY,IX)=0.
           GUECO(IY,IX)=((GCUS(IY,IX)-GCUE(IY,IX))/GCUL(IY,IX))*
     1                 (GCUL(IY,IX)+GCUE(IY,IX)+GCUS(IY,IX))
           IUEVAL(IY,IX)=10
C  i.e. Phi = @SL - edge coeff
           GO TO 89010
89002      GCC(IY,IX)=0.
           GUPCO(IY,IX)=(GCUS(IY,IX)/GCUL(IY,IX))*(GCUL(IY,IX)+
     2                GCUE(IY,IX)+GCUS(IY,IX))
           IUPVAL(IY,IX)=8
C  i.e. Phi = @SEL - corner coeff.
           GO TO 89010
C ***********************************************************************
89003   IF(GCUH(IY,IX).GE.GCUS(IY,IX).AND.GCUH(IY,IX).GE.GCUE(IY,IX))
     1  GO TO 89007
        IF(GCUE(IY,IX).GE.GCUS(IY,IX).AND.GCUE(IY,IX).GE.GCUH(IY,IX))
     2  GO TO 89005
         IF(GCUE(IY,IX).GE.GCUH(IY,IX).AND.GCUH(IY,IX).GE.0.0)
     3   GO TO 89004
         GCC(IY,IX)=(1.0-(GCUH(IY,IX)/GCUS(IY,IX)))*(GCUH(IY,IX)+
     4            GCUE(IY,IX)+GCUS(IY,IX))
         GO TO 89010
89004    GCC(IY,IX)=(1.0-(GCUE(IY,IX)/GCUS(IY,IX)))*(GCUH(IY,IX)+
     5            GCUE(IY,IX)+GCUS(IY,IX))
         GO TO 89010
89005    IF(GCUS(IY,IX).GE.GCUH(IY,IX).AND.GCUH(IY,IX).GE.0.0)
     6   GO TO 89006
         GCC(IY,IX)=0.
         GUPCO(IY,IX)=(GCUS(IY,IX)/GCUE(IY,IX))*(GCUH(IY,IX)+
     7              GCUE(IY,IX)+GCUS(IY,IX))
         IUPVAL(IY,IX)=4
```

```
C  i.e. Phi = @SEH - corner coeff.
            GO TO 89010
89006       GCC(IY,IX)=0.
            GUECO(IY,IX)=((GCUS(IY,IX)-GCUH(IY,IX))/GCUE(IY,IX))*
     8                     (GCUH(IY,IX)+GCUE(IY,IX)+GCUS(IY,IX))
            IUEVAL(IY,IX)=4
C  i.e. Phi = @SE - edge coeff.
            GO TO 89010
89007       IF(GCUE(IY,IX).GE.GCUS(IY,IX).AND.GCUS(IY,IX).GE.0.0)
     9      GO TO 89008
            GCC(IY,IX)=0.
            GUECO(IY,IX)=((GCUS(IY,IX)-GCUE(IY,IX))/GCUH(IY,IX))*
     1                     (GCUH(IY,IX)+GCUE(IY,IX)+GCUS(IY,IX))
            IUEVAL(IY,IX)=6
C  i.e. Phi = @SH - edge coeff
            GO TO 89010
89008       GCC(IY,IX)=0.
            GUPCO(IY,IX)=(GCUS(IY,IX)/GCUH(IY,IX))*(GCUH(IY,IX)+
     2                   GCUE(IY,IX)+GCUS(IY,IX))
            IUPVAL(IY,IX)=4
C  i.e. Phi = @SEH - corner coeff.
89010 CONTINUE
      CALL SETYX(LD12,GCC,NYDIM,NXDIM)
C      CALL PRN('CCSS',LD12)
      RETURN
C
C    East-West
C
89020 IF(NDIREC.NE.3) GO TO 89100
C   i.e. consider flow through East and West faces.
          CALL GETYX(LD11,GCUE,NYDIM,NXDIM)
          CALL GETYX(LD12,GCUW,NYDIM,NXDIM)
          IF(ISWEEP.EQ.FSWEEP) GO TO 89100
C
C    East Coefficient
C
      DO 89060 IX=1,NX-1
      DO 89060 IY=1,NY
       IF(GCUE(IY,IX).GT.0.0.AND.GCUW(IY,IX).EQ.0.0) GO TO 89025
89022    GCC(IY,IX)=GCUE(IY,IX)
         GO TO 89060
89025  IF(GCUN(IY,IX).GT.GCUS(IY,IX)) GO TO 89037
        IF(GCUH(IY,IX).GT.GCUL(IY,IX)) GO TO 89031
         IF(GCUL(IY,IX).GE.GCUE(IY,IX).AND.GCUL(IY,IX).GE.GCUS(IY,IX))
     1   GO TO 89029
         IF(GCUS(IY,IX).GE.GCUE(IY,IX).AND.GCUS(IY,IX).GE.GCUL(IY,IX))
     2   GO TO 89027
          IF(GCUS(IY,IX).GE.GCUL(IY,IX).AND.GCUL(IY,IX).GE.0.0)
     3    GO TO 89026
          GCC(IY,IX)=(1.0-(GCUL(IY,IX)/GCUE(IY,IX)))*(GCUL(IY,IX)+
     4                GCUE(IY,IX)+GCUS(IY,IX))
          GO TO 89060
89026     GCC(IY,IX)=(1.0-(GCUS(IY,IX)/GCUE(IY,IX)))*(GCUL(IY,IX)+
     5                GCUE(IY,IX)+GCUS(IY,IX))
          GO TO 89060
89027     IF(GCUE(IY,IX).GE.GCUL(IY,IX).AND.GCUL(IY,IX).GE.0.0)
     6    GO TO 89028
          GCC(IY,IX)=0.
          GUPCO(IY,IX)=(GCUE(IY,IX)/GCUS(IY,IX))*(GCUL(IY,IX)+
     7                   GCUE(IY,IX)+GCUS(IY,IX))
          IUPVAL(IY,IX)=8
C  i.e. Phi = @SEL - corner coeff.
          GO TO 89060
89028     GCC(IY,IX)=0.
          GUECO(IY,IX)=((GCUE(IY,IX)-GCUL(IY,IX))/GCUS(IY,IX))*
     8                   (GCUL(IY,IX)+GCUE(IY,IX)+GCUS(IY,IX))
```

```
         IUEVAL(IY,IX)=4
C  i.e. Phi = @SE - edge coeff.
         GO TO 89060
89029    IF(GCUE(IY,IX).GE.GCUS(IY,IX).AND.GCUS(IY,IX).GE.0.0)
     9   GO TO 89030
         GCC(IY,IX)=0.
         GUPCO(IY,IX)=(GCUE(IY,IX)/GCUL(IY,IX))*(GCUL(IY,IX)+
     2            GCUE(IY,IX)+GCUS(IY,IX))
         IUPVAL(IY,IX)=8
C  i.e. Phi = @SEL - corner coeff.
         GO TO 89060
89030    GCC(IY,IX)=0.
         GUECO(IY,IX)=((GCUE(IY,IX)-GCUS(IY,IX))/GCUL(IY,IX))*
     1             (GCUL(IY,IX)+GCUE(IY,IX)+GCUS(IY,IX))
         IUEVAL(IY,IX)=11
C  i.e. Phi = @EL - edge coeff
         GO TO 89060
C *********************************************************************
89031   IF(GCUH(IY,IX).GE.GCUE(IY,IX).AND.GCUH(IY,IX).GE.GCUS(IY,IX))
     1  GO TO 89035
        IF(GCUS(IY,IX).GE.GCUE(IY,IX).AND.GCUS(IY,IX).GE.GCUH(IY,IX))
     2  GO TO 89033
         IF(GCUS(IY,IX).GE.GCUH(IY,IX).AND.GCUH(IY,IX).GE.0.0)
     3   GO TO 89032
         GCC(IY,IX)=(1.0-(GCUH(IY,IX)/GCUE(IY,IX)))*(GCUH(IY,IX)+
     4          GCUE(IY,IX)+GCUS(IY,IX))
89032    GCC(IY,IX)=(1.0-(GCUS(IY,IX)/GCUE(IY,IX)))*(GCUH(IY,IX)+
     5          GCUE(IY,IX)+GCUS(IY,IX))
         GO TO 89060
89033    IF(GCUE(IY,IX).GE.GCUH(IY,IX).AND.GCUH(IY,IX).GE.0.0)
     6   GO TO 89034
         GCC(IY,IX)=0.
         GUPCO(IY,IX)=(GCUE(IY,IX)/GCUS(IY,IX))*(GCUH(IY,IX)+
     7            GCUE(IY,IX)+GCUS(IY,IX))
         IUPVAL(IY,IX)=4
C  i.e. Phi = @SEH - corner coeff.
         GO TO 89060
89034    GCC(IY,IX)=0.
         GUECO(IY,IX)=((GCUE(IY,IX)-GCUH(IY,IX))/GCUS(IY,IX))*
     8             (GCUH(IY,IX)+GCUE(IY,IX)+GCUS(IY,IX))
         IUEVAL(IY,IX)=4
C  i.e. Phi = @SE - edge coeff.
         GO TO 89060
89035    IF(GCUE(IY,IX).GE.GCUS(IY,IX).AND.GCUS(IY,IX).GE.0.0)
     9   GO TO 89036
         GCC(IY,IX)=0.
         GUPCO(IY,IX)=(GCUE(IY,IX)/GCUH(IY,IX))*(GCUH(IY,IX)+
     2            GCUE(IY,IX)+GCUS(IY,IX))
         IUPVAL(IY,IX)=4
C  i.e. Phi = @SEH - corner coeff.
         GO TO 89060
89036    GCC(IY,IX)=0.
         GUECO(IY,IX)=((GCUE(IY,IX)-GCUS(IY,IX))/GCUH(IY,IX))*
     1             (GCUH(IY,IX)+GCUE(IY,IX)+GCUS(IY,IX))
         IUEVAL(IY,IX)=7
C  i.e. Phi = @EH - edge coeff
         GO TO 89060
C *********************************************************************
89037  IF(GCUH(IY,IX).GT.GCUL(IY,IX)) GO TO 89043
        IF(GCUL(IY,IX).GE.GCUN(IY,IX).AND.GCUL(IY,IX).GE.GCUE(IY,IX))
     1  GO TO 89041
        IF(GCUN(IY,IX).GE.GCUE(IY,IX).AND.GCUN(IY,IX).GE.GCUL(IY,IX))
     2  GO TO 89039
         IF(GCUN(IY,IX).GE.GCUL(IY,IX).AND.GCUL(IY,IX).GE.0.0)
     3   GO TO 89038
         GCC(IY,IX)=(1.0-(GCUL(IY,IX)/GCUE(IY,IX)))*(GCUL(IY,IX)+
```

```
     4               GCUE(IY,IX)+GCUN(IY,IX))
           GO TO 89060
89038      GCC(IY,IX)=(1.0-(GCUN(IY,IX)/GCUE(IY,IX)))*(GCUL(IY,IX)+
     5               GCUE(IY,IX)+GCUN(IY,IX))
           GO TO 89060
89039      IF(GCUE(IY,IX).GE.GCUL(IY,IX).AND.GCUL(IY,IX).GE.0.0)
     6     GO TO 89040
           GCC(IY,IX)=0.
           GUPCO(IY,IX)=(GCUE(IY,IX)/GCUN(IY,IX))*(GCUL(IY,IX)+
     7                 GCUE(IY,IX)+GCUN(IY,IX))
           IUPVAL(IY,IX)=6
C  i.e. Phi = @NEL - corner coeff.
           GO TO 89060
89040      GCC(IY,IX)=0.
           GUECO(IY,IX)=((GCUE(IY,IX)-GCUL(IY,IX))/GCUN(IY,IX))*
     8                  (GCUL(IY,IX)+GCUE(IY,IX)+GCUN(IY,IX))
           IUEVAL(IY,IX)=2
C  i.e. Phi = @NE - edge coeff.
           GO TO 89060
89041      IF(GCUE(IY,IX).GE.GCUN(IY,IX).AND.GCUN(IY,IX).GE.0.0)
     9     GO TO 89042
           GCC(IY,IX)=0.
           GUPCO(IY,IX)=(GCUE(IY,IX)/GCUL(IY,IX))*(GCUL(IY,IX)+
     2                 GCUE(IY,IX)+GCUN(IY,IX))
           IUPVAL(IY,IX)=6
C  i.e. Phi = @NEL - corner coeff.
           GO TO 89060
89042      GCC(IY,IX)=0.
           GUECO(IY,IX)=((GCUE(IY,IX)-GCUN(IY,IX))/GCUL(IY,IX))*
     1                  (GCUL(IY,IX)+GCUE(IY,IX)+GCUN(IY,IX))
           IUEVAL(IY,IX)=11
C  i.e. Phi = @EL - edge coeff
           GO TO 89060
C ****************************************************************************
89043     IF(GCUH(IY,IX).GE.GCUN(IY,IX).AND.GCUH(IY,IX).GE.GCUE(IY,IX))
     1    GO TO 89047
          IF(GCUN(IY,IX).GE.GCUE(IY,IX).AND.GCUN(IY,IX).GE.GCUH(IY,IX))
     2    GO TO 89045
           IF(GCUN(IY,IX).GE.GCUH(IY,IX).AND.GCUH(IY,IX).GE.0.0)
     3     GO TO 89044
           GCC(IY,IX)=(1.0-(GCUH(IY,IX)/GCUE(IY,IX)))*(GCUH(IY,IX)+
     4               GCUE(IY,IX)+GCUN(IY,IX))
           GO TO 89060
89044      GCC(IY,IX)=(1.0-(GCUN(IY,IX)/GCUE(IY,IX)))*(GCUH(IY,IX)+
     5               GCUE(IY,IX)+GCUN(IY,IX))
           GO TO 89060
89045      IF(GCUE(IY,IX).GE.GCUH(IY,IX).AND.GCUH(IY,IX).GE.0.0)
     6     GO TO 89046
           GCC(IY,IX)=0.
           GUPCO(IY,IX)=(GCUE(IY,IX)/GCUN(IY,IX))*(GCUH(IY,IX)+
     7                 GCUE(IY,IX)+GCUN(IY,IX))
           IUPVAL(IY,IX)=2
C  i.e. Phi = @NEH - corner coeff.
           GO TO 89060
89046      GCC(IY,IX)=0.
           GUECO(IY,IX)=((GCUE(IY,IX)-GCUH(IY,IX))/GCUN(IY,IX))*
     8                  (GCUH(IY,IX)+GCUE(IY,IX)+GCUN(IY,IX))
           IUEVAL(IY,IX)=2
C  i.e. Phi = @NE - edge coeff.
           GO TO 89060
89047      IF(GCUE(IY,IX).GE.GCUN(IY,IX).AND.GCUN(IY,IX).GE.0.0)
     9     GO TO 89048
           GUPCO(IY,IX)=(GCUE(IY,IX)/GCUH(IY,IX))*(GCUH(IY,IX)+
     2                 GCUE(IY,IX)+GCUN(IY,IX))
           IUPVAL(IY,IX)=2
C  i.e. Phi = @NEH - corner coeff.
```

```
           GO TO 89060
89048      GCC(IY,IX)=0.
           GUECO(IY,IX)=((GCUE(IY,IX)-GCUN(IY,IX))/GCUH(IY,IX))*
     1                   (GCUH(IY,IX)+GCUE(IY,IX)+GCUN(IY,IX))
           IUEVAL(IY,IX)=7
C  i.e. Phi = @EH - edge coeff.
89060 CONTINUE
       CALL SETYX(LD11,GCC,NYDIM,NXDIM)
C       CALL PRN('CCSE',LD11)
C
C     West Coefficient
C
       DO 89090 IX=1,NX-1
       DO 89090 IY=1,NY
        IF(GCUW(IY,IX).GT.0.0.AND.GCUE(IY,IX).EQ.0.0) GO TO 89065
89062     GCC(IY,IX)=GCUW(IY,IX)
          GO TO 89090
89065   IF(GCUN(IY,IX).GT.GCUS(IY,IX)) GO TO 89077
         IF(GCUH(IY,IX).GT.GCUL(IY,IX)) GO TO 89071
          IF(GCUL(IY,IX).GE.GCUW(IY,IX).AND.GCUL(IY,IX).GE.GCUS(IY,IX))
     1    GO TO 89069
          IF(GCUS(IY,IX).GE.GCUW(IY,IX).AND.GCUS(IY,IX).GE.GCUL(IY,IX))
     2    GO TO 89067
           IF(GCUS(IY,IX).GE.GCUL(IY,IX).AND.GCUL(IY,IX).GE.0.0)
     3     GO TO 89066
           GCC(IY,IX)=(1.0-(GCUL(IY,IX)/GCUW(IY,IX)))*(GCUL(IY,IX)+
     4               GCUW(IY,IX)+GCUS(IY,IX))
           GO TO 89090
89066      GCC(IY,IX)=(1.0-(GCUS(IY,IX)/GCUW(IY,IX)))*(GCUL(IY,IX)+
     5               GCUW(IY,IX)+GCUS(IY,IX))
           GO TO 89090
89067      IF(GCUW(IY,IX).GE.GCUL(IY,IX).AND.GCUL(IY,IX).GE.0.0)
     6     GO TO 89068
           GCC(IY,IX)=0.
           GUPCO(IY,IX)=(GCUW(IY,IX)/GCUS(IY,IX))*(GCUL(IY,IX)+
     7                  GCUW(IY,IX)+GCUS(IY,IX))
           IUPVAL(IY,IX)=7
C  i.e. Phi = @SWL - corner coeff.
           GO TO 89090
89068      GCC(IY,IX)=0.
           GUECO(IY,IX)=((GCUW(IY,IX)-GCUL(IY,IX))/GCUS(IY,IX))*
     8                   (GCUL(IY,IX)+GCUW(IY,IX)+GCUS(IY,IX))
           IUEVAL(IY,IX)=3
C  i.e. Phi = @SW - edge coeff.
           GO TO 89090
89069      IF(GCUW(IY,IX).GE.GCUS(IY,IX).AND.GCUS(IY,IX).GE.0.0)
     9     GO TO 89070
           GCC(IY,IX)=0.
           GUPCO(IY,IX)=(GCUW(IY,IX)/GCUL(IY,IX))*(GCUL(IY,IX)+
     2                  GCUW(IY,IX)+GCUS(IY,IX))
           IUPVAL(IY,IX)=7
C  i.e. Phi = @SWL - corner coeff.
           GO TO 89090
89070      GCC(IY,IX)=0.
           GUECO(IY,IX)=((GCUW(IY,IX)-GCUS(IY,IX))/GCUL(IY,IX))*
     1                   (GCUL(IY,IX)+GCUW(IY,IX)+GCUS(IY,IX))
           IUEVAL(IY,IX)=12
C  i.e. Phi = @WL - edge coeff
           GO TO 89090
C *********************************************************************
89071     IF(GCUH(IY,IX).GE.GCUW(IY,IX).AND.GCUH(IY,IX).GE.GCUS(IY,IX))
     1    GO TO 89075
          IF(GCUS(IY,IX).GE.GCUW(IY,IX).AND.GCUS(IY,IX).GE.GCUH(IY,IX))
     2    GO TO 89073
           IF(GCUS(IY,IX).GE.GCUH(IY,IX).AND.GCUH(IY,IX).GE.0.0)
     3     GO TO 89072
```

```
      GCC(IY,IX)=(1.0-(GCUH(IY,IX)/GCUW(IY,IX)))*(GCUH(IY,IX)+
     4           GCUW(IY,IX)+GCUS(IY,IX))
      GO TO 89090
89072 GCC(IY,IX)=(1.0-(GCUS(IY,IX)/GCUW(IY,IX)))*(GCUH(IY,IX)+
     5           GCUW(IY,IX)+GCUS(IY,IX))
      GO TO 89090
89073 IF(GCUW(IY,IX).GE.GCUH(IY,IX).AND.GCUH(IY,IX).GE.0.0)
     6 GO TO 89074
      GCC(IY,IX)=0.
      GUPCO(IY,IX)=(GCUW(IY,IX)/GCUS(IY,IX))*(GCUH(IY,IX)+
     7             GCUW(IY,IX)+GCUS(IY,IX))
      IUPVAL(IY,IX)=3
C  i.e. Phi = @SWH - corner coeff.
      GO TO 89090
89074 GCC(IY,IX)=0.
      GUECO(IY,IX)=((GCUW(IY,IX)-GCUH(IY,IX))/GCUS(IY,IX))*
     8             (GCUH(IY,IX)+GCUW(IY,IX)+GCUS(IY,IX))
      IUEVAL(IY,IX)=3
C  i.e. Phi = @SW - edge coeff.
      GO TO 89090
89075 IF(GCUW(IY,IX).GE.GCUS(IY,IX).AND.GCUS(IY,IX).GE.0.0)
     9 GO TO 89076
      GCC(IY,IX)=0.
      GUPCO(IY,IX)=(GCUW(IY,IX)/GCUH(IY,IX))*(GCUH(IY,IX)+
     2             GCUW(IY,IX)+GCUS(IY,IX))
      IUPVAL(IY,IX)=3
C  i.e. Phi = @SWH - corner coeff.
      GO TO 89090
89076 GCC(IY,IX)=0.
      GUECO(IY,IX)=((GCUW(IY,IX)-GCUS(IY,IX))/GCUH(IY,IX))*
     1             (GCUH(IY,IX)+GCUW(IY,IX)+GCUS(IY,IX))
      IUEVAL(IY,IX)=8
C  i.e. Phi = @WH - edge coeff
      GO TO 89090
C ***********************************************************************
89077 IF(GCUH(IY,IX).GT.GCUL(IY,IX)) GO TO 89083
       IF(GCUL(IY,IX).GE.GCUN(IY,IX).AND.GCUL(IY,IX).GE.GCUW(IY,IX))
     1 GO TO 89081
       IF(GCUN(IY,IX).GE.GCUW(IY,IX).AND.GCUN(IY,IX).GE.GCUL(IY,IX))
     2 GO TO 89079
      IF(GCUN(IY,IX).GE.GCUL(IY,IX).AND.GCUL(IY,IX).GE.0.0)
     3 GO TO 89078
      GCC(IY,IX)=(1.0-(GCUL(IY,IX)/GCUW(IY,IX)))*(GCUL(IY,IX)+
     4           GCUW(IY,IX)+GCUN(IY,IX))
      GO TO 89090
89078 GCC(IY,IX)=(1.0-(GCUN(IY,IX)/GCUW(IY,IX)))*(GCUL(IY,IX)+
     5           GCUW(IY,IX)+GCUN(IY,IX))
      GO TO 89090
89079 IF(GCUW(IY,IX).GE.GCUL(IY,IX).AND.GCUL(IY,IX).GE.0.0)
     6 GO TO 89080
      GCC(IY,IX)=0.
      GUPCO(IY,IX)=(GCUW(IY,IX)/GCUN(IY,IX))*(GCUL(IY,IX)+
     7             GCUW(IY,IX)+GCUN(IY,IX))
      IUPVAL(IY,IX)=5
C  i.e. Phi = @NWL - corner coeff.
      GO TO 89090
89080 GCC(IY,IX)=0.
      GUECO(IY,IX)=((GCUW(IY,IX)-GCUL(IY,IX))/GCUN(IY,IX))*
     8             (GCUL(IY,IX)+GCUW(IY,IX)+GCUN(IY,IX))
      IUEVAL(IY,IX)=1
C  i.e. Phi = @NW - edge coeff.
      GO TO 89090
89081 IF(GCUW(IY,IX).GE.GCUN(IY,IX).AND.GCUN(IY,IX).GE.0.0)
     9 GO TO 89082
      GCC(IY,IX)=0.
      GUPCO(IY,IX)=(GCUW(IY,IX)/GCUL(IY,IX))*(GCUL(IY,IX)+
```

```
     2                 GCUW(IY,IX)+GCUN(IY,IX))
          IUPVAL(IY,IX)=5
C  i.e. Phi = @NWL - corner coeff.
          GO TO 89090
89082     GCC(IY,IX)=0.
          GUECO(IY,IX)=((GCUW(IY,IX)-GCUN(IY,IX))/GCUL(IY,IX))*
     1                  (GCUL(IY,IX)+GCUW(IY,IX)+GCUN(IY,IX))
          IUEVAL(IY,IX)=12
C  i.e. Phi = @WL - edge coeff
          GO TO 89090
C ***********************************************************************
89083    IF(GCUH(IY,IX).GE.GCUN(IY,IX).AND.GCUH(IY,IX).GE.GCUW(IY,IX))
     1   GO TO 89087
         IF(GCUN(IY,IX).GE.GCUW(IY,IX).AND.GCUN(IY,IX).GE.GCUH(IY,IX))
     2   GO TO 89085
          IF(GCUN(IY,IX).GE.GCUH(IY,IX).AND.GCUH(IY,IX).GE.0.0)
     3    GO TO 89084
          GCC(IY,IX)=(1.0-(GCUH(IY,IX)/GCUW(IY,IX)))*(GCUH(IY,IX)+
     4              GCUW(IY,IX)+GCUN(IY,IX))
          GO TO 89090
89084     GCC(IY,IX)=(1.0-(GCUN(IY,IX)/GCUW(IY,IX)))*(GCUH(IY,IX)+
     5              GCUW(IY,IX)+GCUN(IY,IX))
          GO TO 89090
89085     IF(GCUW(IY,IX).GE.GCUH(IY,IX).AND.GCUH(IY,IX).GE.0.0)
     6    GO TO 89086
          GCC(IY,IX)=0.
          GUPCO(IY,IX)=(GCUW(IY,IX)/GCUN(IY,IX))*(GCUH(IY,IX)+
     7                GCUW(IY,IX)+GCUN(IY,IX))
          IUPVAL(IY,IX)=1
C  i.e. Phi = @NWH - corner coeff.
          GO TO 89090
89086     GCC(IY,IX)=0.
          GUECO(IY,IX)=((GCUW(IY,IX)-GCUH(IY,IX))/GCUN(IY,IX))*
     8                  (GCUH(IY,IX)+GCUW(IY,IX)+GCUN(IY,IX))
          IUEVAL(IY,IX)=1
C  i.e. Phi = @NW - edge coeff.
          GO TO 89090
89087     IF(GCUW(IY,IX).GE.GCUN(IY,IX).AND.GCUN(IY,IX).GE.0.0)
     9    GO TO 89088
          GCC(IY,IX)=0.
          GUPCO(IY,IX)=(GCUW(IY,IX)/GCUH(IY,IX))*(GCUH(IY,IX)+
     2                GCUW(IY,IX)+GCUN(IY,IX))
          IUPVAL(IY,IX)=1
C  i.e. Phi = @NWH - corner coeff.
          GO TO 89090
89088     GCC(IY,IX)=0.
          GUECO(IY,IX)=((GCUW(IY,IX)-GCUN(IY,IX))/GCUH(IY,IX))*
     1                  (GCUH(IY,IX)+GCUW(IY,IX)+GCUN(IY,IX))
          IUEVAL(IY,IX)=8
C  i.e. Phi = @WH - edge coeff.
89090 CONTINUE
      CALL SETYX(LD12,GCC,NYDIM,NXDIM)
C      CALL PRN('CCSW',LD12)
      RETURN
C
C    High-Low
C
89100 IF(NDIREC.NE.5) GO TO 89140
C   i.e. consider flow through High face.
         CALL GETYX(LD2,GCUH,NYDIM,NXDIM)
         IF(ISWEEP.EQ.FSWEEP) GO TO 89140
C    High Coefficient
C
      DO 89130 IX=1,NX-1
      DO 89130 IY=1,NY
       IF(GCUH(IY,IX).GT.0.0.AND.GCUL(IY,IX).EQ.0.0) GO TO 89105
```

```
          GCC(IY,IX)=GCUH(IY,IX)
          GO TO 89130
89105   IF(GCUN(IY,IX).GT.GCUS(IY,IX)) GO TO 89117
         IF(GCUE(IY,IX).GT.GCUW(IY,IX)) GO TO 89111
          IF(GCUS(IY,IX).GE.GCUH(IY,IX).AND.GCUS(IY,IX).GE.GCUW(IY,IX))
     1    GO TO 89109
          IF(GCUW(IY,IX).GE.GCUH(IY,IX).AND.GCUW(IY,IX).GE.GCUS(IY,IX))
     2    GO TO 89107
           IF(GCUW(IY,IX).GE.GCUS(IY,IX).AND.GCUS(IY,IX).GE.0.0)
     3     GO TO 89106
           GCC(IY,IX)=(1.0-(GCUS(IY,IX)/GCUH(IY,IX)))*(GCUH(IY,IX)+
     4              GCUW(IY,IX)+GCUS(IY,IX))
           GO TO 89130
89106      GCC(IY,IX)=(1.0-(GCUW(IY,IX)/GCUH(IY,IX)))*(GCUH(IY,IX)+
     5              GCUW(IY,IX)+GCUS(IY,IX))
           GO TO 89130
89107      IF(GCUS(IY,IX).GE.GCUH(IY,IX).AND.GCUH(IY,IX).GE.0.0)
     9     GO TO 89108
           GCC(IY,IX)=0.
           GUECO(IY,IX)=((GCUH(IY,IX)-GCUS(IY,IX))/GCUW(IY,IX))*
     1                 (GCUH(IY,IX)+GCUW(IY,IX)+GCUS(IY,IX))
           IUEVAL(IY,IX)=8
C  i.e. Phi = @WH - edge coeff
           GO TO 89130
89108      GCC(IY,IX)=0.
           GUPCO(IY,IX)=(GCUH(IY,IX)/GCUW(IY,IX))*(GCUH(IY,IX)+
     2                GCUW(IY,IX)+GCUS(IY,IX))
           IUPVAL(IY,IX)=3
C  i.e. Phi = @SWH - corner coeff.
           GO TO 89130
89109      IF(GCUW(IY,IX).GE.GCUH(IY,IX).AND.GCUH(IY,IX).GE.0.0)
     9     GO TO 89110
           GCC(IY,IX)=0.
           GUECO(IY,IX)=((GCUH(IY,IX)-GCUW(IY,IX))/GCUS(IY,IX))*
     1                 (GCUH(IY,IX)+GCUW(IY,IX)+GCUS(IY,IX))
           IUEVAL(IY,IX)=6
C  i.e. Phi = @SH - edge coeff
           GO TO 89130
89110      GCC(IY,IX)=0.
           GUPCO(IY,IX)=(GCUH(IY,IX)/GCUS(IY,IX))*(GCUH(IY,IX)+
     2                GCUW(IY,IX)+GCUS(IY,IX))
           IUPVAL(IY,IX)=3
C  i.e. Phi = @SWH - corner coeff.
           GO TO 89130
C **********************************************************************
89111     IF(GCUS(IY,IX).GE.GCUH(IY,IX).AND.GCUS(IY,IX).GE.GCUE(IY,IX))
     1    GO TO 89115
          IF(GCUE(IY,IX).GE.GCUH(IY,IX).AND.GCUE(IY,IX).GE.GCUS(IY,IX))
     2    GO TO 89113
           IF(GCUE(IY,IX).GE.GCUS(IY,IX).AND.GCUS(IY,IX).GE.0.0)
     3     GO TO 89112
           GCC(IY,IX)=(1.0-(GCUS(IY,IX)/GCUH(IY,IX)))*(GCUH(IY,IX)+
     4              GCUE(IY,IX)+GCUS(IY,IX))
           GO TO 89130
89112      GCC(IY,IX)=(1.0-(GCUE(IY,IX)/GCUH(IY,IX)))*(GCUH(IY,IX)+
     5              GCUE(IY,IX)+GCUS(IY,IX))
           GO TO 89130
89113      IF(GCUS(IY,IX).GE.GCUH(IY,IX).AND.GCUH(IY,IX).GE.0.0)
     9     GO TO 89114
           GCC(IY,IX)=0.
           GUECO(IY,IX)=((GCUH(IY,IX)-GCUS(IY,IX))/GCUE(IY,IX))*
     1                 (GCUH(IY,IX)+GCUE(IY,IX)+GCUS(IY,IX))
           IUEVAL(IY,IX)=7
C  i.e. Phi = @EH - edge coeff
           GO TO 89130
89114      GCC(IY,IX)=0.
```

```
          GCC(IY,IX)=GCUH(IY,IX)
          GO TO 89130
89105   IF(GCUN(IY,IX).GT.GCUS(IY,IX)) GO TO 89117
         IF(GCUE(IY,IX).GT.GCUW(IY,IX)) GO TO 89111
          IF(GCUS(IY,IX).GE.GCUH(IY,IX).AND.GCUS(IY,IX).GE.GCUW(IY,IX))
     1    GO TO 89109
          IF(GCUW(IY,IX).GE.GCUH(IY,IX).AND.GCUW(IY,IX).GE.GCUS(IY,IX))
     2    GO TO 89107
           IF(GCUW(IY,IX).GE.GCUS(IY,IX).AND.GCUS(IY,IX).GE.0.0)
     3     GO TO 89106
           GCC(IY,IX)=(1.0-(GCUS(IY,IX)/GCUH(IY,IX)))*(GCUH(IY,IX)+
     4              GCUW(IY,IX)+GCUS(IY,IX))
           GO TO 89130
89106      GCC(IY,IX)=(1.0-(GCUW(IY,IX)/GCUH(IY,IX)))*(GCUH(IY,IX)+
     5              GCUW(IY,IX)+GCUS(IY,IX))
           GO TO 89130
89107      IF(GCUS(IY,IX).GE.GCUH(IY,IX).AND.GCUH(IY,IX).GE.0.0)
     9     GO TO 89108
           GCC(IY,IX)=0.
           GUECO(IY,IX)=((GCUH(IY,IX)-GCUS(IY,IX))/GCUW(IY,IX))*
     1                  (GCUH(IY,IX)+GCUW(IY,IX)+GCUS(IY,IX))
           IUEVAL(IY,IX)=8
C  i.e. Phi = @WH - edge coeff
           GO TO 89130
89108      GCC(IY,IX)=0.
           GUPCO(IY,IX)=(GCUH(IY,IX)/GCUW(IY,IX))*(GCUH(IY,IX)+
     2                GCUW(IY,IX)+GCUS(IY,IX))
           IUPVAL(IY,IX)=3
C  i.e. Phi = @SWH - corner coeff.
           GO TO 89130
89109      IF(GCUW(IY,IX).GE.GCUH(IY,IX).AND.GCUH(IY,IX).GE.0.0)
     9     GO TO 89110
           GCC(IY,IX)=0.
           GUECO(IY,IX)=((GCUH(IY,IX)-GCUW(IY,IX))/GCUS(IY,IX))*
     1                  (GCUH(IY,IX)+GCUW(IY,IX)+GCUS(IY,IX))
           IUEVAL(IY,IX)=6
C  i.e. Phi = @SH - edge coeff
           GO TO 89130
89110      GCC(IY,IX)=0.
           GUPCO(IY,IX)=(GCUH(IY,IX)/GCUS(IY,IX))*(GCUH(IY,IX)+
     2                GCUW(IY,IX)+GCUS(IY,IX))
           IUPVAL(IY,IX)=3
C  i.e. Phi = @SWH - corner coeff.
           GO TO 89130
C *****************************************************************
89111     IF(GCUS(IY,IX).GE.GCUH(IY,IX).AND.GCUS(IY,IX).GE.GCUE(IY,IX))
     1    GO TO 89115
          IF(GCUE(IY,IX).GE.GCUH(IY,IX).AND.GCUE(IY,IX).GE.GCUS(IY,IX))
     2    GO TO 89113
           IF(GCUE(IY,IX).GE.GCUS(IY,IX).AND.GCUS(IY,IX).GE.0.0)
     3     GO TO 89112
           GCC(IY,IX)=(1.0-(GCUS(IY,IX)/GCUH(IY,IX)))*(GCUH(IY,IX)+
     4              GCUE(IY,IX)+GCUS(IY,IX))
           GO TO 89130
89112      GCC(IY,IX)=(1.0-(GCUE(IY,IX)/GCUH(IY,IX)))*(GCUH(IY,IX)+
     5              GCUE(IY,IX)+GCUS(IY,IX))
           GO TO 89130
89113      IF(GCUS(IY,IX).GE.GCUH(IY,IX).AND.GCUH(IY,IX).GE.0.0)
     9     GO TO 89114
           GCC(IY,IX)=0.
           GUECO(IY,IX)=((GCUH(IY,IX)-GCUS(IY,IX))/GCUE(IY,IX))*
     1                  (GCUH(IY,IX)+GCUE(IY,IX)+GCUS(IY,IX))
           IUEVAL(IY,IX)=7
C  i.e. Phi = @EH - edge coeff
           GO TO 89130
89114      GCC(IY,IX)=0.
```

```
         GUPCO(IY,IX)=(GCUH(IY,IX)/GCUE(IY,IX))*(GCUH(IY,IX)+
     2               GCUE(IY,IX)+GCUS(IY,IX))
         IUPVAL(IY,IX)=4
C  i.e. Phi = @SEH - corner coeff.
         GO TO 89130
89115    IF(GCUE(IY,IX).GE.GCUH(IY,IX).AND.GCUH(IY,IX).GE.0.0)
     9   GO TO 89116
         GCC(IY,IX)=0.
         GUECO(IY,IX)=((GCUH(IY,IX)-GCUE(IY,IX))/GCUS(IY,IX))*
     1                 (GCUH(IY,IX)+GCUE(IY,IX)+GCUS(IY,IX))
         IUEVAL(IY,IX)=6
C  i.e. Phi = @SH - edge coeff
         GO TO 89130
89116    GCC(IY,IX)=0.
         GUPCO(IY,IX)=(GCUH(IY,IX)/GCUS(IY,IX))*(GCUH(IY,IX)+
     2               GCUE(IY,IX)+GCUS(IY,IX))
         IUPVAL(IY,IX)=4
C  i.e. Phi = @SEH - corner coeff.
         GO TO 89130
C ***********************************************************************
89117  IF(GCUE(IY,IX).GT.GCUW(IY,IX)) GO TO 89123
        IF(GCUN(IY,IX).GE.GCUH(IY,IX).AND.GCUN(IY,IX).GE.GCUW(IY,IX))
     1  GO TO 89121
        IF(GCUW(IY,IX).GE.GCUH(IY,IX).AND.GCUW(IY,IX).GE.GCUN(IY,IX))
     2  GO TO 89119
         IF(GCUW(IY,IX).GE.GCUN(IY,IX).AND.GCUN(IY,IX).GE.0.0)
     3   GO TO 89118
         GCC(IY,IX)=(1.0-(GCUN(IY,IX)/GCUH(IY,IX)))*(GCUH(IY,IX)+
     4            GCUW(IY,IX)+GCUN(IY,IX))
         GO TO 89130
89118    GCC(IY,IX)=(1.0-(GCUW(IY,IX)/GCUH(IY,IX)))*(GCUH(IY,IX)+
     5            GCUW(IY,IX)+GCUN(IY,IX))
         GO TO 89130
89119    IF(GCUN(IY,IX).GE.GCUH(IY,IX).AND.GCUH(IY,IX).GE.0.0)
     9   GO TO 89120
         GCC(IY,IX)=0.
         GUECO(IY,IX)=((GCUH(IY,IX)-GCUN(IY,IX))/GCUW(IY,IX))*
     1                 (GCUH(IY,IX)+GCUW(IY,IX)+GCUN(IY,IX))
         IUEVAL(IY,IX)=8
C  i.e. Phi = @WH - edge coeff
         GO TO 89130
89120    GCC(IY,IX)=0.
         GUPCO(IY,IX)=(GCUH(IY,IX)/GCUW(IY,IX))*(GCUH(IY,IX)+
     2               GCUW(IY,IX)+GCUN(IY,IX))
         IUPVAL(IY,IX)=1
C  i.e. Phi = @NWH - corner coeff.
         GO TO 89130
89121    IF(GCUW(IY,IX).GE.GCUH(IY,IX).AND.GCUH(IY,IX).GE.0.0)
     9   GO TO 89122
         GCC(IY,IX)=0.
         GUECO(IY,IX)=((GCUH(IY,IX)-GCUW(IY,IX))/GCUN(IY,IX))*
     1                 (GCUH(IY,IX)+GCUW(IY,IX)+GCUN(IY,IX))
         IUEVAL(IY,IX)=5
C  i.e. Phi = @NH - edge coeff
         GO TO 89130
89122    GCC(IY,IX)=0.
         GUPCO(IY,IX)=(GCUW(IY,IX)/GCUN(IY,IX))*(GCUH(IY,IX)+
     2               GCUW(IY,IX)+GCUN(IY,IX))
         IUPVAL(IY,IX)=1
C  i.e. Phi = @NWH - corner coeff.
         GO TO 89130
C ***********************************************************************
89123   IF(GCUN(IY,IX).GE.GCUH(IY,IX).AND.GCUN(IY,IX).GE.GCUE(IY,IX))
     1  GO TO 89127
        IF(GCUE(IY,IX).GE.GCUH(IY,IX).AND.GCUE(IY,IX).GE.GCUN(IY,IX))
     2  GO TO 89125
```

```
          IF(GCUE(IY,IX).GE.GCUN(IY,IX).AND.GCUN(IY,IX).GE.0.0)
     3    GO TO 89124
          GCC(IY,IX)=(1.0-(GCUN(IY,IX)/GCUH(IY,IX)))*(GCUH(IY,IX)+
     4              GCUE(IY,IX)+GCUN(IY,IX))
          GO TO 89130
89124     GCC(IY,IX)=(1.0-(GCUE(IY,IX)/GCUH(IY,IX)))*(GCUH(IY,IX)+
     5              GCUE(IY,IX)+GCUN(IY,IX))
          GO TO 89130
89125     IF(GCUN(IY,IX).GE.GCUH(IY,IX).AND.GCUH(IY,IX).GE.0.0)
     9    GO TO 89126
          GCC(IY,IX)=0.
          GUECO(IY,IX)=((GCUH(IY,IX)-GCUN(IY,IX))/GCUE(IY,IX))*
     1                 (GCUH(IY,IX)+GCUE(IY,IX)+GCUN(IY,IX))
          IUEVAL(IY,IX)=7
C  i.e. Phi = @EH - edge coeff
          GO TO 89130
89126     GCC(IY,IX)=0.
          GUPCO(IY,IX)=(GCUH(IY,IX)/GCUE(IY,IX))*(GCUH(IY,IX)+
     2                GCUE(IY,IX)+GCUN(IY,IX))
          IUPVAL(IY,IX)=2
C  i.e. Phi = @NEH - corner coeff.
          GO TO 89130
89127     IF(GCUE(IY,IX).GE.GCUH(IY,IX).AND.GCUH(IY,IX).GE.0.0)
     9    GO TO 89128
          GCC(IY,IX)=0.
          GUECO(IY,IX)=((GCUH(IY,IX)-GCUE(IY,IX))/GCUN(IY,IX))*
     1                 (GCUH(IY,IX)+GCUE(IY,IX)+GCUN(IY,IX))
          IUEVAL(IY,IX)=5
C  i.e. Phi = @NH - edge coeff
          GO TO 89130
89128     GCC(IY,IX)=0.
          GUPCO(IY,IX)=(GCUH(IY,IX)/GCUN(IY,IX))*(GCUH(IY,IX)+
     2                GCUE(IY,IX)+GCUN(IY,IX))
          IUPVAL(IY,IX)=2
C  i.e. Phi = @NEH - corner coeff.
89130 CONTINUE
      CALL SETYX(LD2,GCC,NYDIM,NXDIM)
C      CALL PRN('CCSH',LD2)
      RETURN
C
C   Low Coefficient
C
89140 IF(NDIREC.NE.6) RETURN
C   i.e. consider flow through Low face.
         CALL GETYX(LD2,GCUL,NYDIM,NXDIM)
         IF(ISWEEP.EQ.FSWEEP) RETURN
      DO 89200 IX=1,NX-1
      DO 89200 IY=1,NY
       IF(GCUL(IY,IX).GT.0.0.AND.GCUH(IY,IX).EQ.0.0) GO TO 89145
         GCC(IY,IX)=GCUL(IY,IX)
         GO TO 89200
89145  IF(GCUN(IY,IX).GT.GCUS(IY,IX)) GO TO 89157
        IF(GCUE(IY,IX).GT.GCUW(IY,IX)) GO TO 89151
         IF(GCUS(IY,IX).GE.GCUL(IY,IX).AND.GCUS(IY,IX).GE.GCUW(IY,IX))
     1   GO TO 89149
         IF(GCUW(IY,IX).GE.GCUL(IY,IX).AND.GCUW(IY,IX).GE.GCUS(IY,IX))
     2   GO TO 89147
          IF(GCUW(IY,IX).GE.GCUS(IY,IX).AND.GCUS(IY,IX).GE.0.0)
     3    GO TO 89146
          GCC(IY,IX)=(1.0-(GCUS(IY,IX)/GCUL(IY,IX)))*(GCUL(IY,IX)+
     4              GCUW(IY,IX)+GCUS(IY,IX))
          GO TO 89200
89146     GCC(IY,IX)=(1.0-(GCUW(IY,IX)/GCUL(IY,IX)))*(GCUL(IY,IX)+
     5              GCUW(IY,IX)+GCUS(IY,IX))
          GO TO 89200
89147     IF(GCUS(IY,IX).GE.GCUL(IY,IX).AND.GCUL(IY,IX).GE.0.0)
```

```
      9    GO TO 89148
           GCC(IY,IX)=0.
           GUECO(IY,IX)=((GCUL(IY,IX)-GCUS(IY,IX))/GCUW(IY,IX))*
      1                 (GCUL(IY,IX)+GCUW(IY,IX)+GCUS(IY,IX))
           IUEVAL(IY,IX)=12
C  i.e. Phi = @WL - edge coeff
           GO TO 89200
89148      GCC(IY,IX)=0.
           GUPCO(IY,IX)=(GCUL(IY,IX)/GCUW(IY,IX))*(GCUL(IY,IX)+
      2                GCUW(IY,IX)+GCUS(IY,IX))
           IUPVAL(IY,IX)=7
C  i.e. Phi = @SWL - corner coeff.
           GO TO 89200
89149      IF(GCUW(IY,IX).GE.GCUL(IY,IX).AND.GCUL(IY,IX).GE.0.0)
      9    GO TO 89150
           GCC(IY,IX)=0.
           GUECO(IY,IX)=((GCUL(IY,IX)-GCUW(IY,IX))/GCUS(IY,IX))*
      1                 (GCUL(IY,IX)+GCUW(IY,IX)+GCUS(IY,IX))
           IUEVAL(IY,IX)=10
C  i.e. Phi = @SL - edge coeff
           GO TO 89200
89150      GCC(IY,IX)=0.
           GUPCO(IY,IX)=(GCUL(IY,IX)/GCUS(IY,IX))*(GCUL(IY,IX)+
      2                GCUW(IY,IX)+GCUS(IY,IX))
           IUPVAL(IY,IX)=7
C  i.e. Phi = @SWL - corner coeff.
           GO TO 89200
C ***********************************************************************
89151     IF(GCUS(IY,IX).GE.GCUL(IY,IX).AND.GCUS(IY,IX).GE.GCUE(IY,IX))
      1   GO TO 89155
          IF(GCUE(IY,IX).GE.GCUL(IY,IX).AND.GCUE(IY,IX).GE.GCUS(IY,IX))
      2   GO TO 89153
           IF(GCUE(IY,IX).GE.GCUS(IY,IX).AND.GCUS(IY,IX).GE.0.0)
      3    GO TO 89152
           GCC(IY,IX)=(1.0-(GCUS(IY,IX)/GCUL(IY,IX)))*(GCUL(IY,IX)+
      4              GCUE(IY,IX)+GCUS(IY,IX))
           GO TO 89200
89152      GCC(IY,IX)=(1.0-(GCUE(IY,IX)/GCUL(IY,IX)))*(GCUL(IY,IX)+
      5              GCUE(IY,IX)+GCUS(IY,IX))
           GO TO 89200
89153      IF(GCUS(IY,IX).GE.GCUL(IY,IX).AND.GCUL(IY,IX).GE.0.0)
      9    GO TO 89154
           GCC(IY,IX)=0.
           GUECO(IY,IX)=((GCUL(IY,IX)-GCUS(IY,IX))/GCUE(IY,IX))*
      1                 (GCUL(IY,IX)+GCUE(IY,IX)+GCUS(IY,IX))
           IUEVAL(IY,IX)=11
C  i.e. Phi = @EL - edge coeff
           GO TO 89200
89154      GCC(IY,IX)=0.
           GUPCO(IY,IX)=(GCUL(IY,IX)/GCUE(IY,IX))*(GCUL(IY,IX)+
      2                GCUE(IY,IX)+GCUS(IY,IX))
           IUPVAL(IY,IX)=8
C  i.e. Phi = @SEL - corner coeff.
           GO TO 89200
89155      IF(GCUE(IY,IX).GE.GCUL(IY,IX).AND.GCUL(IY,IX).GE.0.0)
      9    GO TO 89156
           GCC(IY,IX)=0.
           GUECO(IY,IX)=((GCUL(IY,IX)-GCUE(IY,IX))/GCUS(IY,IX))*
      1                 (GCUL(IY,IX)+GCUE(IY,IX)+GCUS(IY,IX))
           IUEVAL(IY,IX)=10
C  i.e. Phi = @SL - edge coeff
           GO TO 89200
89156      GCC(IY,IX)=0.
           GUPCO(IY,IX)=(GCUL(IY,IX)/GCUS(IY,IX))*(GCUL(IY,IX)+
      2                GCUE(IY,IX)+GCUS(IY,IX))
           IUPVAL(IY,IX)=8
```

```
C  i.e. Phi = @SEL - corner coeff.
         GO TO 89200
C ***********************************************************************
89157   IF(GCUE(IY,IX).GT.GCUW(IY,IX)) GO TO 89163
         IF(GCUN(IY,IX).GE.GCUL(IY,IX).AND.GCUN(IY,IX).GE.GCUW(IY,IX))
     1   GO TO 89161
         IF(GCUW(IY,IX).GE.GCUL(IY,IX).AND.GCUW(IY,IX).GE.GCUN(IY,IX))
     2   GO TO 89159
          IF(GCUW(IY,IX).GE.GCUN(IY,IX).AND.GCUN(IY,IX).GE.0.0)
     3    GO TO 89158
          GCC(IY,IX)=(1.0-(GCUN(IY,IX)/GCUL(IY,IX)))*(GCUL(IY,IX)+
     4              GCUW(IY,IX)+GCUN(IY,IX))
          GO TO 89200
89158     GCC(IY,IX)=(1.0-(GCUW(IY,IX)/GCUL(IY,IX)))*(GCUL(IY,IX)+
     5              GCUW(IY,IX)+GCUN(IY,IX))
          GO TO 89200
89159     IF(GCUN(IY,IX).GE.GCUL(IY,IX).AND.GCUL(IY,IX).GE.0.0)
     9    GO TO 89160
          GCC(IY,IX)=0.
          GUECO(IY,IX)=((GCUL(IY,IX)-GCUN(IY,IX))/GCUW(IY,IX))*
     1                 (GCUL(IY,IX)+GCUW(IY,IX)+GCUN(IY,IX))
          IUEVAL(IY,IX)=12
C  i.e. Phi = @WL - edge coeff
          GO TO 89200
89160     GCC(IY,IX)=0.
          GUPCO(IY,IX)=(GCUL(IY,IX)/GCUW(IY,IX))*(GCUL(IY,IX)+
     2                GCUW(IY,IX)+GCUN(IY,IX))
          IUPVAL(IY,IX)=5
C  i.e. Phi = @NWL - corner coeff.
          GO TO 89200
89161     IF(GCUW(IY,IX).GE.GCUL(IY,IX).AND.GCUL(IY,IX).GE.0.0)
     9    GO TO 89162
          GCC(IY,IX)=0.
          GUECO(IY,IX)=((GCUL(IY,IX)-GCUW(IY,IX))/GCUN(IY,IX))*
     1                 (GCUL(IY,IX)+GCUW(IY,IX)+GCUN(IY,IX))
          IUEVAL(IY,IX)=9
C  i.e. Phi = @NL - edge coeff
          GO TO 89200
89162     GCC(IY,IX)=0.
          GUPCO(IY,IX)=(GCUL(IY,IX)/GCUN(IY,IX))*(GCUL(IY,IX)+
     2                GCUW(IY,IX)+GCUN(IY,IX))
          IUPVAL(IY,IX)=5
C  i.e. Phi = @NWL - corner coeff.
          GO TO 89200
C ***********************************************************************
89163    IF(GCUN(IY,IX).GE.GCUL(IY,IX).AND.GCUN(IY,IX).GE.GCUE(IY,IX))
     1   GO TO 89167
         IF(GCUE(IY,IX).GE.GCUL(IY,IX).AND.GCUE(IY,IX).GE.GCUN(IY,IX))
     2   GO TO 89165
          IF(GCUE(IY,IX).GE.GCUN(IY,IX).AND.GCUN(IY,IX).GE.0.0)
     3    GO TO 89164
          GCC(IY,IX)=(1.0-(GCUN(IY,IX)/GCUL(IY,IX)))*(GCUL(IY,IX)+
     4              GCUE(IY,IX)+GCUN(IY,IX))
          GO TO 89200
89164     GCC(IY,IX)=(1.0-(GCUE(IY,IX)/GCUL(IY,IX)))*(GCUL(IY,IX)+
     5              GCUE(IY,IX)+GCUN(IY,IX))
          GO TO 89200
89165     IF(GCUN(IY,IX).GE.GCUL(IY,IX).AND.GCUL(IY,IX).GE.0.0)
     9    GO TO 89166
          GCC(IY,IX)=0.
          GUECO(IY,IX)=((GCUL(IY,IX)-GCUN(IY,IX))/GCUE(IY,IX))*
     1                 (GCUL(IY,IX)+GCUE(IY,IX)+GCUN(IY,IX))
          IUEVAL(IY,IX)=11
C  i.e. Phi = @EL - edge coeff
          GO TO 89200
89166     GCC(IY,IX)=0.
```

```
          GUPCO(IY,IX)=(GCUL(IY,IX)[illegible]GCUE(IY,IX))*(GCUL(IY,IX)+
     2                GCUE(IY,IX)+GCUN(IY,IX))
          IUPVAL(IY,IX)=6
C  i.e. Phi = @NEL - corner coeff.
          GO TO 89200
89167     IF(GCUE(IY,IX).GE.GCUL(IY,IX).AND.GCUL(IY,IX).GE.0.0)
     9    GO TO 89168
          GCC(IY,IX)=0.
          GUECO(IY,IX)=((GCUL(IY,IX)-GCUE(IY,IX))/GCUN(IY,IX))*
     1                 (GCUL(IY,IX)+GCUE(IY,IX)+GCUN(IY,IX))
          IUEVAL(IY,IX)=9
C  i.e. Phi = @NL - edge coeff
          GO TO 89200
89168     GCC(IY,IX)=0.
          GUPCO(IY,IX)=(GCUL(IY,IX)/GCUN(IY,IX))*(GCUL(IY,IX)+
     2                GCUE(IY,IX)+GCUN(IY,IX))
          IUPVAL(IY,IX)=6
C  i.e. Phi = @NEL - corner coeff.
89200 CONTINUE
      CALL SETYX(LD2,GCC,NYDIM,NXDIM)
C      CALL PRN('CCSL',LD2)
      RETURN
      END
```

```
C***********************************************************************
      SUBROUTINE GROUND2
      INCLUDE 'lp16/d_earth/SATEAR'
      INCLUDE 'lp16/d_earth/GRDLOC'
      INCLUDE 'lp16/d_earth/GRDEAR'
      INCLUDE 'lp16/d_earth/GRDBFC'
CXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX USER SECTION STARTS:
C
C 1   Set dimensions of data-for-GROUND arrays here. WARNING: the
C     corresponding arrays in the MAIN program of the satellite
C     and EARTH must have the same dimensions.
      PARAMETER (NLG=20, NIG=20, NRG=100, NCG=10)
C
      COMMON/LGRND/LG(NLG)/IGRND/IG(NIG)/RGRND/RG(NRG)/CGRND/CG(NCG)
      LOGICAL LG
      CHARACTER*4 CG
      COMMON/GUGRND/GCSN(60,80),GCSS(60,80),GCSE(60,80),GCSW(60,80),
     7              GCSL(60,80),GCSH(60,80),GSPCO(60,80),GSECO(60,80),
     8              GCUN(60,80),GCUS(60,80),GCUE(60,80),GCUW(60,80),
     7              GCUL(60,80),GCUH(60,80),GUPCO(60,80),GUECO(60,80),
     8              GCVN(60,80),GCVS(60,80),GCVE(60,80),GCVW(60,80),
     7              GCVL(60,80),GCVH(60,80),GVPCO(60,80),GVECO(60,80),
     8              GCWN(60,80),GCWS(60,80),GCWE(60,80),GCWW(60,80),
     7              GCWL(60,80),GCWH(60,80),GWPCO(60,80),GWECO(60,80)
      COMMON/IUGRND/ISPVAL(60,80),ISEVAL(60,80),IUPVAL(60,80),
     7              IUEVAL(60,80),IVEVAL(60,80),IVPVAL(60,80),
     8              IWEVAL(60,80),IWPVAL(60,80)
C
C 2   User dimensions own arrays here, for example:
C     DIMENSION GUH(10,10),GUC(10,10),GUX(10,10),GUZ(10)
      DIMENSION GCC(60,80),GCMH(60,80),GCML(60,80),
     1          GCE(60,80),GEMP(60,80),GEMH(60,80),GEML(60,80)
C
C 3   User places his data statements here, for example:
      DATA NXDIM,NYDIM/80,60/
C
C 4   Insert own coding below as desired, guided by GREX examples.
C     Note that the satellite-to-GREX special data in the labelled
C     COMMONs /RSG/, /ISG/, /LSG/ and /CSG/ can be included and
C     used below but the user must check GREX for any conflicting
C     uses. The same comment applies to the EARTH-spare working
C     arrays EASP1, EASP2,....EASP20. In addition to the EASPs,
C     there are 10 GRound-earth SPare arrays, GRSP1,...,GRSP10,
C     supplied solely for the user, which are not used by GREX. If
C     the call to GREX has been deactivated then all of the arrays
C     may be used without reservation.
C
      IF(.NOT.(LG(2))) RETURN
      IF(INDVAR.EQ.5) GO TO 88949
      RETURN
88949 CONTINUE
C
C  Corrected Coefficients For U-Equation
C  ***************************************
C
C  North-South
C
      IF(NDIREC.NE.1) GO TO 89020
C  i.e. consider flow through North and South faces.
        CALL GETYX(LD11,GCVN,NYDIM,NXDIM)
        CALL GETYX(LD12,GCVS,NYDIM,NXDIM)
        IF(ISWEEP.EQ.FSWEEP) GO TO 89020
C
C
      DO 88980 IX=1,NX
      DO 88980 IY=1,NY-1
```

```
      IF(GCVN(IY,IX).GT.0.0.AND.GCVS(IY,IX).EQ.0.0) GO TO 88955
C......i.e.consider scheme ONLY with a max. of 3 +'ve fluxes...else..
88950  GCC(IY,IX)=GCVN(IY,IX)
       GO TO 88980
88955 IF(GCVE(IY,IX).GT.GCVW(IY,IX)) GO TO 88967
       IF(GCVH(IY,IX).GT.GCVL(IY,IX)) GO TO 88961
C......i.e.m(N),m(W)and m(L) involved...
       IF(GCVL(IY,IX).GE.GCVN(IY,IX).AND.GCVL(IY,IX).GE.GCVW(IY,IX))
     1 GO TO 88959
C......i.e.m(L) dominant...
       IF(GCVW(IY,IX).GE.GCVN(IY,IX).AND.GCVW(IY,IX).GE.GCVL(IY,IX))
     2 GO TO 88957
C......i.e.m(W) dominant...otherwise m(N) is dominant...
        IF(GCVW(IY,IX).GE.GCVL(IY,IX).AND.GCVL(IY,IX).GE.0.0)
     3  GO TO 88956
C......i.e.flow specification is m(N)>m(W)>m(L)>=0.0...otherwise..
C..........flow specification is m(N)>m(L)>m(W)>=0.0...
        GCC(IY,IX)=(1.0-(GCVL(IY,IX)/GCVN(IY,IX)))*(GCVL(IY,IX)+
     4          GCVW(IY,IX)+GCVN(IY,IX))
        GO TO 88980
88956   GCC(IY,IX)=(1.0-(GCVW(IY,IX)/GCVN(IY,IX)))*(GCVL(IY,IX)+
     5          GCVW(IY,IX)+GCVN(IY,IX))
        GO TO 88980
88957   IF(GCVN(IY,IX).GE.GCVL(IY,IX).AND.GCVL(IY,IX).GE.0.0)
     6  GO TO 88958
        GCC(IY,IX)=0.
        GVPCO(IY,IX)=(GCVN(IY,IX)/GCVW(IY,IX))*(GCVL(IY,IX)+
     7            GCVW(IY,IX)+GCVN(IY,IX))
        IVPVAL(IY,IX)=5
C  i.e. Phi = @NWL - corner coeff.
        GO TO 88980
88958   GCC(IY,IX)=0.
        GVECO(IY,IX)=((GCVN(IY,IX)-GCVL(IY,IX))/GCVW(IY,IX))*
     8            (GCVL(IY,IX)+GCVW(IY,IX)+GCVN(IY,IX))
        IVEVAL(IY,IX)=1
C  i.e. Phi = @NW - edge coeff.
        GO TO 88980
88959   IF(GCVW(IY,IX).GE.GCVN(IY,IX).AND.GCVN(IY,IX).GE.0.0)
     9  GO TO 88960
        GCC(IY,IX)=0.
        GVECO(IY,IX)=((GCVN(IY,IX)-GCVW(IY,IX))/GCVL(IY,IX))*
     1            (GCVL(IY,IX)+GCVW(IY,IX)+GCVN(IY,IX))
        IVEVAL(IY,IX)=9
C  i.e. Phi = @NL - edge coeff
        GO TO 88980
88960   GCC(IY,IX)=0.
        GVPCO(IY,IX)=(GCVN(IY,IX)/GCVL(IY,IX))*(GCVL(IY,IX)+
     2            GCVW(IY,IX)+GCVN(IY,IX))
        IVPVAL(IY,IX)=5
C  i.e. Phi = @NWL - corner coeff.
        GO TO 88980
C ********************************************************************
88961  IF(GCVH(IY,IX).GE.GCVN(IY,IX).AND.GCVH(IY,IX).GE.GCVW(IY,IX))
     1 GO TO 88965
       IF(GCVW(IY,IX).GE.GCVN(IY,IX).AND.GCVW(IY,IX).GE.GCVH(IY,IX))
     2 GO TO 88963
        IF(GCVW(IY,IX).GE.GCVH(IY,IX).AND.GCVH(IY,IX).GE.0.0)
     3  GO TO 88962
        GCC(IY,IX)=(1.0-(GCVH(IY,IX)/GCVN(IY,IX)))*(GCVH(IY,IX)+
     4          GCVW(IY,IX)+GCVN(IY,IX))
        GO TO 88980
88962   GCC(IY,IX)=(1.0-(GCVW(IY,IX)/GCVN(IY,IX)))*(GCVH(IY,IX)+
     5          GCVW(IY,IX)+GCVN(IY,IX))
        GO TO 88980
88963   IF(GCVN(IY,IX).GE.GCVH(IY,IX).AND.GCVH(IY,IX).GE.0.0)
     6  GO TO 88964
```

```
          GCC(IY,IX)=0.
          GVPCO(IY,IX)=(GCVN(IY,IX)/GCVW(IY,IX))*(GCVH(IY,IX)+
     7                 GCVW(IY,IX)+GCVN(IY,IX))
          IVPVAL(IY,IX)=1
C  i.e. Phi = @NWH - corner coeff.
          GO TO 88980
88964     GCC(IY,IX)=0.
          GVECO(IY,IX)=((GCVN(IY,IX)-GCVH(IY,IX))/GCVW(IY,IX))*
     8                   (GCVH(IY,IX)+GCVW(IY,IX)+GCVN(IY,IX))
          IVEVAL(IY,IX)=1
C  i.e. Phi = @NW - edge coeff.
          GO TO 88980
88965     IF(GCVW(IY,IX).GE.GCVN(IY,IX).AND.GCVN(IY,IX).GE.0.0)
     9    GO TO 88966
          GCC(IY,IX)=0.
          GVECO(IY,IX)=((GCVN(IY,IX)-GCVW(IY,IX))/GCVH(IY,IX))*
     1                   (GCVH(IY,IX)+GCVW(IY,IX)+GCVN(IY,IX))
          IVEVAL(IY,IX)=5
C  i.e. Phi = @NH - edge coeff
          GO TO 88980
88966     GCC(IY,IX)=0.
          GVPCO(IY,IX)=(GCVN(IY,IX)/GCVH(IY,IX))*(GCVH(IY,IX)+
     2                 GCVW(IY,IX)+GCVN(IY,IX))
          IVPVAL(IY,IX)=1
C  i.e. Phi = @NWH - corner coeff.
          GO TO 88980
C ***********************************************************************
88967   IF(GCVH(IY,IX).GT.GCVL(IY,IX)) GO TO 88973
         IF(GCVL(IY,IX).GE.GCVN(IY,IX).AND.GCVL(IY,IX).GE.GCVE(IY,IX))
     1   GO TO 88971
         IF(GCVE(IY,IX).GE.GCVN(IY,IX).AND.GCVE(IY,IX).GE.GCVL(IY,IX))
     2   GO TO 88969
          IF(GCVE(IY,IX).GE.GCVL(IY,IX).AND.GCVL(IY,IX).GE.0.0)
     3    GO TO 88968
          GCC(IY,IX)=(1.0-(GCVL(IY,IX)/GCVN(IY,IX)))*(GCVL(IY,IX)+
     4               GCVE(IY,IX)+GCVN(IY,IX))
          GO TO 88980
88968     GCC(IY,IX)=(1.0-(GCVE(IY,IX)/GCVN(IY,IX)))*(GCVL(IY,IX)+
     5               GCVE(IY,IX)+GCVN(IY,IX))
          GO TO 88980
88969     IF(GCVN(IY,IX).GE.GCVL(IY,IX).AND.GCVL(IY,IX).GE.0.0)
     6    GO TO 88970
          GCC(IY,IX)=0.
          GVPCO(IY,IX)=(GCVN(IY,IX)/GCVE(IY,IX))*(GCVL(IY,IX)+
     7                 GCVE(IY,IX)+GCVN(IY,IX))
          IVPVAL(IY,IX)=6
C  i.e. Phi = @NEL - corner coeff.
          GO TO 88980
88970     GCC(IY,IX)=0.
          GVECO(IY,IX)=((GCVN(IY,IX)-GCVL(IY,IX))/GCVE(IY,IX))*
     8                   (GCVL(IY,IX)+GCVE(IY,IX)+GCVN(IY,IX))
          IVEVAL(IY,IX)=2
C  i.e. Phi = @NE - edge coeff.
          GO TO 88980
88971     IF(GCVE(IY,IX).GE.GCVN(IY,IX).AND.GCVN(IY,IX).GE.0.0)
     9    GO TO 88972
          GCC(IY,IX)=0.
          GVECO(IY,IX)=((GCVN(IY,IX)-GCVE(IY,IX))/GCVL(IY,IX))*
     1                   (GCVL(IY,IX)+GCVE(IY,IX)+GCVN(IY,IX))
          IVEVAL(IY,IX)=9
C  i.e. Phi = @NL - edge coeff
          GO TO 88980
88972     GCC(IY,IX)=0.
          GVPCO(IY,IX)=(GCVN(IY,IX)/GCVL(IY,IX))*(GCVL(IY,IX)+
     2                 GCVE(IY,IX)+GCVN(IY,IX))
          IVPVAL(IY,IX)=6
```

```
C  i.e. Phi = @NEL - corner coeff.
          GO TO 88980
C ***********************************************************************
88973    IF(GCVH(IY,IX).GE.GCVN(IY,IX).AND.GCVH(IY,IX).GE.GCVE(IY,IX))
     1   GO TO 88977
         IF(GCVE(IY,IX).GE.GCVN(IY,IX).AND.GCVE(IY,IX).GE.GCVH(IY,IX))
     2   GO TO 88975
          IF(GCVE(IY,IX).GE.GCVH(IY,IX).AND.GCVH(IY,IX).GE.0.0)
     3    GO TO 88974
          GCC(IY,IX)=(1.0-(GCVH(IY,IX)/GCVN(IY,IX)))*(GCVH(IY,IX)+
     4              GCVE(IY,IX)+GCVN(IY,IX))
          GO TO 88980
88974     GCC(IY,IX)=(1.0-(GCVE(IY,IX)/GCVN(IY,IX)))*(GCVH(IY,IX)+
     5              GCVE(IY,IX)+GCVN(IY,IX))
          GO TO 88980
88975     IF(GCVN(IY,IX).GE.GCVH(IY,IX).AND.GCVH(IY,IX).GE.0.0)
     6    GO TO 88976
          GCC(IY,IX)=0.
          GVPCO(IY,IX)=(GCVN(IY,IX)/GCVE(IY,IX))*(GCVH(IY,IX)+
     7                GCVE(IY,IX)+GCVN(IY,IX))
          IVPVAL(IY,IX)=2
C  i.e. Phi = @NEH - corner coeff.
          GO TO 88980
88976     GCC(IY,IX)=0.
          GVECO(IY,IX)=((GCVN(IY,IX)-GCVH(IY,IX))/GCVE(IY,IX))*
     8                 (GCVH(IY,IX)+GCVE(IY,IX)+GCVN(IY,IX))
          IVEVAL(IY,IX)=2
C  i.e. Phi = @NE - edge coeff.
          GO TO 88980
88977     IF(GCVE(IY,IX).GE.GCVN(IY,IX).AND.GCVN(IY,IX).GE.0.0)
     9    GO TO 88978
          GCC(IY,IX)=0.
          GVECO(IY,IX)=((GCVN(IY,IX)-GCVE(IY,IX))/GCVH(IY,IX))*
     1                 (GCVH(IY,IX)+GCVE(IY,IX)+GCVN(IY,IX))
          IVEVAL(IY,IX)=5
C  i.e. Phi = @NH - edge coeff
          GO TO 88980
88978     GCC(IY,IX)=0.
          GVPCO(IY,IX)=(GCVN(IY,IX)/GCVH(IY,IX))*(GCVH(IY,IX)+
     2                GCVE(IY,IX)+GCVN(IY,IX))
          IVPVAL(IY,IX)=2
C  i.e. Phi = @NEH - corner coeff.
88980 CONTINUE
      CALL SETYX(LD11,GCC,NYDIM,NXDIM)
C      CALL PRN('CCSN',LD11)
C
C   South Coefficient
C
      DO 89010 IX=1,NX
      DO 89010 IY=1,NY-1
       IF(GCVS(IY,IX).GT.0.0.AND.GCVN(IY,IX).EQ.0.0) GO TO 88985
88982    GCC(IY,IX)=GCVS(IY,IX)
         GO TO 89010
88985  IF(GCVE(IY,IX).GT.GCVW(IY,IX)) GO TO 88997
        IF(GCVH(IY,IX).GT.GCVL(IY,IX)) GO TO 88991
         IF(GCVL(IY,IX).GE.GCVS(IY,IX).AND.GCVL(IY,IX).GE.GCVW(IY,IX))
     1   GO TO 88989
         IF(GCVW(IY,IX).GE.GCVS(IY,IX).AND.GCVW(IY,IX).GE.GCVL(IY,IX))
     2   GO TO 88987
          IF(GCVW(IY,IX).GE.GCVL(IY,IX).AND.GCVL(IY,IX).GE.0.0)
     3    GO TO 88986
          GCC(IY,IX)=(1.0-(GCVL(IY,IX)/GCVS(IY,IX)))*(GCVL(IY,IX)+
     4              GCVW(IY,IX)+GCVS(IY,IX))
          GO TO 89010
88986     GCC(IY,IX)=(1.0-(GCVW(IY,IX)/GCVS(IY,IX)))*(GCVL(IY,IX)+
     5              GCVW(IY,IX)+GCVS(IY,IX))
```

```
            GO TO 89010
88987       IF(GCVS(IY,IX).GE.GCVL(IY,IX).AND.GCVL(IY,IX).GE.0.0)
     6      GO TO 88988
            GCC(IY,IX)=0.
            GVPCO(IY,IX)=(GCVS(IY,IX)/GCVW(IY,IX))*(GCVL(IY,IX)+
     7                 GCVW(IY,IX)+GCVS(IY,IX))
            IVPVAL(IY,IX)=7
C  i.e. Phi = @SWL - corner coeff.
            GO TO 89010
88988       GCC(IY,IX)=0.
            GVECO(IY,IX)=((GCVS(IY,IX)-GCVL(IY,IX))/GCVW(IY,IX))*
     8                    (GCVL(IY,IX)+GCVW(IY,IX)+GCVS(IY,IX))
            IVEVAL(IY,IX)=3
C  i.e. Phi = @SW - edge coeff.
            GO TO 89010
88989       IF(GCVW(IY,IX).GE.GCVS(IY,IX).AND.GCVS(IY,IX).GE.0.0)
     9      GO TO 88990
            GCC(IY,IX)=0.
            GVECO(IY,IX)=((GCVS(IY,IX)-GCVW(IY,IX))/GCVL(IY,IX))*
     1                    (GCVL(IY,IX)+GCVW(IY,IX)+GCVS(IY,IX))
            IVEVAL(IY,IX)=10
C  i.e. Phi = @SL - edge coeff
            GO TO 89010
88990       GCC(IY,IX)=0.
            GVPCO(IY,IX)=(GCVS(IY,IX)/GCVL(IY,IX))*(GCVL(IY,IX)+
     2                 GCVW(IY,IX)+GCVS(IY,IX))
            IVPVAL(IY,IX)=7
C  i.e. Phi = @SWL - corner coeff.
            GO TO 89010
C ***********************************************************************
88991       IF(GCVH(IY,IX).GE.GCVS(IY,IX).AND.GCVH(IY,IX).GE.GCVW(IY,IX))
     1     GO TO 88995
           IF(GCVW(IY,IX).GE.GCVS(IY,IX).AND.GCVW(IY,IX).GE.GCVH(IY,IX))
     2     GO TO 88993
            IF(GCVW(IY,IX).GE.GCVH(IY,IX).AND.GCVH(IY,IX).GE.0.0)
     3      GO TO 88992
            GCC(IY,IX)=(1.0-(GCVH(IY,IX)/GCVS(IY,IX)))*(GCVH(IY,IX)+
     4               GCVW(IY,IX)+GCVS(IY,IX))
            GO TO 89010
88992       GCC(IY,IX)=(1.0-(GCVW(IY,IX)/GCVS(IY,IX)))*(GCVH(IY,IX)+
     5               GCVW(IY,IX)+GCVS(IY,IX))
            GO TO 89010
88993       IF(GCVS(IY,IX).GE.GCVH(IY,IX).AND.GCVH(IY,IX).GE.0.0)
     6      GO TO 88994
            GCC(IY,IX)=0.
            GVPCO(IY,IX)=(GCVS(IY,IX)/GCVW(IY,IX))*(GCVH(IY,IX)+
     7                 GCVW(IY,IX)+GCVS(IY,IX))
            IVPVAL(IY,IX)=3
C  i.e. Phi = @SWH - corner coeff.
            GO TO 89010
88994       GCC(IY,IX)=0.
            GVECO(IY,IX)=((GCVS(IY,IX)-GCVH(IY,IX))/GCVW(IY,IX))*
     8                    (GCVH(IY,IX)+GCVW(IY,IX)+GCVS(IY,IX))
            IVEVAL(IY,IX)=3
C  i.e. Phi = @SW - edge coeff.
            GO TO 89010
88995       IF(GCVW(IY,IX).GE.GCVS(IY,IX).AND.GCVS(IY,IX).GE.0.0)
     9      GO TO 88996
            GCC(IY,IX)=0.
            GVECO(IY,IX)=((GCVS(IY,IX)-GCVW(IY,IX))/GCVH(IY,IX))*
     1                    (GCVH(IY,IX)+GCVW(IY,IX)+GCVS(IY,IX))
            IVEVAL(IY,IX)=6
C  i.e. Phi = @SH - edge coeff
            GO TO 89010
88996       GCC(IY,IX)=0.
            GVPCO(IY,IX)=(GCVS(IY,IX)/GCVH(IY,IX))*(GCVH(IY,IX)+
```

```
      GO TO 89010
88987 IF(GCVS(IY,IX).GE.GCVL(IY,IX).AND.GCVL(IY,IX).GE.0.0)
     6 GO TO 88988
      GCC(IY,IX)=0.
      GVPCO(IY,IX)=(GCVS(IY,IX)/GCVW(IY,IX))*(GCVL(IY,IX)+
     7            GCVW(IY,IX)+GCVS(IY,IX))
      IVPVAL(IY,IX)=7
C  i.e. Phi = @SWL - corner coeff.
      GO TO 89010
88988 GCC(IY,IX)=0.
      GVECO(IY,IX)=((GCVS(IY,IX)-GCVL(IY,IX))/GCVW(IY,IX))*
     8             (GCVL(IY,IX)+GCVW(IY,IX)+GCVS(IY,IX))
      IVEVAL(IY,IX)=3
C  i.e. Phi = @SW - edge coeff.
      GO TO 89010
88989 IF(GCVW(IY,IX).GE.GCVS(IY,IX).AND.GCVS(IY,IX).GE.0.0)
     9 GO TO 88990
      GCC(IY,IX)=0.
      GVECO(IY,IX)=((GCVS(IY,IX)-GCVW(IY,IX))/GCVL(IY,IX))*
     1             (GCVL(IY,IX)+GCVW(IY,IX)+GCVS(IY,IX))
      IVEVAL(IY,IX)=10
C  i.e. Phi = @SL - edge coeff
      GO TO 89010
88990 GCC(IY,IX)=0.
      GVPCO(IY,IX)=(GCVS(IY,IX)/GCVL(IY,IX))*(GCVL(IY,IX)+
     2            GCVW(IY,IX)+GCVS(IY,IX))
      IVPVAL(IY,IX)=7
C  i.e. Phi = @SWL - corner coeff.
      GO TO 89010
C **********************************************************************
88991 IF(GCVH(IY,IX).GE.GCVS(IY,IX).AND.GCVH(IY,IX).GE.GCVW(IY,IX))
     1GO TO 88995
     IF(GCVW(IY,IX).GE.GCVS(IY,IX).AND.GCVW(IY,IX).GE.GCVH(IY,IX))
     2GO TO 88993
      IF(GCVW(IY,IX).GE.GCVH(IY,IX).AND.GCVH(IY,IX).GE.0.0)
     3 GO TO 88992
      GCC(IY,IX)=(1.0-(GCVH(IY,IX)/GCVS(IY,IX)))*(GCVH(IY,IX)+
     4           GCVW(IY,IX)+GCVS(IY,IX))
      GO TO 89010
88992 GCC(IY,IX)=(1.0-(GCVW(IY,IX)/GCVS(IY,IX)))*(GCVH(IY,IX)+
     5           GCVW(IY,IX)+GCVS(IY,IX))
      GO TO 89010
88993 IF(GCVS(IY,IX).GE.GCVH(IY,IX).AND.GCVH(IY,IX).GE.0.0)
     6 GO TO 88994
      GCC(IY,IX)=0.
      GVPCO(IY,IX)=(GCVS(IY,IX)/GCVW(IY,IX))*(GCVH(IY,IX)+
     7            GCVW(IY,IX)+GCVS(IY,IX))
      IVPVAL(IY,IX)=3
C  i.e. Phi = @SWH - corner coeff.
      GO TO 89010
88994 GCC(IY,IX)=0.
      GVECO(IY,IX)=((GCVS(IY,IX)-GCVH(IY,IX))/GCVW(IY,IX))*
     8             (GCVH(IY,IX)+GCVW(IY,IX)+GCVS(IY,IX))
      IVEVAL(IY,IX)=3
C  i.e. Phi = @SW - edge coeff.
      GO TO 89010
88995 IF(GCVW(IY,IX).GE.GCVS(IY,IX).AND.GCVS(IY,IX).GE.0.0)
     9 GO TO 88996
      GCC(IY,IX)=0.
      GVECO(IY,IX)=((GCVS(IY,IX)-GCVW(IY,IX))/GCVH(IY,IX))*
     1             (GCVH(IY,IX)+GCVW(IY,IX)+GCVS(IY,IX))
      IVEVAL(IY,IX)=6
C  i.e. Phi = @SH - edge coeff
      GO TO 89010
88996 GCC(IY,IX)=0.
      GVPCO(IY,IX)=(GCVS(IY,IX)/GCVH(IY,IX))*(GCVH(IY,IX)+
```

```
          GO TO 89010
89006     GCC(IY,IX)=0.
          GVECO(IY,IX)=((GCVS(IY,IX)-GCVH(IY,IX))/GCVE(IY,IX))*
     8                 (GCVH(IY,IX)+GCVE(IY,IX)+GCVS(IY,IX))
          IVEVAL(IY,IX)=4
C  i.e. Phi = @SE - edge coeff.
          GO TO 89010
89007     IF(GCVE(IY,IX).GE.GCVS(IY,IX).AND.GCVS(IY,IX).GE.0.0)
     9    GO TO 89008
          GCC(IY,IX)=0.
          GVECO(IY,IX)=((GCVS(IY,IX)-GCVE(IY,IX))/GCVH(IY,IX))*
     1                 (GCVH(IY,IX)+GCVE(IY,IX)+GCVS(IY,IX))
          IVEVAL(IY,IX)=6
C  i.e. Phi = @SH - edge coeff
          GO TO 89010
89008     GCC(IY,IX)=0.
          GVPCO(IY,IX)=(GCVS(IY,IX)/GCVH(IY,IX))*(GCVH(IY,IX)+
     2                GCVE(IY,IX)+GCVS(IY,IX))
          IVPVAL(IY,IX)=4
C  i.e. Phi = @SEH - corner coeff.
89010 CONTINUE
      CALL SETYX(LD12,GCC,NYDIM,NXDIM)
C      CALL PRN('CCSS',LD12)
      RETURN
C
C    East-West
C
89020 IF(NDIREC.NE.3) GO TO 89100
C   i.e. consider flow through East and West faces.
         CALL GETYX(LD11,GCVE,NYDIM,NXDIM)
         CALL GETYX(LD12,GCVW,NYDIM,NXDIM)
         IF(ISWEEP.EQ.FSWEEP) GO TO 89100
C
C    East Coefficient
C
      DO 89060 IX=1,NX
      DO 89060 IY=1,NY-1
       IF(GCVE(IY,IX).GT.0.0.AND.GCVW(IY,IX).EQ.0.0) GO TO 89025
89022    GCC(IY,IX)=GCVE(IY,IX)
         GO TO 89060
89025  IF(GCVN(IY,IX).GT.GCVS(IY,IX)) GO TO 89037
        IF(GCVH(IY,IX).GT.GCVL(IY,IX)) GO TO 89031
         IF(GCVL(IY,IX).GE.GCVE(IY,IX).AND.GCVL(IY,IX).GE.GCVS(IY,IX))
     1   GO TO 89029
         IF(GCVS(IY,IX).GE.GCVE(IY,IX).AND.GCVS(IY,IX).GE.GCVL(IY,IX))
     2   GO TO 89027
          IF(GCVS(IY,IX).GE.GCVL(IY,IX).AND.GCVL(IY,IX).GE.0.0)
     3    GO TO 89026
          GCC(IY,IX)=(1.0-(GCVL(IY,IX)/GCVE(IY,IX)))*(GCVL(IY,IX)+
     4              GCVE(IY,IX)+GCVS(IY,IX))
          GO TO 89060
89026     GCC(IY,IX)=(1.0-(GCVS(IY,IX)/GCVE(IY,IX)))*(GCVL(IY,IX)+
     5              GCVE(IY,IX)+GCVS(IY,IX))
          GO TO 89060
89027     IF(GCVE(IY,IX).GE.GCVL(IY,IX).AND.GCVL(IY,IX).GE.0.0)
     6    GO TO 89028
          GCC(IY,IX)=0.
          GVPCO(IY,IX)=(GCVE(IY,IX)/GCVS(IY,IX))*(GCVL(IY,IX)+
     7                GCVE(IY,IX)+GCVS(IY,IX))
          IVPVAL(IY,IX)=8
C  i.e. Phi = @SEL - corner coeff.
          GO TO 89060
89028     GCC(IY,IX)=0.
          GVECO(IY,IX)=((GCVE(IY,IX)-GCVL(IY,IX))/GCVS(IY,IX))*
     8                 (GCVL(IY,IX)+GCVE(IY,IX)+GCVS(IY,IX))
          IVEVAL(IY,IX)=4
```

```
C  i.e. Phi = @SE - edge coeff.
          GO TO 89060
89029     IF(GCVE(IY,IX).GE.GCVS(IY,IX).AND.GCVS(IY,IX).GE.0.0)
     9    GO TO 89030
          GCC(IY,IX)=0.
          GVPCO(IY,IX)=(GCVE(IY,IX)/GCVL(IY,IX))*(GCVL(IY,IX)+
     2                 GCVE(IY,IX)+GCVS(IY,IX))
          IVPVAL(IY,IX)=8
C  i.e. Phi = @SEL - corner coeff.
          GO TO 89060
89030     GCC(IY,IX)=0.
          GVECO(IY,IX)=((GCVE(IY,IX)-GCVS(IY,IX))/GCVL(IY,IX))*
     1                   (GCVL(IY,IX)+GCVE(IY,IX)+GCVS(IY,IX))
          IVEVAL(IY,IX)=11
C  i.e. Phi = @EL - edge coeff
          GO TO 89060
C ***********************************************************************
89031    IF(GCVH(IY,IX).GE.GCVE(IY,IX).AND.GCVH(IY,IX).GE.GCVS(IY,IX))
     1   GO TO 89035
         IF(GCVS(IY,IX).GE.GCVE(IY,IX).AND.GCVS(IY,IX).GE.GCVH(IY,IX))
     2   GO TO 89033
          IF(GCVS(IY,IX).GE.GCVH(IY,IX).AND.GCVH(IY,IX).GE.0.0)
     3    GO TO 89032
          GCC(IY,IX)=(1.0-(GCVH(IY,IX)/GCVE(IY,IX)))*(GCVH(IY,IX)+
     4               GCVE(IY,IX)+GCVS(IY,IX))
89032     GCC(IY,IX)=(1.0-(GCVS(IY,IX)/GCVE(IY,IX)))*(GCVH(IY,IX)+
     5               GCVE(IY,IX)+GCVS(IY,IX))
          GO TO 89060
89033     IF(GCVE(IY,IX).GE.GCVH(IY,IX).AND.GCVH(IY,IX).GE.0.0)
     6    GO TO 89034
          GCC(IY,IX)=0.
          GVPCO(IY,IX)=(GCVE(IY,IX)/GCVS(IY,IX))*(GCVH(IY,IX)+
     7                 GCVE(IY,IX)+GCVS(IY,IX))
          IVPVAL(IY,IX)=4
C  i.e. Phi = @SEH - corner coeff.
          GO TO 89060
89034     GCC(IY,IX)=0.
          GVECO(IY,IX)=((GCVE(IY,IX)-GCVH(IY,IX))/GCVS(IY,IX))*
     8                   (GCVH(IY,IX)+GCVE(IY,IX)+GCVS(IY,IX))
          IVEVAL(IY,IX)=4
C  i.e. Phi = @SE - edge coeff.
          GO TO 89060
89035     IF(GCVE(IY,IX).GE.GCVS(IY,IX).AND.GCVS(IY,IX).GE.0.0)
     9    GO TO 89036
          GCC(IY,IX)=0.
          GVPCO(IY,IX)=(GCVE(IY,IX)/GCVH(IY,IX))*(GCVH(IY,IX)+
     2                 GCVE(IY,IX)+GCVS(IY,IX))
          IVPVAL(IY,IX)=4
C  i.e. Phi = @SEH - corner coeff.
          GO TO 89060
89036     GCC(IY,IX)=0.
          GVECO(IY,IX)=((GCVE(IY,IX)-GCVS(IY,IX))/GCVH(IY,IX))*
     1                   (GCVH(IY,IX)+GCVE(IY,IX)+GCVS(IY,IX))
          IVEVAL(IY,IX)=7
C  i.e. Phi = @EH - edge coeff
          GO TO 89060
C ***********************************************************************
89037   IF(GCVH(IY,IX).GT.GCVL(IY,IX)) GO TO 89043
         IF(GCVL(IY,IX).GE.GCVN(IY,IX).AND.GCVL(IY,IX).GE.GCVE(IY,IX))
     1   GO TO 89041
         IF(GCVN(IY,IX).GE.GCVE(IY,IX).AND.GCVN(IY,IX).GE.GCVL(IY,IX))
     2   GO TO 89039
          IF(GCVN(IY,IX).GE.GCVL(IY,IX).AND.GCVL(IY,IX).GE.0.0)
     3    GO TO 89038
          GCC(IY,IX)=(1.0-(GCVL(IY,IX)/GCVE(IY,IX)))*(GCVL(IY,IX)+
     4               GCVE(IY,IX)+GCVN(IY,IX))
```

```
      GO TO 89060
89038 GCC(IY,IX)=(1.0-(GCVN(IY,IX)/GCVE(IY,IX)))*(GCVL(IY,IX)+
     5          GCVE(IY,IX)+GCVN(IY,IX))
      GO TO 89060
89039 IF(GCVE(IY,IX).GE.GCVL(IY,IX).AND.GCVL(IY,IX).GE.0.0)
     6 GO TO 89040
      GCC(IY,IX)=0.
      GVPCO(IY,IX)=(GCVE(IY,IX)/GCVN(IY,IX))*(GCVL(IY,IX)+
     7            GCVE(IY,IX)+GCVN(IY,IX))
      IVPVAL(IY,IX)=6
C  i.e. Phi = @NEL - corner coeff.
      GO TO 89060
89040 GCC(IY,IX)=0.
      GVECO(IY,IX)=((GCVE(IY,IX)-GCVL(IY,IX))/GCVN(IY,IX))*
     8             (GCVL(IY,IX)+GCVE(IY,IX)+GCVN(IY,IX))
      IVEVAL(IY,IX)=2
C  i.e. Phi = @NE - edge coeff.
      GO TO 89060
89041 IF(GCVE(IY,IX).GE.GCVN(IY,IX).AND.GCVN(IY,IX).GE.0.0)
     9 GO TO 89042
      GCC(IY,IX)=0.
      GVPCO(IY,IX)=(GCVE(IY,IX)/GCVL(IY,IX))*(GCVL(IY,IX)+
     2            GCVE(IY,IX)+GCVN(IY,IX))
      IVPVAL(IY,IX)=6
C  i.e. Phi = @NEL - corner coeff.
      GO TO 89060
89042 GCC(IY,IX)=0.
      GVECO(IY,IX)=((GCVE(IY,IX)-GCVN(IY,IX))/GCVL(IY,IX))*
     1             (GCVL(IY,IX)+GCVE(IY,IX)+GCVN(IY,IX))
      IVEVAL(IY,IX)=11
C  i.e. Phi = @EL - edge coeff
      GO TO 89060
C **********************************************************************
89043 IF(GCVH(IY,IX).GE.GCVN(IY,IX).AND.GCVH(IY,IX).GE.GCVE(IY,IX))
     1 GO TO 89047
      IF(GCVN(IY,IX).GE.GCVE(IY,IX).AND.GCVN(IY,IX).GE.GCVH(IY,IX))
     2 GO TO 89045
      IF(GCVN(IY,IX).GE.GCVH(IY,IX).AND.GCVH(IY,IX).GE.0.0)
     3 GO TO 89044
      GCC(IY,IX)=(1.0-(GCVH(IY,IX)/GCVE(IY,IX)))*(GCVH(IY,IX)+
     4          GCVE(IY,IX)+GCVN(IY,IX))
      GO TO 89060
89044 GCC(IY,IX)=(1.0-(GCVN(IY,IX)/GCVE(IY,IX)))*(GCVH(IY,IX)+
     5          GCVE(IY,IX)+GCVN(IY,IX))
      GO TO 89060
89045 IF(GCVE(IY,IX).GE.GCVH(IY,IX).AND.GCVH(IY,IX).GE.0.0)
     6 GO TO 89046
      GCC(IY,IX)=0.
      GVPCO(IY,IX)=(GCVE(IY,IX)/GCVN(IY,IX))*(GCVH(IY,IX)+
     7            GCVE(IY,IX)+GCVN(IY,IX))
      IVPVAL(IY,IX)=2
C  i.e. Phi = @NEH - corner coeff.
      GO TO 89060
89046 GCC(IY,IX)=0.
      GVECO(IY,IX)=((GCVE(IY,IX)-GCVH(IY,IX))/GCVN(IY,IX))*
     8             (GCVH(IY,IX)+GCVE(IY,IX)+GCVN(IY,IX))
      IVEVAL(IY,IX)=2
C  i.e. Phi = @NE - edge coeff.
      GO TO 89060
89047 IF(GCVE(IY,IX).GE.GCVN(IY,IX).AND.GCVN(IY,IX).GE.0.0)
     9 GO TO 89048
      GVPCO(IY,IX)=(GCVE(IY,IX)/GCVH(IY,IX))*(GCVH(IY,IX)+
     2            GCVE(IY,IX)+GCVN(IY,IX))
      IVPVAL(IY,IX)=2
C  i.e. Phi = @NEH - corner coeff.
      GO TO 89060
```

```
89048     GCC(IY,IX)=0.
          GVECO(IY,IX)=((GCVE(IY,IX)-GCVN(IY,IX))/GCVH(IY,IX))*
     1                 (GCVH(IY,IX)+GCVE(IY,IX)+GCVN(IY,IX))
          IVEVAL(IY,IX)=7
C  i.e. Phi = @EH - edge coeff.
89060 CONTINUE
      CALL SETYX(LD11,GCC,NYDIM,NXDIM)
C      CALL PRN('CCSE',LD11)
C
C    West Coefficient
C
      DO 89090 IX=1,NX
      DO 89090 IY=1,NY-1
       IF(GCVW(IY,IX).GT.0.0.AND.GCVE(IY,IX).EQ.0.0) GO TO 89065
89062    GCC(IY,IX)=GCVW(IY,IX)
         GO TO 89090
89065  IF(GCVN(IY,IX).GT.GCVS(IY,IX)) GO TO 89077
        IF(GCVH(IY,IX).GT.GCVL(IY,IX)) GO TO 89071
         IF(GCVL(IY,IX).GE.GCVW(IY,IX).AND.GCVL(IY,IX).GE.GCVS(IY,IX))
     1   GO TO 89069
         IF(GCVS(IY,IX).GE.GCVW(IY,IX).AND.GCVS(IY,IX).GE.GCVL(IY,IX))
     2   GO TO 89067
          IF(GCVS(IY,IX).GE.GCVL(IY,IX).AND.GCVL(IY,IX).GE.0.0)
     3    GO TO 89066
          GCC(IY,IX)=(1.0-(GCVL(IY,IX)/GCVW(IY,IX)))*(GCVL(IY,IX)+
     4              GCVW(IY,IX)+GCVS(IY,IX))
          GO TO 89090
89066     GCC(IY,IX)=(1.0-(GCVS(IY,IX)/GCVW(IY,IX)))*(GCVL(IY,IX)+
     5              GCVW(IY,IX)+GCVS(IY,IX))
          GO TO 89090
89067     IF(GCVW(IY,IX).GE.GCVL(IY,IX).AND.GCVL(IY,IX).GE.0.0)
     6    GO TO 89068
          GCC(IY,IX)=0.
          GVPCO(IY,IX)=(GCVW(IY,IX)/GCVS(IY,IX))*(GCVL(IY,IX)+
     7                GCVW(IY,IX)+GCVS(IY,IX))
          IVPVAL(IY,IX)=7
C  i.e. Phi = @SWL - corner coeff.
          GO TO 89090
89068     GCC(IY,IX)=0.
          GVECO(IY,IX)=((GCVW(IY,IX)-GCVL(IY,IX))/GCVS(IY,IX))*
     8                 (GCVL(IY,IX)+GCVW(IY,IX)+GCVS(IY,IX))
          IVEVAL(IY,IX)=3
C  i.e. Phi = @SW - edge coeff.
          GO TO 89090
89069     IF(GCVW(IY,IX).GE.GCVS(IY,IX).AND.GCVS(IY,IX).GE.0.0)
     9    GO TO 89070
          GCC(IY,IX)=0.
          GVPCO(IY,IX)=(GCVW(IY,IX)/GCVL(IY,IX))*(GCVL(IY,IX)+
     2                GCVW(IY,IX)+GCVS(IY,IX))
          IVPVAL(IY,IX)=7
C  i.e. Phi = @SWL - corner coeff.
          GO TO 89090
89070     GCC(IY,IX)=0.
          GVECO(IY,IX)=((GCVW(IY,IX)-GCVS(IY,IX))/GCVL(IY,IX))*
     1                 (GCVL(IY,IX)+GCVW(IY,IX)+GCVS(IY,IX))
          IVEVAL(IY,IX)=12
C  i.e. Phi = @WL - edge coeff
          GO TO 89090
C ********************************************************************
89071    IF(GCVH(IY,IX).GE.GCVW(IY,IX).AND.GCVH(IY,IX).GE.GCVS(IY,IX))
     1   GO TO 89075
         IF(GCVS(IY,IX).GE.GCVW(IY,IX).AND.GCVS(IY,IX).GE.GCVH(IY,IX))
     2   GO TO 89073
          IF(GCVS(IY,IX).GE.GCVH(IY,IX).AND.GCVH(IY,IX).GE.0.0)
     3    GO TO 89072
          GCC(IY,IX)=(1.0-(GCVH(IY,IX)/GCVW(IY,IX)))*(GCVH(IY,IX)+
```

```
     4            GCVW(IY,IX)+GCVS(IY,IX))
          GO TO 89090
89072     GCC(IY,IX)=(1.0-(GCVS(IY,IX)/GCVW(IY,IX)))*(GCVH(IY,IX)+
     5            GCVW(IY,IX)+GCVS(IY,IX))
          GO TO 89090
89073     IF(GCVW(IY,IX).GE.GCVH(IY,IX).AND.GCVH(IY,IX).GE.0.0)
     6    GO TO 89074
          GCC(IY,IX)=0.
          GVPCO(IY,IX)=(GCVW(IY,IX)/GCVS(IY,IX))*(GCVH(IY,IX)+
     7              GCVW(IY,IX)+GCVS(IY,IX))
          IVPVAL(IY,IX)=3
C  i.e. Phi = @SWH - corner coeff.
          GO TO 89090
89074     GCC(IY,IX)=0.
          GVECO(IY,IX)=((GCVW(IY,IX)-GCVH(IY,IX))/GCVS(IY,IX))*
     8               (GCVH(IY,IX)+GCVW(IY,IX)+GCVS(IY,IX))
          IVEVAL(IY,IX)=3
C  i.e. Phi = @SW - edge coeff.
          GO TO 89090
89075     IF(GCVW(IY,IX).GE.GCVS(IY,IX).AND.GCVS(IY,IX).GE.0.0)
     9    GO TO 89076
          GCC(IY,IX)=0.
          GVPCO(IY,IX)=(GCVW(IY,IX)/GCVH(IY,IX))*(GCVH(IY,IX)+
     2              GCVW(IY,IX)+GCVS(IY,IX))
          IVPVAL(IY,IX)=3
C  i.e. Phi = @SWH - corner coeff.
          GO TO 89090
89076     GCC(IY,IX)=0.
          GVECO(IY,IX)=((GCVW(IY,IX)-GCVS(IY,IX))/GCVH(IY,IX))*
     1               (GCVH(IY,IX)+GCVW(IY,IX)+GCVS(IY,IX))
          IVEVAL(IY,IX)=8
C  i.e. Phi = @WH - edge coeff
          GO TO 89090
C ************************************************************************
89077   IF(GCVH(IY,IX).GT.GCVL(IY,IX)) GO TO 89083
         IF(GCVL(IY,IX).GE.GCVN(IY,IX).AND.GCVL(IY,IX).GE.GCVW(IY,IX))
     1   GO TO 89081
         IF(GCVN(IY,IX).GE.GCVW(IY,IX).AND.GCVN(IY,IX).GE.GCVL(IY,IX))
     2   GO TO 89079
          IF(GCVN(IY,IX).GE.GCVL(IY,IX).AND.GCVL(IY,IX).GE.0.0)
     3    GO TO 89078
          GCC(IY,IX)=(1.0-(GCVL(IY,IX)/GCVW(IY,IX)))*(GCVL(IY,IX)+
     4            GCVW(IY,IX)+GCVN(IY,IX))
          GO TO 89090
89078     GCC(IY,IX)=(1.0-(GCVN(IY,IX)/GCVW(IY,IX)))*(GCVL(IY,IX)+
     5            GCVW(IY,IX)+GCVN(IY,IX))
          GO TO 89090
89079     IF(GCVW(IY,IX).GE.GCVL(IY,IX).AND.GCVL(IY,IX).GE.0.0)
     6    GO TO 89080
          GCC(IY,IX)=0.
          GVPCO(IY,IX)=(GCVW(IY,IX)/GCVN(IY,IX))*(GCVL(IY,IX)+
     7              GCVW(IY,IX)+GCVN(IY,IX))
          IVPVAL(IY,IX)=5
C  i.e. Phi = @NWL - corner coeff.
          GO TO 89090
89080     GCC(IY,IX)=0.
          GVECO(IY,IX)=((GCVW(IY,IX)-GCVL(IY,IX))/GCVN(IY,IX))*
     8               (GCVL(IY,IX)+GCVW(IY,IX)+GCVN(IY,IX))
          IVEVAL(IY,IX)=1
C  i.e. Phi = @NW - edge coeff.
          GO TO 89090
89081     IF(GCVW(IY,IX).GE.GCVN(IY,IX).AND.GCVN(IY,IX).GE.0.0)
     9    GO TO 89082
          GCC(IY,IX)=0.
          GVPCO(IY,IX)=(GCVW(IY,IX)/GCVL(IY,IX))*(GCVL(IY,IX)+
     2              GCVW(IY,IX)+GCVN(IY,IX))
```

```
          IVPVAL(IY,IX)=5
C  i.e. Phi = @NWL - corner coeff.
          GO TO 89090
89082     GCC(IY,IX)=0.
          GVECO(IY,IX)=((GCVW(IY,IX)-GCVN(IY,IX))/GCVL(IY,IX))*
     1                  (GCVL(IY,IX)+GCVW(IY,IX)+GCVN(IY,IX))
          IVEVAL(IY,IX)=12
C  i.e. Phi = @WL - edge coeff
          GO TO 89090
C ***********************************************************************
89083    IF(GCVH(IY,IX).GE.GCVN(IY,IX).AND.GCVH(IY,IX).GE.GCVW(IY,IX))
     1   GO TO 89087
         IF(GCVN(IY,IX).GE.GCVW(IY,IX).AND.GCVN(IY,IX).GE.GCVH(IY,IX))
     2   GO TO 89085
          IF(GCVN(IY,IX).GE.GCVH(IY,IX).AND.GCVH(IY,IX).GE.0.0)
     3    GO TO 89084
          GCC(IY,IX)=(1.0-(GCVH(IY,IX)/GCVW(IY,IX)))*(GCVH(IY,IX)+
     4              GCVW(IY,IX)+GCVN(IY,IX))
          GO TO 89090
89084     GCC(IY,IX)=(1.0-(GCVN(IY,IX)/GCVW(IY,IX)))*(GCVH(IY,IX)+
     5              GCVW(IY,IX)+GCVN(IY,IX))
          GO TO 89090
89085     IF(GCVW(IY,IX).GE.GCVH(IY,IX).AND.GCVH(IY,IX).GE.0.0)
     6    GO TO 89086
          GCC(IY,IX)=0.
          GVPCO(IY,IX)=(GCVW(IY,IX)/GCVN(IY,IX))*(GCVH(IY,IX)+
     7                GCVW(IY,IX)+GCVN(IY,IX))
          IVPVAL(IY,IX)=1
C  i.e. Phi = @NWH - corner coeff.
          GO TO 89090
89086     GCC(IY,IX)=0.
          GVECO(IY,IX)=((GCVW(IY,IX)-GCVH(IY,IX))/GCVN(IY,IX))*
     8                  (GCVH(IY,IX)+GCVW(IY,IX)+GCVN(IY,IX))
          IVEVAL(IY,IX)=1
C  i.e. Phi = @NW - edge coeff.
          GO TO 89090
89087     IF(GCVW(IY,IX).GE.GCVN(IY,IX).AND.GCVN(IY,IX).GE.0.0)
     9    GO TO 89088
          GCC(IY,IX)=0.
          GVPCO(IY,IX)=(GCVW(IY,IX)/GCVH(IY,IX))*(GCVH(IY,IX)+
     2                GCVW(IY,IX)+GCVN(IY,IX))
          IVPVAL(IY,IX)=1
C  i.e. Phi = @NWH - corner coeff.
          GO TO 89090
89088     GCC(IY,IX)=0.
          GVECO(IY,IX)=((GCVW(IY,IX)-GCVN(IY,IX))/GCVH(IY,IX))*
     1                  (GCVH(IY,IX)+GCVW(IY,IX)+GCVN(IY,IX))
          IVEVAL(IY,IX)=8
C  i.e. Phi = @WH - edge coeff.
89090 CONTINUE
      CALL SETYX(LD12,GCC,NYDIM,NXDIM)
C      CALL PRN('CCSW',LD12)
      RETURN
C
C    High-Low
C
89100 IF(NDIREC.NE.5) GO TO 89140
C   i.e. consider flow through High face.
         CALL GETYX(LD2,GCVH,NYDIM,NXDIM)
         IF(ISWEEP.EQ.FSWEEP) GO TO 89140
C    High Coefficient
C
      DO 89130 IX=1,NX
      DO 89130 IY=1,NY-1
       IF(GCVH(IY,IX).GT.0.0.AND.GCVL(IY,IX).EQ.0.0) GO TO 89105
         GCC(IY,IX)=GCVH(IY,IX)
```

```
         GO TO 89130
89105  IF(GCVN(IY,IX).GT.GCVS(IY,IX)) GO TO 89117
        IF(GCVE(IY,IX).GT.GCVW(IY,IX)) GO TO 89111
         IF(GCVS(IY,IX).GE.GCVH(IY,IX).AND.GCVS(IY,IX).GE.GCVW(IY,IX))
     1   GO TO 89109
         IF(GCVW(IY,IX).GE.GCVH(IY,IX).AND.GCVW(IY,IX).GE.GCVS(IY,IX))
     2   GO TO 89107
          IF(GCVW(IY,IX).GE.GCVS(IY,IX).AND.GCVS(IY,IX).GE.0.0)
     3    GO TO 89106
          GCC(IY,IX)=(1.0-(GCVS(IY,IX)/GCVH(IY,IX)))*(GCVH(IY,IX)+
     4             GCVW(IY,IX)+GCVS(IY,IX))
          GO TO 89130
89106     GCC(IY,IX)=(1.0-(GCVW(IY,IX)/GCVH(IY,IX)))*(GCVH(IY,IX)+
     5             GCVW(IY,IX)+GCVS(IY,IX))
          GO TO 89130
89107     IF(GCVS(IY,IX).GE.GCVH(IY,IX).AND.GCVH(IY,IX).GE.0.0)
     9    GO TO 89108
          GCC(IY,IX)=0.
          GVECO(IY,IX)=((GCVH(IY,IX)-GCVS(IY,IX))/GCVW(IY,IX))*
     1                 (GCVH(IY,IX)+GCVW(IY,IX)+GCVS(IY,IX))
          IVEVAL(IY,IX)=8
C  i.e. Phi = @WH - edge coeff
          GO TO 89130
89108     GCC(IY,IX)=0.
          GVPCO(IY,IX)=(GCVH(IY,IX)/GCVW(IY,IX))*(GCVH(IY,IX)+
     2                GCVW(IY,IX)+GCVS(IY,IX))
          IVPVAL(IY,IX)=3
C  i.e. Phi = @SWH - corner coeff.
          GO TO 89130
89109     IF(GCVW(IY,IX).GE.GCVH(IY,IX).AND.GCVH(IY,IX).GE.0.0)
     9    GO TO 89110
          GCC(IY,IX)=0.
          GVECO(IY,IX)=((GCVH(IY,IX)-GCVW(IY,IX))/GCVS(IY,IX))*
     1                 (GCVH(IY,IX)+GCVW(IY,IX)+GCVS(IY,IX))
          IVEVAL(IY,IX)=6
C  i.e. Phi = @SH - edge coeff
          GO TO 89130
89110     GCC(IY,IX)=0.
          GVPCO(IY,IX)=(GCVH(IY,IX)/GCVS(IY,IX))*(GCVH(IY,IX)+
     2                GCVW(IY,IX)+GCVS(IY,IX))
          IVPVAL(IY,IX)=3
C  i.e. Phi = @SWH - corner coeff.
          GO TO 89130
C ***********************************************************************
89111    IF(GCVS(IY,IX).GE.GCVH(IY,IX).AND.GCVS(IY,IX).GE.GCVE(IY,IX))
     1   GO TO 89115
         IF(GCVE(IY,IX).GE.GCVH(IY,IX).AND.GCVE(IY,IX).GE.GCVS(IY,IX))
     2   GO TO 89113
          IF(GCVE(IY,IX).GE.GCVS(IY,IX).AND.GCVS(IY,IX).GE.0.0)
     3    GO TO 89112
          GCC(IY,IX)=(1.0-(GCVS(IY,IX)/GCVH(IY,IX)))*(GCVH(IY,IX)+
     4             GCVE(IY,IX)+GCVS(IY,IX))
          GO TO 89130
89112     GCC(IY,IX)=(1.0-(GCVE(IY,IX)/GCVH(IY,IX)))*(GCVH(IY,IX)+
     5             GCVE(IY,IX)+GCVS(IY,IX))
          GO TO 89130
89113     IF(GCVS(IY,IX).GE.GCVH(IY,IX).AND.GCVH(IY,IX).GE.0.0)
     9    GO TO 89114
          GCC(IY,IX)=0.
          GVECO(IY,IX)=((GCVH(IY,IX)-GCVS(IY,IX))/GCVE(IY,IX))*
     1                 (GCVH(IY,IX)+GCVE(IY,IX)+GCVS(IY,IX))
          IVEVAL(IY,IX)=7
C  i.e. Phi = @EH - edge coeff
          GO TO 89130
89114     GCC(IY,IX)=0.
          GVPCO(IY,IX)=(GCVH(IY,IX)/GCVE(IY,IX))*(GCVH(IY,IX)+
```

```
     2              GCVE(IY,IX)+GCVS(IY,IX))
          IVPVAL(IY,IX)=4
C  i.e. Phi = @SEH - corner coeff.
          GO TO 89130
89115     IF(GCVE(IY,IX).GE.GCVH(IY,IX).AND.GCVH(IY,IX).GE.0.0)
     9    GO TO 89116
          GCC(IY,IX)=0.
          GVECO(IY,IX)=((GCVH(IY,IX)-GCVE(IY,IX))/GCVS(IY,IX))*
     1                 (GCVH(IY,IX)+GCVE(IY,IX)+GCVS(IY,IX))
          IVEVAL(IY,IX)=6
C  i.e. Phi = @SH - edge coeff
          GO TO 89130
89116     GCC(IY,IX)=0.
          GVPCO(IY,IX)=(GCVH(IY,IX)/GCVS(IY,IX))*(GCVH(IY,IX)+
     2              GCVE(IY,IX)+GCVS(IY,IX))
          IVPVAL(IY,IX)=4
C  i.e. Phi = @SEH - corner coeff.
          GO TO 89130
C **********************************************************************
89117   IF(GCVE(IY,IX).GT.GCVW(IY,IX)) GO TO 89123
         IF(GCVN(IY,IX).GE.GCVH(IY,IX).AND.GCVN(IY,IX).GE.GCVW(IY,IX))
     1   GO TO 89121
         IF(GCVW(IY,IX).GE.GCVH(IY,IX).AND.GCVW(IY,IX).GE.GCVN(IY,IX))
     2   GO TO 89119
          IF(GCVW(IY,IX).GE.GCVN(IY,IX).AND.GCVN(IY,IX).GE.0.0)
     3    GO TO 89118
          GCC(IY,IX)=(1.0-(GCVN(IY,IX)/GCVH(IY,IX)))*(GCVH(IY,IX)+
     4            GCVW(IY,IX)+GCVN(IY,IX))
          GO TO 89130
89118     GCC(IY,IX)=(1.0-(GCVW(IY,IX)/GCVH(IY,IX)))*(GCVH(IY,IX)+
     5            GCVW(IY,IX)+GCVN(IY,IX))
          GO TO 89130
89119     IF(GCVN(IY,IX).GE.GCVH(IY,IX).AND.GCVH(IY,IX).GE.0.0)
     9    GO TO 89120
          GCC(IY,IX)=0.
          GVECO(IY,IX)=((GCVH(IY,IX)-GCVN(IY,IX))/GCVW(IY,IX))*
     1                 (GCVH(IY,IX)+GCVW(IY,IX)+GCVN(IY,IX))
          IVEVAL(IY,IX)=8
C  i.e. Phi = @WH - edge coeff
          GO TO 89130
89120     GCC(IY,IX)=0.
          GVPCO(IY,IX)=(GCVH(IY,IX)/GCVW(IY,IX))*(GCVH(IY,IX)+
     2              GCVW(IY,IX)+GCVN(IY,IX))
          IVPVAL(IY,IX)=1
C  i.e. Phi = @NWH - corner coeff.
          GO TO 89130
89121     IF(GCVW(IY,IX).GE.GCVH(IY,IX).AND.GCVH(IY,IX).GE.0.0)
     9    GO TO 89122
          GCC(IY,IX)=0.
          GVECO(IY,IX)=((GCVH(IY,IX)-GCVW(IY,IX))/GCVN(IY,IX))*
     1                 (GCVH(IY,IX)+GCVW(IY,IX)+GCVN(IY,IX))
          IVEVAL(IY,IX)=5
C  i.e. Phi = @NH - edge coeff
          GO TO 89130
89122     GCC(IY,IX)=0.
          GVPCO(IY,IX)=(GCVW(IY,IX)/GCVN(IY,IX))*(GCVH(IY,IX)+
     2              GCVW(IY,IX)+GCVN(IY,IX))
          IVPVAL(IY,IX)=1
C  i.e. Phi = @NWH - corner coeff.
          GO TO 89130
C ***********************************************************************
89123    IF(GCVN(IY,IX).GE.GCVH(IY,IX).AND.GCVN(IY,IX).GE.GCVE(IY,IX))
     1   GO TO 89127
         IF(GCVE(IY,IX).GE.GCVH(IY,IX).AND.GCVE(IY,IX).GE.GCVN(IY,IX))
     2   GO TO 89125
          IF(GCVE(IY,IX).GE.GCVN(IY,IX).AND.GCVN(IY,IX).GE.0.0)
```

```
     3    GO TO 89124
          GCC(IY,IX)=(1.0-(GCVN(IY,IX)/GCVH(IY,IX)))*(GCVH(IY,IX)+
     4              GCVE(IY,IX)+GCVN(IY,IX))
          GO TO 89130
89124     GCC(IY,IX)=(1.0-(GCVE(IY,IX)/GCVH(IY,IX)))*(GCVH(IY,IX)+
     5              GCVE(IY,IX)+GCVN(IY,IX))
          GO TO 89130
89125     IF(GCVN(IY,IX).GE.GCVH(IY,IX).AND.GCVH(IY,IX).GE.0.0)
     9    GO TO 89126
          GCC(IY,IX)=0.
          GVECO(IY,IX)=((GCVH(IY,IX)-GCVN(IY,IX))/GCVE(IY,IX))*
     1                 (GCVH(IY,IX)+GCVE(IY,IX)+GCVN(IY,IX))
          IVEVAL(IY,IX)=7
C  i.e. Phi = @EH - edge coeff
          GO TO 89130
89126     GCC(IY,IX)=0.
          GVPCO(IY,IX)=(GCVH(IY,IX)/GCVE(IY,IX))*(GCVH(IY,IX)+
     2                GCVE(IY,IX)+GCVN(IY,IX))
          IVPVAL(IY,IX)=2
C  i.e. Phi = @NEH - corner coeff.
          GO TO 89130
89127     IF(GCVE(IY,IX).GE.GCVH(IY,IX).AND.GCVH(IY,IX).GE.0.0)
     9    GO TO 89128
          GCC(IY,IX)=0.
          GVECO(IY,IX)=((GCVH(IY,IX)-GCVE(IY,IX))/GCVN(IY,IX))*
     1                 (GCVH(IY,IX)+GCVE(IY,IX)+GCVN(IY,IX))
          IVEVAL(IY,IX)=5
C  i.e. Phi = @NH - edge coeff
          GO TO 89130
89128     GCC(IY,IX)=0.
          GVPCO(IY,IX)=(GCVH(IY,IX)/GCVN(IY,IX))*(GCVH(IY,IX)+
     2                GCVE(IY,IX)+GCVN(IY,IX))
          IVPVAL(IY,IX)=2
C  i.e. Phi = @NEH - corner coeff.
89130 CONTINUE
      CALL SETYX(LD2,GCC,NYDIM,NXDIM)
C      CALL PRN('CCSH',LD2)
      RETURN
C
C   Low Coefficient
C
89140 IF(NDIREC.NE.6) RETURN
C   i.e. consider flow through Low face.
         CALL GETYX(LD2,GCVL,NYDIM,NXDIM)
         IF(ISWEEP.EQ.FSWEEP) RETURN
      DO 89200 IX=1,NX
      DO 89200 IY=1,NY-1
       IF(GCVL(IY,IX).GT.0.0.AND.GCVH(IY,IX).EQ.0.0) GO TO 89145
         GCC(IY,IX)=GCVL(IY,IX)
         GO TO 89200
89145  IF(GCVN(IY,IX).GT.GCVS(IY,IX)) GO TO 89157
        IF(GCVE(IY,IX).GT.GCVW(IY,IX)) GO TO 89151
         IF(GCVS(IY,IX).GE.GCVL(IY,IX).AND.GCVS(IY,IX).GE.GCVW(IY,IX))
     1   GO TO 89149
         IF(GCVW(IY,IX).GE.GCVL(IY,IX).AND.GCVW(IY,IX).GE.GCVS(IY,IX))
     2   GO TO 89147
          IF(GCVW(IY,IX).GE.GCVS(IY,IX).AND.GCVS(IY,IX).GE.0.0)
     3    GO TO 89146
          GCC(IY,IX)=(1.0-(GCVS(IY,IX)/GCVL(IY,IX)))*(GCVL(IY,IX)+
     4              GCVW(IY,IX)+GCVS(IY,IX))
          GO TO 89200
89146     GCC(IY,IX)=(1.0-(GCVW(IY,IX)/GCVL(IY,IX)))*(GCVL(IY,IX)+
     5              GCVW(IY,IX)+GCVS(IY,IX))
          GO TO 89200
89147     IF(GCVS(IY,IX).GE.GCVL(IY,IX).AND.GCVL(IY,IX).GE.0.0)
     9    GO TO 89148
```

```
            GCC(IY,IX)=0.
            GVECO(IY,IX)=((GCVL(IY,IX)-GCVS(IY,IX))/GCVW(IY,IX))*
     1                 (GCVL(IY,IX)+GCVW(IY,IX)+GCVS(IY,IX))
            IVEVAL(IY,IX)=12
C  i.e. Phi = @WL - edge coeff
            GO TO 89200
89148       GCC(IY,IX)=0.
            GVPCO(IY,IX)=(GCVL(IY,IX)/GCVW(IY,IX))*(GCVL(IY,IX)+
     2                GCVW(IY,IX)+GCVS(IY,IX))
            IVPVAL(IY,IX)=7
C  i.e. Phi = @SWL - corner coeff.
            GO TO 89200
89149       IF(GCVW(IY,IX).GE.GCVL(IY,IX).AND.GCVL(IY,IX).GE.0.0)
     9      GO TO 89150
            GCC(IY,IX)=0.
            GVECO(IY,IX)=((GCVL(IY,IX)-GCVW(IY,IX))/GCVS(IY,IX))*
     1                 (GCVL(IY,IX)+GCVW(IY,IX)+GCVS(IY,IX))
            IVEVAL(IY,IX)=10
C  i.e. Phi = @SL - edge coeff
            GO TO 89200
89150       GCC(IY,IX)=0.
            GVPCO(IY,IX)=(GCVL(IY,IX)/GCVS(IY,IX))*(GCVL(IY,IX)+
     2                GCVW(IY,IX)+GCVS(IY,IX))
            IVPVAL(IY,IX)=7
C  i.e. Phi = @SWL - corner coeff.
            GO TO 89200
C ***********************************************************************
89151      IF(GCVS(IY,IX).GE.GCVL(IY,IX).AND.GCVS(IY,IX).GE.GCVE(IY,IX))
     1     GO TO 89155
           IF(GCVE(IY,IX).GE.GCVL(IY,IX).AND.GCVE(IY,IX).GE.GCVS(IY,IX))
     2     GO TO 89153
            IF(GCVE(IY,IX).GE.GCVS(IY,IX).AND.GCVS(IY,IX).GE.0.0)
     3      GO TO 89152
            GCC(IY,IX)=(1.0-(GCVS(IY,IX)/GCVL(IY,IX)))*(GCVL(IY,IX)+
     4               GCVE(IY,IX)+GCVS(IY,IX))
            GO TO 89200
89152       GCC(IY,IX)=(1.0-(GCVE(IY,IX)/GCVL(IY,IX)))*(GCVL(IY,IX)+
     5               GCVE(IY,IX)+GCVS(IY,IX))
            GO TO 89200
89153       IF(GCVS(IY,IX).GE.GCVL(IY,IX).AND.GCVL(IY,IX).GE.0.0)
     9      GO TO 89154
            GCC(IY,IX)=0.
            GVECO(IY,IX)=((GCVL(IY,IX)-GCVS(IY,IX))/GCVE(IY,IX))*
     1                 (GCVL(IY,IX)+GCVE(IY,IX)+GCVS(IY,IX))
            IVEVAL(IY,IX)=11
C  i.e. Phi = @EL - edge coeff
            GO TO 89200
89154       GCC(IY,IX)=0.
            GVPCO(IY,IX)=(GCVL(IY,IX)/GCVE(IY,IX))*(GCVL(IY,IX)+
     2                GCVE(IY,IX)+GCVS(IY,IX))
            IVPVAL(IY,IX)=8
C  i.e. Phi = @SEL - corner coeff.
            GO TO 89200
89155       IF(GCVE(IY,IX).GE.GCVL(IY,IX).AND.GCVL(IY,IX).GE.0.0)
     9      GO TO 89156
            GCC(IY,IX)=0.
            GVECO(IY,IX)=((GCVL(IY,IX)-GCVE(IY,IX))/GCVS(IY,IX))*
     1                 (GCVL(IY,IX)+GCVE(IY,IX)+GCVS(IY,IX))
            IVEVAL(IY,IX)=10
C  i.e. Phi = @SL - edge coeff
            GO TO 89200
89156       GCC(IY,IX)=0.
            GVPCO(IY,IX)=(GCVL(IY,IX)/GCVS(IY,IX))*(GCVL(IY,IX)+
     2                GCVE(IY,IX)+GCVS(IY,IX))
            IVPVAL(IY,IX)=8
C  i.e. Phi = @SEL - corner coeff.
```

```
      GO TO 89200
C **********************************************************************
89157 IF(GCVE(IY,IX).GT.GCVW(IY,IX)) GO TO 89163
       IF(GCVN(IY,IX).GE.GCVL(IY,IX).AND.GCVN(IY,IX).GE.GCVW(IY,IX))
     1 GO TO 89161
       IF(GCVW(IY,IX).GE.GCVL(IY,IX).AND.GCVW(IY,IX).GE.GCVN(IY,IX))
     2 GO TO 89159
        IF(GCVW(IY,IX).GE.GCVN(IY,IX).AND.GCVN(IY,IX).GE.0.0)
     3  GO TO 89158
        GCC(IY,IX)=(1.0-(GCVN(IY,IX)/GCVL(IY,IX)))*(GCVL(IY,IX)+
     4            GCVW(IY,IX)+GCVN(IY,IX))
        GO TO 89200
89158   GCC(IY,IX)=(1.0-(GCVW(IY,IX)/GCVL(IY,IX)))*(GCVL(IY,IX)+
     5            GCVW(IY,IX)+GCVN(IY,IX))
        GO TO 89200
89159   IF(GCVN(IY,IX).GE.GCVL(IY,IX).AND.GCVL(IY,IX).GE.0.0)
     9  GO TO 89160
        GCC(IY,IX)=0.
        GVECO(IY,IX)=((GCVL(IY,IX)-GCVN(IY,IX))/GCVW(IY,IX))*
     1                (GCVL(IY,IX)+GCVW(IY,IX)+GCVN(IY,IX))
        IVEVAL(IY,IX)=12
C  i.e. Phi = @WL - edge coeff
        GO TO 89200
89160   GCC(IY,IX)=0.
        GVPCO(IY,IX)=(GCVL(IY,IX)/GCVW(IY,IX))*(GCVL(IY,IX)+
     2              GCVW(IY,IX)+GCVN(IY,IX))
        IVPVAL(IY,IX)=5
C  i.e. Phi = @NWL - corner coeff.
        GO TO 89200
89161   IF(GCVW(IY,IX).GE.GCVL(IY,IX).AND.GCVL(IY,IX).GE.0.0)
     9  GO TO 89162
        GCC(IY,IX)=0.
        GVECO(IY,IX)=((GCVL(IY,IX)-GCVW(IY,IX))/GCVN(IY,IX))*
     1                (GCVL(IY,IX)+GCVW(IY,IX)+GCVN(IY,IX))
        IVEVAL(IY,IX)=9
C  i.e. Phi = @NL - edge coeff
        GO TO 89200
89162   GCC(IY,IX)=0.
        GVPCO(IY,IX)=(GCVL(IY,IX)/GCVN(IY,IX))*(GCVL(IY,IX)+
     2              GCVW(IY,IX)+GCVN(IY,IX))
        IVPVAL(IY,IX)=5
C  i.e. Phi = @NWL - corner coeff.
        GO TO 89200
C ***********************************************************************
89163  IF(GCVN(IY,IX).GE.GCVL(IY,IX).AND.GCVN(IY,IX).GE.GCVE(IY,IX))
     1 GO TO 89167
       IF(GCVE(IY,IX).GE.GCVL(IY,IX).AND.GCVE(IY,IX).GE.GCVN(IY,IX))
     2 GO TO 89165
        IF(GCVE(IY,IX).GE.GCVN(IY,IX).AND.GCVN(IY,IX).GE.0.0)
     3  GO TO 89164
        GCC(IY,IX)=(1.0-(GCVN(IY,IX)/GCVL(IY,IX)))*(GCVL(IY,IX)+
     4            GCVE(IY,IX)+GCVN(IY,IX))
        GO TO 89200
89164   GCC(IY,IX)=(1.0-(GCVE(IY,IX)/GCVL(IY,IX)))*(GCVL(IY,IX)+
     5            GCVE(IY,IX)+GCVN(IY,IX))
        GO TO 89200
89165   IF(GCVN(IY,IX).GE.GCVL(IY,IX).AND.GCVL(IY,IX).GE.0.0)
     9  GO TO 89166
        GCC(IY,IX)=0.
        GVECO(IY,IX)=((GCVL(IY,IX)-GCVN(IY,IX))/GCVE(IY,IX))*
     1                (GCVL(IY,IX)+GCVE(IY,IX)+GCVN(IY,IX))
        IVEVAL(IY,IX)=11
C  i.e. Phi = @EL - edge coeff
        GO TO 89200
89166   GCC(IY,IX)=0.
        GVPCO(IY,IX)=(GCVL(IY,IX)/GCVE(IY,IX))*(GCVL(IY,IX)+
```

```
     2                GCVE(IY,IX)+GCVN(IY,IX))
          IVPVAL(IY,IX)=6
C  i.e. Phi = @NEL - corner coeff.
          GO TO 89200
89167     IF(GCVE(IY,IX).GE.GCVL(IY,IX).AND.GCVL(IY,IX).GE.0.0)
     9    GO TO 89168
          GCC(IY,IX)=0.
          GVECO(IY,IX)=((GCVL(IY,IX)-GCVE(IY,IX))/GCVN(IY,IX))*
     1                  (GCVL(IY,IX)+GCVE(IY,IX)+GCVN(IY,IX))
          IVEVAL(IY,IX)=9
C  i.e. Phi = @NL - edge coeff
          GO TO 89200
89168     GCC(IY,IX)=0.
          GVPCO(IY,IX)=(GCVL(IY,IX)/GCVN(IY,IX))*(GCVL(IY,IX)+
     2                GCVE(IY,IX)+GCVN(IY,IX))
          IVPVAL(IY,IX)=6
C  i.e. Phi = @NEL - corner coeff.
89200 CONTINUE
      CALL SETYX(LD2,GCC,NYDIM,NXDIM)
C      CALL PRN('CCSL',LD2)
      RETURN
      END
```

```
C****************************************************************
      SUBROUTINE GROUND3
      INCLUDE 'lp16/d_earth/SATEAR'
      INCLUDE 'lp16/d_earth/GRDLOC'
      INCLUDE 'lp16/d_earth/GRDEAR'
      INCLUDE 'lp16/d_earth/GRDBFC'
CXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX USER SECTION STARTS:
C
C 1   Set dimensions of data-for-GROUND arrays here. WARNING: the
C     corresponding arrays in the MAIN program of the satellite
C     and EARTH must have the same dimensions.
      PARAMETER (NLG=20, NIG=20, NRG=100, NCG=10)
C
      COMMON/LGRND/LG(NLG)/IGRND/IG(NIG)/RGRND/RG(NRG)/CGRND/CG(NCG)
      LOGICAL LG
      CHARACTER*4 CG
      COMMON/GUGRND/GCSN(60,80),GCSS(60,80),GCSE(60,80),GCSW(60,80),
     7              GCSL(60,80),GCSH(60,80),GSPCO(60,80),GSECO(60,80),
     8              GCUN(60,80),GCUS(60,80),GCUE(60,80),GCUW(60,80),
     7              GCUL(60,80),GCUH(60,80),GUPCO(60,80),GUECO(60,80),
     8              GCVN(60,80),GCVS(60,80),GCVE(60,80),GCVW(60,80),
     7              GCVL(60,80),GCVH(60,80),GVPCO(60,80),GVECO(60,80),
     8              GCWN(60,80),GCWS(60,80),GCWE(60,80),GCWW(60,80),
     7              GCWL(60,80),GCWH(60,80),GWPCO(60,80),GWECO(60,80)
      COMMON/IUGRND/ISPVAL(60,80),ISEVAL(60,80),IUPVAL(60,80),
     7              IUEVAL(60,80),IVEVAL(60,80),IVPVAL(60,80),
     8              IWEVAL(60,80),IWPVAL(60,80)
C
C 2   User dimensions own arrays here, for example:
C     DIMENSION GUH(10,10),GUC(10,10),GUX(10,10),GUZ(10)
      DIMENSION GCC(60,80),GCMH(60,80),GCML(60,80),
     1          GCE(60,80),GEMP(60,80),GEMH(60,80),GEML(60,80)
C
C 3   User places his data statements here, for example:
      DATA NXDIM,NYDIM/80,60/
C
C 4   Insert own coding below as desired, guided by GREX examples.
C     Note that the satellite-to-GREX special data in the labelled
C     COMMONs /RSG/, /ISG/, /LSG/ and /CSG/ can be included and
C     used below but the user must check GREX for any conflicting
C     uses. The same comment applies to the EARTH-spare working
C     arrays EASP1, EASP2,....EASP20. In addition to the EASPs,
C     there are 10 GRound-earth SPare arrays, GRSP1,...,GRSP10,
C     supplied solely for the user, which are not used by GREX. If
C     the call to GREX has been deactivated then all of the arrays
C     may be used without reservation.
C
      IF(.NOT.(LG(3))) RETURN
      IF(INDVAR.EQ.7) GO TO 88949
      RETURN
88949 CONTINUE
C
C  Corrected Coefficients For W-Equation
C  **************************************
C
C  North-South
C
      IF(NDIREC.NE.1) GO TO 89020
C  i.e. consider flow through North and South faces.
         CALL GETYX(LD11,GCWN,NYDIM,NXDIM)
         CALL GETYX(LD12,GCWS,NYDIM,NXDIM)
         IF(ISWEEP.EQ.FSWEEP) GO TO 89020
C
C
C  North Coefficient
      DO 88980 IX=1,NX
```

```
         DO 88980 IY=1,NY
          IF(GCWN(IY,IX).GT.0.0.AND.GCWS(IY,IX).EQ.0.0) GO TO 88955
C......i.e.consider scheme ONLY with a max. of 3 +'ve fluxes...else..
88950      GCC(IY,IX)=GCWN(IY,IX)
           GO TO 88980
88955   IF(GCWE(IY,IX).GT.GCWW(IY,IX)) GO TO 88967
          IF(GCWH(IY,IX).GT.GCWL(IY,IX)) GO TO 88961
C......i.e.m(N),m(W)and m(L) involved...
           IF(GCWL(IY,IX).GE.GCWN(IY,IX).AND.GCWL(IY,IX).GE.GCWW(IY,IX))
     1     GO TO 88959
C......i.e.m(L) dominant...
           IF(GCWW(IY,IX).GE.GCWN(IY,IX).AND.GCWW(IY,IX).GE.GCWL(IY,IX))
     2     GO TO 88957
C......i.e.m(W) dominant...otherwise m(N) is dominant...
            IF(GCWW(IY,IX).GE.GCWL(IY,IX).AND.GCWL(IY,IX).GE.0.0)
     3      GO TO 88956
C......i.e.flow specification is m(N)>m(W)>m(L)>=0.0...otherwise..
C.........flow specification is m(N)>m(L)>m(W)>=0.0...
            GCC(IY,IX)=(1.0-(GCWL(IY,IX)/GCWN(IY,IX)))*(GCWL(IY,IX)+
     4                 GCWW(IY,IX)+GCWN(IY,IX))
            GO TO 88980
88956       GCC(IY,IX)=(1.0-(GCWW(IY,IX)/GCWN(IY,IX)))*(GCWL(IY,IX)+
     5                 GCWW(IY,IX)+GCWN(IY,IX))
            GO TO 88980
88957       IF(GCWN(IY,IX).GE.GCWL(IY,IX).AND.GCWL(IY,IX).GE.0.0)
     6      GO TO 88958
            GCC(IY,IX)=0.
            GWPCO(IY,IX)=(GCWN(IY,IX)/GCWW(IY,IX))*(GCWL(IY,IX)+
     7                   GCWW(IY,IX)+GCWN(IY,IX))
            IWPVAL(IY,IX)=5
C  i.e. Phi = @NWL - corner coeff.
            GO TO 88980
88958       GCC(IY,IX)=0.
            GWECO(IY,IX)=((GCWN(IY,IX)-GCWL(IY,IX))/GCWW(IY,IX))*
     8                   (GCWL(IY,IX)+GCWW(IY,IX)+GCWN(IY,IX))
            IWEVAL(IY,IX)=1
C  i.e. Phi = @NW - edge coeff.
            GO TO 88980
88959       IF(GCWW(IY,IX).GE.GCWN(IY,IX).AND.GCWN(IY,IX).GE.0.0)
     9      GO TO 88960
            GCC(IY,IX)=0.
            GWECO(IY,IX)=((GCWN(IY,IX)-GCWW(IY,IX))/GCWL(IY,IX))*
     1                   (GCWL(IY,IX)+GCWW(IY,IX)+GCWN(IY,IX))
            IWEVAL(IY,IX)=9
C  i.e. Phi = @NL - edge coeff
            GO TO 88980
88960       GCC(IY,IX)=0.
            GWPCO(IY,IX)=(GCWN(IY,IX)/GCWL(IY,IX))*(GCWL(IY,IX)+
     2                   GCWW(IY,IX)+GCWN(IY,IX))
            IWPVAL(IY,IX)=5
C  i.e. Phi = @NWL - corner coeff.
            GO TO 88980
C ***********************************************************************
88961      IF(GCWH(IY,IX).GE.GCWN(IY,IX).AND.GCWH(IY,IX).GE.GCWW(IY,IX))
     1     GO TO 88965
           IF(GCWW(IY,IX).GE.GCWN(IY,IX).AND.GCWW(IY,IX).GE.GCWH(IY,IX))
     2     GO TO 88963
            IF(GCWW(IY,IX).GE.GCWH(IY,IX).AND.GCWH(IY,IX).GE.0.0)
     3      GO TO 88962
            GCC(IY,IX)=(1.0-(GCWH(IY,IX)/GCWN(IY,IX)))*(GCWH(IY,IX)+
     4                 GCWW(IY,IX)+GCWN(IY,IX))
            GO TO 88980
88962       GCC(IY,IX)=(1.0-(GCWW(IY,IX)/GCWN(IY,IX)))*(GCWH(IY,IX)+
     5                 GCWW(IY,IX)+GCWN(IY,IX))
            GO TO 88980
88963       IF(GCWN(IY,IX).GE.GCWH(IY,IX).AND.GCWH(IY,IX).GE.0.0)
```

```
     6    GO TO 88964
          GCC(IY,IX)=0.
          GWPCO(IY,IX)=(GCWN(IY,IX)/GCWW(IY,IX))*(GCWH(IY,IX)+
     7                 GCWW(IY,IX)+GCWN(IY,IX))
          IWPVAL(IY,IX)=1
C  i.e. Phi = @NWH - corner coeff.
          GO TO 88980
88964     GCC(IY,IX)=0.
          GWECO(IY,IX)=((GCWN(IY,IX)-GCWH(IY,IX))/GCWW(IY,IX))*
     8                   (GCWH(IY,IX)+GCWW(IY,IX)+GCWN(IY,IX))
          IWEVAL(IY,IX)=1
C  i.e. Phi = @NW - edge coeff.
          GO TO 88980
88965     IF(GCWW(IY,IX).GE.GCWN(IY,IX).AND.GCWN(IY,IX).GE.0.0)
     9    GO TO 88966
          GCC(IY,IX)=0.
          GWECO(IY,IX)=((GCWN(IY,IX)-GCWW(IY,IX))/GCWH(IY,IX))*
     1                   (GCWH(IY,IX)+GCWW(IY,IX)+GCWN(IY,IX))
          IWEVAL(IY,IX)=5
C  i.e. Phi = @NH - edge coeff
          GO TO 88980
88966     GCC(IY,IX)=0.
          GWPCO(IY,IX)=(GCWN(IY,IX)/GCWH(IY,IX))*(GCWH(IY,IX)+
     2                 GCWW(IY,IX)+GCWN(IY,IX))
          IWPVAL(IY,IX)=1
C  i.e. Phi = @NWH - corner coeff.
          GO TO 88980
C **********************************************************************
88967   IF(GCWH(IY,IX).GT.GCWL(IY,IX)) GO TO 88973
         IF(GCWL(IY,IX).GE.GCWN(IY,IX).AND.GCWL(IY,IX).GE.GCWE(IY,IX))
     1   GO TO 88971
         IF(GCWE(IY,IX).GE.GCWN(IY,IX).AND.GCWE(IY,IX).GE.GCWL(IY,IX))
     2   GO TO 88969
          IF(GCWE(IY,IX).GE.GCWL(IY,IX).AND.GCWL(IY,IX).GE.0.0)
     3    GO TO 88968
          GCC(IY,IX)=(1.0-(GCWL(IY,IX)/GCWN(IY,IX)))*(GCWL(IY,IX)+
     4               GCWE(IY,IX)+GCWN(IY,IX))
          GO TO 88980
88968     GCC(IY,IX)=(1.0-(GCWE(IY,IX)/GCWN(IY,IX)))*(GCWL(IY,IX)+
     5               GCWE(IY,IX)+GCWN(IY,IX))
          GO TO 88980
88969     IF(GCWN(IY,IX).GE.GCWL(IY,IX).AND.GCWL(IY,IX).GE.0.0)
     6    GO TO 88970
          GCC(IY,IX)=0.
          GWPCO(IY,IX)=(GCWN(IY,IX)/GCWE(IY,IX))*(GCWL(IY,IX)+
     7                 GCWE(IY,IX)+GCWN(IY,IX))
          IWPVAL(IY,IX)=6
C  i.e. Phi = @NEL - corner coeff.
          GO TO 88980
88970     GCC(IY,IX)=0.
          GWECO(IY,IX)=((GCWN(IY,IX)-GCWL(IY,IX))/GCWE(IY,IX))*
     8                   (GCWL(IY,IX)+GCWE(IY,IX)+GCWN(IY,IX))
          IWEVAL(IY,IX)=2
C  i.e. Phi = @NE - edge coeff.
          GO TO 88980
88971     IF(GCWE(IY,IX).GE.GCWN(IY,IX).AND.GCWN(IY,IX).GE.0.0)
     9    GO TO 88972
          GCC(IY,IX)=0.
          GWECO(IY,IX)=((GCWN(IY,IX)-GCWE(IY,IX))/GCWL(IY,IX))*
     1                   (GCWL(IY,IX)+GCWE(IY,IX)+GCWN(IY,IX))
          IWEVAL(IY,IX)=9
C  i.e. Phi = @NL - edge coeff
          GO TO 88980
88972     GCC(IY,IX)=0.
          GWPCO(IY,IX)=(GCWN(IY,IX)/GCWL(IY,IX))*(GCWL(IY,IX)+
     2                 GCWE(IY,IX)+GCWN(IY,IX))
```

```
          IWPVAL(IY,IX)=6
C  i.e. Phi = @NEL - corner coeff.
          GO TO 88980
C **********************************************************************
88973    IF(GCWH(IY,IX).GE.GCWN(IY,IX).AND.GCWH(IY,IX).GE.GCWE(IY,IX))
     1   GO TO 88977
         IF(GCWE(IY,IX).GE.GCWN(IY,IX).AND.GCWE(IY,IX).GE.GCWH(IY,IX))
     2   GO TO 88975
          IF(GCWE(IY,IX).GE.GCWH(IY,IX).AND.GCWH(IY,IX).GE.0.0)
     3    GO TO 88974
          GCC(IY,IX)=(1.0-(GCWH(IY,IX)/GCWN(IY,IX)))*(GCWH(IY,IX)+
     4               GCWE(IY,IX)+GCWN(IY,IX))
          GO TO 88980
88974     GCC(IY,IX)=(1.0-(GCWE(IY,IX)/GCWN(IY,IX)))*(GCWH(IY,IX)+
     5               GCWE(IY,IX)+GCWN(IY,IX))
          GO TO 88980
88975     IF(GCWN(IY,IX).GE.GCWH(IY,IX).AND.GCWH(IY,IX).GE.0.0)
     6    GO TO 88976
          GCC(IY,IX)=0.
          GWPCO(IY,IX)=(GCWN(IY,IX)/GCWE(IY,IX))*(GCWH(IY,IX)+
     7                 GCWE(IY,IX)+GCWN(IY,IX))
          IWPVAL(IY,IX)=2
C  i.e. Phi = @NEH - corner coeff.
          GO TO 88980
88976     GCC(IY,IX)=0.
          GWECO(IY,IX)=((GCWN(IY,IX)-GCWH(IY,IX))/GCWE(IY,IX))*
     8                  (GCWH(IY,IX)+GCWE(IY,IX)+GCWN(IY,IX))
          IWEVAL(IY,IX)=2
C  i.e. Phi = @NE - edge coeff.
          GO TO 88980
88977     IF(GCWE(IY,IX).GE.GCWN(IY,IX).AND.GCWN(IY,IX).GE.0.0)
     9    GO TO 88978
          GCC(IY,IX)=0.
          GWECO(IY,IX)=((GCWN(IY,IX)-GCWE(IY,IX))/GCWH(IY,IX))*
     1                  (GCWH(IY,IX)+GCWE(IY,IX)+GCWN(IY,IX))
          IWEVAL(IY,IX)=5
C  i.e. Phi = @NH - edge coeff
          GO TO 88980
88978     GCC(IY,IX)=0.
          GWPCO(IY,IX)=(GCWN(IY,IX)/GCWH(IY,IX))*(GCWH(IY,IX)+
     2                 GCWE(IY,IX)+GCWN(IY,IX))
          IWPVAL(IY,IX)=2
C  i.e. Phi = @NEH - corner coeff.
88980 CONTINUE
      CALL SETYX(LD11,GCC,NYDIM,NXDIM)
C      CALL PRN('CCSN',LD11)
C
C   South Coefficient
C
      DO 89010 IX=1,NX
      DO 89010 IY=1,NY
       IF(GCWS(IY,IX).GT.0.0.AND.GCWN(IY,IX).EQ.0.0) GO TO 88985
88982    GCC(IY,IX)=GCWS(IY,IX)
         GO TO 89010
88985  IF(GCWE(IY,IX).GT.GCWW(IY,IX)) GO TO 88997
        IF(GCWH(IY,IX).GT.GCWL(IY,IX)) GO TO 88991
         IF(GCWL(IY,IX).GE.GCWS(IY,IX).AND.GCWL(IY,IX).GE.GCWW(IY,IX))
     1   GO TO 88989
         IF(GCWW(IY,IX).GE.GCWS(IY,IX).AND.GCWW(IY,IX).GE.GCWL(IY,IX))
     2   GO TO 88987
          IF(GCWW(IY,IX).GE.GCWL(IY,IX).AND.GCWL(IY,IX).GE.0.0)
     3    GO TO 88986
          GCC(IY,IX)=(1.0-(GCWL(IY,IX)/GCWS(IY,IX)))*(GCWL(IY,IX)+
     4               GCWW(IY,IX)+GCWS(IY,IX))
          GO TO 89010
88986     GCC(IY,IX)=(1.0-(GCWW(IY,IX)/GCWS(IY,IX)))*(GCWL(IY,IX)+
```

```
     5                GCWW(IY,IX)+GCWS(IY,IX))
          GO TO 89010
88987     IF(GCWS(IY,IX).GE.GCWL(IY,IX).AND.GCWL(IY,IX).GE.0.0)
     6    GO TO 88988
          GCC(IY,IX)=0.
          GWPCO(IY,IX)=(GCWS(IY,IX)/GCWW(IY,IX))*(GCWL(IY,IX)+
     7                GCWW(IY,IX)+GCWS(IY,IX))
          IWPVAL(IY,IX)=7
C  i.e. Phi = @SWL - corner coeff.
          GO TO 89010
88988     GCC(IY,IX)=0.
          GWECO(IY,IX)=((GCWS(IY,IX)-GCWL(IY,IX))/GCWW(IY,IX))*
     8                 (GCWL(IY,IX)+GCWW(IY,IX)+GCWS(IY,IX))
          IWEVAL(IY,IX)=3
C  i.e. Phi = @SW - edge coeff.
          GO TO 89010
88989     IF(GCWW(IY,IX).GE.GCWS(IY,IX).AND.GCWS(IY,IX).GE.0.0)
     9    GO TO 88990
          GCC(IY,IX)=0.
          GWECO(IY,IX)=((GCWS(IY,IX)-GCWW(IY,IX))/GCWL(IY,IX))*
     1                 (GCWL(IY,IX)+GCWW(IY,IX)+GCWS(IY,IX))
          IWEVAL(IY,IX)=10
C  i.e. Phi = @SL - edge coeff
          GO TO 89010
88990     GCC(IY,IX)=0.
          GWPCO(IY,IX)=(GCWS(IY,IX)/GCWL(IY,IX))*(GCWL(IY,IX)+
     2                GCWW(IY,IX)+GCWS(IY,IX))
          IWPVAL(IY,IX)=7
C  i.e. Phi = @SWL - corner coeff.
          GO TO 89010
C *******************************************************************
88991    IF(GCWH(IY,IX).GE.GCWS(IY,IX).AND.GCWH(IY,IX).GE.GCWW(IY,IX))
     1   GO TO 88995
         IF(GCWW(IY,IX).GE.GCWS(IY,IX).AND.GCWW(IY,IX).GE.GCWH(IY,IX))
     2   GO TO 88993
          IF(GCWW(IY,IX).GE.GCWH(IY,IX).AND.GCWH(IY,IX).GE.0.0)
     3    GO TO 88992
          GCC(IY,IX)=(1.0-(GCWH(IY,IX)/GCWS(IY,IX)))*(GCWH(IY,IX)+
     4               GCWW(IY,IX)+GCWS(IY,IX))
          GO TO 89010
88992     GCC(IY,IX)=(1.0-(GCWW(IY,IX)/GCWS(IY,IX)))*(GCWH(IY,IX)+
     5               GCWW(IY,IX)+GCWS(IY,IX))
          GO TO 89010
88993     IF(GCWS(IY,IX).GE.GCWH(IY,IX).AND.GCWH(IY,IX).GE.0.0)
     6    GO TO 88994
          GCC(IY,IX)=0.
          GWPCO(IY,IX)=(GCWS(IY,IX)/GCWW(IY,IX))*(GCWH(IY,IX)+
     7                GCWW(IY,IX)+GCWS(IY,IX))
          IWPVAL(IY,IX)=3
C  i.e. Phi = @SWH - corner coeff.
          GO TO 89010
88994     GCC(IY,IX)=0.
          GWECO(IY,IX)=((GCWS(IY,IX)-GCWH(IY,IX))/GCWW(IY,IX))*
     8                 (GCWH(IY,IX)+GCWW(IY,IX)+GCWS(IY,IX))
          IWEVAL(IY,IX)=3
C  i.e. Phi = @SW - edge coeff.
          GO TO 89010
88995     IF(GCWW(IY,IX).GE.GCWS(IY,IX).AND.GCWS(IY,IX).GE.0.0)
     9    GO TO 88996
          GCC(IY,IX)=0.
          GWECO(IY,IX)=((GCWS(IY,IX)-GCWW(IY,IX))/GCWH(IY,IX))*
     1                 (GCWH(IY,IX)+GCWW(IY,IX)+GCWS(IY,IX))
          IWEVAL(IY,IX)=6
C  i.e. Phi = @SH - edge coeff
          GO TO 89010
88996     GCC(IY,IX)=0.
```

```
        GWPCO(IY,IX)=(GCWS(IY,IX)/GCWH(IY,IX))*(GCWH(IY,IX)+
     2               GCWW(IY,IX)+GCWS(IY,IX))
        IWPVAL(IY,IX)=3
C  i.e. Phi = @SWH - corner coeff.
        GO TO 89010
C ********************************************************************
88997  IF(GCWH(IY,IX).GT.GCWL(IY,IX)) GO TO 89003
       IF(GCWL(IY,IX).GE.GCWS(IY,IX).AND.GCWL(IY,IX).GE.GCWE(IY,IX))
     1 GO TO 89001
       IF(GCWE(IY,IX).GE.GCWS(IY,IX).AND.GCWE(IY,IX).GE.GCWL(IY,IX))
     2 GO TO 88999
        IF(GCWE(IY,IX).GE.GCWL(IY,IX).AND.GCWL(IY,IX).GE.0.0)
     3  GO TO 88998
        GCC(IY,IX)=(1.0-(GCWL(IY,IX)/GCWS(IY,IX)))*(GCWL(IY,IX)+
     4            GCWE(IY,IX)+GCWS(IY,IX))
        GO TO 89010
88998   GCC(IY,IX)=(1.0-(GCWE(IY,IX)/GCWS(IY,IX)))*(GCWL(IY,IX)+
     5            GCWE(IY,IX)+GCWS(IY,IX))
        GO TO 89010
88999   IF(GCWS(IY,IX).GE.GCWL(IY,IX).AND.GCWL(IY,IX).GE.0.0)
     6  GO TO 89000
        GCC(IY,IX)=0.
        GWPCO(IY,IX)=(GCWS(IY,IX)/GCWE(IY,IX))*(GCWL(IY,IX)+
     7              GCWE(IY,IX)+GCWS(IY,IX))
        IWPVAL(IY,IX)=8
C  i.e. Phi = @SEL - corner coeff.
        GO TO 89010
89000   GCC(IY,IX)=0.
        GWECO(IY,IX)=((GCWS(IY,IX)-GCWL(IY,IX))/GCWE(IY,IX))*
     8                (GCWL(IY,IX)+GCWE(IY,IX)+GCWS(IY,IX))
        IWEVAL(IY,IX)=4
C  i.e. Phi = @SE - edge coeff.
        GO TO 89010
89001   IF(GCWE(IY,IX).GE.GCWS(IY,IX).AND.GCWS(IY,IX).GE.0.0)
     9  GO TO 89002
        GCC(IY,IX)=0.
        GWECO(IY,IX)=((GCWS(IY,IX)-GCWE(IY,IX))/GCWL(IY,IX))*
     1                (GCWL(IY,IX)+GCWE(IY,IX)+GCWS(IY,IX))
        IWEVAL(IY,IX)=10
C  i.e. Phi = @SL - edge coeff
        GO TO 89010
89002   GCC(IY,IX)=0.
        GWPCO(IY,IX)=(GCWS(IY,IX)/GCWL(IY,IX))*(GCWL(IY,IX)+
     2              GCWE(IY,IX)+GCWS(IY,IX))
        IWPVAL(IY,IX)=8
C  i.e. Phi = @SEL - corner coeff.
        GO TO 89010
C ********************************************************************
89003  IF(GCWH(IY,IX).GE.GCWS(IY,IX).AND.GCWH(IY,IX).GE.GCWE(IY,IX))
     1 GO TO 89007
       IF(GCWE(IY,IX).GE.GCWS(IY,IX).AND.GCWE(IY,IX).GE.GCWH(IY,IX))
     2 GO TO 89005
        IF(GCWE(IY,IX).GE.GCWH(IY,IX).AND.GCWH(IY,IX).GE.0.0)
     3  GO TO 89004
        GCC(IY,IX)=(1.0-(GCWH(IY,IX)/GCWS(IY,IX)))*(GCWH(IY,IX)+
     4            GCWE(IY,IX)+GCWS(IY,IX))
        GO TO 89010
89004   GCC(IY,IX)=(1.0-(GCWE(IY,IX)/GCWS(IY,IX)))*(GCWH(IY,IX)+
     5            GCWE(IY,IX)+GCWS(IY,IX))
        GO TO 89010
89005   IF(GCWS(IY,IX).GE.GCWH(IY,IX).AND.GCWH(IY,IX).GE.0.0)
     6  GO TO 89006
        GCC(IY,IX)=0.
        GWPCO(IY,IX)=(GCWS(IY,IX)/GCWE(IY,IX))*(GCWH(IY,IX)+
     7              GCWE(IY,IX)+GCWS(IY,IX))
        IWPVAL(IY,IX)=4
```

```
C  i.e. Phi = @SEH - corner coeff.
          GO TO 89010
89006     GCC(IY,IX)=0.
          GWECO(IY,IX)=((GCWS(IY,IX)-GCWH(IY,IX))/GCWE(IY,IX))*
     8                 (GCWH(IY,IX)+GCWE(IY,IX)+GCWS(IY,IX))
          IWEVAL(IY,IX)=4
C  i.e. Phi = @SE - edge coeff.
          GO TO 89010
89007     IF(GCWE(IY,IX).GE.GCWS(IY,IX).AND.GCWS(IY,IX).GE.0.0)
     9    GO TO 89008
          GCC(IY,IX)=0.
          GWECO(IY,IX)=((GCWS(IY,IX)-GCWE(IY,IX))/GCWH(IY,IX))*
     1                 (GCWH(IY,IX)+GCWE(IY,IX)+GCWS(IY,IX))
          IWEVAL(IY,IX)=6
C  i.e. Phi = @SH - edge coeff
          GO TO 89010
89008     GCC(IY,IX)=0.
          GWPCO(IY,IX)=(GCWS(IY,IX)/GCWH(IY,IX))*(GCWH(IY,IX)+
     2                GCWE(IY,IX)+GCWS(IY,IX))
          IWPVAL(IY,IX)=4
C  i.e. Phi = @SEH - corner coeff.
89010 CONTINUE
      CALL SETYX(LD12,GCC,NYDIM,NXDIM)
C      CALL PRN('CCSS',LD12)
      RETURN
C
C    East-West
C
89020 IF(NDIREC.NE.3) GO TO 89100
C   i.e. consider flow through East and West faces.
         CALL GETYX(LD11,GCWE,NYDIM,NXDIM)
         CALL GETYX(LD12,GCWW,NYDIM,NXDIM)
         IF(ISWEEP.EQ.FSWEEP) GO TO 89100
C
C    East Coefficient
C
      DO 89060 IX=1,NX
      DO 89060 IY=1,NY
       IF(GCWE(IY,IX).GT.0.0.AND.GCWW(IY,IX).EQ.0.0) GO TO 89025
89022    GCC(IY,IX)=GCWE(IY,IX)
         GO TO 89060
89025  IF(GCWN(IY,IX).GT.GCWS(IY,IX)) GO TO 89037
        IF(GCWH(IY,IX).GT.GCWL(IY,IX)) GO TO 89031
         IF(GCWL(IY,IX).GE.GCWE(IY,IX).AND.GCWL(IY,IX).GE.GCWS(IY,IX))
     1   GO TO 89029
         IF(GCWS(IY,IX).GE.GCWE(IY,IX).AND.GCWS(IY,IX).GE.GCWL(IY,IX))
     2   GO TO 89027
          IF(GCWS(IY,IX).GE.GCWL(IY,IX).AND.GCWL(IY,IX).GE.0.0)
     3    GO TO 89026
          GCC(IY,IX)=(1.0-(GCWL(IY,IX)/GCWE(IY,IX)))*(GCWL(IY,IX)+
     4              GCWE(IY,IX)+GCWS(IY,IX))
          GO TO 89060
89026     GCC(IY,IX)=(1.0-(GCWS(IY,IX)/GCWE(IY,IX)))*(GCWL(IY,IX)+
     5              GCWE(IY,IX)+GCWS(IY,IX))
          GO TO 89060
89027     IF(GCWE(IY,IX).GE.GCWL(IY,IX).AND.GCWL(IY,IX).GE.0.0)
     6    GO TO 89028
          GCC(IY,IX)=0.
          GWPCO(IY,IX)=(GCWE(IY,IX)/GCWS(IY,IX))*(GCWL(IY,IX)+
     7                GCWE(IY,IX)+GCWS(IY,IX))
          IWPVAL(IY,IX)=8
C  i.e. Phi = @SEL - corner coeff.
          GO TO 89060
89028     GCC(IY,IX)=0.
          GWECO(IY,IX)=((GCWE(IY,IX)-GCWL(IY,IX))/GCWS(IY,IX))*
     8                 (GCWL(IY,IX)+GCWE(IY,IX)+GCWS(IY,IX))
```

```
      IWEVAL(IY,IX)=4
C  i.e. Phi = @SE - edge coeff.
      GO TO 89060
89029 IF(GCWE(IY,IX).GE.GCWS(IY,IX).AND.GCWS(IY,IX).GE.0.0)
     9 GO TO 89030
      GCC(IY,IX)=0.
      GWPCO(IY,IX)=(GCWE(IY,IX)/GCWL(IY,IX))*(GCWL(IY,IX)+
     2             GCWE(IY,IX)+GCWS(IY,IX))
      IWPVAL(IY,IX)=8
C  i.e. Phi = @SEL - corner coeff.
      GO TO 89060
89030 GCC(IY,IX)=0.
      GWECO(IY,IX)=((GCWE(IY,IX)-GCWS(IY,IX))/GCWL(IY,IX))*
     1              (GCWL(IY,IX)+GCWE(IY,IX)+GCWS(IY,IX))
      IWEVAL(IY,IX)=11
C  i.e. Phi = @EL - edge coeff
      GO TO 89060
C *******************************************************************
89031 IF(GCWH(IY,IX).GE.GCWE(IY,IX).AND.GCWH(IY,IX).GE.GCWS(IY,IX))
     1 GO TO 89035
      IF(GCWS(IY,IX).GE.GCWE(IY,IX).AND.GCWS(IY,IX).GE.GCWH(IY,IX))
     2 GO TO 89033
      IF(GCWS(IY,IX).GE.GCWH(IY,IX).AND.GCWH(IY,IX).GE.0.0)
     3 GO TO 89032
      GCC(IY,IX)=(1.0-(GCWH(IY,IX)/GCWE(IY,IX)))*(GCWH(IY,IX)+
     4           GCWE(IY,IX)+GCWS(IY,IX))
89032 GCC(IY,IX)=(1.0-(GCWS(IY,IX)/GCWE(IY,IX)))*(GCWH(IY,IX)+
     5           GCWE(IY,IX)+GCWS(IY,IX))
      GO TO 89060
89033 IF(GCWE(IY,IX).GE.GCWH(IY,IX).AND.GCWH(IY,IX).GE.0.0)
     6 GO TO 89034
      GCC(IY,IX)=0.
      GWPCO(IY,IX)=(GCWE(IY,IX)/GCWS(IY,IX))*(GCWH(IY,IX)+
     7             GCWE(IY,IX)+GCWS(IY,IX))
      IWPVAL(IY,IX)=4
C  i.e. Phi = @SEH - corner coeff.
      GO TO 89060
89034 GCC(IY,IX)=0.
      GWECO(IY,IX)=((GCWE(IY,IX)-GCWH(IY,IX))/GCWS(IY,IX))*
     8              (GCWH(IY,IX)+GCWE(IY,IX)+GCWS(IY,IX))
      IWEVAL(IY,IX)=4
C  i.e. Phi = @SE - edge coeff.
      GO TO 89060
89035 IF(GCWE(IY,IX).GE.GCWS(IY,IX).AND.GCWS(IY,IX).GE.0.0)
     9 GO TO 89036
      GCC(IY,IX)=0.
      GWPCO(IY,IX)=(GCWE(IY,IX)/GCWH(IY,IX))*(GCWH(IY,IX)+
     2             GCWE(IY,IX)+GCWS(IY,IX))
      IWPVAL(IY,IX)=4
C  i.e. Phi = @SEH - corner coeff.
      GO TO 89060
89036 GCC(IY,IX)=0.
      GWECO(IY,IX)=((GCWE(IY,IX)-GCWS(IY,IX))/GCWH(IY,IX))*
     1              (GCWH(IY,IX)+GCWE(IY,IX)+GCWS(IY,IX))
      IWEVAL(IY,IX)=7
C  i.e. Phi = @EH - edge coeff
      GO TO 89060
C *******************************************************************
89037 IF(GCWH(IY,IX).GT.GCWL(IY,IX)) GO TO 89043
      IF(GCWL(IY,IX).GE.GCWN(IY,IX).AND.GCWL(IY,IX).GE.GCWE(IY,IX))
     1 GO TO 89041
      IF(GCWN(IY,IX).GE.GCWE(IY,IX).AND.GCWN(IY,IX).GE.GCWL(IY,IX))
     2 GO TO 89039
      IF(GCWN(IY,IX).GE.GCWL(IY,IX).AND.GCWL(IY,IX).GE.0.0)
     3 GO TO 89038
      GCC(IY,IX)=(1.0-(GCWL(IY,IX)/GCWE(IY,IX)))*(GCWL(IY,IX)+
```

```
     4              GCWE(IY,IX)+GCWN(IY,IX))
          GO TO 89060
89038     GCC(IY,IX)=(1.0-(GCWN(IY,IX)/GCWE(IY,IX)))*(GCWL(IY,IX)+
     5              GCWE(IY,IX)+GCWN(IY,IX))
          GO TO 89060
89039     IF(GCWE(IY,IX).GE.GCWL(IY,IX).AND.GCWL(IY,IX).GE.0.0)
     6    GO TO 89040
          GCC(IY,IX)=0.
          GWPCO(IY,IX)=(GCWE(IY,IX)/GCWN(IY,IX))*(GCWL(IY,IX)+
     7                GCWE(IY,IX)+GCWN(IY,IX))
          IWPVAL(IY,IX)=6
C  i.e. Phi = @NEL - corner coeff.
          GO TO 89060
89040     GCC(IY,IX)=0.
          GWECO(IY,IX)=((GCWE(IY,IX)-GCWL(IY,IX))/GCWN(IY,IX))*
     8                 (GCWL(IY,IX)+GCWE(IY,IX)+GCWN(IY,IX))
          IWEVAL(IY,IX)=2
C  i.e. Phi = @NE - edge coeff.
          GO TO 89060
89041     IF(GCWE(IY,IX).GE.GCWN(IY,IX).AND.GCWN(IY,IX).GE.0.0)
     9    GO TO 89042
          GCC(IY,IX)=0.
          GWPCO(IY,IX)=(GCWE(IY,IX)/GCWL(IY,IX))*(GCWL(IY,IX)+
     2                GCWE(IY,IX)+GCWN(IY,IX))
          IWPVAL(IY,IX)=6
C  i.e. Phi = @NEL - corner coeff.
          GO TO 89060
89042     GCC(IY,IX)=0.
          GWECO(IY,IX)=((GCWE(IY,IX)-GCWN(IY,IX))/GCWL(IY,IX))*
     1                 (GCWL(IY,IX)+GCWE(IY,IX)+GCWN(IY,IX))
          IWEVAL(IY,IX)=11
C  i.e. Phi = @EL - edge coeff
          GO TO 89060
C ************************************************************************
89043    IF(GCWH(IY,IX).GE.GCWN(IY,IX).AND.GCWH(IY,IX).GE.GCWE(IY,IX))
     1   GO TO 89047
         IF(GCWN(IY,IX).GE.GCWE(IY,IX).AND.GCWN(IY,IX).GE.GCWH(IY,IX))
     2   GO TO 89045
          IF(GCWN(IY,IX).GE.GCWH(IY,IX).AND.GCWH(IY,IX).GE.0.0)
     3    GO TO 89044
          GCC(IY,IX)=(1.0-(GCWH(IY,IX)/GCWE(IY,IX)))*(GCWH(IY,IX)+
     4              GCWE(IY,IX)+GCWN(IY,IX))
          GO TO 89060
89044     GCC(IY,IX)=(1.0-(GCWN(IY,IX)/GCWE(IY,IX)))*(GCWH(IY,IX)+
     5              GCWE(IY,IX)+GCWN(IY,IX))
          GO TO 89060
89045     IF(GCWE(IY,IX).GE.GCWH(IY,IX).AND.GCWH(IY,IX).GE.0.0)
     6    GO TO 89046
          GCC(IY,IX)=0.
          GWPCO(IY,IX)=(GCWE(IY,IX)/GCWN(IY,IX))*(GCWH(IY,IX)+
     7                GCWE(IY,IX)+GCWN(IY,IX))
          IWPVAL(IY,IX)=2
C  i.e. Phi = @NEH - corner coeff.
          GO TO 89060
89046     GCC(IY,IX)=0.
          GWECO(IY,IX)=((GCWE(IY,IX)-GCWH(IY,IX))/GCWN(IY,IX))*
     8                 (GCWH(IY,IX)+GCWE(IY,IX)+GCWN(IY,IX))
          IWEVAL(IY,IX)=2
C  i.e. Phi = @NE - edge coeff.
          GO TO 89060
89047     IF(GCWE(IY,IX).GE.GCWN(IY,IX).AND.GCWN(IY,IX).GE.0.0)
     9    GO TO 89048
          GWPCO(IY,IX)=(GCWE(IY,IX)/GCWH(IY,IX))*(GCWH(IY,IX)+
     2                GCWE(IY,IX)+GCWN(IY,IX))
          IWPVAL(IY,IX)=2
C  i.e. Phi = @NEH - corner coeff.
```

```
          GO TO 89060
89048     GCC(IY,IX)=0.
          GWECO(IY,IX)=((GCWE(IY,IX)-GCWN(IY,IX))/GCWH(IY,IX))*
     1                  (GCWH(IY,IX)+GCWE(IY,IX)+GCWN(IY,IX))
          IWEVAL(IY,IX)=7
C  i.e. Phi = @EH - edge coeff.
89060 CONTINUE
      CALL SETYX(LD11,GCC,NYDIM,NXDIM)
C      CALL PRN('CCSE',LD11)
C
C    West Coefficient
C
      DO 89090 IX=1,NX
      DO 89090 IY=1,NY
       IF(GCWW(IY,IX).GT.0.0.AND.GCWE(IY,IX).EQ.0.0) GO TO 89065
89062    GCC(IY,IX)=GCWW(IY,IX)
         GO TO 89090
89065  IF(GCWN(IY,IX).GT.GCWS(IY,IX)) GO TO 89077
        IF(GCWH(IY,IX).GT.GCWL(IY,IX)) GO TO 89071
         IF(GCWL(IY,IX).GE.GCWW(IY,IX).AND.GCWL(IY,IX).GE.GCWS(IY,IX))
     1   GO TO 89069
         IF(GCWS(IY,IX).GE.GCWW(IY,IX).AND.GCWS(IY,IX).GE.GCWL(IY,IX))
     2   GO TO 89067
          IF(GCWS(IY,IX).GE.GCWL(IY,IX).AND.GCWL(IY,IX).GE.0.0)
     3    GO TO 89066
          GCC(IY,IX)=(1.0-(GCWL(IY,IX)/GCWW(IY,IX)))*(GCWL(IY,IX)+
     4              GCWW(IY,IX)+GCWS(IY,IX))
          GO TO 89090
89066     GCC(IY,IX)=(1.0-(GCWS(IY,IX)/GCWW(IY,IX)))*(GCWL(IY,IX)+
     5              GCWW(IY,IX)+GCWS(IY,IX))
          GO TO 89090
89067     IF(GCWW(IY,IX).GE.GCWL(IY,IX).AND.GCWL(IY,IX).GE.0.0)
     6    GO TO 89068
          GCC(IY,IX)=0.
          GWPCO(IY,IX)=(GCWW(IY,IX)/GCWS(IY,IX))*(GCWL(IY,IX)+
     7                GCWW(IY,IX)+GCWS(IY,IX))
          IWPVAL(IY,IX)=7
C  i.e. Phi = @SWL - corner coeff.
          GO TO 89090
89068     GCC(IY,IX)=0.
          GWECO(IY,IX)=((GCWW(IY,IX)-GCWL(IY,IX))/GCWS(IY,IX))*
     8                  (GCWL(IY,IX)+GCWW(IY,IX)+GCWS(IY,IX))
          IWEVAL(IY,IX)=3
C  i.e. Phi = @SW - edge coeff.
          GO TO 89090
89069     IF(GCWW(IY,IX).GE.GCWS(IY,IX).AND.GCWS(IY,IX).GE.0.0)
     9    GO TO 89070
          GCC(IY,IX)=0.
          GWPCO(IY,IX)=(GCWW(IY,IX)/GCWL(IY,IX))*(GCWL(IY,IX)+
     2                GCWW(IY,IX)+GCWS(IY,IX))
          IWPVAL(IY,IX)=7
C  i.e. Phi = @SWL - corner coeff.
          GO TO 89090
89070     GCC(IY,IX)=0.
          GWECO(IY,IX)=((GCWW(IY,IX)-GCWS(IY,IX))/GCWL(IY,IX))*
     1                  (GCWL(IY,IX)+GCWW(IY,IX)+GCWS(IY,IX))
          IWEVAL(IY,IX)=12
C  i.e. Phi = @WL - edge coeff
          GO TO 89090
C **********************************************************************
89071    IF(GCWH(IY,IX).GE.GCWW(IY,IX).AND.GCWH(IY,IX).GE.GCWS(IY,IX))
     1   GO TO 89075
         IF(GCWS(IY,IX).GE.GCWW(IY,IX).AND.GCWS(IY,IX).GE.GCWH(IY,IX))
     2   GO TO 89073
          IF(GCWS(IY,IX).GE.GCWH(IY,IX).AND.GCWH(IY,IX).GE.0.0)
     3    GO TO 89072
```

```
          GCC(IY,IX)=(1.0-(GCWH(IY,IX)/GCWW(IY,IX)))*(GCWH(IY,IX)+
     4              GCWW(IY,IX)+GCWS(IY,IX))
          GO TO 89090
89072     GCC(IY,IX)=(1.0-(GCWS(IY,IX)/GCWW(IY,IX)))*(GCWH(IY,IX)+
     5              GCWW(IY,IX)+GCWS(IY,IX))
          GO TO 89090
89073     IF(GCWW(IY,IX).GE.GCWH(IY,IX).AND.GCWH(IY,IX).GE.0.0)
     6    GO TO 89074
          GCC(IY,IX)=0.
          GWPCO(IY,IX)=(GCWW(IY,IX)/GCWS(IY,IX))*(GCWH(IY,IX)+
     7                GCWW(IY,IX)+GCWS(IY,IX))
          IWPVAL(IY,IX)=3
C  i.e. Phi = @SWH - corner coeff.
          GO TO 89090
89074     GCC(IY,IX)=0.
          GWECO(IY,IX)=((GCWW(IY,IX)-GCWH(IY,IX))/GCWS(IY,IX))*
     8                  (GCWH(IY,IX)+GCWW(IY,IX)+GCWS(IY,IX))
          IWEVAL(IY,IX)=3
C  i.e. Phi = @SW - edge coeff.
          GO TO 89090
89075     IF(GCWW(IY,IX).GE.GCWS(IY,IX).AND.GCWS(IY,IX).GE.0.0)
     9    GO TO 89076
          GCC(IY,IX)=0.
          GWPCO(IY,IX)=(GCWW(IY,IX)/GCWH(IY,IX))*(GCWH(IY,IX)+
     2                GCWW(IY,IX)+GCWS(IY,IX))
          IWPVAL(IY,IX)=3
C  i.e. Phi = @SWH - corner coeff.
          GO TO 89090
89076     GCC(IY,IX)=0.
          GWECO(IY,IX)=((GCWW(IY,IX)-GCWS(IY,IX))/GCWH(IY,IX))*
     1                  (GCWH(IY,IX)+GCWW(IY,IX)+GCWS(IY,IX))
          IWEVAL(IY,IX)=8
C  i.e. Phi = @WH - edge coeff
          GO TO 89090
C ***********************************************************************
89077   IF(GCWH(IY,IX).GT.GCWL(IY,IX)) GO TO 89083
         IF(GCWL(IY,IX).GE.GCWN(IY,IX).AND.GCWL(IY,IX).GE.GCWW(IY,IX))
     1   GO TO 89081
         IF(GCWN(IY,IX).GE.GCWW(IY,IX).AND.GCWN(IY,IX).GE.GCWL(IY,IX))
     2   GO TO 89079
          IF(GCWN(IY,IX).GE.GCWL(IY,IX).AND.GCWL(IY,IX).GE.0.0)
     3    GO TO 89078
          GCC(IY,IX)=(1.0-(GCWL(IY,IX)/GCWW(IY,IX)))*(GCWL(IY,IX)+
     4              GCWW(IY,IX)+GCWN(IY,IX))
          GO TO 89090
89078     GCC(IY,IX)=(1.0-(GCWN(IY,IX)/GCWW(IY,IX)))*(GCWL(IY,IX)+
     5              GCWW(IY,IX)+GCWN(IY,IX))
          GO TO 89090
89079     IF(GCWW(IY,IX).GE.GCWL(IY,IX).AND.GCWL(IY,IX).GE.0.0)
     6    GO TO 89080
          GCC(IY,IX)=0.
          GWPCO(IY,IX)=(GCWW(IY,IX)/GCWN(IY,IX))*(GCWL(IY,IX)+
     7                GCWW(IY,IX)+GCWN(IY,IX))
          IWPVAL(IY,IX)=5
C  i.e. Phi = @NWL - corner coeff.
          GO TO 89090
89080     GCC(IY,IX)=0.
          GWECO(IY,IX)=((GCWW(IY,IX)-GCWL(IY,IX))/GCWN(IY,IX))*
     8                  (GCWL(IY,IX)+GCWW(IY,IX)+GCWN(IY,IX))
          IWEVAL(IY,IX)=1
C  i.e. Phi = @NW - edge coeff.
          GO TO 89090
89081     IF(GCWW(IY,IX).GE.GCWN(IY,IX).AND.GCWN(IY,IX).GE.0.0)
     9    GO TO 89082
          GCC(IY,IX)=0.
          GWPCO(IY,IX)=(GCWW(IY,IX)/GCWL(IY,IX))*(GCWL(IY,IX)+
```

```
     2                GCWW(IY,IX)+GCWN(IY,IX))
          IWPVAL(IY,IX)=5
C  i.e. Phi = @NWL - corner coeff.
          GO TO 89090
89082     GCC(IY,IX)=0.
          GWECO(IY,IX)=((GCWW(IY,IX)-GCWN(IY,IX))/GCWL(IY,IX))*
     1                 (GCWL(IY,IX)+GCWW(IY,IX)+GCWN(IY,IX))
          IWEVAL(IY,IX)=12
C  i.e. Phi = @WL - edge coeff
          GO TO 89090
C *************************************************************************
89083    IF(GCWH(IY,IX).GE.GCWN(IY,IX).AND.GCWH(IY,IX).GE.GCWW(IY,IX))
     1   GO TO 89087
         IF(GCWN(IY,IX).GE.GCWW(IY,IX).AND.GCWN(IY,IX).GE.GCWH(IY,IX))
     2   GO TO 89085
          IF(GCWN(IY,IX).GE.GCWH(IY,IX).AND.GCWH(IY,IX).GE.0.0)
     3    GO TO 89084
          GCC(IY,IX)=(1.0-(GCWH(IY,IX)/GCWW(IY,IX)))*(GCWH(IY,IX)+
     4              GCWW(IY,IX)+GCWN(IY,IX))
          GO TO 89090
89084     GCC(IY,IX)=(1.0-(GCWN(IY,IX)/GCWW(IY,IX)))*(GCWH(IY,IX)+
     5              GCWW(IY,IX)+GCWN(IY,IX))
          GO TO 89090
89085     IF(GCWW(IY,IX).GE.GCWH(IY,IX).AND.GCWH(IY,IX).GE.0.0)
     6    GO TO 89086
          GCC(IY,IX)=0.
          GWPCO(IY,IX)=(GCWW(IY,IX)/GCWN(IY,IX))*(GCWH(IY,IX)+
     7                GCWW(IY,IX)+GCWN(IY,IX))
          IWPVAL(IY,IX)=1
C  i.e. Phi = @NWH - corner coeff.
          GO TO 89090
89086     GCC(IY,IX)=0.
          GWECO(IY,IX)=((GCWW(IY,IX)-GCWH(IY,IX))/GCWN(IY,IX))*
     8                 (GCWH(IY,IX)+GCWW(IY,IX)+GCWN(IY,IX))
          IWEVAL(IY,IX)=1
C  i.e. Phi = @NW - edge coeff.
          GO TO 89090
89087     IF(GCWW(IY,IX).GE.GCWN(IY,IX).AND.GCWN(IY,IX).GE.0.0)
     9    GO TO 89088
          GCC(IY,IX)=0.
          GWPCO(IY,IX)=(GCWW(IY,IX)/GCWH(IY,IX))*(GCWH(IY,IX)+
     2                GCWW(IY,IX)+GCWN(IY,IX))
          IWPVAL(IY,IX)=1
C  i.e. Phi = @NWH - corner coeff.
          GO TO 89090
89088     GCC(IY,IX)=0.
          GWECO(IY,IX)=((GCWW(IY,IX)-GCWN(IY,IX))/GCWH(IY,IX))*
     1                 (GCWH(IY,IX)+GCWW(IY,IX)+GCWN(IY,IX))
          IWEVAL(IY,IX)=8
C  i.e. Phi = @WH - edge coeff.
89090 CONTINUE
      CALL SETYX(LD12,GCC,NYDIM,NXDIM)
C      CALL PRN('CCSW',LD12)
      RETURN
C
C    High-Low
C
89100 IF(NDIREC.NE.5) GO TO 89140
C   i.e. consider flow through High face.
         CALL GETYX(LD2,GCWH,NYDIM,NXDIM)
         IF(ISWEEP.EQ.FSWEEP) GO TO 89140
C    High Coefficient
C
      DO 89130 IX=1,NX
      DO 89130 IY=1,NY
       IF(GCWH(IY,IX).GT.0.0.AND.GCWL(IY,IX).EQ.0.0) GO TO 89105
```

```
         GCC(IY,IX)=GCWH(IY,IX)
         GO TO 89130
89105  IF(GCWN(IY,IX).GT.GCWS(IY,IX)) GO TO 89117
        IF(GCWE(IY,IX).GT.GCWW(IY,IX)) GO TO 89111
         IF(GCWS(IY,IX).GE.GCWH(IY,IX).AND.GCWS(IY,IX).GE.GCWW(IY,IX))
     1   GO TO 89109
         IF(GCWW(IY,IX).GE.GCWH(IY,IX).AND.GCWW(IY,IX).GE.GCWS(IY,IX))
     2   GO TO 89107
          IF(GCWW(IY,IX).GE.GCWS(IY,IX).AND.GCWS(IY,IX).GE.0.0)
     3    GO TO 89106
          GCC(IY,IX)=(1.0-(GCWS(IY,IX)/GCWH(IY,IX)))*(GCWH(IY,IX)+
     4              GCWW(IY,IX)+GCWS(IY,IX))
          GO TO 89130
89106     GCC(IY,IX)=(1.0-(GCWW(IY,IX)/GCWH(IY,IX)))*(GCWH(IY,IX)+
     5              GCWW(IY,IX)+GCWS(IY,IX))
          GO TO 89130
89107     IF(GCWS(IY,IX).GE.GCWH(IY,IX).AND.GCWH(IY,IX).GE.0.0)
     9    GO TO 89108
          GCC(IY,IX)=0.
          GWECO(IY,IX)=((GCWH(IY,IX)-GCWS(IY,IX))/GCWW(IY,IX))*
     1                  (GCWH(IY,IX)+GCWW(IY,IX)+GCWS(IY,IX))
          IWEVAL(IY,IX)=8
C  i.e. Phi = @WH - edge coeff
          GO TO 89130
89108     GCC(IY,IX)=0.
          GWPCO(IY,IX)=(GCWH(IY,IX)/GCWW(IY,IX))*(GCWH(IY,IX)+
     2                GCWW(IY,IX)+GCWS(IY,IX))
          IWPVAL(IY,IX)=3
C  i.e. Phi = @SWH - corner coeff.
          GO TO 89130
89109     IF(GCWW(IY,IX).GE.GCWH(IY,IX).AND.GCWH(IY,IX).GE.0.0)
     9    GO TO 89110
          GCC(IY,IX)=0.
          GWECO(IY,IX)=((GCWH(IY,IX)-GCWW(IY,IX))/GCWS(IY,IX))*
     1                  (GCWH(IY,IX)+GCWW(IY,IX)+GCWS(IY,IX))
          IWEVAL(IY,IX)=6
C  i.e. Phi = @SH - edge coeff
          GO TO 89130
89110     GCC(IY,IX)=0.
          GWPCO(IY,IX)=(GCWH(IY,IX)/GCWS(IY,IX))*(GCWH(IY,IX)+
     2                GCWW(IY,IX)+GCWS(IY,IX))
          IWPVAL(IY,IX)=3
C  i.e. Phi = @SWH - corner coeff.
          GO TO 89130
C ***********************************************************************
89111     IF(GCWS(IY,IX).GE.GCWH(IY,IX).AND.GCWS(IY,IX).GE.GCWE(IY,IX))
     1    GO TO 89115
          IF(GCWE(IY,IX).GE.GCWH(IY,IX).AND.GCWE(IY,IX).GE.GCWS(IY,IX))
     2    GO TO 89113
          IF(GCWE(IY,IX).GE.GCWS(IY,IX).AND.GCWS(IY,IX).GE.0.0)
     3    GO TO 89112
          GCC(IY,IX)=(1.0-(GCWS(IY,IX)/GCWH(IY,IX)))*(GCWH(IY,IX)+
     4              GCWE(IY,IX)+GCWS(IY,IX))
          GO TO 89130
89112     GCC(IY,IX)=(1.0-(GCWE(IY,IX)/GCWH(IY,IX)))*(GCWH(IY,IX)+
     5              GCWE(IY,IX)+GCWS(IY,IX))
          GO TO 89130
89113     IF(GCWS(IY,IX).GE.GCWH(IY,IX).AND.GCWH(IY,IX).GE.0.0)
     9    GO TO 89114
          GCC(IY,IX)=0.
          GWECO(IY,IX)=((GCWH(IY,IX)-GCWS(IY,IX))/GCWE(IY,IX))*
     1                  (GCWH(IY,IX)+GCWE(IY,IX)+GCWS(IY,IX))
          IWEVAL(IY,IX)=7
C  i.e. Phi = @EH - edge coeff
          GO TO 89130
89114     GCC(IY,IX)=0.
```

```
          GWPCO(IY,IX)=(GCWH(IY,IX)/GCWE(IY,IX))*(GCWH(IY,IX)+
     2                GCWE(IY,IX)+GCWS(IY,IX))
          IWPVAL(IY,IX)=4
C  i.e. Phi = @SEH - corner coeff.
          GO TO 89130
89115     IF(GCWE(IY,IX).GE.GCWH(IY,IX).AND.GCWH(IY,IX).GE.0.0)
     9    GO TO 89116
          GCC(IY,IX)=0.
          GWECO(IY,IX)=((GCWH(IY,IX)-GCWE(IY,IX))/GCWS(IY,IX))*
     1                  (GCWH(IY,IX)+GCWE(IY,IX)+GCWS(IY,IX))
          IWEVAL(IY,IX)=6
C  i.e. Phi = @SH - edge coeff
          GO TO 89130
89116     GCC(IY,IX)=0.
          GWPCO(IY,IX)=(GCWH(IY,IX)/GCWS(IY,IX))*(GCWH(IY,IX)+
     2                GCWE(IY,IX)+GCWS(IY,IX))
          IWPVAL(IY,IX)=4
C  i.e. Phi = @SEH - corner coeff.
          GO TO 89130
C *********************************************************************
89117   IF(GCWE(IY,IX).GT.GCWW(IY,IX)) GO TO 89123
         IF(GCWN(IY,IX).GE.GCWH(IY,IX).AND.GCWN(IY,IX).GE.GCWW(IY,IX))
     1   GO TO 89121
         IF(GCWW(IY,IX).GE.GCWH(IY,IX).AND.GCWW(IY,IX).GE.GCWN(IY,IX))
     2   GO TO 89119
          IF(GCWW(IY,IX).GE.GCWN(IY,IX).AND.GCWN(IY,IX).GE.0.0)
     3    GO TO 89118
          GCC(IY,IX)=(1.0-(GCWN(IY,IX)/GCWH(IY,IX)))*(GCWH(IY,IX)+
     4              GCWW(IY,IX)+GCWN(IY,IX))
          GO TO 89130
89118     GCC(IY,IX)=(1.0-(GCWW(IY,IX)/GCWH(IY,IX)))*(GCWH(IY,IX)+
     5              GCWW(IY,IX)+GCWN(IY,IX))
          GO TO 89130
89119     IF(GCWN(IY,IX).GE.GCWH(IY,IX).AND.GCWH(IY,IX).GE.0.0)
     9    GO TO 89120
          GCC(IY,IX)=0.
          GWECO(IY,IX)=((GCWH(IY,IX)-GCWN(IY,IX))/GCWW(IY,IX))*
     1                  (GCWH(IY,IX)+GCWW(IY,IX)+GCWN(IY,IX))
          IWEVAL(IY,IX)=8
C  i.e. Phi = @WH - edge coeff
          GO TO 89130
89120     GCC(IY,IX)=0.
          GWPCO(IY,IX)=(GCWH(IY,IX)/GCWW(IY,IX))*(GCWH(IY,IX)+
     2                GCWW(IY,IX)+GCWN(IY,IX))
          IWPVAL(IY,IX)=1
C  i.e. Phi = @NWH - corner coeff.
          GO TO 89130
89121     IF(GCWW(IY,IX).GE.GCWH(IY,IX).AND.GCWH(IY,IX).GE.0.0)
     9    GO TO 89122
          GCC(IY,IX)=0.
          GWECO(IY,IX)=((GCWH(IY,IX)-GCWW(IY,IX))/GCWN(IY,IX))*
     1                  (GCWH(IY,IX)+GCWW(IY,IX)+GCWN(IY,IX))
          IWEVAL(IY,IX)=5
C  i.e. Phi = @NH - edge coeff
          GO TO 89130
89122     GCC(IY,IX)=0.
          GWPCO(IY,IX)=(GCWW(IY,IX)/GCWN(IY,IX))*(GCWH(IY,IX)+
     2                GCWW(IY,IX)+GCWN(IY,IX))
          IWPVAL(IY,IX)=1
C  i.e. Phi = @NWH - corner coeff.
          GO TO 89130
C **********************************************************************
89123    IF(GCWN(IY,IX).GE.GCWH(IY,IX).AND.GCWN(IY,IX).GE.GCWE(IY,IX))
     1   GO TO 89127
         IF(GCWE(IY,IX).GE.GCWH(IY,IX).AND.GCWE(IY,IX).GE.GCWN(IY,IX))
     2   GO TO 89125
```

```
          IF(GCWE(IY,IX).GE.GCWN(IY,IX).AND.GCWN(IY,IX).GE.0.0)
     3    GO TO 89124
          GCC(IY,IX)=(1.0-(GCWN(IY,IX)/GCWH(IY,IX)))*(GCWH(IY,IX)+
     4              GCWE(IY,IX)+GCWN(IY,IX))
          GO TO 89130
89124     GCC(IY,IX)=(1.0-(GCWE(IY,IX)/GCWH(IY,IX)))*(GCWH(IY,IX)+
     5              GCWE(IY,IX)+GCWN(IY,IX))
          GO TO 89130
89125     IF(GCWN(IY,IX).GE.GCWH(IY,IX).AND.GCWH(IY,IX).GE.0.0)
     9    GO TO 89126
          GCC(IY,IX)=0.
          GWECO(IY,IX)=((GCWH(IY,IX)-GCWN(IY,IX))/GCWE(IY,IX))*
     1                 (GCWH(IY,IX)+GCWE(IY,IX)+GCWN(IY,IX))
          IWEVAL(IY,IX)=7
C  i.e. Phi = @EH - edge coeff
          GO TO 89130
89126     GCC(IY,IX)=0.
          GWPCO(IY,IX)=(GCWH(IY,IX)/GCWE(IY,IX))*(GCWH(IY,IX)+
     2                GCWE(IY,IX)+GCWN(IY,IX))
          IWPVAL(IY,IX)=2
C  i.e. Phi = @NEH - corner coeff.
          GO TO 89130
89127     IF(GCWE(IY,IX).GE.GCWH(IY,IX).AND.GCWH(IY,IX).GE.0.0)
     9    GO TO 89128
          GCC(IY,IX)=0.
          GWECO(IY,IX)=((GCWH(IY,IX)-GCWE(IY,IX))/GCWN(IY,IX))*
     1                 (GCWH(IY,IX)+GCWE(IY,IX)+GCWN(IY,IX))
          IWEVAL(IY,IX)=5
C  i.e. Phi = @NH - edge coeff
          GO TO 89130
89128     GCC(IY,IX)=0.
          GWPCO(IY,IX)=(GCWH(IY,IX)/GCWN(IY,IX))*(GCWH(IY,IX)+
     2                GCWE(IY,IX)+GCWN(IY,IX))
          IWPVAL(IY,IX)=2
C  i.e. Phi = @NEH - corner coeff.
89130 CONTINUE
      CALL SETYX(LD2,GCC,NYDIM,NXDIM)
C      CALL PRN('CCSH',LD2)
      RETURN
C
C   Low Coefficient
C
89140 IF(NDIREC.NE.6) RETURN
C   i.e. consider flow through Low face.
         CALL GETYX(LD2,GCWL,NYDIM,NXDIM)
         IF(ISWEEP.EQ.FSWEEP) RETURN
      DO 89200 IX=1,NX
      DO 89200 IY=1,NY
       IF(GCWL(IY,IX).GT.0.0.AND.GCWH(IY,IX).EQ.0.0) GO TO 89145
         GCC(IY,IX)=GCWL(IY,IX)
         GO TO 89200
89145  IF(GCWN(IY,IX).GT.GCWS(IY,IX)) GO TO 89157
        IF(GCWE(IY,IX).GT.GCWW(IY,IX)) GO TO 89151
         IF(GCWS(IY,IX).GE.GCWL(IY,IX).AND.GCWS(IY,IX).GE.GCWW(IY,IX))
     1   GO TO 89149
         IF(GCWW(IY,IX).GE.GCWL(IY,IX).AND.GCWW(IY,IX).GE.GCWS(IY,IX))
     2   GO TO 89147
          IF(GCWW(IY,IX).GE.GCWS(IY,IX).AND.GCWS(IY,IX).GE.0.0)
     3    GO TO 89146
          GCC(IY,IX)=(1.0-(GCWS(IY,IX)/GCWL(IY,IX)))*(GCWL(IY,IX)+
     4              GCWW(IY,IX)+GCWS(IY,IX))
          GO TO 89200
89146     GCC(IY,IX)=(1.0-(GCWW(IY,IX)/GCWL(IY,IX)))*(GCWL(IY,IX)+
     5              GCWW(IY,IX)+GCWS(IY,IX))
          GO TO 89200
89147     IF(GCWS(IY,IX).GE.GCWL(IY,IX).AND.GCWL(IY,IX).GE.0.0)
```

```
     9    GO TO 89148
          GCC(IY,IX)=0.
          GWECO(IY,IX)=((GCWL(IY,IX)-GCWS(IY,IX))/GCWW(IY,IX))*
     1                (GCWL(IY,IX)+GCWW(IY,IX)+GCWS(IY,IX))
          IWEVAL(IY,IX)=12
C  i.e. Phi = @WL - edge coeff
          GO TO 89200
89148     GCC(IY,IX)=0.
          GWPCO(IY,IX)=(GCWL(IY,IX)/GCWW(IY,IX))*(GCWL(IY,IX)+
     2               GCWW(IY,IX)+GCWS(IY,IX))
          IWPVAL(IY,IX)=7
C  i.e. Phi = @SWL - corner coeff.
          GO TO 89200
89149     IF(GCWW(IY,IX).GE.GCWL(IY,IX).AND.GCWL(IY,IX).GE.0.0)
     9    GO TO 89150
          GCC(IY,IX)=0.
          GWECO(IY,IX)=((GCWL(IY,IX)-GCWW(IY,IX))/GCWS(IY,IX))*
     1                (GCWL(IY,IX)+GCWW(IY,IX)+GCWS(IY,IX))
          IWEVAL(IY,IX)=10
C  i.e. Phi = @SL - edge coeff
          GO TO 89200
89150     GCC(IY,IX)=0.
          GWPCO(IY,IX)=(GCWL(IY,IX)/GCWS(IY,IX))*(GCWL(IY,IX)+
     2               GCWW(IY,IX)+GCWS(IY,IX))
          IWPVAL(IY,IX)=7
C  i.e. Phi = @SWL - corner coeff.
          GO TO 89200
C ***********************************************************************
89151    IF(GCWS(IY,IX).GE.GCWL(IY,IX).AND.GCWS(IY,IX).GE.GCWE(IY,IX))
     1   GO TO 89155
         IF(GCWE(IY,IX).GE.GCWL(IY,IX).AND.GCWE(IY,IX).GE.GCWS(IY,IX))
     2   GO TO 89153
          IF(GCWE(IY,IX).GE.GCWS(IY,IX).AND.GCWS(IY,IX).GE.0.0)
     3    GO TO 89152
          GCC(IY,IX)=(1.0-(GCWS(IY,IX)/GCWL(IY,IX)))*(GCWL(IY,IX)+
     4             GCWE(IY,IX)+GCWS(IY,IX))
          GO TO 89200
89152     GCC(IY,IX)=(1.0-(GCWE(IY,IX)/GCWL(IY,IX)))*(GCWL(IY,IX)+
     5             GCWE(IY,IX)+GCWS(IY,IX))
          GO TO 89200
89153     IF(GCWS(IY,IX).GE.GCWL(IY,IX).AND.GCWL(IY,IX).GE.0.0)
     9    GO TO 89154
          GCC(IY,IX)=0.
          GWECO(IY,IX)=((GCWL(IY,IX)-GCWS(IY,IX))/GCWE(IY,IX))*
     1                (GCWL(IY,IX)+GCWE(IY,IX)+GCWS(IY,IX))
          IWEVAL(IY,IX)=11
C  i.e. Phi = @EL - edge coeff
          GO TO 89200
89154     GCC(IY,IX)=0.
          GWPCO(IY,IX)=(GCWL(IY,IX)/GCWE(IY,IX))*(GCWL(IY,IX)+
     2               GCWE(IY,IX)+GCWS(IY,IX))
          IWPVAL(IY,IX)=8
C  i.e. Phi = @SEL - corner coeff.
          GO TO 89200
89155     IF(GCWE(IY,IX).GE.GCWL(IY,IX).AND.GCWL(IY,IX).GE.0.0)
     9    GO TO 89156
          GCC(IY,IX)=0.
          GWECO(IY,IX)=((GCWL(IY,IX)-GCWE(IY,IX))/GCWS(IY,IX))*
     1                (GCWL(IY,IX)+GCWE(IY,IX)+GCWS(IY,IX))
          IWEVAL(IY,IX)=10
C  i.e. Phi = @SL - edge coeff
          GO TO 89200
89156     GCC(IY,IX)=0.
          GWPCO(IY,IX)=(GCWL(IY,IX)/GCWS(IY,IX))*(GCWL(IY,IX)+
     2               GCWE(IY,IX)+GCWS(IY,IX))
          IWPVAL(IY,IX)=8
```

```
C  i.e. Phi = @SEL - corner coeff.
          GO TO 89200
C *********************************************************************
89157   IF(GCWE(IY,IX).GT.GCWW(IY,IX)) GO TO 89163
         IF(GCWN(IY,IX).GE.GCWL(IY,IX).AND.GCWN(IY,IX).GE.GCWW(IY,IX))
     1   GO TO 89161
         IF(GCWW(IY,IX).GE.GCWL(IY,IX).AND.GCWW(IY,IX).GE.GCWN(IY,IX))
     2   GO TO 89159
          IF(GCWW(IY,IX).GE.GCWN(IY,IX).AND.GCWN(IY,IX).GE.0.0)
     3    GO TO 89158
          GCC(IY,IX)=(1.0-(GCWN(IY,IX)/GCWL(IY,IX)))*(GCWL(IY,IX)+
     4              GCWW(IY,IX)+GCWN(IY,IX))
          GO TO 89200
89158     GCC(IY,IX)=(1.0-(GCWW(IY,IX)/GCWL(IY,IX)))*(GCWL(IY,IX)+
     5              GCWW(IY,IX)+GCWN(IY,IX))
          GO TO 89200
89159     IF(GCWN(IY,IX).GE.GCWL(IY,IX).AND.GCWL(IY,IX).GE.0.0)
     9    GO TO 89160
          GCC(IY,IX)=0.
          GWECO(IY,IX)=((GCWL(IY,IX)-GCWN(IY,IX))/GCWW(IY,IX))*
     1                 (GCWL(IY,IX)+GCWW(IY,IX)+GCWN(IY,IX))
          IWEVAL(IY,IX)=12
C  i.e. Phi = @WL - edge coeff
          GO TO 89200
89160     GCC(IY,IX)=0.
          GWPCO(IY,IX)=(GCWL(IY,IX)/GCWW(IY,IX))*(GCWL(IY,IX)+
     2                GCWW(IY,IX)+GCWN(IY,IX))
          IWPVAL(IY,IX)=5
C  i.e. Phi = @NWL - corner coeff.
          GO TO 89200
89161     IF(GCWW(IY,IX).GE.GCWL(IY,IX).AND.GCWL(IY,IX).GE.0.0)
     9    GO TO 89162
          GCC(IY,IX)=0.
          GWECO(IY,IX)=((GCWL(IY,IX)-GCWW(IY,IX))/GCWN(IY,IX))*
     1                 (GCWL(IY,IX)+GCWW(IY,IX)+GCWN(IY,IX))
          IWEVAL(IY,IX)=9
C  i.e. Phi = @NL - edge coeff
          GO TO 89200
89162     GCC(IY,IX)=0.
          GWPCO(IY,IX)=(GCWL(IY,IX)/GCWN(IY,IX))*(GCWL(IY,IX)+
     2                GCWW(IY,IX)+GCWN(IY,IX))
          IWPVAL(IY,IX)=5
C  i.e. Phi = @NWL - corner coeff.
          GO TO 89200
C *********************************************************************
89163    IF(GCWN(IY,IX).GE.GCWL(IY,IX).AND.GCWN(IY,IX).GE.GCWE(IY,IX))
     1   GO TO 89167
         IF(GCWE(IY,IX).GE.GCWL(IY,IX).AND.GCWE(IY,IX).GE.GCWN(IY,IX))
     2   GO TO 89165
          IF(GCWE(IY,IX).GE.GCWN(IY,IX).AND.GCWN(IY,IX).GE.0.0)
     3    GO TO 89164
          GCC(IY,IX)=(1.0-(GCWN(IY,IX)/GCWL(IY,IX)))*(GCWL(IY,IX)+
     4              GCWE(IY,IX)+GCWN(IY,IX))
          GO TO 89200
89164     GCC(IY,IX)=(1.0-(GCWE(IY,IX)/GCWL(IY,IX)))*(GCWL(IY,IX)+
     5              GCWE(IY,IX)+GCWN(IY,IX))
          GO TO 89200
89165     IF(GCWN(IY,IX).GE.GCWL(IY,IX).AND.GCWL(IY,IX).GE.0.0)
     9    GO TO 89166
          GCC(IY,IX)=0.
          GWECO(IY,IX)=((GCWL(IY,IX)-GCWN(IY,IX))/GCWE(IY,IX))*
     1                 (GCWL(IY,IX)+GCWE(IY,IX)+GCWN(IY,IX))
          IWEVAL(IY,IX)=11
C  i.e. Phi = @EL - edge coeff
          GO TO 89200
89166     GCC(IY,IX)=0.
```

```
          GWPCO(IY,IX)=(GCWL(IY,IX)/GCWE(IY,IX))*(GCWL(IY,IX)+
     2                GCWE(IY,IX)+GCWN(IY,IX))
          IWPVAL(IY,IX)=6
C  i.e. Phi = @NEL - corner coeff.
          GO TO 89200
89167     IF(GCWE(IY,IX).GE.GCWL(IY,IX).AND.GCWL(IY,IX).GE.0.0)
     9    GO TO 89168
          GCC(IY,IX)=0.
          GWECO(IY,IX)=((GCWL(IY,IX)-GCWE(IY,IX))/GCWN(IY,IX))*
     1                 (GCWL(IY,IX)+GCWE(IY,IX)+GCWN(IY,IX))
          IWEVAL(IY,IX)=9
C  i.e. Phi = @NL - edge coeff
          GO TO 89200
89168     GCC(IY,IX)=0.
          GWPCO(IY,IX)=(GCWL(IY,IX)/GCWN(IY,IX))*(GCWL(IY,IX)+
     2                GCWE(IY,IX)+GCWN(IY,IX))
          IWPVAL(IY,IX)=6
C  i.e. Phi = @NEL - corner coeff.
89200 CONTINUE
      CALL SETYX(LD2,GCC,NYDIM,NXDIM)
C      CALL PRN('CCSL',LD2)
      RETURN
      END
```

```
C*************************************************************
      SUBROUTINE GROUND4
      INCLUDE 'lp16/d_earth/SATEAR'
      INCLUDE 'lp16/d_earth/GRDLOC'
      INCLUDE 'lp16/d_earth/GRDEAR'
      INCLUDE 'lp16/d_earth/GRDBFC'
CXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX USER SECTION STARTS:
C
C 1   Set dimensions of data-for-GROUND arrays here. WARNING: the
C     corresponding arrays in the MAIN program of the satellite
C     and EARTH must have the same dimensions.
      PARAMETER (NLG=20, NIG=20, NRG=100, NCG=10)
C
      COMMON/LGRND/LG(NLG)/IGRND/IG(NIG)/RGRND/RG(NRG)/CGRND/CG(NCG)
      LOGICAL LG
      CHARACTER*4 CG
      COMMON/GUGRND/GCSN(60,80),GCSS(60,80),GCSE(60,80),GCSW(60,80),
     7              GCSL(60,80),GCSH(60,80),GSPCO(60,80),GSECO(60,80),
     8              GCUN(60,80),GCUS(60,80),GCUE(60,80),GCUW(60,80),
     7              GCUL(60,80),GCUH(60,80),GUPCO(60,80),GUECO(60,80),
     8              GCVN(60,80),GCVS(60,80),GCVE(60,80),GCVW(60,80),
     7              GCVL(60,80),GCVH(60,80),GVPCO(60,80),GVECO(60,80),
     8              GCWN(60,80),GCWS(60,80),GCWE(60,80),GCWW(60,80),
     7              GCWL(60,80),GCWH(60,80),GWPCO(60,80),GWECO(60,80)
      COMMON/IUGRND/ISPVAL(60,80),ISEVAL(60,80),IUPVAL(60,80),
     7              IUEVAL(60,80),IVEVAL(60,80),IVPVAL(60,80),
     8              IWEVAL(60,80),IWPVAL(60,80)
C
C 2   User dimensions own arrays here, for example:
C     DIMENSION GUH(10,10),GUC(10,10),GUX(10,10),GUZ(10)
      DIMENSION GCC(60,80),GCMH(60,80),GCML(60,80),
     1          GCE(60,80),GEMP(60,80),GEMH(60,80),GEML(60,80)
C
C 3   User places his data statements here, for example:
      DATA NXDIM,NYDIM/80,60/
C
C 4   Insert own coding below as desired, guided by GREX examples.
C     Note that the satellite-to-GREX special data in the labelled
C     COMMONs /RSG/, /ISG/, /LSG/ and /CSG/ can be included and
C     used below but the user must check GREX for any conflicting
C     uses. The same comment applies to the EARTH-spare working
C     arrays EASP1, EASP2,....EASP20. In addition to the EASPs,
C     there are 10 GRound-earth SPare arrays, GRSP1,...,GRSP10,
C     supplied solely for the user, which are not used by GREX. If
C     the call to GREX has been deactivated then all of the arrays
C     may be used without reservation.
C
      IF(.NOT.(LG(4))) RETURN
      IF(INDVAR.GT.11) GO TO 88949
      RETURN
88949 CONTINUE
C
C  Corrected Coefficients For Scalar-Equation
C  *******************************************
C
C  North-South
C
      IF(NDIREC.NE.1) GO TO 89020
C  i.e. consider flow through North and South faces.
         CALL GETYX(LD11,GCSN,NYDIM,NXDIM)
         CALL GETYX(LD12,GCSS,NYDIM,NXDIM)
         IF(ISWEEP.EQ.FSWEEP) GO TO 89020
C
C
C  North Coefficient
      DO 88980 IX=1,NX
```

```
      DO 88980 IY=1,NY
       IF(GCSN(IY,IX).GT.0.0.AND.GCSS(IY,IX).EQ.0.0) GO TO 88955
C......i.e.consider scheme ONLY with a max. of 3 +'ve fluxes...else..
88950    GCC(IY,IX)=GCSN(IY,IX)
         GO TO 88980
88955  IF(GCSE(IY,IX).GT.GCSW(IY,IX)) GO TO 88967
        IF(GCSH(IY,IX).GT.GCSL(IY,IX)) GO TO 88961
C......i.e.m(N),m(W)and m(L) involved...
         IF(GCSL(IY,IX).GE.GCSN(IY,IX).AND.GCSL(IY,IX).GE.GCSW(IY,IX))
     1   GO TO 88959
C......i.e.m(L) dominant...
         IF(GCSW(IY,IX).GE.GCSN(IY,IX).AND.GCSW(IY,IX).GE.GCSL(IY,IX))
     2   GO TO 88957
C......i.e.m(W) dominant...otherwise m(N) is dominant...
          IF(GCSW(IY,IX).GE.GCSL(IY,IX).AND.GCSL(IY,IX).GE.0.0)
     3    GO TO 88956
C......i.e.flow specification is m(N)>m(W)>m(L)>=0.0...otherwise..
C..........flow specification is m(N)>m(L)>m(W)>=0.0...
          GCC(IY,IX)=(1.0-(GCSL(IY,IX)/GCSN(IY,IX)))*(GCSL(IY,IX)+
     4                GCSW(IY,IX)+GCSN(IY,IX))
          GO TO 88980
88956     GCC(IY,IX)=(1.0-(GCSW(IY,IX)/GCSN(IY,IX)))*(GCSL(IY,IX)+
     5                GCSW(IY,IX)+GCSN(IY,IX))
          GO TO 88980
88957     IF(GCSN(IY,IX).GE.GCSL(IY,IX).AND.GCSL(IY,IX).GE.0.0)
     6    GO TO 88958
          GCC(IY,IX)=0.
          GSPCO(IY,IX)=(GCSN(IY,IX)/GCSW(IY,IX))*(GCSL(IY,IX)+
     7                  GCSW(IY,IX)+GCSN(IY,IX))
          ISPVAL(IY,IX)=5
C  i.e. Phi = @NWL - corner coeff.
          GO TO 88980
88958     GCC(IY,IX)=0.
          GSECO(IY,IX)=((GCSN(IY,IX)-GCSL(IY,IX))/GCSW(IY,IX))*
     8                  (GCSL(IY,IX)+GCSW(IY,IX)+GCSN(IY,IX))
          ISEVAL(IY,IX)=1
C  i.e. Phi = @NW - edge coeff.
          GO TO 88980
88959     IF(GCSW(IY,IX).GE.GCSN(IY,IX).AND.GCSN(IY,IX).GE.0.0)
     9    GO TO 88960
          GCC(IY,IX)=0.
          GSECO(IY,IX)=((GCSN(IY,IX)-GCSW(IY,IX))/GCSL(IY,IX))*
     1                  (GCSL(IY,IX)+GCSW(IY,IX)+GCSN(IY,IX))
          ISEVAL(IY,IX)=9
C  i.e. Phi = @NL - edge coeff
          GO TO 88980
88960     GCC(IY,IX)=0.
          GSPCO(IY,IX)=(GCSN(IY,IX)/GCSL(IY,IX))*(GCSL(IY,IX)+
     2                  GCSW(IY,IX)+GCSN(IY,IX))
          ISPVAL(IY,IX)=5
C  i.e. Phi = @NWL - corner coeff.
          GO TO 88980
C **********************************************************************
88961    IF(GCSH(IY,IX).GE.GCSN(IY,IX).AND.GCSH(IY,IX).GE.GCSW(IY,IX))
     1   GO TO 88965
         IF(GCSW(IY,IX).GE.GCSN(IY,IX).AND.GCSW(IY,IX).GE.GCSH(IY,IX))
     2   GO TO 88963
          IF(GCSW(IY,IX).GE.GCSH(IY,IX).AND.GCSH(IY,IX).GE.0.0)
     3    GO TO 88962
          GCC(IY,IX)=(1.0-(GCSH(IY,IX)/GCSN(IY,IX)))*(GCSH(IY,IX)+
     4                GCSW(IY,IX)+GCSN(IY,IX))
          GO TO 88980
88962     GCC(IY,IX)=(1.0-(GCSW(IY,IX)/GCSN(IY,IX)))*(GCSH(IY,IX)+
     5                GCSW(IY,IX)+GCSN(IY,IX))
          GO TO 88980
88963     IF(GCSN(IY,IX).GE.GCSH(IY,IX).AND.GCSH(IY,IX).GE.0.0)
```

```
     6    GO TO 88964
          GCC(IY,IX)=0.
          GSPCO(IY,IX)=(GCSN(IY,IX)/GCSW(IY,IX))*(GCSH(IY,IX)+
     7                 GCSW(IY,IX)+GCSN(IY,IX))
          ISPVAL(IY,IX)=1
C  i.e. Phi = @NWH - corner coeff.
          GO TO 88980
88964     GCC(IY,IX)=0.
          GSECO(IY,IX)=((GCSN(IY,IX)-GCSH(IY,IX))/GCSW(IY,IX))*
     8                 (GCSH(IY,IX)+GCSW(IY,IX)+GCSN(IY,IX))
          ISEVAL(IY,IX)=1
C  i.e. Phi = @NW - edge coeff.
          GO TO 88980
88965     IF(GCSW(IY,IX).GE.GCSN(IY,IX).AND.GCSN(IY,IX).GE.0.0)
     9    GO TO 88966
          GCC(IY,IX)=0.
          GSECO(IY,IX)=((GCSN(IY,IX)-GCSW(IY,IX))/GCSH(IY,IX))*
     1                 (GCSH(IY,IX)+GCSW(IY,IX)+GCSN(IY,IX))
          ISEVAL(IY,IX)=5
C  i.e. Phi = @NH - edge coeff
          GO TO 88980
88966     GCC(IY,IX)=0.
          GSPCO(IY,IX)=(GCSN(IY,IX)/GCSH(IY,IX))*(GCSH(IY,IX)+
     2                 GCSW(IY,IX)+GCSN(IY,IX))
          ISPVAL(IY,IX)=1
C  i.e. Phi = @NWH - corner coeff.
          GO TO 88980
C ********************************************************************
88967   IF(GCSH(IY,IX).GT.GCSL(IY,IX)) GO TO 88973
         IF(GCSL(IY,IX).GE.GCSN(IY,IX).AND.GCSL(IY,IX).GE.GCSE(IY,IX))
     1   GO TO 88971
         IF(GCSE(IY,IX).GE.GCSN(IY,IX).AND.GCSE(IY,IX).GE.GCSL(IY,IX))
     2   GO TO 88969
          IF(GCSE(IY,IX).GE.GCSL(IY,IX).AND.GCSL(IY,IX).GE.0.0)
     3    GO TO 88968
          GCC(IY,IX)=(1.0-(GCSL(IY,IX)/GCSN(IY,IX)))*(GCSL(IY,IX)+
     4               GCSE(IY,IX)+GCSN(IY,IX))
          GO TO 88980
88968     GCC(IY,IX)=(1.0-(GCSE(IY,IX)/GCSN(IY,IX)))*(GCSL(IY,IX)+
     5               GCSE(IY,IX)+GCSN(IY,IX))
          GO TO 88980
88969     IF(GCSN(IY,IX).GE.GCSL(IY,IX).AND.GCSL(IY,IX).GE.0.0)
     6    GO TO 88970
          GCC(IY,IX)=0.
          GSPCO(IY,IX)=(GCSN(IY,IX)/GCSE(IY,IX))*(GCSL(IY,IX)+
     7                 GCSE(IY,IX)+GCSN(IY,IX))
          ISPVAL(IY,IX)=6
C  i.e. Phi = @NEL - corner coeff.
          GO TO 88980
88970     GCC(IY,IX)=0.
          GSECO(IY,IX)=((GCSN(IY,IX)-GCSL(IY,IX))/GCSE(IY,IX))*
     8                 (GCSL(IY,IX)+GCSE(IY,IX)+GCSN(IY,IX))
          ISEVAL(IY,IX)=2
C  i.e. Phi = @NE - edge coeff.
          GO TO 88980
88971     IF(GCSE(IY,IX).GE.GCSN(IY,IX).AND.GCSN(IY,IX).GE.0.0)
     9    GO TO 88972
          GCC(IY,IX)=0.
          GSECO(IY,IX)=((GCSN(IY,IX)-GCSE(IY,IX))/GCSL(IY,IX))*
     1                 (GCSL(IY,IX)+GCSE(IY,IX)+GCSN(IY,IX))
          ISEVAL(IY,IX)=9
C  i.e. Phi = @NL - edge coeff
          GO TO 88980
88972     GCC(IY,IX)=0.
          GSPCO(IY,IX)=(GCSN(IY,IX)/GCSL(IY,IX))*(GCSL(IY,IX)+
     2                 GCSE(IY,IX)+GCSN(IY,IX))
```

```
          ISPVAL(IY,IX)=6
C  i.e. Phi = @NEL - corner coeff.
          GO TO 88980
C ************************************************************************
88973    IF(GCSH(IY,IX).GE.GCSN(IY,IX).AND.GCSH(IY,IX).GE.GCSE(IY,IX))
     1   GO TO 88977
         IF(GCSE(IY,IX).GE.GCSN(IY,IX).AND.GCSE(IY,IX).GE.GCSH(IY,IX))
     2   GO TO 88975
          IF(GCSE(IY,IX).GE.GCSH(IY,IX).AND.GCSH(IY,IX).GE.0.0)
     3    GO TO 88974
          GCC(IY,IX)=(1.0-(GCSH(IY,IX)/GCSN(IY,IX)))*(GCSH(IY,IX)+
     4               GCSE(IY,IX)+GCSN(IY,IX))
          GO TO 88980
88974     GCC(IY,IX)=(1.0-(GCSE(IY,IX)/GCSN(IY,IX)))*(GCSH(IY,IX)+
     5               GCSE(IY,IX)+GCSN(IY,IX))
          GO TO 88980
88975     IF(GCSN(IY,IX).GE.GCSH(IY,IX).AND.GCSH(IY,IX).GE.0.0)
     6    GO TO 88976
          GCC(IY,IX)=0.
          GSPCO(IY,IX)=(GCSN(IY,IX)/GCSE(IY,IX))*(GCSH(IY,IX)+
     7                 GCSE(IY,IX)+GCSN(IY,IX))
          ISPVAL(IY,IX)=2
C  i.e. Phi = @NEH - corner coeff.
          GO TO 88980
88976     GCC(IY,IX)=0.
          GSECO(IY,IX)=((GCSN(IY,IX)-GCSH(IY,IX))/GCSE(IY,IX))*
     8                 (GCSH(IY,IX)+GCSE(IY,IX)+GCSN(IY,IX))
          ISEVAL(IY,IX)=2
C  i.e. Phi = @NE - edge coeff.
          GO TO 88980
88977     IF(GCSE(IY,IX).GE.GCSN(IY,IX).AND.GCSN(IY,IX).GE.0.0)
     9    GO TO 88978
          GCC(IY,IX)=0.
          GSECO(IY,IX)=((GCSN(IY,IX)-GCSE(IY,IX))/GCSH(IY,IX))*
     1                 (GCSH(IY,IX)+GCSE(IY,IX)+GCSN(IY,IX))
          ISEVAL(IY,IX)=5
C  i.e. Phi = @NH - edge coeff
          GO TO 88980
88978     GCC(IY,IX)=0.
          GSPCO(IY,IX)=(GCSN(IY,IX)/GCSH(IY,IX))*(GCSH(IY,IX)+
     2                 GCSE(IY,IX)+GCSN(IY,IX))
          ISPVAL(IY,IX)=2
C  i.e. Phi = @NEH - corner coeff.
88980 CONTINUE
      CALL SETYX(LD11,GCC,NYDIM,NXDIM)
C      CALL PRN('CCSN',LD11)
C
C   South Coefficient
C
      DO 89010 IX=1,NX
      DO 89010 IY=1,NY
       IF(GCSS(IY,IX).GT.0.0.AND.GCSN(IY,IX).EQ.0.0) GO TO 88985
88982    GCC(IY,IX)=GCSS(IY,IX)
         GO TO 89010
88985  IF(GCSE(IY,IX).GT.GCSW(IY,IX)) GO TO 88997
        IF(GCSH(IY,IX).GT.GCSL(IY,IX)) GO TO 88991
         IF(GCSL(IY,IX).GE.GCSS(IY,IX).AND.GCSL(IY,IX).GE.GCSW(IY,IX))
     1   GO TO 88989
         IF(GCSW(IY,IX).GE.GCSS(IY,IX).AND.GCSW(IY,IX).GE.GCSL(IY,IX))
     2   GO TO 88987
          IF(GCSW(IY,IX).GE.GCSL(IY,IX).AND.GCSL(IY,IX).GE.0.0)
     3    GO TO 88986
          GCC(IY,IX)=(1.0-(GCSL(IY,IX)/GCSS(IY,IX)))*(GCSL(IY,IX)+
     4               GCSW(IY,IX)+GCSS(IY,IX))
          GO TO 89010
88986     GCC(IY,IX)=(1.0-(GCSW(IY,IX)/GCSS(IY,IX)))*(GCSL(IY,IX)+
```

```
     5                GCSW(IY,IX)+GCSS(IY,IX))
          GO TO 89010
88987     IF(GCSS(IY,IX).GE.GCSL(IY,IX).AND.GCSL(IY,IX).GE.0.0)
     6    GO TO 88988
          GCC(IY,IX)=0.
          GSPCO(IY,IX)=(GCSS(IY,IX)/GCSW(IY,IX))*(GCSL(IY,IX)+
     7                  GCSW(IY,IX)+GCSS(IY,IX))
          ISPVAL(IY,IX)=7
C  i.e. Phi = @SWL - corner coeff.
          GO TO 89010
88988     GCC(IY,IX)=0.
          GSECO(IY,IX)=((GCSS(IY,IX)-GCSL(IY,IX))/GCSW(IY,IX))*
     8                  (GCSL(IY,IX)+GCSW(IY,IX)+GCSS(IY,IX))
          ISEVAL(IY,IX)=3
C  i.e. Phi = @SW - edge coeff.
          GO TO 89010
88989     IF(GCSW(IY,IX).GE.GCSS(IY,IX).AND.GCSS(IY,IX).GE.0.0)
     9    GO TO 88990
          GCC(IY,IX)=0.
          GSECO(IY,IX)=((GCSS(IY,IX)-GCSW(IY,IX))/GCSL(IY,IX))*
     1                  (GCSL(IY,IX)+GCSW(IY,IX)+GCSS(IY,IX))
          ISEVAL(IY,IX)=10
C  i.e. Phi = @SL - edge coeff
          GO TO 89010
88990     GCC(IY,IX)=0.
          GSPCO(IY,IX)=(GCSS(IY,IX)/GCSL(IY,IX))*(GCSL(IY,IX)+
     2                  GCSW(IY,IX)+GCSS(IY,IX))
          ISPVAL(IY,IX)=7
C  i.e. Phi = @SWL - corner coeff.
          GO TO 89010
C ********************************************************************
88991    IF(GCSH(IY,IX).GE.GCSS(IY,IX).AND.GCSH(IY,IX).GE.GCSW(IY,IX))
     1   GO TO 88995
         IF(GCSW(IY,IX).GE.GCSS(IY,IX).AND.GCSW(IY,IX).GE.GCSH(IY,IX))
     2   GO TO 88993
          IF(GCSW(IY,IX).GE.GCSH(IY,IX).AND.GCSH(IY,IX).GE.0.0)
     3    GO TO 88992
          GCC(IY,IX)=(1.0-(GCSH(IY,IX)/GCSS(IY,IX)))*(GCSH(IY,IX)+
     4                GCSW(IY,IX)+GCSS(IY,IX))
          GO TO 89010
88992     GCC(IY,IX)=(1.0-(GCSW(IY,IX)/GCSS(IY,IX)))*(GCSH(IY,IX)+
     5                GCSW(IY,IX)+GCSS(IY,IX))
          GO TO 89010
88993     IF(GCSS(IY,IX).GE.GCSH(IY,IX).AND.GCSH(IY,IX).GE.0.0)
     6    GO TO 88994
          GCC(IY,IX)=0.
          GSPCO(IY,IX)=(GCSS(IY,IX)/GCSW(IY,IX))*(GCSH(IY,IX)+
     7                  GCSW(IY,IX)+GCSS(IY,IX))
          ISPVAL(IY,IX)=3
C  i.e. Phi = @SWH - corner coeff.
          GO TO 89010
88994     GCC(IY,IX)=0.
          GSECO(IY,IX)=((GCSS(IY,IX)-GCSH(IY,IX))/GCSW(IY,IX))*
     8                  (GCSH(IY,IX)+GCSW(IY,IX)+GCSS(IY,IX))
          ISEVAL(IY,IX)=3
C  i.e. Phi = @SW - edge coeff.
          GO TO 89010
88995     IF(GCSW(IY,IX).GE.GCSS(IY,IX).AND.GCSS(IY,IX).GE.0.0)
     9    GO TO 88996
          GCC(IY,IX)=0.
          GSECO(IY,IX)=((GCSS(IY,IX)-GCSW(IY,IX))/GCSH(IY,IX))*
     1                  (GCSH(IY,IX)+GCSW(IY,IX)+GCSS(IY,IX))
          ISEVAL(IY,IX)=6
C  i.e. Phi = @SH - edge coeff
          GO TO 89010
88996     GCC(IY,IX)=0.
```

```
          GSPCO(IY,IX)=(GCSS(IY,IX)/GCSH(IY,IX))*(GCSH(IY,IX)+
     2                  GCSW(IY,IX)+GCSS(IY,IX))
          ISPVAL(IY,IX)=3
C  i.e. Phi = @SWH - corner coeff.
          GO TO 89010
C **********************************************************************
88997   IF(GCSH(IY,IX).GT.GCSL(IY,IX)) GO TO 89003
         IF(GCSL(IY,IX).GE.GCSS(IY,IX).AND.GCSL(IY,IX).GE.GCSE(IY,IX))
     1   GO TO 89001
         IF(GCSE(IY,IX).GE.GCSS(IY,IX).AND.GCSE(IY,IX).GE.GCSL(IY,IX))
     2   GO TO 88999
          IF(GCSE(IY,IX).GE.GCSL(IY,IX).AND.GCSL(IY,IX).GE.0.0)
     3    GO TO 88998
          GCC(IY,IX)=(1.0-(GCSL(IY,IX)/GCSS(IY,IX)))*(GCSL(IY,IX)+
     4                GCSE(IY,IX)+GCSS(IY,IX))
          GO TO 89010
88998     GCC(IY,IX)=(1.0-(GCSE(IY,IX)/GCSS(IY,IX)))*(GCSL(IY,IX)+
     5                GCSE(IY,IX)+GCSS(IY,IX))
          GO TO 89010
88999     IF(GCSS(IY,IX).GE.GCSL(IY,IX).AND.GCSL(IY,IX).GE.0.0)
     6    GO TO 89000
          GCC(IY,IX)=0.
          GSPCO(IY;IX)=(GCSS(IY,IX)/GCSE(IY,IX))*(GCSL(IY,IX)+
     7                  GCSE(IY,IX)+GCSS(IY,IX))
          ISPVAL(IY,IX)=8
C  i.e. Phi = @SEL - corner coeff.
          GO TO 89010
89000     GCC(IY,IX)=0.
          GSECO(IY,IX)=((GCSS(IY,IX)-GCSL(IY,IX))/GCSE(IY,IX))*
     8                  (GCSL(IY,IX)+GCSE(IY,IX)+GCSS(IY,IX))
          ISEVAL(IY,IX)=4
C  i.e. Phi = @SE - edge coeff.
          GO TO 89010
89001     IF(GCSE(IY,IX).GE.GCSS(IY,IX).AND.GCSS(IY,IX).GE.0.0)
     9    GO TO 89002
          GCC(IY,IX)=0.
          GSECO(IY,IX)=((GCSS(IY,IX)-GCSE(IY,IX))/GCSL(IY,IX))*
     1                  (GCSL(IY,IX)+GCSE(IY,IX)+GCSS(IY,IX))
          ISEVAL(IY,IX)=10
C  i.e. Phi = @SL - edge coeff
          GO TO 89010
89002     GCC(IY,IX)=0.
          GSPCO(IY,IX)=(GCSS(IY,IX)/GCSL(IY,IX))*(GCSL(IY,IX)+
     2                  GCSE(IY,IX)+GCSS(IY,IX))
          ISPVAL(IY,IX)=8
C  i.e. Phi = @SEL - corner coeff.
          GO TO 89010
C ***********************************************************************
89003    IF(GCSH(IY,IX).GE.GCSS(IY,IX).AND.GCSH(IY,IX).GE.GCSE(IY,IX))
     1   GO TO 89007
         IF(GCSE(IY,IX).GE.GCSS(IY,IX).AND.GCSE(IY,IX).GE.GCSH(IY,IX))
     2   GO TO 89005
          IF(GCSE(IY,IX).GE.GCSH(IY,IX).AND.GCSH(IY,IX).GE.0.0)
     3    GO TO 89004
          GCC(IY,IX)=(1.0-(GCSH(IY,IX)/GCSS(IY,IX)))*(GCSH(IY,IX)+
     4                GCSE(IY,IX)+GCSS(IY,IX))
          GO TO 89010
89004     GCC(IY,IX)=(1.0-(GCSE(IY,IX)/GCSS(IY,IX)))*(GCSH(IY,IX)+
     5                GCSE(IY,IX)+GCSS(IY,IX))
          GO TO 89010
89005     IF(GCSS(IY,IX).GE.GCSH(IY,IX).AND.GCSH(IY,IX).GE.0.0)
     6    GO TO 89006
          GCC(IY,IX)=0.
          GSPCO(IY,IX)=(GCSS(IY,IX)/GCSE(IY,IX))*(GCSH(IY,IX)+
     7                  GCSE(IY,IX)+GCSS(IY,IX))
          ISPVAL(IY,IX)=4
```

```
C  i.e. Phi = @SEH - corner coeff.
          GO TO 89010
89006     GCC(IY,IX)=0.
          GSECO(IY,IX)=((GCSS(IY,IX)-GCSH(IY,IX))/GCSE(IY,IX))*
     8                  (GCSH(IY,IX)+GCSE(IY,IX)+GCSS(IY,IX))
          ISEVAL(IY,IX)=4
C  i.e. Phi = @SE - edge coeff.
          GO TO 89010
89007     IF(GCSE(IY,IX).GE.GCSS(IY,IX).AND.GCSS(IY,IX).GE.0.0)
     9    GO TO 89008
          GCC(IY,IX)=0.
          GSECO(IY,IX)=((GCSS(IY,IX)-GCSE(IY,IX))/GCSH(IY,IX))*
     1                  (GCSH(IY,IX)+GCSE(IY,IX)+GCSS(IY,IX))
          ISEVAL(IY,IX)=6
C  i.e. Phi = @SH - edge coeff
          GO TO 89010
89008     GCC(IY,IX)=0.
          GSPCO(IY,IX)=(GCSS(IY,IX)/GCSH(IY,IX))*(GCSH(IY,IX)+
     2                 GCSE(IY,IX)+GCSS(IY,IX))
          ISPVAL(IY,IX)=4
C  i.e. Phi = @SEH - corner coeff.
89010 CONTINUE
      CALL SETYX(LD12,GCC,NYDIM,NXDIM)
C      CALL PRN('CCSS',LD12)
      RETURN
C
C    East-West
C
89020 IF(NDIREC.NE.3) GO TO 89100
C   i.e. consider flow through East and West faces.
         CALL GETYX(LD11,GCSE,NYDIM,NXDIM)
         CALL GETYX(LD12,GCSW,NYDIM,NXDIM)
         IF(ISWEEP.EQ.FSWEEP) GO TO 89100
C
C    East Coefficient
C
      DO 89060 IX=1,NX
      DO 89060 IY=1,NY
       IF(GCSE(IY,IX).GT.0.0.AND.GCSW(IY,IX).EQ.0.0) GO TO 89025
89022    GCC(IY,IX)=GCSE(IY,IX)
         GO TO 89060
89025  IF(GCSN(IY,IX).GT.GCSS(IY,IX)) GO TO 89037
        IF(GCSH(IY,IX).GT.GCSL(IY,IX)) GO TO 89031
         IF(GCSL(IY,IX).GE.GCSE(IY,IX).AND.GCSL(IY,IX).GE.GCSS(IY,IX))
     1   GO TO 89029
         IF(GCSS(IY,IX).GE.GCSE(IY,IX).AND.GCSS(IY,IX).GE.GCSL(IY,IX))
     2   GO TO 89027
          IF(GCSS(IY,IX).GE.GCSL(IY,IX).AND.GCSL(IY,IX).GE.0.0)
     3    GO TO 89026
          GCC(IY,IX)=(1.0-(GCSL(IY,IX)/GCSE(IY,IX)))*(GCSL(IY,IX)+
     4               GCSE(IY,IX)+GCSS(IY,IX))
          GO TO 89060
89026     GCC(IY,IX)=(1.0-(GCSS(IY,IX)/GCSE(IY,IX)))*(GCSL(IY,IX)+
     5               GCSE(IY,IX)+GCSS(IY,IX))
          GO TO 89060
89027     IF(GCSE(IY,IX).GE.GCSL(IY,IX).AND.GCSL(IY,IX).GE.0.0)
     6    GO TO 89028
          GCC(IY,IX)=0.
          GSPCO(IY,IX)=(GCSE(IY,IX)/GCSS(IY,IX))*(GCSL(IY,IX)+
     7                 GCSE(IY,IX)+GCSS(IY,IX))
          ISPVAL(IY,IX)=8
C  i.e. Phi = @SEL - corner coeff.
          GO TO 89060
89028     GCC(IY,IX)=0.
          GSECO(IY,IX)=((GCSE(IY,IX)-GCSL(IY,IX))/GCSS(IY,IX))*
     8                  (GCSL(IY,IX)+GCSE(IY,IX)+GCSS(IY,IX))
```

```
          ISEVAL(IY,IX)=4
C  i.e. Phi = @SE - edge coeff.
          GO TO 89060
89029     IF(GCSE(IY,IX).GE.GCSS(IY,IX).AND.GCSS(IY,IX).GE.0.0)
     9    GO TO 89030
          GCC(IY,IX)=0.
          GSPCO(IY,IX)=(GCSE(IY,IX)/GCSL(IY,IX))*(GCSL(IY,IX)+
     2                  GCSE(IY,IX)+GCSS(IY,IX))
          ISPVAL(IY,IX)=8
C  i.e. Phi = @SEL - corner coeff.
          GO TO 89060
89030     GCC(IY,IX)=0.
          GSECO(IY,IX)=((GCSE(IY,IX)-GCSS(IY,IX))/GCSL(IY,IX))*
     1                  (GCSL(IY,IX)+GCSE(IY,IX)+GCSS(IY,IX))
          ISEVAL(IY,IX)=11
C  i.e. Phi = @EL - edge coeff
          GO TO 89060
C **********************************************************************
89031    IF(GCSH(IY,IX).GE.GCSE(IY,IX).AND.GCSH(IY,IX).GE.GCSS(IY,IX))
     1   GO TO 89035
         IF(GCSS(IY,IX).GE.GCSE(IY,IX).AND.GCSS(IY,IX).GE.GCSH(IY,IX))
     2   GO TO 89033
          IF(GCSS(IY,IX).GE.GCSH(IY,IX).AND.GCSH(IY,IX).GE.0.0)
     3    GO TO 89032
          GCC(IY,IX)=(1.0-(GCSH(IY,IX)/GCSE(IY,IX)))*(GCSH(IY,IX)+
     4                GCSE(IY,IX)+GCSS(IY,IX))
89032     GCC(IY,IX)=(1.0-(GCSS(IY,IX)/GCSE(IY,IX)))*(GCSH(IY,IX)+
     5                GCSE(IY,IX)+GCSS(IY,IX))
          GO TO 89060
89033     IF(GCSE(IY,IX).GE.GCSH(IY,IX).AND.GCSH(IY,IX).GE.0.0)
     6    GO TO 89034
          GCC(IY,IX)=0.
          GSPCO(IY,IX)=(GCSE(IY,IX)/GCSS(IY,IX))*(GCSH(IY,IX)+
     7                  GCSE(IY,IX)+GCSS(IY,IX))
          ISPVAL(IY,IX)=4
C  i.e. Phi = @SEH - corner coeff.
          GO TO 89060
89034     GCC(IY,IX)=0.
          GSECO(IY,IX)=((GCSE(IY,IX)-GCSH(IY,IX))/GCSS(IY,IX))*
     8                  (GCSH(IY,IX)+GCSE(IY,IX)+GCSS(IY,IX))
          ISEVAL(IY,IX)=4
C  i.e. Phi = @SE - edge coeff.
          GO TO 89060
89035     IF(GCSE(IY,IX).GE.GCSS(IY,IX).AND.GCSS(IY,IX).GE.0.0)
     9    GO TO 89036
          GCC(IY,IX)=0.
          GSPCO(IY,IX)=(GCSE(IY,IX)/GCSH(IY,IX))*(GCSH(IY,IX)+
     2                  GCSE(IY,IX)+GCSS(IY,IX))
          ISPVAL(IY,IX)=4
C  i.e. Phi = @SEH - corner coeff.
          GO TO 89060
89036     GCC(IY,IX)=0.
          GSECO(IY,IX)=((GCSE(IY,IX)-GCSS(IY,IX))/GCSH(IY,IX))*
     1                  (GCSH(IY,IX)+GCSE(IY,IX)+GCSS(IY,IX))
          ISEVAL(IY,IX)=7
C  i.e. Phi = @EH - edge coeff
          GO TO 89060
C **********************************************************************
89037   IF(GCSH(IY,IX).GT.GCSL(IY,IX)) GO TO 89043
         IF(GCSL(IY,IX).GE.GCSN(IY,IX).AND.GCSL(IY,IX).GE.GCSE(IY,IX))
     1   GO TO 89041
         IF(GCSN(IY,IX).GE.GCSE(IY,IX).AND.GCSN(IY,IX).GE.GCSL(IY,IX))
     2   GO TO 89039
          IF(GCSN(IY,IX).GE.GCSL(IY,IX).AND.GCSL(IY,IX).GE.0.0)
     3    GO TO 89038
          GCC(IY,IX)=(1.0-(GCSL(IY,IX)/GCSE(IY,IX)))*(GCSL(IY,IX)+
```

```
    4               GCSE(IY,IX)+GCSN(IY,IX))
          GO TO 89060
89038     GCC(IY,IX)=(1.0-(GCSN(IY,IX)/GCSE(IY,IX)))*(GCSL(IY,IX)+
    5               GCSE(IY,IX)+GCSN(IY,IX))
          GO TO 89060
89039     IF(GCSE(IY,IX).GE.GCSL(IY,IX).AND.GCSL(IY,IX).GE.0.0)
    6     GO TO 89040
          GCC(IY,IX)=0.
          GSPCO(IY,IX)=(GCSE(IY,IX)/GCSN(IY,IX))*(GCSL(IY,IX)+
    7                 GCSE(IY,IX)+GCSN(IY,IX))
          ISPVAL(IY,IX)=6
C  i.e. Phi = @NEL - corner coeff.
          GO TO 89060
89040     GCC(IY,IX)=0.
          GSECO(IY,IX)=((GCSE(IY,IX)-GCSL(IY,IX))/GCSN(IY,IX))*
    8                 (GCSL(IY,IX)+GCSE(IY,IX)+GCSN(IY,IX))
          ISEVAL(IY,IX)=2
C  i.e. Phi = @NE - edge coeff.
          GO TO 89060
89041     IF(GCSE(IY,IX).GE.GCSN(IY,IX).AND.GCSN(IY,IX).GE.0.0)
    9     GO TO 89042
          GCC(IY,IX)=0.
          GSPCO(IY,IX)=(GCSE(IY,IX)/GCSL(IY,IX))*(GCSL(IY,IX)+
    2                 GCSE(IY,IX)+GCSN(IY,IX))
          ISPVAL(IY,IX)=6
C  i.e. Phi = @NEL - corner coeff.
          GO TO 89060
89042     GCC(IY,IX)=0.
          GSECO(IY,IX)=((GCSE(IY,IX)-GCSN(IY,IX))/GCSL(IY,IX))*
    1                 (GCSL(IY,IX)+GCSE(IY,IX)+GCSN(IY,IX))
          ISEVAL(IY,IX)=11
C  i.e. Phi = @EL - edge coeff
          GO TO 89060
C ***********************************************************************
89043    IF(GCSH(IY,IX).GE.GCSN(IY,IX).AND.GCSH(IY,IX).GE.GCSE(IY,IX))
    1    GO TO 89047
         IF(GCSN(IY,IX).GE.GCSE(IY,IX).AND.GCSN(IY,IX).GE.GCSH(IY,IX))
    2    GO TO 89045
          IF(GCSN(IY,IX).GE.GCSH(IY,IX).AND.GCSH(IY,IX).GE.0.0)
    3     GO TO 89044
          GCC(IY,IX)=(1.0-(GCSH(IY,IX)/GCSE(IY,IX)))*(GCSH(IY,IX)+
    4               GCSE(IY,IX)+GCSN(IY,IX))
          GO TO 89060
89044     GCC(IY,IX)=(1.0-(GCSN(IY,IX)/GCSE(IY,IX)))*(GCSH(IY,IX)+
    5               GCSE(IY,IX)+GCSN(IY,IX))
          GO TO 89060
89045     IF(GCSE(IY,IX).GE.GCSH(IY,IX).AND.GCSH(IY,IX).GE.0.0)
    6     GO TO 89046
          GCC(IY,IX)=0.
          GSPCO(IY,IX)=(GCSE(IY,IX)/GCSN(IY,IX))*(GCSH(IY,IX)+
    7                 GCSE(IY,IX)+GCSN(IY,IX))
          ISPVAL(IY,IX)=2
C  i.e. Phi = @NEH - corner coeff.
          GO TO 89060
89046     GCC(IY,IX)=0.
          GSECO(IY,IX)=((GCSE(IY,IX)-GCSH(IY,IX))/GCSN(IY,IX))*
    8                 (GCSH(IY,IX)+GCSE(IY,IX)+GCSN(IY,IX))
          ISEVAL(IY,IX)=2
C  i.e. Phi = @NE - edge coeff.
          GO TO 89060
89047     IF(GCSE(IY,IX).GE.GCSN(IY,IX).AND.GCSN(IY,IX).GE.0.0)
    9     GO TO 89048
          GSPCO(IY,IX)=(GCSE(IY,IX)/GCSH(IY,IX))*(GCSH(IY,IX)+
    2                 GCSE(IY,IX)+GCSN(IY,IX))
          ISPVAL(IY,IX)=2
C  i.e. Phi = @NEH - corner coeff.
```

```
          GO TO 89060
89048     GCC(IY,IX)=0.
          GSECO(IY,IX)=((GCSE(IY,IX)-GCSN(IY,IX))/GCSH(IY,IX))*
     1                  (GCSH(IY,IX)+GCSE(IY,IX)+GCSN(IY,IX))
          ISEVAL(IY,IX)=7
C  i.e. Phi = @EH - edge coeff.
89060 CONTINUE
      CALL SETYX(LD11,GCC,NYDIM,NXDIM)
C      CALL PRN('CCSE',LD11)
C
C    West Coefficient
C
      DO 89090 IX=1,NX
      DO 89090 IY=1,NY
       IF(GCSW(IY,IX).GT.0.0.AND.GCSE(IY,IX).EQ.0.0) GO TO 89065
89062    GCC(IY,IX)=GCSW(IY,IX)
         GO TO 89090
89065  IF(GCSN(IY,IX).GT.GCSS(IY,IX)) GO TO 89077
        IF(GCSH(IY,IX).GT.GCSL(IY,IX)) GO TO 89071
         IF(GCSL(IY,IX).GE.GCSW(IY,IX).AND.GCSL(IY,IX).GE.GCSS(IY,IX))
     1   GO TO 89069
         IF(GCSS(IY,IX).GE.GCSW(IY,IX).AND.GCSS(IY,IX).GE.GCSL(IY,IX))
     2   GO TO 89067
          IF(GCSS(IY,IX).GE.GCSL(IY,IX).AND.GCSL(IY,IX).GE.0.0)
     3    GO TO 89066
          GCC(IY,IX)=(1.0-(GCSL(IY,IX)/GCSW(IY,IX)))*(GCSL(IY,IX)+
     4               GCSW(IY,IX)+GCSS(IY,IX))
          GO TO 89090
89066     GCC(IY,IX)=(1.0-(GCSS(IY,IX)/GCSW(IY,IX)))*(GCSL(IY,IX)+
     5               GCSW(IY,IX)+GCSS(IY,IX))
          GO TO 89090
89067     IF(GCSW(IY,IX).GE.GCSL(IY,IX).AND.GCSL(IY,IX).GE.0.0)
     6    GO TO 89068
          GCC(IY,IX)=0.
          GSPCO(IY,IX)=(GCSW(IY,IX)/GCSS(IY,IX))*(GCSL(IY,IX)+
     7                 GCSW(IY,IX)+GCSS(IY,IX))
          ISPVAL(IY,IX)=7
C  i.e. Phi = @SWL - corner coeff.
          GO TO 89090
89068     GCC(IY,IX)=0.
          GSECO(IY;IX)=((GCSW(IY,IX)-GCSL(IY,IX))/GCSS(IY,IX))*
     8                  (GCSL(IY,IX)+GCSW(IY,IX)+GCSS(IY,IX))
          ISEVAL(IY,IX)=3
C  i.e. Phi = @SW - edge coeff.
          GO TO 89090
89069     IF(GCSW(IY,IX).GE.GCSS(IY,IX).AND.GCSS(IY,IX).GE.0.0)
     9    GO TO 89070
          GCC(IY,IX)=0.
          GSPCO(IY,IX)=(GCSW(IY,IX)/GCSL(IY,IX))*(GCSL(IY,IX)+
     2                 GCSW(IY,IX)+GCSS(IY,IX))
          ISPVAL(IY,IX)=7
C  i.e. Phi = @SWL - corner coeff.
          GO TO 89090
89070     GCC(IY,IX)=0.
          GSECO(IY,IX)=((GCSW(IY,IX)-GCSS(IY,IX))/GCSL(IY,IX))*
     1                  (GCSL(IY,IX)+GCSW(IY,IX)+GCSS(IY,IX))
          ISEVAL(IY,IX)=12
C  i.e. Phi = @WL - edge coeff
          GO TO 89090
C **********************************************************************
89071    IF(GCSH(IY,IX).GE.GCSW(IY,IX).AND.GCSH(IY,IX).GE.GCSS(IY,IX))
     1   GO TO 89075
         IF(GCSS(IY,IX).GE.GCSW(IY,IX).AND.GCSS(IY,IX).GE.GCSH(IY,IX))
     2   GO TO 89073
          IF(GCSS(IY,IX).GE.GCSH(IY,IX).AND.GCSH(IY,IX).GE.0.0)
     3    GO TO 89072
```

```
          GCC(IY,IX)=(1.0-(GCSH(IY,IX)/GCSW(IY,IX)))*(GCSH(IY,IX)+
     4               GCSW(IY,IX)+GCSS(IY,IX))
          GO TO 89090
89072     GCC(IY,IX)=(1.0-(GCSS(IY,IX)/GCSW(IY,IX)))*(GCSH(IY,IX)+
     5               GCSW(IY,IX)+GCSS(IY,IX))
          GO TO 89090
89073     IF(GCSW(IY,IX).GE.GCSH(IY,IX).AND.GCSH(IY,IX).GE.0.0)
     6    GO TO 89074
          GCC(IY,IX)=0.
          GSPCO(IY,IX)=(GCSW(IY,IX)/GCSS(IY,IX))*(GCSH(IY,IX)+
     7                 GCSW(IY,IX)+GCSS(IY,IX))
          ISPVAL(IY,IX)=3
C  i.e. Phi = @SWH - corner coeff.
          GO TO 89090
89074     GCC(IY,IX)=0.
          GSECO(IY,IX)=((GCSW(IY,IX)-GCSH(IY,IX))/GCSS(IY,IX))*
     8                 (GCSH(IY,IX)+GCSW(IY,IX)+GCSS(IY,IX))
          ISEVAL(IY,IX)=3
C  i.e. Phi = @SW - edge coeff.
          GO TO 89090
89075     IF(GCSW(IY,IX).GE.GCSS(IY,IX).AND.GCSS(IY,IX).GE.0.0)
     9    GO TO 89076
          GCC(IY,IX)=0.
          GSPCO(IY,IX)=(GCSW(IY,IX)/GCSH(IY,IX))*(GCSH(IY,IX)+
     2                 GCSW(IY,IX)+GCSS(IY,IX))
          ISPVAL(IY,IX)=3
C  i.e. Phi = @SWH - corner coeff.
          GO TO 89090
89076     GCC(IY,IX)=0.
          GSECO(IY,IX)=((GCSW(IY,IX)-GCSS(IY,IX))/GCSH(IY,IX))*
     1                 (GCSH(IY,IX)+GCSW(IY,IX)+GCSS(IY,IX))
          ISEVAL(IY,IX)=8
C  i.e. Phi = @WH - edge coeff
          GO TO 89090
C ***********************************************************************
89077   IF(GCSH(IY,IX).GT.GCSL(IY,IX)) GO TO 89083
         IF(GCSL(IY,IX).GE.GCSN(IY,IX).AND.GCSL(IY,IX).GE.GCSW(IY,IX))
     1   GO TO 89081
         IF(GCSN(IY,IX).GE.GCSW(IY,IX).AND.GCSN(IY,IX).GE.GCSL(IY,IX))
     2   GO TO 89079
          IF(GCSN(IY,IX).GE.GCSL(IY,IX).AND.GCSL(IY,IX).GE.0.0)
     3    GO TO 89078
          GCC(IY,IX)=(1.0-(GCSL(IY,IX)/GCSW(IY,IX)))*(GCSL(IY,IX)+
     4               GCSW(IY,IX)+GCSN(IY,IX))
          GO TO 89090
89078     GCC(IY,IX)=(1.0-(GCSN(IY,IX)/GCSW(IY,IX)))*(GCSL(IY,IX)+
     5               GCSW(IY,IX)+GCSN(IY,IX))
          GO TO 89090
89079     IF(GCSW(IY,IX).GE.GCSL(IY,IX).AND.GCSL(IY,IX).GE.0.0)
     6    GO TO 89080
          GCC(IY,IX)=0.
          GSPCO(IY,IX)=(GCSW(IY,IX)/GCSN(IY,IX))*(GCSL(IY,IX)+
     7                 GCSW(IY,IX)+GCSN(IY,IX))
          ISPVAL(IY,IX)=5
C  i.e. Phi = @NWL - corner coeff.
          GO TO 89090
89080     GCC(IY,IX)=0.
          GSECO(IY,IX)=((GCSW(IY,IX)-GCSL(IY,IX))/GCSN(IY,IX))*
     8                 (GCSL(IY,IX)+GCSW(IY,IX)+GCSN(IY,IX))
          ISEVAL(IY,IX)=1
C  i.e. Phi = @NW - edge coeff.
          GO TO 89090
89081     IF(GCSW(IY,IX).GE.GCSN(IY,IX).AND.GCSN(IY,IX).GE.0.0)
     9    GO TO 89082
          GCC(IY,IX)=0.
          GSPCO(IY,IX)=(GCSW(IY,IX)/GCSL(IY,IX))*(GCSL(IY,IX)+
```

```
     2                      GCSW(IY,IX)+GCSN(IY,IX))
          ISPVAL(IY,IX)=5
C  i.e. Phi = @NWL - corner coeff.
          GO TO 89090
89082     GCC(IY,IX)=0.
          GSECO(IY,IX)=((GCSW(IY,IX)-GCSN(IY,IX))/GCSL(IY,IX))*
     1                   (GCSL(IY,IX)+GCSW(IY,IX)+GCSN(IY,IX))
          ISEVAL(IY,IX)=12
C  i.e. Phi = @WL - edge coeff
          GO TO 89090
C ***********************************************************************
89083    IF(GCSH(IY,IX).GE.GCSN(IY,IX).AND.GCSH(IY,IX).GE.GCSW(IY,IX))
     1   GO TO 89087
         IF(GCSN(IY,IX).GE.GCSW(IY,IX).AND.GCSN(IY,IX).GE.GCSH(IY,IX))
     2   GO TO 89085
          IF(GCSN(IY,IX).GE.GCSH(IY,IX).AND.GCSH(IY,IX).GE.0.0)
     3    GO TO 89084
          GCC(IY,IX)=(1.0-(GCSH(IY,IX)/GCSW(IY,IX)))*(GCSH(IY,IX)+
     4                GCSW(IY,IX)+GCSN(IY,IX))
          GO TO 89090
89084     GCC(IY,IX)=(1.0-(GCSN(IY,IX)/GCSW(IY,IX)))*(GCSH(IY,IX)+
     5                GCSW(IY,IX)+GCSN(IY,IX))
          GO TO 89090
89085     IF(GCSW(IY,IX).GE.GCSH(IY,IX).AND.GCSH(IY,IX).GE.0.0)
     6    GO TO 89086
          GCC(IY,IX)=0.
          GSPCO(IY,IX)=(GCSW(IY,IX)/GCSN(IY,IX))*(GCSH(IY,IX)+
     7                  GCSW(IY,IX)+GCSN(IY,IX))
          ISPVAL(IY,IX)=1
C  i.e. Phi = @NWH - corner coeff.
          GO TO 89090
89086     GCC(IY,IX)=0.
          GSECO(IY,IX)=((GCSW(IY,IX)-GCSH(IY,IX))/GCSN(IY,IX))*
     8                  (GCSH(IY,IX)+GCSW(IY,IX)+GCSN(IY,IX))
          ISEVAL(IY,IX)=1
C  i.e. Phi = @NW - edge coeff.
          GO TO 89090
89087     IF(GCSW(IY,IX).GE.GCSN(IY,IX).AND.GCSN(IY,IX).GE.0.0)
     9    GO TO 89088
          GCC(IY,IX)=0.
          GSPCO(IY,IX)=(GCSW(IY,IX)/GCSH(IY,IX))*(GCSH(IY,IX)+
     2                  GCSW(IY,IX)+GCSN(IY,IX))
          ISPVAL(IY,IX)=1
C  i.e. Phi = @NWH - corner coeff.
          GO TO 89090
89088     GCC(IY,IX)=0.
          GSECO(IY,IX)=((GCSW(IY,IX)-GCSN(IY,IX))/GCSH(IY,IX))*
     1                  (GCSH(IY,IX)+GCSW(IY,IX)+GCSN(IY,IX))
          ISEVAL(IY,IX)=8
C  i.e. Phi = @WH - edge coeff.
89090 CONTINUE
      CALL SETYX(LD12,GCC,NYDIM,NXDIM)
C      CALL PRN('CCSW',LD12)
      RETURN
C
C     High-Low
C
89100 IF(NDIREC.NE.5) GO TO 89140
C   i.e. consider flow through High face.
         CALL GETYX(LD2,GCSH,NYDIM,NXDIM)
         IF(ISWEEP.EQ.FSWEEP) GO TO 89140
C     High Coefficient
C
      DO 89130 IX=1,NX
      DO 89130 IY=1,NY
       IF(GCSH(IY,IX).GT.0.0.AND.GCSL(IY,IX).EQ.0.0) GO TO 89105
```

```
         GCC(IY,IX)=GCSH(IY,IX)
         GO TO 89130
89105   IF(GCSN(IY,IX).GT.GCSS(IY,IX)) GO TO 89117
         IF(GCSE(IY,IX).GT.GCSW(IY,IX)) GO TO 89111
         IF(GCSS(IY,IX).GE.GCSH(IY,IX).AND.GCSS(IY,IX).GE.GCSW(IY,IX))
     1   GO TO 89109
         IF(GCSW(IY,IX).GE.GCSH(IY,IX).AND.GCSW(IY,IX).GE.GCSS(IY,IX))
     2   GO TO 89107
          IF(GCSW(IY,IX).GE.GCSS(IY,IX).AND.GCSS(IY,IX).GE.0.0)
     3    GO TO 89106
          GCC(IY,IX)=(1.0-(GCSS(IY,IX)/GCSH(IY,IX)))*(GCSH(IY,IX)+
     4               GCSW(IY,IX)+GCSS(IY,IX))
          GO TO 89130
89106     GCC(IY,IX)=(1.0-(GCSW(IY,IX)/GCSH(IY,IX)))*(GCSH(IY,IX)+
     5               GCSW(IY,IX)+GCSS(IY,IX))
          GO TO 89130
89107     IF(GCSS(IY,IX).GE.GCSH(IY,IX).AND.GCSH(IY,IX).GE.0.0)
     9    GO TO 89108
          GCC(IY,IX)=0.
          GSECO(IY,IX)=((GCSH(IY,IX)-GCSS(IY,IX))/GCSW(IY,IX))*
     1                 (GCSH(IY,IX)+GCSW(IY,IX)+GCSS(IY,IX))
          ISEVAL(IY,IX)=8
C  i.e. Phi = @WH - edge coeff
          GO TO 89130
89108     GCC(IY,IX)=0.
          GSPCO(IY,IX)=(GCSH(IY,IX)/GCSW(IY,IX))*(GCSH(IY,IX)+
     2                 GCSW(IY,IX)+GCSS(IY,IX))
          ISPVAL(IY,IX)=3
C  i.e. Phi = @SWH - corner coeff.
          GO TO 89130
89109     IF(GCSW(IY,IX).GE.GCSH(IY,IX).AND.GCSH(IY,IX).GE.0.0)
     9    GO TO 89110
          GCC(IY,IX)=0.
          GSECO(IY,IX)=((GCSH(IY,IX)-GCSW(IY,IX))/GCSS(IY,IX))*
     1                 (GCSH(IY,IX)+GCSW(IY,IX)+GCSS(IY,IX))
          ISEVAL(IY,IX)=6
C  i.e. Phi = @SH - edge coeff
          GO TO 89130
89110     GCC(IY,IX)=0.
          GSPCO(IY,IX)=(GCSH(IY,IX)/GCSS(IY,IX))*(GCSH(IY,IX)+
     2                 GCSW(IY,IX)+GCSS(IY,IX))
          ISPVAL(IY,IX)=3
C  i.e. Phi = @SWH - corner coeff.
          GO TO 89130
C *********************************************************************
89111    IF(GCSS(IY,IX).GE.GCSH(IY,IX).AND.GCSS(IY,IX).GE.GCSE(IY,IX))
     1   GO TO 89115
         IF(GCSE(IY,IX).GE.GCSH(IY,IX).AND.GCSE(IY,IX).GE.GCSS(IY,IX))
     2   GO TO 89113
          IF(GCSE(IY,IX).GE.GCSS(IY,IX).AND.GCSS(IY,IX).GE.0.0)
     3    GO TO 89112
          GCC(IY,IX)=(1.0-(GCSS(IY,IX)/GCSH(IY,IX)))*(GCSH(IY,IX)+
     4               GCSE(IY,IX)+GCSS(IY,IX))
          GO TO 89130
89112     GCC(IY,IX)=(1.0-(GCSE(IY,IX)/GCSH(IY,IX)))*(GCSH(IY,IX)+
     5               GCSE(IY,IX)+GCSS(IY,IX))
          GO TO 89130
89113     IF(GCSS(IY,IX).GE.GCSH(IY,IX).AND.GCSH(IY,IX).GE.0.0)
     9    GO TO 89114
          GCC(IY,IX)=0.
          GSECO(IY,IX)=((GCSH(IY,IX)-GCSS(IY,IX))/GCSE(IY,IX))*
     1                 (GCSH(IY,IX)+GCSE(IY,IX)+GCSS(IY,IX))
          ISEVAL(IY,IX)=7
C  i.e. Phi = @EH - edge coeff
          GO TO 89130
89114     GCC(IY,IX)=0.
```

```
      GSPCO(IY,IX)=(GCSH(IY,IX)/GCSE(IY,IX))*(GCSH(IY,IX)+
     2             GCSE(IY,IX)+GCSS(IY,IX))
      ISPVAL(IY,IX)=4
C  i.e. Phi = @SEH - corner coeff.
      GO TO 89130
89115 IF(GCSE(IY,IX).GE.GCSH(IY,IX).AND.GCSH(IY,IX).GE.0.0)
     9 GO TO 89116
      GCC(IY,IX)=0.
      GSECO(IY,IX)=((GCSH(IY,IX)-GCSE(IY,IX))/GCSS(IY,IX))*
     1              (GCSH(IY,IX)+GCSE(IY,IX)+GCSS(IY,IX))
      ISEVAL(IY,IX)=6
C  i.e. Phi = @SH - edge coeff
      GO TO 89130
89116 GCC(IY,IX)=0.
      GSPCO(IY,IX)=(GCSH(IY,IX)/GCSS(IY,IX))*(GCSH(IY,IX)+
     2             GCSE(IY,IX)+GCSS(IY,IX))
      ISPVAL(IY,IX)=4
C  i.e. Phi = @SEH - corner coeff.
      GO TO 89130
C ************************************************************************
89117 IF(GCSE(IY,IX).GT.GCSW(IY,IX)) GO TO 89123
      IF(GCSN(IY,IX).GE.GCSH(IY,IX).AND.GCSN(IY,IX).GE.GCSW(IY,IX))
     1 GO TO 89121
      IF(GCSW(IY,IX).GE.GCSH(IY,IX).AND.GCSW(IY,IX).GE.GCSN(IY,IX))
     2 GO TO 89119
      IF(GCSW(IY,IX).GE.GCSN(IY,IX).AND.GCSN(IY,IX).GE.0.0)
     3 GO TO 89118
      GCC(IY,IX)=(1.0-(GCSN(IY,IX)/GCSH(IY,IX)))*(GCSH(IY,IX)+
     4            GCSW(IY,IX)+GCSN(IY,IX))
      GO TO 89130
89118 GCC(IY,IX)=(1.0-(GCSW(IY,IX)/GCSH(IY,IX)))*(GCSH(IY,IX)+
     5            GCSW(IY,IX)+GCSN(IY,IX))
      GO TO 89130
89119 IF(GCSN(IY,IX).GE.GCSH(IY,IX).AND.GCSH(IY,IX).GE.0.0)
     9 GO TO 89120
      GCC(IY,IX)=0.
      GSECO(IY,IX)=((GCSH(IY,IX)-GCSN(IY,IX))/GCSW(IY,IX))*
     1              (GCSH(IY,IX)+GCSW(IY,IX)+GCSN(IY,IX))
      ISEVAL(IY,IX)=8
C  i.e. Phi = @WH - edge coeff
      GO TO 89130
89120 GCC(IY,IX)=0.
      GSPCO(IY,IX)=(GCSH(IY,IX)/GCSW(IY,IX))*(GCSH(IY,IX)+
     2             GCSW(IY,IX)+GCSN(IY,IX))
      ISPVAL(IY,IX)=1
C  i.e. Phi = @NWH - corner coeff.
      GO TO 89130
89121 IF(GCSW(IY,IX).GE.GCSH(IY,IX).AND.GCSH(IY,IX).GE.0.0)
     9 GO TO 89122
      GCC(IY,IX)=0.
      GSECO(IY,IX)=((GCSH(IY,IX)-GCSW(IY,IX))/GCSN(IY,IX))*
     1              (GCSH(IY,IX)+GCSW(IY,IX)+GCSN(IY,IX))
      ISEVAL(IY,IX)=5
C  i.e. Phi = @NH - edge coeff
      GO TO 89130
89122 GCC(IY,IX)=0.
      GSPCO(IY,IX)=(GCSW(IY,IX)/GCSN(IY,IX))*(GCSH(IY,IX)+
     2             GCSW(IY,IX)+GCSN(IY,IX))
      ISPVAL(IY,IX)=1
C  i.e. Phi = @NWH - corner coeff.
      GO TO 89130
C ************************************************************************
89123 IF(GCSN(IY,IX).GE.GCSH(IY,IX).AND.GCSN(IY,IX).GE.GCSE(IY,IX))
     1 GO TO 89127
      IF(GCSE(IY,IX).GE.GCSH(IY,IX).AND.GCSE(IY,IX).GE.GCSN(IY,IX))
     2 GO TO 89125
```

```
          IF(GCSE(IY,IX).GE.GCSN(IY,IX).AND.GCSN(IY,IX).GE.0.0)
     3    GO TO 89124
          GCC(IY,IX)=(1.0-(GCSN(IY,IX)/GCSH(IY,IX)))*(GCSH(IY,IX)+
     4              GCSE(IY,IX)+GCSN(IY,IX))
          GO TO 89130
89124     GCC(IY,IX)=(1.0-(GCSE(IY,IX)/GCSH(IY,IX)))*(GCSH(IY,IX)+
     5              GCSE(IY,IX)+GCSN(IY,IX))
          GO TO 89130
89125     IF(GCSN(IY,IX).GE.GCSH(IY,IX).AND.GCSH(IY,IX).GE.0.0)
     9    GO TO 89126
          GCC(IY,IX)=0.
          GSECO(IY,IX)=((GCSH(IY,IX)-GCSN(IY,IX))/GCSE(IY,IX))*
     1                (GCSH(IY,IX)+GCSE(IY,IX)+GCSN(IY,IX))
          ISEVAL(IY,IX)=7
C  i.e. Phi = @EH - edge coeff
          GO TO 89130
89126     GCC(IY,IX)=0.
          GSPCO(IY,IX)=(GCSH(IY,IX)/GCSE(IY,IX))*(GCSH(IY,IX)+
     2                GCSE(IY,IX)+GCSN(IY,IX))
          ISPVAL(IY,IX)=2
C  i.e. Phi = @NEH - corner coeff.
          GO TO 89130
89127     IF(GCSE(IY,IX).GE.GCSH(IY,IX).AND.GCSH(IY,IX).GE.0.0)
     9    GO TO 89128
          GCC(IY,IX)=0.
          GSECO(IY,IX)=((GCSH(IY,IX)-GCSE(IY,IX))/GCSN(IY,IX))*
     1                (GCSH(IY,IX)+GCSE(IY,IX)+GCSN(IY,IX))
          ISEVAL(IY,IX)=5
C  i.e. Phi = @NH - edge coeff
          GO TO 89130
89128     GCC(IY,IX)=0.
          GSPCO(IY,IX)=(GCSH(IY,IX)/GCSN(IY,IX))*(GCSH(IY,IX)+
     2                GCSE(IY,IX)+GCSN(IY,IX))
          ISPVAL(IY,IX)=2
C  i.e. Phi = @NEH - corner coeff.
89130 CONTINUE
      CALL SETYX(LD2,GCC,NYDIM,NXDIM)
C      CALL PRN('CCSH',LD2)
      RETURN
C
C   Low Coefficient
C
89140 IF(NDIREC.NE.6) RETURN
C   i.e. consider flow through Low face.
         CALL GETYX(LD2,GCSL,NYDIM,NXDIM)
         IF(ISWEEP.EQ.FSWEEP) RETURN
      DO 89200 IX=1,NX
      DO 89200 IY=1,NY
       IF(GCSL(IY,IX).GT.0.0.AND.GCSH(IY,IX).EQ.0.0) GO TO 89145
         GCC(IY,IX)=GCSL(IY,IX)
         GO TO 89200
89145  IF(GCSN(IY,IX).GT.GCSS(IY,IX)) GO TO 89157
        IF(GCSE(IY,IX).GT.GCSW(IY,IX)) GO TO 89151
         IF(GCSS(IY,IX).GE.GCSL(IY,IX).AND.GCSS(IY,IX).GE.GCSW(IY,IX))
     1   GO TO 89149
         IF(GCSW(IY,IX).GE.GCSL(IY,IX).AND.GCSW(IY,IX).GE.GCSS(IY,IX))
     2   GO TO 89147
          IF(GCSW(IY,IX).GE.GCSS(IY,IX).AND.GCSS(IY,IX).GE.0.0)
     3    GO TO 89146
          GCC(IY,IX)=(1.0-(GCSS(IY,IX)/GCSL(IY,IX)))*(GCSL(IY,IX)+
     4              GCSW(IY,IX)+GCSS(IY,IX))
          GO TO 89200
89146     GCC(IY,IX)=(1.0-(GCSW(IY,IX)/GCSL(IY,IX)))*(GCSL(IY,IX)+
     5              GCSW(IY,IX)+GCSS(IY,IX))
          GO TO 89200
89147     IF(GCSS(IY,IX).GE.GCSL(IY,IX).AND.GCSL(IY,IX).GE.0.0)
```

```
     9    GO TO 89148
          GCC(IY,IX)=0.
          GSECO(IY,IX)=((GCSL(IY,IX)-GCSS(IY,IX))/GCSW(IY,IX))*
     1                 (GCSL(IY,IX)+GCSW(IY,IX)+GCSS(IY,IX))
          ISEVAL(IY,IX)=12
C  i.e. Phi = @WL - edge coeff
          GO TO 89200
89148     GCC(IY,IX)=0.
          GSPCO(IY,IX)=(GCSL(IY,IX)/GCSW(IY,IX))*(GCSL(IY,IX)+
     2                 GCSW(IY,IX)+GCSS(IY,IX))
          ISPVAL(IY,IX)=7
C  i.e. Phi = @SWL - corner coeff.
          GO TO 89200
89149     IF(GCSW(IY,IX).GE.GCSL(IY,IX).AND.GCSL(IY,IX).GE.0.0)
     9    GO TO 89150
          GCC(IY,IX)=0.
          GSECO(IY,IX)=((GCSL(IY,IX)-GCSW(IY,IX))/GCSS(IY,IX))*
     1                 (GCSL(IY,IX)+GCSW(IY,IX)+GCSS(IY,IX))
          ISEVAL(IY,IX)=10
C  i.e. Phi = @SL - edge coeff
          GO TO 89200
89150     GCC(IY,IX)=0.
          GSPCO(IY,IX)=(GCSL(IY,IX)/GCSS(IY,IX))*(GCSL(IY,IX)+
     2                 GCSW(IY,IX)+GCSS(IY,IX))
          ISPVAL(IY,IX)=7
C  i.e. Phi = @SWL - corner coeff.
          GO TO 89200
C **********************************************************************
89151    IF(GCSS(IY,IX).GE.GCSL(IY,IX).AND.GCSS(IY,IX).GE.GCSE(IY,IX))
     1   GO TO 89155
         IF(GCSE(IY,IX).GE.GCSL(IY,IX).AND.GCSE(IY,IX).GE.GCSS(IY,IX))
     2   GO TO 89153
          IF(GCSE(IY,IX).GE.GCSS(IY,IX).AND.GCSS(IY,IX).GE.0.0)
     3    GO TO 89152
          GCC(IY,IX)=(1.0-(GCSS(IY,IX)/GCSL(IY,IX)))*(GCSL(IY,IX)+
     4               GCSE(IY,IX)+GCSS(IY,IX))
          GO TO 89200
89152     GCC(IY,IX)=(1.0-(GCSE(IY,IX)/GCSL(IY,IX)))*(GCSL(IY,IX)+
     5               GCSE(IY,IX)+GCSS(IY,IX))
          GO TO 89200
89153     IF(GCSS(IY,IX).GE.GCSL(IY,IX).AND.GCSL(IY,IX).GE.0.0)
     9    GO TO 89154
          GCC(IY,IX)=0.
          GSECO(IY,IX)=((GCSL(IY,IX)-GCSS(IY,IX))/GCSE(IY,IX))*
     1                 (GCSL(IY,IX)+GCSE(IY,IX)+GCSS(IY,IX))
          ISEVAL(IY,IX)=11
C  i.e. Phi = @EL - edge coeff
          GO TO 89200
89154     GCC(IY,IX)=0.
          GSPCO(IY,IX)=(GCSL(IY,IX)/GCSE(IY,IX))*(GCSL(IY,IX)+
     2                 GCSE(IY,IX)+GCSS(IY,IX))
          ISPVAL(IY,IX)=8
C  i.e. Phi = @SEL - corner coeff.
          GO TO 89200
89155     IF(GCSE(IY,IX).GE.GCSL(IY,IX).AND.GCSL(IY,IX).GE.0.0)
     9    GO TO 89156
          GCC(IY,IX)=0.
          GSECO(IY,IX)=((GCSL(IY,IX)-GCSE(IY,IX))/GCSS(IY,IX))*
     1                 (GCSL(IY,IX)+GCSE(IY,IX)+GCSS(IY,IX))
          ISEVAL(IY,IX)=10
C  i.e. Phi = @SL - edge coeff
          GO TO 89200
89156     GCC(IY,IX)=0.
          GSPCO(IY,IX)=(GCSL(IY,IX)/GCSS(IY,IX))*(GCSL(IY,IX)+
     2                 GCSE(IY,IX)+GCSS(IY,IX))
          ISPVAL(IY,IX)=8
```

```
C  i.e. Phi = @SEL - corner coeff.
        GO TO 89200
C ***********************************************************************
89157  IF(GCSE(IY,IX).GT.GCSW(IY,IX)) GO TO 89163
       IF(GCSN(IY,IX).GE.GCSL(IY,IX).AND.GCSN(IY,IX).GE.GCSW(IY,IX))
     1 GO TO 89161
       IF(GCSW(IY,IX).GE.GCSL(IY,IX).AND.GCSW(IY,IX).GE.GCSN(IY,IX))
     2 GO TO 89159
        IF(GCSW(IY,IX).GE.GCSN(IY,IX).AND.GCSN(IY,IX).GE.0.0)
     3  GO TO 89158
        GCC(IY,IX)=(1.0-(GCSN(IY,IX)/GCSL(IY,IX)))*(GCSL(IY,IX)+
     4             GCSW(IY,IX)+GCSN(IY,IX))
        GO TO 89200
89158   GCC(IY,IX)=(1.0-(GCSW(IY,IX)/GCSL(IY,IX)))*(GCSL(IY,IX)+
     5             GCSW(IY,IX)+GCSN(IY,IX))
        GO TO 89200
89159   IF(GCSN(IY,IX).GE.GCSL(IY,IX).AND.GCSL(IY,IX).GE.0.0)
     9  GO TO 89160
        GCC(IY,IX)=0.
        GSECO(IY,IX)=((GCSL(IY,IX)-GCSN(IY,IX))/GCSW(IY,IX))*
     1               (GCSL(IY,IX)+GCSW(IY,IX)+GCSN(IY,IX))
        ISEVAL(IY,IX)=12
C  i.e. Phi = @WL - edge coeff
        GO TO 89200
89160   GCC(IY,IX)=0.
        GSPCO(IY,IX)=(GCSL(IY,IX)/GCSW(IY,IX))*(GCSL(IY,IX)+
     2               GCSW(IY,IX)+GCSN(IY,IX))
        ISPVAL(IY,IX)=5
C  i.e. Phi = @NWL - corner coeff.
        GO TO 89200
89161   IF(GCSW(IY,IX).GE.GCSL(IY,IX).AND.GCSL(IY,IX).GE.0.0)
     9  GO TO 89162
        GCC(IY,IX)=0.
        GSECO(IY,IX)=((GCSL(IY,IX)-GCSW(IY,IX))/GCSN(IY,IX))*
     1               (GCSL(IY,IX)+GCSW(IY,IX)+GCSN(IY,IX))
        ISEVAL(IY,IX)=9
C  i.e. Phi = @NL - edge coeff
        GO TO 89200
89162   GCC(IY,IX)=0.
        GSPCO(IY,IX)=(GCSL(IY,IX)/GCSN(IY,IX))*(GCSL(IY,IX)+
     2               GCSW(IY,IX)+GCSN(IY,IX))
        ISPVAL(IY,IX)=5
C  i.e. Phi = @NWL - corner coeff.
        GO TO 89200
C ***********************************************************************
89163  IF(GCSN(IY,IX).GE.GCSL(IY,IX).AND.GCSN(IY,IX).GE.GCSE(IY,IX))
     1 GO TO 89167
       IF(GCSE(IY,IX).GE.GCSL(IY,IX).AND.GCSE(IY,IX).GE.GCSN(IY,IX))
     2 GO TO 89165
        IF(GCSE(IY,IX).GE.GCSN(IY,IX).AND.GCSN(IY,IX).GE.0.0)
     3  GO TO 89164
        GCC(IY,IX)=(1.0-(GCSN(IY,IX)/GCSL(IY,IX)))*(GCSL(IY,IX)+
     4             GCSE(IY,IX)+GCSN(IY,IX))
        GO TO 89200
89164   GCC(IY,IX)=(1.0-(GCSE(IY,IX)/GCSL(IY,IX)))*(GCSL(IY,IX)+
     5             GCSE(IY,IX)+GCSN(IY,IX))
        GO TO 89200
89165   IF(GCSN(IY,IX).GE.GCSL(IY,IX).AND.GCSL(IY,IX).GE.0.0)
     9  GO TO 89166
        GCC(IY,IX)=0.
        GSECO(IY,IX)=((GCSL(IY,IX)-GCSN(IY,IX))/GCSE(IY,IX))*
     1               (GCSL(IY,IX)+GCSE(IY,IX)+GCSN(IY,IX))
        ISEVAL(IY,IX)=11
C  i.e. Phi = @EL - edge coeff
        GO TO 89200
89166   GCC(IY,IX)=0.
```

```
          GSPCO(IY,IX)=(GCSL(IY,IX)/GCSE(IY,IX))*(GCSL(IY,IX)+
     2               .    GCSE(IY,IX)+GCSN(IY,IX))
          ISPVAL(IY,IX)=6
C  i.e. Phi = @NEL - corner coeff.
          GO TO 89200
89167     IF(GCSE(IY,IX).GE.GCSL(IY,IX).AND.GCSL(IY,IX).GE.0.0)
     9    GO TO 89168
          GCC(IY,IX)=0.
          GSECO(IY,IX)=((GCSL(IY,IX)-GCSE(IY,IX))/GCSN(IY,IX))*
     1                    (GCSL(IY,IX)+GCSE(IY,IX)+GCSN(IY,IX))
          ISEVAL(IY,IX)=9
C  i.e. Phi = @NL - edge coeff
          GO TO 89200
89168     GCC(IY,IX)=0.
          GSPCO(IY,IX)=(GCSL(IY,IX)/GCSN(IY,IX))*(GCSL(IY,IX)+
     2                    GCSE(IY,IX)+GCSN(IY,IX))
          ISPVAL(IY,IX)=6
C  i.e. Phi = @NEL - corner coeff.
89200 CONTINUE
      CALL SETYX(LD2,GCC,NYDIM,NXDIM)
C       CALL PRN('CCSL',LD2)
      RETURN
      END
```

REFERENCES

1. KARMAN, T.von. *Gottinger Nachrichten*, pp.547-556, 1912.
2. WHITE, F.W.L and KENNEDY, A. Design philosophy and performance of a vortex shedding flow meter installation used in the offshore mass measurement of NGL's. *Proceedings of the International Conference on the Metering of Petroleum*. Oyez Scientific and Technical Services Ltd., London, March, 1985.
3. COUSINS, T. Vortex shedding detection using ultrasound. As ref. 2.
4. MOTTRAM, R.C. The measurement of pulsating flow. *Conference on the Basic Principles and Practice of Flow Measurement*. National Engineering Laboratory, East Kilbride, Glasgow, 1986.
5. MOTTRAM, R.C. and ROBATI, B. The effect of pulsation on vortex flowmeters. As ref. 2.
6. PHOENICS Version 1.6, CHAM Ltd., Wimbledon, London, 1992.
7. STROUHAL, V. Uber eine besondre Art der Tonnerregung, Annalen der Physik und Chemie, *Neu Folge*, Vol. 5, pp. 126-251, 1878.
8. MALLOCK, A. *Proc. Roy, Soc. A*, Vol. 79, pp. 262-265, 1907.
9. BENARD, H. *Comptes Rendus*, Vol. 147, pp. 970-972, 1908.
10. KARMAN, T.von. and RUBACH, H.L. *Physik. Zeitschr.* Vol. 13, pp. 49-59, 1912.
11. *Modern Developments in Fluid Dynamics*, ed. S.GOLDSTEIN. An Account of the Theory and Experiment Relating to Boundary Layers, Turbulent Motion and Wakes. Vol. II, pp. 557, The Clarendon Press, Oxford, 1938.

12. FAGE, A., and JOHANSEN, F.C. *Phil. Mag. (7)*, Vol. 5, pp. 417-441, 1928.

13. SCHILLER, L. and LINKE, W. *Zeitschr.f. Flugtechn. u. Motorluftschiffart*, Vol. 24, pp. 193-198, 1933.

14. As ref. 11. pp. 556-557.

15. RAYLEIGH, Lord. *Phil. Mag. (6)*, Vol. 29, pp. 433-533, 1915.

16. TYLER, E. *Phil. Mag. (7)*, Vol. 11, pp. 849-890, 1931.

17. ROSENHEAD, L. and SCHWABE, M. *Proc. Roy. Soc. A*, Vol. 129, pp.115-135, 1930.

18. KIRCHOFF, G. As ref. 11. pp. 553.

19. ROSHKO, A. *N.A.C.A Tech. Note*, No. 2913, 1953.

20. ROSHKO, A. *N.A.C.A Tech. Note*, No. 3169, 1954.

21. GERRARD, J.H. The mechanics of the formation region of vortices behind bluff bodies. *J. Fluid Mech.*, Vol. 25, part 2, pp. 401-413, 1966.

22. GERRARD, J.H. The three-dimensional structure of the wake of a circular cylinder. *J. Fluid Mech.*, Vol. 25, part 1, pp. 143-164, 1966.

23. PAPAILIOU, D.D. and LYKOUDIS, P.S. Turbulent vortex streets and the entrainment mechanism of the turbulent wake. *J. Fluid Mech.*, Vol. 62, part 1, pp. 11-31, 1974.

24. CANTWELL, B.J. A flying hot-wire study of the turbulent near wake of a circular cylinder at a Reynolds number of 140000. Ph.D. thesis, California Institute of Technology, 1975.

25. PERRY, A.E. and WATMUFF, J.H. The phase-averaged large-scale structures in three-dimensional turbulent wakes. *J. Fluid Mech.*, Vol. 103, pp. 33-51, 1981.

26. PERRY, A.E., LIM, T.T. and CHONG, M.S. The instantaneous velocity fields of coherent structures in coflowing jets and wakes. *J. Fluid Mech.*, Vol. 101, pp. 243, 1980.

27. BRAZA, M., CHASSAING, P. and MINH, H. HA. Numerical study and physical analysis of the pressure and velocity fields in the near wake of a circular cylinder. *J. Fluid Mech.*, Vol. 165, pp. 79-130, 1986.

28. SONG, C.C.S. and YUAN, M. Simulation of vortex-shedding flow about a circular cylinder at high Reynolds numbers. *Journal of Fluids Engineering*. Vol. 112, pp. 155-163, June, 1990.

29. SMITH, P.A. and STANSBY, P.K. Postcritical flow around a circular cylinder by the vortex method. *Journal of Fluids and Structures*. Vol 3., pp. 275-291, 1989.

30. MAEKAWA, I. Numerical diffusion in single-phase multi-dimensional thermal-hydraulic analysis. *Nuclear Engineering and Design*. Vol. 120, pp. 323-339, 1990.

31. LESCHZINER, M.A. Practical evaluation of three finite difference schemes for the computation of steady-state recirculating flows. *Computer Methods in Applied Mechanics and Engineering*. Vol. 23, pp. 293-312, 1980.

32. DAVIS, R.W. and MOORE, E.F. A numerical study of vortex shedding from rectangles. *J. Fluid Mech.* Vol. 116, pp. 475-506, 1982.

33. DAVIS, R.W., MOORE, E.F. and PURTELL, L. A numerical-experimental study of confined flow around rectangular cylinders. *Phys. Fluids*, 27(1), pp. 46-59, January, 1984.

34. LEONARD, B.P. A stable and accurate convective modelling procedure based on quadratic upstream interpolation. *Computer Methods in Applied Mechanics and Engineering*. Vol. 19, pp. 59-98, 1979.

35. PATANKAR, S.V. *Numerical Heat Transfer and Fluid Flow.* New York, Hemisphere, 1980.

36. PATEL, M.K., MARKATOS, N.C. and CROSS, M. Technical note - Method of reducing false-diffusion errors in convection-diffusion problems. *Applied Mathematical Modelling.* Vol. 9, pp. 302-306, August, 1985.

37. LESCHZINER, M.A. Modeling turbulent recirculating flows by finite-volume methods - current status and future directions. *International Journal of Heat and Fluid Flow*. Vol. 10, No. 3, pp. 186-202, September, 1989.

38. SPALDING, D.B. PHOENICS Instruction Courses, Course notes, CHAM TR/300, 17-21 April, 1989.

39. MARKATOS, N.C. Computer simulation techniques for turbulent flows. *Encyclopedia of Fluid Mechanics.* Vol. 6 - Complex Flow Phenomena and Modeling, Ed. N.P. CHEREMISINOFF, pp. 1221-1275, 1987.

40. TENNEKES, H.L. and LUMLEY, J.L. *A First Course in Turbulence*. M.I.T Press, 1972.

41. PRANDTL, L. and WEIGHARDT, K. Uber ein neues Formelsystem fur die ausgibildete Tubulenz. Nachr. Akad. Wiss., Gottingen, *Math - Phys. K1.*, pp. 16, 1945.

42. KOLMOGOROV, A.N. Equations of turbulent motion of an incompressible fluid. Izv. Akad. Nauk. SSSR, Ser. Phys. 6 No. 1/2, pp. 56/58, 1942; (transl. Engl. by D.B.Spalding as *Imperial College, Mech. Engng. Dept. Report No. ON/6*, 1968).

43. LAUNDER, B.E. and SPALDING, D.B. The Numerical Computation of Turbulent Flows. *Comp. Meths. Appl. Mech. Engrng.*, Vol. 3, pp. 269-289, 1974.

44. DYKE, M.van. *An Album of Fluid Motion.* The Parabolic Press, Stanford, CA., 1982.

45. ANDERSON, C.R. *et al.* On the accurate calculation of vortex shedding. *Phys. Fluids*, A 2(6), pp. 883-885, 1990.

46. PATEL, M.K., CROSS, M. and MARKATOS, N.C. An Assessment of Flow Oriented Schemes for Reducing 'False Diffusion'. *International Journal for Numerical Methods in Engineering* , Vol. 26, pp. 2279-2304, 1988.

47. RAITHBY, G.D. Skew-Upstream Differencing Schemes for Problems Involving Fluid Flow. *Computer Methods in Applied Mechanics and Engineering.* Vol. 9, pp. 151-162, 1976.

48. RICE, J.G. and SCHNIPKE, R.J. A Streamline Upwind Finite Element Method for Laminar and Turbulent Flow. *University of Virginia, School of Engineering and Applied Science.* Report No. UVA/643092/MAE86/342, June, 1986.

49. ARMALY, B.F., DURST,F., PEREIVA, J.C.F. and SCHONUNG, B. Experimental and Theoretical Investigation of Backward-Facing Step Flow. *Journal of Fluid Mechanics.* Vol. 127, pp. 473-496, 1983.

50. SPALDING, D.B. A conservative low-dispersion algorithm for the reduction of numerical diffusion. Paper presented at the *ICHMT Conference on Numerical Heat Transfer*, 1991.

51. LESCHZINER, M.A. and RODI,W. Calculation of Annular and Twin Parallel Jets Using Various Discretization Schemes and Turbulence Model Variations. *Journal of Fluids Engineering.*, Vol. 103, pp. 352-359, June, 1981.

52. MOTTRAM, R.C. and RAWAT, M.S. Installation effects on vortex flowmeters. *Measurement and Control*, Vol. 21, pp. 241-246, October, 1988.

53. de VAHL DAVIS, G. and MALLINSON, G. D. False Diffusion in Numerical Fluid Mechanics, University of New South Wales, School of Mech. and Ind. Eng., Report No. 1972/FMT/1, 1972.

54. HUMPHREYS, J.S. *Journal of Fluid Mechanics.* Vol. 9, pp. 603, 1960.

55. MORKOVIN, M.V. Flow around a circular cylinder: a kaleidoscope of challenging fluid phenomena. *ASME Fluids Eng. Div. Conf. Symp. on Fully Separated Flows*. A.S.M.E., 1964.

56. ROSHKO, A. *Journal of Fluid Mechanics.* Vol. 10, 1961.

57. REDDY, J.N. On penalty function methods in finite element analysis of flow problems. *International Journal for Numerical Methods in Fluids*. Vol. 2, pp. 151-171, 1982.

58. VAN DE VOSSE, F.N. *et al* A Finite Element Approximation of the Unsteady Two-Dimensional Navier-Stokes Equations. *International Journal for Numerical Methods in Fluids*. Vol. 6, pp. 427-443, 1986.