NUMERICAL SIMULATION OF LAMINAR SEPARATED FLOWS ON ADAPTIVE TRI-TREE GRIDS WITH THE FINITE VOLUME METHOD

by

Zheng Zheng Hu

A thesis submitted for degree

of Doctor of Philosophy

Department of Mechanical Engineering

University College London

2000

UNIVERSITY OF LONDON THESIS

Degree PhD Year 2000 Name of Author Hu, Z.Z.

To my dear father

ABSTRACT

In this work, a code has been developed that solves the Navier-Stokes equations using the finite volume method with unstructured triangular grids. A cell-centred, finite volume method is used and the pressure-velocity coupling is treated using both the SIMPLE and the MAC algorithms. The major advantage of using triangular grids is their applicability to complex geometry. A special treatment is developed to ensure good quality triangular elements around the boundaries. The numerical simulation of incompressible flow at low Reynolds number is studied in this thesis.

A code for generating triangular grids using the tri-tree algorithm has been written and an adaptive finite volume method developed for calculating laminar fluid flow. The grid is locally adapted at each time step, with grid refinement and derefinement dependent on the vorticity magnitude. The resulting grids have fine local resolution and are economical in reducing the numerical simulation time.

The discretised equations are solved by using an iterative point by point Gauss-Seidel solver. For calculating the values of velocity and pressure at vertices of triangular grids, special interpolation schemes (averaged linear-interpolation and scattered interpolation) are used to increase the accuracy. To avoid the well known checkerboard error problems, i.e., the oscillations occurring in the pressure field, third derivative terms in pressure, first introduced by Rhie-chow (1983), are added to the mass flux velocity. Convective terms are approximated using a QUICK (Quadratic Upstream Interpolation for Convective Kinematics) differencing scheme which has been developed here in for unstructured grids.

Three cases of two-dimensional viscous incompressible fluid flow have been investigated: the first is channel flow, in which the numerical results are compared with the analytical solution; the second case is the backward-facing step flow; and the third case is flow past circular cylinders at low Reynolds number (*Re*). The numerical results

obtained for the last two cases are compared with published data. The evolution of vortex shedding is presented for the case of unidirectional flow past a circular cylinder at *Re*=200. In addition, drag and lift force coefficients are calculated and compared for single and multiple cylinders in unidirectional flow.

ACKNOWLEDGEMENTS

I would first like to express my sincere gratitude to my supervisor Dr. G.X. Wu for his supervision and his valuable suggestions, especially his guidance in using the MAC algorithm.

I would also like to express my sincere gratitude to my supervisor Dr. D.M.Greaves for her regular valuable guidance, advice and constant encouragement throughout my studies.

In addition, I should like to express my appreciation to Dr. A.G.L. Borthwick and Professor M.Giles of the University of Oxford for their invaluable suggestions and discussions in my work. I am also very grateful for the support provided by Dr. D. Davis, who read Chapter 2 and Mr. C. Santos, who read Chapters 3, 7 and 8. Their suggestions resulted in significant improvement.

I have received the most influential and unconditional support from my family, my husband and my son in a way of endless patience. Finally, I would like to take this opportunity to pay my sincere tribute to my dear father, who passed away two years ago during the busy time of my study. I could not determine to complete my Ph.D. without his encouragement and wishes.

CONTENTS

FIGURES

TABLES

NOMENCLATURE

(a) *Roman Symbols*

A area of each triangular element
A_j discretisation coefficient of the part neighbouring cell element for QUICK scheme
A_{fj} face area of each triangular element
a position of the vortex cores in x-direction
a_p discretisation coefficient of the given cell element for central differencing scheme
a_{fj} discretisation coefficient of the neighbouring cell element for central differencing scheme
a/d_c non-dimensional vortex cores in x-direction
B_j discretisation coefficient of the part neighbouring cell element for QUICK scheme
b half channel width; position of the vortex cores in y-direction
b/d_c non-dimensional vortex cores in y-direction
C_D dimensionless drag force coefficient
C_L dimensionless lift force coefficient
C_j discretisation coefficient of the given cell element for QUICK scheme
C_d drag force
C_l lift force
D length of the face of the initial triangle
D_{fj} diffusion coefficient
D_i diameter of the circle of seeding point
d difference operator; face area of triangular element; characteristic length for channel flow
d_c diameter of circular cylinder
f vortex shedding frequency
h distance associated with the optimum ratio; step size for backward-facing step
i_j neighbours of given cell element
$\mathbf{J}$ momentum flux vector containing convection and diffusion
K^i sum of successive powers (i=5 is used to tri-tree decimal grids)
K_{fj} diffusion term
k parameter depending on the maximum division level
L length of channel flow; wake of length
Listc cell index information
Listf face index information
L_k ratio between the area of the three triangles and the area of the triangular element
L_d downstream channel length for backward-facing step
L_u upstream channel length for backward-facing step
L/d_c ratio of close-wake length versus the diameter of cylinder
m tri-tree division level
m_{fj} mass flux value
m^*_{fj} guessed mass flux value

m'_{fj}	correction mass flux value
N	reference number of given cell
NL	generation level of cell
N_p	parent reference number
n	maximum tri-tree division level
n_x	x-components of n
n_y	y-components of n
n_{xfj}	x-face components of n
n_{yfj}	y-face components of n
$\vec{n}_j$	face normal vector
P_c	Peclet number
P_k	pressure value corresponding the cell element
p	pressure of the fluid
p_i	pressure value of the given cell element
p_{ij}	neighbour pressure value of given cell element
p'	pressure correction field
p^*	pressure guessed field
p'_i	pressure correction value at the given cell element
p'_f	pressure correction value at the face of the element lying on the boundary
P/D	non-dimensional distance between the centres of the cylinders in tandem arrangement
R_e	Reynolds number
$R_e e$	element Reynolds number
r	the optimum ratio for the stretching technique
S	Strouhal nember
S^ϕ	total pressure gradient source in the control volume
S^ϕ_{non}	non-orthogonal diffusion term
$\bar{s}^\phi$	a source per unit volume
T_i	triangular element numbering
t	time
t^*	non-dimensional time
T/d_c	non-dimensional distance between the centres of the cylinders in transverse arrangement
U_o	free stream velocity
U^{fj}	face normal velocity
U'^{fj}	correction face normal velocity
u_{avg}	average inlet velocity for backward-facing step
u_{fj}	face velocity in x-direction
$u_{max.}$	maximum velocity in x-direction
u_{jst}	velocity value at starting node of each element face in x-direction
u_{jend}	velocity value at ending node of each element face in x-direction
u_i^*	velocity guessed field in x-direction
V	volume of the given triangular element
V_{fj}	enclosed area (a-b-c-d) of diffusion volume
v	velocity in y-direction

v_{jst}	velocity value at starting node of each element face in y-direction
v_{jend}	velocity value at ending node of each element face in y-direction
v_i^*	velocity guessed field in y-direction
w_{fj}	face coefficient of pressure gradient correction
w_k	inverse distance weighting function for scattered data interpolation
X_R	reattachment length
x_k	co-ordinate of the centre element
X_R^*	dimensionless reattachment length

(b) Greek Symbols

α	under-relaxation factor
ρ	density of the fluid
ν	kinematic viscosity
μ	dynamic viscosity
μ_ϕ	dynamic viscosity of general variable
λ_j	weight function for linear interpolation
Δ	Laplace operator
∇	gradient operator
Δx	general cell element width
Δy	general cell element height
Δp	pressure at regular interval
$\Delta \omega$	vorticity at regular interval
$\nabla \phi$	vector operator of general variable
$\Delta \psi$	streamline faction at regular spacing
Δt^*	non-dimensional time step
$\Delta \xi$	small distance along ξ direction
ϕ	general variable represents u, v and p
ϕ_f	face general variable of the element lying on the boundary
ϕ_i	general variable of the given cell element
ϕ_{ij}	general neighbour variable of the given cell element
ϕ_r	general variable of interpolation
∂	partial difference operator
$\frac{\partial}{\partial t}$	partial time derivative
β	the angle of distorted triangle; the angle of notation associated with Poisson equation
θ_s	separation angle
ω_i	vorticity value of given cell element
ω_{max}	maximum vorticity limit for adaptation
ω_{min}	minimum vorticity limit for adaptation

(c) Acronyms

CFD	Computational Fluid Dynamics
MAC	Marker And Cell method
PDE	Partial Differential Equation
PEP	Poisson Equation for Pressure
QUICK	Quadratic Upstream Interpolation for Convective Kinematics
SIMPLE	Semi-Implicit Method for Pressure Linked Equations

(d) Subscripts

(Applicable to all symbols used within the thesis)

i	given cell element
ij	given cell element at *j*th neighbour
jst	starting point of each face triangular element
jend	ending point of each face triangular element
il	left cell element of given cell
ir	right cell element of given cell
fj	face number at *j*th face

(e) Superscripts

(Applicable to all symbols used within the thesis)

*	non-dimensional value; guessed value
'	correction

CHAPTER 1

INTRODUCTION

1.1 Scope of the Thesis

In 1989 P. D. Lax predicted, in the special report at the centenary ceremony of the American Mathematical Seminar (AMS), that fluid dynamics would become one of the central study fields in applied mathematics. The use of computational fluid dynamics (CFD) to predict flows has risen dramatically in the past decade. The codes that are now on the market may be powerful, but their capabilities are limited to simple industrially relevant applications. They still require improvements in order to obtain meaningful results for complex modelling. Therefore, much of the effort surrounding CFD research is to provide effective and efficient numerical solution algorithms to enhance the problem solving capabilities.

In ocean engineering and naval architecture, assessment of the response for design purposes is complex due to various factors including flow separation, vortex shedding, turbulence and body motions. In some applications, such as a ship with forward speed, large areas of flow separation have to be considered. Accurate numerical simulation of separated flow plays an important role for this purpose. Turbulent flow exists in most engineering situations, however the nature of turbulent flow is complex and CFD turbulence models tend to be problem dependent. The study of separated flow in this thesis is based on laminar flow. Although the laminar flow can be exactly expressed by the Navier-Stokes equations, the development of numerical algorithms for CFD implementation is still at the state of validation and improvement.

Offshore structure design needs to have some knowledge of fluid flow around a bluff body. A circular cylinder in fluid flow has been a subject of intense interest for a long time, because this topic has a great practical background and important academic significance. Steel offshore structures typically comprise tubular cylinders connected together at nodal junctions and other slender elements of offshore installations include risers and umbilical pipes; all of these elements are treated as circular cylinders in hydrodynamics. A circular cylinder has geometric simplicity enabling models to be easily manufactured for experimental research. Its simple form also makes it attractive as a representative of a bluff body for CFD investigations.

For viscous flow past a cylinder, the Reynolds number uniquely describes the flow and is given by

$$Re = \frac{U_0 d}{\nu}, \tag{1-1}$$

where d is the diameter of the cylinder, U_0 is the free stream velocity and ν is the kinematic viscosity of the fluid.

With regarding to the Reynolds number, Roger (1980) pointed out that the flow will not separate if Reynolds number is less than 5. Above this value, the flow separates and two steady Föppl vortices will appear behind the cylinder when it reaches about Re=46. Upon reaching the critical Reynolds number, an unsteady wake will start to form behind the cylinder. This unsteadiness moves towards the cylinder until vortex shedding begins. Laminar alternate vortex shedding is observed at Reynolds numbers up to approximately Re=100,000 as described by Saalehi (1995). The flow separation and vortex shedding is simulated and validated in this work.

As mentioned above, fluid flows around a circular cylinder are characterised mainly by separation flows and vortex shedding. The dominant contribution to the drag force is due to separation, which occurs near the midplane of the body. As a result of separation, there is a substantial pressure difference on the fore-body and after-body. Vortex shedding behind the cylinder results in oscillatory drag and lift forces, which may cause the cylinder to vibrate if it is flexibly mounted. Thus, the ability to calculate the force exerted by fluid flow past any immersed body (such as a circular cylinder) accurately and efficiently by numerical computation will be invaluable in the design of offshore installations.

This work involves developing an efficient triangular grid generation procedure which can provide fast adaptation locally and approximate a curved boundary properly without losing its computational quality. Grid generation is an essential part of the numerical simulation of engineering problems. Unstructured tri-tree grids are employed in this

study, which can use grid adaptation to provide a rational distribution according to the local flow solution.

The hierarchical tri-tree grid is ideal for this application as it can be automatically adapted by adding and removing cell elements according to the internal flow solution. This technique ensures good quality elements and maintains the tree data structure. In this work, the numerical schemes have first been validated using grids in which quadrilaterals are split into triangles for cases involving simple geometry (channel flow). Then the hierarchical tri-tree grids with and without grid adaptation are applied to other cases (backward-facing step and fluid flow past circular cylinders).

The Navier-Stokes equations are solved using the finite volume method with tri-tree grids. Fluid flow is governed by the Navier-Stokes momentum equations coupled with the continuity equation. The finite volume method is applied to calculate the velocities and pressure distribution in a given domain which may include a body, such as the circular cylinder. Then, fluid flow forces which include the drag and lift can be calculated by integrating the pressure and shear stresses around the surface of the body.

The finite volume method has been chosen for use in this study. Finite volume formulations involve discretisation of the calculation domain by integration of the flow equations over each element. The principal advantage of the finite volume method is unconditional satisfaction of the mass conservation principle so that the discretisation of the flow-governing transport equations guarantees the calculation process against false mass flux across the boundaries. The finite volume method has most commonly been used on structured quadrilateral grids. There are only a limited number of papers in the literature which describe the finite volume method using triangular grids such as Jameson et al. (1986), Davidson (1996), Thomadakis and Leschziner (1996), Minyard and Kallinderis (1998) as well as Chan *et al.* (1999), because the discretised mathematical formulations are more complicated than for quadrilateral grids. For this reason, much of the work presented here is focused on the derivation of the finite volume method on triangular grids.

The fluid flow variables are assigned either at cell centres or at the staggered locations of each triangular control volume. In principle the staggered scheme has better accuracy than the cell centred scheme, because in the cell centred scheme, interpolation is necessary to calculate the values at vertices and faces of the triangular elements. However, the staggered scheme has the disadvantage that the coefficients of the convection term in the discretised equations are not the same for all the variables. This results in a low efficiency computer programme. In contrast the cell centred scheme can have a higher efficiency, because the coefficients of the convection term are the same for all the variables.

A QUICK (Quadratic Upstream Interpolation for Convective Kinematics) differencing scheme for unstructured grids is derived in this work, which calculates the cell face values by fitting a quadratic profile through three nodes surrounding each face. The scheme which satisfies the transportiveness requirement is stable and possesses good accuracy associated with the non-linear convective term of the Navier-Stokes equations.

Fluid dynamic interaction commonly occurs in multiple cylinders and is an important phenomenon in some engineering flows. For example, it occurs in flows that involve offshore structures, transmission cables with twin conductors, twin chimney stacks and heat exchanger tubes. Much research has been made into understanding the interactions for flow past a single cylinder in the past decades. However, results for flow past multiple cylinders are not so numerous. Therefore, numerical results are obtained for fluid flow past two and three circular cylinders in this thesis.

1.2 Aim of the Thesis

The aim of the work is to create a code for numerical simulation of laminar separated flows on adaptive unstructured grids with the finite volume method.

1.3 Objectives

The main objectives of the PhD work are:

(i) to derive the discretised mathematical formulations for solving the Navier-Stokes equations using the finite volume method with triangular grids;
(ii) to study an automatic tri-tree grid generation algorithm;
(iii) to study an efficient adaptive grid generation algorithm, using hierarchical trees, that allows high local resolution;
(iv) to simulate steady and unsteady viscous separated flow past circular cylinders using adaptive tri-tree grids and to calculate the drag and lift forces which result.

1.4 Synopsis

The review and discussion of previous work in grid generation, numerical algorithms and adaptive grid generation is described in Chapter 2. Chapter 3 describes the development of an efficient triangular grid generation procedure which can provide fast adaptation locally using hierachical tri-tree grids. A special treatment is developed to ensure good quality triangular elements around the boundaries of the domain. In Chapter 4, the Navier-Stokes equations governing the two-dimensional flow of viscous incompressible fluids are introduced and their discretised equations using a collocated finite volume method are derived. SIMPLE and MAC solution algorithms for the Navier-Stokes equations are also described in Chapter 4. A QUICK differencing scheme for unstructured grids is developed in this work.

Chapter 5 presents results from the first application of channel flow for assessment of SIMPLE and MAC algorithms and the differencing schemes. The results for flow over a backward-facing step using tri-tree grids are presented and compared with published data in Chapter 6. In Chapter 7, the application of adaptive tri-tree based grids to simulation of flow past single and multiple cylinders is described. Grid convergence tests for unidirectional flow past a circular cylinder at Re=2.04 are first presented. The results of flow past a circular cylinder at Reynolds number 10, 40 and 200 are presented and

compared with published data, as well as the results at *Re*=40 past two and three circular cylinders. The evolution of vortex shedding is presented for the case of unidirectional flow past a circular cylinder at *Re*=200. In addition, the drag and lift force coefficients are calculated and compared for single and multiple cylinders in unidirectional flow. These results give significant insight into the understanding of the interactions of the laminar flows. Finally, conclusions and recommendations are given in Chapter 8. The project is implemented in a software package written in *FORTRAN 77*.

CHAPTER 2

LITERATURE REVIEW

When solving problems in fluid mechanics, we should be aware that a good understanding of numerical algorithms is important. The CFD code, comprising both a grid generator and a numerical solver, is structured around the numerical algorithms used to simulate the flow. Clearly, there exist various methods for grid generation as well as an assortment of methods for numerical solution and some of these will be discussed in this Chapter.

The present study, using the finite volume method with unstructured tri-tree grids, is found to be computationally effective and numerically stable for treating incompressible viscous fluid flow. This Chapter will review and discuss previous studies and techniques, which focus on general grid generation, numerical methods and adaptive grid generation.

2.1 Grid Generation

The first stage in the development of a numerical flow simulation is the creation of a suitable grid. For viscous flow problems involving complex boundaries and time dependent flow, it is generally preferred to use a grid generation algorithm that automatically generates a grid about boundaries and subdivides the domain into elements. The grid generator should also have the capability of local refinements based on the solution of the governing equations at each time step. Many grid generation methods have been proposed to satisfy some, or all, of these requirements. Structured grid generation, unstructured grid generation and hybrid grid generation will be described in the following Sections.

2.1.1 Structured grid generation

Structured grids have an implied co-ordinate direction, and are typically composed of quadrilaterals in two-dimensions and hexahedra in three-dimensions. Numerical grid generation, algebraic grid generation and multi-block methods are represented within this class of grid generation.

2.1.1.1 *Numerical grid generation*

The numerical grid generation technique involves solution of a Partial Differential Equation (PDE) and has proved to be a powerful approach for creating grids around complex shapes. Thompson *et al.* (1982*b*) used elliptic equations to generate grids in both two- and three-dimensions. In order to relate the computational domain to the physical domain, one of three types (*O*-type, *C*-type and *H*-type) of grid geometry are usually used as distinguished by George (1991). Application of these types of grid generators can be found, for example, in Thompson (1987), Franke *et al.* (1990), Mittal *et al.* (1997) and Chen *et al.* (1999).

The advantage of the method is that it can generate grids with orthogonality of elements. These grids have the direction normal to the surface so they simplify the numerical method without concerning non-orthogonal terms. Chen *et al.*(1999) have been successful in simulating of flow past circular cylinder up to Reynolds number, *Re*=200, with combination of the *O* and *H*-type grids as shown in Figure 2.1; here, the first grid spacing is 0.005 cylinder diameters, which suffices to keep the first grid within the region of the viscous boundary layer. The dimensions of this layer are governed by the Reynolds number.

However, it is more difficult to generate a rational distribution of elements within the numerical grid generation technique when two or more different bodies are together in the domain. For example, for a domain containing two or more cylinders, it is a difficult challenge to produce a structured grid that is aligned with all solid surfaces. The reason is that the transformation function (PDF) is very difficult to compute which maps the real domain onto a reference domain past two bodies. Even for a domain with a simple body that uses an adaptive grid, a large amount of memory and CPU time are needed to regenerate entire grids.

2.1.1.2 *Algebraic grid generation*

Algebraic grid generation or transport-mapping methods are based on the mapping of a reference grid corresponding to an elementary geometry (such as a quadrilateral or a

triangle in two dimensions). The reference grid is mapped onto the real domain via a transport function. This function, say F, is constructed to a polygonal contour, and defines a one-to-one mapping f_j from a real domain Ω to the reference domain $\hat{\Omega}$ as shown in Figure 2.2. The final computational results are converted from the reference grid into the real domain. Algebraic methods, based on surface fitting by Eiseman (1985), transfinite interpolation by Eriksson (1982) and sequential mapping by Baker (1987), have been used to approximate certain geometric shapes in two and three dimensions.

Westermann (1992) constructs the mapping f_j using linear and bilinear interpolation for curved quadrilaterals. However, the scheme may experience difficulties in seeking the right element where the point is located in the real domain. This is demonstrated by a numerical example presented by Allievi and Bermej (1997). Higher order interpolation functions to define the mapping f_j to curved elements is used by Allievi and Bermej (1997), in which arbitrary grids composed of both three and four sided elements are constructed as shown in Figure 2.3.

2.1.1.3 *Multi-block methods*

The above mentioned two methods encounter difficulties when being applied to complex geometry. Multi-block methods or super-element methods, introduced by George (1991), have been shown to be suitable for complex geometry; where in, the domain is firstly divided into a series of blocks of simple geometry. Then, using one of the methods described above, grids are generated in each block until the whole domain is discretised. In some cases, introduced by Weatherill and Forsey (1985) and Yu *et al.* (1987), the grid is required to blend smoothly together at block interfaces to provide a grid that can be viewed as a single block by the flow solver. In other cases, as introduced by Benek *et al.* (1985) and Flores *et al.* (1987), the grid is not required to connect smoothly at the interfaces and interpolation is used to transfer flow information between different blocks.

2.1.2 Unstructured grid generation

Unstructured grids do not have an implied co-ordinate direction and are typically composed of quadrilaterals or triangles in two-dimensions and hexahedra or tetrahedra in three-dimensions. Two major advantages of unstructured grids compared to structured grids are firstly, their greater flexibility for generating grids about complex configurations and secondly, the ease with which grid adaptation can be incorporated. Several algorithms have been devised for this class of grid generation.

2.1.2.1 *Hierarchical quadrilateral or hexahedral grid generation*

Hierarchical grid generation is conceptually easy to understand with the grids being created by recursive subdivision about a set of seeding points forming the boundary of the domain. Figure 2.4 shows a typical quad-tree generated grid, as taken from work by Gáspár *et al.* (1991). This quad-tree grid is generated about a single point and restrictions have been applied so that the maximum ratio of cell side length of each cell to any of its neighbours is 2:1. An advantage of hierarchical grids is that the grid information is stored in a tree, which can be traversed according to simple rules. Furthermore, hierarchical grids are suitable for adaptation, where the grid can adjust density locally within the domain according to the flow solution.

Samet (1990) described the use of quad-trees for the spatial decomposition of computer images, which leads to a substantial reduction in computer storage requirements. Subsequently, the quad-tree technique has been employed for grid generation in various applications including grid adaptation. Examples include the modelling of shock waves by Schmidt (1991) and Evans (1993); vortex tracking calculations by Van Dommelen and Rundensteiner (1989), who exploited the fast adaptive nature of quad-tree grids; pollutant transport by Gáspár *et al.* (1991), who applied multi-grid techniques in the quad-tree context; and wind-induced flow patterns in shallow lakes by Józsa and Gáspár (1992). Moreover, Greaves and Borthwick (1998) applied adaptive quad-tree grids for flow past a circular cylinder and a square cylinder in two-dimensions. They also demonstrated hexahedral oc-tree grids in three-dimensions.

The main disadvantages associated with the use of quad-tree grids in two-dimensions are firstly, the occurrence of hanging nodes within the grids and, secondly, if not further divided into triangles, poor approximation to curved boundaries. A variety of ideas on how to deal with the hanging node problem have emerged. These include triangularisation of the grid; a scattered data-type integration, such as described by Kochavi and Segev (1991) and Morgan *et al.* (1994); or some form of averaging of the neighbour cells. The quad-tree grids are divided into triangles by joining the centres of adjacent quadrants in order to eliminate hanging nodes from the grids by Greaves (1995) as illustrated in Figure 2.5. Saalehi's (1995) special treatment around a circular boundary using 'filling' elements is shown in Figure 2.6.

A major advantage of using triangular grids is their applicability to approximate curved boundaries and complex configurations. Also, when using the finite element method, a disadvantage of rectangular elements is that the derivation of shape functions, element matrices and vectors becomes more complicated than for linear triangular elements as noted by Saalehi (1995). Jameson and Baker (1987) explain how triangular elements may be used for a variety of flow problems. Several algorithms building on triangular grid techniques have subsequently been developed.

2.1.2.2 *Delaunay triangulation*

Delaunay triangulation was first used for magnetic field problems by Cendes *et al.* (1983). It is usually defined in terms of an auxiliary diagram called the Voronoi diagram (see Voronoi (1908)). Each point is assigned to a territory that is closer to that point than to any other point. Such regions for each point are called Voronoi neighbourhoods, and if points with common boundaries of Voronoi neighbourhoods are connected, then the Delaunay triangulation is obtained. An example of a Voronoi diagram for 6 points is shown in Figure 2.7 by Cendes *et al.* (1983). Details of the implementation of this approach have been given by Jameson and Baker (1987).

Several algorithms have been proposed for generating the Delaunay triangulation of a set of points in the plane. In order to achieve an efficient algorithm, some improved

techniques have been adopted. There include divide-and-conquer strategies by Shamos (1975) and a sweep-line approach by Forture (1986). The use of Delaunay triangulation to connect an arbitrary set of points to produce the grids around aerofoil (2D) and the surface of a nacelle (3D) were initiated by Weatherill *et al.* (1994), who successfully constructed an algorithm for arbitrary two-dimensional configurations. In three-dimensions the successful implementation of this approach has proved difficult, because of the incorporation of directional refinement, which is an essential ingredient in any optimal three-dimensional algorithm for incompressible flows. Preliminary results of a finite element solution of the Euler equations using a tetrahedral grid were reported by Jameson and Baker (1987). At that time, however, the triangulation algorithm was extremely time consuming and there were difficulties with the procedures for identifying and improving the triangulation of the aircraft structure. Later, Baker's (1987) implementation and optimisation of the Voronoi algorithm showed that fast and reliable grid generators for tetrahedral grids can be produced. Delaunay triangulation has been successfully employed by Holmes and Lamson (1986) for two- and three-dimensional simulations of steady inviscid flows in aeronautical applications.

Delaunay triangulation can be an effective grid generator. However, it may fail to generate exact boundaries, such as non-convex boundaries. These failures are caused by the Delaunay triangulation itself, since the method merely subdivides the convex domain occupied by the nodes into triangles rather than the nodes outside the triangles. Details of those are described by Bowyer (1981), Watson (1981) and Sloan (1984). Sloan (1987) proposed an algorithm to modify the Delaunay triangulation so that it could be applied to arbitrary domains in two-dimensions (i.e. includes non-convex boundary curves). Taniguch *et al.* (1992) has studied ways to recognise the boundary of an arbitrary domain using nodes, and also how to introduce it into the Delaunay triangulation. However, Wille (1996) pointed out that the major difficulty associated with Delaunay triangulation is generating points for a non-convex boundary, which can introduce degenerate elements in three-dimensions.

2.1.2.3 *Advancing front triangulation*

The incorporation of directional refinement in the Voronoi algorithm appears to be difficult, unless the reconnection of points based on the purely geometrical Delaunay criterion (see Baker (1987)) is substituted by some other criterion that incorporates directionality into the triangulation. The advancing front (sometimes called moving front) technique can easily be used for grid regeneration with directional refinement as introduced by Peraire *et al.* (1987), because the direction is chosen to coincide with a defined 'normal' direction at each triangular surface.

It starts with the boundary as a front and generates triangles or tetrahedrons by selecting new points inside the domain and making linear connections with them. As new triangles or tetrahedrons arise, the front, which is the base for generating new elements, moves further inside the domain, until finally the entire domain is triangulated. An example of such a grid is shown in Figure 2.8, which describes the wing body surface layer in three-dimensions by Hassan and Morgan (1996). Löhner and Parikh (1989) describe the detail of three-dimensional advancing front grid generation which uses tetrahedral elements, and its application on the pathfinder configuration in a wind tunnel and missile launcher. Peraire *et al.* (1987) used such grids successfully for two-dimensional and three-dimensional flow simulations.

However, in some cases (for example, when the initial boundary is not smoothly discretised) the technique may not converge. This approach also faces major implementational difficulties when an extension to three-dimensions is contemplated. These difficulties arise because, in three-dimensions, grid movement is not a straightforward procedure and adaptive grid generation entails an unacceptable expansion in the number of grid points, while its application involves time consuming interpolation between grids. Although Hassan and Morgan (1996) avoided these difficulties by addressing the alternative approach of generating some distorted grids directly, which used some elements with an aspect ratio greater than one in the domain, a demonstration of the validity of the generated grid has only been shown for laminar viscous flow analysis.

2.1.2.4 *Hierarchical triangular or tetrahedral grid generation*

Triangular grids generated using a hierarchical tri-tree method and tetrahedral grid generation using a hierarchical tetra-tree method are described by Wille (1992), and can be used for adaptive applications in two-and three-dimensions. In the search algorithms, an equilateral triangle or tetrahedron is used as the basic domain. The equilateral triangle in two-dimensions is divided into four equilateral triangles, while the equilateral tetrahedron in three-dimensions is divided into eight tetrahedra. The algorithm is repeated until the required resolution is achieved. The storage of tri-tree reference numbers allows the algorithm to perform both up- and down-searches for adaptive applications as required. The tri-tree algorithm has been applied on adaptive multi-grids for the cavity case while the tetra-tree algorithm has been used to model the North Sea-Skagerraka (see Figure 2.9) by Wille (1996).

In the present study, it is intended to study and apply the hierarchical tri-tree grid generation in two-dimensions. Tri-tree grid generation ensures good quality elements and in principle, a straightforward extension to three-dimensions using tetra-tree grid generation. These properties are extremely desirable, especially with the increasing capability of CFD to tackle a wider range of engineering problems in fluid mechanics.

2.1.3 Hybrid grid generation

Hybrid grids usually consist of quadrilaterals and triangles in two-dimensions, and prisms and tetrahedrons in three-dimensions. Figure 2.10 (see Kallinderis (1998)) shows that quadrilaterals are employed to model boundary layers, while triangles cover the rest of the domain. Hybrid grids are intended to provide flexibility in combining essential features of the two main types of grids, namely the structured and the unstructured grids. Hybrid grids in two-dimensions consisting of triangles and quadrilaterals have been employed by Kallinderis and Nakajima (1994), Hufford and Mitchell (1995), and Banks *et al.* (1996). In three-dimensions, hybrid grid techniques involve generating a grid made up of tetrahedral and prismatic elements and then destructuring the prisms to form tetrahedra as introduced by Connell and Braaten (1995)

and Pirzadeh (1994). Adaptation and load balancing computation of hybrid grids have been considered by Parthasarathy *et al.* (1996) and Minyard and Kallinderis (1998).

The accuracy and stability of a numerical method are almost always a function of the size and shape of the grid elements that fill the computational domain. Therefore, obtaining a grid that can follow complex surface geometry and flow patterns is of importance. Unstructured grids can provide flexibility and can cover complicated topologies more easily than structured grids. However, employment of triangular or tetrahedral elements for some regions of the fluid flow such as boundary layers is quite expensive. In these regions, strong solution gradients usually occur in the direction normal to the surface, so large-aspect-ratio elements are commonly employed to resolve the boundary layer. Structured grids are superior in capturing the directionality of the flow field in these viscous regions, since they can be aligned with the boundary layer. Therefore, the hybrid grid can be used to exploit the advantages of these two kinds of grids.

Hybrid grid generation exploits the benefits of both structured grid elements generated in the direction following a body surface and unstructured grid elements generated in a rational distribution in the domain. It would be especially desirable to apply these grids for high Reynolds number viscous flows, which are of great practical interest. However, Mavriplis (1991) has indicated that this type of compromise limits the flexibility of the original unstructured approach. He explained that "Geometry with close tolerance, where confluent wakes and boundary layers occur, may prove difficult to discretize with a hybrid approach, and the task of performing adaptive remeshing throughout the viscous and inviscid regions of flow can be considerably more complex than otherwise".

2.2 Numerical Methods

The Navier-Stokes equations with appropriate boundary conditions accurately describe viscous Newtonian fluid motion. However, the numerical solution of these equations is not straightforward since many difficulties are inherent, such as treatment of the coupling of the velocity through the pressure field as discussed by Versteeg and

Malalasekera (1995). A further numerical difficulty is how to accurately treat the mass-conservation condition. Moreover, the ability of the methodology to produce uniformly accurate solutions must make it possible to obtain numerically exact solutions (i.e. grid-independent) to the equations of motion, so that the mathematical models of the physical phenomenon under scrutiny can be evaluated with confidence.

Different numerical methods, including the finite element, finite difference, finite volume and spectral methods, have been employed to obtain numerical solutions of the Navier-stokes equations in various applications. The main differences between these numerical methods concern the way in which the flow variables are approximated and the fluid domain is discretised. The first three of the aforementioned methods will be described in the following Sections.

2.2.1 Finite element method

The finite element method was originally developed for structural mechanics. A characteristic feature of the finite element method is the partitioning of the domain considered into polygonal elements, with each element being assigned a certain number of nodes. In two-dimensions, the most popular element type is the triangle or quadrilateral, and in three-dimensions their respective counterparts are tetrahedron and hexahedron. The unknown flow variables are then approximated in each element by piecewise interpolation functions (usually linear or quadratic polygon). The approximated flow variables are substituted into the governing equations, which leads to a linear matrix system with the nodal values of the variables as unknowns. Zienkiewicz and Taylor (1991) discussed the standard work for application of the finite element method.

Many researchers have used the finite element method for solving the Navier-Stokes equations. Baruzzi *et al.* (1995) used the finite element approach, in which the fourth-order dissipation is recast as the difference of two Laplacian operators, allowing the use of bilinear elements. Hill and Baskharone (1995) have implemented quadratic interpolation for both the velocity and pressure fields with streamline upwinding, and

solved the equations by means of a multi-block technique, which employs active and inactive blocks with respect to the direct equation solver.

For large problems, the use of a direct solver is prohibitive due to both the large storage requirement and the computational time necessary to obtain a solution (see Gustafasson (1978)). Here, iterative methods can be more advantageous compared to direct solvers, as found by Langtangen (1986), Barragy (1988) and Carey (1989). However, the variation in success of most iterative solvers seems to be very much dependent on using a good preconditioner. For example, Sonneveld (1987) pointed out that the iterative preconditioned CGSTAB solver (a fast Lanczos-type solver) has previously demonstrated favourable convergence properties both for symmetric and asymmetric systems of equations. Wille (1995) used the CGSTAB iterative solver with ILU (presenting incomplete factorisation) preconditioner for Navier-Stokes equations on tri-tree grids, and applied that to modelling the North Sea-Skagerrak area and to driven cavity flow problems.

It is well known that the standard Galerkin finite element approximation leads to instability for problems at high Reynolds numbers often characterised by 'wiggles' in the flow. This is caused by the fact that the scheme uses a central-difference-type approximation for the convection term. Thus, the question is how to effectively approximate the convection term to avoid this instability. Several first-order so called 'upwind' approximations have been proposed by Thomasset (1981), Ohmori (1984) and Tabata (1986). Although they are stable, their accuracy to the Reynolds number is not good on account of the additional viscosity included. Kondo *et al.* (1988) presented a finite element scheme with third-order upwinding based on the Petrov-Galerkin method by Hughes and Brooks (1982). Giraldo (1997) used the Lagrange-Galerkin method on unstructured spherical geodesic grids, which is accurate to a high order and highly efficient. Tabata and Fujima (1991) have approximated the convection term to third-order accuracy, and employed a numerical algorithm based on solving a pressure Poisson equation for flows in a cavity and past a circular cylinder. Mittal *et al.* (1997) used a stabilised finite element formulation applied to flows past a pair of cylinders with staggered arrangements.

2.2.2 Finite difference method

The finite difference method confines the values of the unknown variables to the nodal points on a grid composed of orthogonal co-ordinate lines; the co-ordinate system is normally Cartesian or curvilinear. As explained by Versteeg and Malalasekera (1995) "truncated Taylor series expansions are often used to generate finite difference approximations of derivatives of variables in terms of point samples of variables at each grid point and its immediate neighbours. Those derivatives appearing in the governing equations are replaced by finite differences yielding an algebraic equation for the values of variables at each grid point."

Early numerical solutions for flow past a cylinder were introduced by Jaime *et al.* (1969) using quadrilateral grids. However, Taylor series truncation errors are inherent in the finite difference method, which also lacks the flux conserving properties of integral formulations. Two natural procedures to reduce the truncation error are to use sufficiently small spacing or higher order derivatives, both at a cost of greater CPU time and storage. Another problem is to avoid convective instability, and methods such as upwind differencing, the Lax-Wendroff scheme and the QUICK scheme (as described by Leonard (1980), Abbott and Basco (1989) and Wan *et al.* (1996)) help to do this.

Over the past decade various finite difference methods for incompressible viscous flows have been developed. Extensive results can be found in the literature. Some researchers proposed mixed methods for improving the numerical results, e.g. Zhang and Dalton (1998) combined finite difference and spectral methods to calculate steady approach flow past a fixed circular cylinder with low Reynolds number. A finite difference procedure was obtained in the radial direction because of its relative simplicity to implement while Fourier spectral approximation was applied in the axial direction, due to its periodic boundary conditions. A hybrid conservative finite difference and finite element scheme is proposed for the solution of the unsteady incompressible Navier-Stokes equations by Le *et al.* (1997). The scheme is derived from the finite element discretization obtained using the Galerkin method with piecewise bilinear polynomial basis functions defined on quadrilateral elements.

2.2.3 Finite volume method

Finite volume formulations involve discretisation of the flow domain of interest and then integration of the flow equations over each elemental volume. The finite volume technique enables correct flux transport across cell boundaries and conserves momentum throughout the grid.

One method of dealing with the pressure-velocity coupling in the Navier-Stokes equations is to use pressure correction. The origin of most pressure correction methods lies in the Semi-Implicit Method for Pressure Linked Equations (SIMPLE) algorithm introduced by Patankar and Spalding (1970), which involves solving the pressure equation using pseudo velocities to obtain the pressure field. The pressure correction equation is then solved to correct the velocities without further correcting the pressure. Kelkar and Patankar (1992) used a finite volume method based on SIMPLE for prediction of vortex shedding behind a square cylinder. Okajima *et al.* (1992) have applied SIMPLE to the solution of laminar flows, and a finite volume method incorporating QUICK for approximation of the convective terms for turbulent flow simulations past rectangular cylinders. Sivaloganthan and Shaw (1988) reported that the SIMPLE scheme appears to be robust over a wider range of flows than its competitors.

The finite volume method was first used on unstructured grids during the mid-80's when Jameson *et al.* (1986) presented full Euler explicit flow simulations for flow over an aircraft. Triangular grids are used as computational grids for the finite volume method by Jiang *et al.* (1994). Unsteady flow over a circular cylinder is simulated with collocated variable grids. To avoid checkerboard oscillations in the velocity and pressure solution, fourth order pressure terms are added. Solutions using the finite volume method with grids composed of triangular and quadrilateral cell elements are given by Thomadakis *et al.* (1996). This method adopts semi-staggered grids where pressure is held at the cell centre and the velocities at the cell vertices. The Rhie-Chow (1983) interpolation scheme was modified for use in the pressure correction equation to avoid oscillations appearing in the result. Unsteady flow was simulated by Thomadakis *et al.* (1996) for two-dimensional configurations: cavity flow and backward-facing step flow. A finite volume method, for which the number of faces on the cell elements can

vary between three and six, is used by Davidson (1996), wherein unsteady flow is simulated using collocated variable grids to calculate numerical solutions to two-dimensional configurations: backward-facing step, diverging channel and skewed driven-cavity flows.

More recently Greaves (1998) used quad-tree grids directly as computational grids for the adaptive finite volume solution of the Navier-Stokes equations in two-dimensions. Speares and Berzins (1997) presented three-dimensional adaptive tetrahedral grids for the solution of problems in gas dynamics using a second-order accurate finite volume method. Chan and Anastasion (1999) have used a cell centred finite volume method on unstructured triangular grids, with second-order upwinding applied to the convective term to obtain solutions to flow over a backward-facing step and flow past a cylinder. Minyard and Kallinderis (1998) used a parallel finite volume method for the Navier-Stokes equations with adaptive hybrid prismatic and tetrahedral grids.

The MAC method was first described by Harlow and Welch (1965), in which the pressure for each cell is obtained by solving a Poisson equation describing it. The source term is a function of the velocity, derived by substituting the momentum equations into the continuity equation. The full Navier-Stokes equations are then used to find new velocities throughout the grid. The MAC method was used by Viecelli (1969), Schumann (1976) and Naruhn *et al.* (1976).

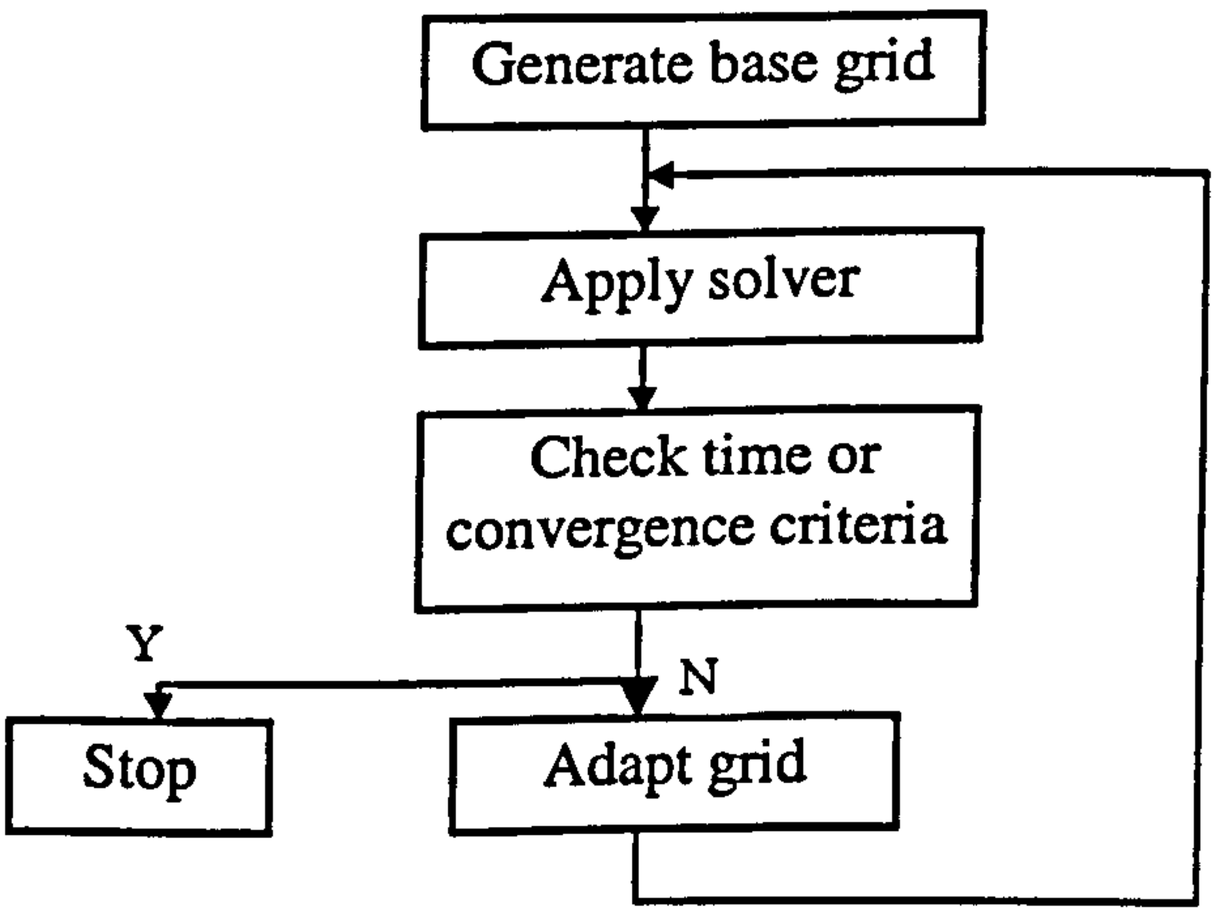
In the work presented here, the finite volume method is applied on computational grids, composed of tri-tree elements. Both the SIMPLE pressure-velocity coupling scheme with collocated variables following Davidson (1996) and the MAC algorithm are adopted here for solving the Navier-Stokes equations. Steady incompressible laminar fluid flow is simulated for the following two-dimensional configurations: channel flow, backward-facing step flow and flow past a circular cylinder. Unsteady fluid motion is simulated for flow past one, two and three circular cylinders. The formulations and results are presented in Chapters 4, 5, 6 and 7.

2.3 Adaptive Grid Generation

The initial grids used for simulation of viscous flows may not always yield the desired accuracy for the solution, so that improvement of the grid may be necessary. Adaptive grids can provide more accurate solutions with less cost by automatically clustering elements around flow features of interest such as vortex shedding, boundary layers, reattachment points and wakes. Therefore, adaptive grid generation is a technique that is being increasingly used in CFD applications, as the flows of interest are increasingly complicated.

In the first stage, a grid of the domain is created by one of the methods proposed previously (see Section 2.1), after which a computational method for solving the problem is applied (see Section 2.2). After choice of a pertinent criterion involving the solution, the initial grid of the domain is regenerated by *refinement*, *derefinement* or local *remeshing*. This is generally an iterative process with the application of adaptation and solvers being simultaneous until some objectives are achieved. The complete numerical simulation can be expressed in the following algorithm:

Algorithm:



There are several ways of achieving adaptive grids. George (1991) and Weatherill and Marchant (1994) have concluded that grid adaptation techniques fall into the categories of (i) P-refinement, where an increase in the degree of the polynomial interpolation function leads to improved accuracy; (ii) R-refinement, where the number of nodes

remains fixed but the nodes are relocated to improve resolution around the features of interest, discussed above; (iii) H-refinement, where points are added or deleted; and (iv) remeshing methods where, for given information on a grid, a completely new grid is derived from the information. In addition, nested grids and multi-grid techniques are further possibilities, a comprehensive study of which has been given by Hackbusah and Trottenberg (1982). Some research, such as that conducted by Weatherill and Marchant (1994) used a combination of H-refinement and remeshing, both of which are controlled by given input parameters.

Adaptive grid generation can make use of the local flow field solution itself to determine where high spatial grid resolution is required and then employ some strategy to increase the grid resolution in those regions. This enables the high spatial grid resolution to be concentrated around the important flow features rather than being wasted on parts of the computational domain where the flow activity is relatively unimportant. There are many indicators for adaptive grid criteria, e.g. error indication, pressure and Mach number, density gradients, element Reynolds number, seeding points and vorticity parameter. How these criteria are chosen has important consequences for the overall operation of the adaptive solver and will be discussed below.

2.3.1 Error indication

An error indicator may be used to find the elements within the grids which require refinement. Berzins (1992) studied gas dynamics using a tetra-tree grid adaptation local solution error indicator as a means of controlling the grid adaptation. Yahia *et al.* (1996) use a directionally adaptive methodology using an edge-based error estimate on quadrilateral grids.

The error estimator used by Yahia *et al.* (1996) was first introduced by Zienkiewicz and Zhu (1987) and is based on the observation that the diffusion fluxes of the finite element are discontinued across element faces while the exact fluxes are continuous. The difference between the two is a measure of the accuracy of the numerical solution. Zienkiewicz and Zhu (1992) have shown that the exact derivatives can be replaced by a

continuous least square approximation: $\|e\| = \sum_{i=1}^{m} \|e\|_i^{1/2}$, where i represents an element contribution, m the total element number and e evaluates the various error forms. Indeed for an 'optimal' grid they shall generally try to make the contributions to this error estimate formula equal for all elements. Pelletier *et al.* (1997) describes two error estimators: a projection error estimator and a local PDE (Partial Differential Equation) problem for the error. The former approximates the true fluxes by a least square projection of the finite element fluxes, while the latter derives the velocity, pressure and temperature errors directly from the Navier-Stokes equations. Ilinca *et al.* (1998) used the error estimator for solving incompressible turbulent flows with H-refinement strategy.

Weatherill and Marchant (1994) used error indication to generate additional points and then connected the points with a local Delaunay algorithm. Weatherill and Marchant's (1994) adaptive procedure involves firstly setting three error indicators: the negative and positive components of the gradient in the direction of the velocity vector and the magnitude of all the gradients in all directions normal to the velocity vector. These error indicators are then converted into grid generation sources that, as indicated, locally reduce the relative element size during grid generation. The grid generation sources are created at grid points and minimum spacing is required where an error indicator is greater than the corresponding error limit; if it is less, then no sources are created and the grid is unaltered.

Sampaio *et al.* (1993) have used an error estimator for their adaptive finite element solution to the Navier-Stokes equations. However, Strouboulis and Haque (1992) have shown this approach is not very robust, because the error estimate obtained by computing may not be convergent due to the error estimate formula itself. Examples are given in Strouboulis and Haque (1992).

2.3.2 Pressure and Mach number

Mavriplis's (1991) refinement criterion is based on the local pressure gradients and Mach number for computing compressible turbulent viscous flows about aerofoil geometry in two-dimensions.

Pressure gradients provide a good indication of inviscid flow phenomena such as shocks and expansions, while local Mach number variations can be used to identify viscous phenomena such as boundary layers and wakes. The variations of local pressure and Mach number are first examined by Mavriplis (1991) within each triangular element of the grid. If these are larger than some fraction of the average variation of pressure and Mach number over some cells of the grid, then new points are inserted into the grid by locally restructuring the grid according to Bowyer's (1981) algorithm in stretched space. Subsequently, the existing structure in this region is removed and a new structure is created by joining the new points to all the vertices of this polygonal region, as shown in Figure 2.11. This is an adaptive H-refinement strategy and based on a modified Delaunay triangulation technique by Mavriplis (1991).

However, the Delaunay maximum and minimum angle criterion (see Mavriplis (1989)) is only applied in the stretched space, where only new points along existing grid stations are permitted and where a new boundary point is inserted. In other words, for curved boundary surfaces this will not coincide with the original boundary grid edge, so that the removal of all grid edges in this region is required and the new structure has to adapt to replace the previous grids. This can be considerably more complex and inconvenient than, for example, tri-tree adaptive grid in which cells can be added or removed locally within the tree structure.

2.3.3 Density gradients

Speares and Berzins's (1997) tetrahedron-based, H-refinement strategy for the solution of problems in three-dimensional gas dynamics uses a density based adaptive indicator. The Euler flow solver is combined with the adaptive algorithm by flagging regions of the grid with low or high-density gradients for *definement* or *refinement*, with the calculation

of local flow gradients being performed across element faces. Where the face normal density gradient falls below or exceeds a chosen tolerance, the edges on the face are flagged to define or refine. In addition, a 'safety layer' of refinement flagging is employed to ensure a directional refinement, simply by defining a second type of regular refinement based upon a directional dissection of the parent tetrahedron. However, the density gradient indicator only applies to compressible flow simulation.

2.3.4 Element Reynolds number

Wille's (1998) adaptive multi-grid generation is determined by the element Reynolds number, which is calculated from the nodal values and weighted by the basis functions evaluated at the geometrical centre. The size of the element Reynolds number, Re_e, indicates the degree of non-linearity in the equation system. An element will be refined if the element corresponding Reynolds number is above the refinement limit, the value of which is given based on experimental observation. Otherwise an element is made terminal. Wille (1997) explored the adaptive limit of the element Reynolds number using a finite element solver of the Navier-Stokes equations for driven cavity flow.

2.3.5 Seeding point

The use of seeding points, as recommended by Gáspár *et al.* (1991), is another possible adaptation technique. In this case, the adaptation is controlled by the seeding points, which adjust their location according to the updated flow velocity components after each time step. The refined grid is determined by the new position of the seeding points. This method involves refining the base grid through the addition of new cells according to their locations.

However, the seeding point method is likely to be computationally expensive, due to the cost of advection of the seeding points at each time step, which require interpolation. Greaves (1995) compared the seeding points and vorticity methods (see below) as adaptive indicators to simulate flow past a circular cylinder with the finite volume method. The conclusion is that the vorticity method adapted more of the grid than the slow-moving seeding point method.

2.3.6 Vorticity parameter

Greaves (1995) has developed grid adaptation based on a vorticity parameter for solving the Navier-Stokes equations with quad-tree grids. An example of which is the adapted quad-tree grid for fluid flow past a square cylinder at *Re*=250 shown in Figure 2.12 (see also Greaves (1998)). This is achieved through the addition of new cell elements in regions of high absolute vorticity and removal of cell elements not included in the base grid in regions of low absolute vorticity. Maximum and minimum vorticity limits are specified in advance, and, after each time step, the vorticity associated with each cell element is calculated. If the absolute vorticity in a given element is greater than the prescribed maximum, then that element is divided, whereas if the absolute vorticity is less than the prescribed minimum, then that element may be removed.

It should be noted that grids of a hierarchical nature are ideally suited to local remeshing, because the refinement and removal of leaf cell elements does not disrupt the tree data structure. This is fast adaptive technology in comparison with those methods previously described and hence saves CPU.

The vorticity parameter as an adaptation indicator, in which the grid is successively remeshed to reach predetermined standards of accuracy, is central to the effective use of the finite volume technique applied in the present work. However, this approach is not a good indicator to grid refinement as one gets closer to the vortex centre. The grid refinement is based on the prescribed maximum vorticity of which the value is predetermined by the observation of grids. Therefore, the optimum values of the prescribed vorticity would require further investigation. Furthermore, as stated earlier, the tri-tree adaptive grid generation is implemented. A tri-tree methodology for flexible, automatic and adaptive grid generation is described in Chapter 3.

CHAPTER 3

THE TRI-TREE GRID GENERATION

Grid generation is an essential pre-requisite for the numerical analysis of engineering problems using the finite volume method. When large size or geometrically complicated problems are considered, grid generation can become a very difficult and time-consuming task. Therefore, developing tri-tree grids to approximate the boundary properly and using fast adaptive grids locally to save CPU time have been considered in this study. Details of these methods are given in the following sections.

3.1 Tri-Tree Grid Algorithm

In the tri-tree grid algorithm, equilateral triangles are used as the basic domain. A diagram of the tri-tree structure is shown in Figure 3.1. An initial equilateral triangle is divided into four new equilateral triangles. Each of these triangles can then be divided into another four equilateral triangles, and so on.

The tri-tree grid generation algorithm has previously been developed and described by Wille (1995). The author stored the information for each grid element in a record consisting of nine integers in two dimensions. These records contain the level of refinement, the node numbers and pointers to the parent and children elements if they exist. The reference numbering system is shown in Figure 3.2.

Numerical methods using hierarchical grid generation have been applied successfully by Kallinderis (1992) and Greaves (1995). These authors used the hierarchical technique of quad- tree grid generation, in which the numbering system can be stored as a single integer (such as suggested by Samet (1990)) in an efficient and easily accessible manner. The Samet-type numbering system was successfully used for quad-tree grids by Greaves (1995) and a similar system is used for the tri-tree grid generation in the present work.

In this work, special treatment is developed to achieve a smooth approximation to curved boundaries. Where single or multiple circular cylinders are included in the domain, elements close to the cylinder are stretched so that nodes lie on the boundary.

Any obtuse angled triangles that result from the technique of elimination of obtuse angles are removed. This treatment ensures good quality elements around the cylinder boundary.

3.2 Samet-Type Numbering System

Every triangle in the tri-tree grid may be identified uniquely by means of its element numbering, T_n. This process is illustrated in Figure 3.1. T_1 is divided into four triangular elements T_2 to T_5 (first level), and then these are subdivided into T_6 to T_{21} (second level) and so on. It should be noted that the element numbering is implemented concurrently for each element in turn.

Each element also has a reference number which indicates the position of the element within the tri-tree hierarchical data structure. At generation level, NL, the reference number of parent element can be worked out from those of its children, or the other way round.

The reference numbering system for this study is adopted from Samet (1990), which enables the reference numbers to be stored efficiently as a decimal integer. The reference numbering system can be summarised as

$$N = \sum_{i=0}^{m-1} N_i K^i \tag{3-1}$$

where m is the division level of the grid and N_i takes the integer value 1, 2, 3, or 4 depending on the position in which the element is located within its parent triangle, as shown in Figure 3.3(a) for a rightward facing triangle and 3.3(b) for a leftward facing triangle. The tri-tree grid reference numbers are stored as the sum of successive powers of K. In this case, $K = 5$ is used for tri-tree decimal grids in two dimensions. Similarly, $K = 9$ could be used for tetra-tree decimal grids in three dimensions where eight new elements are generated with each subdivision.

As each division occurs, a given element produces four children in two-dimensions, and the reference numbers of the children are stored in the tree. For example, the division shown in Figure 3.3 results in grid elements of division level 1 having reference numbers 1, 2, 3, and 4 or 1×5^0, 2×5^0, 3×5^0 and 4×5^0 using equation 3.1. When each division takes place, the reference numbers of the four children are obtained by adding to the parents reference number. In Figure 3.4, grid 1 is divided, which results in elements of division level 2 having reference number 6, 11, 16 and 21 or $1\times5^0+1\times5^1$, $1\times5^0+2\times5^1$, $1\times5^0+3\times5^1$ and $1\times5^0+4\times5^1$ using equation 3.1.

Generation Level, *NL*: From equation (3-1) it can be shown the generation level is obtained by successively dividing the reference number by 5 (K=5) until the remaining number is less than 1. For example, in Figure 3.5, the reference number of the element marked * is 121. Then successive divisions by 5 yield:

First division = 24.2
Second division = 4.84
Third division = 0.968,

which is less than 1 and so the operation is halted. Thus three divisions were carried out indicating that the generation level $NL=3$.

Parent element: To obtain the reference number of the parent element of a given triangle is a two-stage process. Firstly, the remainder on division of the given element reference number by 5^{NL-1} (where NL is the generation level of the element) is obtained. This is then multiplied by 5^{NL-1}. The procedure is summarised by,

$$N_p = \mathrm{mod}(\frac{N}{5^{NL-1}})\times5^{NL-1} \tag{3-2}$$

where N_p is the reference number of the parent of a given element, *mod* is the remainder operator, and N is the given element reference number. For example, in

Figure 3.5, the reference number of the element marked * is 121; the generation level is 3, $\frac{N}{5^{NL-1}} = \frac{121}{25} = 4.84$ and the remainder is equal to 0.84. Then the reference number of the parent element gives $0.84 \times 5^2 = 21$ corresponding to Figure 3.4.

3.3 Methodology

The grid generation process consists of the following steps:

1) Define an initial equilateral triangle, within which the desired grid will lie.
2) Define the set of seeding points about which the grid will be generated.
3) Subdivide about the seeding points. If the triangle contains a point, divide the triangle otherwise check the next triangle.
4) Repeat for all points 3) until the maximum division level has been reached.
5) Subdivide all grid elements to the minimum level.
6) Apply face regulation.
7) Eliminate hanging nodes.
8) Apply special boundary treatment around curved boundaries.
9) Apply corner regulation.
10) Reorder element and node numbers and store grid information.

3.3.1 Initial grid

Figure 3.6 shows an equilateral triangle as the root of the tri-tree. A right hand Cartesian co-ordinate system is used, with axes emanating from one vertex of the triangle. Both the reference number and the generation level are zero. The global nodal numbers (p_1, p_2, p_3) and the local nodal numbers (n_1, n_2, n_3) are defined in the anti-clockwise direction.

3.3.2 The seeding points

The triangulation starts with the outer and interior boundaries shown in Figure 3.7. For the case of flow past a circular cylinder, the boundary is circumscribed by the initial triangle and defined by a set of seeding points as a unit square with two small circles. For the interior boundary cylinder, the diameter of the first circle of seeding points is defined by $D_1 = 0.05$ in this instance. Figure 3.8 visualises the arrangement of the seeding points around the cylinder boundary. The diameter of the second circle of seeding points is slightly larger than the first one to allow for the boundary treatment described in Section 3.3.7. The diameter of the second cylinder is defined $D_2 = D_1 + k*2^{-mend}$, where *mend* is the maximum level of refinement and k is a parameter depending on *mend* .

3.3.3 Subdivision to the maximum level

Subdivision on these boundaries is successively performed until the maximum level of refinement is reached. For every point on the boundaries, the tri-tree is searched to find whether or not each triangle contains a point. By looking at the refinement level of this triangle, decision is made on further refinement. This process is an iterative process, which continues until the triangles containing the boundary points are sufficiently refined.

A general algorithm for deciding whether a point is inside or outside a given triangle is adopted from Wille (1995). For example, referring to Figure 3.9, consider point *s* with co-ordinate (x, y) and the triangle defined by vertices *a*, *b* and *c* with co-ordinates (x_k, y_k); for k =1, 2, 3. By joining the point s with vertices *a*, *b* and *c* three additional triangles are created. L_k is the ratio between the areas of the three triangles to the area of Δabc (*e.g.* $L_1 = \frac{\Delta abs}{\Delta abc}$). If $0<L_k<1$ (k =1,2,3) and the sum of L_k is equal to 1, then point *s* is inside the triangle. Otherwise if the sum of L_k is larger than 1, point *s* lies outside and if $L_k = 0$, point *s* lies on a node or a face of Δabc .

Subdivision of the tri-tree grid about the seeding points takes the following steps. The domain is first divided into four subregions or elements as shown in Figure 3.10. Each

element is then considered in turn: it is subdivided to create four new elements if it contains a point and the element is not divided further otherwise. The second subdivision results in the grids shown in Figure 3.11 and the third subdivision is shown in Figure 3.12. The tri-tree algorithm produces selectively refined Cartesian grids with a recognisable hierarchical tree structure. Figure 3.13 illustrates the tree structure corresponding to the grids in Figure 3.12. After that the algorithm is repeated until the maximum division level 9 has been achieved as shown in Figure 3.14. In the mean time, all tri-tree elements with all three nodes outside the square boundary or inside the cylinder are deleted. Elements with two nodes inside the cylinder and the other at a prescribed distance normal (see Section 3.6) to the inner wall, are also deleted and not considered for further processing.

In Figure 3.14, the size of the faces of the finest tri-tree elements is $1/2^8$ (e.g Maximum division level = 9) times the length of the face of the initial triangle. The general formula for this length is

$$d = \frac{D}{2^n}, \tag{3-3}$$

where d is the size of the face of the finest element, D is the length of the face of the initial triangle and n is the level of the maximum refinement.

When a triangle is divided, the co-ordinates of the midpoint on each face are calculated and these will be the nodal co-ordinates of the new children elements. Two nodal numbering systems are being formed in each triangle element. One system is the local node numbering which runs from 1 to 3 counting in the anti-clockwise direction. The other system is the global node numbering which accounts for all grid nodes. When a element needs to be divided, it is necessary to check whether the required new nodes already exist. If they do not, they are added to the list of points. The decimal trees of the two nodal numbering systems are stored.

3.3.4 Subdivision to the minimum level

Additional elements are generated to achieve a minimum subdivision level throughout the grid. Figure 3.15 shows the subdivision of all elements to minimum level 5 and Figure 3.16 shows the close-up of the elements in the vicinity of the cylinder. As in Section 3.3.3, the process of the deleting elements outside boundaries is carried out simultaneously.

3.3.5 The face regulation

After subdivision to the minimum level is achieved, from Figure 3.16, it can be seen that some triangles contain more than one node along one of their faces. This node distribution will not lead to well-formed triangles, which will be disadvantageous when integrating the differential equations over the elements. It is desirable to generate additional grid elements such that the grid size ratio between two adjacent triangles does not exceed two. This constraint is applied in this process to elements sharing a face as shown in Figure 3.18. The face regulation is achieved by considering each grid element in turn. If more than one node already exists along any of the faces of an element, then that element is subdivided. The search is repeated in a recursive manner until no further elements are generated. The triangulation in Figure 3.17 shows the outcome of this procedure.

3.3.6 Elimination of hanging nodes

After face regulation, a maximum of one hanging node on each element face remains. These nodes lie on the faces of adjacent elements, which renders the discretisation of the Navier-Stokes equations across these elements difficult. An advantage of tri-tree grids is that they provide an easy way to deal with this problem. The process is implemented by first searching for an element in which a hanging node exists such as the large triangle illustrated in Figure 3.19(a) which has hanging node P. The equilateral triangle is then split into two right-angled triangles as shown in Figure 3.19(b). Figure 3.20 shows the result of this process applied to the entire grid. After this process, only two types of triangles are used in the domain: equilateral and right angled triangles.

3.3.7 The special boundary treatment around interior boundaries

After the process of deleting the elements outside boundaries, illustrated in Figure 3.21, a special boundary treatment around the interior boundary is used. Two steps are taken, as follows:

The Stretching Technique: First the boundary nodes that lie close to the cylinder are found. Lines are extended from each of these nodes to the centre of the cylinder. The intersection between the line and the cylinder circumference will be the new node position, which fits accurately the boundary curve shown in Figure 3.22. Of course, if the node already lies on the cylinder circumference then it is position is not changed.

Elimination of Obtuse Angles: After the stretching technique, some obtuse angles are often generated in the distorted elements around the cylinder, which causes the accuracy of the numerical simulation to suffer. These large angle elements can increase errors in nodal variables, which change quickly around the cylinder (such as the pressure value). It is necessary to transform these angles from obtuse into acute or right angles. The correction procedure is implemented by firstly identifying these large angle elements such as in Figure 3.23 Δ_{abp}, then defining lines from the nodes (e.g. node a) that are outside the cylinder to the centre of the cylinder. The intersection between the line and the cylinder circumference will be the new node p, which fits the accurate boundary condition and forms two well-formed triangles Δ_{abp} and Δ_{apc} in this instance. Figure 3.24 shows the final result of this treatment applied to the entire grid.

In this way, the special boundary treatment around the cylinder boundary is performed and the quality of these triangles will be discussed in Section 3.6. This technique may be adapted to arbitrary boundaries, using alternative methods to determine the normal direction to the boundaries.

3.3.8 The corner regulation

The generation of additional elements to achieve a side length ratio of 2 throughout the gird is not only applied to adjacent elements but also to those sharing a single node. Such abrupt changes from large to small elements which share a node may cause large errors in interpolation calculation, which are of course unacceptable. Each node is considered in turn. Firstly, the maximum division level of the elements which share the node is determined. Then, elements with a division level two or more below the maximum division level are divided, as illustrated in Figure 3.26. In Figure 3.26(a) node P has elements A and B connected to it which are two division levels different in size and so element B is divided as shown in Figure 3.26(b). After this, face regulation and elimination of hanging nodes must be performed again in an iterative process which stops once no more new elements or nodes are generated. The final grid is illustrated in Figure 3.25 and the entire tri-tree grid is shown in Figure 3.27.

3.3.9 Grid information storage

Element Numbering: When a triangular element is divided to produce four new elements, the parent element number is deleted when the children are generated. It should be noted that the total number of elements varies up and down during the tri-tree grid generation. For example, outside elements are taken out so that the total element number is reduced. During each generation procedure, the number of newly generated elements is added to the record of existing elements.

Nodal Numbering: The above triangularisation technique results in a maximum and minimum refinement level in a given tri-tree grid. Two node numbering systems have to be used for each triangle: local and global node numbering. The local node numbers of an element range from 1 to 3 in the anti-clockwise direction for both rightward and leftward facing triangles as shown in Figure 3.29. The global nodal number is increased when a new node is generated in the refinement process. Figure 3.28 gives a visualisation of global node numbering. The global nodes (p_1, p_4, p_6) belong to element 1 and concurrently correspond to local nodes (n_1, n_2, n_3). Similarly, node p_4

belongs to elements 1, 2, and 4 and corresponds to n_2 for element 1, n_1 for element 2 and n_3 for element 4. If some elements are deleted in the grid generation process, some global node numbers will have disappeared accordingly. After the final procedure is completed, each node is checked to verify if it is one of the nodes of an existing element. If not, this node number is deleted and subsequent nodes have their global nodal numbers reduced by 1.

After the above the procedures have been carried out, the base tri-tree grid is complete. The co-ordinates of the nodes, the local and global node numbering, the boundary indicator information and element reference numbers are all stored. The tree structure so defined is sufficient to completely specify the element remeshing history for the purpose of adaptation. This will be discussed further in Section 3.5.

3.4 Pointer System

In this Section, methods for neighbour finding, grid indexing and for calculating element centre co-ordinates and element areas are described. Almost all quantities (e.g. convection, diffusion and derivatives) are computed by looping over each element or each face and assigned at element centres. The processes of determining these variables are described in this Section. The grid indexing system is based on that described by Davidson (1996).

3.4.1 Centre co-ordinates

The co-ordinates (x_c, y_c) of the centres of the tri-tree grid elements can be calculated from their three nodal co-ordinates. The formulation used is

$$x_c = \frac{1}{3}\sum_{k=1}^{3} x_k \quad \text{and} \quad y_c = \frac{1}{3}\sum_{k=1}^{3} y_k \,, \tag{3-4}$$

where (x_k, y_k), for k = 1, 2 and 3, are the co-ordinates of vertices *a, b, c* of the element shown in Figure 3.30.

3.4.2 The area of an element

The area Δ_{abc} of each element is expressed by

$$\Delta_{abc} = \frac{1}{2}\begin{vmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{vmatrix}, \qquad (3\text{-}5)$$

3.4.3 Neighbour finding

Three neighbours of any triangular element can be determined from the corner co-ordinates. This is a sweeping process through each node of each element in turn. Three neighbours are defined for each element. The link of elements is searched to see which contains the nodes on each of the three faces of the element in question.

It should be mentioned that the calculation of element centre co-ordinates, the area of the elements and neighbour finding in quad-tree grids are all performed by manipulating the quad-tree reference number (Van Dommelen and Rundensteiner (1989)). However, tri-tree grids present major difficulties in implementing this technique. For the example, the case of neighbour finding, determining whether the triangle is rightward facing or leftward facing and the existence of equilateral or right angled triangles, all complicate the tree search. In other words, tri-tree grids do not lend themselves to the use of the same VDR reference number manipulations. The only tri-tree search for parents and children is carried out using the Samet-type hierarchical reference numbering system described in Section 3.2.

3.4.4 Face indexing system

A pointer is required for each face to give information about which grid points (element vertices) form the starting and ending points of the face and which two

elements are the adjacent elements. This information is stored in *listf* (see Figure 3.31), which has the form (see Davidson (1996)):

$$\begin{aligned} listf(i,1) &= il \\ listf(i,2) &= ist \\ listf(i,3) &= iend \\ listf(i,4) &= ir \end{aligned} \tag{3-6}$$

where i is the index of the face, *ist* and *iend* are the starting and ending grid points and *il* and *ir* are centres of the left and right elements respectively (see Figure 3.31).

3.4.5 Element indexing system

When solving the discretised equations, we also need a pointer for each element and its neighbouring elements. The number of neighbours is three for triangular grids. For element i, this information is stored in $listc(i, j)$, which has the form:

$$\begin{aligned} listc(i,1) &= i_1\,(neighbour1) \\ listc(i,2) &= i_2\,(neighbour2) \\ listc(i,3) &= i_3\,(neighbour3) \end{aligned} \tag{3-7}$$

where the neighbouring elements for element i are i_1, i_2 and i_3. $j =$ 1, 2 or 3 are determined by the order in which the connecting faces f_1, f_2 and f_3 are visited when looping over all faces in the calculation domain (see Figure 3.32).

The triangular grid indexing systems are adopted for the present work. However, the systems can also be employed in general circumstances where a element may have an arbitrary number of faces.

3.5 Tri-Tree Grid Adaptation

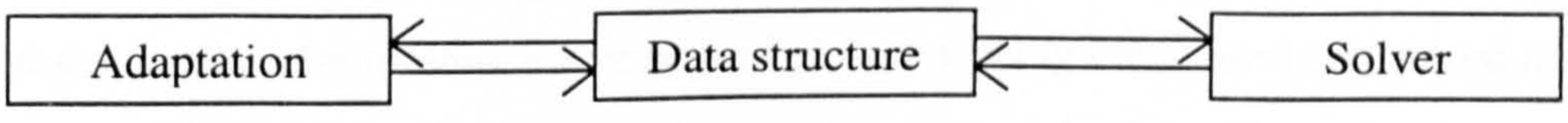
3.5.1 General description

The tri-tree grid algorithm was described in Section 3.1 above. The tri-tree has a naturally hierarchical data structure which is suitable for adaptation. The grid data structure contains all flow and connectivity information sufficient to adapt the grid, either by a local grid *refinement* or by a local grid *derefinement* process.

The grid adaptation strategy assumes that there exists a good quality initial triangle grid of the computational domain, which is taken to be the invariant base grid. The adaptation of the grid is based on a vorticity parameter rather than the seeding points used in base grid generation. The reference process involves adding nodes to this base grid by face and element subdivision. Each change in the computational grid is tracked within the hierarchical data structure. The *derefinement* process is the inverse process to *refinement*, where nodes, faces and elements are removed from the grid leaving the parent element.

The process of grid adaptation is invoked automatically in response to flow solution criteria. It is desirable that refinement regions correspond to regions of significant flow activity requiring increased resolution. Conversely, derefinement is applied in regions of insignificant flow activity where reduced resolution is desired for the sake of CPU efficiency. These criteria have important consequences for the overall operation of the adaptive solver. The complete numerical simulation may be thought of as follows:

Algorithm:



Thus the adaptation and solver can be thought of as two distinct processes that are applied to the data structure. The former alters the grid in response to local flow solution features and the latter advances the flow solution parameters in time, with the application of adaptation alternating, interactively with the solver until the objectives are achieved. Below is given a full account of how the data structure and adaptive algorithm are constructed. The solver algorithm will be described in Chapter 4.

3.5.2 The tri-tree data structure

An advantage of using tri-tree grids is that the hierarchical data structure and numbering system are retained during the *refinement* and *derefinement* operations. We define a set of data structures and a pointer system in the tri-tree grid generation. These are the following:

Element Objects: There are three different data objects associated with each element. One records the node numbers corresponding to its vertices, along with a flag indicating the presence and position of possible boundaries. In a second object, the element indices of its neighbours are stored. Finally, a third data object holds the element reference number plus a flag identifying whether it is an equilateral or a right angled triangle.

Face Object: The face data object contains four data elements for each element: a) the start node number; b) the end node number; c) the element number that lies to the left of the face; d) the element number that lies to the right of the face.

Node Object: This contains the $x-y$ co-ordinates of each node, plus an additional flag to indicate on which boundary, if any, the node lies.

The grid is adapted and the new information stored using the Samet-type numbering system. Data handing during the process of *refinement* is similar to that during base grid generation; the reference numbers of children being calculated and stored and the new element numbers added to the list of element numbering. Extra faces and nodes are created at a given element, nodal and face numbers are added to the list. The process of *derefinement,* however, has no parallel in base grid generation. The reference number of the parent element is calculated and replaces the reference number of the given element and its siblings. The generation level reduces by 1 and the list of elements is shortened. Faces and nodes are deleted if they do not belong to any element in the updated list.

3.5.3 The tri-tree grid adaptation algorithm

Before the grid is adapted, it is necessary to calculate the vorticity value of each element, which is the indicator used for driving the remeshing routines. The vorticity at element i using collocated variables is calculated as follows,

$$\omega_i = \left| (\frac{du}{dy})_i - (\frac{dv}{dx})_i \right| , \tag{3-8}$$

where ω_i is the absolute vorticity value, and $(\frac{du}{dy})_i$ and $(\frac{dv}{dx})_i$ are velocity gradients at the element i of y and x respectively, expressions for which are described in Chapter 4.

The maximum vorticity $\omega_{\max}$ and the minimum vorticity $\omega_{\min}$ values are specified before the time dependent simulation starts. The grid is adapted regularly throughout the simulation at a prescribed number of time steps.

Element Refinement: In this case, if the vorticity of a given element is greater than the prescribed maximum, then that element must be divided. There are two types of subdivision as there are two types of triangle. One is subdivision of equilateral triangles, which is regular division similar to the base grid subdivision shown in Figure 3.33. The other is the subdivision of right angle triangles, which is performed by firstly replacing the two right angle triangles with their equilateral parent element; this is then divided into four elements as explained above and shown in Figure 3.34.

Element Derefinement: In this case, if the vorticity of a given element and all its siblings are less than the prescribed minimum and the element is not at the minimum division level, then its parent element is placed on the list of elements and the children are deleted, as shown in Figures 3.35 and 3.36 with both types of triangle.

Once the remeshing process is complete, face regulation, elimination of hanging nodes and corner regulation are performed. The grid is then reordered, nodes are numbered and the pointer system updated. All these procedures are similar to the base grid generation.

Interpolation within the Grid: The new elements and unchanged elements make up a new entire grid. In order to complete the adapted grid in preparation for the next time step calculation, variables (such as velocities and pressure) are assigned to new elements by interpolating values from their neighbouring elements. In this procedure, values at the unchanged elements are assigned directly, whilst interpolation is used at the newly created elements. For example, Figure 3.37 illustrates the interpolation of pressure to new element i from its parent element (i_2) and two neighbour elements i_1 and i_3. The interpolation is summarised by Franke (1982) (and often called Shepard-type scattered data interpolation) and written as

$$p_i = \frac{\sum_{k=1}^{3} w_k(x,y) \times p_k}{\sum_{k=1}^{3} w_k(x,y)} \quad , \tag{3-9}$$

where p_i is the pressure at new element i and p_k, for k=1, 2 and 3, p_2 is the pressure in the parent element i_2 and p_1 and p_3 are the pressure at neighbour elements i_1 and i_3. $w_k(x,y)$ is an inverse distance weighting function given by

$$w_k(x,y) = [(x_i - x_k)^2 + (y_i - y_k)^2]^{-\frac{1}{2}} \quad , \tag{3-10}$$

where (x_i, y_i) are the co-ordinates of the centre of element i and (x_k, y_k) are the co-ordinates of the centre corresponding the elements described above. The interpolation of u and v velocity components to the new element i is similar. This interpolation is used to calculate the values of the velocity and pressure at discrete vertices of triangular elements from its related cell-centre values. A complete set of discrete equations is assembled for those variables by enforcing the appropriate conservation laws on control volumes constructed for each face and node. The influence of flow direction for each node should be considered when using interpolation scheme. For simplicity, the average influence of flow direction for the node is used by the interpolation. For high Reynolds numbers, higher order interpolation taking into account flow direction has to be employed.

The adaptive algorithm is a dynamic procedure and the grid is adapted to the instantaneous solution. The above procedure is repeated until no further change is generated (for steady flows) or some objectives, such as number of iterations or time-steps are achieved for time dependent flows. For example, Figure 3.38 shows the final tri-tree adapted grid in steady flow past a cylinder at Re=40. Further runs through the grid adaptation algorithm produced no changes to this grid.

3.6 Tri-Tree Grid Quality Behaviour

Only two types of triangles have been consistently generated in both base and adapted grids. The major advantage related to tri-tree grids is that refinements during grid adaptation causes no degradation in element quality.

In grid generation, it is often considered essential for accurate solution that no angle of any triangle in the grid should be small. The tri-tree algorithm automatically ensures that no angle in a triangulation is near 180 degrees since only equilateral and right angle triangles are used in the initial tri-tree grid.

Attention is drawn to a small number of distorted triangles around the cylinder wall, whose quality has significant influence in flow simulation. The stretching technique is used to avoid large angles appearing in any element and the process for elimination of obtuse angles guarantees that right angles are normal to the wall surface.

The key to the stretching technique is making a decision on the critical distance normal to the actual cylinder wall. In other words, which elements close to the cylinder wall should be deleted or retained. Figure 3.39(a) shows a point s_1 lying on the cylinder circumference and located inside element i_1, which has two nodes inside the cylinder (a_2 and a_3) and the other a_1 in the opposite direction at distance a_1s_1 from the wall. An optimum ratio is used to decide whether element i_1 is deleted or retained. The formula of this ratio, r, is

$$r = \frac{d_1}{h} \quad , \tag{3-11}$$

where d_1 is the distance $s_1 d$ and h is the distance $a_1 d$. The cross point d is the intersection between the extension of line $a_1 s_1$ and the face $a_2 a_3$.

If the ratio value r, based on the equations derived in this project, is large than 0.423 (optimum value), element i_1 is deleted as shown in Figure 3.39 (b); a_1 is then stretched to s_1 so element i_2 is distorted. It is easily seen that this distortion produces no angles larger than 90^0. If r is less than 0.423, the element i_1 is retained as shown in Figure 3.40; a_2 and a_3 are stretched to s_2 and s_3 so that elements i_1, i_2 and i_3 are distorted.

Consider Figure 3.41 which shows triangle i_1 before and after distortion. Since i_1 is equilateral prior to distortion, we have

$$a / h = 1/\sqrt{3} . \tag{3-12}$$

In order to avoid obtuse angles which impair the quality of the solution, angle β should not exceed 90^0. This implies that

$$\frac{a}{h - d_1} < \tan 45^0 \quad \Rightarrow \quad a < h - d_1 . \tag{3-13}$$

Introducing (3-12) into (3-13) yields,

$$\frac{h}{\sqrt{3}} < h - d_1 \quad \Rightarrow \quad d_1 < 0.423h . \tag{3-14}$$

Hence, r=0.423 will be the deciding value: if $d_1 / h >$ 0.423 the triangle is deleted, otherwise, it is retained and subsequently stretched.

However, the combination of successive element distortions may create obtuse angles. As a result, once the stretching procedure is completed, the elements surrounding the cylinder boundary are searched for obtuse angles, which are then eliminated by modifying the element geometry, as shown in Figure 3.23 and described in Section 3.3.7.

This special treatment around the cylinder boundary is implemented to improve the quality of the distorted triangles. The procedure is only used during base grid generation, because the elements close to the cylinder are not altered during adaptation of the grid.

3.7 Closure

The tri-tree grid generation algorithm and adaptive grid generation algorithm have been described. The tri-tree grids have been chosen as the basis of this work because they are quick to generate, approximate curved boundaries well and can be readily adapted. It would be most desirable for the calculation grid to adapt automatically according to local flow vorticity given by the solution at each time step. This will be demonstrated in Chapter 7 for flow past circular cylinders with tri-tree adaptive grids.

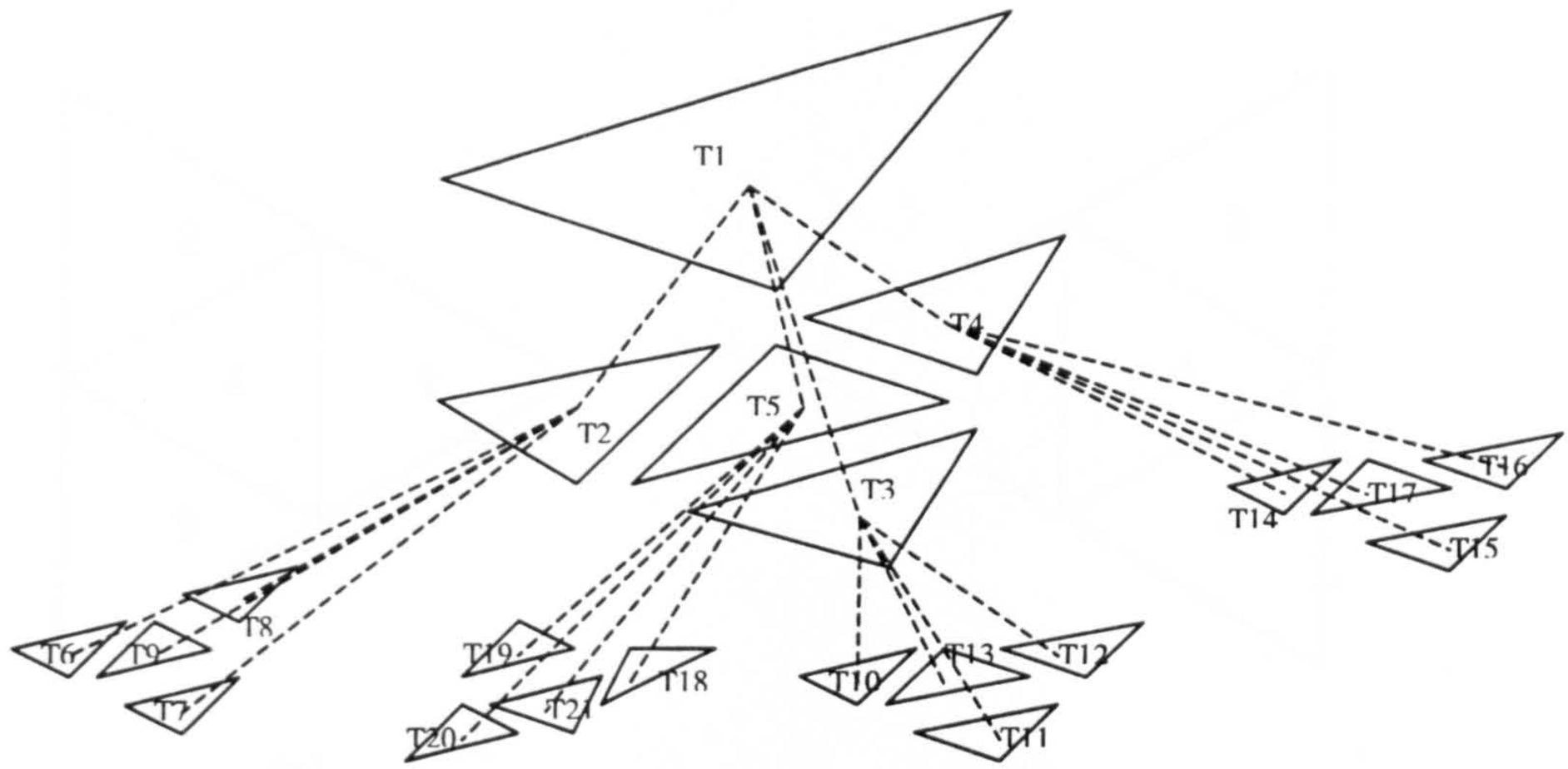
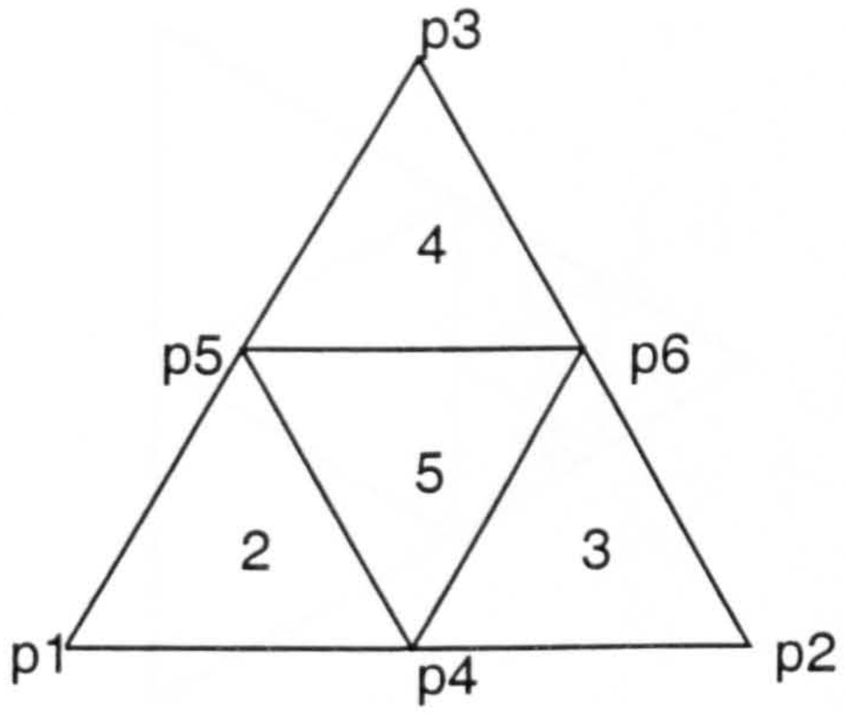


Figure 3.1 Tri-tree structure showing subsequent divisions



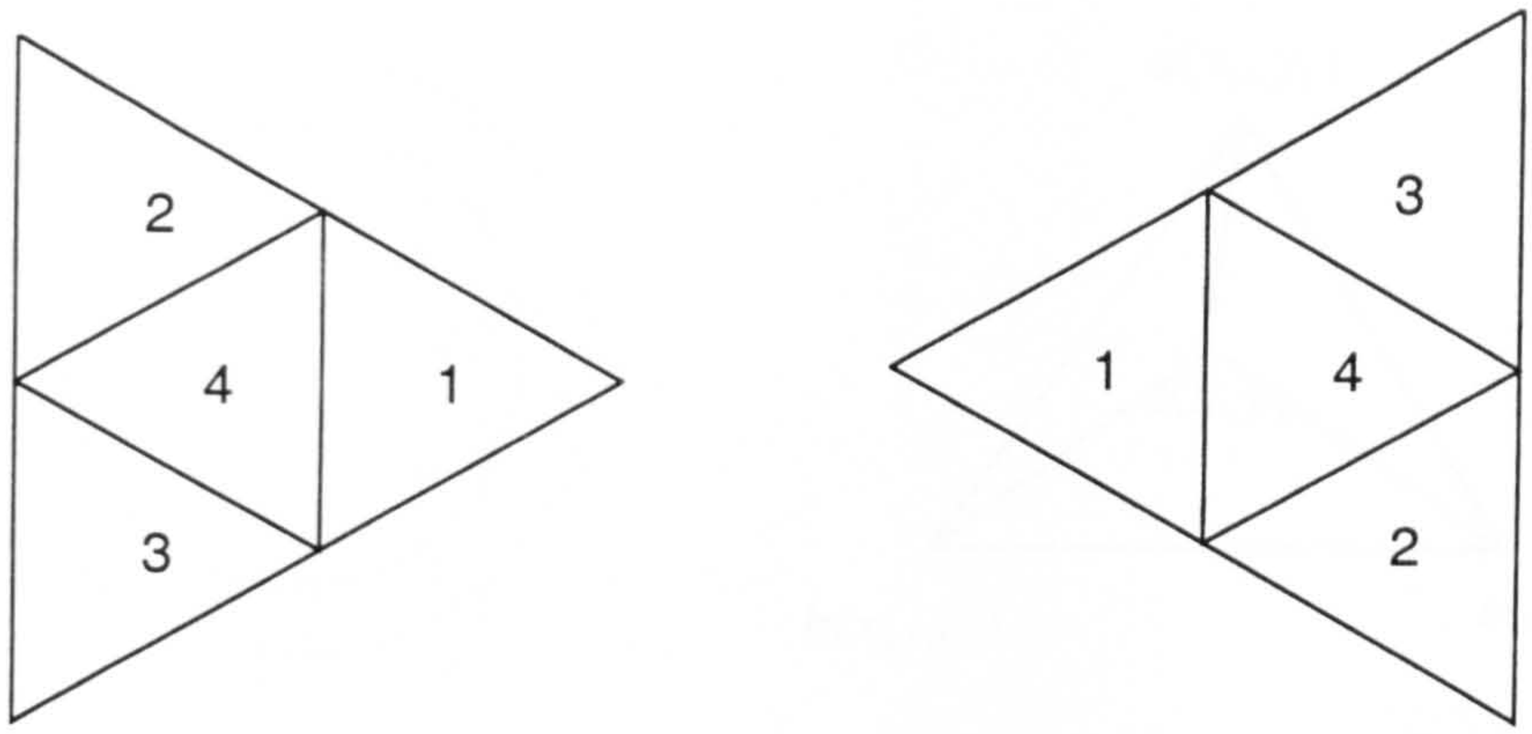
1	*1*	P_1	P_2	P_3	*2*	*3*	4	5	0
2	-2	P_1	P_4	P_5	*5*	*0*	0		1
3	-2	P_4	P_2	P_6	*0*	*5*	0		1
4	-2	P_5	P_6	P_3	*0*	*0*	5		1
5	-2	P_6	P_4	P_5	*2*	*4*	3		1

(a) *(b)*

Figure 3.2 The tri-tree reference number by Wille (1992)

(a) Initial equilateral triangle consists of the corners $T_1 = (p_1, p_2, p_3)$***, which is divided into four new triangles (b) Information in the tri-tree structure***

Wille (1992) noted that "This information is contained in a record consisting of nine integers. The first integer describes the level of refinement. When this integer is negative, it indicates a terminal triangle. The next three integers are the indices to the corners of the present division. If the refinement level integer is positive, the next four integers are pointing to the triangles into which the triangle is refined. If the triangle is terminal, the following three integers are pointers to the neighbour triangles. If one of these integers is zero the triangle has no neighbour in that direction. The last index is pointing to the parent triangle."



(a) (b)

Figure 3.3 (a) The local integer value for a rightward facing triangle (b) The local integer value for a leftward facing triangle

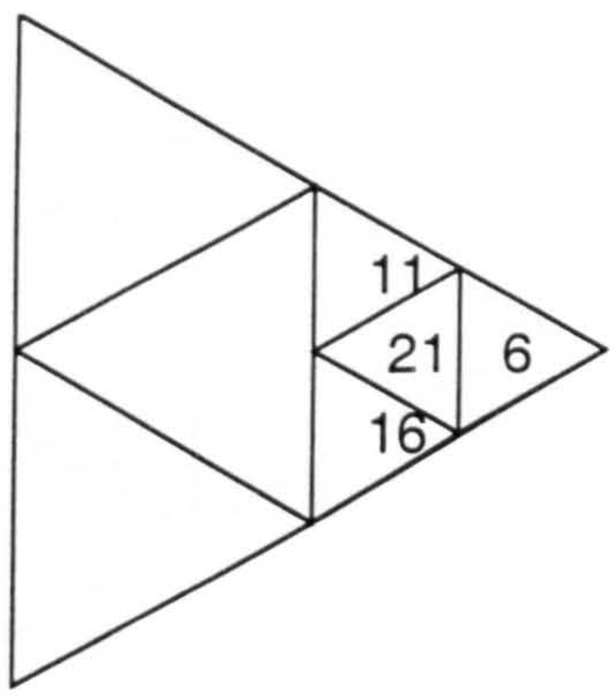


Figure 3.4 The reference numbers

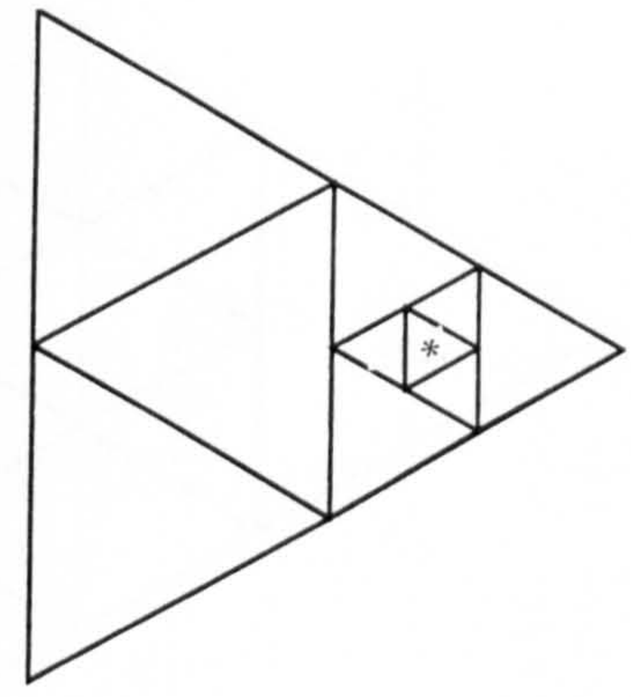


Figure 3.5 Example tri-tree grid

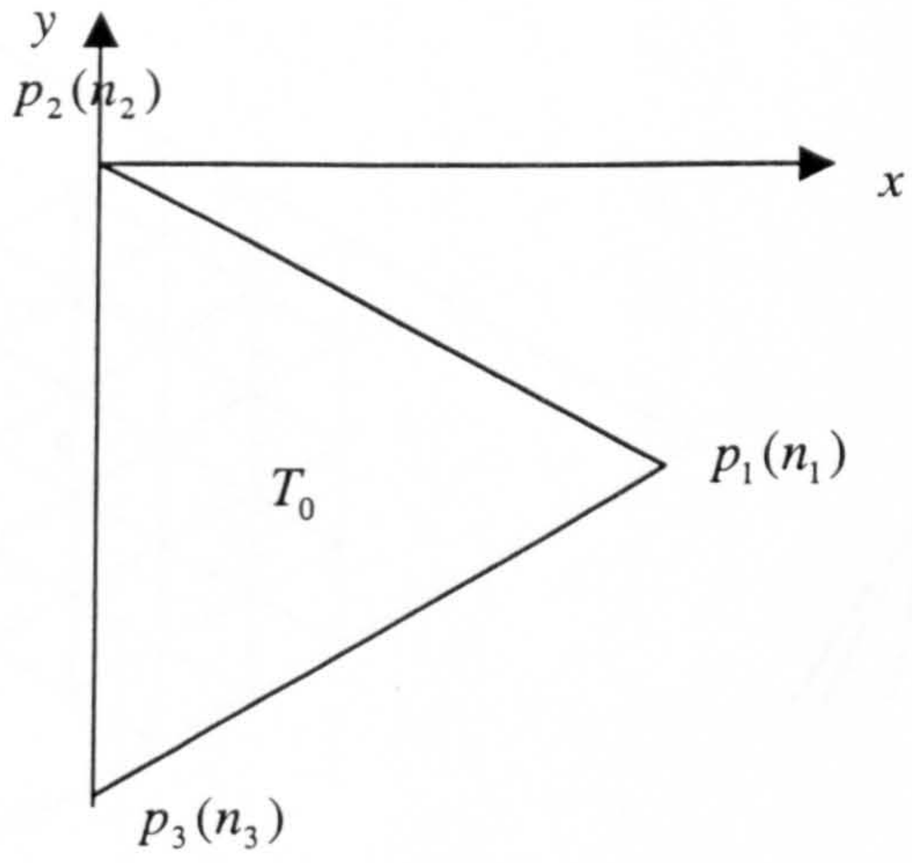


Figure 3.6 Initial equilateral triangle

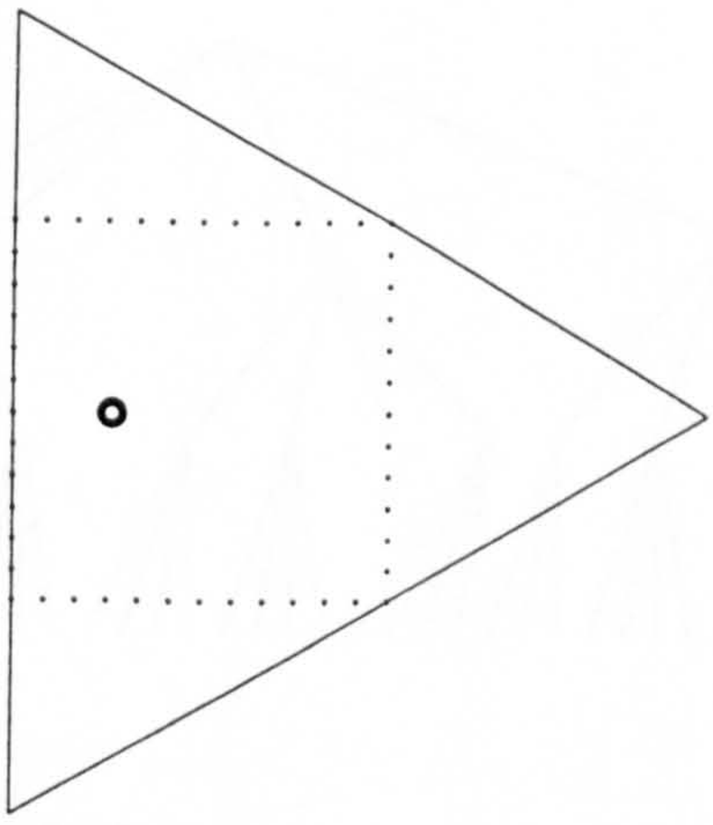


Figure 3.7 Seeding points (note that the interior circles are magnified in Figure 3.8)

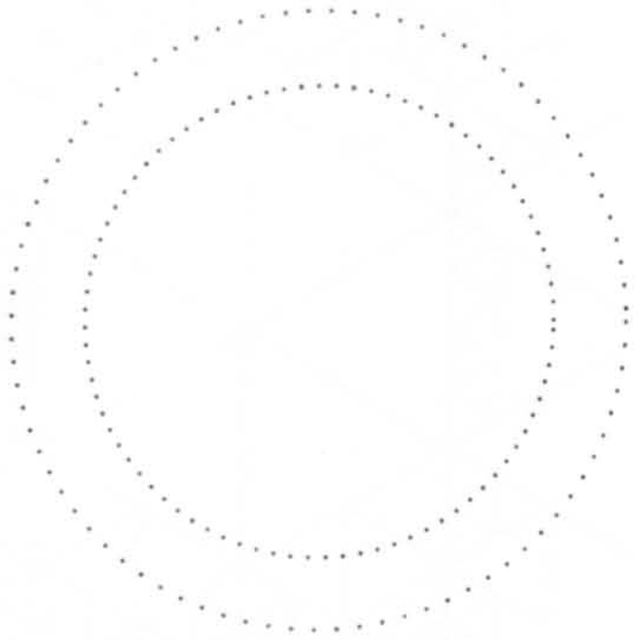


Figure 3.8 The two circles seeding points

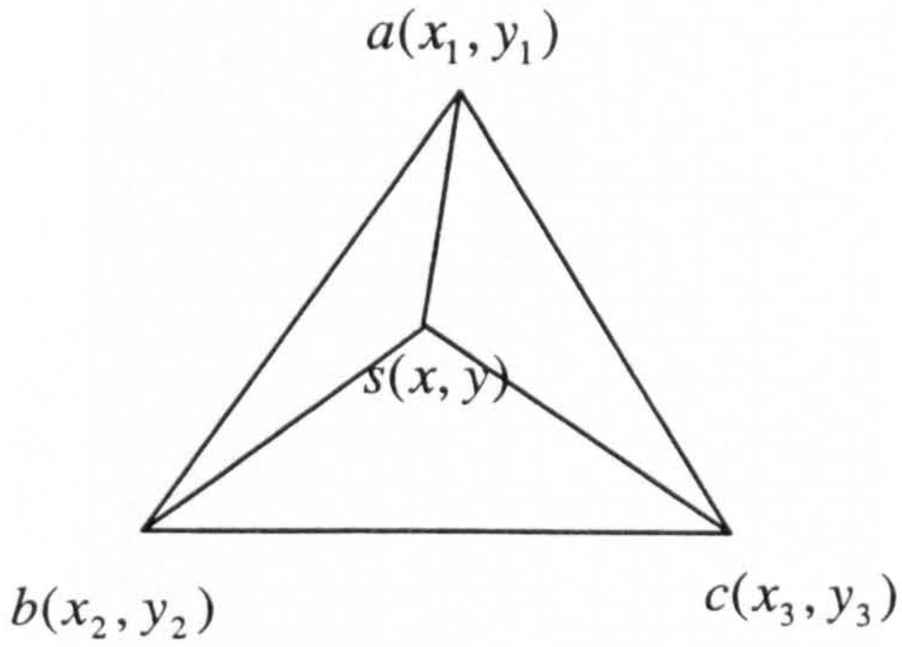


Figure 3.9 A seed point in the triangle

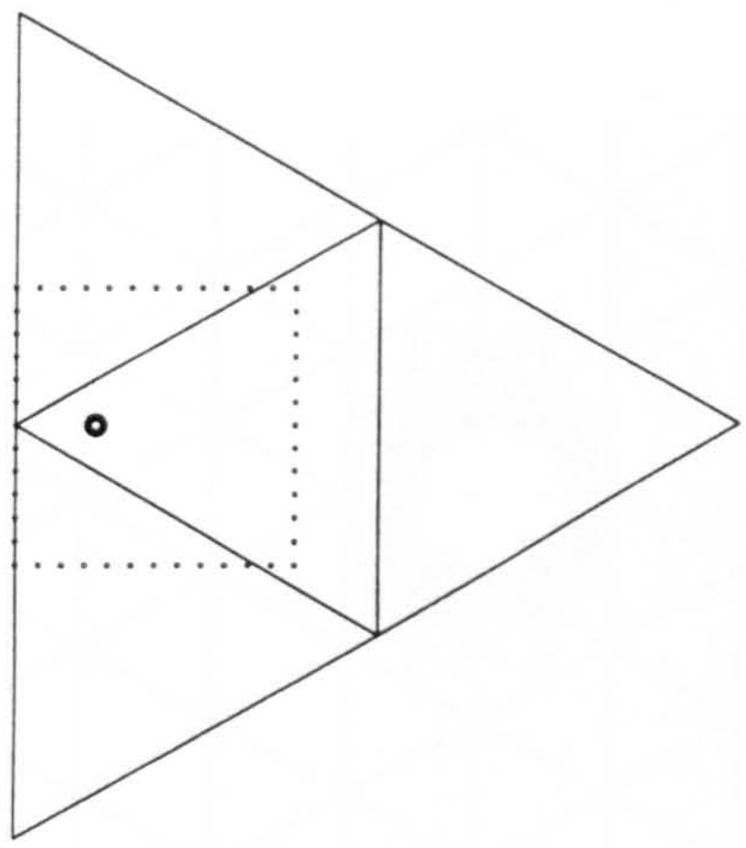


Figure 3.10 First division

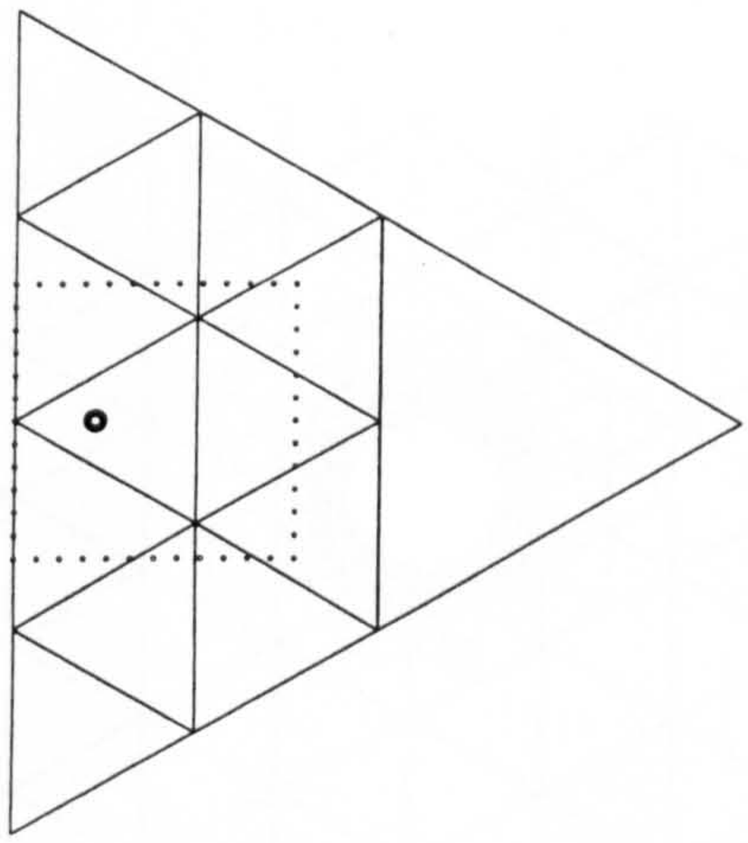


Figure 3.11 Second division

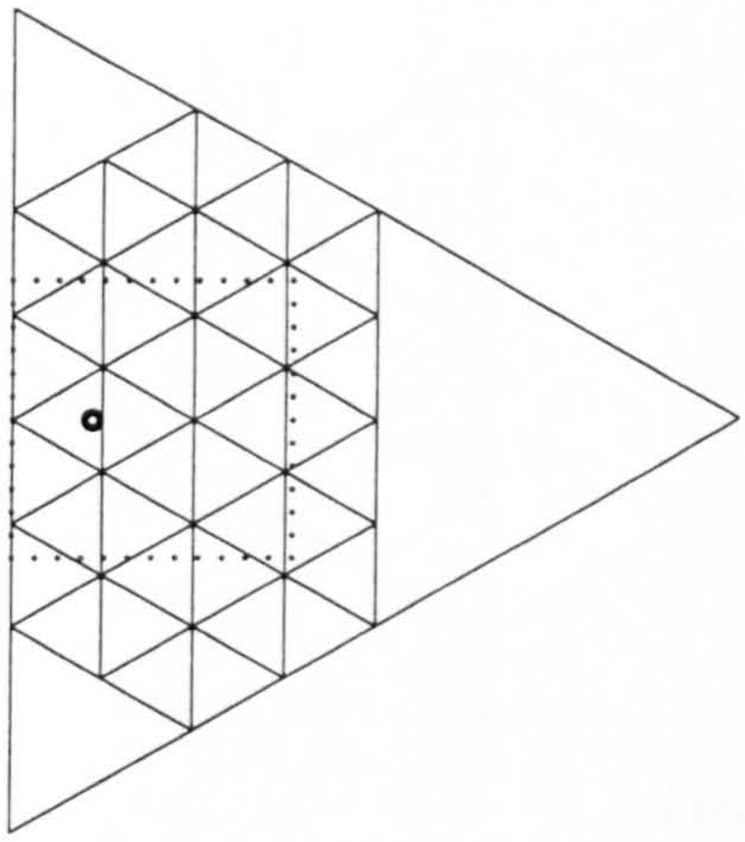


Figure 3.12 3rd division

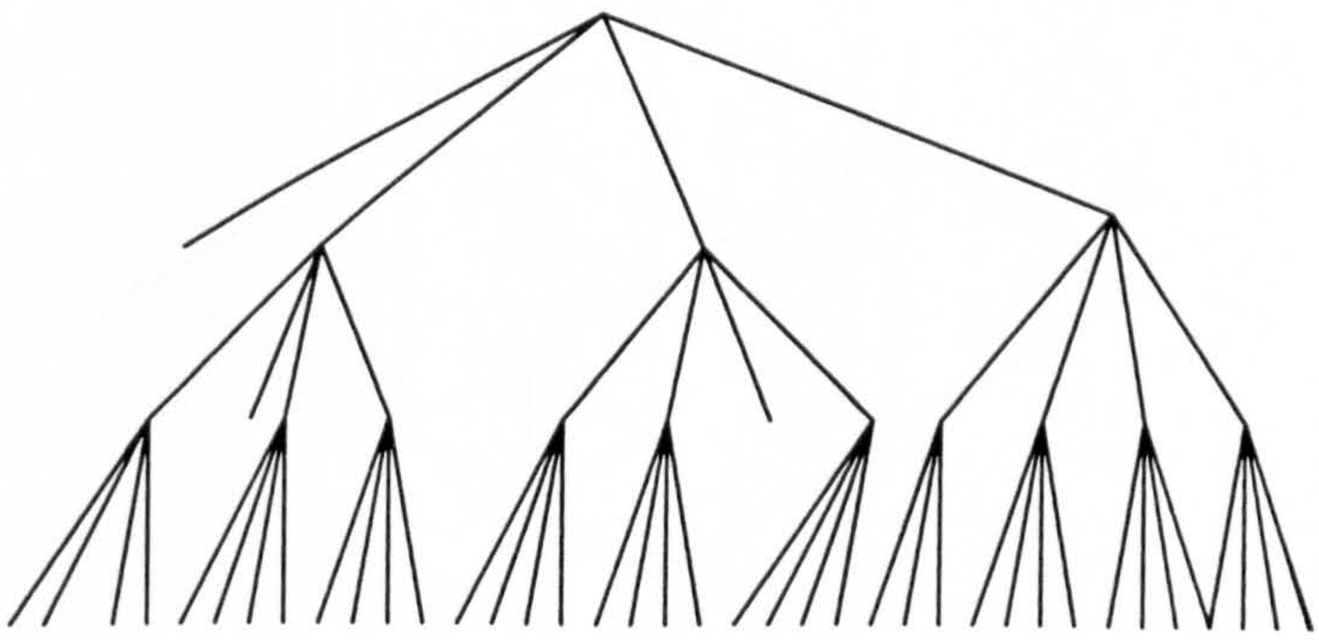


Figure 3.13 Tree structure

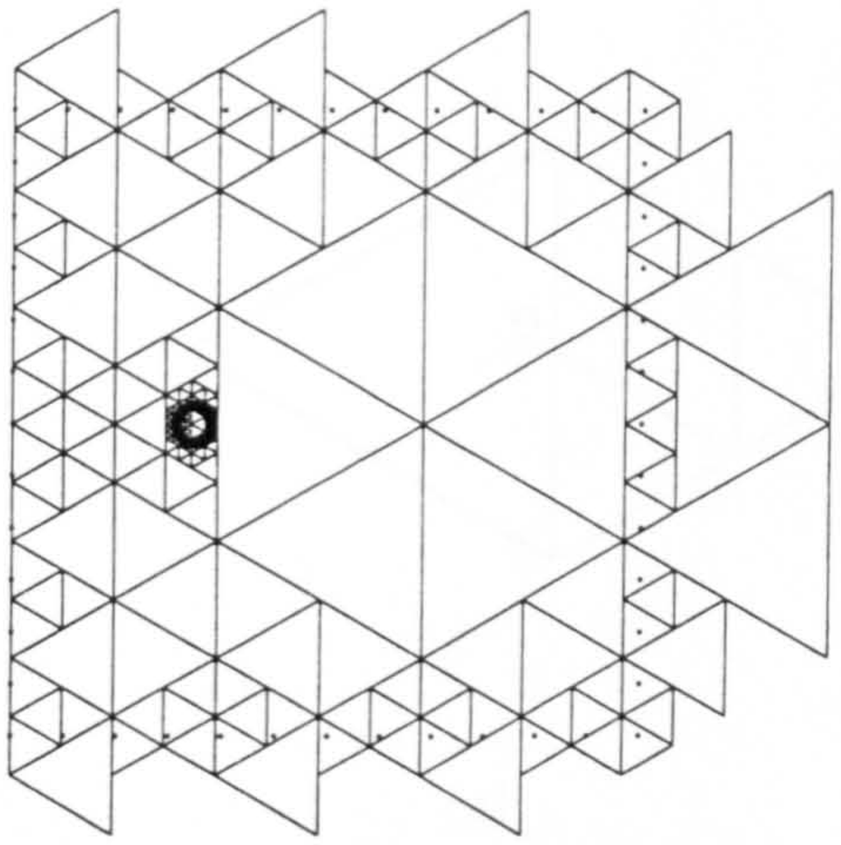


Figure 3.14 9th division

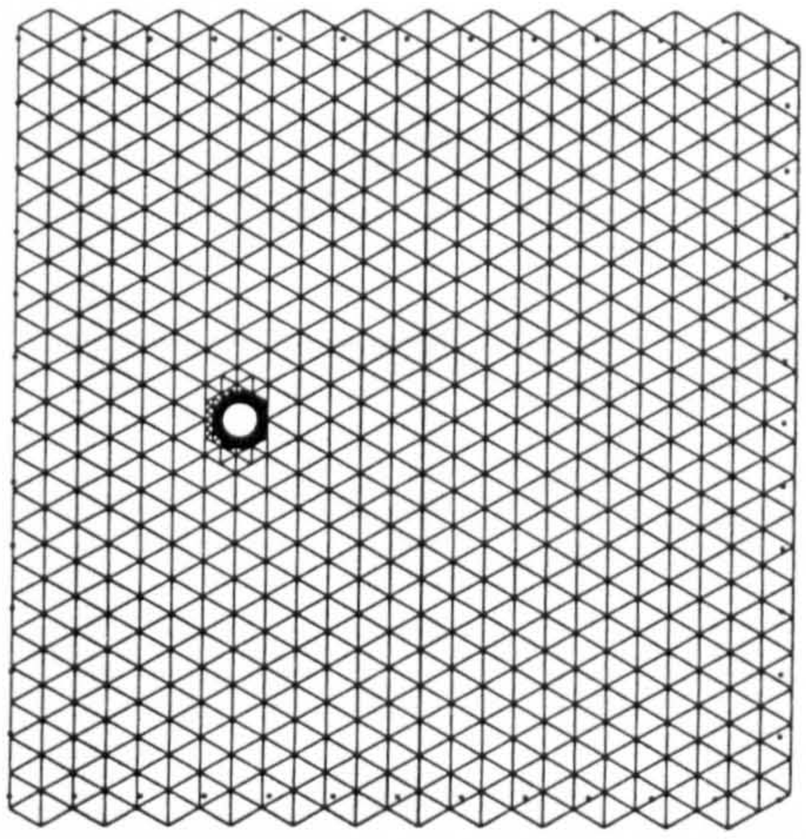


Figure 3.15 Application of minimum level 5

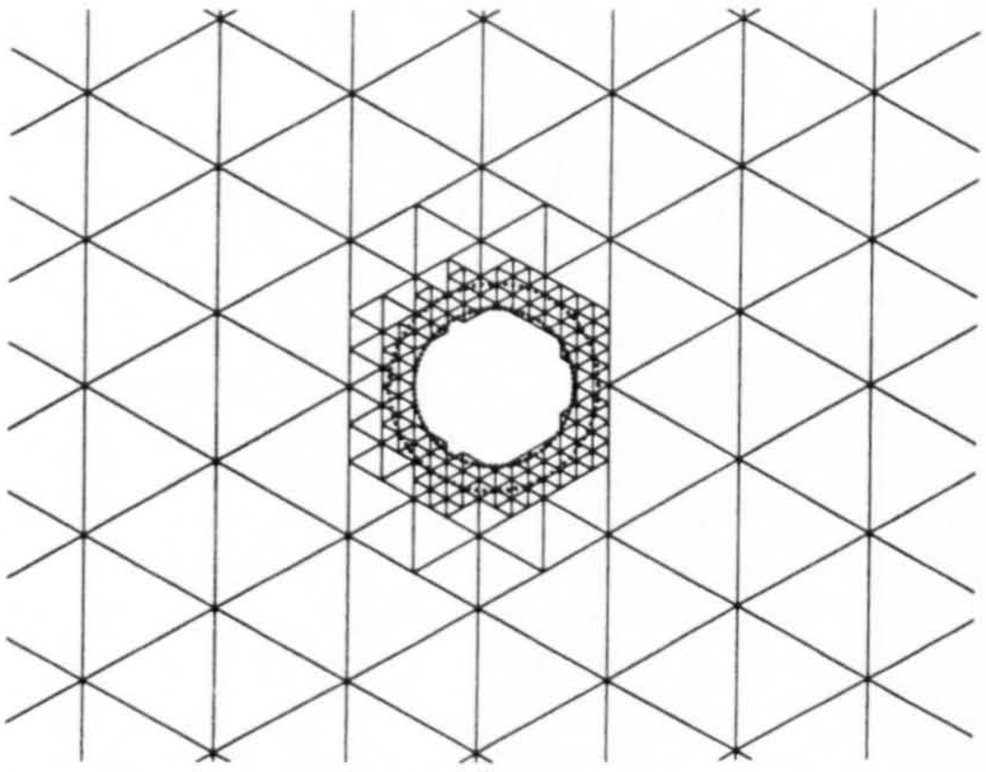


Figure 3.16 Close-up of the cylinder

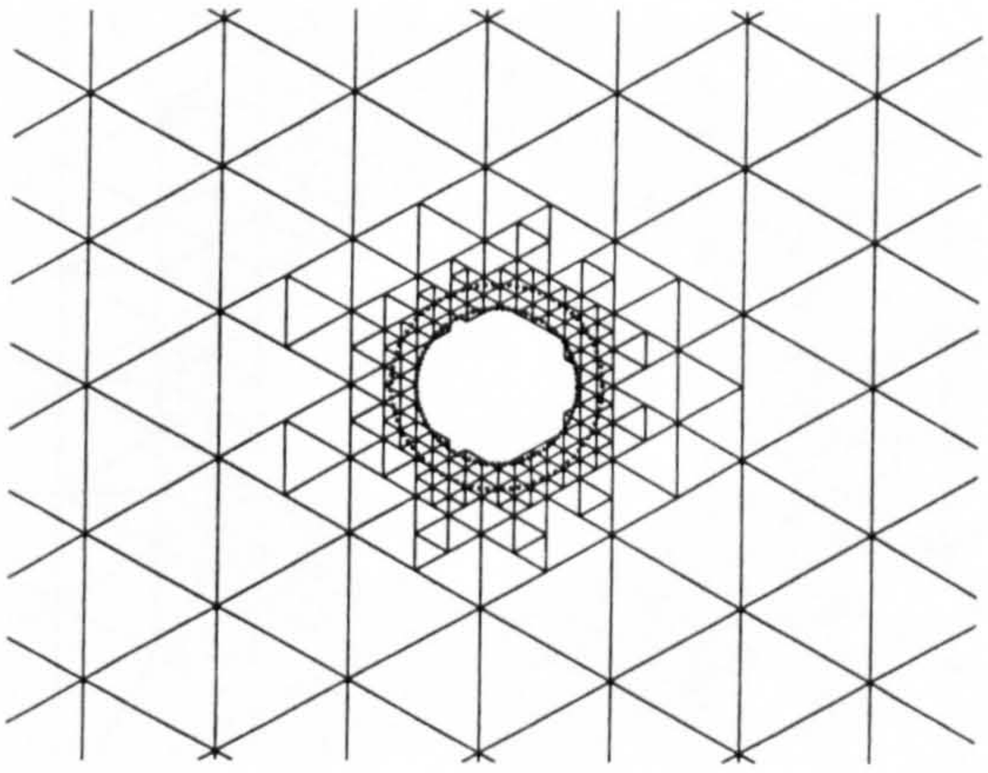
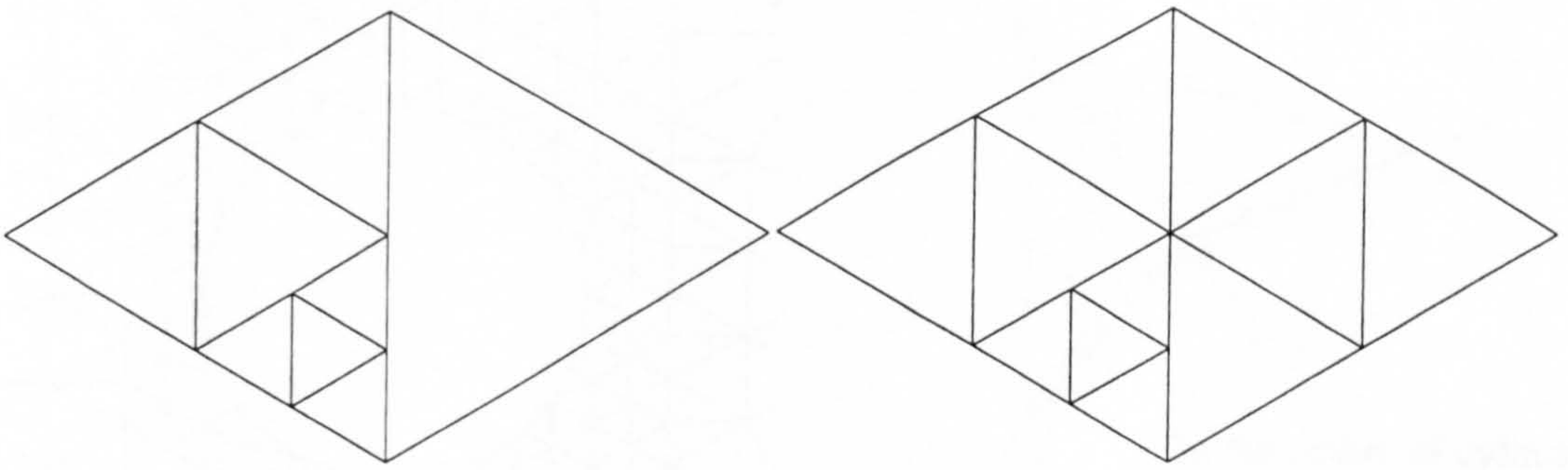


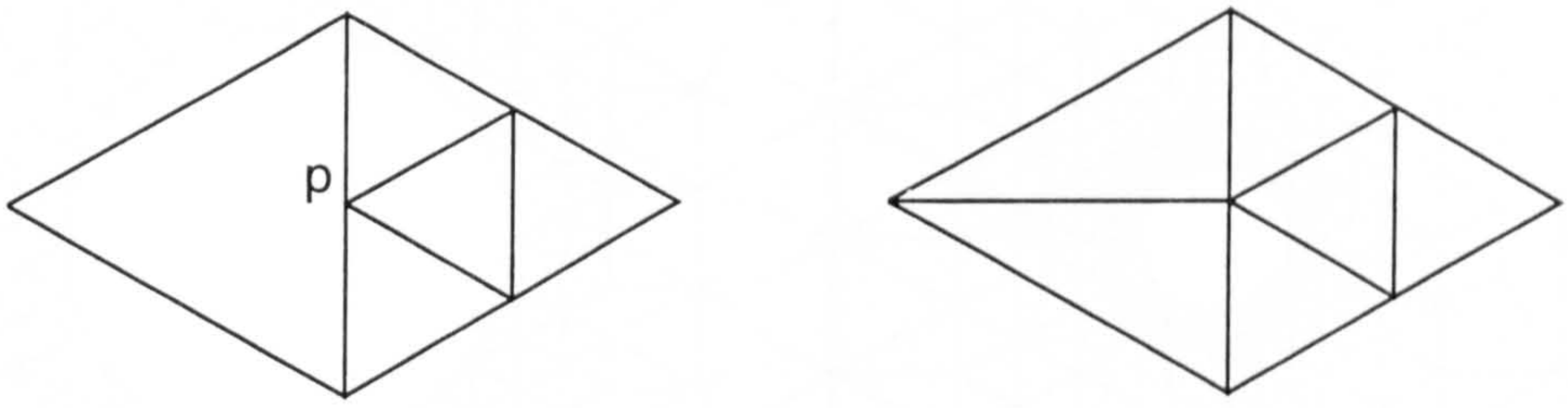
Figure 3.17 Grid after face regulation



(a) *(b)*

Figure 3.18 Face regulation

(a) Abrupt change in sharing a face (b) Face regulation



(a) (b)

Figure 3.19 Eliminating the hanging node

(a) A hanging node (b) Eliminating the hanging node

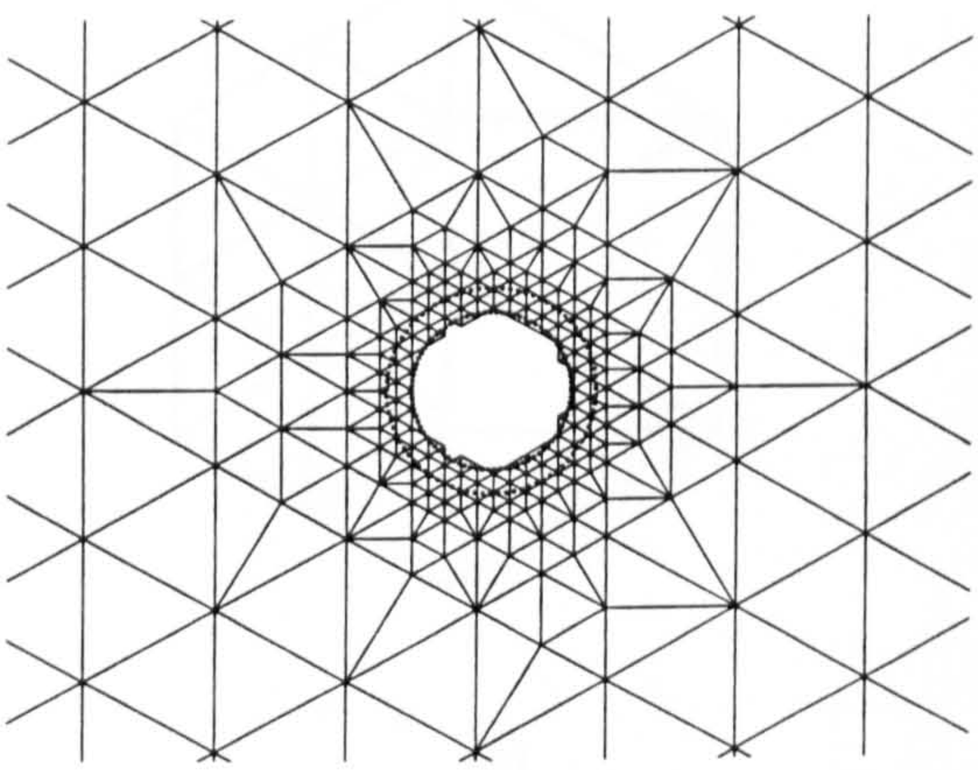


Figure 3.20 Elimination of hanging nodes

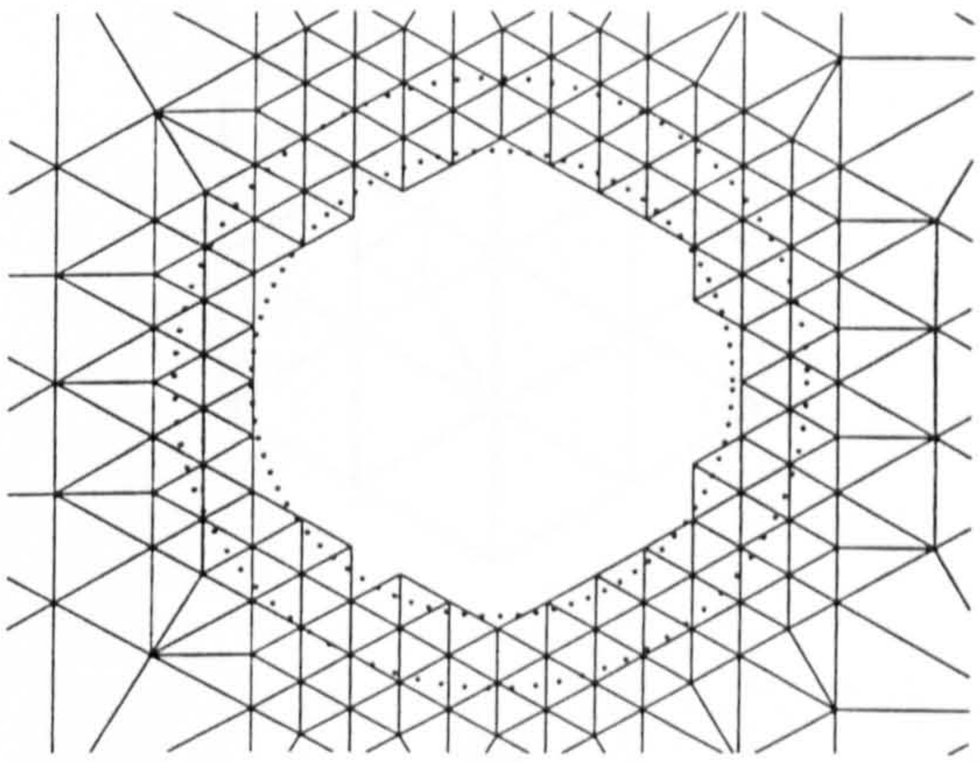


Figure 3.21 The vicinity of the cylinder

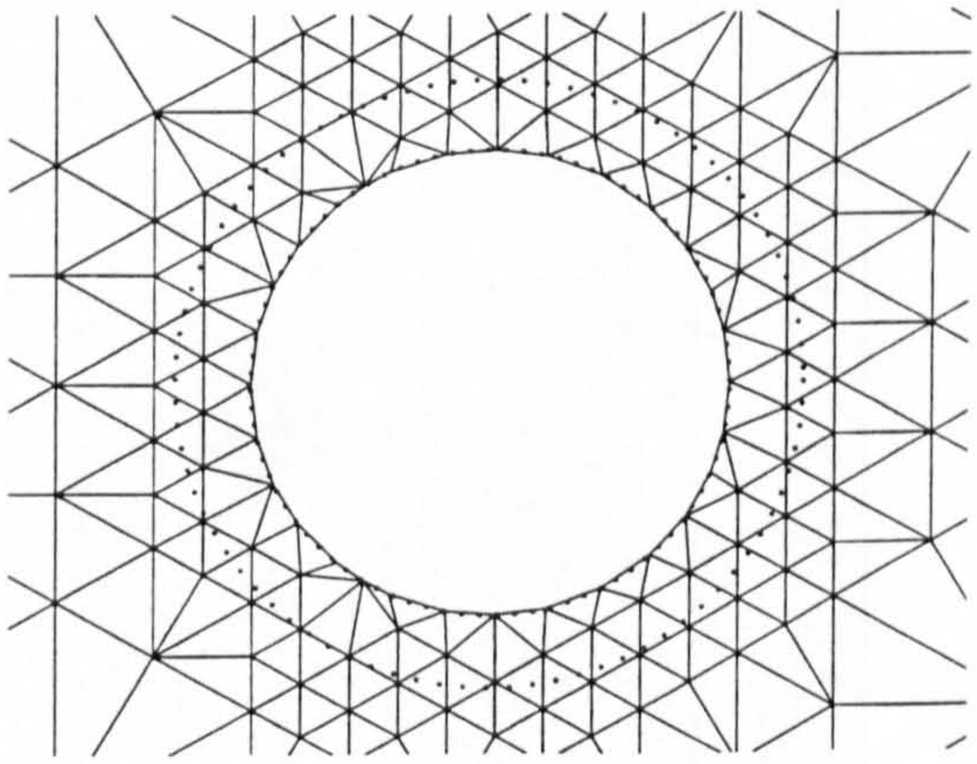


Figure 3.22 The stretching technique

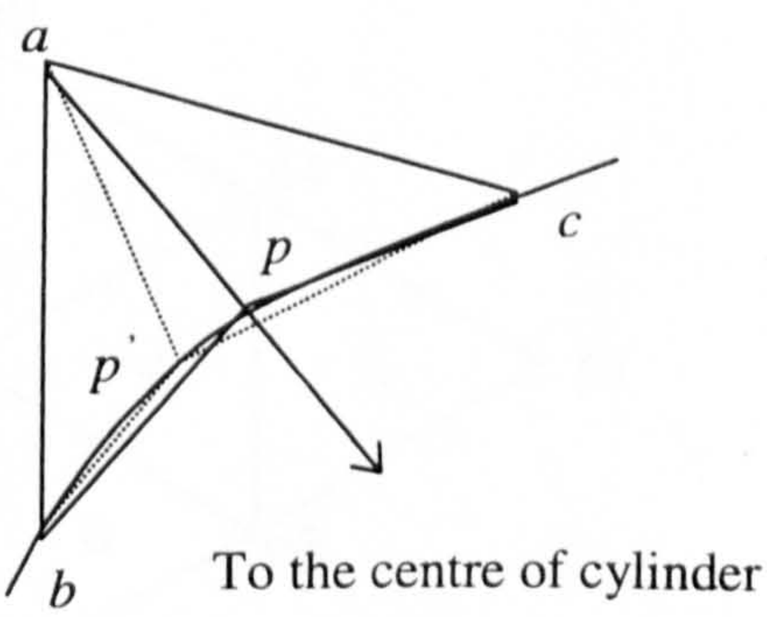


Figure 3.23 Elimination of obtuse angle

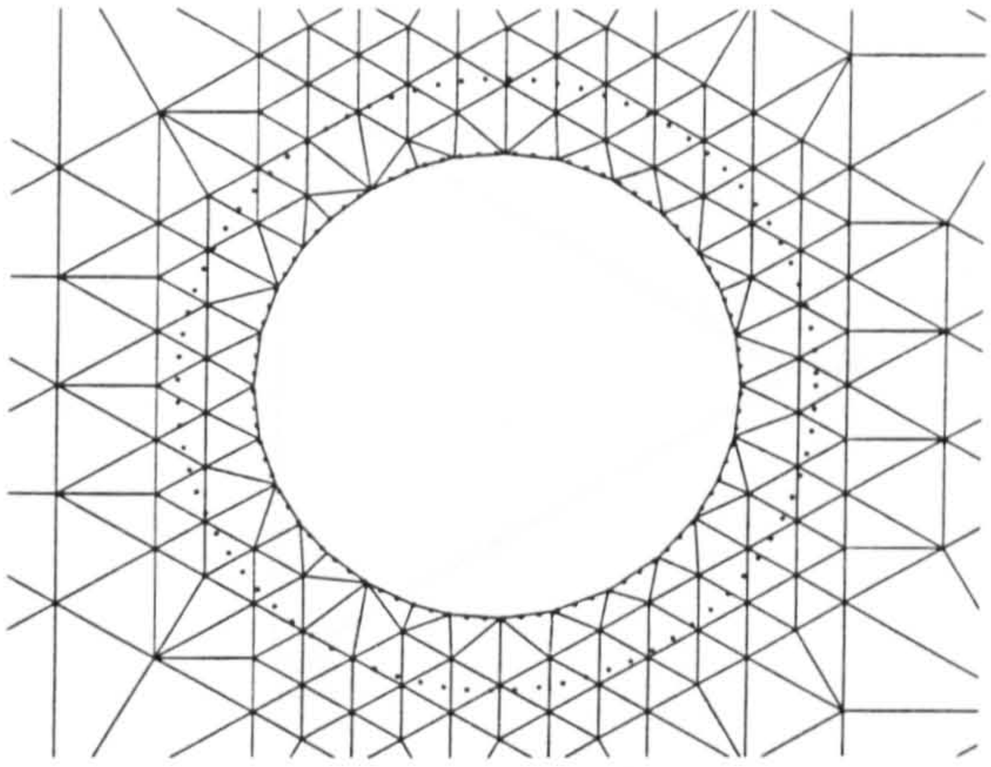


Figure 3.24 After the elimination of obtuse angles

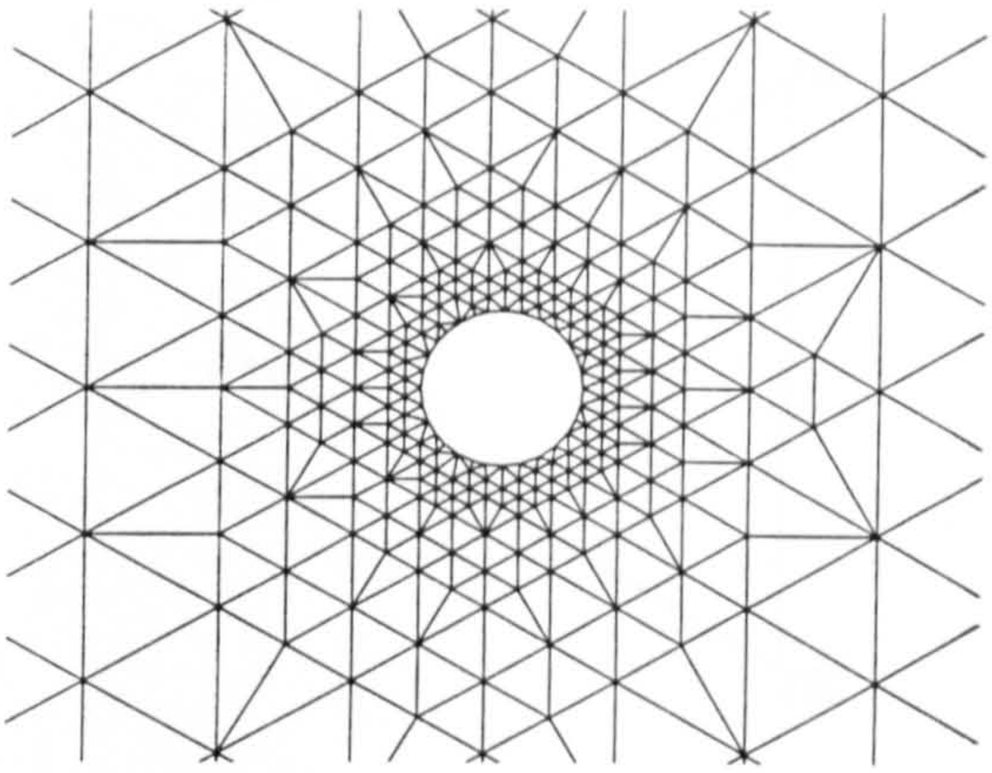
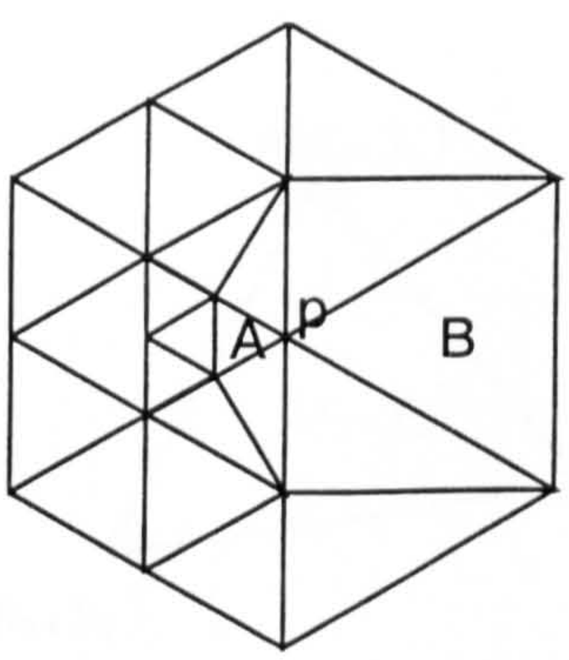
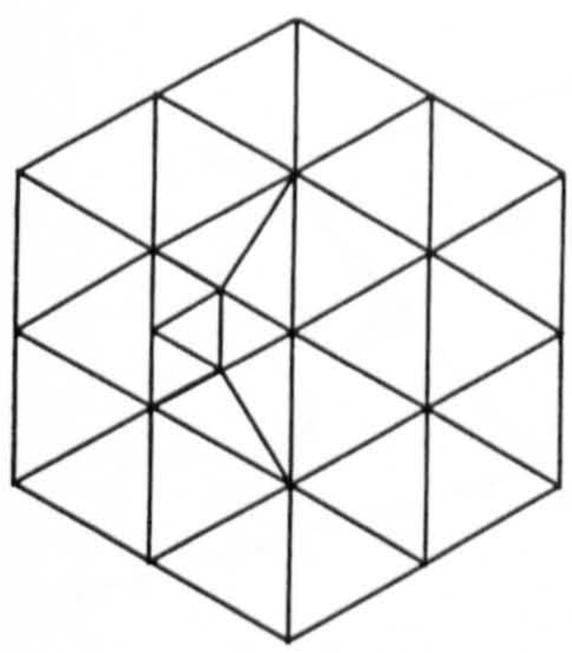


Figure3.25 Grid after corner regulation





(a) (b)

Figure 3.26 (a) Abrupt change in sharing a node (b) corner regulation

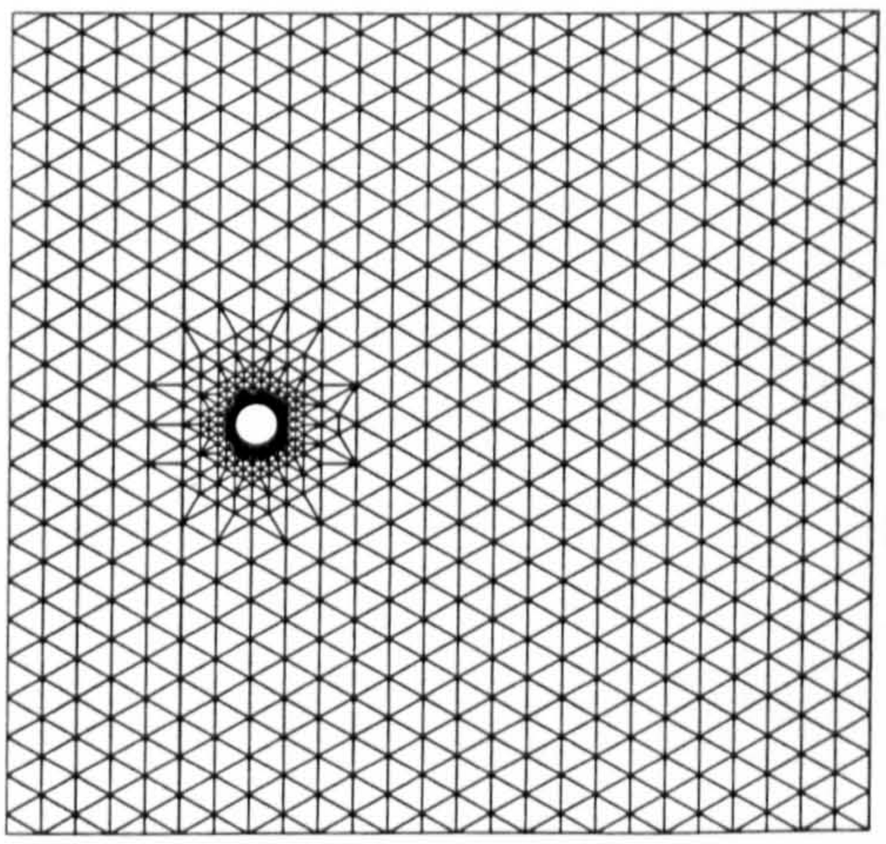


Figure 3.27 Entire grid, level 9

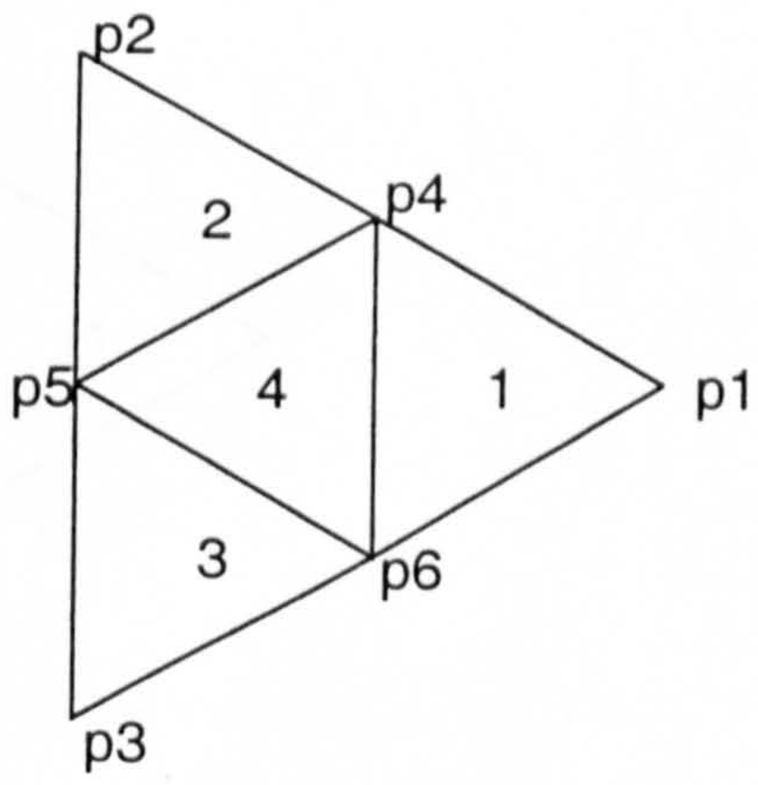
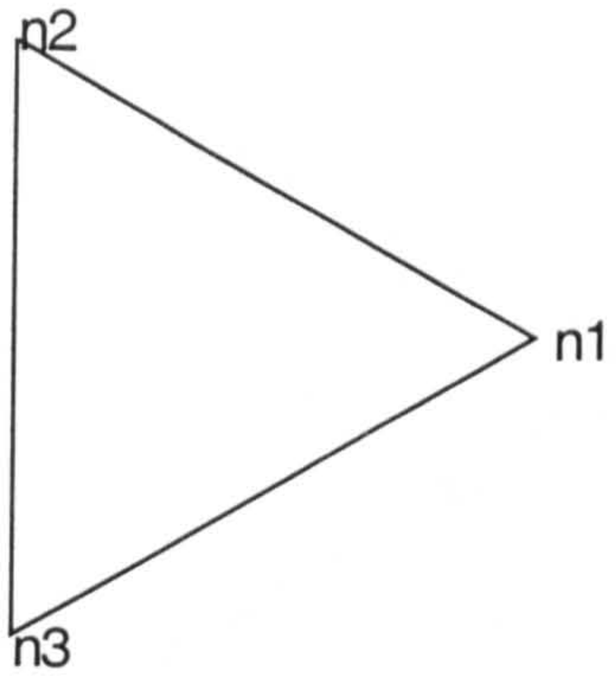
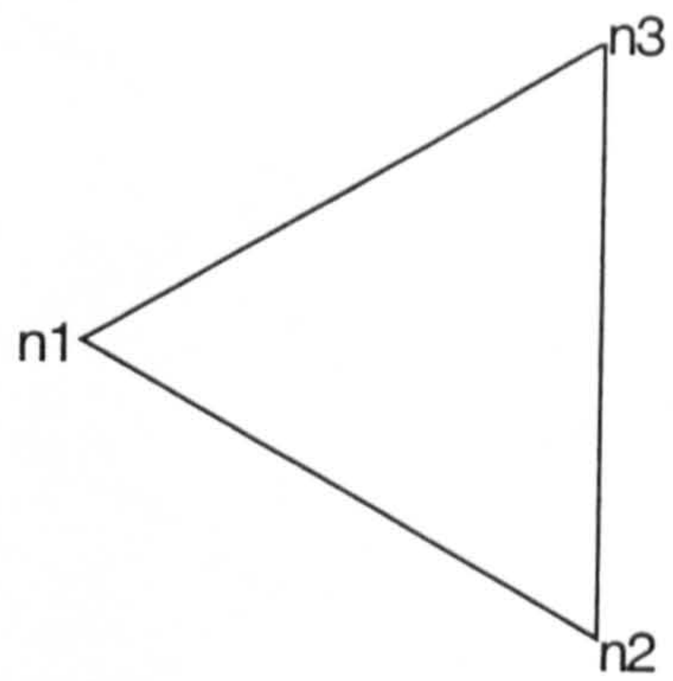


Figure 3.28 Nodal numbering



(a)



(b)

Figure 3.29 (a) The local nodal numbering for a rightward facing triangle (b) The local nodal numbering for a leftward facing triangle

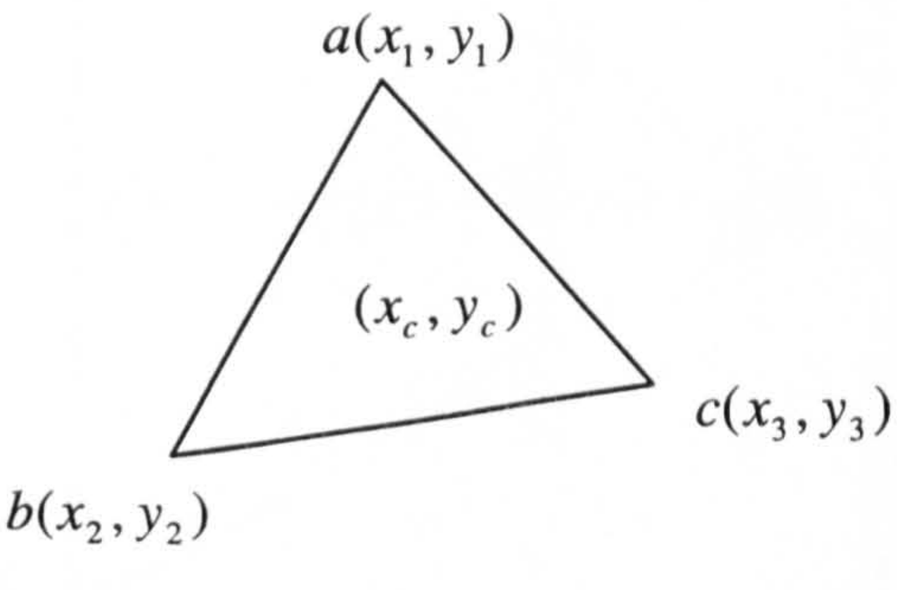


Figure 3.30 Notation associated with equations (3-4) and (3-5)

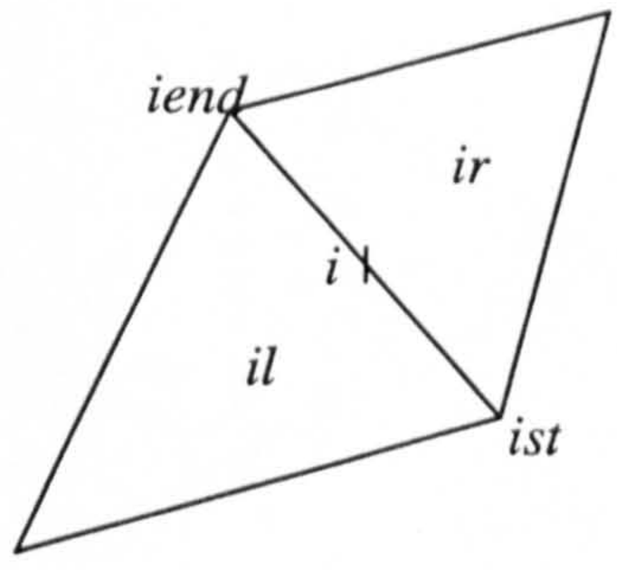


Figure 3.31 The face i indexing system

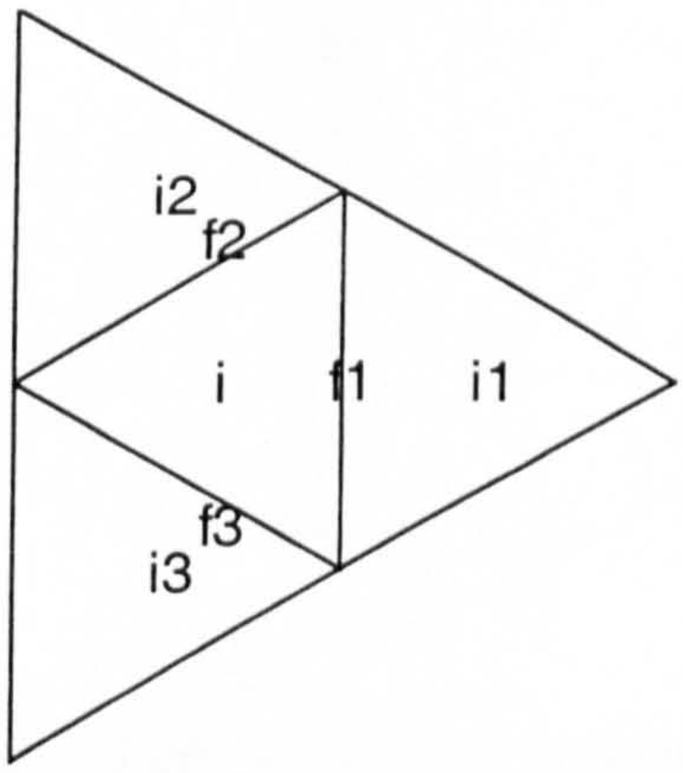


Figure 3.32 A triangular grid i with its three faces and three neighbours

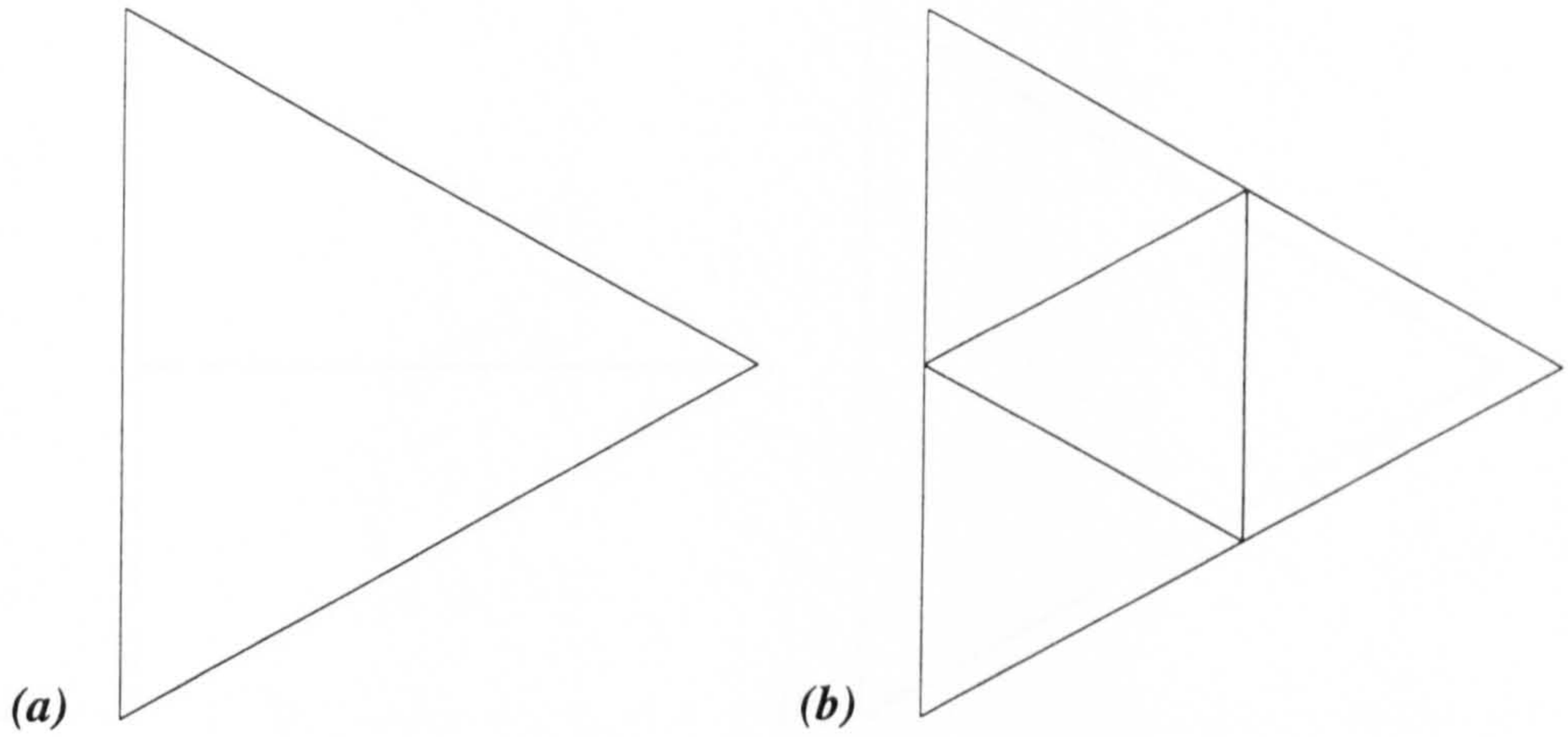


Figure 3.33 (a) Before refining element (b) After refining element

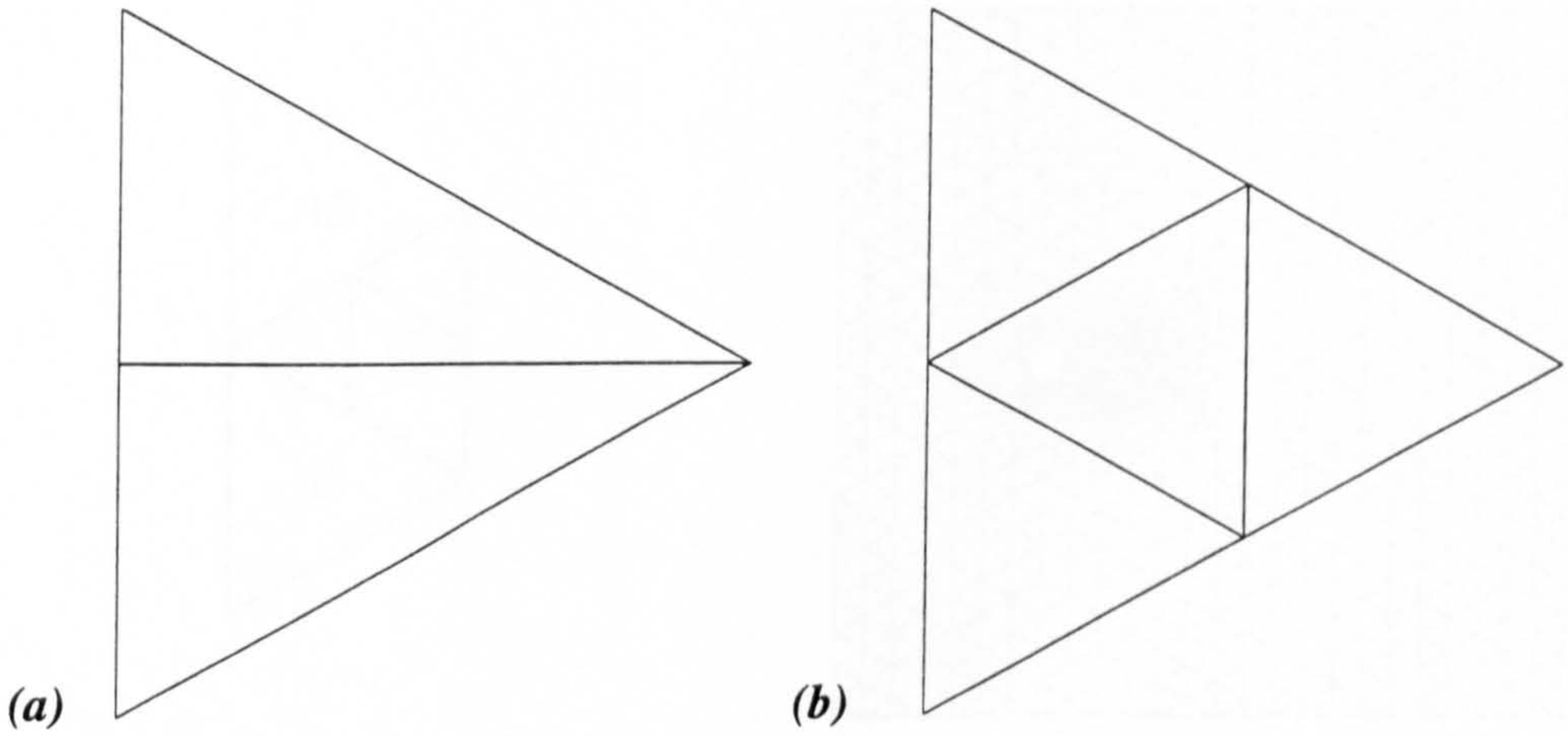


Figure 3.34 (a) Before refining element (b) After refining element

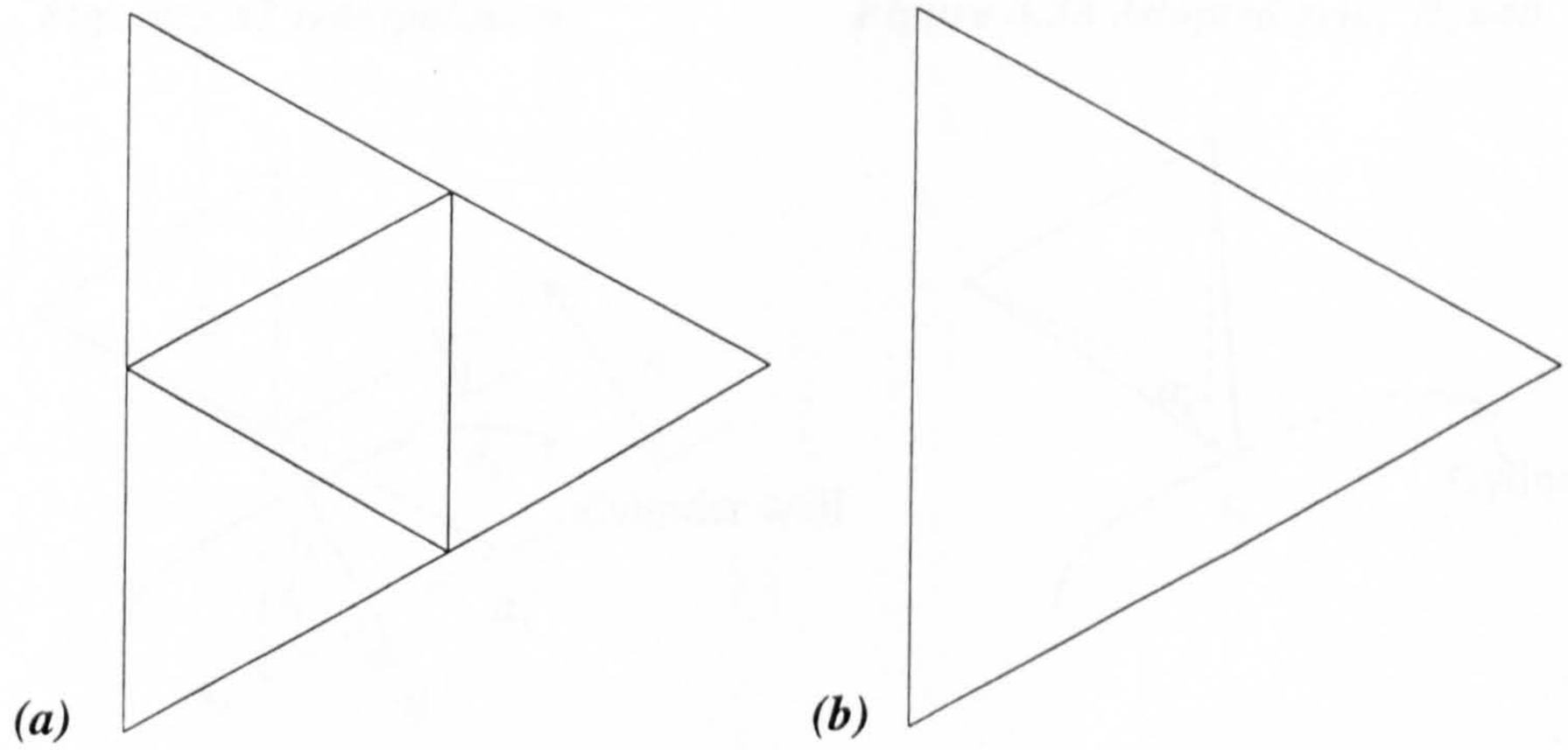
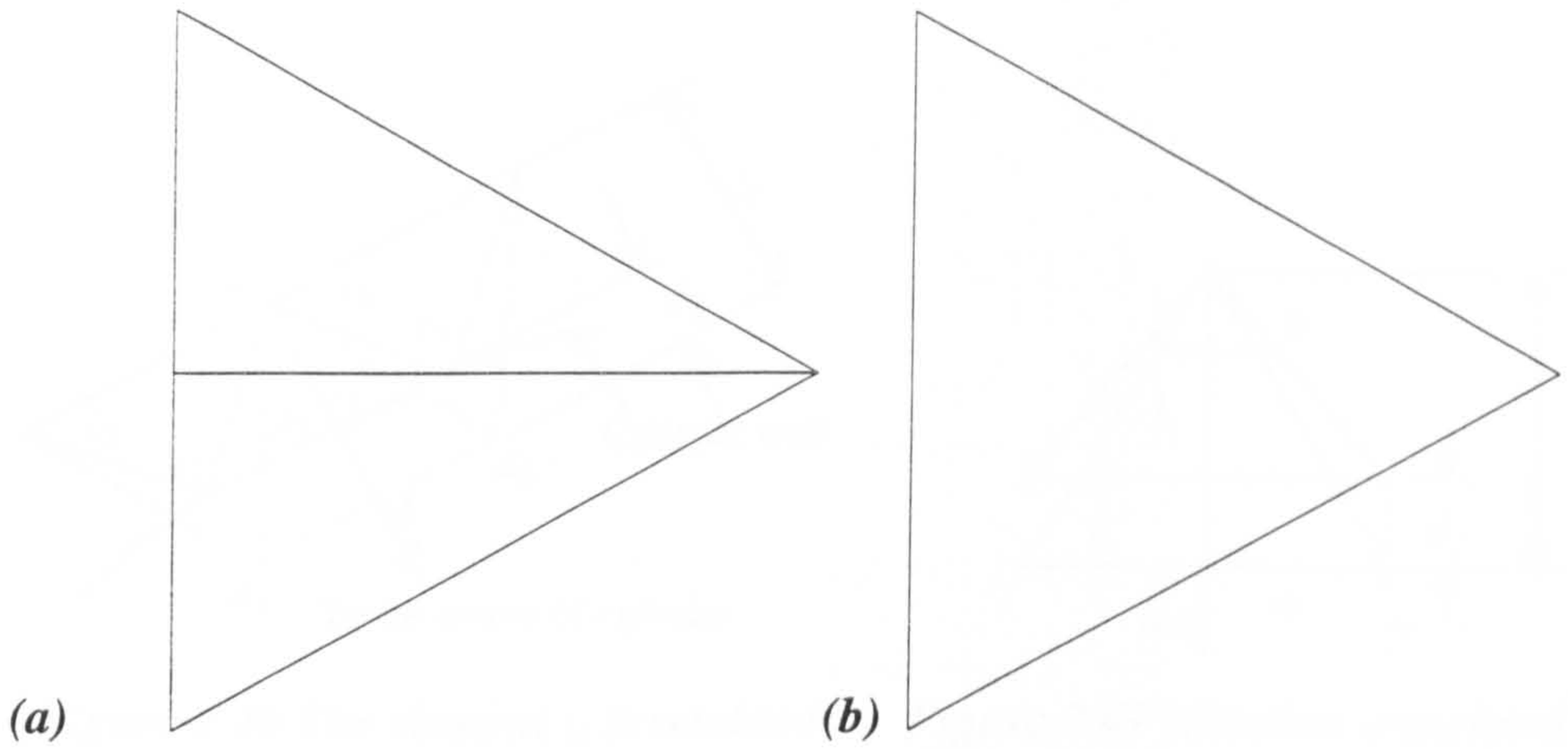


Figure 3.35 (a) Before derefining element (b) After derefining element



(a) (b)

Figure 3.36 (a) Before derefining element (b) After derefining element

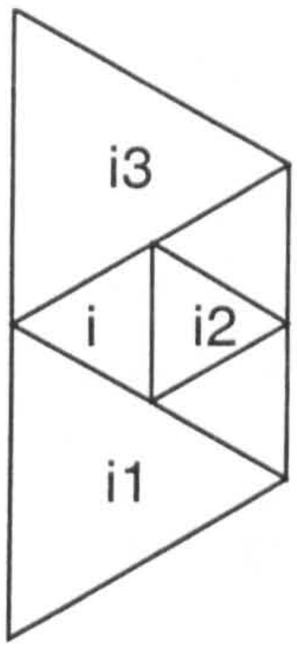


Figure 3.37 Interpolation

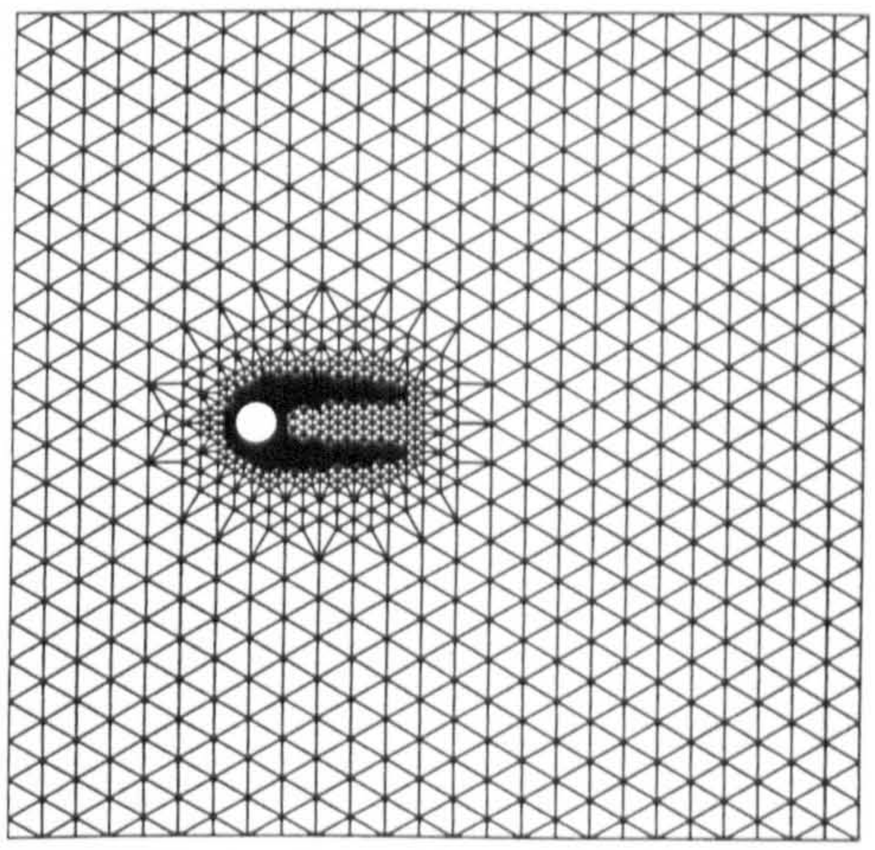
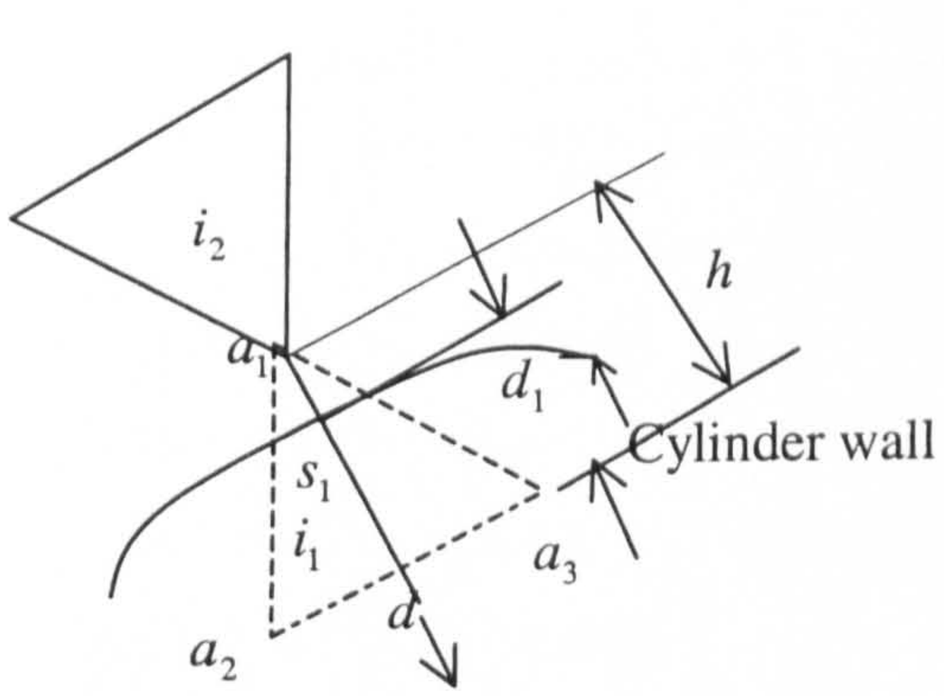
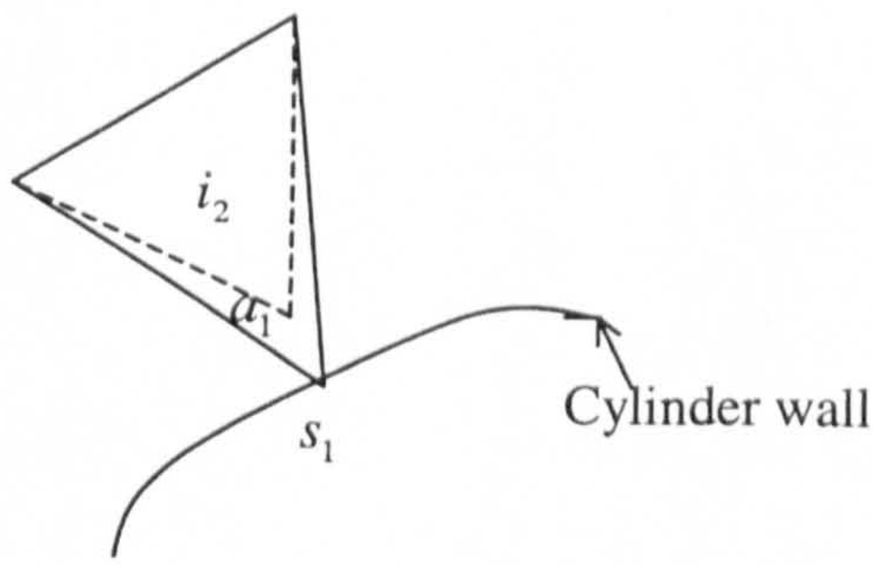


Figure 3.38 Adapted grid, R_e =40



(a)



(b)

Figure 3.39 (a)The element i_1 is deleted (b) The element i_2 is distorted

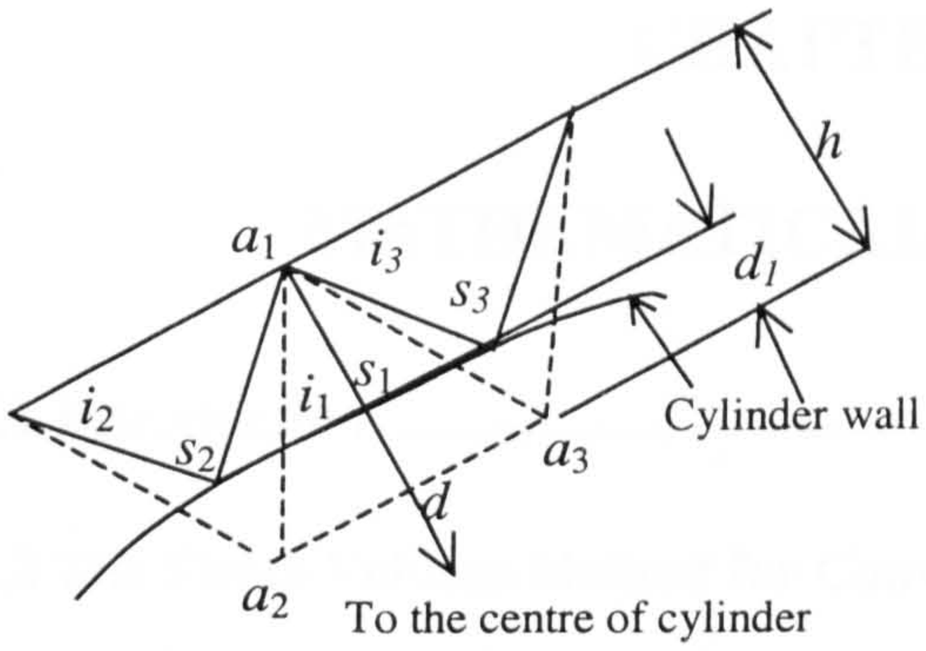


Figure 3.40 The element i_1 is retained and i_1, i_2 and i_3 are distorted

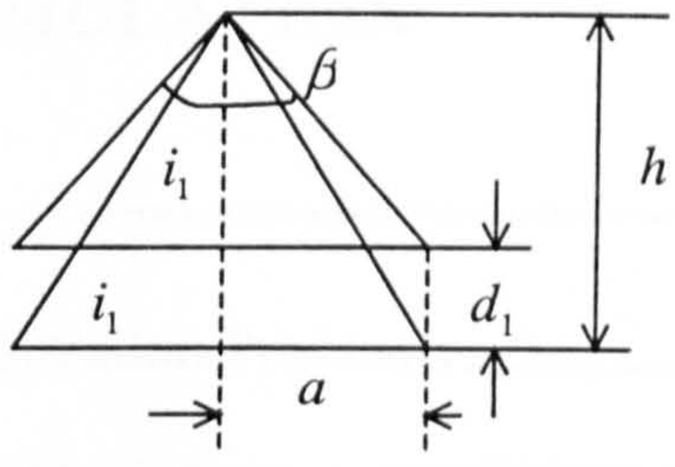


Figure 3.41 Notation associated with the optimum ratio

CHAPTER 4

MATHEMATICAL FORMULATION

4.1 Introduction

In this chapter, the derivation of a finite volume method for solving the Navier-Stokes equations on unstructured triangular grids is presented. A cell-centred scheme is used and the pressure-velocity coupling is treated using both SIMPLE and MAC algorithms. The derivation of the SIMPLE algorithm is based on that described by Davidson (1996) and the MAC algorithm is due to Harlow and Welch (1965). Both algorithms are derived from the basic principles of conservation of mass, momentum and energy of fluid flow. Those formulations lead to the transport equations of fluid flow and can be solved with appropriate boundary conditions. The methods are used here for two-dimensional viscous and incompressible flow. The derivation can also be extended to three-dimensional cases in the future.

4.2 The Finite Volume Method for Convection-Diffusion Problems

4.2.1 Transport equations

Fluid flow is governed by the Navier-Stokes momentum equations coupled with the continuity equation. In two dimensions, the equations can be expressed as follows:

$$\frac{\partial(\rho u)}{\partial t}+\rho(u\frac{\partial u}{\partial x}+v\frac{\partial u}{\partial y})=\mu_\phi(\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2})-\frac{\partial p}{\partial x}$$

$$\frac{\partial(\rho v)}{\partial t}+\rho(u\frac{\partial v}{\partial x}+v\frac{\partial v}{\partial y})=\mu_\phi(\frac{\partial^2 v}{\partial x^2}+\frac{\partial^2 v}{\partial y^2})-\frac{\partial p}{\partial y}$$

$$\frac{\partial(\rho u)}{\partial x}+\frac{\partial(\rho v)}{\partial y}=0$$

where u and v are the velocity components in the x, y dimensions and p is the pressure. ρ is the density of the fluid and μ_ϕ is the coefficient of dynamic viscosity of the fluid.

Let ϕ represent a general variable, such as velocity, pressure or concentration, which can be transported in a fluid. The general transport equation for ϕ can then be written as

$$\frac{\partial(\rho\phi)}{\partial t}+\nabla\cdot(\rho\mathbf{U}\phi)=\nabla\cdot(\mu_\phi\nabla\phi)+\bar{s}^\phi \qquad (4\text{-}1)$$

where $\bar{s}^\phi$ denotes a source per unit volume. The $\frac{\partial(\rho\phi)}{\partial t}$ term signifies the rate of change

of the total amount of fluid property ϕ in the control volume; for the steady case, $\frac{\partial(\rho\phi)}{\partial t} = 0$. If a flux vector **J** containing convection and diffusion is defined as

$$\mathbf{J} = \rho \mathbf{U}\phi - \mu_\phi \nabla\phi , \tag{4-2}$$

then in steady state vector notation the transport equation reads

$$\nabla \cdot \mathbf{J} = \bar{s}^\phi .$$

Integrating this equation over a volume V bounded by surface A and using the Gauss law of divergence gives

$$\int_A \mathbf{J} \cdot d\mathbf{A} = \int_V \bar{s}^\phi dV \tag{4-3}$$

which, for a control volume implies that

$$\sum_{j=1}^{nf} (\mathbf{J} \cdot \mathbf{A})_j = S^\phi$$

where nf is the total number of faces of a element and is three in this case since a triangular grid is used and S^ϕ is the total source in the control volume.

The left-hand side of the equation (4-3) has two parts: the net convective flux and the net diffusive flux. The right hand side represents the generation or destruction of the property ϕ within the control volume. With reference to Figure 4.1, the area denoted by broken lines $(a-b-c-d)$ is used for computing the gradients at the face ff (see Sections 4.2.3, 4.3.1.2 and 4.3.1.4). Lengths ad and cb must be parallel to the face ff and pass through the element centres i and ij, whereas ab and cd must be parallel to the straight line from i to ij and pass through points jst and $jend$.

4.2.2 Convection terms

In convection diffusion problems, convection spreads influence only in the flow direction. The convection term is the first part of the flux vector **J** in equation (4-2). The principal problem in the discretisation of the convective terms is the calculation of the value of the transported property ϕ at control volume faces and its convective flux across the boundaries. The convective flux contains the mass flux m_{fj} multiplied by the scalar property ϕ. The mass flux, $m_{fj} = \rho A_{fj} U^{fj}$ represents the mass flow out of or into the element by convection across the face. The face areas A_{fj} are shown in Figure 4.2, where j=1, 2, 3. The face normal velocities U^{fj} are calculated by:

$$U^{fj} = \{u \cdot n_x + v \cdot n_y\}_{fj} = u_{fj} \cdot n_{xfj} + v_{fj} \cdot n_{yfj} \tag{4-5}$$

where (n_{xfj}, n_{yfj}) are the components of the unit outward-pointing normal vector of the face, computed as the vector cross product between the vectors $(jst)(jend)$ and $\bar{z}$. Note that the vector $\bar{z} = \overline{(i)(ij)} \times \overline{(jst)(jend)}$. With reference to Figure 4.3, we have

$$n_{xfj} = \frac{y_{jend} - y_{jst}}{d_{jst,jend}}, \qquad n_{yfj} = -\frac{x_{jend} - x_{jst}}{d_{jst,jend}}, \tag{4-6}$$

where $$d_{jst,jend} = \sqrt{(y_{jend} - y_{jst})^2 + (x_{jend} - x_{jst})^2} \ .$$

The face velocity components shown in Figure 4.4 are calculated by:

$$u_{fj} = \lambda_j \cdot u_i + (1 - \lambda_j) \cdot u_{ij}$$

and $$v_{fj} = \lambda_j \cdot v_i + (1 - \lambda_j) \cdot v_{ij} \ , \tag{4-7}$$

where λ_j is a weight function for linear interpolation and is defined as

$$\lambda_j = \frac{d_i}{d_{i,ij}},$$

with
$$d_i = \sqrt{(x_{ij} - x_{fj})^2 + (y_{ij} - y_{fj})^2}\ , \qquad (4\text{-}8)$$
$$d_{i,ij} = \sqrt{(x_{ij} - x_i)^2 + (y_{ij} - y_i)^2}.$$

Here (x_{fj}, y_{fj}) are the face co-ordinates of the intersection between $(i)(ij)$ and $(jst)(jend)$ and can be obtained from the following equations,

$$x_{fj} = \frac{b1 \cdot c2 - b2 \cdot c1}{a1 \cdot b2 - a2 \cdot b1}$$

$$y_{fj} = \frac{a2 \cdot c1 - a1 \cdot c2}{a1 \cdot b2 - a2 \cdot b1}$$

where $a1 = y_{jst} - y_{jend}$, $a2 = y_i - y_{ij}$, $b1 = -(x_{jst} - x_{jend})$, $b2 = -(x_i - x_{ij})$, $c1 = -x_{jst} \cdot a1 - y_{jst} \cdot b1$ and $c2 = -x_i \cdot a2 - y_i \cdot b2$.

Then the mass flux value is given by

$$m_{fj} = \rho A_{fj} U^{fj} = \rho A_{fj} (u_{fj} n_{xfj} + v_{fj} n_{yfj})$$

and the convection term is

$$\rho A_{fj} U^{fj} \phi. \qquad (4\text{-}9)$$

4.2.2.1 *The problem with central differencing*

Consider the transport equation (4-1) together with appropriate boundary conditions. This can be interpreted as a two-dimensional convection-diffusion equation for the scalar property ϕ with a well-defined source term S^{ϕ}. The standard control-volume approach uses the central differencing scheme, in which the face values of ϕ are defined by equation (4-7) and the convective term is evaluated using (4-9).

The non-dimensional grid Peclet number is a measure of the relative strengths of convection and diffusion in a particular element and is defined as

$$P_c = \max\left(\frac{u\Delta x}{\nu}, \frac{v\Delta y}{\nu}\right)$$

in which Δx and Δy denote the width and height for the rectangular element and the face length for the triangular element, u and v are the element velocity components and ν is the kinematic viscosity of the fluid.

There is no way of guaranteeing convergence using central differencing for the convection terms for values of P_c greater than 2. When $P_c > 2$, 'wiggles' may appear in the solution because most of the coefficients are negative and the solution contains large under- and overshoots. Therefore, the central differencing scheme does not apply at high P_c because it is influenced by both upstream and downstream values equally, whereas in a strong convective flow upstream values should have a strong influence. The limitations of the central differencing scheme are given in more detail by Versteeg and Malalasekera (1995). Owing to this limitation, central differencing is not a suitable discretisation method for numerous realistic problems. Two alternative schemes are discussed in the following sections.

4.2.2.2 *The upwind differencing scheme*

The problem associated with both central and upwind differencing techniques can be demonstrated by the methods used in estimating the face values of the dependent variables in a control-volume formulation in each case. One of the major inadequacies of the central differencing scheme is that it disregards the flow direction. The value of property ϕ at the face i is always influenced by both ϕ_r and ϕ_l in central differencing. With reference to Figure 4.5, a strongly convective flow from west to east, the central differencing is unsuitable because the west element should receive much stronger influencing from face l than from face r. The upwind differencing scheme (often called the donor-cell technique, see Versteeg and Malalasekera (1995)) takes into account the

flow direction when determining the value at a element face: the convected value of ϕ at the face i is taken to be equal to the value at the upstream element. Figure 4.5 shows the upwind scheme when u_l and u_r are both positive relative to the downstream direction x,

$$\phi_l = \phi_L \quad \text{and} \quad \phi_r = \phi_i .$$

The scheme is based on the backward differencing formula and is only first-order accurate from the truncated Taylor series.

4.2.2.3 *Quadratic upstream interpolation for QUICK differencing*

A natural question arises as to the possibility of constructing an interpolation scheme which simultaneously possesses good accuracy and the directional properties associated with stable convective sensitivity. Here, by using a ***three-point upstream-weighted quadratic interpolation for each face value individually,*** a conservative formulation with stable convective sensitivity can be achieved. Figure 4.6 shows the basic interpolation scheme (see Versteeg and Malalasekera (1995)) when ϕ_l and ϕ_r are both positive. For constant grid spacing the resulting formula is

$$\phi_r = \frac{1}{2}(\phi_i + \phi_{i+1}) - \frac{1}{8}(\phi_{i-1} + \phi_{i+1} - 2\phi_i) \ . \qquad (4\text{-}10)$$

For variable grid spacing, the face value ϕ_r is derived as follows. (see Figure 4.7)

For example, when $\phi_i > 0$ and $\phi_{i+1} > 0$, a quadratic fit through two bracketing nodes $i-1$ and i, and upstream node $i+1$ is used to calculate ϕ_r. The quadratic equation of the form

$$\phi(x) = ax^2 + bx + c \, .$$

is used. The coefficients of a, b and c are calculated by solving the linked equations: (see Figure 4.7)

$$\phi_{i+1} = \phi(h_1) = a(h_1)^2 + b(h_1) + c$$

$$\phi_i = \phi(-h_2) = a(h_2)^2 - b(h_2) + c$$

$$\phi_{i-1} = \phi(-h_3) = a(h_3)^2 - b(h_3) + c,$$

where h_1 is the distance between the positions of face r and node i+1, h_2 is the distance between the positions of face r and node i and h_3 is the distance between the positions of face r and node i -1. The resulting formula is

$$\phi_r = \frac{h_2 h_3}{(h_1 + h_2)(h_1 + h_3)}\phi_{i+1} + \frac{h_1 h_3}{(h_1 + h_2)(h_3 - h_2)}\phi_i - \frac{h_1 h_2}{(h_1 + h_3)(h_3 - h_2)}\phi_{i-1}. \qquad (4\text{-}11)$$

4.2.2.4 *Weight-function quadratic interpolation of QUICK differencing*

In order to calculate face values on the two-dimensional triangular grid, a three-point-upstream-weighted quadratic interpolation is used. The choice of interpolation is very important as it can directly affect the accuracy of the results. Figure 4.8 shows the notation for the interpolation, assuming that the flow direction is from left to right.

The values ϕ_{i1} and ϕ_i are calculated using the most recent element centre values. The third upstream location of the point s is established by extending the line connecting points i_1 and i to the element face. The value of ϕ_s is found by interpolation of the adjacent node and face at $\phi_{j3,jst}$ and ϕ_{f3} respectively using equation (4-7).

4.2.3 Diffusion terms

The second part of the flux vector **J** in equation (4-2) represents diffusion, and has the form

$$-\mu_\phi \mathbf{A} \cdot \nabla\phi$$

Thus, the diffusion across a given face fj is

$$-\{\mu_\phi \mathbf{A}\cdot\nabla\phi\}_{ij} = -\{\mu_\phi(A_x\frac{\partial\phi}{\partial x}+A_y\frac{\partial\phi}{\partial y})\}_{ij} = -\{\mu_\phi|A|(n_x\frac{\partial\phi}{\partial x}+n_y\frac{\partial\phi}{\partial y})\}_{ij} \qquad (4\text{-}12)$$

where A_x is the face-area component in the x direction, and A_y is the face-area component in the y direction; $|A|$ is the magnitude of the length of the face and equals A_{ij} in Figure 4.2. We must now evaluate the derivatives $\frac{\partial\phi}{\partial x}$ and $\frac{\partial\phi}{\partial y}$ at the faces. This is achieved by applying Green's formula to the control volume *a-b-c-d* (see Figure 4.9) surrounding the midpoint of the face, i.e.

$$(\frac{\partial\phi}{\partial x})_{ij} = \frac{1}{V_{ij}}\int_A \phi n_x dA \qquad (\frac{\partial\phi}{\partial y})_{ij} = \frac{1}{V_{ij}}\int_A \phi n_y dA,$$

where A is the length of the surface *a-b-c-d* and V_{ij} is the enclosed volume area.

Here, the gradients in the x and y- component directions of the unit outward pointing normal for face j, are given by:

$$\begin{aligned}(\frac{\partial\phi}{\partial x})_{ij}n_{xij} &= \frac{n_{xij}}{V_{ij}}(\phi_{ij}\cdot n_{xij}A_{ij} + \phi_{jend}n_{xjend}A_{jend} + \phi_i n_{xi}A_i + \phi_{ist}n_{xist}A_{jst})\\ (\frac{\partial\phi}{\partial y})_{ij}n_{yij} &= \frac{n_{yij}}{V_{ij}}(\phi_{ij}n_{yij}A_{ij} + \phi_{jend}n_{yjend}A_{jend} + \phi_i n_{yi}A_i + \phi_{jst}n_{yjst}A_{jst})\end{aligned} \qquad (4\text{-}13)$$

These expressions are summed for each face of a given element. For each element, the face vector and areas are related, i.e. $A_i = A_{ij} = A_{ij}$ and $A_{jst} = A_{jend}$ (see Figure 4.9). The normal vectors $\overline{n}_i$ and $\overline{n}_{ij}$ are parallel to the normal vector of the face $\overline{n}_{ij}$ and the normal vectors $\overline{n}_{jst}$ and $\overline{n}_{jend}$ are normal to the vector $\overline{(i)(ij)}$. Here, (n_{xij}, n_{yij}) is equal to the unit normal vector of the face (n_{xij}, n_{yij}), and is calculated using equation (4-6). The vector (n_{xi}, n_{yi}) has the same magnitude as (n_{xij}, n_{yij}), but lies in the opposite direction, whilst (n_{xjend}, n_{yjend}) is the unit normal vector of the face *cd* centred at the node *jend*. They are taken (see Figure 4.9) as

$$n_{xjend} = \frac{y_i - y_{ij}}{d_{i,ij}}, \qquad n_{yjend} = -\frac{x_i - x_{ij}}{d_{i,ij}}, \tag{4-14}$$

where $$d_{i,ij} = \sqrt{(y_{ij} - y_i)^2 + (x_{ij} - x_i)^2} \ .$$

The vector (n_{xjst}, n_{yjst}) is the unit normal vector of *ab* centred at the node *jst*. It has the same magnitude as (n_{xjend}, n_{yjend}), but lies in the opposite direction.

So, equation (4-13) can be rewritten as

$$\begin{aligned}(\frac{\partial \phi}{\partial x})_{fj} n_{xfj} &= \frac{n_{xfj}}{V_{fj}}(n_{xfj} A_{fj} \phi_{ij} + n_{xjend} A_{jend} \phi_{jend} - n_{xfj} A_{fj} \phi_i - n_{xjend} A_{jend} \phi_{jst}) \\ (\frac{\partial \phi}{\partial y})_{fj} n_{yfj} &= \frac{n_{yfj}}{V_{fj}}(n_{yfj} A_{fj} \phi_{ij} + n_{yjend} A_{jend} \phi_{jend} - n_{yfj} A_{fj} \phi_i - n_{yjend} A_{jend} \phi_{jst})\end{aligned}, \tag{4-15}$$

in which the face areas A_{fj} and A_{jend} are given by:

$$A_{fj} = d_{jst,jend} = \sqrt{(y_{jend} - y_{jst})^2 + (x_{jend} - x_{jst})^2}$$

$$A_{jend} = d_{i,ij} = \sqrt{(x_{ij} - x_i)^2 + (y_{ij} - y_i)^2}$$

and the volume *a*-*b*-*c*-*d* by

$$V_{fj} = \left| d_{jst,jend} \times d_{i,ij} \right| = (y_{jend} - y_{jst})(x_{ij} - x_i) - (x_{jend} - x_{jst})(y_{ij} - y_i).$$

The sum of the two terms in equation (4-15) can be written as

$$(\frac{\partial \phi}{\partial x})_{fj} n_{xfj} + (\frac{\partial \phi}{\partial y})_{fj} n_{yfj} =$$

$$\frac{1}{V_{fj}}[A_{fj}(\phi_{ij} - \phi_i) + A_{jend}(n_{xfj} \cdot n_{xjend} + n_{yfj} \cdot n_{yjend})(\phi_{jend} - \phi_{jst})], \tag{4-16}$$

where $(n_{xfj}^2 + n_{yfj}^2) = 1$ is used. The first term on the right-hand side represents orthogonal diffusion while the second term represents non-orthogonal diffusion, which vanishes on orthogonal grids, i.e. when $\overline{(i)(ij)} \cdot \overline{(jst)(jend)} = 0$. It was noted by Davidson (1996) that "The non-orthogonal terms are mostly negligible on smooth meshes based on quadrilaterals and often also on smooth meshes based on triangles." However, for stretched triangles the non-orthogonal terms are not negligible. The diffusion flux can now be written as

$$-\{\mu_\phi A \cdot \nabla\phi\}_{fj} = -\frac{\mu_\phi A_{fj}}{V_{fj}}[A_{fj}(\phi_{ij} - \phi_i) + A_{jend}(n_{xfj}n_{xjend} + n_{yfj}n_{yjend})(\phi_{jend} - \phi_{jst})], \quad (4\text{-}17)$$

where the first term on the right-hand side will be treated implicitly and the second term on the right-hand side will be treated explicitly using values at the previous iteration. Thus, the non-orthogonal terms will be placed on the right-hand side of the discretised equation, within the source term. For the sake of conciseness we rewrite the orthogonal part of the diffusion as

$$-\{\mu_\phi A \cdot \nabla\phi\}_{fj.ort} = -D_{fj}\mu_\phi(\phi_{ij} - \phi_i) = -K_{fj}(\phi_{ij} - \phi_i), \quad (4\text{-}18)$$

where D_{fj} is the diffusion coefficient

$$D_{fj} = \frac{A_{fj}^2}{V_{fj}}$$

and

$$K_{fj} = D_{fj}\mu_\phi.$$

4.2.4 The discretised equations

The diffusion process affects the distribution of a transported quantity along its gradients in all directions, whereas the influence of convection occurs only in the flow direction. This crucial difference manifests itself in an upper limit to the grid size, that is dependent on the relative strength of convection and diffusion combined in the Peclet

number, for stable convection-diffusion calculations with various differencing schemes. This has been discussed in Section 4.2.2.1.

Combining the expressions for convection and diffusion flux [i.e.equations (4-9), and (4-18)] for a triangular element with three neighbours gives

$$\sum_{j=1}[m_{fj}\,\phi_{fj} - (\mu_\phi A\,\nabla\phi)_{fj}] = S^\psi\,, \tag{4-19}$$

where S^ψ is the total source in the control volume which consists of the total pressure gradient source term S^ϕ and the non-orthogonal diffusion source term S^ϕ_{non}. Expanding equation (4-19), we have

$$\{m_{f1}\,\phi_{f1} - K_{f1}(\phi_{i1} - \phi_i)\} + \{m_{f2}\,\phi_{f2} - K_{f2}(\phi_{i2} - \phi_i)\} +$$

$$+ \{m_{f3}\,\phi_{f3} - K_{f3}(\phi_{i3} - \phi_i)\} = S^\psi\,. \tag{4-20}$$

The relevant notation is shown in Figure 4.10.

The element face values of ϕ_{fj} are evaluated using the weighting function for linear interpolation defined in equation (4-7) (central differencing). Substituting expressions for ϕ_{fj} into equation (4-19) yields

$$\sum_{j=1}^{3}\,[\lambda_j\,m_{fj}\,\phi_i + (1-\lambda_j)\,m_{fj}\,\phi_{ij} - K_{fj}(\phi_{ij} - \phi_i)] = S^\psi\,,$$

where $K_{fj} = D_{fj}\mu_\phi$ as above. Then, expanding the series, we have

$$\lambda_1\,m_{f1}\,\phi_i + (1-\lambda_1)\,m_{f1}\,\phi_{i1} - K_{f1}\phi_{i1} + K_{f1}\phi_i +$$

$$\lambda_2\,m_{f2}\,\phi_i + (1-\lambda_2)\,m_{f2}\,\phi_{i2} - K_{f2}\phi_{i2} + K_{f2}\phi_i +$$

$$\lambda_3\,m_{f3}\,\phi_i + (1-\lambda_3)\,m_{f3}\,\phi_{i3} - K_{f3}\phi_{i3} + K_{f3}\phi_i = S^\psi\,,$$

which, upon rearranging , gives

$$\phi_i[(\lambda_1 m_{f1}+K_{f1})+(\lambda_2 m_{f2}+K_{f2})+(\lambda_3 m_{f3}+K_{f3})]=$$

$$\phi_{i1}[-(1-\lambda_1)m_{f1}+K_{f1}]+\phi_{i2}[-(1-\lambda_2)m_{f2}+K_{f2}]+\phi_{i3}[-(1-\lambda_3)m_{f3}+K_{f3}]+S^{\phi}.$$

Hence, the resulting discretised equation (central differencing) is

$$a_p\,\phi_i = a_{f1}\phi_{il} + a_{f2}\phi_{i2} + a_{f3}\phi_{i3} + S^{\phi}$$

$$a_{f1} = -(1-\lambda_1)m_{f1}+K_{f1}$$

$$a_{f2} = -(1-\lambda_2)m_{f2}+K_{f2} \qquad \text{(4-21a)}$$

$$a_{f3} = -(1-\lambda_3)m_{f3}+K_{f3}$$

$$a_p = (\lambda_1 m_{f1}+K_{f1})+(\lambda_2 m_{f2}+K_{f2})+(\lambda_3 m_{f3}+K_{f3}).$$

Alternatively, if the element face values of ϕ_{fj} are evaluated using the QUICK scheme defined in equation (4-11), substituting expressions for ϕ_{fj} into equation (4-19) yields

$$\sum_{j=1}^{3}\,[(A_j m_{fj}\,\phi_{ij} + B_j m_{fj}\,\phi_i - C_j m_{fj}\phi_p) - K_{fj}(\phi_{ij} - \phi_i)] = S^{\phi},$$

where

$$A_j = [\frac{h_2 h_3}{(h_1+h_2)(h_1+h_3)}]_j, B_j = [\frac{h_1 h_3}{(h_1+h_2)(h_3-h_2)}]_j \text{ and } C_j = [\frac{h_1 h_2}{(h_1+h_3)(h_3-h_2)}]_j.$$

where h_i are defined in equation (4-11).

After expanding the series and rearranging, the resulting discretised equation (QUICK differencing) is

$$a_p\,\phi_i = a_{f1}\phi_{il} + a_{f2}\phi_{i2} + a_{f3}\phi_{i3} + S^{\phi} + \sum_{j=1}^{3} C_j m_{fj}\phi_p$$

$$a_{f1} = -A_1 m_{f1}+K_{f1}$$

$$a_{f2} = -A_2 m_{f2} + K_{f2} \tag{4-21b}$$

$$a_{f3} = -A_3 m_{f3} + K_{f3}$$

$$a_p = (B_1 m_{f1} + K_{f1}) + (B_2 m_{f2} + K_{f2}) + (B_3 m_{f3} + K_{f3}).$$

4.2.5 The source terms

The source terms in the momentum equation consist of the total pressure gradient and the non-orthogonal diffusion term which is defined in Section 4.2.3. The total pressure gradient term S^{ϕ} is the right-hand side of equation (4-3), calculated from:

$$\int_V \bar{s}^{\phi} dV = -\int_V (\frac{\partial p}{\partial x})_i dV = -(\frac{\partial p}{\partial x})_i V_i$$

$$\int_V \bar{s}^{\phi} dV = -\int_V (\frac{\partial p}{\partial y})_i dV = -(\frac{\partial p}{\partial y})_i V_i,$$

where V_i is the triangular element volume, $(\frac{\partial p}{\partial x})_i, (\frac{\partial p}{\partial y})_i$ denote the pressure gradient at the centre of element i in the x and y directions respectively.

4.2.5.1 *The Pressure gradient*

The pressure gradient at a element centre is calculated by applying Green's formula, i.e.

$$(\frac{\partial p}{\partial x})_i = \frac{1}{V_i} \int_A p_f n_x dA$$

$$(\frac{\partial p}{\partial y})_i = \frac{1}{V_i} \int_A p_f n_y dA.$$

For a triangular element with reference to Figure 4.11, we have

$$(\frac{\partial p}{\partial x})_i = \frac{1}{V_i}(p_{f1}n_{xf1}A_{f1} + p_{f2}n_{xf2}A_{f2} + p_{f3}n_{xf3}A_{f3})$$

$$(\frac{\partial p}{\partial y})_i = \frac{1}{V_i}(p_{f1}n_{yf1}A_{f1} + p_{f2}n_{yf2}A_{f2} + p_{f3}n_{yf3}A_{f3}), \qquad (4\text{-}22)$$

where p_{ff} is the face pressure which is interpolated using linear interpolation, i.e.

$$p_{ff} = \lambda_j p_i + (1-\lambda_j)p_{ij}, \qquad (4\text{-}23)$$

where λ_j is defined in equation (4-8) and p_i, p_{ij} are pressures at element centres.

4.2.5.2 *Non-orthogonal terms*

The non-orthogonal diffusion term S^{ϕ}_{non} is defined in Section 4.2.3 above. It can be written as

$$S^{\phi}_{non} = \frac{\mu_{\phi}A_{ff}}{V_{ff}}A_{jend}(n_{xff}n_{xjend} + n_{yff}n_{yjend})(\phi_{jend} - \phi_{jst})$$

where ϕ_{jend} and ϕ_{jst} denote the variables at nodes *jend* and *jst* respectively. The values of these two parameters are found from

$$\phi = ax + by + c \qquad (4\text{-}24)$$

at (x_{jst}, y_{jst}) and (x_{jend}, y_{jend}) respectively. The coefficients a, b, and c are calculated by solving the following equations: (see Figure 4.12)

$$\phi_{f1} = ax_{f1} + by_{f1} + c$$

$$\phi_{f2} = ax_{f2} + by_{f2} + c$$

$$\phi_{f3} = ax_{f3} + by_{f3} + c.$$

4.2.5.3 *Method for calculating the variables at vertices*

In consideration of the cell-centred scheme, interpolation is necessary to calculate the values at vertices from the element centre values of the triangular element.

(a) The averaged linear-interpolation scheme

In the non-orthogonal term, the vertex variables (e.g. u,v,p) need to be calculated. Equation (4-24) is a weighted linear-interpolation function for calculating the vertex variable ϕ. However, each vertex is inevitably connected to more than one adjacent face, each of which will possibly generate a different value for the variable (see Figure 4.13). This can lead to large errors when using an iterative process.

To overcome this problem, an average of all the predicted values is used. Hence, according to this formulation, we write

$$\phi_{jend} = \frac{1}{n}\sum_{j=1}^{n} \phi_{ij,jend} \,, \tag{4-25}$$

in which n is the number of connected adjacent elements ($n = 6$ for the example shown in Figure 4.13).

(b) The Shepard-type scattered data interpolation

An alternative Shepard-type scattered data interpolation has also been described in Chapter 3 (Section 3.5.3). The formulations are introduced at (3-9) and (3-10).

The averaged linear-interpolation is used for the channel flow case but the Shepard-type scattered data interpolation is necessary for other cases.

4.2.5.4 *The discretised equation with source terms included*

Combining equations (4-3) and (4-21a), and replacing ϕ with u and v, the discretised momentum equations for velocity u and v at location i are given by

$$a_p u_i = \sum_{j=1}^{3} a_{fj} u_{ij} - (\frac{\partial p}{\partial x})_i V_i + S_{non}^{\phi} \tag{4-26}$$

$$a_p v_i = \sum_{j=1}^{3} a_{fj} v_{ij} - (\frac{\partial p}{\partial y})_i V_i + S_{non}^{\phi} \tag{4-27}$$

where a_p is the discretised diagonal coefficient, a_{fj} represents influence coefficients from the neighbour, and u_{ij}, v_{ij} are the neighbour velocities. The terms of $(\frac{\partial p}{\partial x})_i V_i$ and $(\frac{\partial p}{\partial y})_i V_i$ are the total pressure gradient source terms given on the right-hand side of equation (4-3), where S_{non}^{ϕ} is the non-orthogonal diffusion term.

The velocities may also be treated with under-relaxation factor α to stabilise the iteration process where α is chosen between 10^{-3} and 1. Equations (4-26) and (4-27) can be written in the form

$$\frac{a_p}{\alpha} u_i^{(n)} = \sum_{j=1}^{3} a_{fj} u_{ij}^{(n)} - (\frac{\partial p}{\partial x})_i^{(n-1)} V_i + S_{non}^{\phi\ (n-1)} + \frac{1-\alpha}{\alpha} a_p u_i^{(n-1)} \tag{4-28}$$

$$\frac{a_p}{\alpha} v_i^{(n)} = \sum_{j=1}^{3} a_{fj} v_{ij}^{(n)} - (\frac{\partial p}{\partial y})_i^{(n-1)} V_i + S_{non}^{\phi\ (n-1)} + \frac{1-\alpha}{\alpha} a_p v_i^{(n-1)}. \tag{4-29}$$

From a given pressure field p, the discretised momentum equations given by (4-28) and (4-29) can be used for each element and solved to obtain the velocity field. If the pressure field were correct, the resulting velocity field would satisfy the continuity equation. However, as the pressure field is unknown, we need a method for calculating pressure.

4.3 Solution Algorithms for Coupled Pressure-Velocity Systems

4.3.1 The SIMPLE algorithm

4.3.1.1 *The correction for velocity*

Patankar and Spalding (1970) suggested using the SIMPLE procedure, which first involves solving the pressure equation using pseudo velocities to obtain the pressure field. The pressure correction equation (defined below) is then solved to update the velocities. The pressure updating is determined by adding the pressure correction to the guessed pressure field. To initiate the process in this case, a pressure field p^* is guessed. The discretised momentum equations (4-26) and (4-27) are then solved using the guessed pressure field to yield velocity components u^* and v^* as follows:

$$a_p u_i^* = \sum_{j=1}^{3} a_{fj} u_{ij}^* - (\frac{\partial p^*}{\partial x})_i V_i + S_{non}^{\phi *} \tag{4-30}$$

$$a_p v_i^* = \sum_{j=1}^{3} a_{fj} v_{ij}^* - (\frac{\partial p^*}{\partial y})_i V_i + S_{non}^{\phi *}. \tag{4-31}$$

Now, we define the correction p' as the difference between the correct pressure field p and the guessed pressure field p^*, so that

$$p = p^* + p'. \tag{4-32}$$

Similarly we define velocity correction u' and v' to relate the correct velocities u and v to the guessed velocities u^* and v^* from equations (4-30) and (4-31):

$$u = u^* + u' \tag{4-33}$$

$$v = v^* + v'. \tag{4-34}$$

Subtraction of equations (4-30) and (4-31) from (4-26) and (4-27), respectively, as well as assumption that the source term $S_{non}^{\phi*}$ is unchanged, gives

$$a_p(u_i - u_i^*) = \sum_{j=1}^{3} a_{fj}(u_{ij} - u_{ij}^*) - (\frac{\partial(p - p^*)}{\partial x})_i V_i \tag{4-35}$$

$$a_p(v_i - v_i^*) = \sum_{j=1}^{3} a_{fj}(v_{ij} - v_{ij}^*) - (\frac{\partial(p - p^*)}{\partial y})_i V_i. \tag{4-36}$$

Using the correction equations ((4-32)-(4-34)), equations (4-35) and (4-36) may be rewritten as follows:

$$a_p u_i' = \sum_{j=1}^{3} a_{fj} u_{ij}' - (\frac{\partial p'}{\partial x})_i V_i \tag{4-37}$$

$$a_p v_i' = \sum_{j=1}^{3} a_{fj} v_{ij}' - (\frac{\partial p'}{\partial y})_i V_i. \tag{4-38}$$

An approximation is introduced herein: $\sum_{j=1}^{3} a_{fj} u_{ij}'$ and $\sum_{j=1}^{3} a_{fj} v_{ij}'$ are dropped to simplify equations (4-37) and (4-38) for the velocity corrections with the assumptions of these being negligible compared to the other terms. Patankar (1980) states that the Omission of these terms is the main approximation of the SIMPLE algorithm. Hence, we write

$$a_p u_i' = -(\frac{\partial p'}{\partial x})_i V_i \tag{4-39}$$

$$a_p v_i' = -(\frac{\partial p'}{\partial y})_i V_i. \tag{4-40}$$

Equations (4-39) and (4-40) describe the corrections to be applied to the velocities which, via (4-33) and (4-34) show that

$$u_i = u_i^* - \frac{V_i}{a_p}(\frac{\partial p'}{\partial x})_i \tag{4-41}$$

$$v_i = v_i^* - \frac{V_i}{a_p}(\frac{\partial p'}{\partial y})_i. \tag{4-42}$$

When the under-relaxation factor α for velocities is used, we obtain

$$u_i = u_i^* - \frac{V_i}{(\frac{a_p}{\alpha})}(\frac{\partial p'}{\partial x})_i \tag{4-43}$$

$$v_i = v_i^* - \frac{V_i}{(\frac{a_p}{\alpha})}(\frac{\partial p'}{\partial y})_i. \tag{4-44}$$

4.3.1.2 *The pressure correction equation*

Up to now we have only considered the momentum equations but as mentioned earlier, the velocity field is also subject to the constraint that it should satisfy the steady continuity equation

$$\frac{\partial}{\partial x}(\rho u) + \frac{\partial}{\partial y}(\rho v) = 0,$$

where ρ is the density of the fluid, which is constant for incompressible flow. Thus the continuity equation may be written here as

$$\rho(\frac{\partial u}{\partial x})_i + \rho(\frac{\partial v}{\partial y})_i = 0. \tag{4-45}$$

We must now evaluate the derivatives $(\frac{\partial u}{\partial x})_i$ and $(\frac{\partial v}{\partial y})_i$ at the element centres. This is done in a manner similar to that used for pressure gradients in Section 4.2.5.1. The result can be written as

$$(\frac{\partial u}{\partial x})_i = \frac{1}{V_i}\sum_{j=1}^{3} u_{fj} n_{xfj} A_{fj}$$

$$(\frac{\partial v}{\partial y})_i = \frac{1}{V_i}\sum_{j=1}^{3} v_{fj} n_{yfj} A_{fj}.$$

Substitution of these into equation (4-45) implies that

$$\rho(\frac{1}{V_i}\sum_{j=1}^{3}(u_{fj} n_{xfj} A_{fj} + v_{fj} n_{xfj} A_{fj})) = 0.$$

Using equation (4-5), the discrete form of the continuity equation for a element with an arbitrary number of faces (neighbours) can be expressed as

$$\rho\sum_{j=1}^{3} U^{fj} A_{fj} = 0.$$

The left-hand side of the equation is equal to the mass flux value described in Section 4.2.2, so that

$$\sum_{j=1}^{3} m_{fj} = \sum_{j=1}^{3} \rho U^{fj} A_{fj} = 0 , \qquad (4\text{-}46)$$

where U^{fj} is the face normal velocity which was defined in Section (4.2.2) and is obtained from:

$$U^{fj} = u_{fj} n_{xfj} + v_{fj} n_{yfj} = (\lambda_j u_i + (1-\lambda_j) u_{ij}) n_{xfj} + (\lambda_j v_i + (1-\lambda_j) v_{ij}) n_{xfj} . \qquad (4\text{-}47)$$

The velocity correction expressions (4-43) and (4-44) are substituted into equation (4-47) and relevant notation is defined in Figure 4.14, which gives

$$U^{fj} = \{\lambda_j(u_i^* - \frac{V_i}{(\frac{a_p}{\alpha})}(\frac{\partial p'}{\partial x})_i) + (1-\lambda_j)(u_{ij}^* - \frac{V_{ij}}{(\frac{a_p}{\alpha})}(\frac{\partial p'}{\partial x})_{ij})\}\, n_{xfj} +$$

$$\{\lambda_j(v_i^* - \frac{V_i}{(\frac{a_p}{\alpha})}(\frac{\partial p'}{\partial y})_i) + (1-\lambda_j)(v_{ij}^* - \frac{V_{ij}}{(\frac{a_p}{\alpha})}(\frac{\partial p'}{\partial y})_{ij})\}\, n_{yfj}$$

$$= \{\lambda_j u_i^* + (1-\lambda_j)u_{ij}^*\}n_{xfj} + \{\lambda_j v_i^* + (1-\lambda_j)v_{ij}^*\}n_{yfj}$$

$$-[\{\lambda_j \frac{V_i}{(\frac{a_p}{\alpha})}(\frac{\partial p'}{\partial x})_i + (1-\lambda_j)(\frac{V_{ij}}{(\frac{a_p}{\alpha})}(\frac{\partial p'}{\partial x})_{ij})\}\, n_{xfj} +$$

$$\{\lambda_j \frac{V_i}{(\frac{a_p}{\alpha})}(\frac{\partial p'}{\partial y})_i + (1-\lambda_j)(\frac{V_{ij}}{(\frac{a_p}{\alpha})}(\frac{\partial p'}{\partial y})_{ij})\}\, n_{yfj}]$$

$$= (u_{fj}^* n_{xfj} + v_{fj}^* n_{yfj}) - W_{fj}[(\frac{\partial p'}{\partial x})_{fj} n_{xfj} + (\frac{\partial p'}{\partial y})_{fj} n_{yfj}] = U^{fj^*} - W_{fj}(\frac{\partial p'}{\partial n})_{fj}, \qquad (4\text{-}48)$$

in which $(\frac{\partial p'}{\partial n})_{fj}$ defines the pressure correction gradient at the face and the coefficient of the face is defined as

$$W_{fj} = \lambda_j \{\frac{V_i}{(\frac{a_p}{\alpha})}\} + (1-\lambda_j)\{\frac{V_{ij}}{(\frac{a_p}{\alpha})}\}. \qquad (4\text{-}49)$$

Expanding equation (4-46), we have

$$\sum_{j=1}^{3} m_{fj} = \sum_{j=1}^{3} \rho U^{fj} A_{fj} = \rho U^{f1} A_{f1} + \rho U^{f2} A_{f2} + \rho U^{f3} A_{f3} = 0. \qquad (4\text{-}50)$$

Next, substituting (4-48) into (4-50) gives

$$\sum_{j=1}^{3} \rho U^{fj} A_{fj} = \rho U^{f1^*} A_{f1} - \rho A_{f1} W_{f1}(\frac{\partial p'}{\partial n})_{f1} + \rho U^{f2^*} A_{f2} - \rho A_{f2} W_{f2}(\frac{\partial p'}{\partial n})_{f2} +$$

$$\rho U^{f3^*} A_{f3} - \rho A_{f3} W_{f3}(\frac{\partial p'}{\partial n})_{f3} = \sum_{j=1}^{3} m_{fj}^* - \sum_{j=1}^{3} \rho A_{fj} W_{fj}(\frac{\partial p'}{\partial n})_{fj} = 0, \qquad (4\text{-}51)$$

where

$$\sum_{j=1}^{3} m_{fj}^{*} = \rho U^{f1^{*}} A_{f1} + \rho U^{f2^{*}} A_{f2} + \rho U^{f3^{*}} A_{f3}, \tag{4-52}$$

in which

$$(\frac{\partial p^{'}}{\partial n})_{fj} = (\frac{\partial p^{'}}{\partial x})_{fj} n_{xfj} + (\frac{\partial p^{'}}{\partial y})_{fj} n_{yfj}. \tag{4-53}$$

We must now evaluate the derivatives $(\frac{\partial p^{'}}{\partial x})_{fj}$ and $(\frac{\partial p^{'}}{\partial y})_{fj}$ at the face. This is achieved in a manner similar to that used for diffusion terms in Section 4.2.3. The result can be obtained as

$$(\frac{\partial p^{'}}{\partial x})_{fj} n_{xfj} = \frac{n_{xfj}}{V_{fj}}[n_{xfj} A_{fj} (p^{'})_{ij} + n_{xjend} A_{jend} (p^{'})_{jend} - n_{xfj} A_{fj} (p^{'})_{i} - n_{xiend} A_{jend} (p^{'})_{jst}]$$

$$(\frac{\partial p^{'}}{\partial y})_{fj} n_{yfj} = \frac{n_{yfj}}{V_{fj}}[n_{yfj} A_{fj} (p^{'})_{ij} + n_{yjend} A_{jend} (p^{'})_{jend} - n_{yfj} A_{fj} (p^{'})_{i} - n_{yjend} A_{jend} (p^{'})_{jst}].$$

Substituting these expressions into equation (4-53) yields

$$(\frac{\partial p^{'}}{\partial n})_{fj} = (\frac{\partial p^{'}}{\partial x})_{fj} n_{xj} + (\frac{\partial p^{'}}{\partial y})_{fj} n_{yj} =$$

$$\frac{1}{V_{fj}}[A_{fj}(p^{'}_{ij} - p^{'}_{i}) + A_{iend}(n_{xfj} n_{xjend} + n_{yfj} n_{yjend})(p^{'}_{jend} - p^{'}_{jst})], \tag{4-54}$$

and hence that

$$-\sum_{j=1}^{3} \rho A_{fj} W_{fj} (\frac{\partial p^{'}}{\partial n})_{fj} = -\sum_{j=1}^{3} \rho A_{fj} W_{fj} \frac{1}{V_{fj}}[A_{fj}(p^{'}_{ij} - p^{'}_{i}) + A_{iend}(n_{xfj} n_{xjend} + n_{yfj} n_{yjend})(p^{'}_{jend} - p^{'}_{jst})]. \tag{4-55a}$$

Expanding equation (4-51) and using equation (4-55a), we have

$$\sum_{j=1}^{3} m_{fj}^{*} + \rho A_{f1}^{2} W_{f1} \frac{1}{V_{f1}} (p_i^{'} - p_{i1}^{'}) + \rho A_{f2}^{2} W_{f2} \frac{1}{V_{f2}} (p_i^{'} - p_{i2}^{'}) +$$

$$\rho A_{f3}^{2} W_{f3} \frac{1}{V_{f3}} (p_i^{'} - p_{i3}^{'}) = S_{non}^{p}, \qquad (4\text{-}55b)$$

where S_{non}^{p} is the pressure correction gradient for non-orthogonal terms, given by

$$S_{non}^{p} = \sum_{j=1}^{nf} \rho A_{fj} W_{fj} \frac{1}{V_{fj}} A_{jend} (n_{xfj} n_{xiend} + n_{yfj} n_{yjend}) (p^{'}_{jend} - p^{'}_{jst}). \qquad (4\text{-}56)$$

Equation (4-55b) can then be written in the discretised form

$$a_i p_i^{'} = a_{pf1} p_{i1}^{'} + a_{pf2} p_{i2}^{'} + a_{pf3} p_{i3}^{'} + S_p,$$

where

$$a_{pf1} = \rho A_{f1}^{2} W_{f1} \frac{1}{V_{f1}}$$

$$a_{pf2} = \rho A_{f2}^{2} W_{f2} \frac{1}{V_{f2}}$$

$$a_{pf3} = \rho A_{f3}^{2} W_{f3} \frac{1}{V_{f3}} \qquad (4\text{-}57)$$

$$a_i = a_{pf1} + a_{pf2} + a_{pf3}$$

$$S_p = -\sum_{j=1}^{nf} m_{fj}^{*} + S_{non}^{p}$$

Having solved the pressure correction equation, the corrected pressure may then be simply computed from

$$p = p^{*} + \alpha_p p^{'}, \qquad (4\text{-}58)$$

where α_p is pressure under-relaxation factor and is chosen between 10^{-3} and 1.

4.3.1.3 *Rhie and Chow pressure term*

Collocated grid systems are often used in structured finite volume methods based on SIMPLE. In order to prevent oscillations in the pressure field, a third derivative term (three first derivative terms) in pressure, suggested by Rhie and Chow (1983), is added to the convection (normal) velocities. This term is added to the convection when calculating the continuity error in the pressure correction equation, but not to the convection in the u- and v- momentum equations. It is often implemented, for example by Davidson (1996), Jameson *et al.* (1986) and Rizzi (1982) as the difference between two first derivative terms, one evaluated at the centre of the element and the other at the element face. This third derivative term is an artificial dissipation term which can be regarded as a stabilising dissipation term.

At the element face, the normal pressure gradient is evaluated using equation (4-54) with reference to Figure 4.15.

$$(\frac{\partial p}{\partial n})_{fj} = \frac{1}{V_{fj}}[A_{fj}(p_{ij} - p_i) + A_{jend}(n_{xfj}n_{xjend} + n_{yfj}n_{yjend})(p_{jend} - p_{jst})]. \qquad (4\text{-}59)$$

The pressure gradient in the x and y component directions are evaluated at the element centres i and ij by

$$(\frac{\partial p}{\partial x})_i = \frac{1}{V_i}\sum_{j=1}^{nf} p_{fj}n_{xfj}A_{fj}, \qquad (\frac{\partial p}{\partial y})_i = \frac{1}{V_i}\sum_{j=1}^{nf} p_{fj}n_{yfj}A_{fj},$$

$$(\frac{\partial p}{\partial x})_{ij} = \frac{1}{V_{ij}}\sum_{j=1}^{nf} p_{fj}n_{xfj}A_{fj}, \qquad (\frac{\partial p}{\partial y})_{ij} = \frac{1}{V_{ij}}\sum_{j=1}^{nf} p_{fj}n_{xfj}A_{fj}. \qquad (4\text{-}60)$$

Then pressure gradients at the element centre are projected on the outward-pointing normal vector at the face fj respectively to give, with reference to Figure 4.15,

$$(\frac{\partial p}{\partial n_j})_i = (\frac{\partial p}{\partial x})_i n_{xfj} + (\frac{\partial p}{\partial y})_i n_{yfj} \tag{4-61}$$

$$(\frac{\partial p}{\partial n_j})_{ij} = (\frac{\partial p}{\partial x})_{ij} n_{xfj} + (\frac{\partial p}{\partial y})_{ij} n_{yfj} . \tag{4-62}$$

The equations (4-59), (4-61) and (4-62) present three first derivative terms. The difference between the pressure gradient at the element face and the pressure gradient at the elements centre, which is uses to calculate the third derivative term (see Davidson (1996)), is given by:

$$(\frac{\partial p}{\partial n})_{fj} - [\lambda_j (\frac{\partial p}{\partial n})_i + (1-\lambda_j)(\frac{\partial p}{\partial n})_{ij}]. \tag{4-63}$$

The third derivative term is then add to the term, S_p, which is the continuity error in the pressure correction equation (see equation (4-57). Now the mass flux m^*_{fj} in the S_p can be written as

$$m^*_{fj} = (\rho A_{fj} U^{fj*}) - \rho A_{fj} W_{fj} \{ (\frac{\partial p}{\partial n})_{fj} - [\lambda_j (\frac{\partial p}{\partial n})_i + (1-\lambda_j)(\frac{\partial p}{\partial n})_{ij}] \}, \tag{4-64}$$

where U^{fj*} is the updated value for the normal velocity at the face.

4.3.1.4 *The corrected mass flux*

Upon solving the pressure correction equation, we next correct the mass flux in the manner

$$m_{fj} = m^*_{fj} + m'_{fj} = m^*_{fj} + \rho A_{fj} U'^{fj} \tag{4-65}$$

where U'^{fj} is the face normal velocity correction, defined by:

$$U'^{fj} = u'_{fj} n_{xfj} + v'_{fj} n_{yfj} = (\lambda_j u'_i + (1-\lambda_j) u'_{ij}) n_{xfj} + (\lambda_j v'_i + (1-\lambda_j) v'_{ij}) n_{yfj} . \tag{4-66}$$

Substituting equations (4-39) and (4-40) into (4-66) yields

$$U'^{fj} = \{\lambda_j(-\frac{V_i}{(\frac{a_p}{\alpha})}(\frac{\partial p'}{\partial x})_i) + (1-\lambda_j)(-\frac{V_{ij}}{(\frac{a_p}{\alpha})}(\frac{\partial p'}{\partial x})_{ij})\}\, n_{xfj} +$$

$$\{\lambda_j(-\frac{V_i}{(\frac{a_p}{\alpha})}(\frac{\partial p'}{\partial y})_i) + (1-\lambda_j)(-\frac{V_{ij}}{(\frac{a_p}{\alpha})}(\frac{\partial p'}{\partial y})_{ij})\}\, n_{yfj}$$

$$= -[\{\lambda_j \frac{V_i}{(\frac{a_p}{\alpha})}(\frac{\partial p'}{\partial x})_i + (1-\lambda_j)(\frac{V_{ij}}{(\frac{a_p}{\alpha})}(\frac{\partial p'}{\partial x})_{ij})\}\, n_{xfj} +$$

$$\{\lambda_j \frac{V_i}{(\frac{a_p}{\alpha})}(\frac{\partial p'}{\partial y})_i + (1-\lambda_j)(\frac{V_{ij}}{(\frac{a_p}{\alpha})}(\frac{\partial p'}{\partial y})_{ij})\}\, n_{yfj}]$$

$$= -\frac{V_{fj}}{(\frac{a_p}{\alpha})}[(\frac{\partial p'}{\partial x})_{fj} n_{xfj} + (\frac{\partial p'}{\partial y})_{fj} n_{yfj}] = -\frac{V_{fj}}{(\frac{a_p}{\alpha})}(\frac{\partial p'}{\partial n})_{fj} = -W_{fj}(\frac{\partial p'}{\partial n})_{fj}. \qquad (4\text{-}67)$$

Inserting equation (4-67) into (4-65) leads to

$$m_{fj} = m_{fj}^* + m_{fj}' = m_{fj}^* + \rho A_{fj} U'^{fj} = \rho A_{fj} U^{fj*} - \rho A_{fj} W_{fj}(\frac{\partial p'}{\partial n})_{fj} \qquad (4\text{-}68)$$

in which $(\frac{\partial p'}{\partial n})_{fj}$ is the pressure correction gradient at the face and can be obtained as equation (4-54).

4.3.1.5 *The solution procedure for SIMPLE*

The whole solution algorithm for SIMPLE is summarised in the following sequence of steps:

(a) initial pressure and velocity fields are defined;

(b) the coefficients a_p and a_{fj} are computed from equation (4-21) (looping over faces), and under-relaxation is introduced;

(c) the momentum equations (4-28) and (4-29) are solved using a point by point Gauss-Seidel solver (looping over elements);

(d) the coefficients for the p' equation are computed from equation (4-57) (looping over faces);

(e) the pressure-correction equations are computed from equation (4-57) (looping over elements);

(f) the pressure is corrected using equation (4-58) and velocities are corrected using equations (4-43) and (4-44) (looping over elements). The mass flux is corrected using equation (4-68); and

(g) steps (a)-(h) are repeated until convergence is achieved, based on the assumption that the mass residuals in all elements or the sum of normalised residuals have fallen below a prescribed value.

4.3.2 The MAC algorithm

One of the earliest, and most widely used, methods for solving the Navier-Stokes equations is the Marker and Cell (MAC) method, due to Harlow and Welch (1965). The MAC method was developed for the time-dependent, numerical solution of confined free-surface flows for a viscous incompressible fluid. The method is applicable to a wide variety of problems.

The purpose of this section is to describe the MAC method for obtaining the pressure field as an alternative to SIMPLE.

4.3.2.1 *The Poisson equation for pressure*

The derivation of the Poisson equation for pressure used in the MAC method starts with the Navier-Stokes equations. The conditions of incompressibility states that the divergence of the fluid velocity must vanish, i.e.

$$\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}=0 \tag{4-69}$$

The fluid pressure, p, satisfies an equation that is derived by taking the divergence of the Navier-Stokes momentum equations, which, written in Cartesian form are:

$$u\frac{\partial u}{\partial x}+v\frac{\partial u}{\partial y}=-\frac{1}{\rho}\frac{\partial p}{\partial x}+\mu\left(\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}\right) \tag{4-70}$$

$$u\frac{\partial v}{\partial x}+v\frac{\partial v}{\partial y}=-\frac{1}{\rho}\frac{\partial p}{\partial x}+\mu\left(\frac{\partial^2 v}{\partial x^2}+\frac{\partial^2 v}{\partial y^2}\right). \tag{4-71}$$

Pressure is implicitly specified by the continuity equation. To determine the pressure explicitly, one must derive an auxiliary equation from a combination of momentum and continuity equations. In the MAC algorithm, which was initially developed for incompressible flow, an equation for pressure is derived as follows.

Adding the $x-$derivative of equation (4-70) to the $y-$derivative of equation (4-71), and using the continuity equation (4-69) for further reduction, leads to

$$\left(\frac{\partial u}{\partial x}\right)^2+\frac{\partial v}{\partial x}\frac{\partial v}{\partial y}+\left(\frac{\partial v}{\partial y}\right)^2+\frac{\partial u}{\partial y}\frac{\partial v}{\partial x}=-\frac{1}{\rho}\nabla^2 p, \tag{4-72}$$

which, in vector notation, reads

$$\nabla^2 p=-\rho\nabla\cdot[(\vec{v}\cdot\nabla)\vec{v}] \tag{4-73}$$

where ∇^2 is the Laplacian and $\vec{v}=u\vec{i}+v\vec{j}$.

Hence, we have derived a Poisson equation for the pressure, which in finite volume form, is most efficiently solved by an iteration process. Since an iteration process must be terminated after a finite number of iterations, some error is introduced. This error corresponds to an error in the total volume of fluid, so it must remain small, in order

that the numerical solution of the Navier-Stokes equations is sufficiently accurate.

4.3.2.2 *Discretisation*

Integrating equation (4-73) over a control volume (see Figure 4.2) yields

$$\sum_{j=1}^{3}(\frac{\partial p}{\partial n})_{fj} A_{fj} = -\rho\sum_{j=1}^{3}[(\bar{v}\cdot\nabla)\bar{v}]\bar{n}_{fj}\cdot A_{fj}$$

$$=-\rho\sum[(u_{fj}(\frac{\partial u}{\partial x})_{fj}+v_{fj}(\frac{\partial u}{\partial y})_{fj})\cdot n_{xfj}+(u_{fj}(\frac{\partial v}{\partial x})_{fj}+v_{fj}(\frac{\partial v}{\partial y})_{fj})\cdot n_{yfj}]\cdot A_{fj}. \qquad (4\text{-}74)$$

On the left-hand side of equation (4-74), $(\frac{\partial p}{\partial n})_{fj}$ is calculated by

$$(\frac{\partial p}{\partial n})_{fj}=\frac{1}{V_{fj}}[A_{fj}(p_{ij}-p_i)+A_{jend}(n_{xfj}n_{xjend}+n_{ufj}n_{yjend})(p_{jend}-p_{jst}). \qquad (4\text{-}75)$$

Substituting equation (4-75) into the left-hand side of the equation (4-74) and expanding the left-hand side expression for a triangular grid gives:

$$\sum_{j=1}^{3}(\frac{\partial p}{\partial n})_{fj} A_{fj} = -p_i(\frac{A_{f1}^2}{V_{f1}}+\frac{A_{f2}^2}{V_{f2}}+\frac{A_{f3}^2}{V_{f3}})+\frac{A_{f1}^2}{V_{f1}}p_{i1}+\frac{A_{f2}^2}{V_{f2}}p_{i2}+\frac{A_{f3}^2}{V_{f3}}p_{i3}$$

$$+\frac{A_{f1}}{V_{f1}}A_{jend1}(n_{xf1}n_{xjend1}+n_{yf1}n_{yjend1})(p_{jend1}-p_{jst1})$$

$$+\frac{A_{f2}}{V_{f2}}A_{jend2}(n_{xf2}n_{xjend2}+n_{yf2}n_{yjend2})(p_{jend2}-p_{jst2})$$

$$+\frac{A_{f3}}{V_{f3}}A_{jend3}(n_{xf3}n_{xjend3}+n_{yf3}n_{yjend3})(p_{jend3}-p_{jst3}). \qquad (4\text{-}76)$$

On the right-hand side of equation (4-74), the velocity-gradient components at the faces $(\frac{\partial u}{\partial x})_{fj},(\frac{\partial u}{\partial y})_{fj},(\frac{\partial v}{\partial x})_{fj},(\frac{\partial v}{\partial y})_{fj}$ are calculated by applying Green's formula to the

volume *a-b-c-d* (see Figure 4-9) surrounding the midpoint of the face, and can be written as:

$$(\frac{\partial u}{\partial x})_{fj} = \frac{1}{V_{fj}}(n_{xfj}A_{fj}u_{ij} + n_{xjend}A_{jend}u_{jend} - n_{xfj}A_{fj}u_i - n_{xjend}A_{jend}u_{jst}) ,$$

$$(\frac{\partial u}{\partial y})_{fj} = \frac{1}{V_{fj}}(n_{yfj}A_{fj}u_{ij} + n_{yjend}A_{jend}u_{jend} - n_{yfj}A_{fj}u_i - n_{yjend}A_{jend}u_{jst}),$$

$$(\frac{\partial v}{\partial x})_{fj} = \frac{1}{V_{fj}}(n_{xfj}A_{fj}v_{ij} + n_{xjend}A_{jend}v_{jend} - n_{xfj}A_{fj}v_i - n_{xjend}A_{jend}v_{jst}) , \quad (4\text{-}77)$$

$$(\frac{\partial v}{\partial y})_{fj} = \frac{1}{V_{fj}}(n_{yfj}A_{fj}v_{ij} + n_{yjend}A_{jend}v_{jend} - n_{yfj}A_{fj}v_i - n_{yjend}A_{jend}v_{jst}).$$

Substituting equation (4-77) into the right-hand side of the equation (4-74) gives

$$-\rho\sum_{j=1}^{3}[u_{fj}A_{fj}\frac{n_{xfj}}{V_{fj}}(n_{xfj}A_{fj}u_{ij} + n_{xjend}A_{jend}u_{jend} - n_{xfj}A_{fj}u_i - n_{xjend}A_{jend}u_{jst})$$

$$+v_{fj}A_{fj}\frac{n_{xfj}}{V_{fj}}(n_{yfj}A_{fj}u_{ij} + n_{yjend}A_{jend}u_{jend} - n_{yfj}A_{fj}u_i - n_{yjend}A_{jend}u_{jst}) \quad (4\text{-}78)$$

$$+u_{fj}A_{fj}\frac{n_{yfj}}{V_{fj}}(n_{xfj}A_{fj}v_{ij} + n_{xjend}A_{jend}v_{jend} - n_{xfj}A_{fj}v_i - n_{xjend}A_{jend}v_{jst})$$

$$+v_{fj}A_{fj}\frac{n_{yfj}}{V_{fj}}(n_{yfj}A_{fj}v_{ij} + n_{yjend}A_{jend}v_{jend} - n_{yfj}A_{fj}v_i - n_{yjend}A_{jend}v_{jst})],$$

and furthermore, when equation (4-76) is used to replace the left-hand side of equation (4-74), we have the discretised equation:

$$a_i p_i = a_{pf1}p_{i1} + a_{pf2}p_{i2} + a_{pf3}p_{i3} + S_p^m . \quad (4\text{-}79)$$

Here $a_i = \frac{A_{f1}^2}{V_{f1}} + \frac{A_{f2}^2}{V_{f2}} + \frac{A_{f3}^2}{V_{f3}}$, $a_{pf1} = \frac{A_{f1}^2}{V_{f1}}$, $a_{pf2} = \frac{A_{f2}^2}{V_{f2}}$, $a_{pf3} = \frac{A_{f3}^2}{V_{f3}}$,

and $S_p^m = \frac{A_{f1}}{V_{f1}}A_{jend1}(n_{xf1}n_{xjend1} + n_{yf1}n_{yjend1})(p_{jend1} - p_{jst1})$

$$+\frac{A_{f2}}{V_{f2}}A_{jend2}(n_{xf2}n_{xjend2}+n_{yf2}n_{yjend2})(p_{jend2}-p_{jst2})$$

$$+\frac{A_{f3}}{V_{f3}}A_{jend3}(n_{xf3}n_{xjend3}+n_{yf3}n_{yjend3})(p_{jend3}-p_{jst3})$$

$$+\rho\sum_{j=1}^{3}[u_{fj}A_{fj}\frac{n_{xfj}}{V_{fj}}(n_{xfj}A_{fj}u_{ij}+n_{xjend}A_{jend}u_{jend}-n_{xfj}A_{fj}u_{i}-n_{xjend}A_{jend}u_{jst})$$

$$+v_{fj}A_{fj}\frac{n_{xfj}}{V_{fj}}(n_{yfj}A_{fj}u_{ij}+n_{yjend}A_{jend}u_{jend}-n_{yfj}A_{fj}u_{i}-n_{yjend}A_{jend}u_{jst})$$

$$+u_{fj}A_{fj}\frac{n_{yfj}}{V_{fj}}(n_{xfj}A_{fj}v_{ij}+n_{xjend}A_{jend}v_{jend}-n_{xfj}A_{fj}v_{i}-n_{xjend}A_{jend}v_{jst})$$

$$+v_{fj}A_{fj}\frac{n_{yfj}}{V_{fj}}(n_{yfj}A_{fj}v_{ij}+n_{yjend}A_{jend}v_{jend}-n_{yfj}A_{fj}v_{i}-n_{yjend}A_{jend}v_{jst})].$$

An alternative form of the Poisson equation for the pressure in equation (4-73) is now described. On the right-hand side of equation (4-74), the continuity equation is used to simplify the velocity-gradient component expressions, i.e. we substitute

$$(\frac{\partial u}{\partial x})_{fj}=-(\frac{\partial v}{\partial y})_{fj} \tag{4-80}$$

into the right-hand side of the equation (4-74), so that each term in the summation there is rewritten as:

$$u_{fj}[-(\frac{\partial v}{\partial y})_{fj}n_{xfj}+(\frac{\partial v}{\partial x})_{fj}n_{yfj}]+v_{fj}[(\frac{\partial u}{\partial y})_{fj}n_{xfj}-(\frac{\partial u}{\partial x})_{fj}n_{yfj}], \tag{4-81}$$

where $n_{xfj}=\sin\beta$, $n_{yfj}=-\cos\beta$ and (n_{xlj},n_{ylj}) is the unit parallel vector of the face (see Figure 4.16). Also, $n_{xlj}=\cos\beta$, $n_{ylj}=\sin\beta$, thus the expression (4-81) can be written as:

$$u_{fj}[-(\frac{\partial v}{\partial y})_{fj}\sin\beta-(\frac{\partial v}{\partial x})_{fj}\cos\beta]+v_{fj}[(\frac{\partial u}{\partial y})_{fj}\sin\beta+(\frac{\partial u}{\partial x})_{fj}\cos\beta] = -u_{fj}\frac{\partial v}{\partial l}+v_{fj}\frac{\partial u}{\partial l}.$$

Thus the right-hand side of equation (4-74) can be written as:

$$-\rho\sum_{j=1}^{3}(-u_{fj}\frac{\partial v}{\partial l}+v_{fj}\frac{\partial u}{\partial l})dl=-\rho[-u_{fj}(v_{jend}-v_{jst})+v_{fj}(u_{jend}-u_{jst})]. \qquad (4\text{-}82)$$

Substituting equation (4-76) into equation (4-74) and replacing its right-hand side with equation (4-82), we have the discretised form:

$$a_i p_i = a_{pf1}p_{i1}+a_{pf2}p_{i2}+a_{pf3}p_{i3}+S_p^m. \qquad (4\text{-}83)$$

where the coefficients a_i, a_{pf1}, a_{pf2} and a_{pf3} are the same as in equation(4-79) and

$$\begin{aligned}S_p^m = {} & \frac{A_{f1}}{V_{f1}}A_{jend1}(n_{xf1}n_{xjend1}+n_{yf1}n_{yjend1})(p_{jend1}-p_{jst1})\\ & +\frac{A_{f2}}{V_{f2}}A_{jend2}(n_{xf2}n_{xjend2}+n_{yf2}n_{yjend2})(p_{jend2}-p_{jst2})\\ & +\frac{A_{f3}}{V_{f3}}A_{jend3}(n_{xf3}n_{xjend3}+n_{yf3}n_{yjend3})(p_{jend3}-p_{jst3})\\ & +\rho[-u_{f1}(v_{jend1}-v_{jst1})+v_{f1}(u_{jend1}-u_{jst1})]\\ & +\rho[-u_{f2}(v_{jend2}-v_{jst2})+v_{f2}(u_{jend2}-u_{jst2})]\\ & +\rho[-u_{f3}(v_{jend3}-v_{jst3})+v_{f3}(u_{jend3}-u_{jst3})].\end{aligned}$$

4.3.2.3 *The solution procedure for the MAC method*

The whole solution algorithm for MAC is summarised in the following sequence of steps:

(a) initial pressure and velocity fields are defined;

(b) the coefficients a_p and a_{fj} are computed from equation (4-21) (looping over faces), and under-relaxation is introduced;

(c) the momentum equations (4-28) and (4-29) are solved using a point by point Gauss-

Seidel solver (looping over elements);

(d) the coefficients for the Poisson pressure equation are computed from equation (4-79) or (4-83) (looping over faces);

(e) the discretised form of Poisson equation for pressure (4-79) or (4-83) is solved using a point by point Gauss-Seidel solver (looping over elements);

(f) steps (a)-(e) are repeated until convergence is achieved, based on the assumption that the mass residuals in all elements value or the sum of normalised residuals have fallen below a prescribed value.

The procedure used for the MAC algorithm is similar in many ways to that used for the SIMPLE algorithm; however, the MAC procedure for calculating the pressure is significantly different.

4.4 Unsteady Flow Calculation

Having derived above the finite volume method for steady flows, the more complex category of time-dependent flow problems is now considered. The first term of equation (4-1) represents the rate of change term and is not equal to zero for unsteady flows. To predict time-dependent flow problems this term must be retained in the discretisation process.

Attention is drawn to the integration over a finite time step Δt, the rest of the integral terms of the finite volume discretisation are the same as in steady flows (convection, diffusion, source terms and pressure). The fully implicit scheme is based on that described by Versteeg and Malalasekera (1995). As explained by Versteeg and Malalasekera (1995) "the implicit method is recommended for general purpose transient calculations because of its robustness and unconditional stability." If the face values are evaluated using the QUICK scheme defined in equation (4-11), the discretised equations are same as those of steady problems apart from minor changes to the central coefficient a_p:

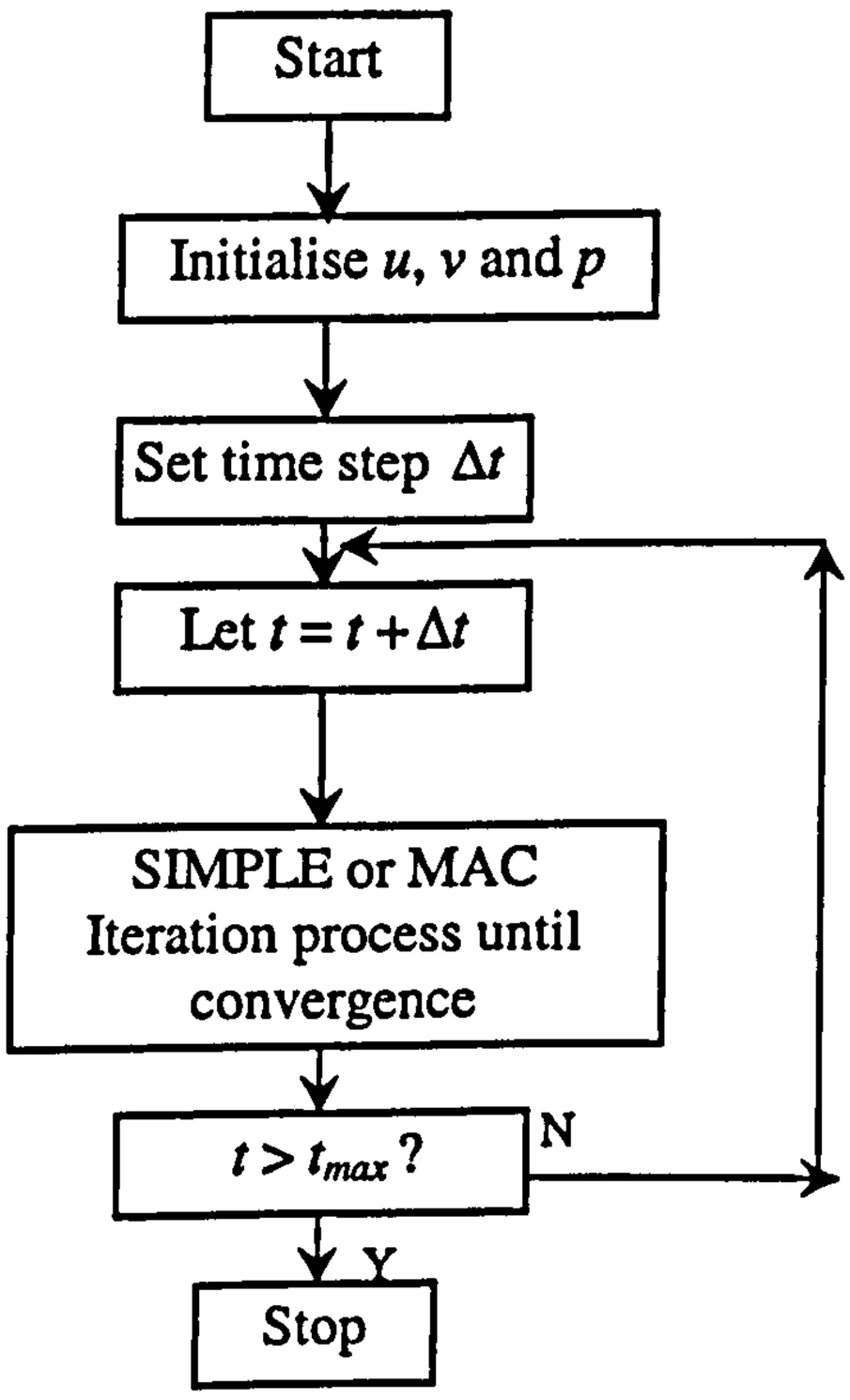
$$a_p \phi_i = a_{f1}\phi_{i1} + a_{f2}\phi_{i2} + a_{f3}\phi_{i3} + S^{\psi} + \sum_{j=1}^{3} C_j m_{fj} \phi_p + a_p^0 \phi_i^{(n-1)}$$

$$a_p = (B_1 m_{f1} + K_{f1}) + (B_2 m_{f2} + K_{f2}) + (B_3 m_{f3} + K_{f3}) + a_p^0 \qquad (4\text{-}84)$$

$$a_p^0 = \rho V_i / \Delta t,$$

where the coefficient a_p^0 is included and $\phi_i^{(n-1)}$ indicates the variable of the previous time step. The rest of the coefficients are practically the same as those of equation (4-21b). The complete algorithm for unsteady flow is summarised as follows,

Algorithm:



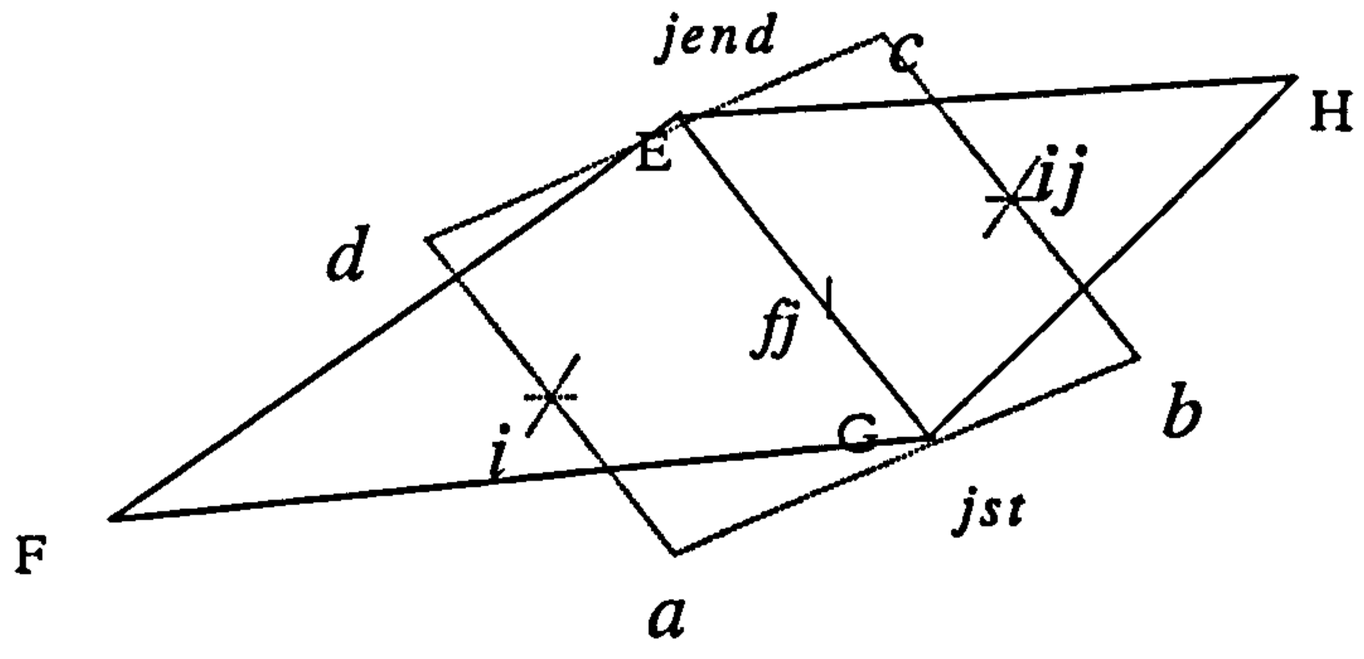


Figure 4.1 Notation for control volumes ΔEFG and ΔFGH. i and ij are the left and right elements respectively. jst and jend are the starting and ending grid points for a given face index fj respectively.

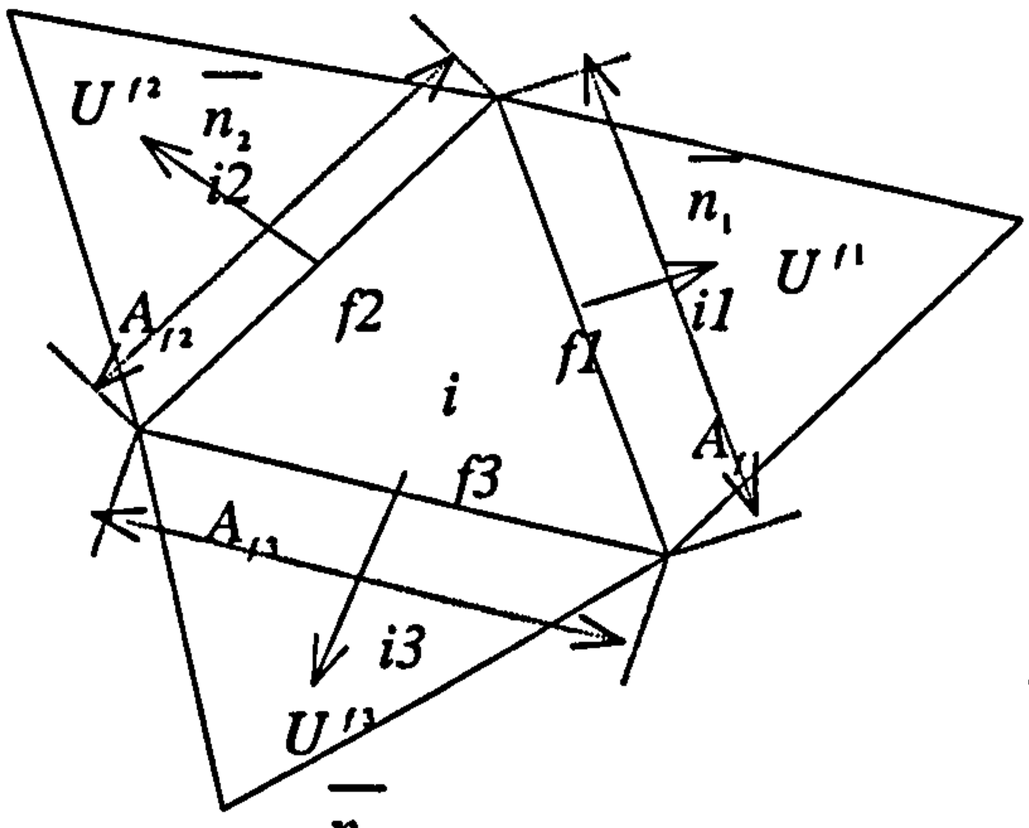


Figure 4.2 A triangular control volume i with its three faces f1, f2, f3 and three neighbours i1, i2, i3.

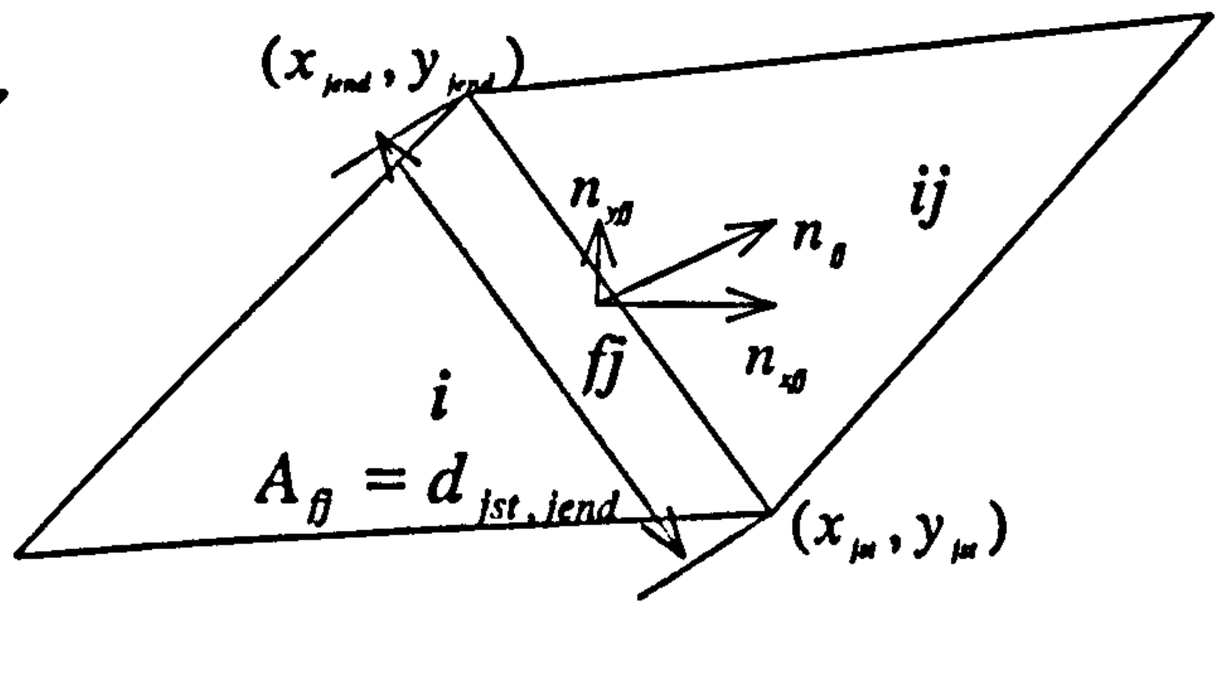


Figure 4.3 Notation associated with (n_{xfj}, n_{xfj})

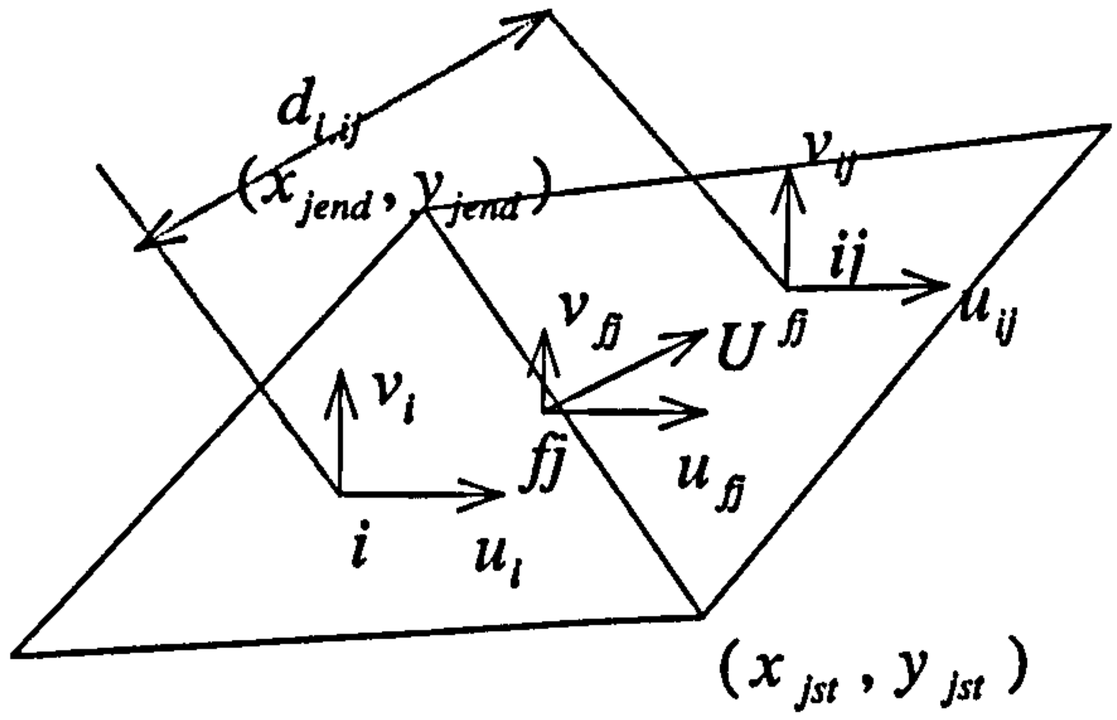


Figure 4.4 Notation associated with face velocity calculation

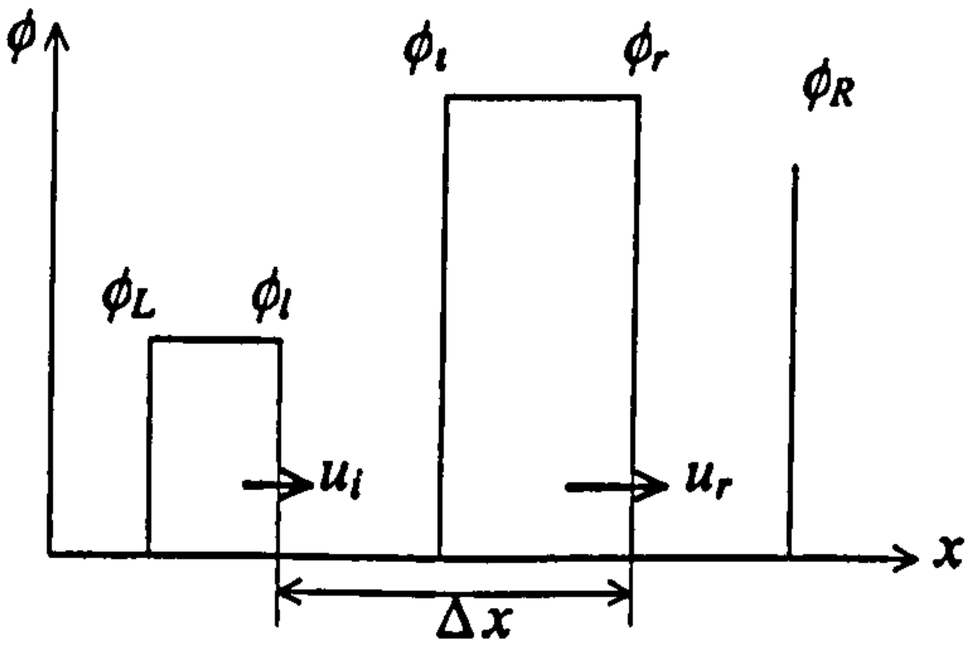


Figure 4.5 cell-face values for upwind (donor-cell) differencing (u_l and $u_r > 0$)

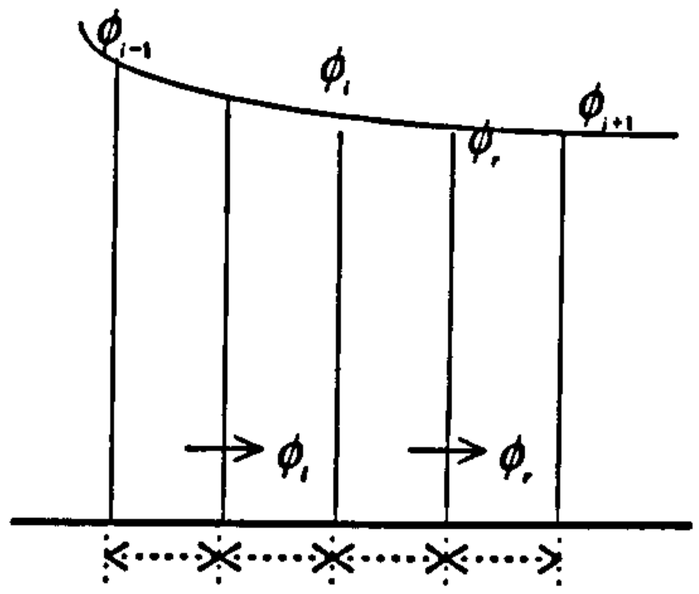


Figure 4.6 Quadratic upstream interpolation for face value ϕ_r

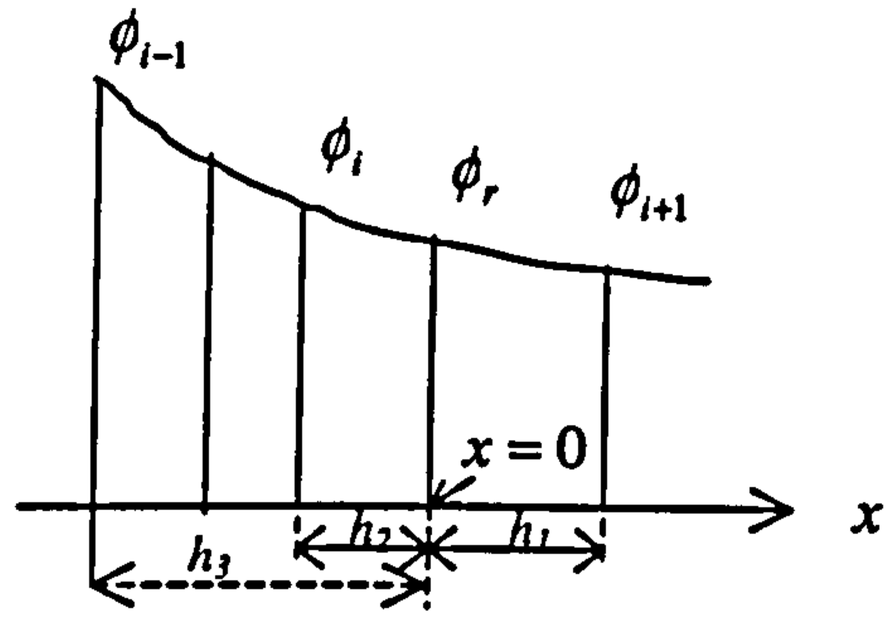


Figure 4.7 Quadratic upstream interpolation for face value ϕ_r

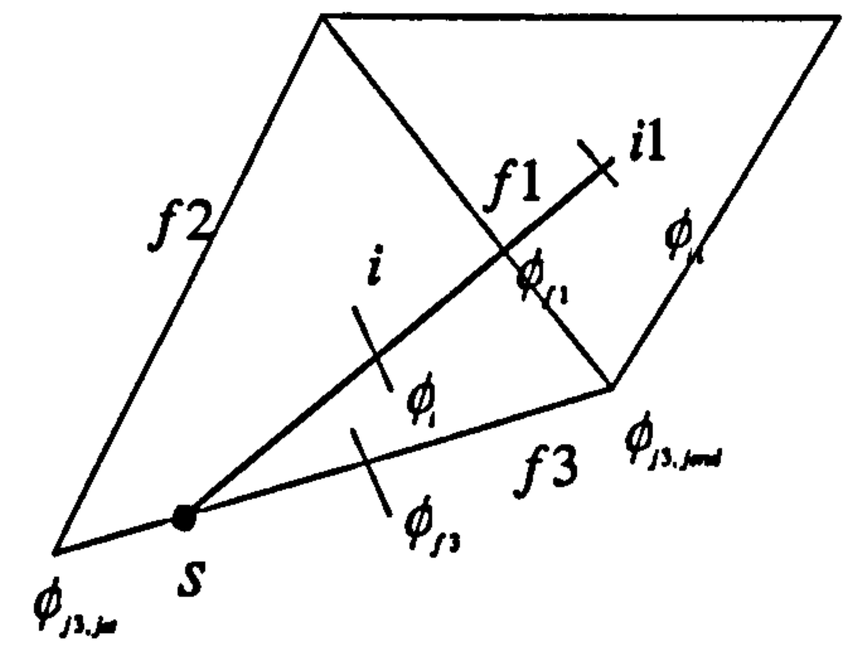


Figure 4.8 Notation associated with a face value when using the QUICK differencing

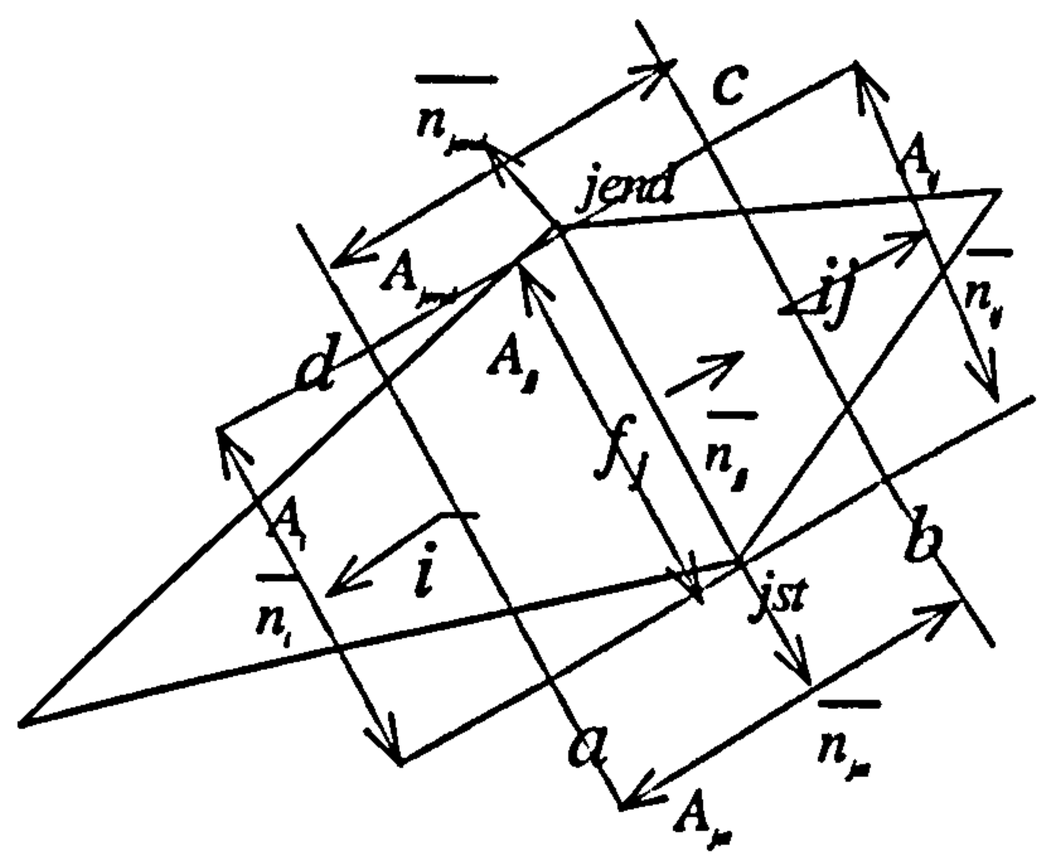


Figure 4.9 Notation associated with equation (4-13)

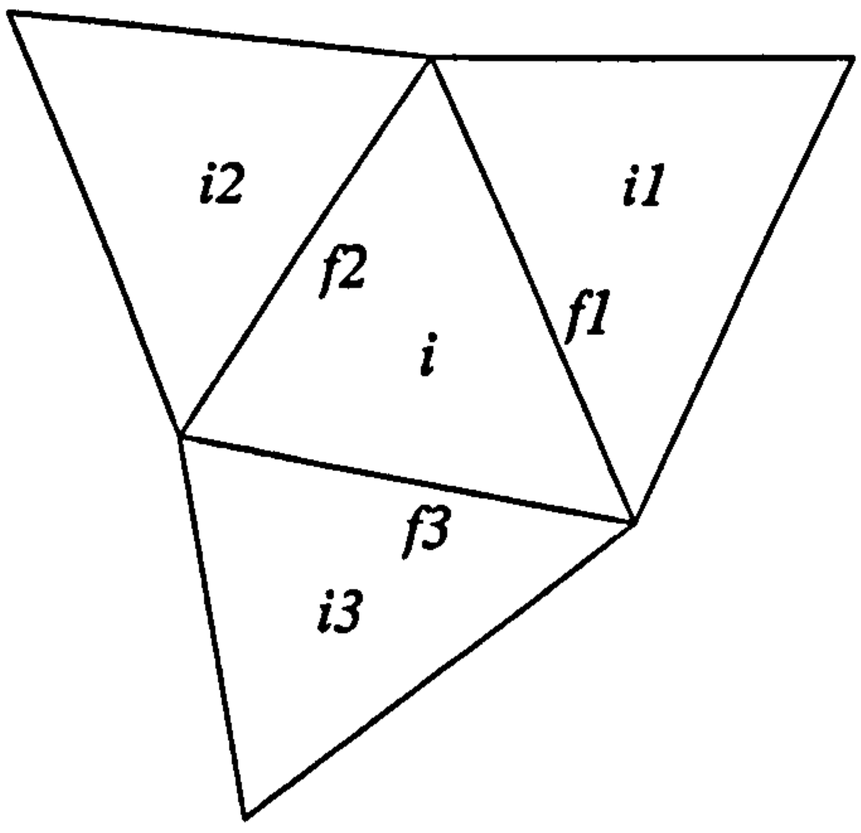


Figure 4.10 Notation associated with equation (4-20)

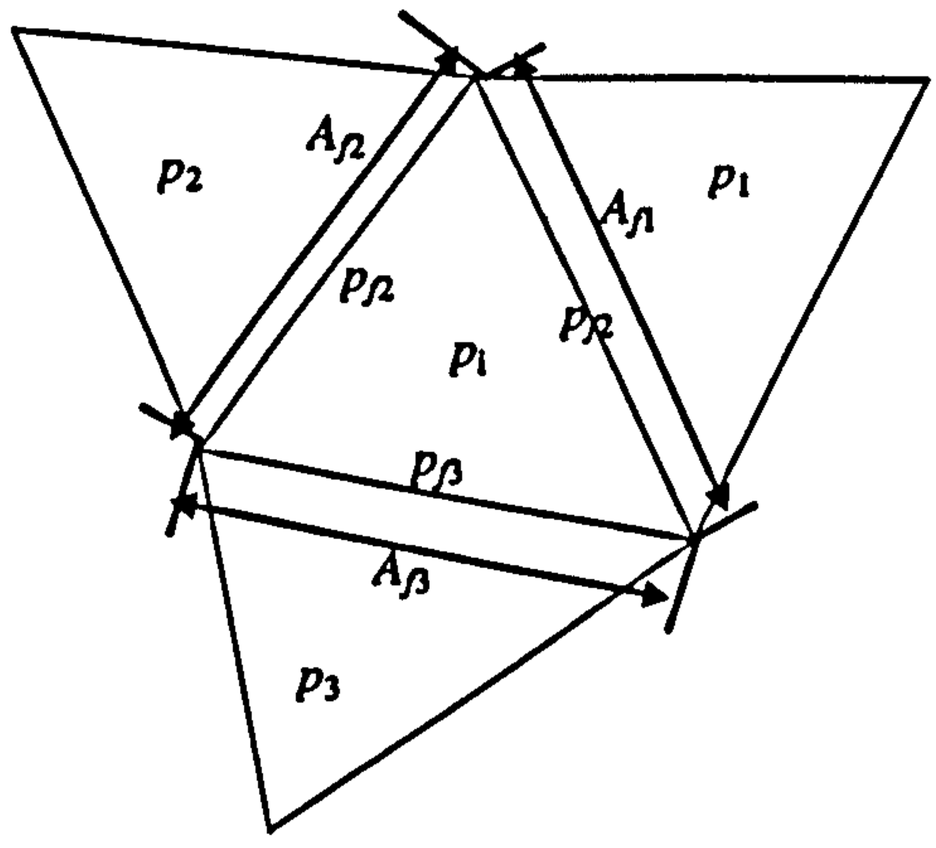


Figure 4.11 Notation associated with equations (4-22) and (4-23)

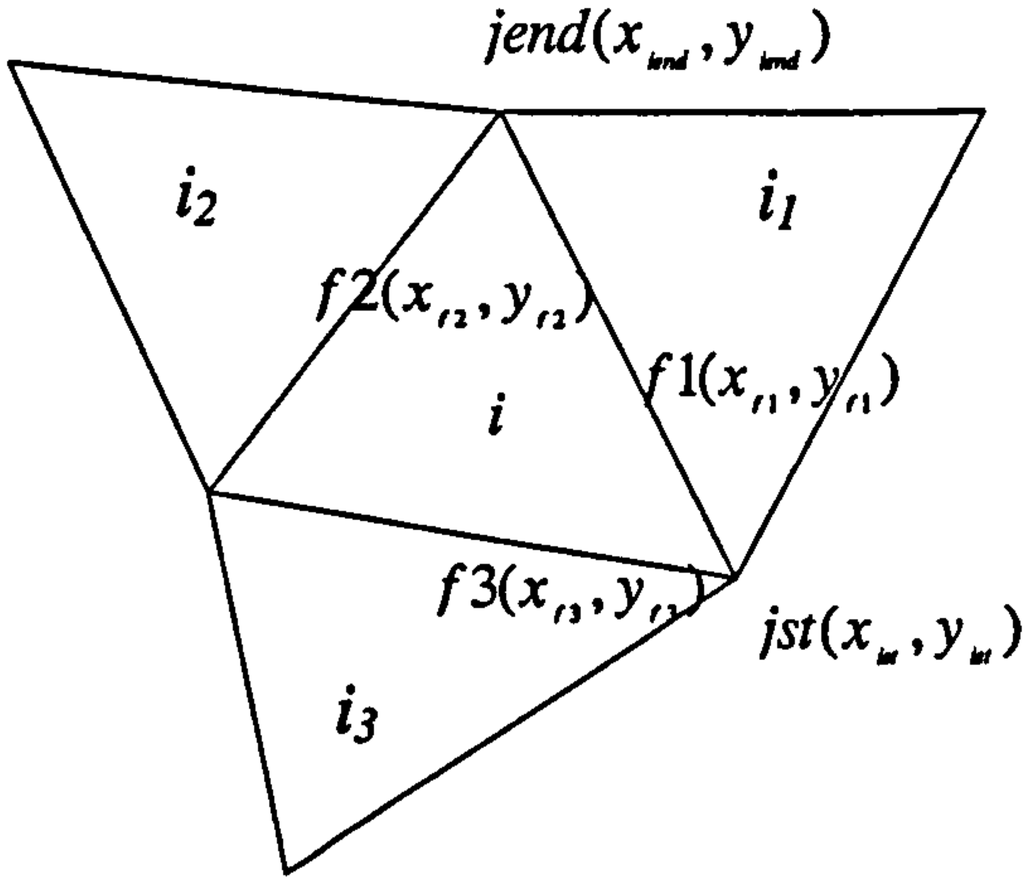


Figure 4.12 Notation associated with ϕ_{jend}, ϕ_{jst}

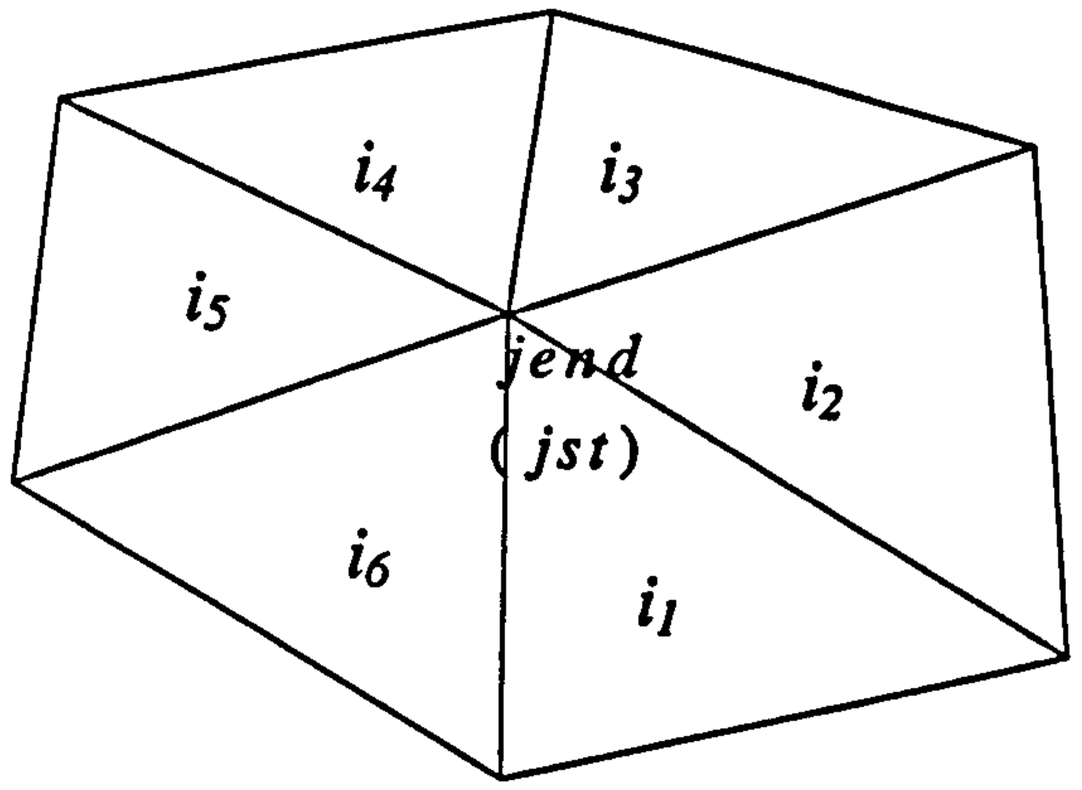


Figure 4.13 Notation associated with averaging method

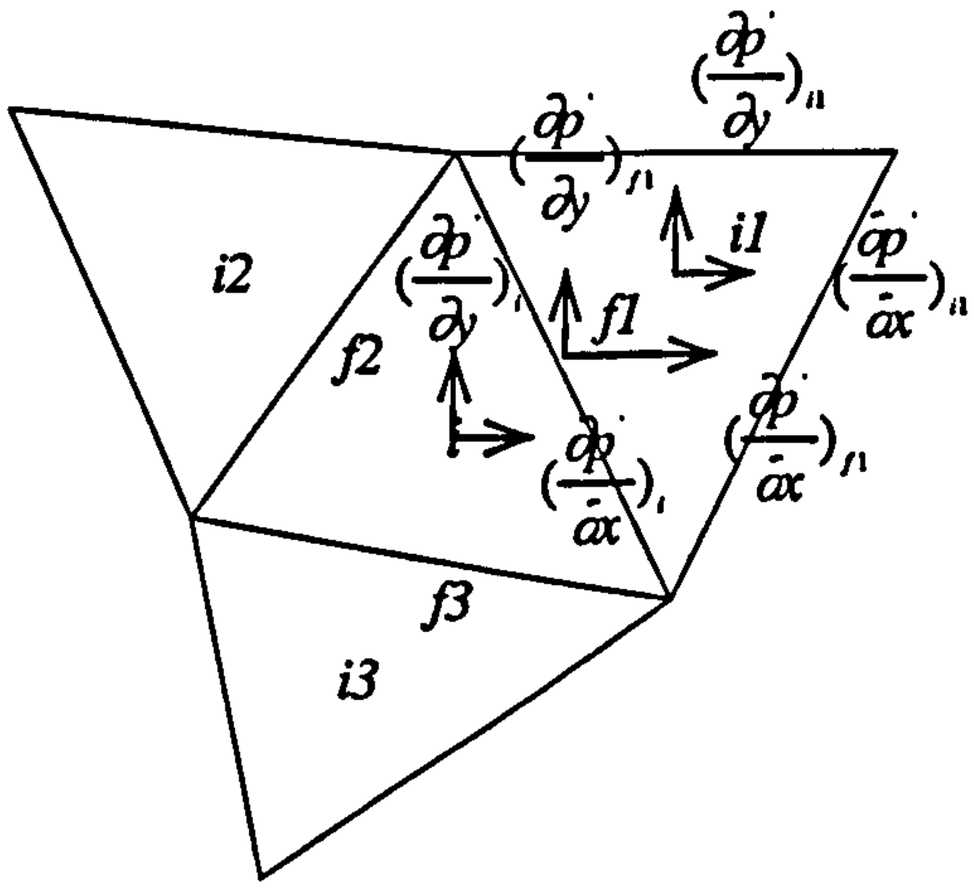


Figure 4.14 Notation associated with equation (4-47)

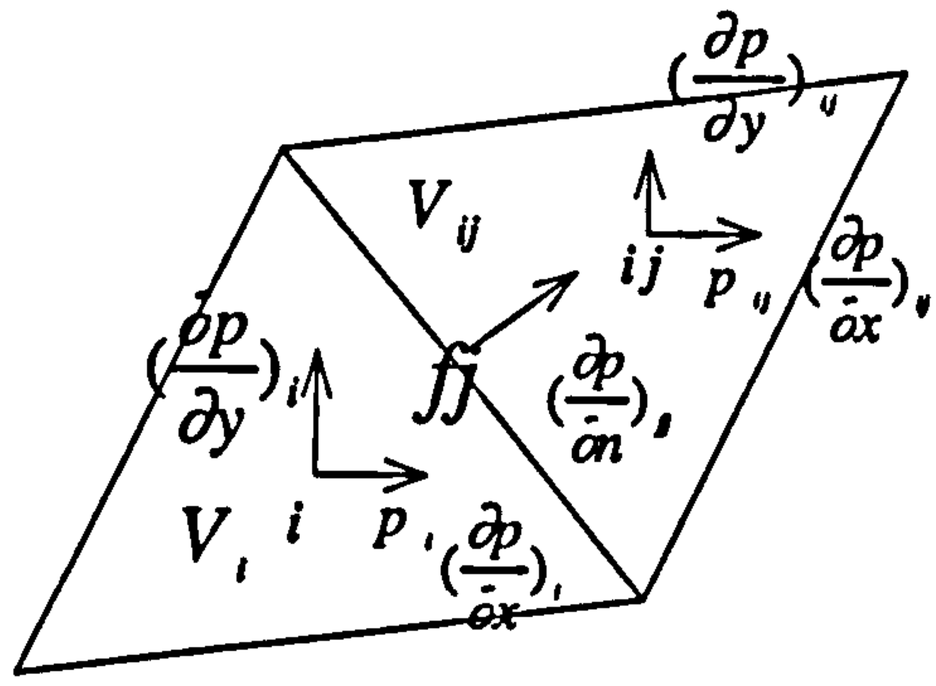


Figure 4.15 Notation associated with the normal pressure gradient at the face and equations (4-59) and (4-60)

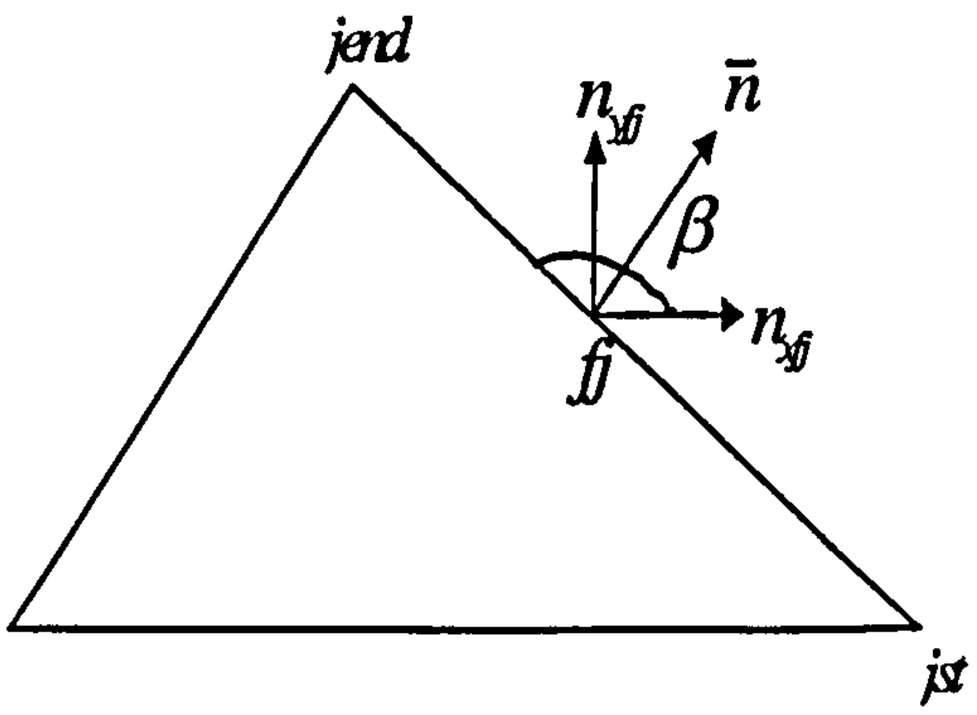


Figure 4.16 Notation associated with equation (4-81)

CHAPTER 5

RESULTS AND DISCUSSION FOR LAMINAR CHANNEL FLOW

5.1 General Description

In this Chapter, the numerical results for two-dimensional incompressible laminar flow are presented. As will be seen, the results are used to show that the accuracy of the code is satisfactory and to assure the use of various schemes within the code.

Figure 5.1 illustrates a simple method in which each rectangular element is divided into two triangular elements. This kind of triangular grid is used for the channel flow case. Section 5.2 gives the definitions of the boundary conditions for the channel flow and Section 5.3 describes the low-Reynolds-number flow. In the preliminary stages of the work, attention was focused on validating the programs of both SIMPLE and MAC algorithms. To do this, we have investigated this simple case. Laminar channel flow has an analytical solution and so it can be used to demonstrate clearly the accuracy of the code. This section provides a variety of results based on both methods, which demonstrate that the numerical results are valid. The SIMPLE algorithm is selected for further work.

Considerations for applying the methods to higher Reynolds number flow (though not turbulent flow) are given in Section 5.4. In this case, it would be desirable to use a better discretisation scheme than central differencing. Assessment of the differencing schemes: central differencing, upwind differencing and QUICK are presented for the channel flow case. Results are presented for Reynolds numbers up to 1000. The QUICK scheme is selected for further calculations.

5.2 Boundary Conditions

In solving the hydrodynamic equations inside a bounded region, when the flow within the domain is variable in time and space, the setting of boundary conditions presents an important problem. The reason is that the flow field is determined by the conditions imposed upon it at its boundaries, and usually the stability and accuracy of the discrete conditions affect the stability and accuracy of the difference scheme itself.

It means in general that a correct boundary condition for numerical computations is quite important.

The physical boundary condition describes the practical fluid flow environment, but in numerous realistic flow problems, the properties on the boundary can not be described in advance. For this reason computational boundary conditions are employed. The computational boundary conditions have to be specified properly so that solutions exist and are unique for each problem and so that the numerical results will approach the exact solutions. In this chapter, the boundary conditions which are necessary for calculating fluid flow are described.

The channel flow problem is used for validation for both the SIMPLE and the MAC algorithms. Two sets of boundary conditions are used. One gives velocities and pressure at all boundaries. These are Dirichlet boundary conditions referred to in the following as boundary condition set (1) and shown in Figure 5.2. The other uses velocity and pressure together with derivatives, which is a combination of Dirichlet and Neumann boundary conditions referred to in the following as boundary condition set (2) and shown in Figure 5.3. The equations used in these two sets of conditions are obtained from the analytical solution.

5.2.1 The analytical solution for fully developed channel flow

The analytical solution for the channel flow can be derived as follows. Consider liquid forced under pressure to move between two fixed parallel lines a distance $2b$ apart, (see Figure 5.2) and take one line to be the x-axis and the other to be at $y = 2b$.

The Navier-Stokes momentum equations and continuity equation have been given in Chapter 4. Suppose, first, that the motion is in the x-direction only, so that if

$$\bar{\phi} = u\bar{i} + v\bar{j},$$

then $v = 0$. Then the equation of continuity is $\partial u / \partial x = 0$, which means that u is independent of x. When the motion is steady, u will therefore be a function of y only. Thus the momentum equation for u becomes

$$-\frac{\partial p}{\partial x} + \mu \frac{\partial^2 u}{\partial y^2} = 0,$$

and for y becomes $-\partial p / \partial y = 0$. Integrating the equation and putting $-\partial p / \partial x = P$, gives

$$u = A + By - \frac{P}{2\mu} y^2 .$$

Since $u = 0$ when $y = 0$ or $y = 2b$, we can work out A and B, which leads to

$$u = \frac{1}{2\mu} y(2b - y)P \quad . \tag{5-2-1}$$

When $y=b$, the maximum velocity $u_{max} = \frac{b^2}{2\mu} P$. Then the pressure gradient has the following form:

$$P = -\frac{\partial p}{\partial x} = u_{max} \frac{2\mu}{b^2} \tag{5-2-2}$$

and (5-2-1) becomes

$$u = u_{max}[1 - (\frac{y-b}{b})^2]. \tag{5-2-3}$$

Equation (5-2-3) is used to define the inlet flow boundary velocities that have a parabolic profile lying along the west side of the domain.

The inlet v-velocity component is zero,

$$v = 0 \ . \tag{5-2-4}$$

The outlet lies at the east side of the domain, where the outlet velocities have analytical values in boundary condition set (1) and the velocity gradients are zero in boundary condition set (2).

For the pressure value, integration of equation (5-2-2) gives

$$p = -\frac{2\mu u_{\max}}{b^2} x + c \, , \tag{5-2-5}$$

where c is a constant, and upon assuming $p = 0$ when $x = L$, we get $c = u_{\max} \frac{2\mu}{b^2} L$.

5.2.2 The definition of boundary pressure with boundary condition set (2)

In numerous realistic flow problems, the pressure is not prescribed on the boundaries. In fact, the absolute values of the pressure solution do not usually bear any significance, and the pressure gradient is more important. To obtain boundary conditions for the pressure, we may use boundary condition set (2) which treats some boundary pressure values as unknowns. In the following an explanation of the computational pressure boundary condition will be given.

To obtain pressure values at the boundary nodes, extrapolation from the interior nodes is used. From Figure 5.4, the Taylor expansion was employed at point 2 to obtain the value at point 1 on the inlet boundary

$$p_1 = p_2 + h(\frac{\partial p}{\partial \xi})_2 + O(h^2)$$

where ξ is defined by the increment with x direction and using $(\frac{\partial p}{\partial \xi})_2 = \frac{p_1 - p_3}{2h}$, the result can be then written as

$$p_1 = 2p_2 - p_3, \qquad (5\text{-}2\text{-}6)$$

In this way, the truncation error at the inlet boundary is uniformly of second order.

Similarly, the extrapolation method can be applied on the parallel wall to satisfy the Neumann pressure boundary condition:

$$\frac{\partial p}{\partial n} = 0.$$

We have: (see Figure 5.5)

$$p_1 = p_2, \qquad (5\text{-}2\text{-}7)$$

where p_2 is the pressure on the interior point and p_1 is the pressure on the wall boundary point. The equation is of first order accuracy.

The outlet boundary pressure for both boundary condition sets (1) and (2) are defined as below: (see Figure 5.2 and Figure 5.3)

$$p_{fj} = 0, \qquad (5\text{-}2\text{-}8)$$

where *fj* is the face number *j* of the boundary element (volume) at the outlet. For calculating the incompressible flow field, the pressure differences between points in the flow field are more significant than the absolute pressure values themselves. The absolute pressure values are usually much higher than the calculated pressure differences in the flow field. If the calculation is based on the absolute pressure a large relative error would be generated in the calculation of the pressure difference. In order to reduce the rounding error in the calculation of pressure correction values,

it is usual to prescribe the absolute pressure values as zero on some points and the pressure on all other points are referenced to these points. For this reason the outlet boundary pressure values are set to zero.

5.2.3 Calculation of the velocity gradient on the boundary

In the momentum equations, it is necessary to calculate the velocity gradient values in the diffusion terms [see equation (4-12)]. Inside the fluid domain, the velocity gradient values are calculated by the discretised equation, but the velocity gradient at the boundary needs special treatment. Four alternative methods for calculating the velocity gradient at the boundary were considered and are summarised in Table 5.1, in which $(\frac{\partial \phi}{\partial x})_f$, $(\frac{\partial \phi}{\partial y})_f$, ϕ_f and y_f are values at the face of an element lying at the boundary, ϕ_i is the value at the centre of element and ϕ_2, ϕ_3 are interior point values. (see Figure 5.6) The four cases are defined as follows:

Case (1). First order accurate extrapolation at all boundaries without consideration of the angle between the element centre and the centre of the element face lying on boundary.

Case (2). Inlet/outlet boundaries use first order accurate extrapolation with consideration of the angle between the element centre and the centre of the element face lying on boundary, Parallel walls use first order accurate extrapolation as case (1).

Case (3). Inlet/outlet boundaries use first order accurate extrapolation as case (1) and the parallel wall uses second order accurate extrapolation.

Case (4). Inlet / outlet boundaries use first order accurate extrapolation as case (2) and the parallel wall uses second order accurate extrapolation as case (3).

The velocity gradient change in the *y*-direction is much greater than that in the *x*-direction. For this reason, second order extrapolation is applied for the parallel wall and first order extrapolation is sufficient for the inlet and outlet.

These four cases with boundary condition set (1) are calculated using the SIMPLE method with the 24×24 grid (a total of 24×24×2 =1152 triangular elements). The results are summarised in Table 5.2, where the accuracy describes the error (*u*, *v* and *p*) between the analytical solution and the numerical results summed over the entire grid and the convergence error indicates the difference in the sum of all variables between two iterative steps.

The results show that case (4) requires the least CPU time and achieves greater accuracy. Thus case (4) will be used for all following calculations.

5.3 Results and Discussion of Low Reynolds Number Simulations

In this section, the objective is to validate the method and the code, and to search for the best scheme for calculating the flow field. Various tests are presented and the best method among them selected at each stage.

The following conditions are used in every test:

The length of domain L is defined to be 3.0×10^{-3}m and the width of domain $2b$ is also defined to be 3.0×10^{-3}m. The initial values of velocities and pressure within the domain are set to be zero, the Reynolds number is 1.5 (defined below). The central difference scheme is used for the convection terms in the momentum equations and stability is ensured as the grids are uniform and the Peclet number is less than 2 (see 4.2.2.1). In each case the calculation is considered to have converged when the sum of normalised residuals falls below 1.0×10^{-6}. The accuracy, CPU and grid convergence are used to assess the quality of the solution for all cases investigated.

5.3.1 The Reynolds number *Re*

The Reynolds number uniquely describes the flow and is given by

$$Re = \frac{u_{max} d}{\nu}, \tag{5-3-1}$$

where d is the characteristic length and $d=2b$ for channel flow. u_{max} is the maximum inflow velocity; $\nu = \frac{\mu}{\rho}$ is the kinematic viscosity of the fluid.

For the following simulations, the Reynolds number is taken as, $Re=1.5$ which corresponds to $d = 3.0\times10^{-3}$ m, $u_{max} = 8.0\times10^{-2}$ m/s, $\nu = 1.6\times10^{-4}$ m^2/s.

5.3.2 Results and discussion for the MAC algorithm

The procedure in the MAC method is similar to that in the SIMPLE method. (see Chapter 4) The same momentum equations are used in both the MAC and the SIMPLE methods. The difference between the MAC and the SIMPLE methods is the technique used to solve the pressure value. The former uses the Poisson Equation for Pressure (PEP) while the latter uses the pressure correction equation.

5.3.2.1 *Optimisation of the solution process*

In order to optimise the solution process, two alternative schemes are investigated in which (PEP) is solved either once or iteratively within each outer iteration using equation (4-83). For both cases boundary condition set (1) is used on the 24×24 grid. The results are summarised in Table 5.3. The results show that the second scheme requires less CPU time and achieves greater accuracy. Thus in all following cases the PEP is solved iteratively within each outer iteration.

5.3.2.2 *Optimisation of the right hand side of the PEP*

Two cases are considered in order to compare equations (4-79) and (4-83) which have different right hand sides. In both cases boundary condition set (1) is used on the 24×24 grid. The results are summarised in Table 5.4. The results show that the PEP of (4-83) requires less CPU time and achieves greater accuracy. Thus in all following cases, equation (4-83) is used for solving the Poisson equation.

5.3.2.3 *Grid convergence for the MAC scheme*

In order to consider grid convergence, four calculations were performed on grids of 6×6, 12×12, 24×24 and 48×48. The boundary condition set (1) is used and the results are summarised in Table 5.5. The results show that the error converges from a 6×6 grid to a 24×24 grid, but the error is larger for a 48×48 grid. The reason is that the gradients of the flow variables at the boundaries do not yield exact values. For example,

$$(\frac{\partial \phi}{\partial n})_f = \frac{\phi_i - \phi_f}{\Delta \xi} + O(\Delta \xi).$$

Here ϕ_i is the value at the centre of the element and is a guessed value when used in numerical calculation procedures. ϕ_f is the value at the face of the element lying on the boundary and can have either an analytical value or a computational boundary value. Therefore, $\phi_i - \phi_f$ has a finite non-zero value in general and is independent of grid size. With an increasingly refined grid, such that $\Delta \xi$ is getting smaller, this implies that $(\frac{\partial \phi}{\partial n})_f$ will get increasingly large. So perhaps, for this reason, the error is larger for the 48×48 grid.

5.3.2.4 *Comparison of boundary condition set (1) and set (2)*

In comparing two boundary condition sets the 24×24 grid is used and the results are summarised in Table 5.6. The results show that the boundary condition set (1) requires less CPU time and achieves greater accuracy than boundary condition set (2). The reason is that the boundary condition set (1) uses the full analytical result and the boundary condition set (2) requires some computational results. The solution using boundary condition set (2) will approach the analytical solution when the grid is refined and will be used for calculations in general where an analytical solution is not available.

5.3.3 Results and discussion for the SIMPLE algorithm

Tests are carried out with the same objectives as above which are to select the best method and scheme with the SIMPLE algorithm at each stage.

5.3.3.1 *Grid convergence for the SIMPLE scheme*

In order to consider convergence with grid, the 6×6, 12×12, 24×24 and 48×48 grids are used with boundary condition set (1). The results are summarised in Table 5.7. The results show that the error decreases as the grid is refined. It can be seen that the total accuracy (u, v and p) is dominated by the accuracy for pressure, p. This indicates that the error of the pressure gradient is larger than the velocity gradient at the boundaries. For this reason, attention should be drawn to the pressure and pressure correction gradient at the boundaries. In contrast with the MAC algorithm, the SIMPLE algorithm uses the pressure correction equation. The pressure correction is not fixed at the boundaries and we have,

$$(\frac{\partial p'}{\partial n})_f = \frac{p'_i - p'_f}{\Delta \xi} + O(\Delta \xi)$$

where p'_i is the value at the centre of the element and is calculated in the pressure-correction equation. p'_f is the value at the face of the element lying on the boundary.

It is found by extrapolation of the interior nodes using equation (5-2-6), thus, $p'_i - p'_f$ is dependent on grid size. For this reason, the error reduces as the grid is refined when using the SIMPLE algorithm.

5.3.3.2 *Comparison of boundary condition set (1) and set (2)*

In comparing the two boundary condition sets, calculations are performed on the 24×24 grid. The results are summarised in Table 5.8. The results show that boundary condition set (1) requires less CPU time and achieves greater accuracy. However, in numerous realistic flow fields, boundary condition set (2) has usually to be employed.

5.3.4 Comparison of the MAC and the SIMPLE algorithms

Both the SIMPLE and MAC algorithms have been applied successfully to calculating channel flow. It is obvious that boundary conditions play a very important part in the numerical solution. Comparison and discussion of the MAC and the SIMPLE algorithms with boundary condition set (2) are given in the following:

5.3.4.1 *CPU time*

The conclusion is that the CPU time used by the SIMPLE method is a little higher than that used by the MAC method for various grids. The results are summarised in Table 5.9. If considering only the CPU time, the MAC algorithm is recommended.

5.3.4.2 *Accuracy*

The accuracy is divided into two parts: the velocity and the pressure.

(a) The velocity accuracy

The u-velocity error across the centre of the domain (x=1.5×10^{-3}m) in the vertical direction with the 24×24 grid is shown in Figure 5.7. The conclusion is that the

accuracy of the *u*-velocity with the SIMPLE algorithm is higher than the MAC algorithm.

(b) The pressure accuracy

The pressure error across the centre of domain (y=1.5×10^{-3}m) in the x-direction on 24×24 grid is shown in the following Figure 5.8. The conclusion is that the pressure accuracy of the MAC algorithm is higher than the SIMPLE algorithm.

(c) The total accuracy

The results of the accuracy with the 24×24 grid are summarised in Table 5.10. The conclusion is the total accuracy (u, v and p) of the MAC algorithm is a little higher than the SIMPLE algorithm. Thus the MAC algorithm is recommended.

5.3.4.3 *The grid convergence*

The conclusion of the tests above (see Section 5.3.2.3 and Section 5.3.3.1) is that grid convergence occurs on the SIMPLE algorithm but not with MAC. This is because of problems in calculation of the gradients of the flow variables at boundaries as discussed in Section 5.3.2.3. Thus the SIMPLE algorithm is recommended.

5.3.4.4 *The conclusion*

We can see that both the SIMPLE and MAC algorithms for calculation of the channel flow work well. In comparing the accuracy, CPU and grid convergence of the MAC and the SIMPLE algorithms, it is difficult to decide which one is best, because both algorithms have advantages and disadvantages. The selection should depend on the nature of the fluid flow, but for the channel flow case, conclusions are as follows:

The total accuracy of the MAC algorithm is a little higher than the SIMPLE algorithm, but it should be noted that the calculation of the pressure using equation (4-82) (that is part of PEP) gives

$$-\rho[-u_{ff}(v_{jend}-v_{jst})+v_{ff}(u_{jend}-u_{jst})]$$

The v-velocity component is zero in this two-dimensional channel flow. Therefore, $u_{ff}(v_{jend}-v_{jst})$ and $v_{ff}(u_{jend}-u_{jst})$ will be of very small values. So perhaps, for this reason the accuracy of the pressure with MAC is higher than that of SIMPLE with channel flow and may not be for other fluid flow calculations.

Although the CPU time of the SIMPLE algorithm is a little higher than the MAC algorithm, grid convergence occurs only with the SIMPLE algorithm. In addition, SIMPLE algorithm is more flexible than the MAC algorithm, under-relaxation can be more easily employed. And versteeg and malalasekera (1995) took the SIMPLE scheme into account for the robust consideration in two-dimensional laminar flow equations. Thus for all following calculations, the SIMPLE algorithm has been selected.

Comparison between numerical and analytical data for the u-velocity profile at the outlet is shown in Figure 5.9 with boundary condition set (2). Pressure contours are plotted in Figure 5.10, the difference between the maximum and minimum contour plotted, $\Delta p = 4.346\times10^{-3}$ Pa.

5.4 Results and Discussion of Higher Reynolds Number Simulations

In this section, results and discussion are given for the simulation of higher Reynolds number channel flow (though not turbulent flow).

5.4.1 Assessment of the differencing schemes

The central differencing scheme will be stable and accurate when $|P_c|<2$. This condition was satisfied for the low Reynolds number flow results in section 5.3. However, if P_c is greater than 2 some coefficients of the discretised equations will be negative space. In one of the examples shown in Table 5.11, P_c=4. The consequence is evident in the results which show divergence.

Owing to this limitation, central differencing is not a suitable discretisation practice for general purpose flow calculations. The upwind differencing scheme is stable when the Peclet number $|P_c|>2$. The first order upwind differencing scheme is tested and results are compared with those from central differencing, as summarised in Table 5.11.

Firstly, the results show that the accuracy of the central differencing scheme is higher than the accuracy of the first order upwind differencing scheme when $|P_c|<2$. This is because the Taylor series truncation error of the central differencing scheme is of second order and of the upwind differencing scheme is of first order.

Secondly, the central differencing scheme fails in the case $|P_c|>2$ and the first order upwind scheme produces a realistic solution that is stable, but has a large error. Therefore, the first order upwind differencing scheme is not suitable for accurate flow calculations and higher order discretisation should be employed. Higher order schemes involve more neighbour points and reduce the discretisation error by bringing in a wider influence. An example is the quadratic upwind differencing scheme (QUICK) (see Chapter 4). For the case $P_c = 4$ (for which central differencing would be unstable), the upwind and QUICK schemes are compared in Table 5.12. The results show that the accuracy of the QUICK scheme is much better than that of the first order upwind scheme. Thus the QUICK scheme will be used in further calculations.

5.4.2 *Re*=1000 flow simulation

For this case, flow variables are defined by $u_{max}=8.0\times10^{-2}$ m/s, $\nu=2.4\times10^{-7}$ m^2/s and $d=3.0\times10^{-3}$m. Thus

$$Re=\frac{u_{max}d}{\nu}=1000$$

The result is summarised in Table 5.13 and the pressure contours are shown in Figure 5.11, $\Delta p = 2.83\times10^{-3}$ Pa between the maximum and minimum values. The results show that the accuracy is acceptable and the pressure field is smooth.

5.5 Closure

A finite volume code for solving the Navier-Stokes equations on triangular grids has been successfully written and validated against the corresponding analytical solution for channel flow. Implementation of the collocated grid arrangement is successfully handled by using the averaged linear-interpolations for calculating the nodal values. Comparison between the SIMPLE and the MAC algorithms for channel flow has been given, as a result of which the SIMPLE algorithm is recommended.

	case 1	case 2	case 3	case 4
The velocity gradient for the inlet/outlet boundary	$(\frac{\partial\phi}{\partial x})_f = \frac{\phi_i - \phi_f}{\Delta x}$	$(\frac{\partial\phi}{\partial x})_f = \frac{\phi_i - \phi_f}{\Delta x} + \frac{2(y_f - b)}{b^2} \times \frac{(y_i - y_f)}{\Delta y}$	$(\frac{\partial\phi}{\partial x})_f = \frac{\phi_i - \phi_f}{\Delta x}$	$(\frac{\partial\phi}{\partial x})_f = \frac{\phi_i - \phi_f}{\Delta x} + \frac{2(y_f - b)}{b^2} \times \frac{(y_i - y_f)}{\Delta y}$
The velocity gradient for the parallel wall	$(\frac{\partial\phi}{\partial y})_f = \frac{\phi_f - \phi_i}{\Delta y}$	$(\frac{\partial\phi}{\partial y})_f = \frac{\phi_f - \phi_i}{\Delta y}$	$(\frac{\partial\phi}{\partial y})_f = (3\phi_f - 4\phi_2 + \phi_3)/2\Delta y$	$(\frac{\partial\phi}{\partial y})_f = (3\phi_f - 4\phi_2 + \phi_3)/2\Delta y$

Table 5.1 The velocity gradient at boundaries

	CPU	Accuracy	Convergence error
case (1)	13 mins. 46.39 s.	1.535E-003	9.812E-007
case (2)	14 mins. 52.21 s.	1.738E-003	9.931E-007
case (3)	13 mins. 54.84 s.	6.139E-004	9.940E-007
case (4)	13 mins. 53.33 s.	5.457E-004	9.974E-007

Table 5.2 Optimisation of the velocity gradient calculation

	CPU	Accuracy	Convergence error
PEP is solved once	28 mins.55.06 s.	4.601E-004	9.981E-007
PEP is solved iteratively	16mins. 47.92 s.	1.307E-004	9.939E-007

Table 5.3 Optimisation of the solution process

	CPU	Accuracy	Convergence error
PEP of (4-79)	32mins. 2 s.	2.217E-004	9.951E-007
PEP of (4-83)	16 mins. 47.92 s.	1.307E-004	9.939E-007

Table 5.4 Optimisation of the right hand side of PEP

	CPU	Accuracy	Convergence error
6×6	7 s.	2.760E-003	9.414E-007
12×12	1 min. 18 s.	7.348E-004	9.924E-007
24×24	16 mins.47.92 s.	1.307E-004	9.939E-007
48×48	5 hrs.39 mins.0.22 s.	5.178E-004	9.984E-007

Table 5.5 Grid convergence

	CPU	Accuracy	Convergence error
boundary condition set (1)	16mins.47.92 s.	1.307E-004	9.939E-007
boundary condition set (2)	30mins.34.71 s.	6.149E-004	9.976E-007

Table5.6 Comparison of boundary condition sets

	CPU	Total accuracy	Accuracy for pressure	Total convergence error
6×6	16 s.	6.485E-003	3.346E-003	9.727E-007
12×12	1 min. 49 s.	1.858E-003	1.045E-003	9.887E-007
24×24	13 mins. 53.33 s.	5.457E-004	3.391E-004	9.974E-007
48×48	3 hrs.6 mins.46.95 s.	1.965E-004	1.417E-004	9.983E-007

Table 5.7 Grid convergence

	CPU	Accuracy	Convergence error
boundary condition set (1)	13mins. 53.33 s.	5.457E-004	9.974E-007
boundary condition set (2)	35mins. 46.86s.	6.817E-004	9.948E-007

Table 5.8 Comparison of boundary conditions

CPU	6×6	12×12	24×24	48×48
SIMPLE	16 s.	3 mins. 6 s.	35mins.46.86 s.	11 hrs.38 mins. 13.49 s.
MAC	12 s.	2 min. 25 s.	30mins.34.71 s.	10 hrs.39 mins. 18.85 s.

Table 5.9 Comparison of the MAC and the SIMPLE algorithms

	Accuracy for *u*-velocity	Accuracy for *v*-velocity	Accuracy for pressure	The total accuracy
MAC	2.237E-004	2.645E-004	1.266E-004	6.148E-004
SIMPLE	1.408E-004	1.899E-005	5.226E-004	6.817E-004

Table 5.10 Comparison of both algorithms

	Accuracy	Convergence
The central differencing $P_c < 2$ (Re=1.5)	6.817E-004	9.948E-007
The first order upwind $P_c < 2$ (Re=1.5)	4.513E-003	9.980E-007
The central differencing $P_c = 4$ (Re=96)	divergence	
The first order upwind $P_c = 4$ (Re=96)	2.561	9.927E-007

Table 5.11 Comparison of differencing schemes (24×24)

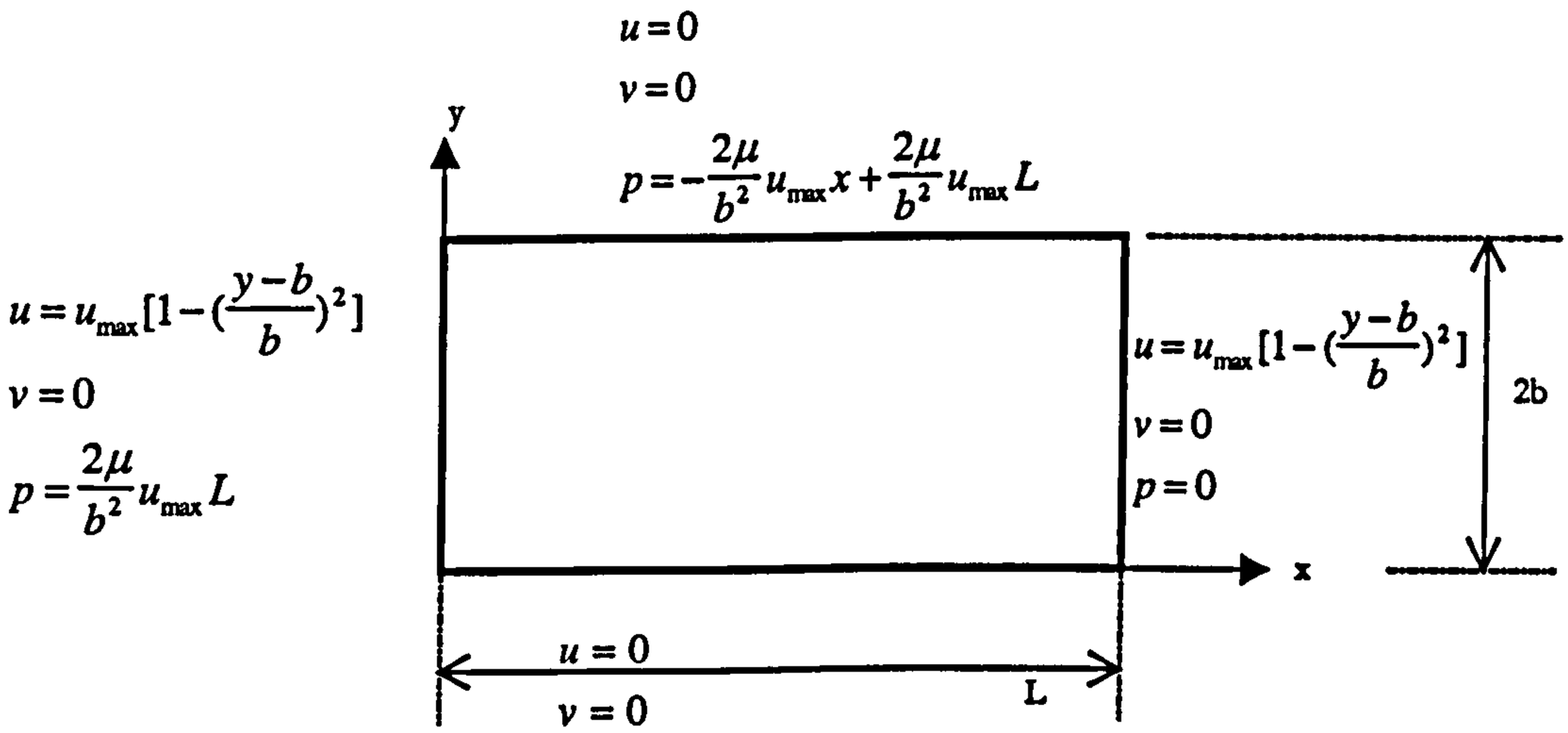
	CPU	Accuracy	Convergence error
The first order upwind P_c=4 (*Re*=96)	1 hrs. 17 mins. 30.39s.	2.561	9.927E-007
The QUICK P_c=4 (*Re*=96)	3 hrs. 1 min. 43.14s.	2.300E-003	9.993E-007

Table5.12 Comparison of differencing schemes (24×24)

	CPU	Accuracy	Convergence error
QUICK (*Re*=1000)	6 hrs. 12mins. 20.55s.	1.555E-002	9.99E-007

Table 5.13 The results of Re=1000 Simulation

Figure 5.1 Simple triangular grid generation



$$p = -\frac{2\mu}{b^2}u_{max}x + \frac{2\mu}{b^2}u_{max}L$$

Figure 5.2 Boundary condition set (1)

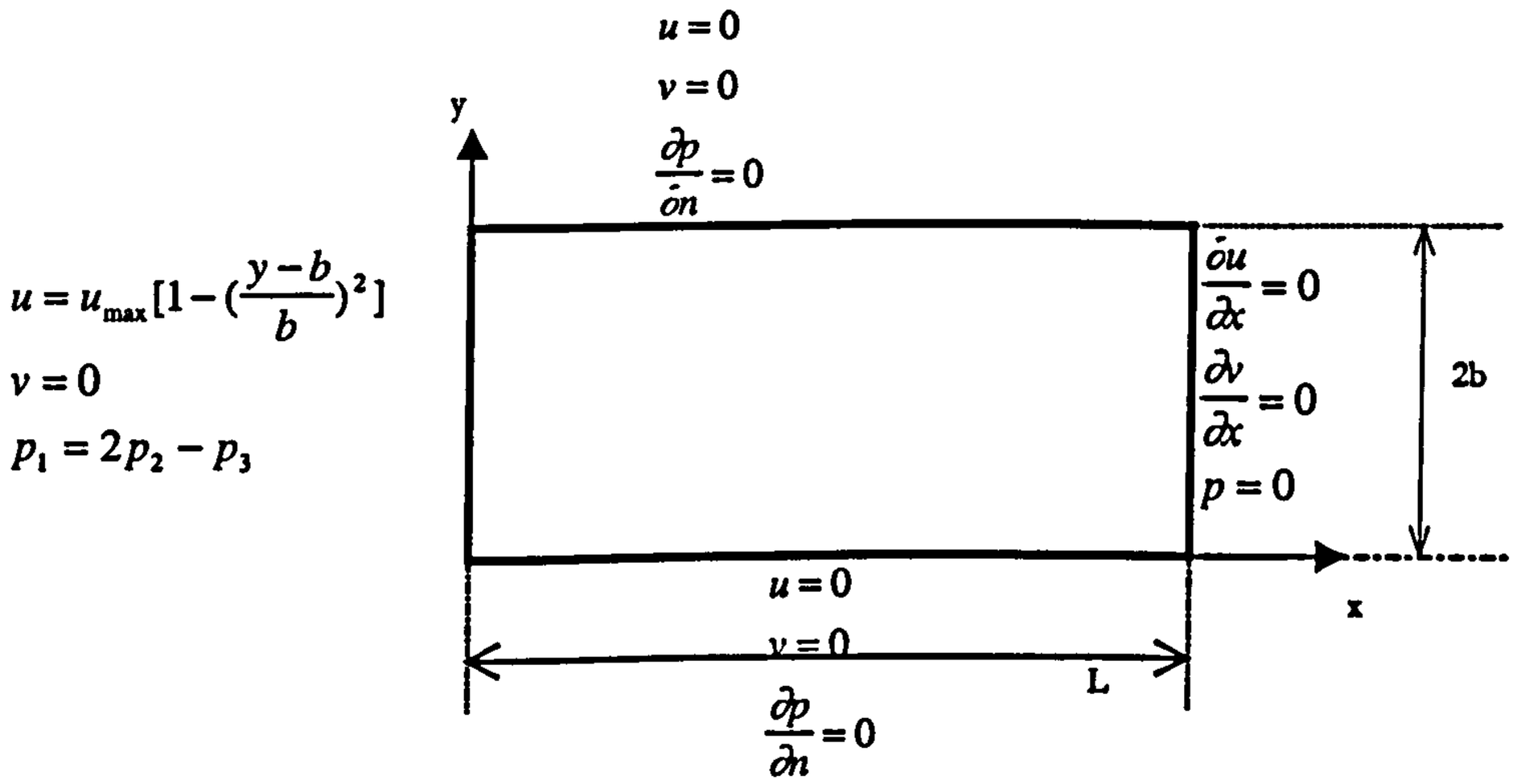


Figure 5.3 Boundary condition set (2)

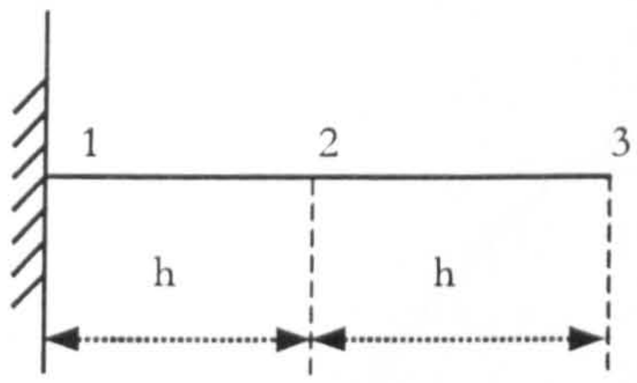
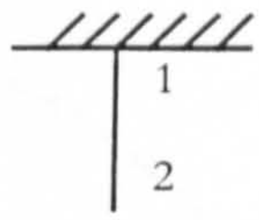


Figure 5.4 Notation associated with equation (5-2-6)



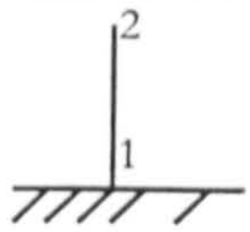


Figure 5.5 Notation associated with equation (5-2-7)

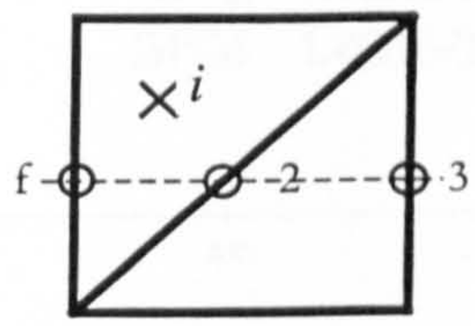


Figure 5.6 Notation associated with the four cases

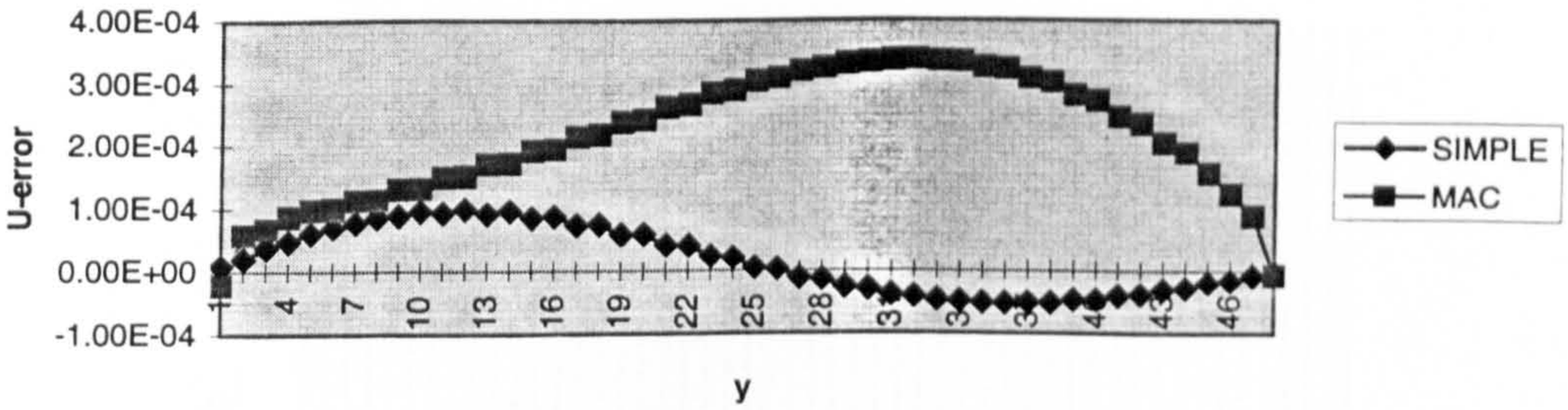


Figure 5.7 Comparison of u-velocity error at $x=1.5\times10^{-3}m$ with MAC and SIMPLE algorithms

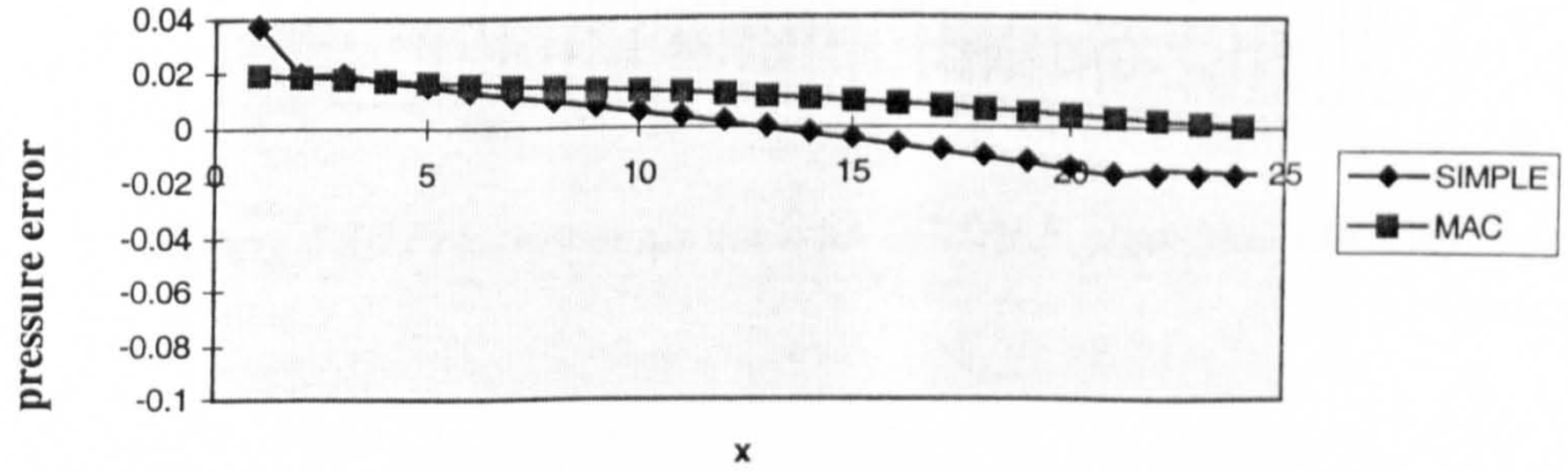


Figure 5.8 Comparison of MAC and SIMPLE algorithms

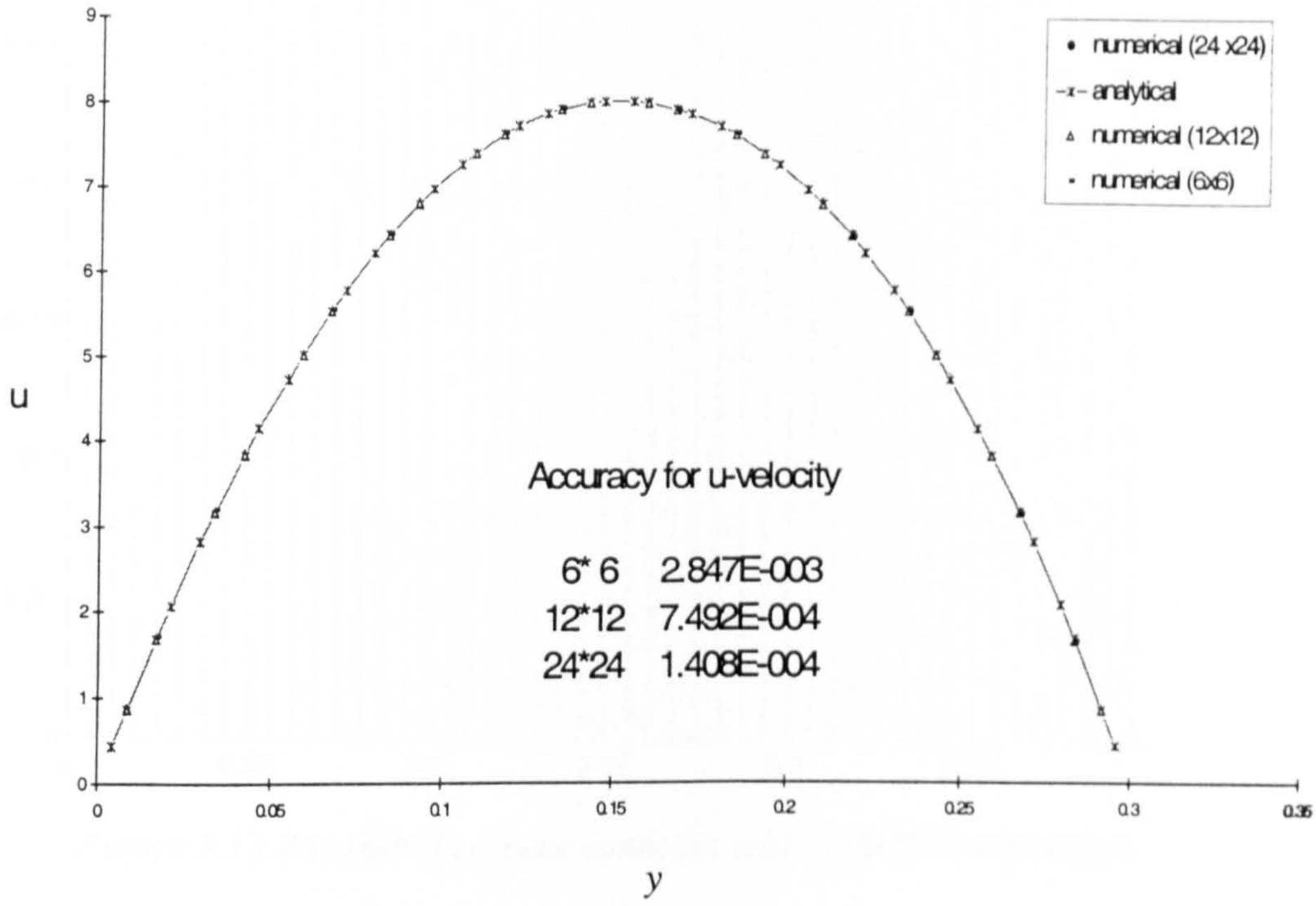


Figure 5.9 u-velocity profile at outlet (6×6, 12×12 and 24×24) with SIMPLE algorithm

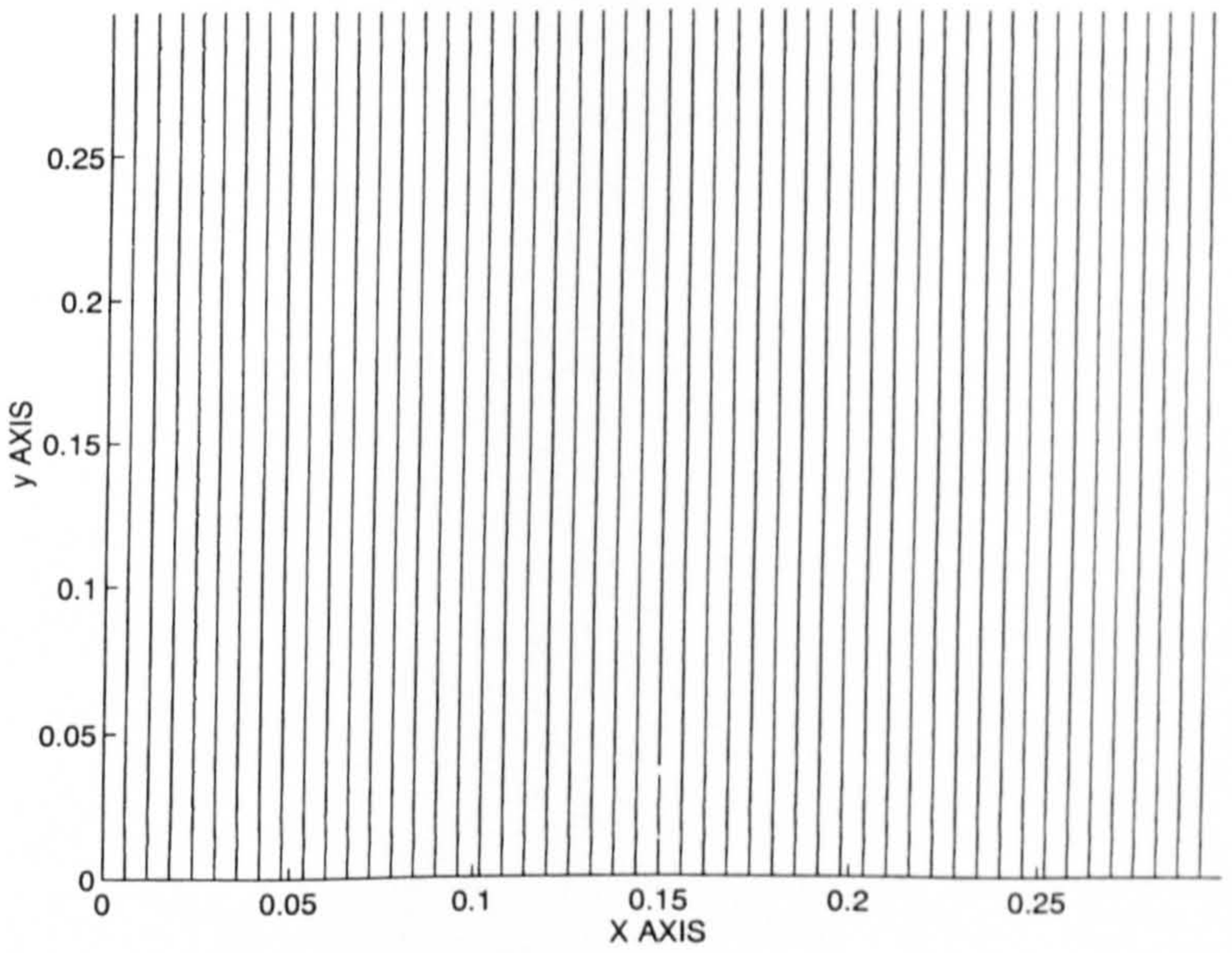


Figure 5.10 Pressure contours with SIMPLE algorithm (24×24)

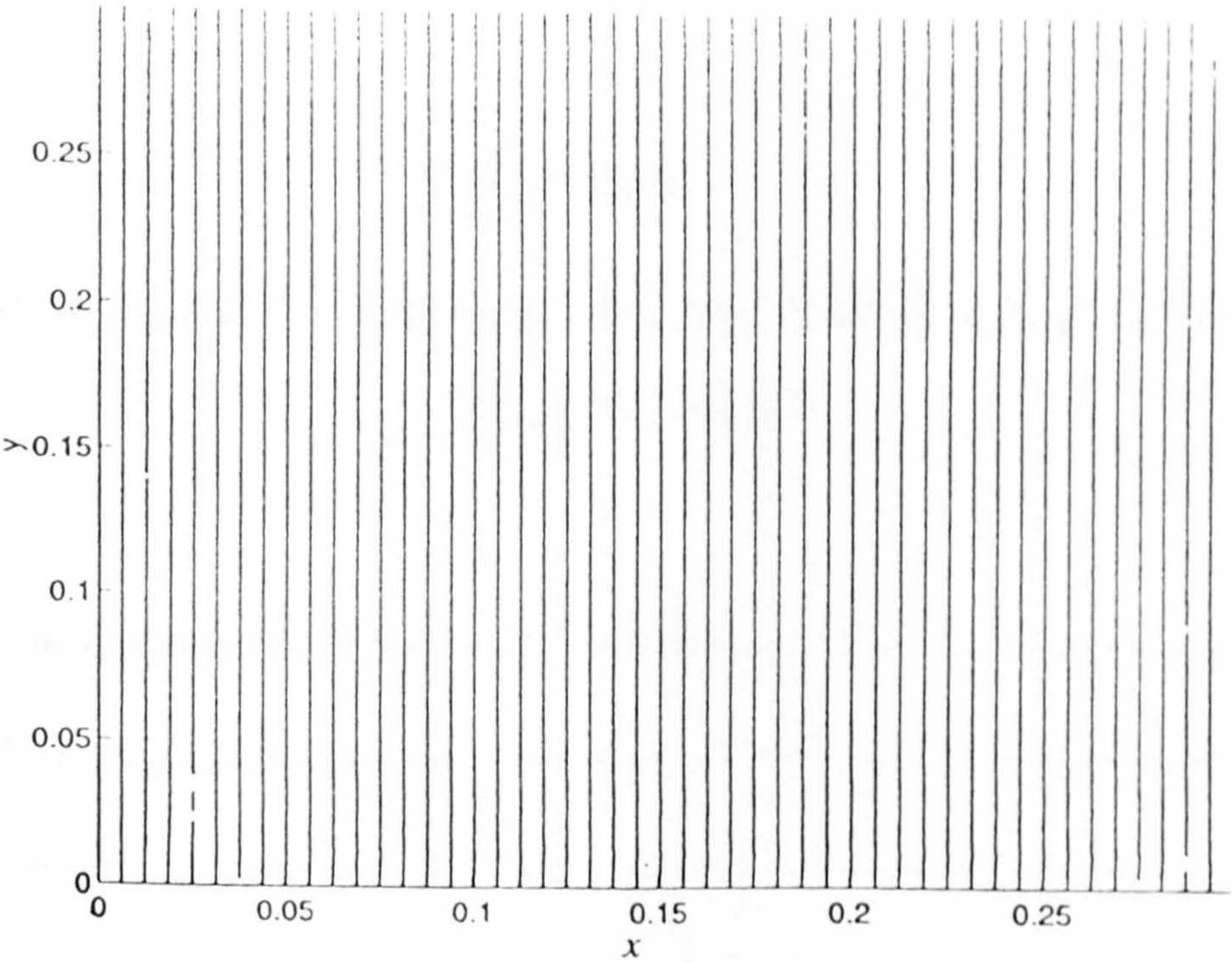


Figure 5.11 Re=1000 Pressure contours with SIMPLE algorithm

CHAPTER 6

RESULTS FOR LAMINAR FLOW OVER A BACKWARD-FACING STEP

Flow separation is a fundamental phenomenon in fluid flows. The problem involving steady viscous incompressible flow over a two-dimensional backward-facing step is a standard test problem and widely used for benchmark validation of CFD methods. At the step a recirculation is formed and the reattachment length in this laminar flow is a sensitive parameter that has historically been used to assess the overall predictive capability of CFD codes.

Previous related work on laminar separated flows has been carried out by several workers. Detailed experimental data are available from Denham and Patrick (1974). Also Borthwick and Kaar (1993), Atkins *et al.* (1980), Hackman *et al.* (1984), Barton (1994) and Chan and Anastasion (1999) have presented numerical calculations for this configuration.

6.1 The Geometry and Boundary Conditions

Figure 6.1 illustrates an initial equilateral triangle containing the set of seeding points, which define the boundary of a backward-facing step. The tri-tree grids are generated by recursive subdivision and for level 7 and 8 are shown in Figure 6.2 and 6.3.

A schematic diagram of the geometry for the backward-facing step is shown in Figure 6.4, which is characterised by separation at the step followed by a recirculation zone. A measure of computational accuracy is given by the prediction of the reattachment point, defined by a velocity sign change. The downstream channel was defined to have height $3h$ with a step height, h=0.9525 m and upstream inlet region set equal to $2h$. The upstream channel length is set to L_u=1.15h and the downstream channel length is taken as L_d=16h. The co-ordinate system for describing locations in the channel is at the step corner with the x-co-ordinate being defined as positive in the downstream direction and the y-co-ordinate across the channel.

The boundary conditions for the step geometry include the usual non-slip velocity specification for all solid surfaces except the outlet boundary as shown in Figure 6.5.

The velocity gradients and pressure of the outlet boundary are all defined to be zero. The inlet velocity field is specified as a parallel flow with a parabolic horizontal component given by

$$u(y) = u_{max}\, y(2h - y)\,. \qquad (0 \le y \le 2h) \qquad (6\text{-}1)$$

The Reynolds number is defined by $Re = u_{avg} \times h/\nu$, where u_{avg} is the average inlet velocity. Results are presented for inlet Reynolds numbers 73, 125, 191 and 229, corresponding to the experiments and the numerical calculations published. The calculation is considered to have converged when the sum of normalised residuals falls below 1.0×10^{-4}.

6.2 Results

The tri-tree grids used in the investigation to validate the laminar fluid flow over a backward facing step are listed in Table 6.1. For case A, the grid shown in Figure 6.2 has been used in the calculation with the Reynolds numbers 73 and 125. For case B, the grid shown in Figure 6.3 has been used in the calculation with the Reynolds numbers 191 and 229.

Calculation of the stream function and of streamlines: For calculation of the streamlines in laminar flow, it is convenient to represent the stream function ψ by its integration from a known value at a boundary using

$$\psi = \int u dy - \int v dx \qquad (6\text{-}2)$$

and the relationship

$$u = \frac{\partial \psi}{\partial y} \text{ and } v = -\frac{\partial \psi}{\partial x}. \qquad (6\text{-}3)$$

For this work, the stream function is calculated by first defining its value at the lower wall of the downstream channel to be equal to zero, and then calculating ψ at each element in the domain. At each element, the integration is based on cell-centre velocity components. The calculation is therefore repeated iteratively until convergence of the error between update iteration and previous one has fallen below 1.0×10^{-5}.

After the stream function has been calculated on each cell-centre, the nodal values of stream function are evaluated by interpolation using equation (3-9). The streamlines are then plotted as contour lines through each triangle in turn.

Calculation of the vorticity: The flow structure can be observed in the presence of vorticity, and understood by studying the evolution of the vorticity field. The vorticity is calculated using equation (3-8) at cell-centre for each element. After that, all procedures for plotting vorticity contours are similar to those for streamlines.

Velocity vectors: The visualisation of velocity, **u**, is achieved using components u and v. The process is implemented by MATLAB™ software, which from velocity u and v at element centres and the x y co-ordinates of these automatically produces a graphic depiction of the velocity field.

Numerical solution is carried out with Re=73 and the results in Figure 6.6. (a) show the stream function contours, where streamlines are at equal intervals of 3.83×10^{-2} m^2/s, (b) gives the vorticity contours at $\Delta\omega$ =0.39 /s intervals, and (c) and (d) give the velocity vectors and the detail of velocity vectors at the recirculation region.. The flow pattern shows good agreement with experimental data by Denham and Patrick (1974) and the numerical simulation by Borthwick and Kaar (1993). As would be expected, the main flow pattern consists of a primary recirculating flow immediately behind the backward-facing step which shows a slowing down of fluid in this region.

Figure 6.7 gives similar plots for the flow at Re=229. (a) shows the stream function contours, (b) gives the vorticity contours, and (c) and (d) give the velocity vectors and

the detail of velocity vectors at the recirculation region. These flow patterns agree well with the numerical simulation by Borthwick and Kaar (1993) in Figure 6.8.

The reattachment length is used to compare the effectiveness of the numerical methods. Laminar flow over a backward-facing step is characterised by separation at the step followed by a reattachment point, defined by a velocity sign change. The dimensionless reattachment length $X_R^* = X_R / h$ is shown in Table 6.2 and the comparison with experimental and numerical data is illustrated in Figure 6.9. The present results show good agreement with experimental and numerical data by Denham and Patrick (1974), O'Leary and Mueller (1969) and Borthwick and Kaar (1993). Also, the predictions for the length of separation within experimental error band compare satisfactorily with those shown by Tan *et al.* (1999).

6.3 Closure

A finite volume code for predicting recirculating flow with tri-tree grids has been successfully written and validated against published data. This case is used for comparison with published data as a basis to check the numerical solution procedure in a laminar flow field with recirculation.

Model and case	Cells	Nodes	The size of the element face
For *Re*=73 and 125 and case A	3926	2092	0.16
For *Re*=191 and 229 and case B	15484	7999	0.10

Table 6.1 Grid systems

Reynolds number	*Re*=73	*Re*=125	*Re*=191	*Re*=229
Dimensionless rettachment length X_R^*	3.52	6.24	8.06	8.89

Table 6.2 The numerical dimensionless reattachment length X_R^*

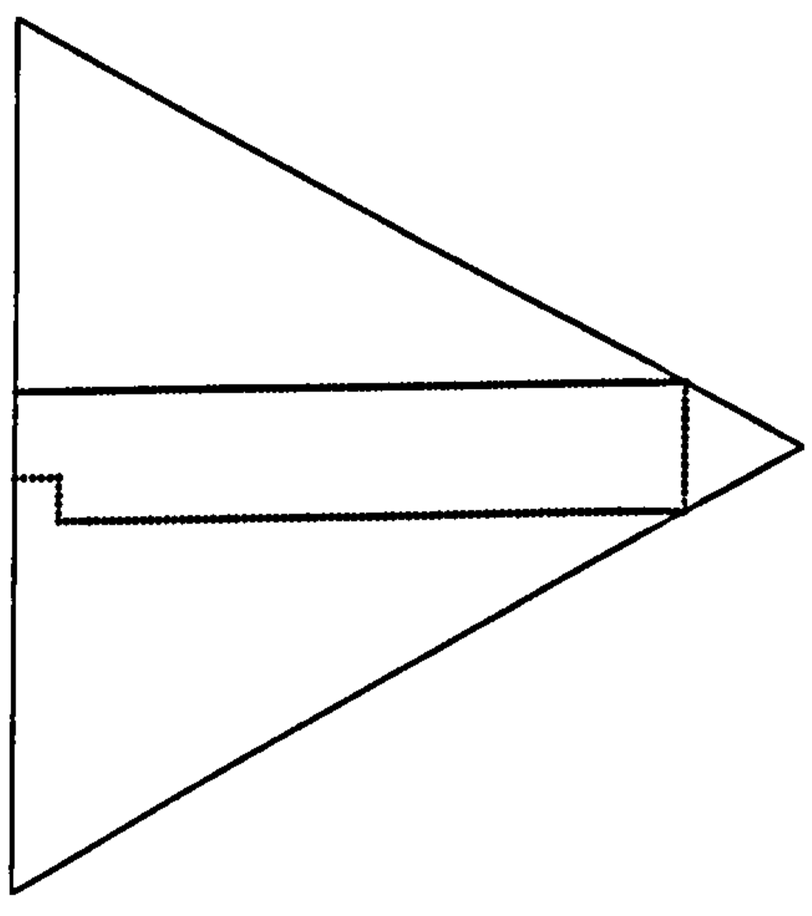


Figure 6.1 An initial triangle with seeding points

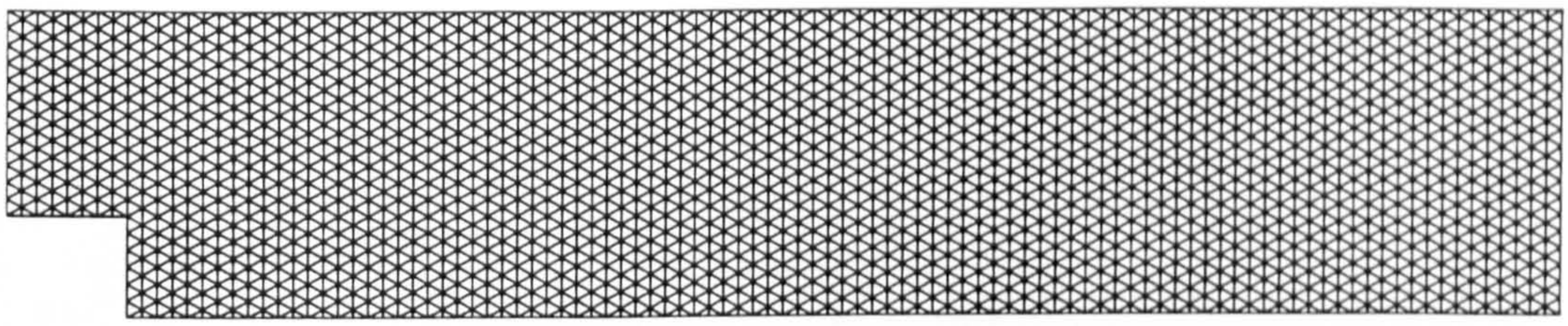


Figure 6.2 Tri-tree grids (level 7, elements = 3926 and nodes = 2092)

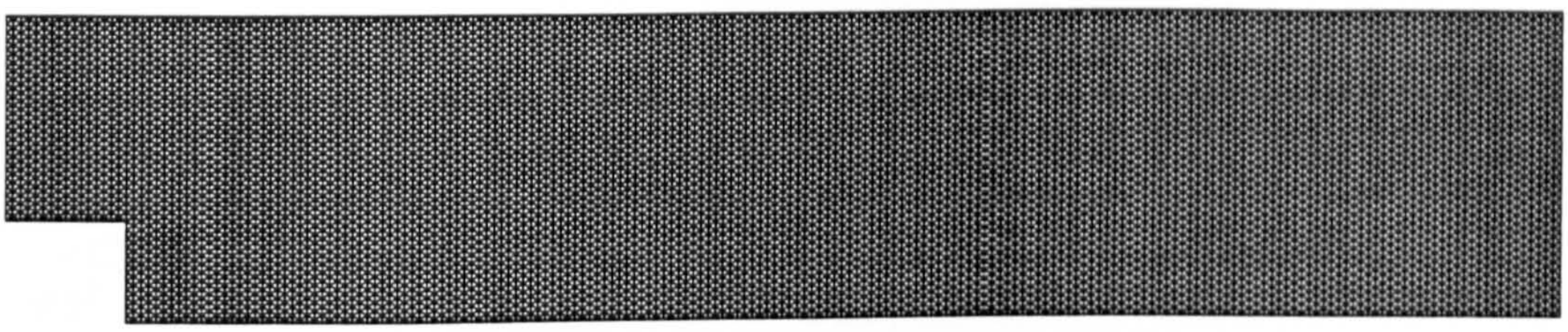


Figure 6.3 Tri-tree grids (level 8, elements = 15484 and nodes = 7999)

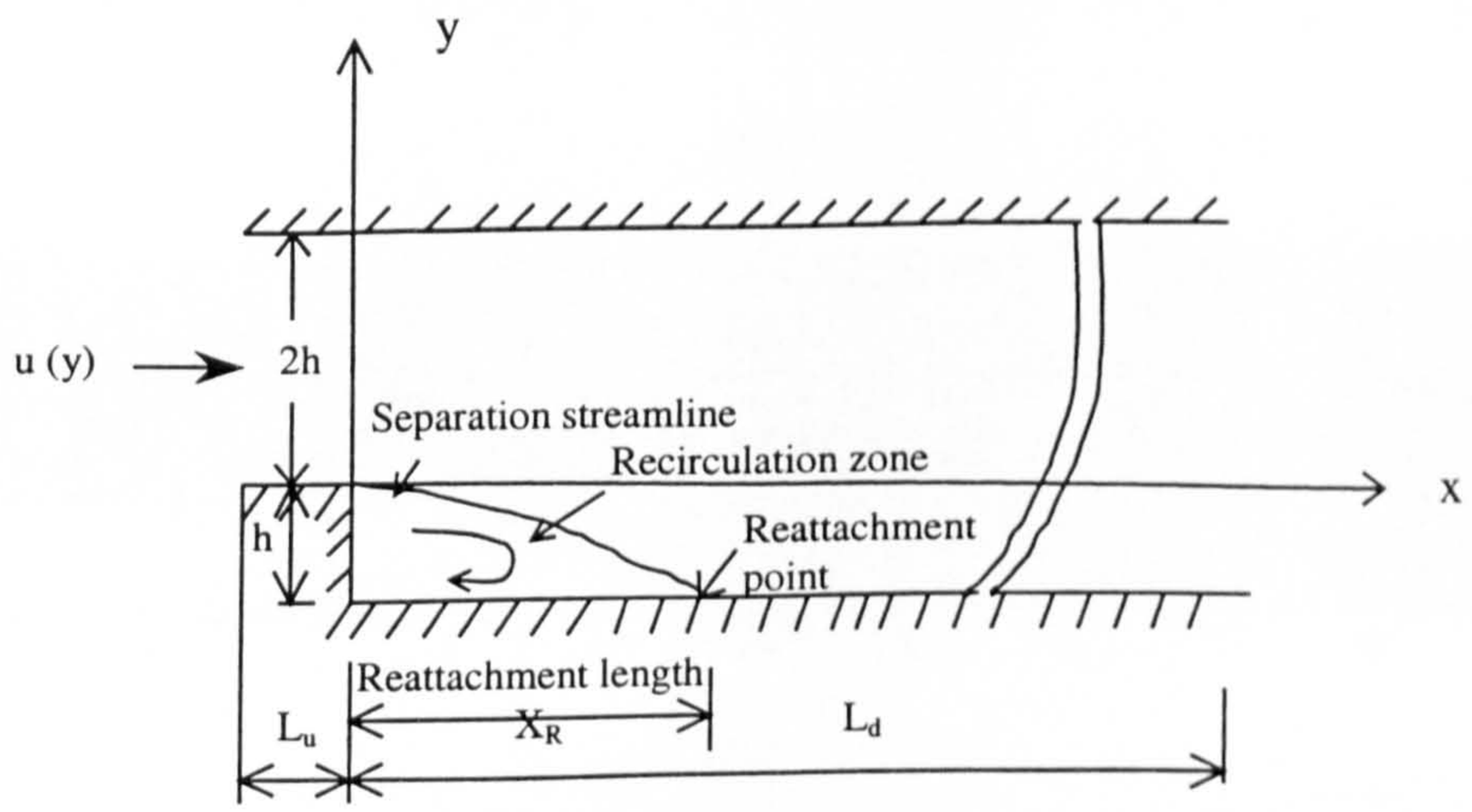


Figure 6.4 Physical configuration and co-ordinate system

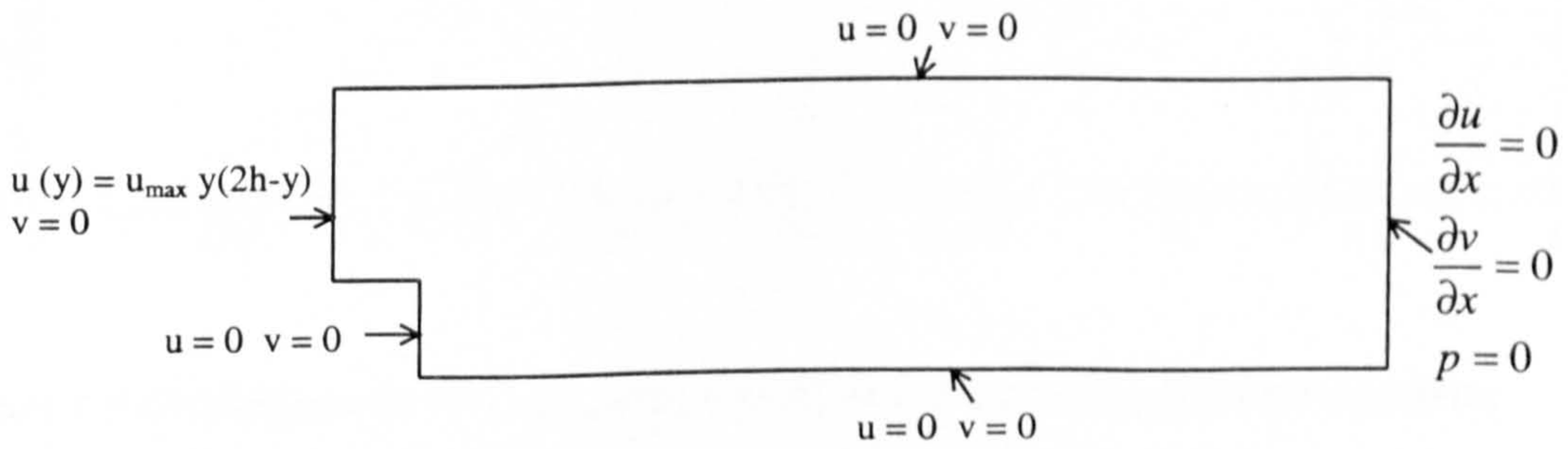


Figure 6.5 Boundary conditions

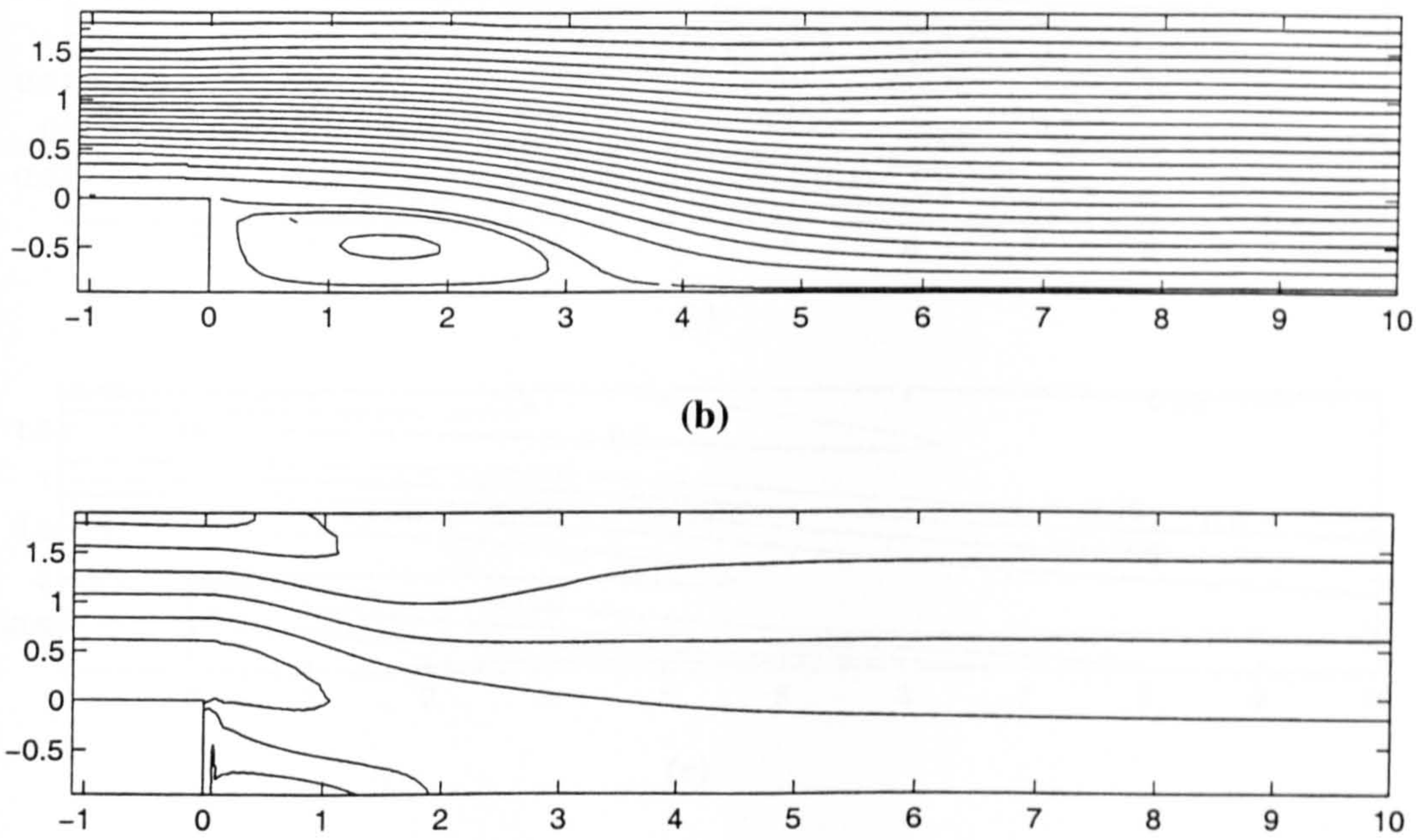


Figure 6.6 Present results for flow pattern at Re=73: (a) stream function contours; (b) vorticity contours; (c) velocity vectors; (d) detail of velocity vectors at the recirculation region.

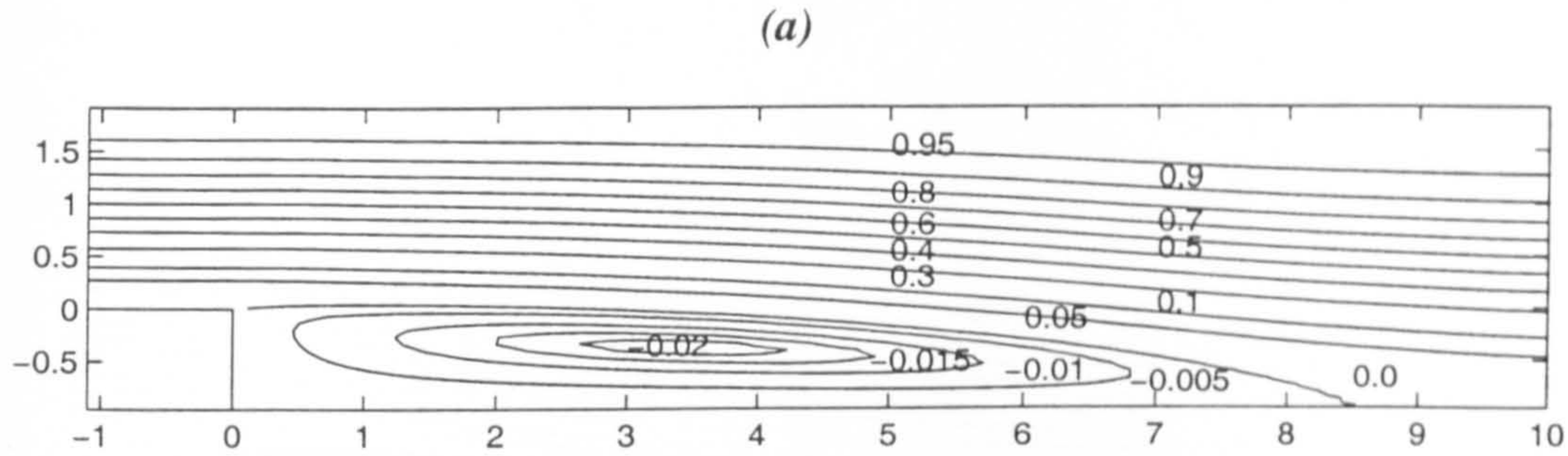


Figure 6.7 Present results for flow pattern at Re=229: (a) stream function contours; (b) vorticity contours; (c) velocity vectors; (d) detail of velocity vectors at the recirculation region.

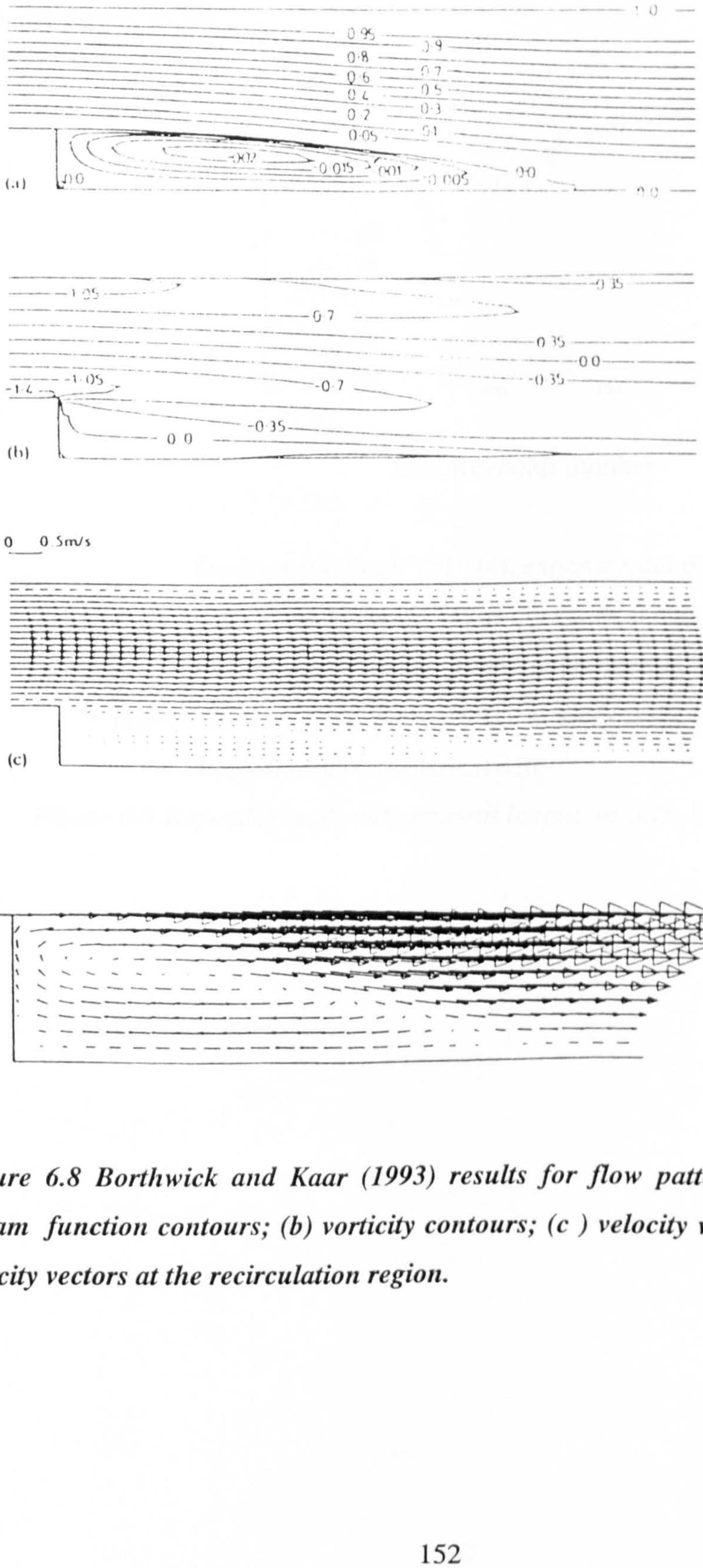
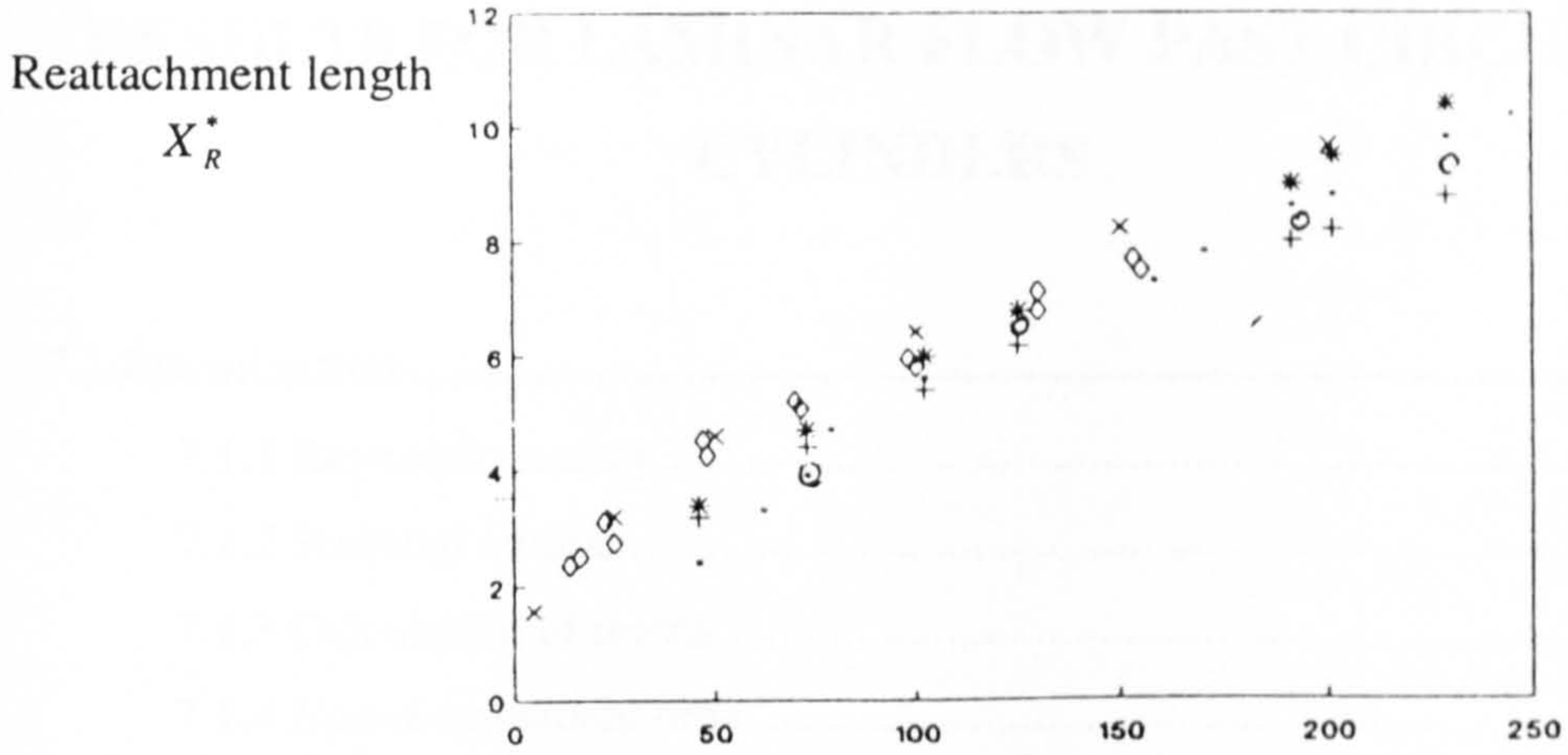


Figure 6.8 Borthwick and Kaar (1993) results for flow pattern at Re=229: (a) stream function contours; (b) vorticity contours; (c) velocity vectors; (d) detail of velocity vectors at the recirculation region.



· Denham and Patrick (1974), experimental data

× O'Leary and Mueller (1969), numerical data

◊ O'Leary and Mueller (1969), experimental data

* and + Borthwick and Kaar (1993), numerical data

∘ Present results, numerical data

Figure 6.9 Dependence of reattachment length on inlet Reynolds number

CHAPTER 7

RESULTS FOR LAMINAR FLOW PAST CIRCULAR CYLINDERS

7.1 Introduction

In this Chapter, results and discussions are presented concerning the simulation of incompressible flow past circular cylinders with adaptive tri-tree grids. The steady flow past a cylinder at Reynolds numbers equal to 2.04, 10 and 40 is predicted by ignoring time dependent terms in equation (4-21). Unsteady laminar flow past a cylinder at Reynolds numbers equal to 40 and 200, as well as flow past two and three cylinders at Reynolds number equal to 40 are simulated by equation (4-84).

These studies are aimed at obtaining approximate solutions that represent real flow systems as exactly as possible. The method behind these numerical simulations may essentially differ from experimental tests and other numerical simulations. Whichever method is used, the author agrees that the results should be similar to the experimental data that provides verification and guidance of visualised results and other numerical methods. Therefore, a comparison with previous numerical and experimental results is presented and some remarks are made to validate the algorithms proposed up to the present moment.

The great advantage of numerical simulation is the availability to show details on all aspects of the flow at every stage of development rather than at a few limited points available from experimental facilities. It is easy to obtain parameters such as the angle of separation and vortex centre, which are difficult with experimental studies. The purpose of the present study is not only to calculate some particular property of the flows, examples of which include Strouhal number, drag and lift coefficient (defined below), but also the development of periodic vortex shedding.

7.1.1 Reynolds number

The Reynolds number uniquely describes the flow past cylinders and is given by

$$Re = \frac{U_0 \times d_c}{\nu}, \tag{7-1}$$

where d_c is the diameter of the cylinder, U_0 is free stream velocity and ν is the kinematic viscosity of the fluid.

7.1.2 Strouhal number

The vortex shedding frequency, f, is related to the diameter of the cylinder, d_c, and the free stream velocity, U_0, by the dimensionless parameter

$$S = \frac{f \times d_c}{U_0}. \tag{7-2}$$

S is known as the Strouhal number.

7.1.3 Calculation of forces

Fluid flow exerts forces on a cylinder immersed which are due to the integrated fluid pressures acting perpendicular to the cylinder surface, and skin friction resulting from tangential viscous shear stresses. Let p and τ denote the pressure and shear stress on a surface element dA of the cylinder. The pdA and $\tau\, dA$ are, respectively, the pressure force and shear force acting on the surface element dA. Therefore, the total force is obtained through integration of the pressure and shear forces around the whole contour of the cylinder and is resolved into two components: the drag force, C_d, is the force component in x direction and lift force, C_l, represents y direction in a Cartesian co-ordinate as follows

$$C_d = \oint p n_x dA + \oint \tau n_x dA$$

$$C_l = \oint p n_y dA + \oint \tau n_y dA,$$

where (n_x, n_y) are the components of the unit vector of the cylinder face. The drag and lift forces, C_d and C_l are determined by the following formula (based on Acheson (1990)):

$$C_d = \oint_{cylinder} \{-p dx + \mu[2\frac{\partial u}{\partial x}dx + (\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x})dy]\} \tag{7-3}$$

$$C_l = \oint_{cylinder} \{-p dy + \mu[(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x})dx + 2\frac{\partial v}{\partial y}dy]\}, \tag{7-4}$$

where p denotes the pressure at the cylindrical surface and $\frac{\partial u}{\partial x}$, $\frac{\partial u}{\partial y}$, $\frac{\partial v}{\partial x}$ and $\frac{\partial v}{\partial y}$ define the velocity gradient at the cylindrical surface of x and y respectively.

It is usually the total drag and lift forces that are of interest, and they can be expressed in terms of a dimensionless drag coefficient, C_D, and a dimensionless lift coefficient, C_L, defined by the following equations:

$$C_D = \frac{C_d}{\frac{1}{2}\rho U_0^2 d_c}, \tag{7-5}$$

$$C_L = \frac{C_l}{\frac{1}{2}\rho U_0^2 d_c}, \tag{7-6}$$

where d_c is the diameter of the cylinder, U_0 is the free stream velocity and ρ is the density of the fluid.

7.1.4 Non-dimensional time t^*

For the time dependent flow simulations, the initial condition is zero velocity everywhere and the inlet velocity is constant throughout the simulation, $u = U_0$, $v = 0$. This is equivalent to the cylinder starting impulsively from rest in a stationary fluid. Once the flow has been started, the prediction is taken at different non-dimensional time values, t^*, which is defined as the ratio between the true time t and the parameter d_c / U_0 by

$$t^* = \frac{U_0 \times t}{d_c}. \tag{7-7}$$

7.1.5 Vortex shedding

As mentioned before, one of the objectives of simulation of flow past a circular cylinder is to predict the alternate vortex shedding that occurs behind the circular cylinder regarding the Reynolds number. Let us consider the flow development with respect to non-dimensional time, t^*, and the Reynolds number, *Re*.

If the value of *Re* remains below 4.4 or time t^* is less than a certain limiting value for *Re*>4.4, the flow develops without visible distortion, as described previously by Coutanceau and Bouard (1977). For *Re*>4.4 and after a period of time which is shorter as the Reynolds number becomes greater, separation of the flow appears giving rise to a recirculating zone that grows (in width and length) with time in the rear of the cylinder.

The first successful numerical solution of the complete equations of motion in two-dimensions was obtained by Thom (1928) for flow past a circular cylinder at *Re*=10. Later Thom (1933) improved his numerical method by a transformation of the physical plane and obtained a solution at *Re*=20 which agrees with available experimental results. Kawaguti and Jain (1966) reported numerical solutions of the time dependent equations for *Re*=10, 20, 30, 40 and 50. Dennis and Chang (1970) simulated steady flow past a circular cylinder at Reynolds number up to 100. Their observations are consistent with experimental results for *Re*>20, whereby 'standing eddies' or 'twin vortices' appear very clearly and the flow develops two symmetrical Föppl vortices behind the cylinder up to *Re*=40. Figure 7.1 shows that the geometrical parameters of the region can be measured: the length L and the separation angle θ_s, as well as the position of the vortex cores by the distances a and b: a represents the distance between the vortex cores and rear stagnation point, and b is the distance between the cores.

The estimated critical value of primary instability is Re=46.389 regardless of time t^* by Ding and Kawahara (1999). The value is very close to the observation data by Williamson (1989), Morzynski and Thiele (1991) and Jackson (1987). Bouard and Coutanceau (1980) pointed out that for Re=50 and Re=60 the flow is still stable up to t^*=3.5. But later, at a time t^* which is shorter as R_e is greater, the wake will become unstable.

In addition some experimental results by Coutanceau and Bouard (1977) for $5<Re\leq 40$ and Grove *et al.* (1963) for $Re<170$ recorded the main features for flow past a circular cylinder, which include length of the closed wake and the separation angle against R_e, the shape of the wake boundary, velocity and pressure distribution on the cylinder and the position of the vortex cores.

When the Reynolds number, Re, or time, t^*, continues to increase, the velocities in the recirculating zone increase rapidly and vortex shedding occurs in time. The attached vortices become asymmetric and are shed alternately at a well-defined frequency, the Strouhal frequency S, to form the Kàrmàn vortex street behind the cylinder. Many people using quite different numerical methods have studied numerical simulations of two-dimensional vortex shedding. For example, Jaime and Thomas (1969) used the finite difference method for the time dependent equation and simulated the flow past a circular cylinder at Re=200 and Re=500. Franke *et al.* (1990) applied the unsteady Navier-Stokes equations using a finite volume method for flows past a circular cylinder at Re=200 with structured grids. Chen *et al.* (1999) obtained flow patterns past a circular cylinder at Re=200 during the vortex shedding cycle using weighted essentially non-oscillatory (WENO) schemes with combined structured grids of O-H-grid system and O-grid system. Chan and Anastasion's (1999) results presented the evolution of vortex shedding in time past a circular cylinder at Re=200 with an unstructured triangular grid based on a cell-centred and second order upwind finite volume formulation. Min and Chang (1998) used the finite volume method on the fully staggered arrangement to simulate the evolution of vortex shedding in time past a circular cylinder at Re=1000 with structured grids. Zhang and Dalton (1998)

applied a fractional-step finite difference/spectral element method to predict Re=100 and Re=200 with two- and three-dimensional structured grids. Experimental work has also been published by Honji and Taneda (1969) and Taneda (1977), which produced experimental data and visualisation for Reynolds numbers up to 1800 and Dimopoulos and Hanratty (1968) presented experimental results for 60<Re<360 using an electrochemical technique.

With regard to the Reynolds number, as pointed out by Coutanceau and Bouard (1977), it is possible to distinguish three categories of flow time evolutions which correspond successively to small (4.4<Re<60), moderate (60<Re<500) and high Reynolds numbers (Re>500). Consideration of small and moderate (low Reynolds numbers) simulations are the area of this study.

7.2 Initial Values and Boundary Conditions

It should be noted that the boundary layer is the region adjacent to a streamlined body surface in viscous flow and that boundary layer flows are assumed to be laminar in this study. Due to the viscosity of the fluid, it sticks to the wall (no-slip condition), and the fluid velocity at the wall is equal to zero. Thus within the cylinder boundary layer, viscous effects dominate the flow. Attention is drawn to the initial values and boundary conditions, whose definition has significant influence in flow simulation.

The inlet boundary velocities U_0 are initially set as constant values based on the Reynolds number ($Re=U_0 \times d_c / \nu$). ν is the kinematic viscosity of the fluid and is defined as the viscosity of water, and d_c is the diameter of the cylinder and is taken as 0.05m (e.g. 1/20 of the domain length for Re=2.04, Re=10 and 40) in these simulations. To provide the required Reynolds number, the velocity of the incoming stream U_0 is adjusted in concert with the specified ν and d_c. Initial values of the velocities, u, v, and the pressure, p, are set to zero everywhere apart from the boundaries, at which flows have an impulsive start.

The grid adaptation methodology, described in Chapter 3, could be controlled by the value of vorticity. The maximum vorticity and minimum vorticity (e.g. ω_{max}=3.0/s and ω_{min}=0.1/s for Re=10) are specified before the adaptation. If an element vorticity is greater than ω_{max} then the element is divided and four new elements created. Otherwise, if an element vorticity is less than ω_{min} then the element and its siblings are removed and replaced by their parent element. Different values of the maximum vorticity and minimum vorticity were experimented with Reynolds numbers, which should be satisfied to give sufficient resolution in separated regions without creating an excessive number of elements.

The no-slip conditions are imposed for the velocity on the cylinder boundary. Neumann boundary conditions are used for the pressure gradient normal to the cylinder boundary which is set equal to zero ($\partial p/\partial n$=0). A prescribed pressure value (p= 0) is utilised at the outlet for scaling the pressure values in the computational domain. A Neumann boundary condition $\partial\phi/\partial n = 0\,(\phi = u, v, p)$ is utilised for upper and lower open boundaries and a Neumann boundary condition $\partial\phi/\partial n = 0\,(\phi = u, v)$ is utilised for the outlet boundary. These boundary conditions are illustrated in Figure 7.2. The calculation is considered to have converged when the sum of normalised residuals falls below 1.0×10^{-4}.

7.3 Low Reynolds Number Fluid Flow Past a Circular Cylinder

The simulation of steady flow at Reynolds number equal to 2.04 without adapting the tri-tree grid is first used to test the grid convergence in this Section. Then the simulations of the steady flows at Reynolds number of 10 and 40 and unsteady flows at Reynolds number of 40 with adaptive tri-tree grids are presented in this Section. Comparisons with a variety of numerical methods and experimental tests have been considered for these problems, with the intent of validating the numerical algorithms

and the use of adaptive tri-tree grids.

The computational configuration for flow past a circular cylinder at Re=2.04, 10 and 40 is shown in Figure 7.3. The flow is from left to right. The centre of the cylinder is located a distance equal to $5d_c$ downstream from the inlet and $15d_c$ upstream from the outlet open boundary of the domain. The upper and lower lateral open boundaries of the domain are each $10d_c$ away from the centre of the cylinder.

7.3.1 Re=2.04 simulations

The lower limit for separation of the boundary layer is reported as Re=3.2 by Nisi and Porter (1923), Re=5.0 by Taneda (1956), Re=6.0 by Homann (1936) and Re=7 by Dennis and Chang (1970) for flow past a circular cylinder. Therefore, studies of the grid convergence of the solution have been carried out for the case of Re=2.04 where the boundary layer remains attached. Figures 7.4 to 7.8 show details near the cylinder surface of grids, each having different maximum and minimum division levels.

The number of elements, nodes and nodes on the cylinder obtained from each grid, as well as the CPU time spent in obtaining a solution on each grid are listed in Table 7.1. The drag and lift coefficients are presented in Table 7.2. Grid convergence was occurred after a maximum division level of 12 and a minimum of 6 and comparison with the experimental data by Tritton (1959) shows good agreement. The results proved that, the smaller elements (or the maximum division level) are a significant fact or in obtaining an accurate solution. For comparison with other data, the Reynolds number and drag coefficient are expressed in logarithmic form, $log_{10}Re$=0.31 (Re=2.04) and $log_{10}C_D$=0.86 (C_D=7.26) and plotted against one another. Experimental and numerical data collected together by Batchelor (1994) are plotted in Figure 7.10 together with the present result.

Based on grid convergence and CPU time, maximum and minimum division levels of 12 and 6 respectively, will be used for base grid generation in the following

simulations. Figure 7.9 shows the base grid for the whole domain and Figure 7.11 shows the grid in the vicinity of the cylinder, which has 238 triangular elements and 237 nodes on the cylinder surface.

Flow patterns: Figure 7.12 illustrates the velocity vectors in the vicinity of the cylinder. Figure 7.13 shows the streamlines at regular spacing equal to $\Delta\psi = 1.99\times10^{-2}\,m^2/s$, where ψ is the stream function and is calculated by equation (6-2). There is no flow separation in this case.

7.3.2 *Re*=10 simulations

Steady flow simulation at Reynolds number equal to 10 past a circular cylinder with adaptive grids is presented. In this case, flow separation behind the cylinder froms two symmetric vortices.

The base grid is shown in Figure 7.9, which contains 3957 grids and 2142 nodes. The adapted maximum division level of 11 and the adapted minimum level of 6 are specified before the adaptation. If an element vorticity is greater than ω_{max} and the division level of the element is less than the maximum adapted level then the element is divided. Otherwise, if the element vorticity is less than ω_{min} and the division level of the element is greater than the minmum adapted level then the element and its siblings are removed. After the base grid solver is run, the adaptive routine and the solver are alternately and iteratively run until no more new elements are generated. The total causes 5 iterations and the final adapted grid contains 4125 grids and 2226 nodes shown in Figure 7.14, and 8640 seconds of CPU were consumed.

The drag coefficient is calculated to be C_D =2.99, the lift coefficient is $C_L = -6.66\times10^{-3}$. The Reynolds number and the drag coefficient are expressed in logarithmic form, $log_{10}Re$=1.0 (Re=10) and $log_{10}C_D$=0.47 (C_D=2.99), and plotted in Figure 7.10. It shows that there is a close correlation between the present result and the published results. However, the present drag coefficient is little higher than the

numerical results obtained by Collins and Dennis (1973), C_D=2.94.

The calculation of the drag coefficient will be discussed here. It is well known that in most cases no analytical solution exists for the flow variables (velocity and pressure) and the force values in viscous flow. The fluid flow exerts forces on the immersed cylinder body which are due to the integrated fluid pressure acting perpendicular to the cylinder surface, and skin friction resulting from tangential viscous shear stresses (see formulae (7-3) and (7-4)). There are a number of factors influencing the drag coefficient calculations, namely the cylinder surface roughness and the type of numerical method being applied. For example, the present work approximates the curved cylinder surface with straight segments and the interpolation error increased by the distorted triangles around the cylinder. Depending on the fineness of the grid, a convergent trend can be found when the maximum level is increased. However, the increase of elements also brought about an increased in computational time and storage. Due to present computational storage availability, it is impossible to further increase the number of elements above 40,000.

Flow patterns: It is well known that, when a fluid flows past a bluff body such as a circular cylinder, separation occurs on each side of the cylinder and two symmetrical recirculations develop in its near wake, the length of this recirculation increases with time and Reynolds number. These recirculations are called Föppl vortices. Figure 7.15 shows the details of the separation angle, θ_s, measured from the centre line of the cylinder to the line of the centre of cylinder to the separation point (x =0.022m, y =0.013m) on the cylinder (defined in Figure 7.1). The wake length, L, is the distance measured from the rear cylinder surface to the rear stagnation point (x=0.0405m, y =0.0m) where velocities are zero. Comparison of these data is shown in Table 7.3 and 7.4, with those reported by Kawaguti and Jain (1966), Dennis and Chang (1970) and Saalehi (1995), shows good agreement.

Figure 7.16 illustrates the velocity vectors in the vicinity of the cylinder. Figure 7.17 shows the streamlines at regular spacing equal to $\Delta\psi = 2.15\times10^{-2}\,m^2/s$. Figure 7.18

shows pressure contours at $\Delta p = 2.73\times10^{-1}$Pa between the maximum and minimum values and Figure 7.19 gives the vorticity contours at $\Delta\omega$=0.25/s intervals, calculated from equation (3-9). These flow patterns show good agreement with numerical simulations provided by Greaves (1995) and Saalehi (1995).

7.3.3 *Re*=40 simulations

The flow past a circular cylinder at *Re*=40 has become a standard benchmark problem. The simulations with steady and unsteady flows are presented herein to validate the capability of our code.

7.3.3.1 *Steady flow simulation*

The base grid is shown in Figure 7.9. Figure 7.20 shows the velocity vectors in the vicinity of the cylinder and Figure 7.21 shows the streamlines at regular intervals equal to $\Delta\psi = 8.62\times10^{-2}$ m^2/s, calculated using the base grid. The drag coefficient is C_D=1.75 and the lift coefficient is $C_L = -5.89\times10^{-4}$. The recirculation length of Föppl vortices and the separation angles have been list in Table 7.3 and 7.4 and show that the base grid simulation could not achieve very high accuracy.

After the base grid solver is run, the adaptive routine and solver are alternately and iteratively run until no more new elements are generated. The total causes 17 iterations and cost 111,420 seconds of CPU time. The adapted grid contains 10465 elements and 5396 nodes shown in Figure 7.22. Of the total CPU time, the amount used for generation of the adapted grid (17 iterations of the adaptive routine) was 5,100 seconds, which is much less than the CPU time needed to obtain the flow solution. The drag coefficient is C_D=1.69 and the lift coefficient is $C_L = -3.80\times10^{-3}$. The Reynolds number and the drag coefficient are expressed in logarithmic form, $log_{10}Re$=1.6 (*Re*=40) and $log_{10}C_D$=0.22 (C_D=1.69) and plotted in Figure 7.10. There is a close correlation between the present result and the published results.

Flow patterns: For the adapted grid simulation, Figure 7.23 illustrates the velocity vectors in vicinity of the cylinder, Figure 7.24 shows the streamlines at regular spacing equal to $\Delta\psi = 9.69\times10^{-2}\,\mathrm{m^2/s}$ and the symmetric recirculations that exist behind the cylinder can be seen clearly. Figure 7.25 shows pressure contours at $\Delta p = 3.832\,\mathrm{Pa}$ between the maximum and minimum values and Figure 7.26 gives the vorticity contours at $\Delta\omega$=0.32/s intervals. These flow patterns show good agreement with numerical simulations provided by Ding and Kawahara (1999), Greaves (1995) and Saalehi (1995).

Figure 7.27 shows the details of the separation angles, θ_s measured from the centre line of cylinder to the line of the centre of cylinder to the separation point (x=0.015m, y=0.021m) on the cylinder, and the wake length, L, is the distance measured from the rear cylinder surface to the rear stagnation point (x=0.138m, y=0.0m) where velocities are zero. Comparison between recirculation lengths of Föppl vortices obtained herein and those in published data is shown in Table 7.3. Table 7.4 compares the separation angles, θ_s, with published data by Kawaguti and Jain (1966), Collins and Dennis (1973), Dennis and Chang (1970) and Saalehi (1995). It is clear that grid adaptation can increase the numerical accuracy of the solution compared with the base grid simulation.

7.3.3.2 *Unsteady flow simulation*

Although the flow past a circular cylinder for Re=40 is steady, simulations were carried using the unsteady formulation to validate the time-dependent finite volume method. The final results should be the same for both tests and should agree with published data.

The particular refinement of the numerical simulation allows the main geometrical parameters of the closed wake with time to be visualised, even during the first stages of its evolution. For example, the simulation used the unsteady formulation with non-dimensional time step $\Delta t^* = 1.0$ from the impulsive start of the flow. Steady state is

presumed to occur after the sum of normalised residuals falls below 1.0×10^{-4} for each successive time step. The simulation took 14 adaptations and stopped at non-dimensional time $t^*=14$ at which point no more new elements or nodes are generated. Figure 7.28 depicts the time history of the drag coefficient C_D in order to observe the time convergence of the method. The finial drag coefficient is $C_D=1.69$ and the lift coefficient is $C_L=-3.84\times10^{-3}$, which are both same as for the steady flow simulation. These data have been listed in Table 7.8.

Flow patterns: The final adapted grid contains 10539 elements and 5461 nodes as shown in Figure 7.29 and consumed 140,400 seconds of CPU time. As would be expected, symmetric of Föppl vortices are also generated by the unsteady simulation. Figure 7.30 illustrates the velocity vectors in the vicinity of the cylinder.

The comparison of some critical parameters with experimental data, is carried out by considering the following aspects. The evolution in time of the close-wake length L versus the diameter of cylinder d_c, L/d_c, is shown in Figure 7.31. We see that the closed-wake length is almost a linear function of time at the beginning of the establishment period. Then the curves tend towards their asymptotic values. The parameter has shown good agreement between the present results and those of the experimental data given by Coutanceau and Bouard (1977) and numerical results by Chen *et al.* (1999). The evolution in time of the position of the vortex cores (a, b) versus the diameter of cylinder d_c are presented in Figure 7.32 and 7.33. The evolution in time of the separation angle, θ_s, is shown in Figure 7.34. These were found to change rapidly at the start of the motion, with the position of the vortex cores and separation angle tending asymptotically toward their final period. Again, these parameters have shown good agreement between the present results and those of the experimental data given by Coutanceau and Bouard (1977). Those final parameters are the same as for the steady flow simulation and are shown in Table 7.3 and Table 7.4.

7.4 Unsteady Flow Past Two and Three Circular Cylinders at *Re*=40

Fluid flow structure interaction commonly occurs in arrays of cylinders and influences the force experienced by them. Investigation of such flow interaction is important in the area of flow induced vibration (if the cylinder is flexibly mounted). It is well known that these vibrations can sometimes lead to resonance of the structure. Oscillation of the cylinder can also cause the vortex shedding pattern to change and the vortex shedding frequency (Strouhal number) to shift. Arrays of cylinders which may be subject to flow interaction occur in various industrial applications, such as offshore jacket structures, radar mast vibrations and heat exchange tube vibrations.

In this Section, unsteady flow simulations at Reynolds number equal to 40 past fixed multiple circular cylinders are predicted. Although this Reynolds number is very low compared with that occurring in practical applications, the basic flow characteristics will be similar to those predicted here. The influence on flow patterns and force coefficients is assessed by changing the distance between the cylinders. The results obtained here are compared with numerical data reported by Greaves and Borthwick(1997).

Unlike flow past a single cylinder, the study of flow interactions past two or three cylinders depends on the non-dimensional distance between the centres of the cylinders, denoted by T/d_c in the cross-flow direction and P/d_c in the flow direction. The configuration for flow past two cylinders with respect to the free stream flow direction can be broadly classified in three groups: in one group, the cylinders face the flow side by side at any transverse spacing as shown in Figure 7.35; in the second group, the cylinders are in a tandem arrangement, one behind the other at any longitudinal spacing as shown in Figure 7.36; and all other combinations of longitudinal and transverse spacing represent staggered arrangements, the third group. The transverse and tandem arrangements for flow past two cylinders will be discussed in this study. Simulation of flow past three cylinders is also covered here.

The computational domains for flow past two cylinders are defined herein. The d_c is the diameter of the equal-size cylinders and is taken to be 0.05m (1/25 of the domain length) in these simulations. The upstream boundary is located at $5d_c$ from the centre of the first cylinder. The downstream boundary is located at $20d_c$ from the centre of the upstream cylinder. The upper and lower lateral open boundaries of the domain are each $7d_c$ for transverse and $8d_c$ for tandem arrangement away from the centre of the cylinder. The simulation used the unsteady formulation with non-dimensional time step Δt^*=1.0 from the impulsive start of the flow. No-slip conditions are imposed for the velocity on the cylinder walls and the rest of the boundary conditions as well as simulation conditions are the same as for the simulation of flow past a single cylinder.

7.4.1 Flow past two cylinders in transverse arrangement

The transverse arrangement will be considered first. The flow interaction between two cylinders depends on their spacing. The arrangements at T/d_c=2.0 and T/d_c=5.0 have been predicted here.

Transverse arrangement T/d_c=2.0: The initial grid for two circular cylinders placed in transverse arrangement at T/d_c=2.0, contains 7034 cells and 3797 nodes as shown in Figure 7.37 and Figure 7.38. The simulation took 6 adaptations (the interior of the grid is adapted every three time steps) and stopped at non-dimensional time t^*=20 at which point no more new elements or nodes are generated. The final adapted grid contains 19640 cells and 10100 nodes as shown in Figure 7.39. The calculation used 262,800 seconds of CPU time.

Flow patterns: Figure 7.40 illustrates the velocity vectors in the vicinity of the cylinder, Figure 7.41 shows pressure contours at $\Delta p = 5.13\,\text{Pa}$ between the maximum and minimum values and Figure 7.42 shows the streamlines at regular spacing equal to $\Delta\psi = 7.36\times10^{-2}\,\text{m}^2/\text{s}$. Two recirculations exist behind each cylinder but each has an individual asymmetric wake structure. These flow patterns have shown good agreement with numerical simulations by Greaves and Borthwick(1997).

Drag and lift coefficients: The force coefficients for the upper cylinder are calculated as C_D=2.18 and C_L=0.63 and for the lower cylinder as C_D=2.18 and C_L=-0.68. These data are final again with data given by Greaves and Borthwick (1997), where the force coefficients are C_D=1.77 and C_L=0.37 for the upstream cylinder and C_D=1.77 and C_L=-0.39 for the downstream cylinder. The reason for the difference is most likely due to the quality of the grids at the circular cylinder boundaries. Greaves and Borthwick(1997) used quadtree grids with a stepped profile at the cylinder boundaries, whereas smooth cylinder boundaries are achieved in this work by using tri-tree grids.

Transverse arrangement T/d_c=5.0: The initial grid for two circular cylinders placed in transverse arrangement at T/d_c=5.0, contains 7466 cells and 4021 nodes as shown in Figure 7.43 and Figure 7.44. The simulation took 5 adaptations (the interior of the grid is adapted every three time steps) and stopped at non-dimensional time t^*=17 at which point no more new elements or nodes are generated. The finial adaptive grid contains 20006 cells and 10283 nodes as shown in Figure 7.45. The calculation used 295,200 seconds of CPU time.

Flow patterns: Figure 7.46 illustrates the velocity vectors in the vicinity of the cylinder, Figure 7.47 shows pressure contours at $\Delta p = 4.63\,\text{Pa}$ between the maximum and minimum values and Figure 7.48 shows the streamlines at regular spacing equal to $\Delta\psi = 9.86\times10^{-2}\,\text{m}^2/\text{s}$. The separated regions behind each cylinder can be seen to be slightly asymmetric.

Drag and lift coefficients: The force coefficients for the upper cylinder are calculated as C_D=1.87 and C_L=0.12 and for the lower cylinder as C_D=1.87 and C_L=-0.16.

From the comparison of results for flow past two cylinders at arrangements T/d_c=2.0 and T/d_c=5.0 as well as a single cylinder, some remarks on the interaction can be noted, as follows:

Flow characteristics in the recirculating zone: The cylinder spacing affects the flow patterns around the two cylinders. The flow patterns can be distinguished by examining the closed wake length L, the position of the vortex cores (a, b) and the separation angles, θ_s. These parameters are all listed in Table 7.5. Examination of the separated region of each cylinder shows that the asymmetry in the wake at T/d_c=2.0 is larger than at T/d_c=5.0 because the flow interaction is greater when the cylinders are spaced closer together. T/d_c=5.0 is more similar to the single cylinder than T/d_c=2.0 because the flow interaction is less pronounced.

Drag and lift force coefficients on two cylinders. The remarkable fact regarding the interference between the two cylinders is that the drag and lift coefficients are all higher than those of a single cylinder. The reason is that the gap flow biased to one side will produce a resultant force on the cylinder which is deflected relative to the free stream direction. Therefore, there will be a component of the force acting perpendicular to the free stream direction which may increase the lift force. Table 7.8 shows that the drag and lift coefficients at T/d_c=2.0 are larger than at T/d_c=5.0.

7.4.2 Flow past two cylinders in tandem arrangement

Flow past two cylinders in tandem arrangement P/d_c=2.0 and P/d_c=5.0 are presented in this Section. There are no references corresponding to the exact cases. However, some research workers have studied and reported on similar cases. Zdravkovich's (1977) experimental observation for higher Reynolds number predicted the flow interference between two cylinders in transverse, tandem and staggered arrangements and Tanida *et.al*'s (1973) experimental observation in tandem at R_e=80 and R_e=3400 measured the drag coefficient, lift coefficient and Strouhal number with the distance between their centres from 2.5 to 20 diameters. Numerical investigations, including Chambaral *et al.* (1991) studied the vortex shedding for Re=100 using the finite element method and Mittal *et al.* (1997) presented the vortex shedding for Re=100 and Re=1000 in tandem arrangements with the distance between their centres varying between 2.5 and 5.5 diameters.

Tandem arrangement P/d_c=2.0: The initial grid for two cylinders placed in tandem arrangement at P/d_c=2.0, contains 7027 cells and 3793 nodes as shown in Figure 7.49 and 7.50. The grid is adapted every time step until no more new elements are generated. The final adapted grid contains 13263 cells and 6911 nodes as shown in Figure 7.51. The simulation required 4 time steps and used 115,200 seconds of CPU time.

Flow patterns: Figure 7.52 illustrates the velocity vectors close to the cylinder, Figure 7.53 shows pressure contours at $\Delta p = 3.42\,Pa$ between the maximum and minimum values and Figure 7.54 shows the streamlines at regular spacing equal to $\Delta\psi = 1.05\times10^{-1}\,\mathrm{m^2/s}$. Symmetric separated regions can be seen behind each cylinder. The vortices which have formed behind the upstream cylinder are attached to the downstream cylinder.

Drag and lift coefficients: The force coefficients for the upstream cylinder are calculated as C_D=1.63 and C_L=-6.50×10^{-3} and for the downstream cylinder as C_D=0.15 and C_L=-4.37×10^{-1}. The drag coefficient on the upstream cylinder is much higher than on the downstream cylinder. This is because the vortices behind the upstream cylinder do not have sufficient room to grow and to develop. Since the downstream cylinder is in the attached vortex region of the upstream cylinder, its equivalent oncoming 'free-stream' velocity is quite weak. This leads to a weaker wake behind it than that of flow past a single cylinder.

Tandem arrangement P/d_c=5.0: When the spacing between the two cylinders is increased the separated region behind the upstream cylinder is elongated. For example, the initial grid for two circular cylinders placed in tandem arrangement at P/d_c=5.0, contains 7187 cells and 3877 nodes and is shown in Figure 7.55 and Figure 7.56. The grid is adapted every time step until no more new elements are generated. The final adapted grid contains 12671 cells and 6619 nodes as shown in Figure 7.57. The simulation required 15 time steps and consumed 158,400 seconds of CPU time.

Flow patterns: Figure 7.58 illustrates the velocity vectors close to the cylinder, Figure 7.59 shows pressure contours at $\Delta p = 3.83\ Pa$ between the maximum and minimum values and Figure 7.60 shows the streamlines at regular spacing equal to $\Delta\psi = 7.93\times10^{-2}\ m^2/s$. The separated region behind the upstream cylinder has elongated, but is still attached to the downstream cylinder.

Drag and lift fore coefficients: The force coefficients for the upstream cylinder are calculated as C_D=1.66 and C_L=-9.72×10^{-3} and for the downstream cylinder as C_D=0.44 and C_L=9.68×10^{-2}. Compared with the arrangement at P/d_c=2.0, the drag coefficients for two cylinders are higher because the flow interaction is reduced as the cylinder spacing is increased.

The results obtained enable us to draw the following conclusions about flow patterns for two cylinders placed in the tandem arrangement:

Flow characteristics in the recirculating zone: As discussed above, the two cylinders in tandem arrangement have symmetric separated flow regions. In both cases, the separated region behind the upstream cylinder is attached to the downstream cylinder. It is clear that larger spacing (P/d_c=5.0) has less flow interference between two cylinders than smaller spacing (P/d_c=2.0). The flow region can be distinguished by examining the closed-wake length L, the position of the vortex cores (a, b) and the separation angles, θ_s. These parameters are all listed in Table 7.6.

Drag and lift force coefficients on two cylinders. The clear fact concerning flow interference between the two cylinders is that the drag coefficients for the upstream cylinder are lower than for a single cylinder. The reason is that the vortex flow between the two cylinders will produce large negative force on the upstream cylinder. Therefore, there will be a component of the force acting parallel to the free stream direction which may reduce the drag force. The drag coefficient for the downstream

cylinder for P/d_c=5.0 has a higher value than for P/d_c=2.0. This is because the behaviour of the upstream cylinder is similar to that of a single cylinder. There is less flow interaction and so it does not affect the downstream cylinder much. This data has been listed in Table 7.8.

7.4.3 Flow past three cylinders

The computational configuration for flow past three cylinders is shown in Figure 7.61. The initial grid for three circular cylinders placed in arrangement at T/d_c=2.0 and P/d_c=2.5, in which the third cylinder is shielded from the incoming flow behind the pair of transverse cylinders, contains 10834 cells and 5829 nodes and is shown in Figure 7.62 and Figure 7.63. The grid is adapted every time step until no more new elements are generated. The final adapted grid contains 23528 grids and 12176 nodes as shown in Figure 7.64. The simulation required 9 time steps and consumed 298,800 seconds of CPU time.

Flow patterns: Figure 7.65 illustrates the velocity vectors close to the cylinder, Figure 7.66 shows pressure contours at $\Delta p = 5.01 Pa$ between the maximum and minimum values and Figure 7.67 shows the streamlines at regular spacing equal to $\Delta\psi = 6.93\times10^{-2}$ m^2/s. From these Figures it can be seen that the flow interaction between these three cylinders is complicated, with the pair of transverse cylinders affecting each other as well as affecting the third cylinder. These flow patterns show good agreement with numerical simulations by Greaves and Borthwick(1997).

Drag and lift coefficients: For the upper cylinder, the force coefficients are calculated to be C_D=2.03 C_L=0.67 for the lower cylinder C_D=2.04 and C_L=-0.69 and for the third cylinder C_D=0.93 and C_L=-1.02×10^{-4}. Greaves and Borthwick(1997) reported that the force coefficients are C_D=1.63 and C_L=0.36 for the upper cylinder, C_D=1.64 and C_L=-0.38 for the lower cylinder and C_D=0.71 and C_L=4.0×10^{-3} for the third cylinder.

The numerical simulation of flow past multiple cylinders with *Re*=40 has been carried out. The interaction of flow has been studied on the basis of the flow patterns and force coefficients. The aim of these results is to help in the understanding of the physical phenomena involving multiple cylinders even at small Reynolds number.

It should be noted that the final adapted grids are often slightly asymmetric in the present simulations. The reason for this is connected with numerical inaccuracies during the simulation especially related to processes such as interpolation. It follows from this argument that the more refined the grids (i.e. the smaller the grid elements are) the greater the likelihood of the final grid being asymmetric. This is demonstrated in Figures 7.68 and 7.69 which have final adapted grids for unsteady flow simulation past a single cylinder at *Re*=40. Figure 7.68 was obtained using maximum and minimum adaptive division levels of 11 and 6, and is slightly asymmetric. Figure 7.69 was obtained using maximum and minimum adaptive division levels of 10 and 6, and is symmetric. Another factor is that a fine balance must be struck when selecting the maximum value of vorticity to be used for grid adaptation. Too small a vorticity value will result in too fine, and most likely an asymmetric, grid, whilst a large value of vorticity value will entail loss in accuracy.

7.5 Unsteady Flow Past a Circular Cylinder at *Re*=200

The physical flows at *Re*=200 have an unsteady wake formed by shed vortices. The computational configuration for flow past a circular cylinder at *Re*=200 is shown in Figure 7.70. The diameter of the cylinder, d_c, is taken as 0.05m (1/28 of the domain length). The centre of the cylinder is located a distance equal to $5d_c$ downstream from the inlet and $23d_c$ upstream from the outlet open boundary of the domain. The upper and lower lateral open boundaries of the domain are each $8d_c$ away from the centre of the cylinder.

The base grid used is shown in Figure 7.71, in which the domain has 4115 cells and 2227 nodes. The simulation is advanced in non-dimensional time step $\Delta t^* = 0.25$. The

interior of the grid is adapted every 4 time steps up to t^*=5, after that it is adapted every 20 time steps to save CPU time.

Evolution with time of the closed wake: The flow has an impulsive start and the simulation was run until periodic shedding of vortices occurred. The plots of the adapted grids and streamlines obtained from simulation of the early stages (t^*=1.0 and t^*=2.0) are shown from Figure 7.72 to Figure 7.75. The shape of the closed wake is stable and symmetric. Steady state is presumed to occur after the sum of normalised residuals falls below 1.0×10^{-4} for each successive time step.

At t^*=2.25, the asymmetry in the wake and the shifting of the main eddy core appears clearly in Figure 7.76. After this time, the flow never becomes steady again since the wake continues to grow and the vortices become unstable and eventually are shed. Here, computational convergence is considered to have occurred once the iterations after 10,000. This restricts the amount of CPU time used which otherwise becomes prohibitively expensive. Figure 7.77 shows the adapted grid at t^*=3.0, in which the asymmetric grid can be seen. The next goal is to predict the time evolution of vortex shedding phenomena.

Vortex shedding: Figure 7.78 shows the adapted grid at t^*=66.0, which has refinements in the vortex shedding area. Figure 7.80 shows the streamlines in sequence for one complete cycle of Kàmàn vortex shedding. These Figures are all plotted with the streamlines at regular spacing equal to $\Delta\psi = 0.32\,\mathrm{m^2/s}$. Figure 7.80(*a*) shows a vortex forming on the upper surface of the cylinder and a vortex from the lower surface that has just been shed into the wake. In Figure 7.80(*b*) the vortex from the upper surface has grown but remains attached; it is shed in Figure 7.80(*c*) and a new vortex is forming at the lower surface of the cylinder. In Figure 7.80(*d*) the lower vortex has grown but remains attached whereas the upper vortex has been swept away in the wake. Figure 7.80(*e*) is similar to 7.80(*a*) and shows the lower vortex just having detached from the rear of the cylinder and a new vortex forming at the upper surface. These patterns are in good agreement with numerical simulations

provided by Chen *et al.* (1999) shown in Figure 7.81.

The drag and lift coefficients for the streamline patterns shown in Figure 7.80 are calculated as following: (*a*) C_D=1.32, C_L=5.54×10^{-2} (*b*) C_D=1.50, C_L=-0.59 (*c*) C_D=1.42, C_L=-0.21 (*d*) C_D=1.44, C_L=0.48 and (*e*) C_D=1.32, C_L=4.27×10^{-2}. Figure 7.79 shows the evolution of lift and drag coefficients on the cylinder with non-dimensional time. Table 7.9 shows a comparison between the present results (C_D, C_L and S) and those obtained by other experimental and numerical results. The present numerical data are in reasonable agreement with the other published data. The present period for the vortex shedding cycle calculated here is 5.0 (the corresponding Strouhal number is 0.20) and Chen *et al.* (1999)'s period is 5.072 (the corresponding Strouhal number is 0.197).

7.6 Closure

The results for the flow past circular cylinders are successfully obtained and validated against published data at low Reynolds number. All of the results presented in this thesis have been obtained on Alpha workstations.

Subdivision level		Cells	Nodes	Nodes on the cylinder	CPU time
Max.	Min.				
10	6	1824	987	60	24mins.
11	6	2518	1363	117	36mins.
12	6	3957	2142	237	52mins.
12	7	7061	3741	237	2hrs.50mins.
13	6	8890	4726	472	4 hrs.49mins.

Table 7.1 Grid convergence test for Re=2.04 flow simulation showing variation in grids, nodes and CPU times

Subdivision level		Drag coefficient, C_D,		Lift coefficient, C_L,	
Max.	Min.	present study	published data	present study	published data
10	6	7.77	7.26[1]	1.87×10^{-3}	–
11	6	7.33		1.96×10^{-3}	
12	6	7.26		-1.13×10^{-3}	
12	7	7.27		-1.12×10^{-3}	
13	6	7.25		-1.18×10^{-3}	

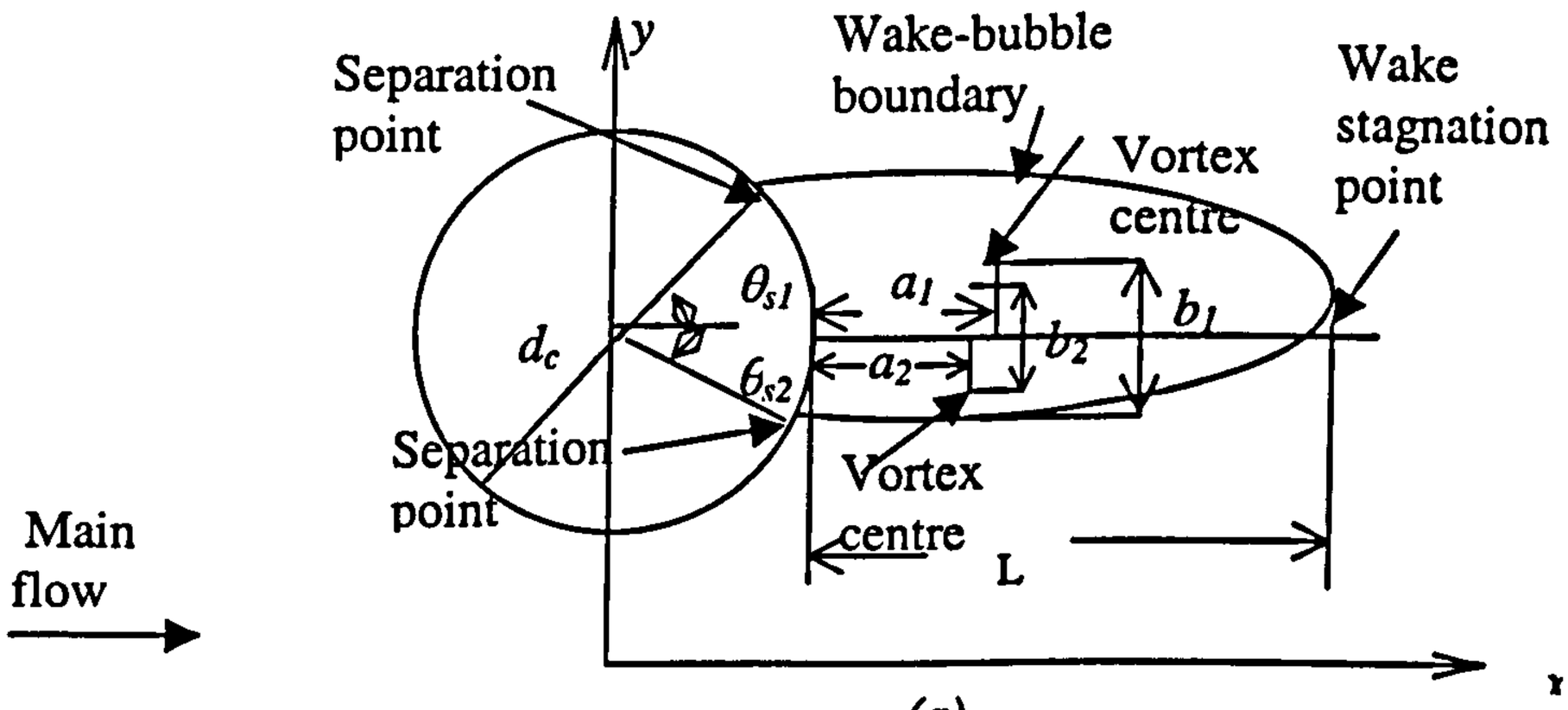
Table 7.2 Grid convergence test for Re=2.04 flow simulation showing variation in drag and lift with increasing grid resolution and comparison with experimental data reported by [1]Tritton (1959)

Re	L/d_c Present study	L/d_c Published data
10	0.30 (with adaptation)	0.30[1] 0.265[2] 0.28[3]
40	2.26 (with adaptation) 1.84 (base grids)	2.515[1] 2.16[2] 2.18-3.02[3]

Table 7.3 Comparison between recirculation lengths of Föppl vortices obtained with data reported by [1]Kawaguti and Jain (1966), [2]Collins and Dennis (1973) and [3]Saalehi (1995) for steady flow past a cylinder at Re=10 and Re=40 simulations

Re	θ_s Present study	θ_s Published data
10	30.5^0 (with adaptation)	30.8^0 [1] 29.6^0 [2] 29.5^0 [3]
40	53.56^0 (with adaptation) 52.16^0 (base grids)	53.7^0 [1] 53.8^0 [2] 53.5^0 [3]

Table 7.4 Comparison between separation angles obtained with data reported by [1]Kawaguti and Jain (1966), [2]Collins and Dennis (1973) and[3]Saalehi (1995) for steady flow past a cylinder at Re=10 and Re=40 simulations

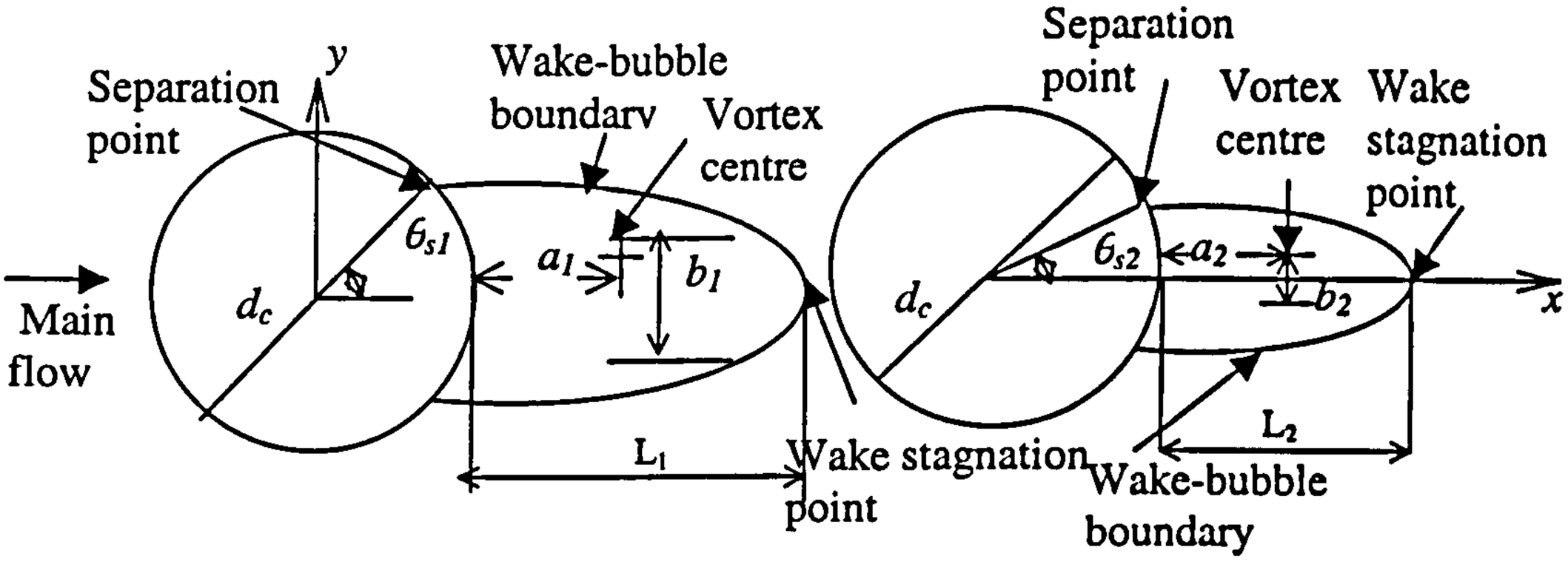


(a)

Re=40	L/d_c	θ_{s1}	θ_{s2}	a_1/d_c b_1/d_c	a_2/d_c b_2/d_c
A single cylinder	2.26	53.56^0	-	0.75 0.58	-
T/d_c=2.0	2.0	55.88^0	-43.24^0	0.6 0.5	0.70 0.24
T/d_c=5.0	2.20	52.43^0	-48.92^0	0.78 0.62	0.60 0.55

(b)

Table 7.5 (a) Geometrical parameters of the closed wake (b) The present numerical values of the closed-wake geometrical parameter for a single cylinder and two cylinders in transverse arrangement at Re=40

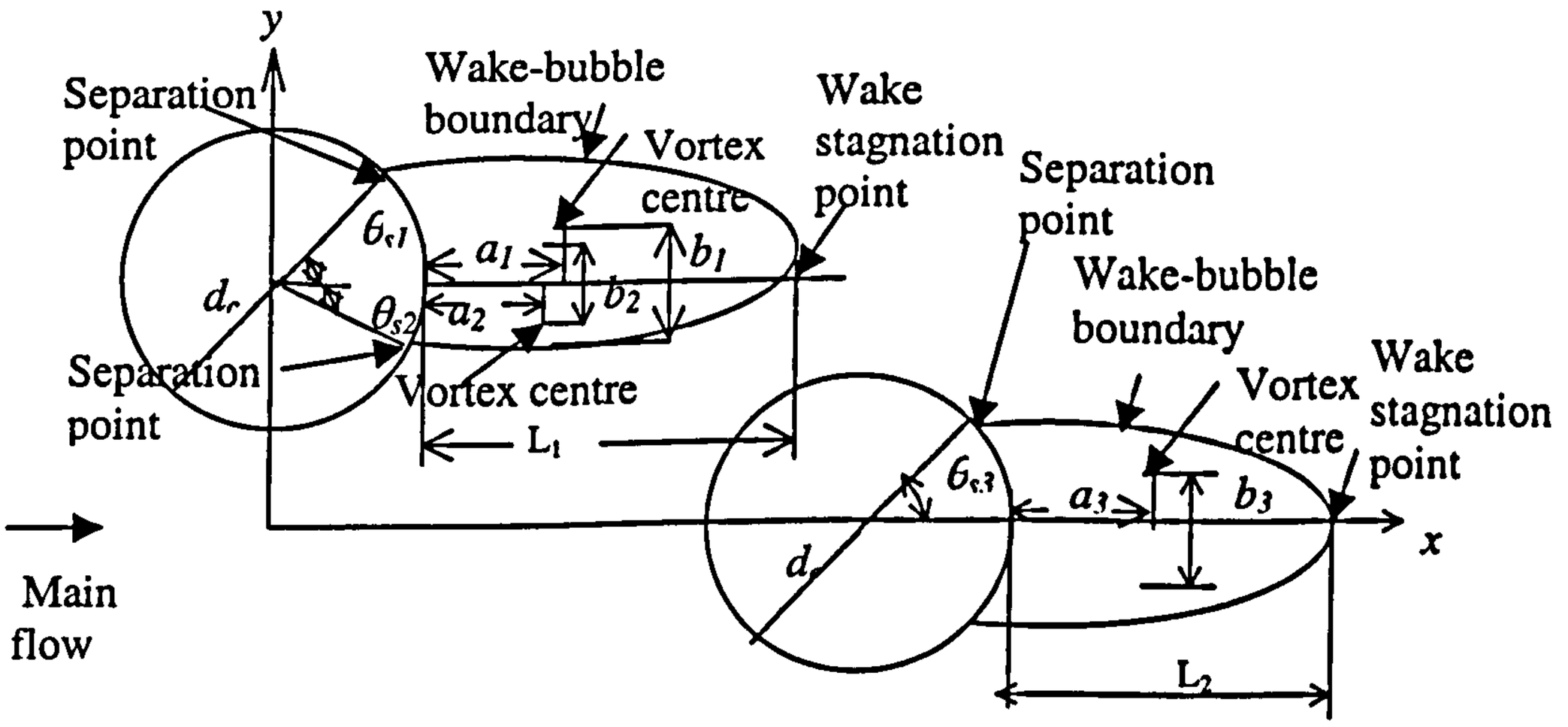


(a)

Re=40	L_1/d_c	L_2/d_c	θ_{s1}	θ_{s2}	a_1/d_c b_1/d_c	a_2/d_c b_2/d_c
A single cylinder	2.26	-	53.56^0	-	0.75 0.58	-
P/d_c=2.0	-	1.74	54.32^0	40.36^0	0.5 0.65	0.7 0.48
P/d_c=5.0	2.9	0.9	54.32^0	40.74^0	1.4 0.56	0.6 0.2

(b)

Table 7.6 (a) Geometrical parameters of the closed wake (b) The present numerical values of the closed-wake geometrical parameter for a single cylinder and two cylinders in tandem arrangement at Re=40



(a)

Re=40	L_1/d_c	L_2/d_c	θ_{s1}	θ_{s2}	θ_{s3}	a_1/d_c b_1/d_c	a_2/d_c b_2/d_c	a_3/d_c b_3/d_c
T/d_c=2.0 P/d_c=2.5	1.1	8.6	60.25^0	-43.03^0	72.08^0	0.34 0.868	0.62 0.18	3.8 0.936

(b)

Table 7.7 (a) Geometrical parameters of the closed wake (b) The present numerical values of the closed-wake geometrical parameter for three cylinders at Re=40

Re=40	Drag coefficient, C_D,						Lift coefficient, C_L,					
	present study			published data			present study			published data		
	1	2	3	1	2	3	1	2	3	1	2	3
a single cylinder	1.69			1.62[1]			-3.84 E-03					
two cylinders T/d_c=2.0	2.18	2.18		1.77[2]	1.77[2]		0.63	-0.68		0.37[2]	-0.39[2]	
two cylinders T/d_c=5.0	1.87	1.87					0.12	-0.16				
two cylinders P/d_c=2.0	1.63	0.15					-6.50 E-03	-4.37 E-03				
two cylinders P/d_c=5.0	1.66	0.44					-9.72 E-03	-9.68 E-02				
three cylinders T/d_c=2.0 P/d_c=2.5	2.03	2.04	0.93	1.63[2]	1.64[2]	0.71[2]	0.67	-0.69	-1.02 E-4	0.36[2]	-0.38[2]	4.0 E-3[2]

Table 7.8 The drag and lift coefficients for Re=40 flow past a single cylinder and multi-cylinders comparison with numerical data reported by [1]Dennis and Chang (1970) and [2] Greaves and Borthwick (1997)

	C_D	C_L	S
Present study	1.41 ± 0.09	+0.71 -0.59	0.200
Wille (1960) (experimental) Roshko (1954) (experimental)	1.3		0.19
Lecointe and Piquet (1984)	1.46 ± 0.04	± 0.70	0.227
Chen *et al.*(1999)	1.33 ± 0.04	± 0.72	0.197
Chan and Anastasion (1999)	1.48 ± 0.05	± 0.63	0.183

Table 7.9 The values of the drag and lift coefficients, as well as the Strouhal number for Re=200 flow past a single cylinder

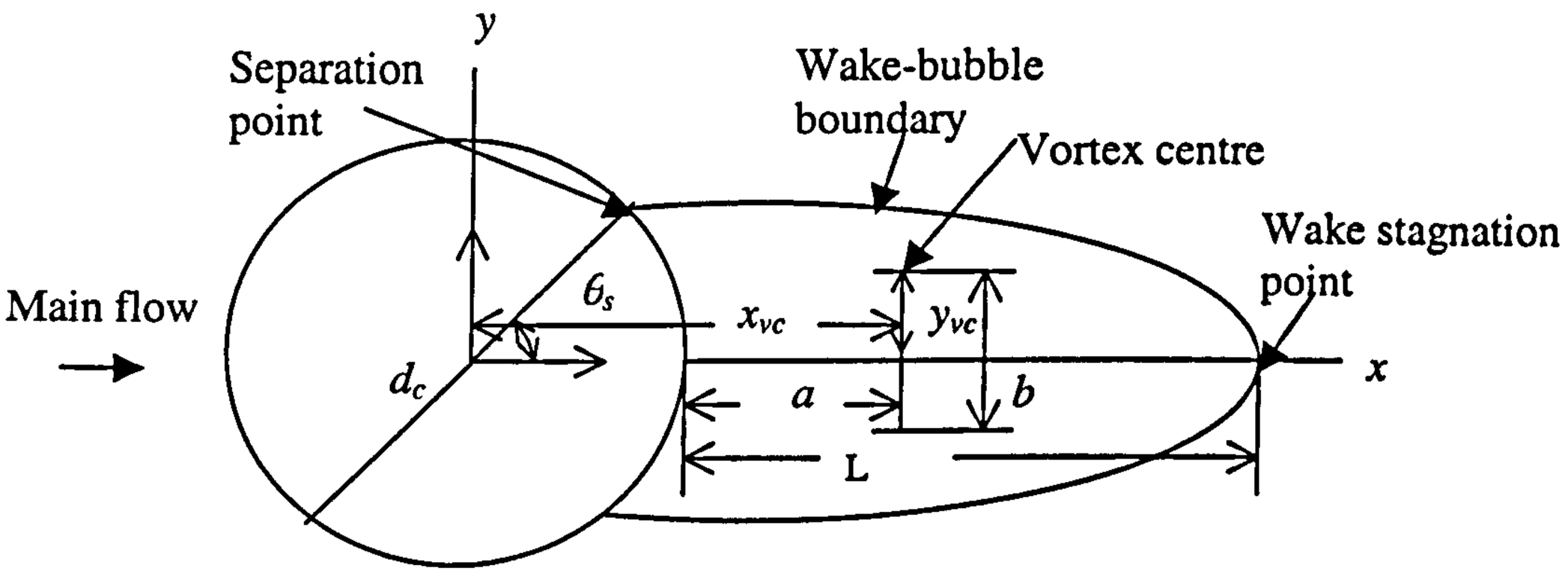


Figure 7.1 Geometrical parameters of the closed wake

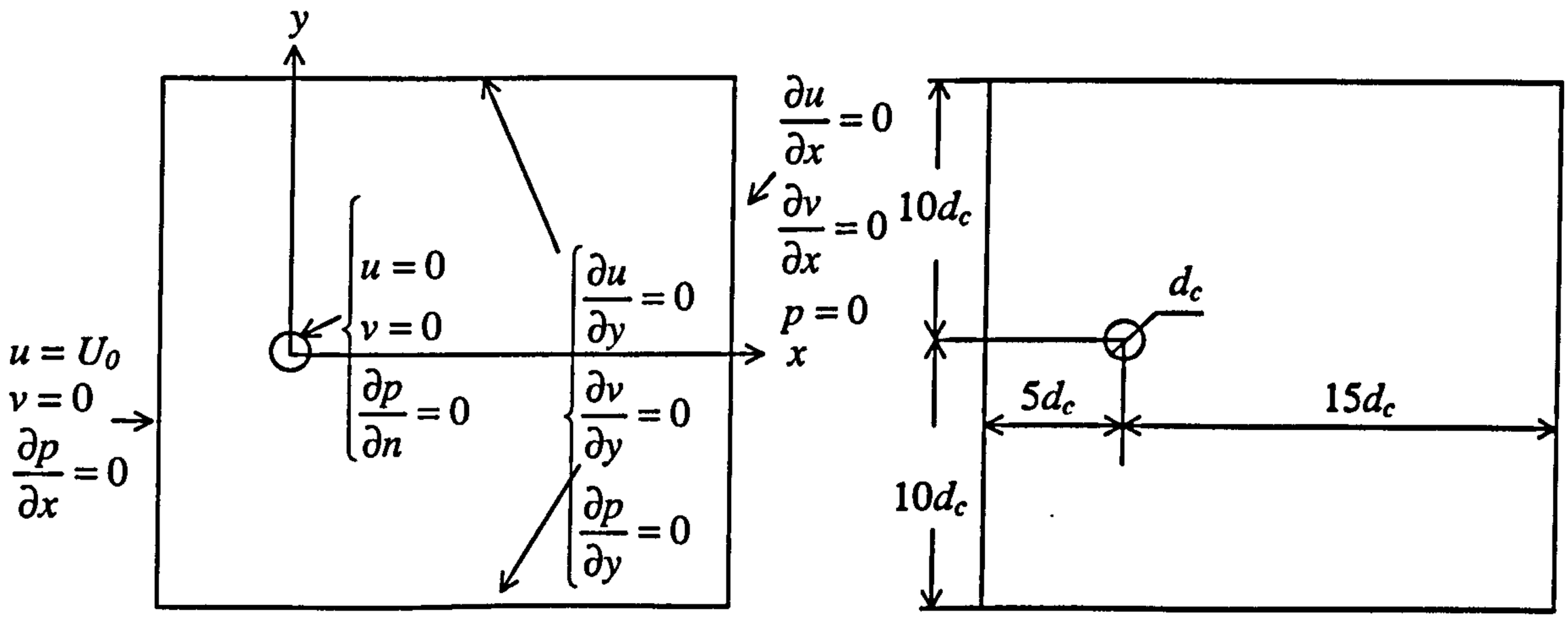


Figure 7.2 Boundary conditions

Figure 7.3 Computational configuration

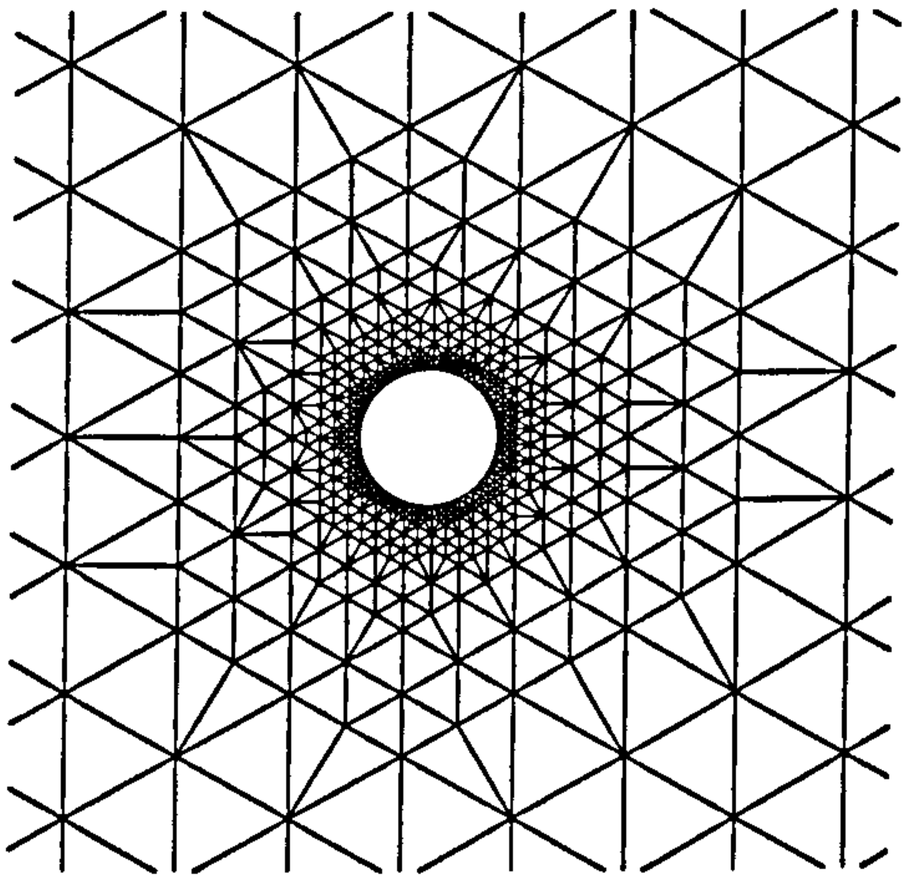


Figure 7.4 Detail at the cylinder of base grid, $level_{max}$= 10 and $level_{min}$=6

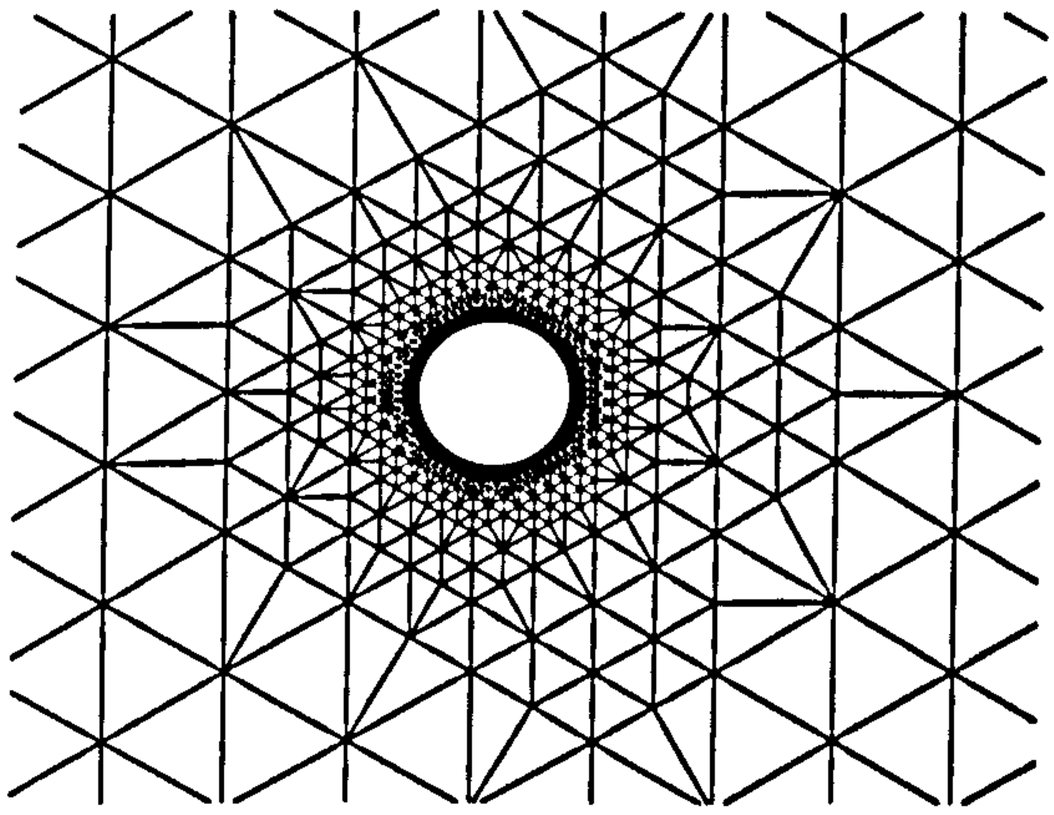


Figure 7.5 Detail at the cylinder of base grid, $level_{max}$= 11 and $level_{min}$=6

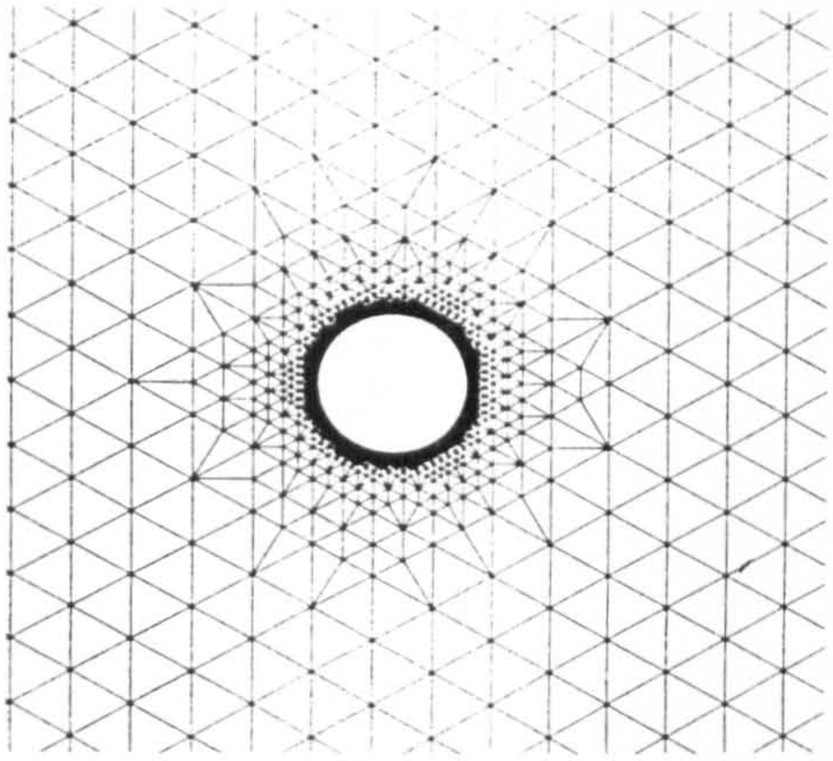


Figure 7.6 Detail at the cylinder of base grid, $level_{max}$=12 and $level_{min}$=7

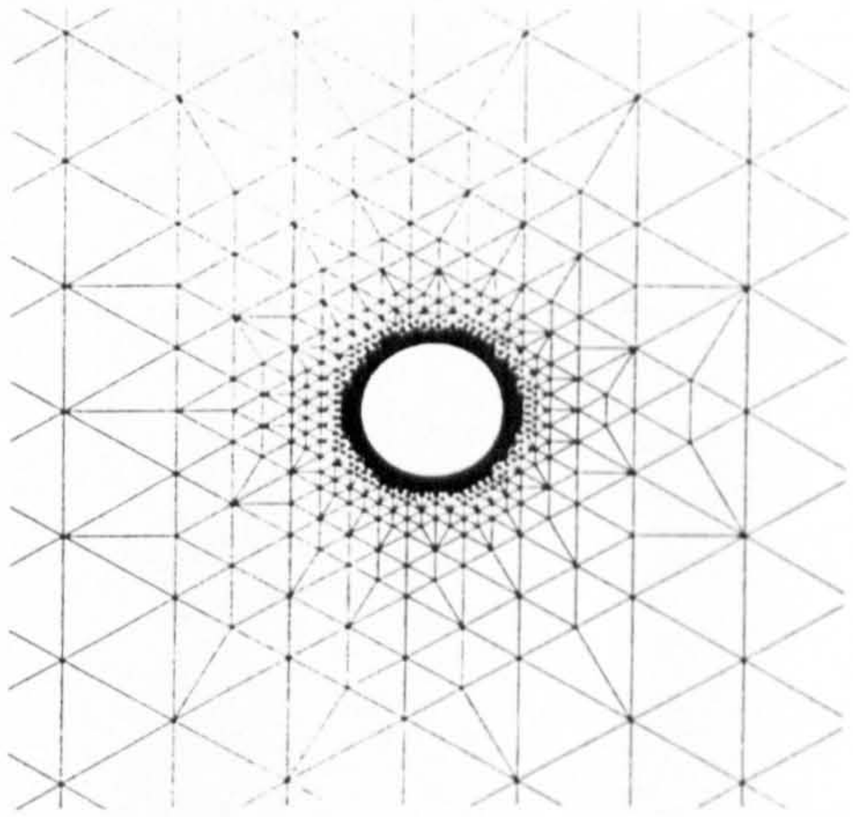


Figure 7.7 Detail at the cylinder of base grid, $level_{max}$=13 and $level_{min}$=6

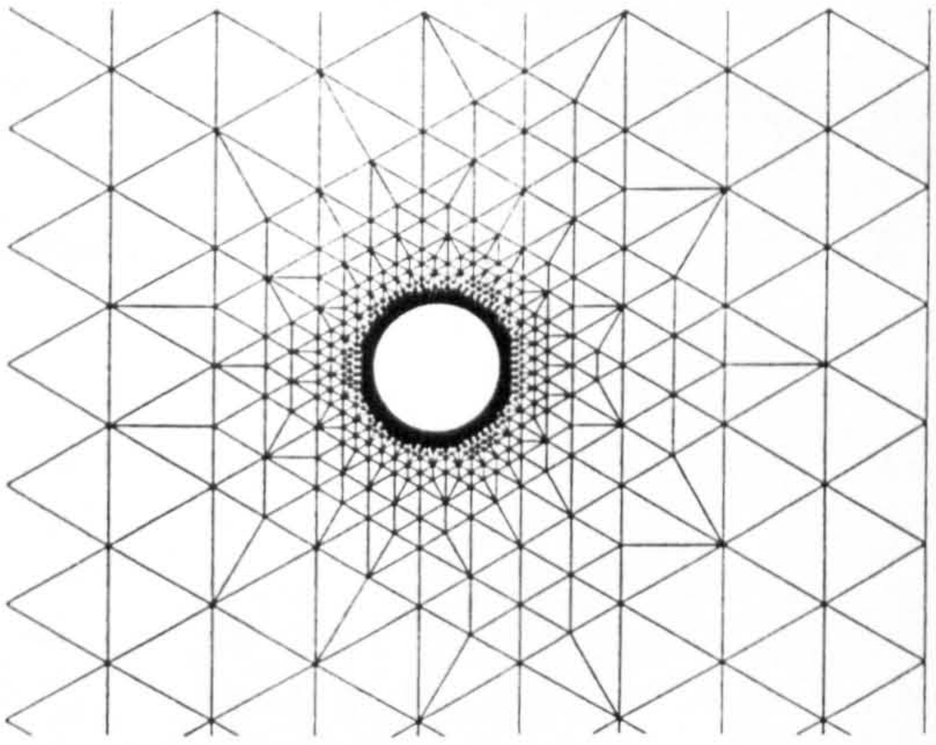


Figure 7.8 Detail at the cylinder of base grid, $level_{max}$=12 and $level_{min}$=6

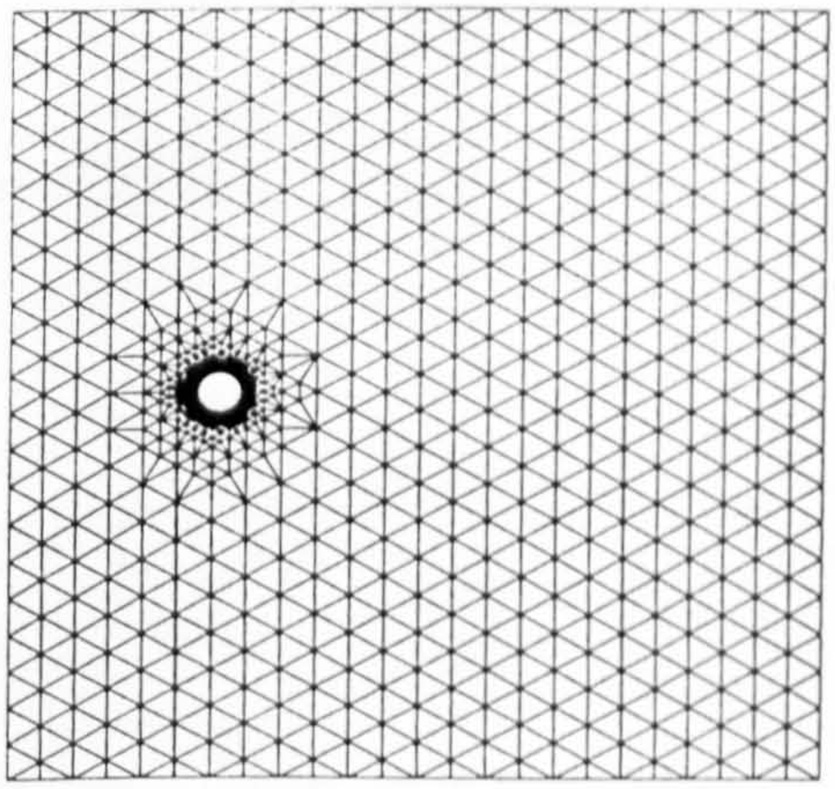
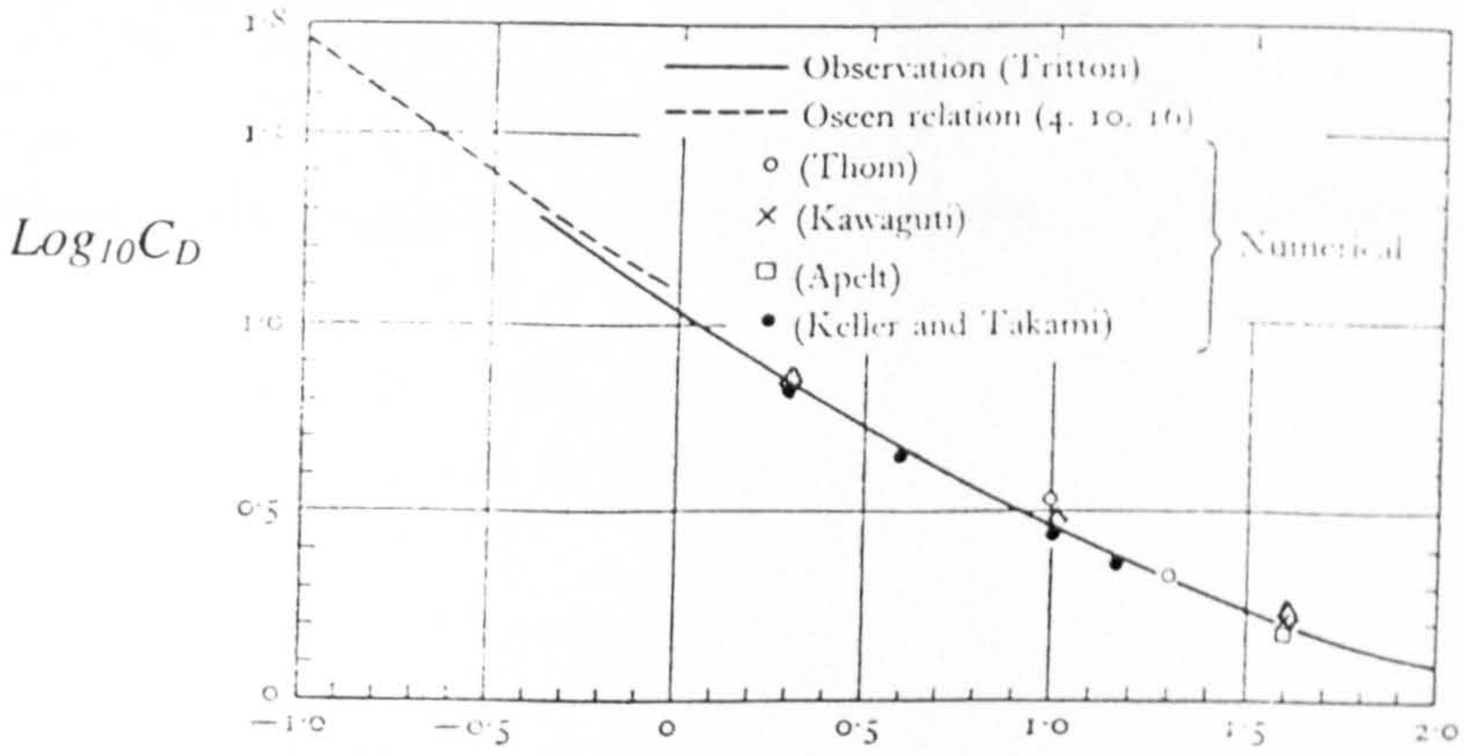


Figure 7.9 Base grid with cells=3957, nodes=2142, $level_{max}$= 12 and $level_{min}$=6



◊ Present results, numerical data

Figure 7.10 Drag coefficient exerted on a circular cylinder

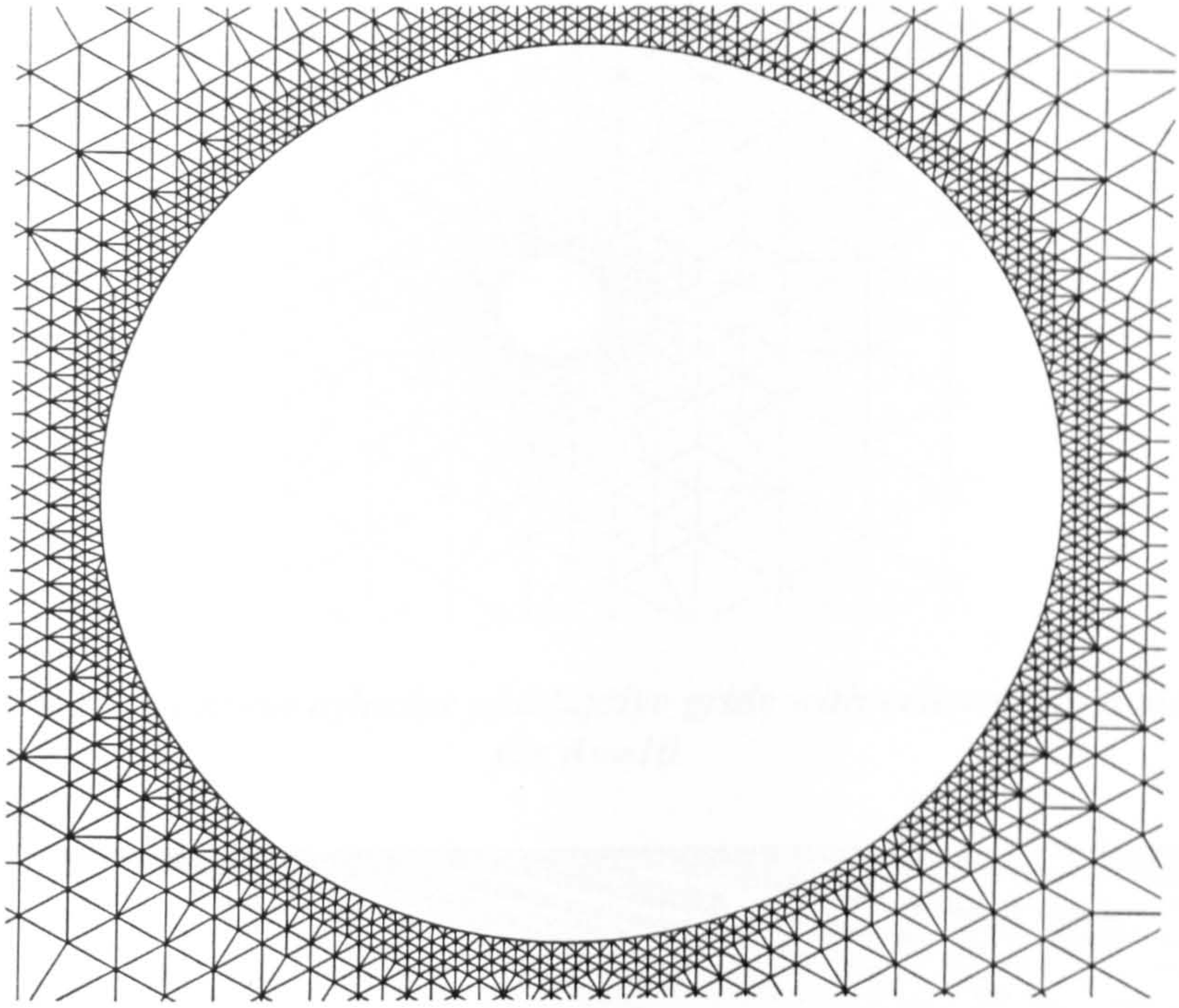


Figure 7.11 Base grid in the vicinity of the cylinder $level_{max}$= 12 and $level_{min}$=6

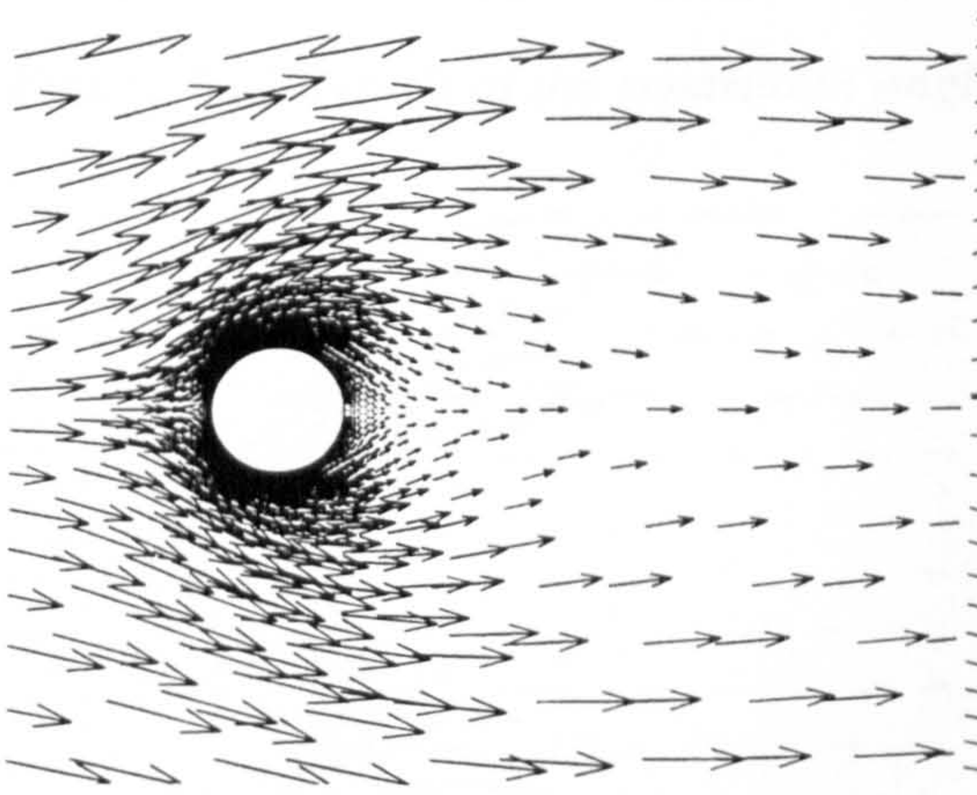


Figure 7.12 Velocity vectors Re=2.04

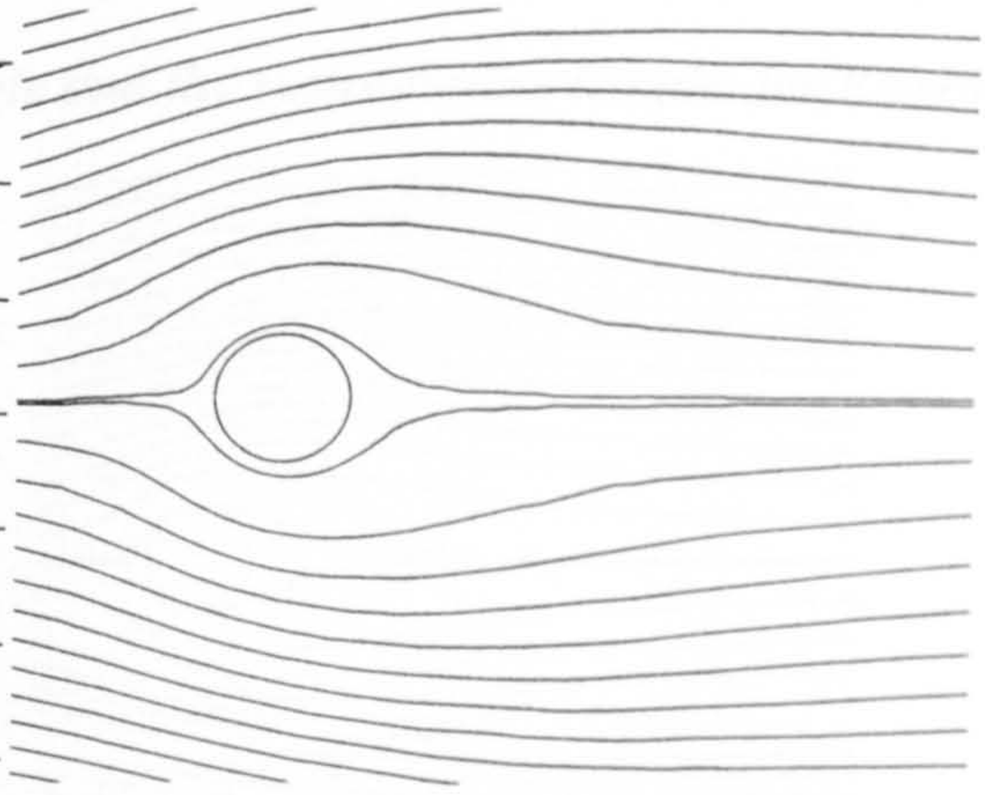


Figure 7.13 Streamlines for Re=2.04

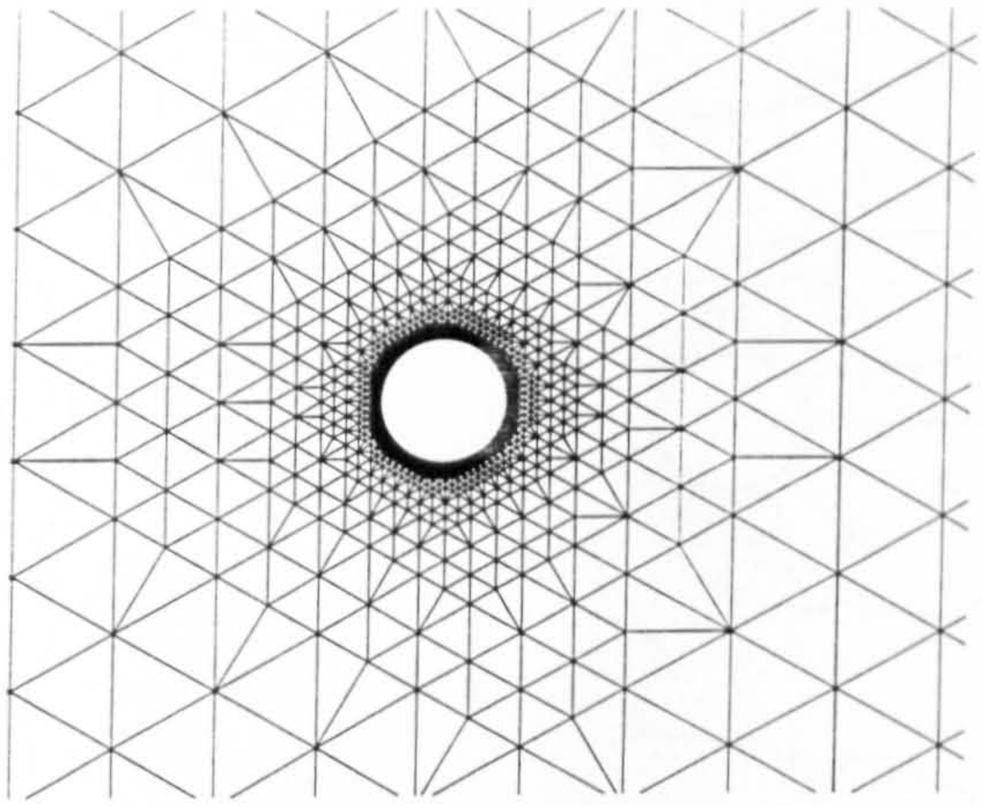


Figure 7.14 Detail at the cylinder of adaptive grids with cells=4125, nodes=2226 for Re=10

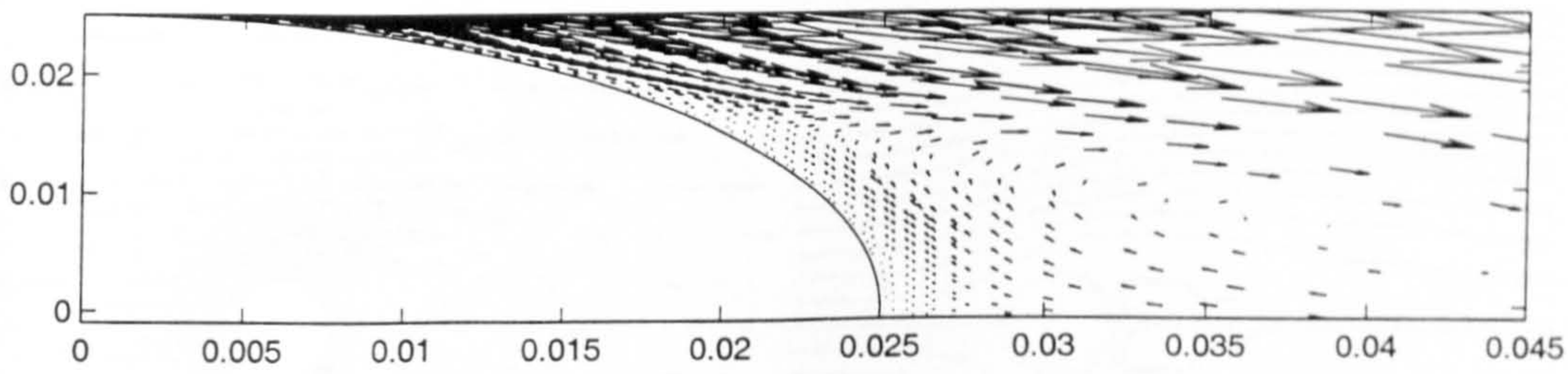


Figure 7.15 Details of the separation angle and the recirculation lengths for Re=10

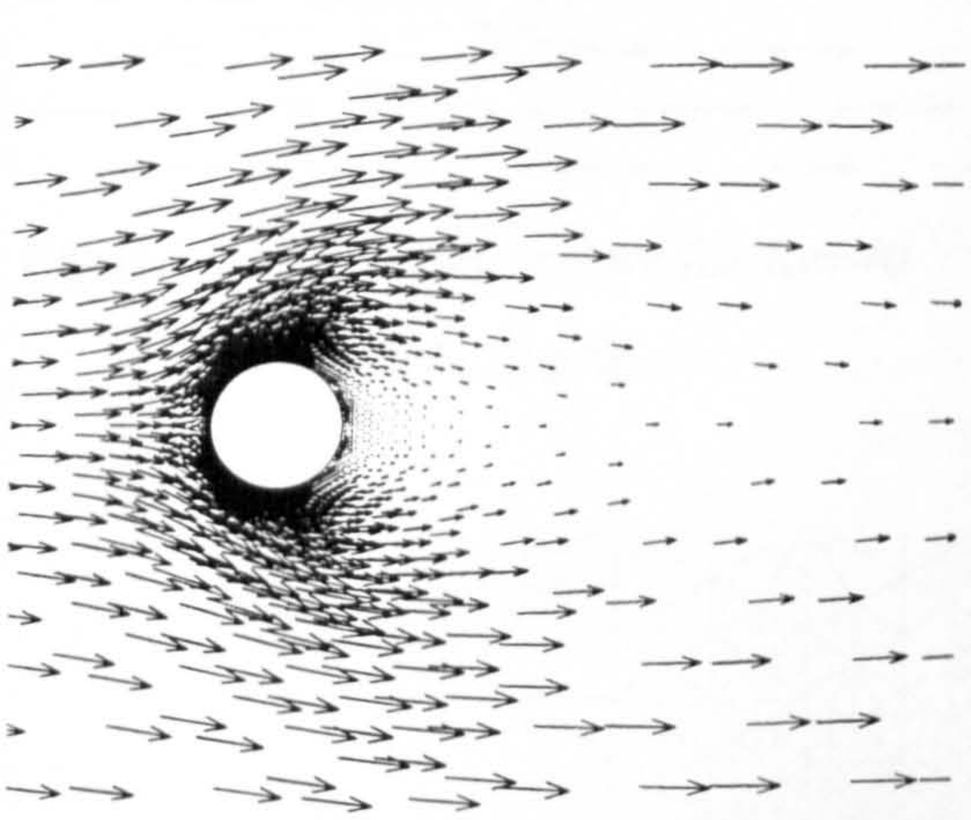


Figure 7.16 Velocity vectors Re=10

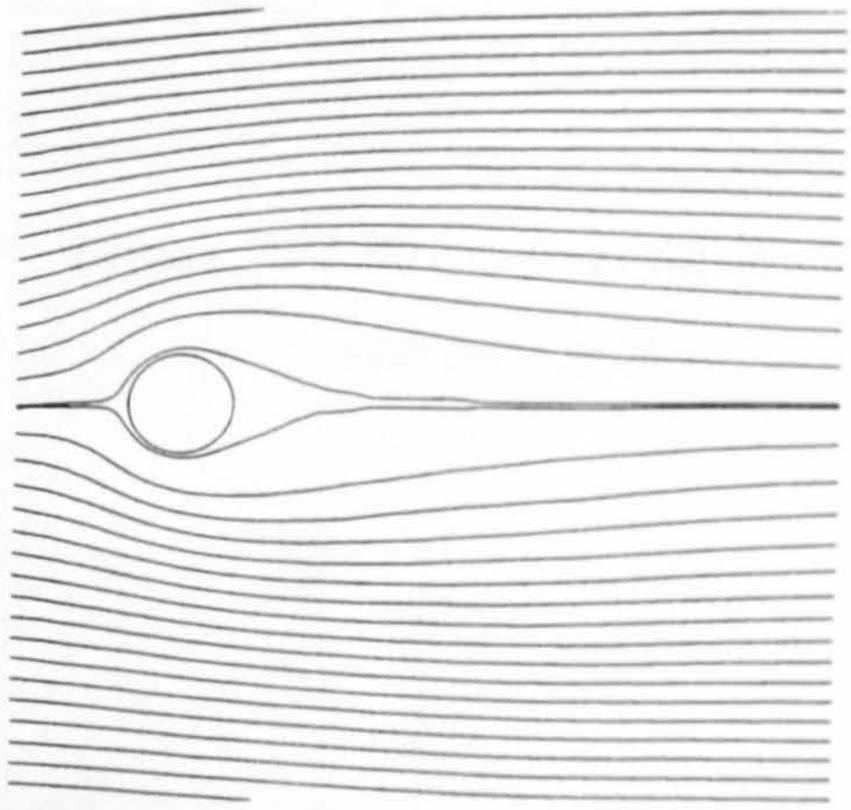


Figure 7.17 Streamlines for Re=10

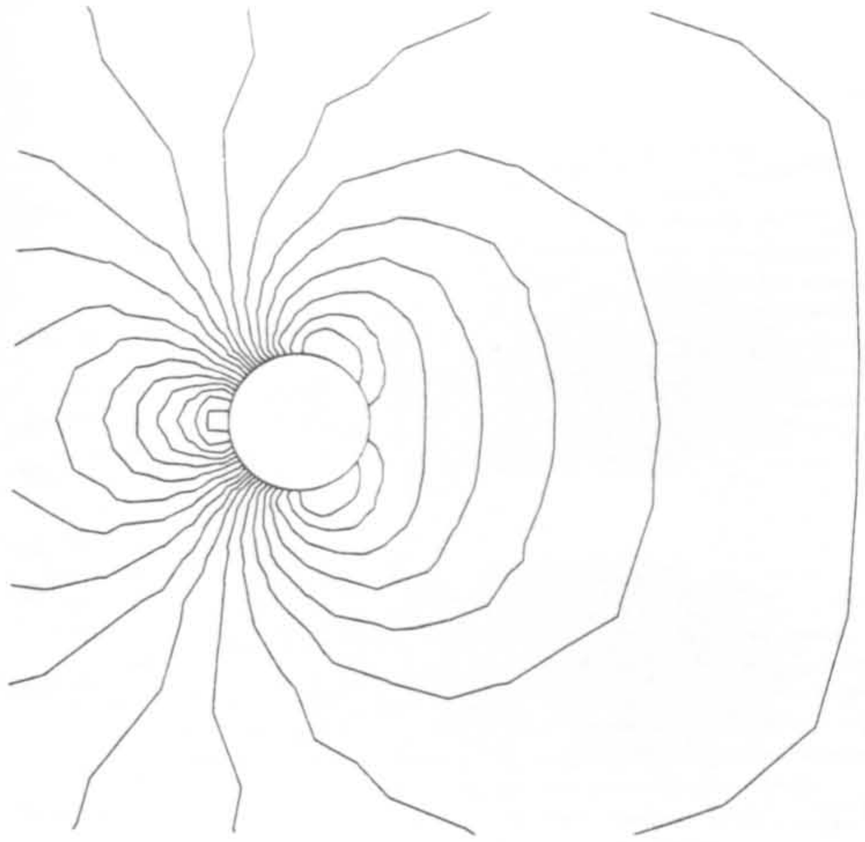


Figure 7.18 Pressure contours for Re=10

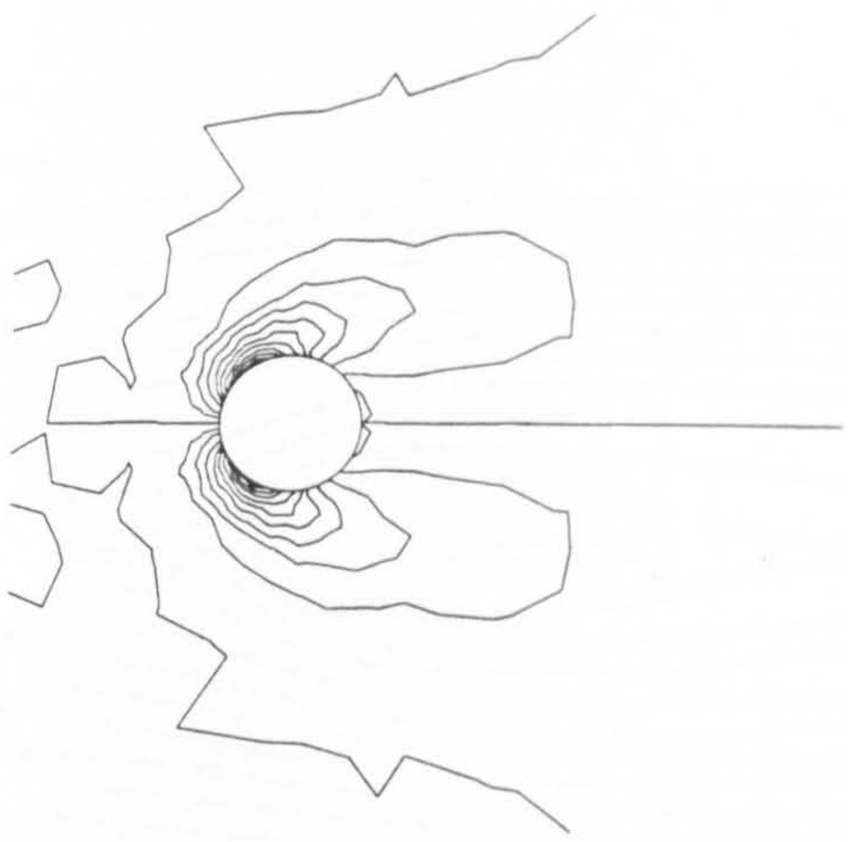


Figure 7.19 Vorticity contours for Re=10

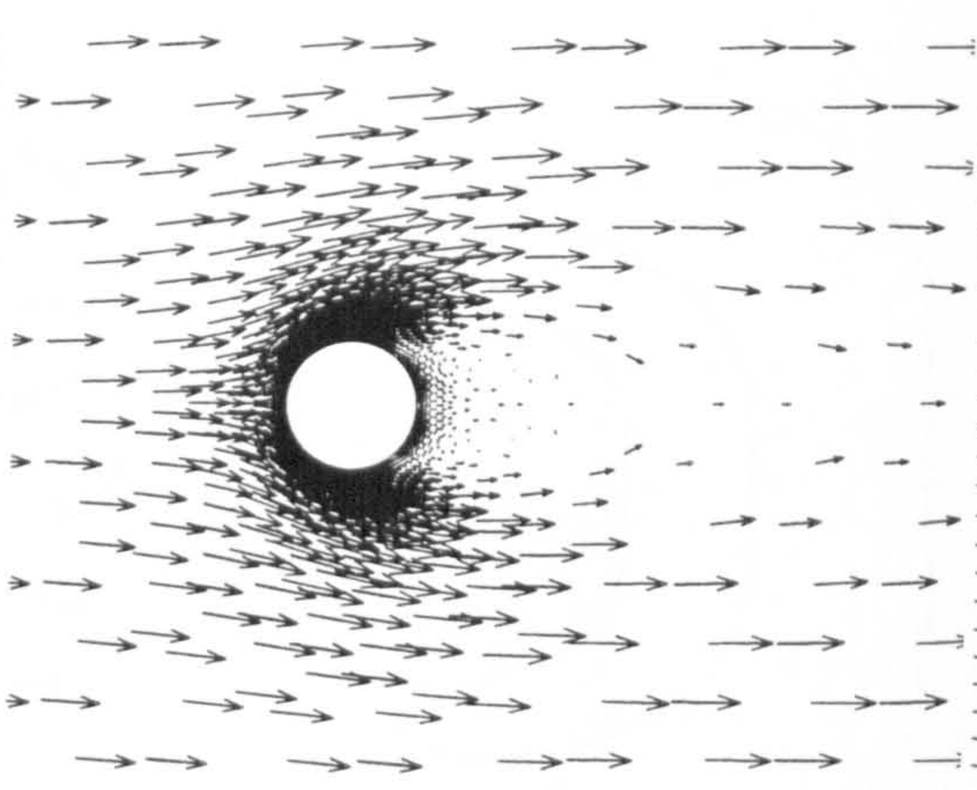


Figure 7.20 Velocity vectors for Re=40 with base grid

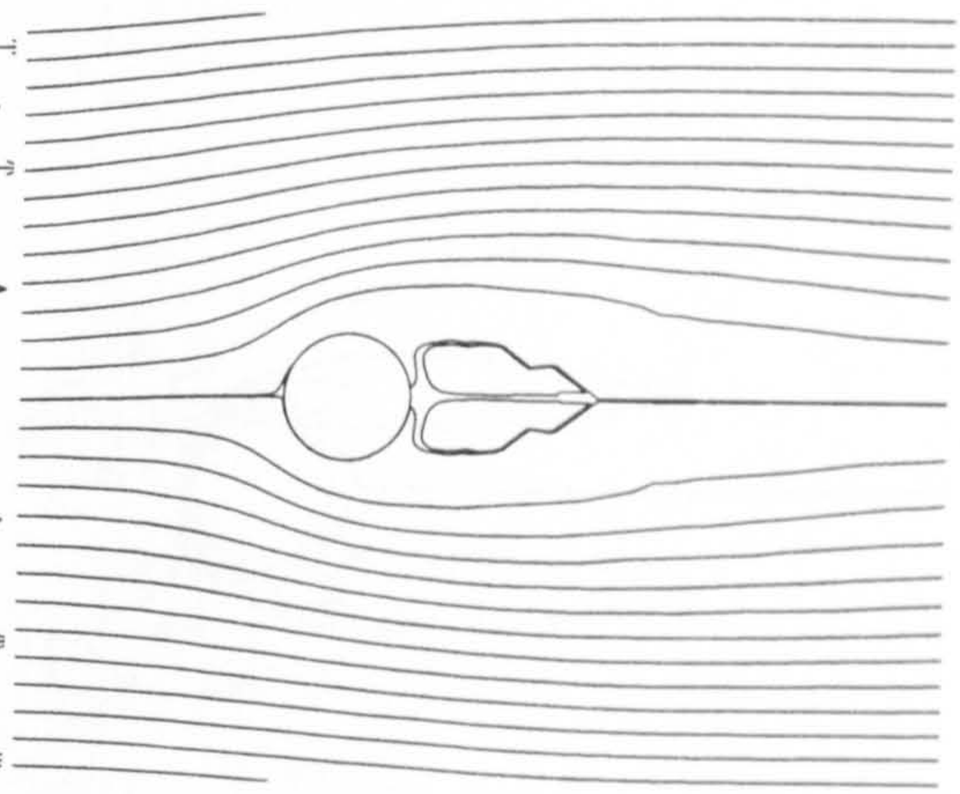


Figure 7.21 Streamlines for Re=40 with base grid

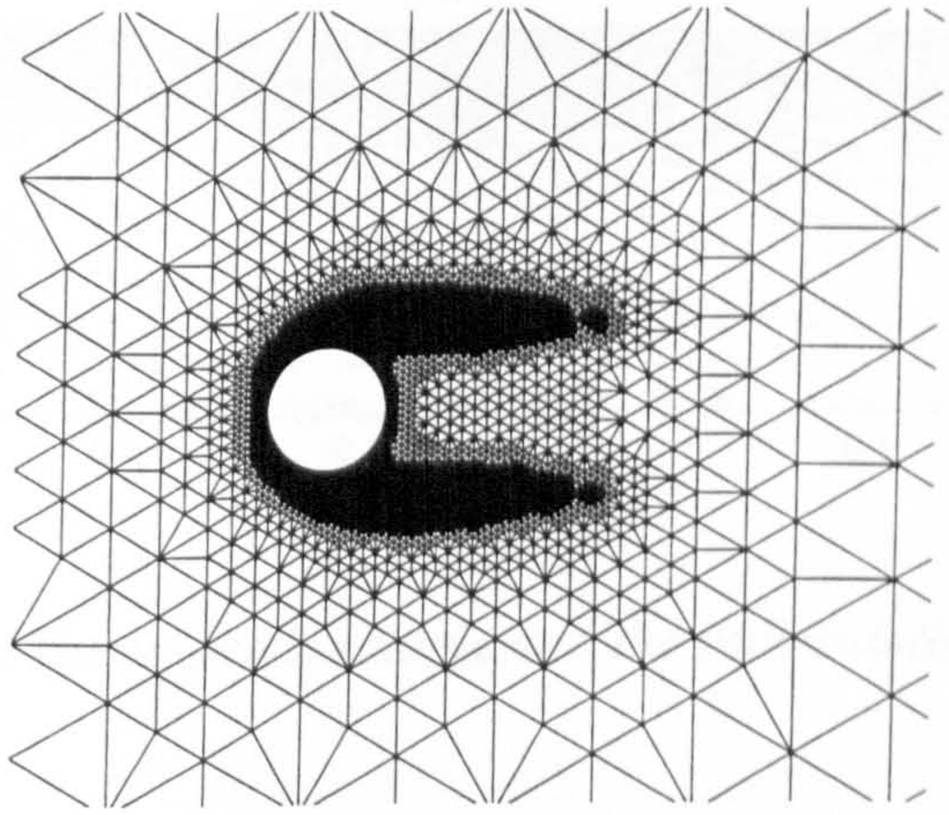


Figure 7.22 Detail of adapted grid for Re=40 of steady flow simulation (cells=10465, nodes=5396)

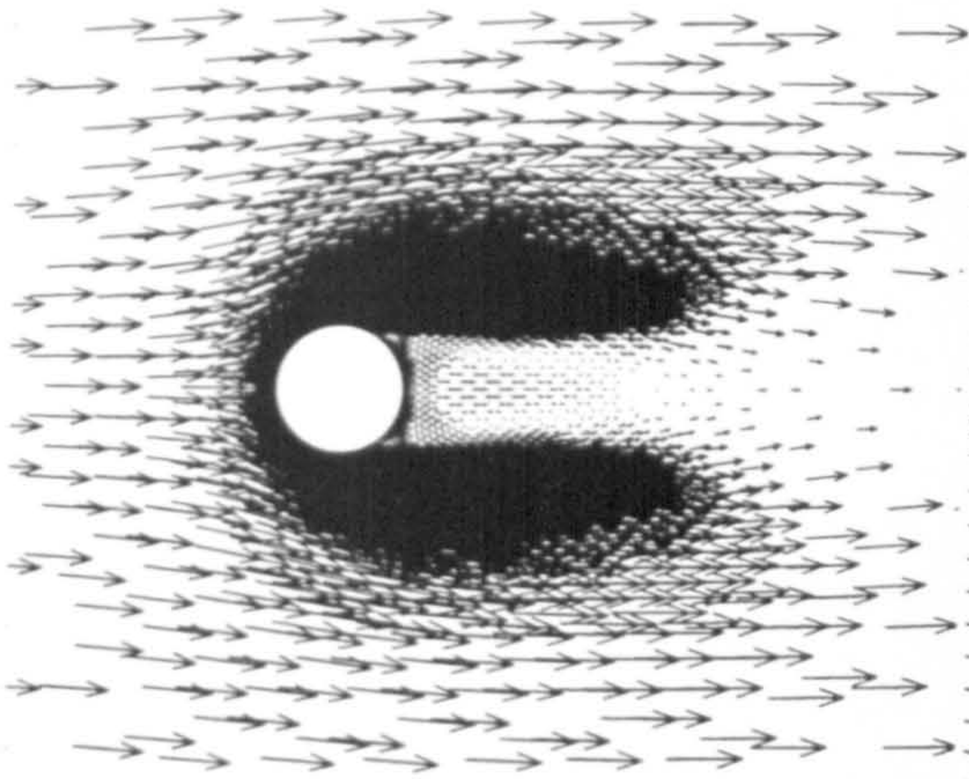


Figure 7.23 Velocity vectors for Re=40

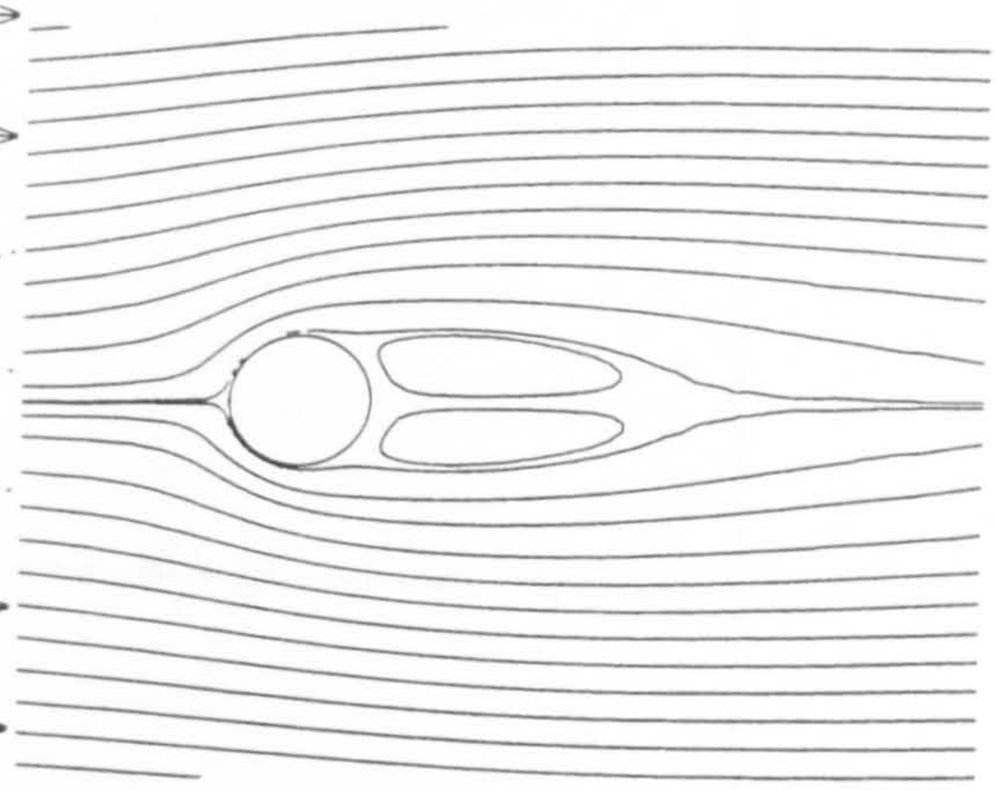


Figure 7.24 Streamlines for Re=40

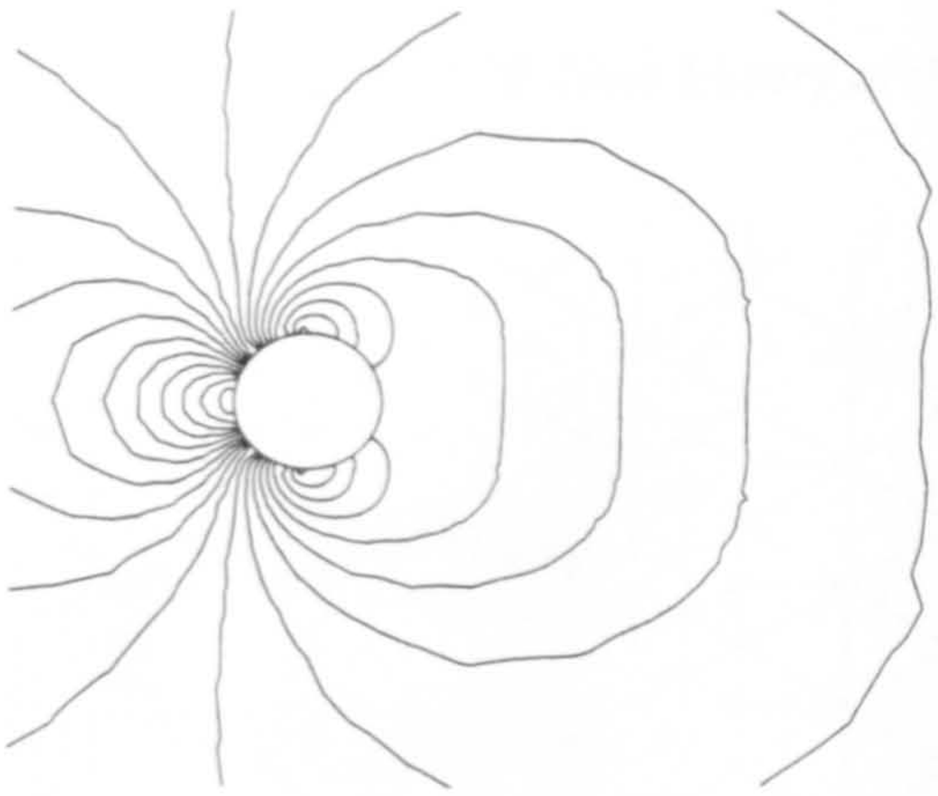


Figure 7.25 Pressure contours for Re=40

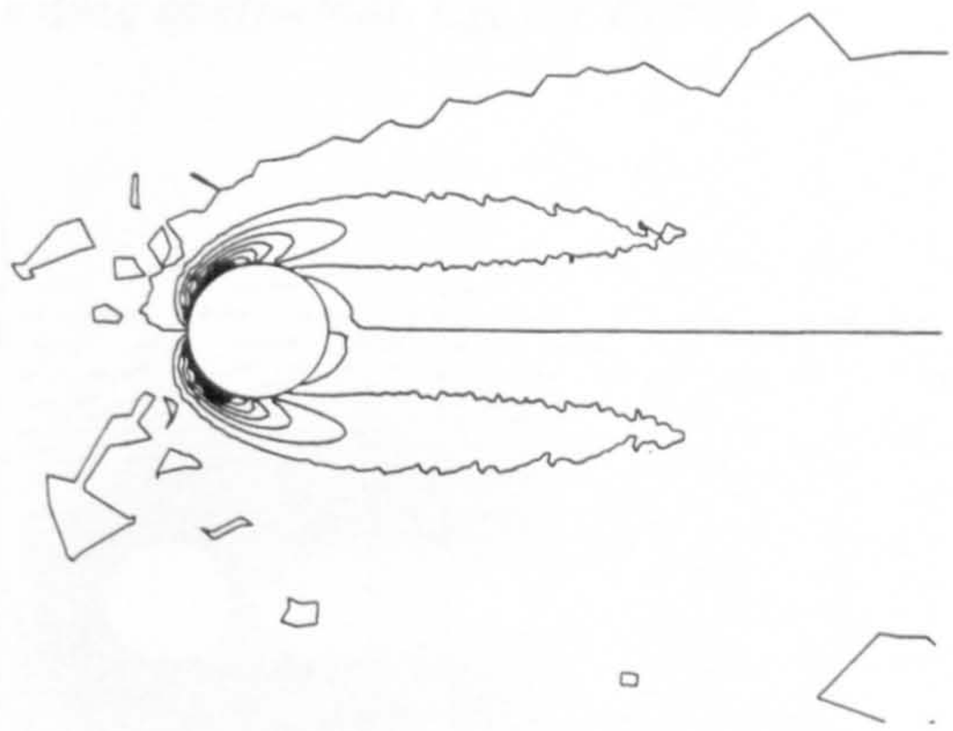


Figure 7.26 Vorticity contours for Re=40

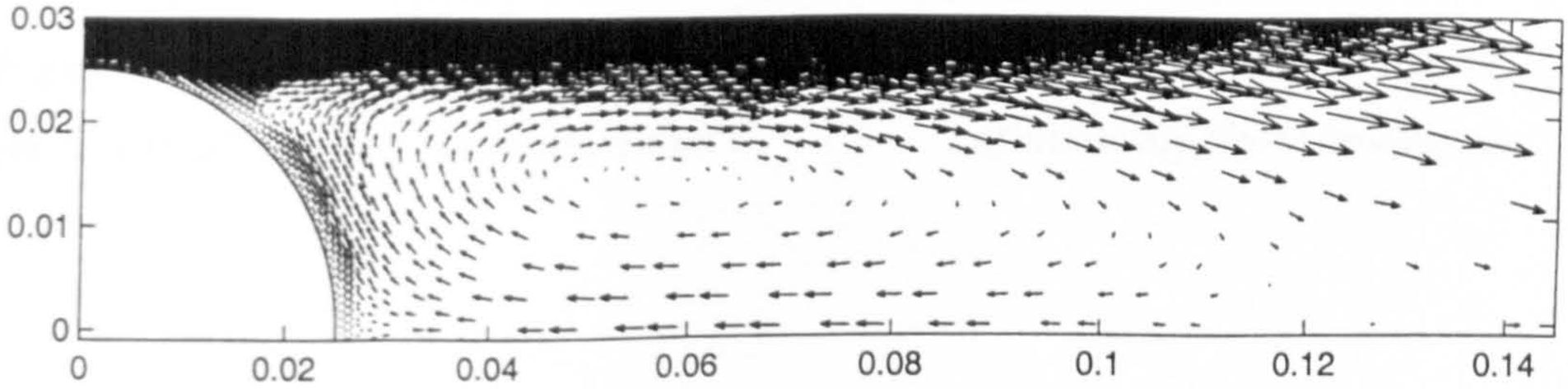


Figure 7.27 Details of the separation angle and the recirculation lengths for Re=40

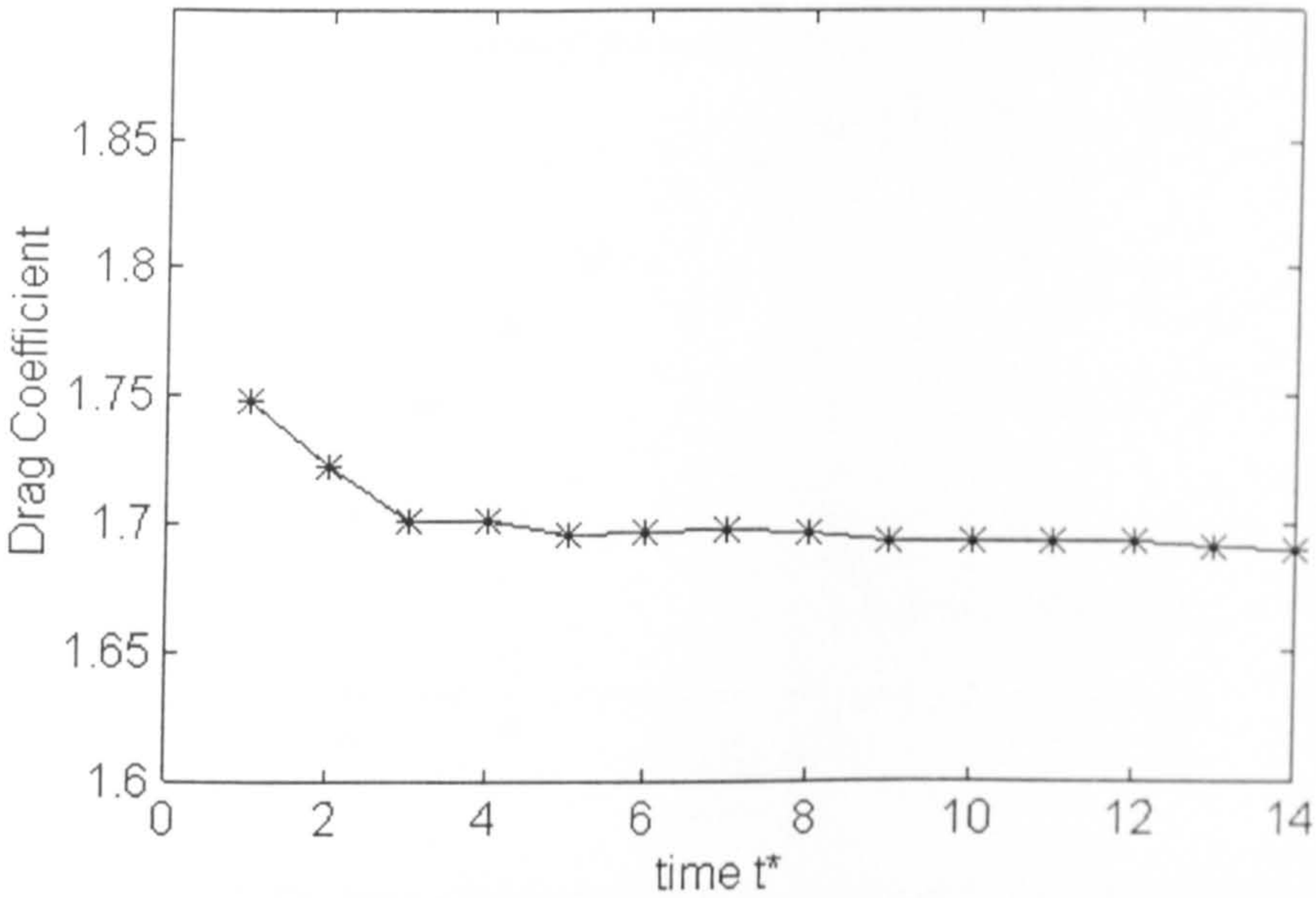


Figure 7.28 Time history of the drag coefficient, C_D, for Re=40

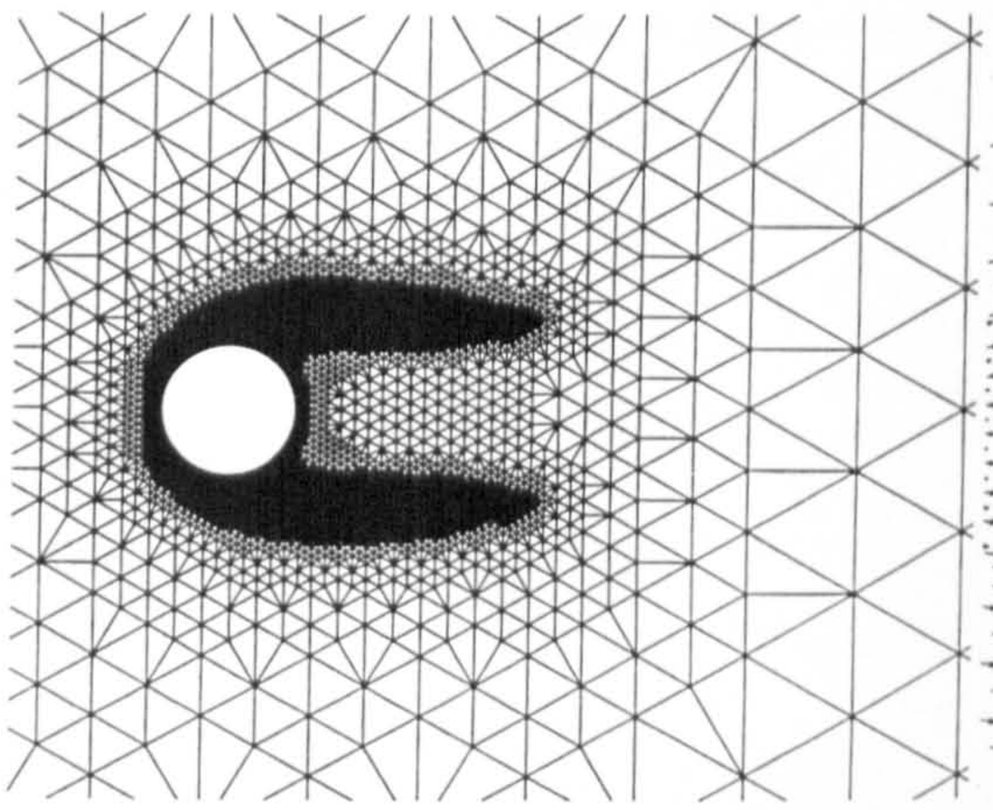


Figure 7.29 Detail of adapted grid for Re=40 of unsteady flow simulation

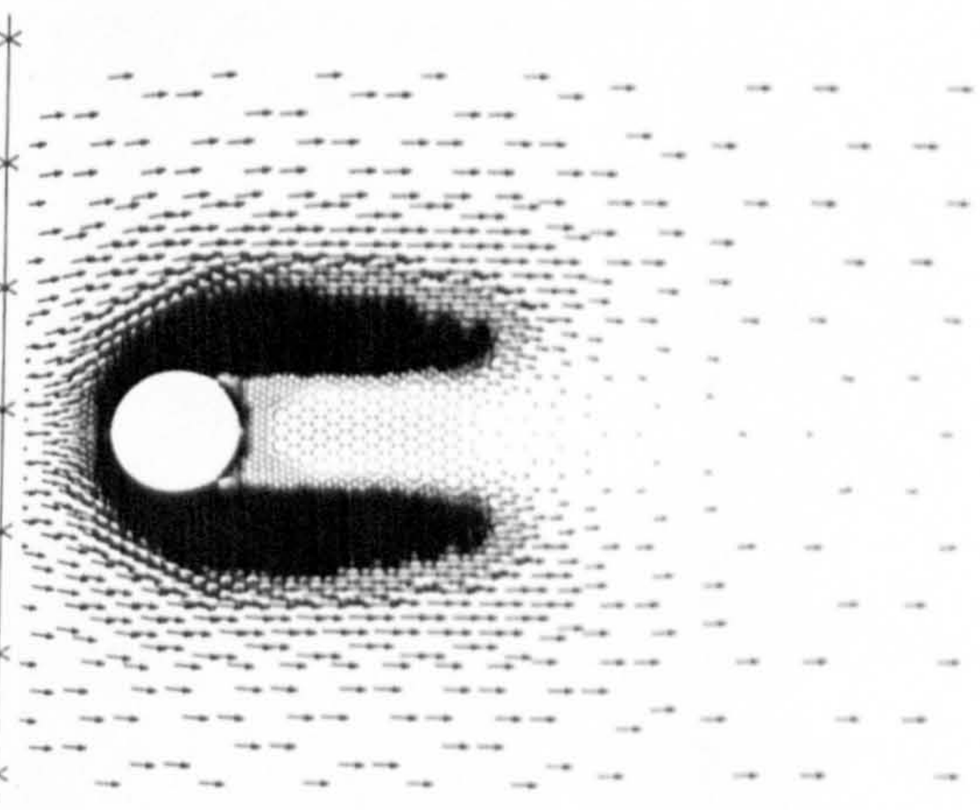


Figure 7.30 Velocity vectors for Re=40 of unsteady flow simulation

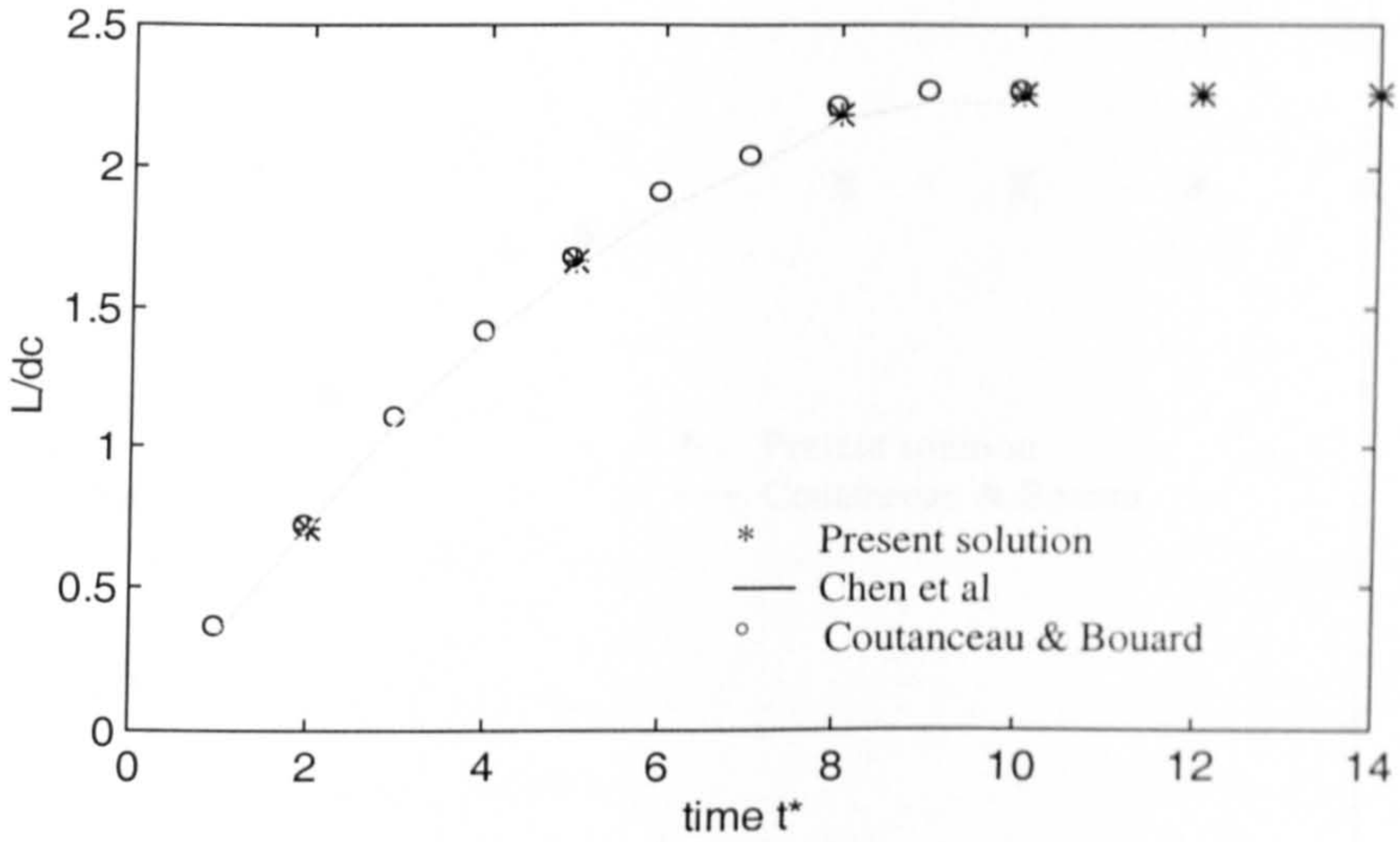


Figure 7.31 Evolution with time of the closed-wake length for Re=40

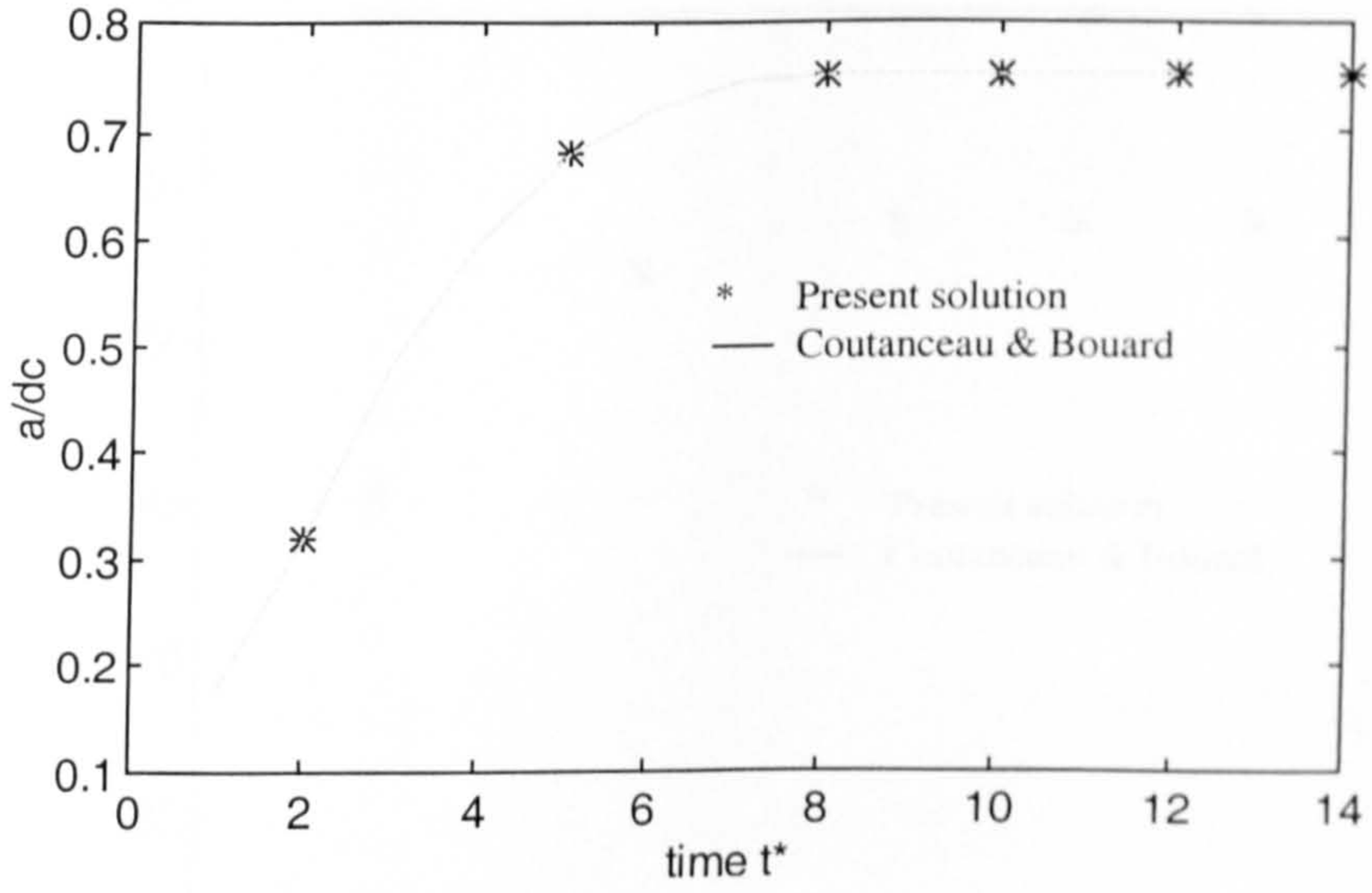


Figure 7.32 Evolution with time of the closed-wake core ordinate for Re=40

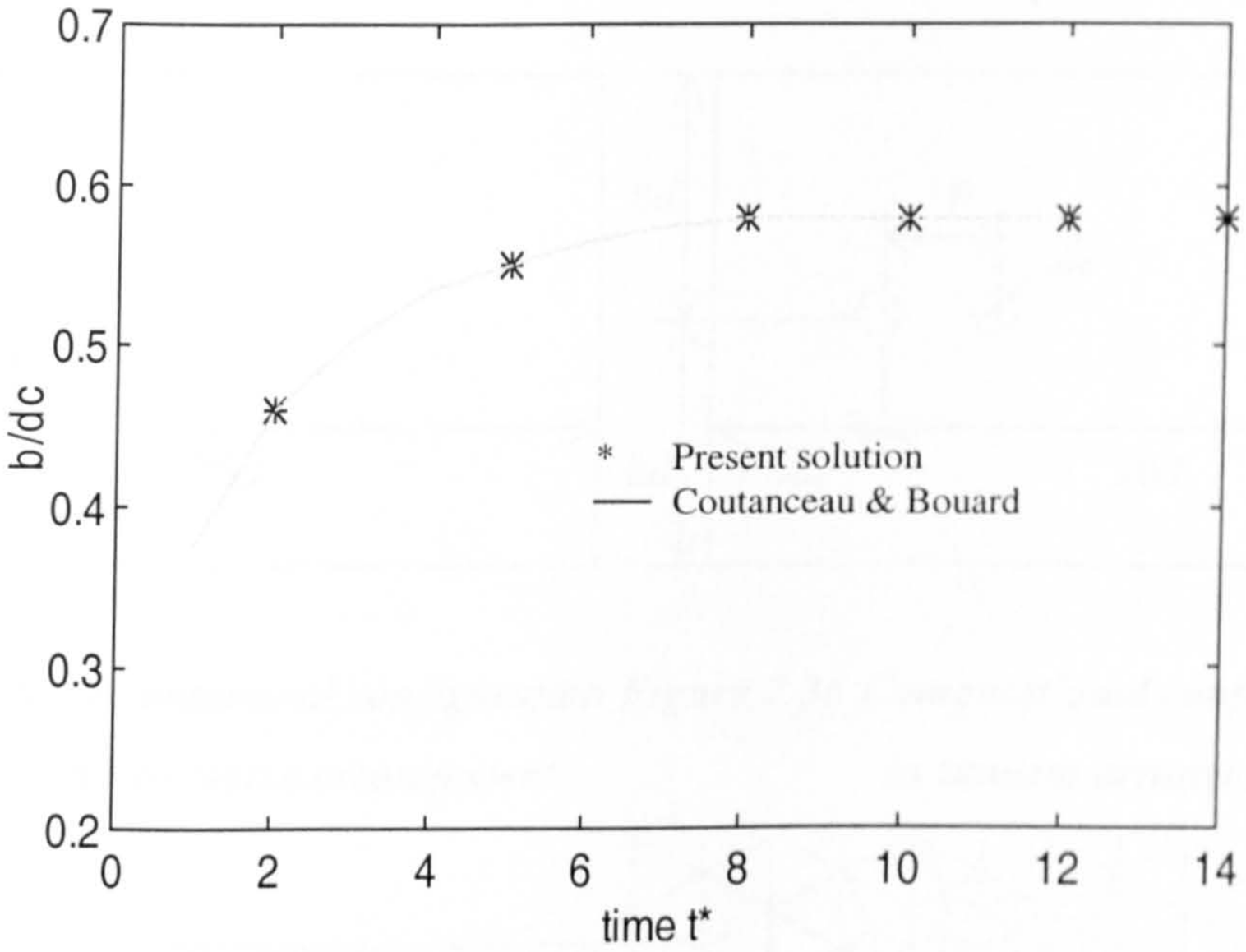


Figure 7.33 Evolution with time of the closed-wake core ordinate for Re=40

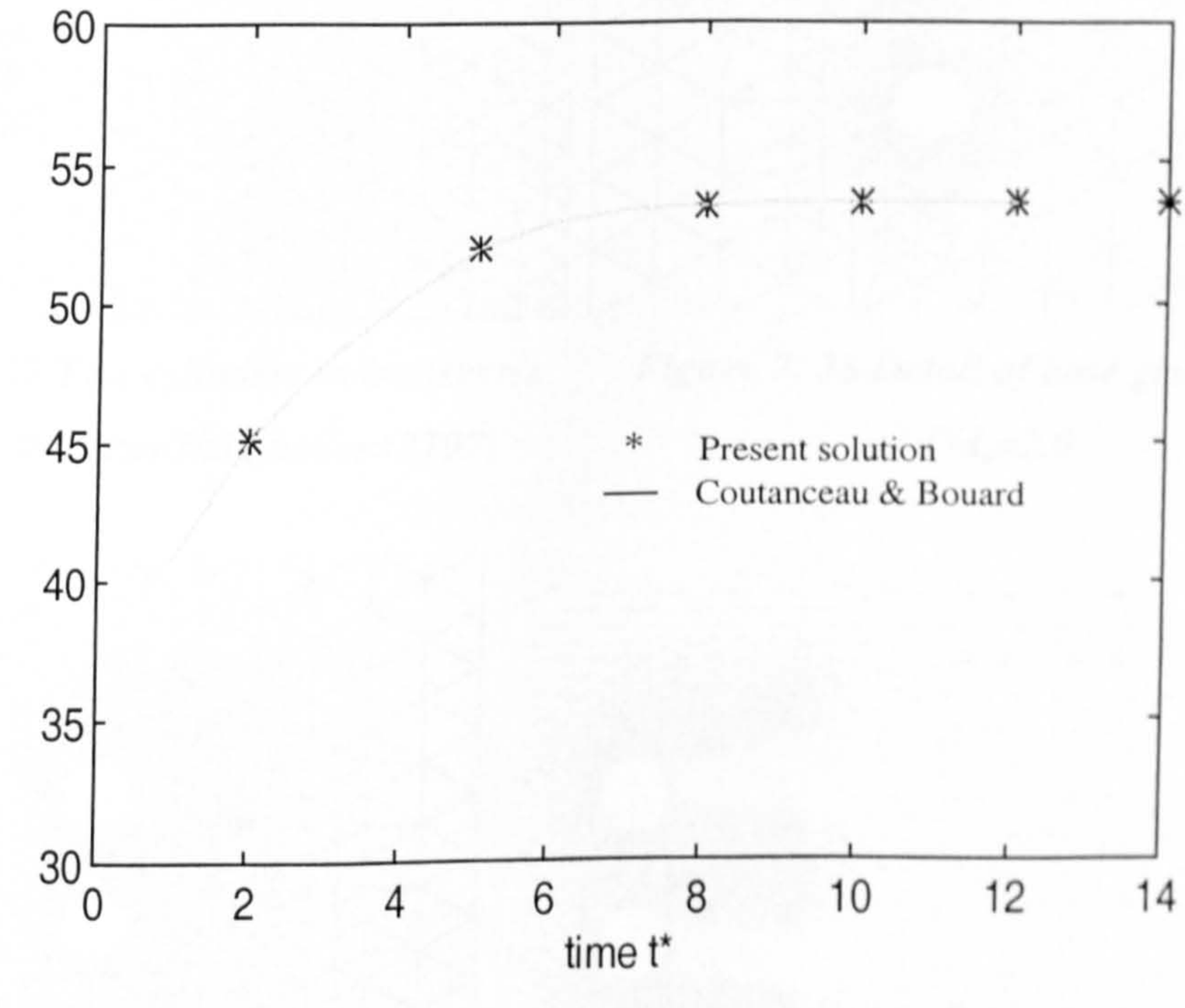


Figure 7.34 Evolution with time of the cylinder separation angle for Re =40

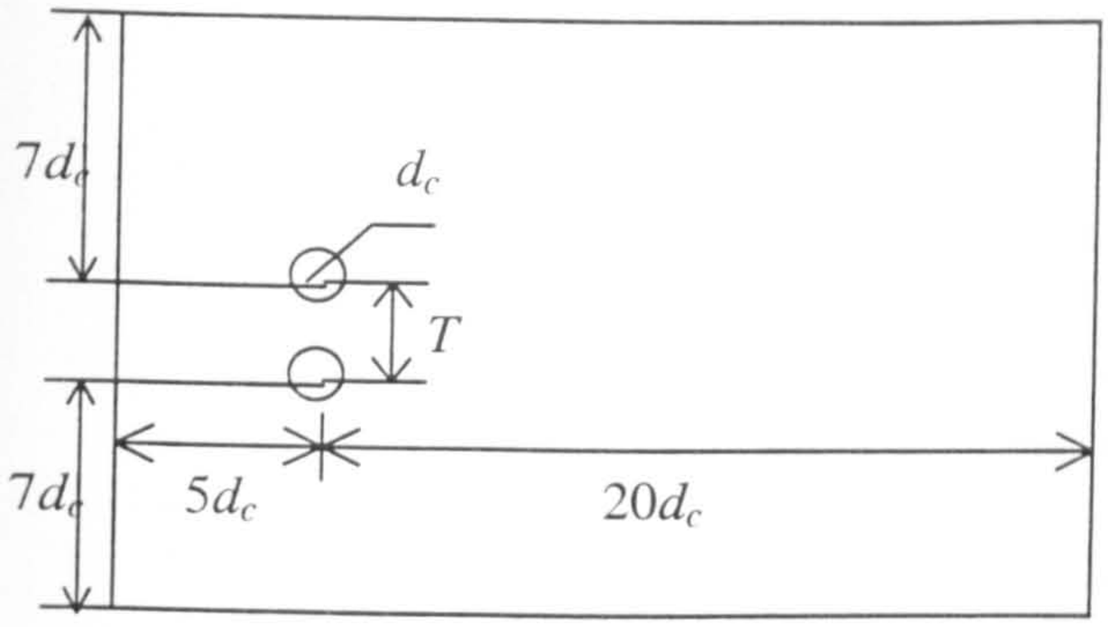


Figure7.35 Computational configuration in transverse arrangement

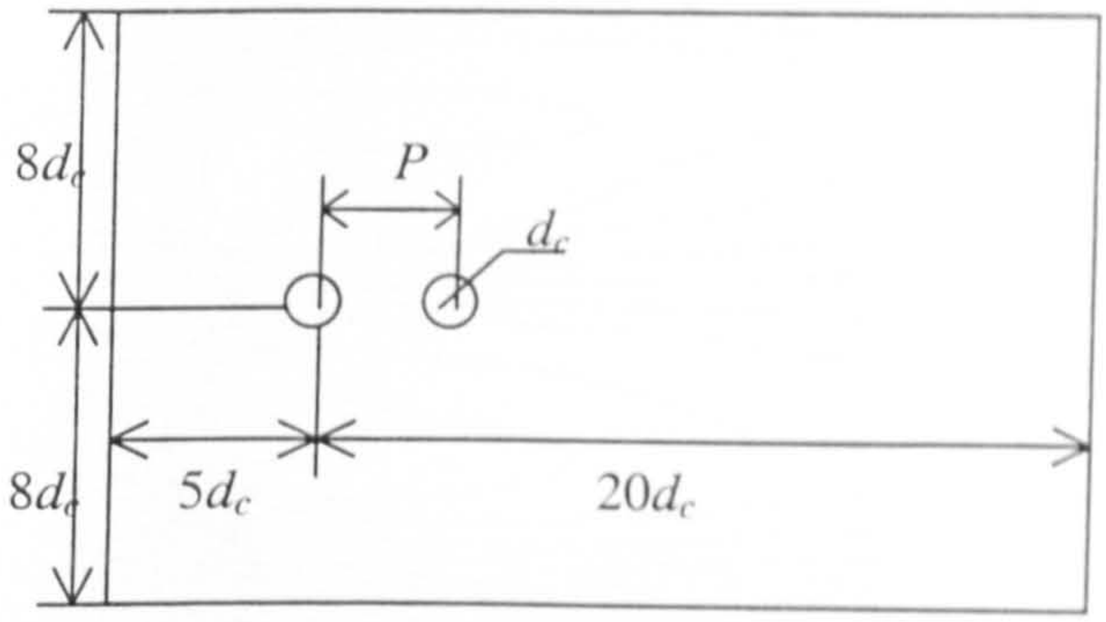


Figure 7.36 Computational configuration in tandem arrangement

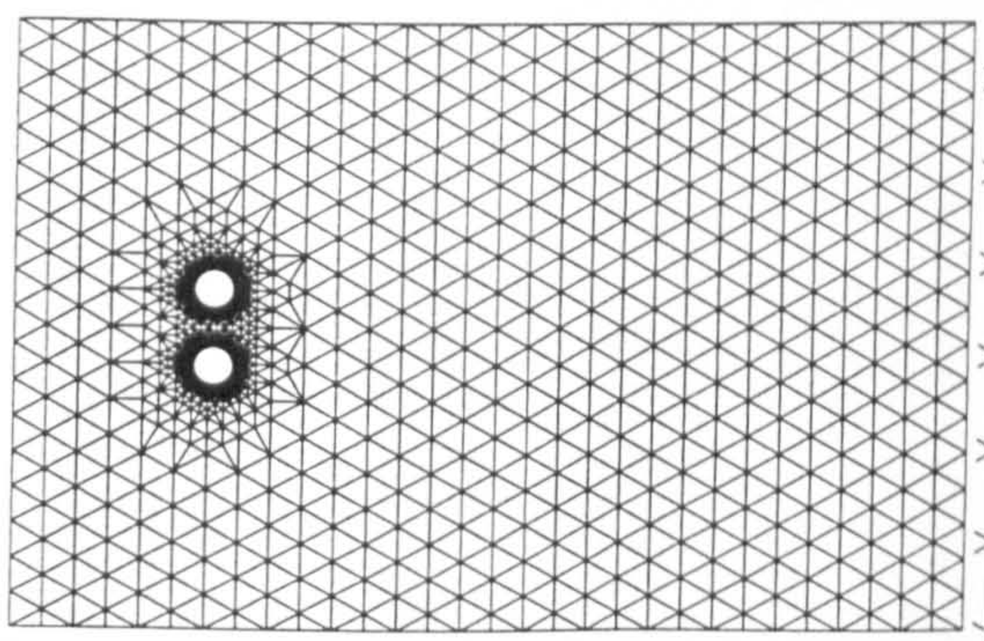


Figure 7. 37 Two cylinders in transverse at T/d_c=2.0 (cells=7034,nodes=3797)

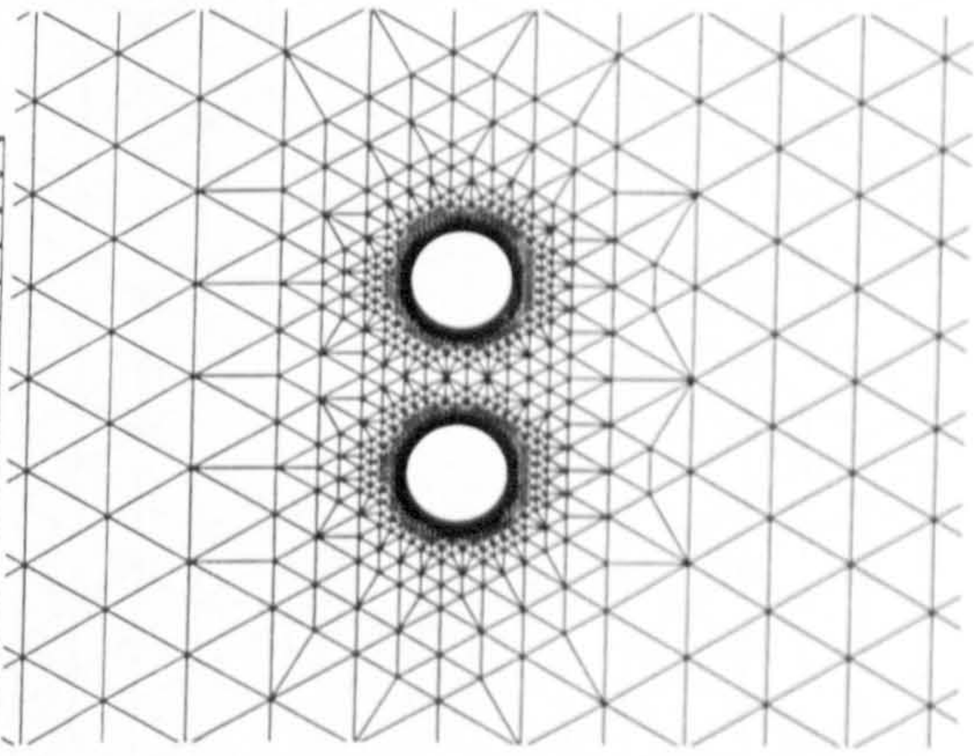


Figure 7. 38 Detail of base grid at T/d_c=2.0

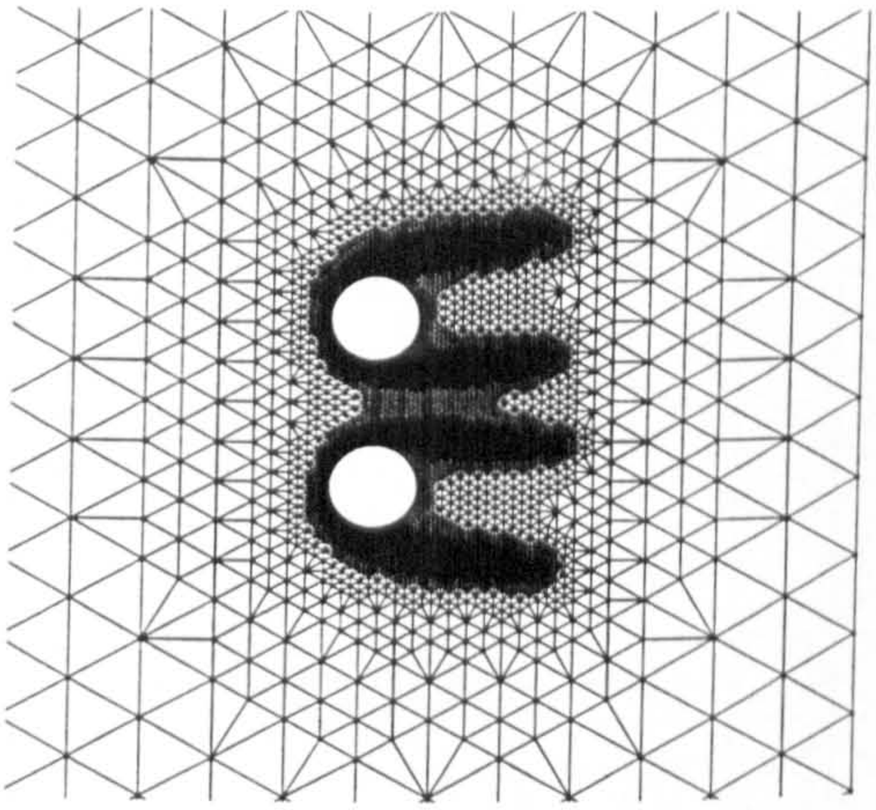


Figure 7. 39 Detail of adapted grid at T/d_c=2.0 (cells=19640,nodes=10100)

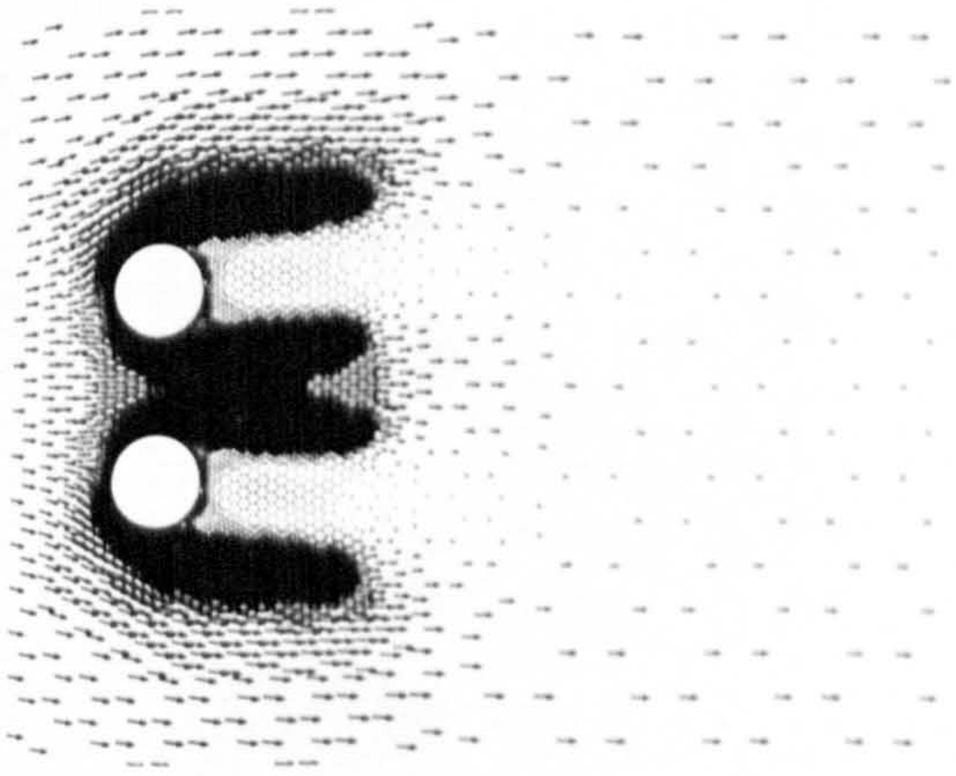


Figure 7. 40 Velocity vectors at T/d_c=2.0

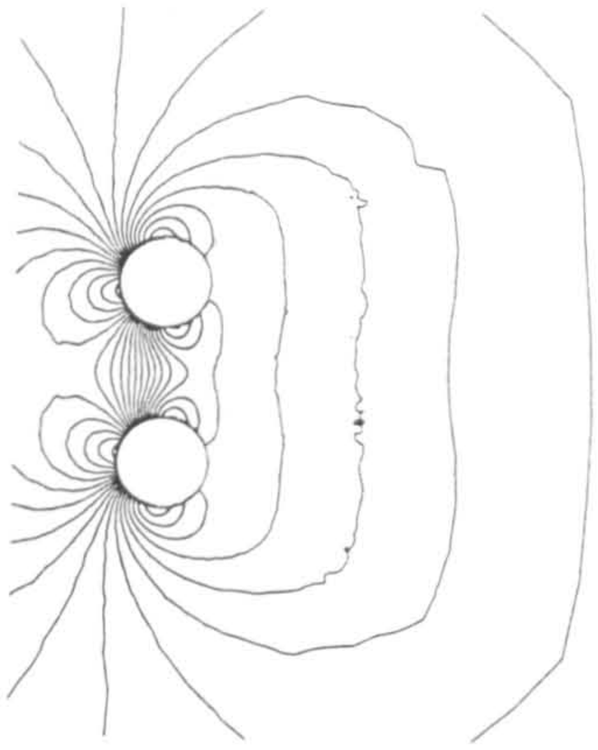


Figure 7. 41 Pressure contours at T/d_c=2.0

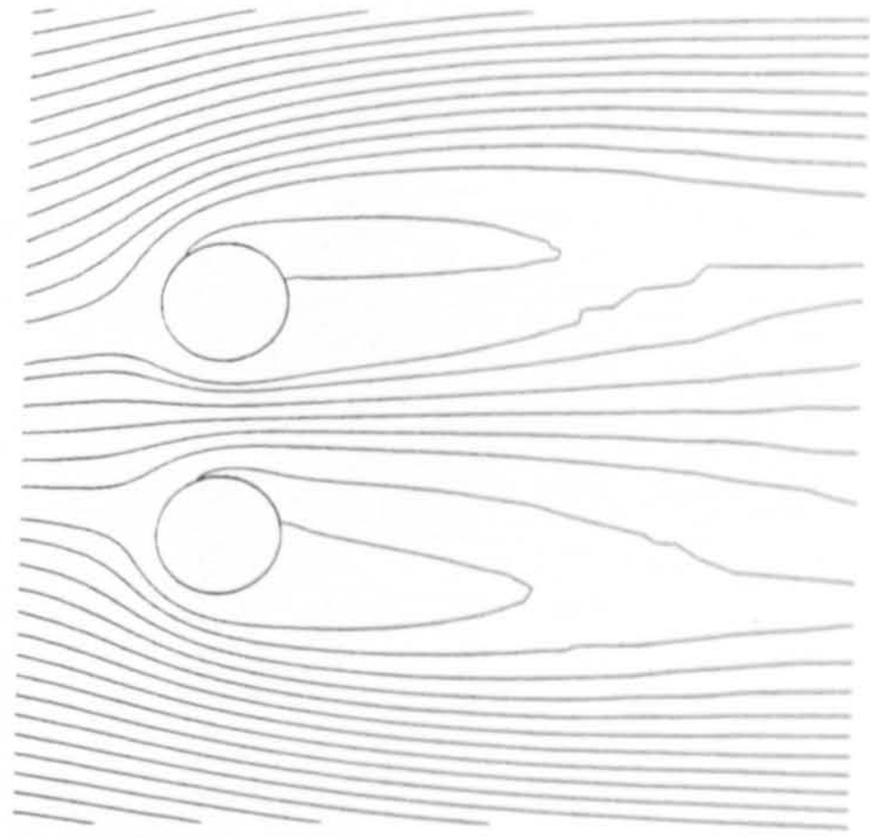


Figure 7. 42 Streamlines at T/d_c=2.0

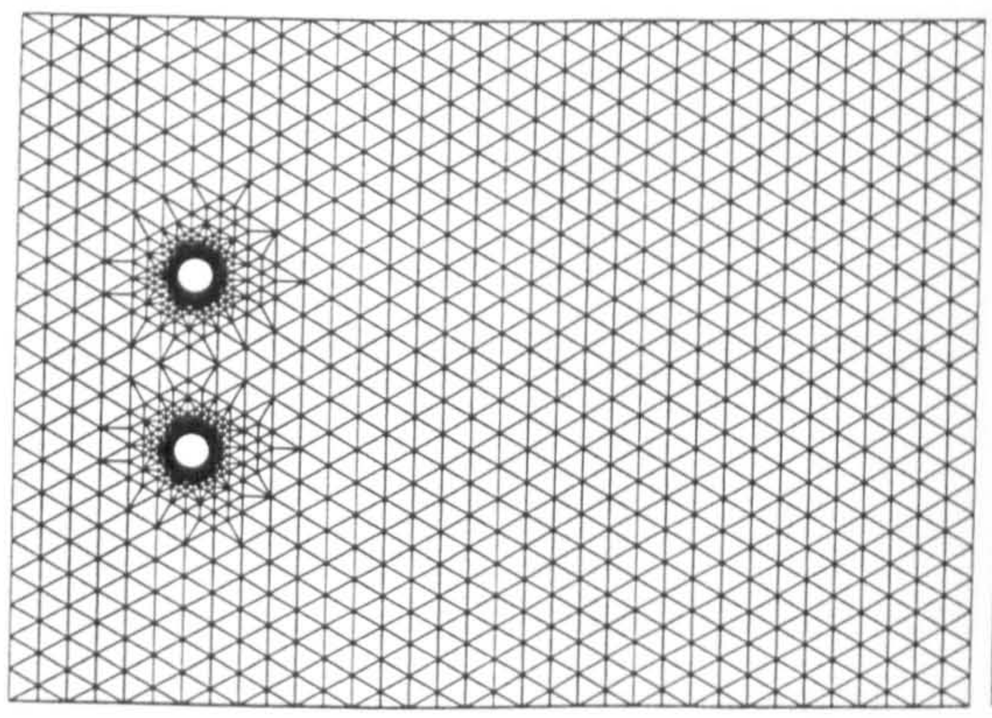


Figure 7. 43 Two cylinders in transverse at T/d_c=5.0 (cells=7466,nodes=4021)

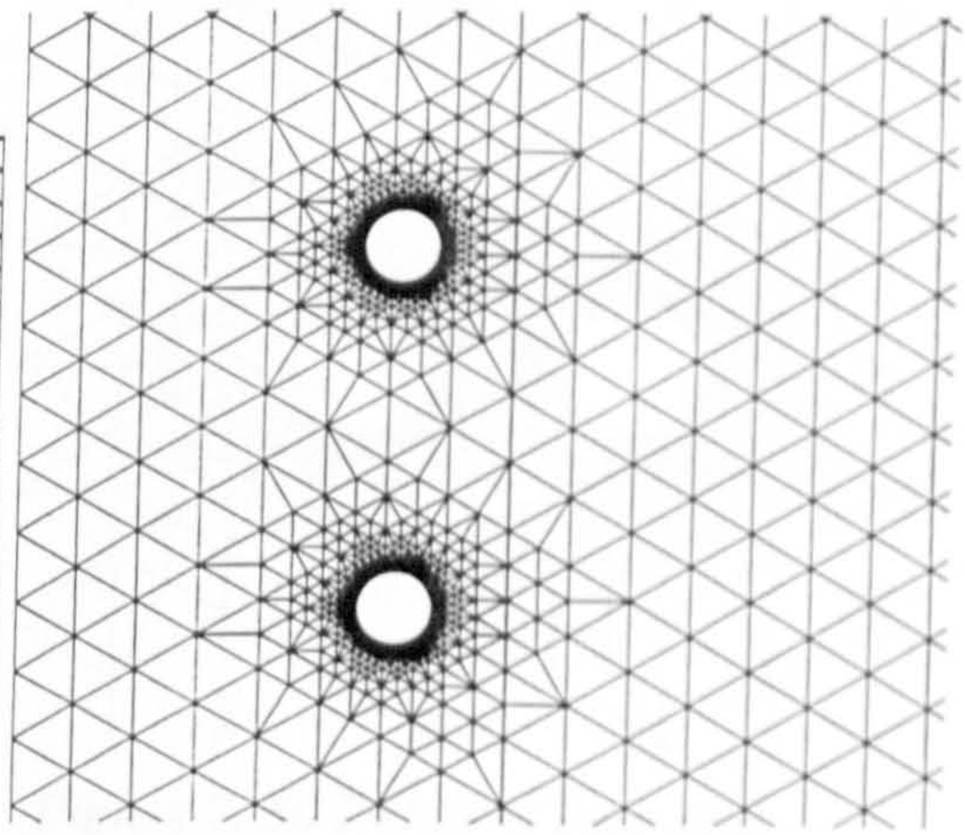


Figure 7. 44 Detail of base grid at T/d_c=5.0

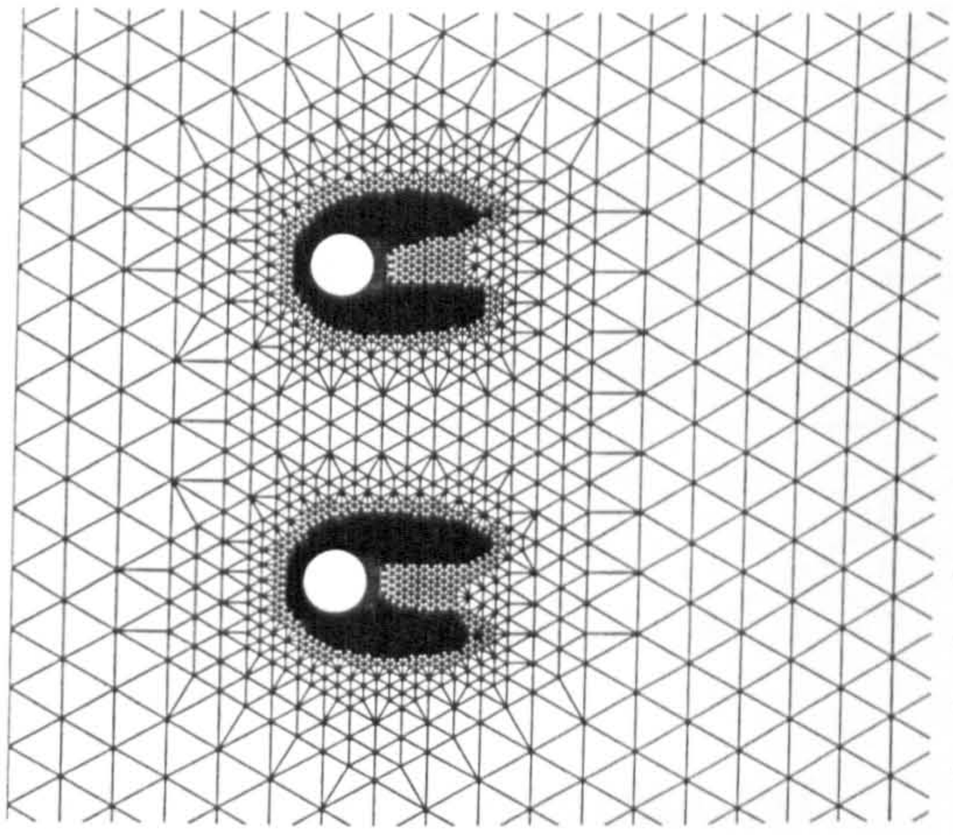


Figure 7. 45 Detail of adapted grid at T/d_c=5.0 (cells=20006,nodes=10283)

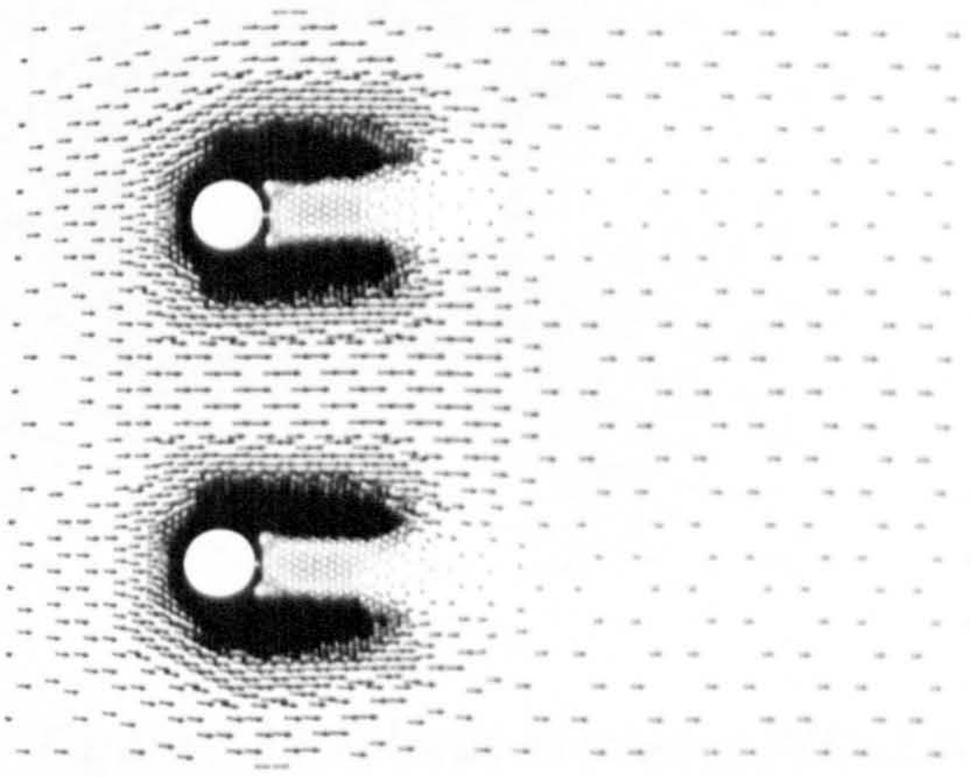


Figure 7. 46 Velocity vectors at T/d_c=5.0

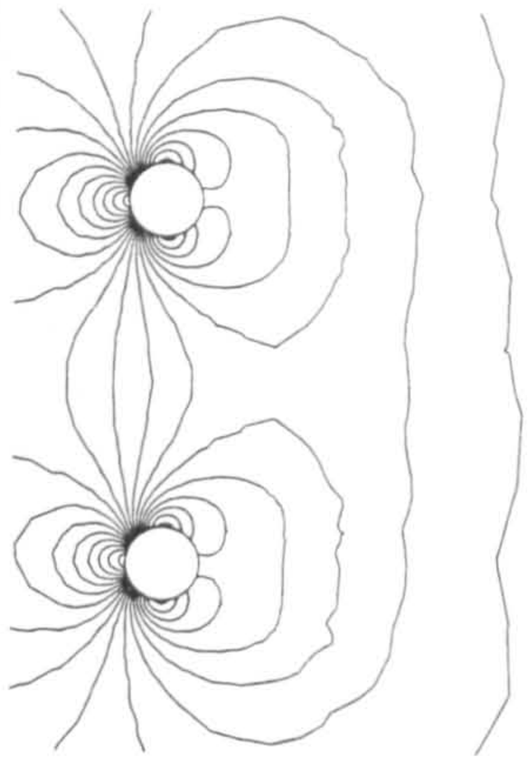


Figure 7. 47 Pressure contours at T/d_c=5.0

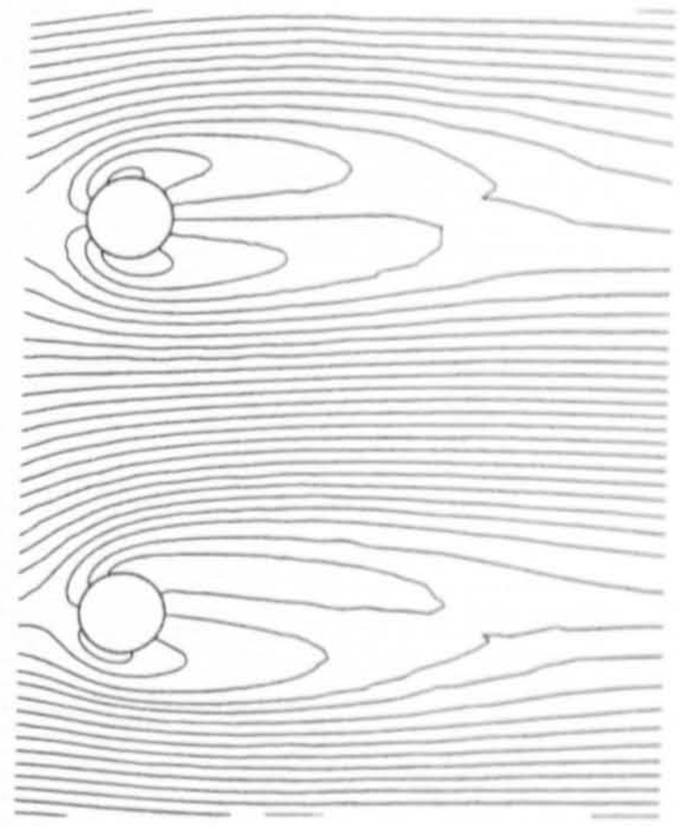


Figure 7. 48 Streamlines at T/d_c=5.0

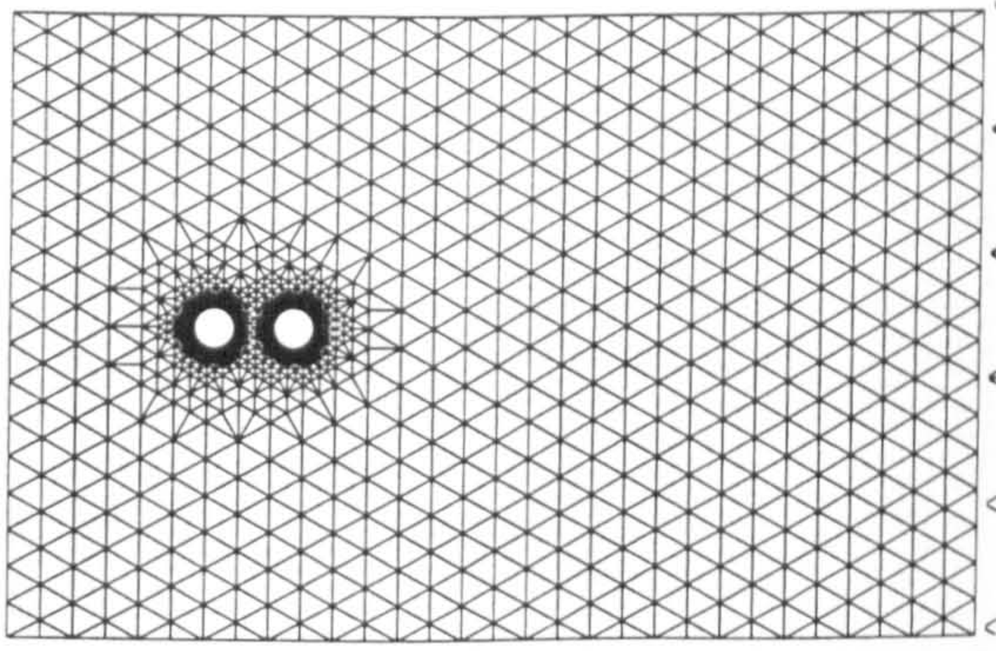


Figure 7. 49 Two cylinders in tandem at P/d_c=2.0 (cells=7027,nodes=3793)

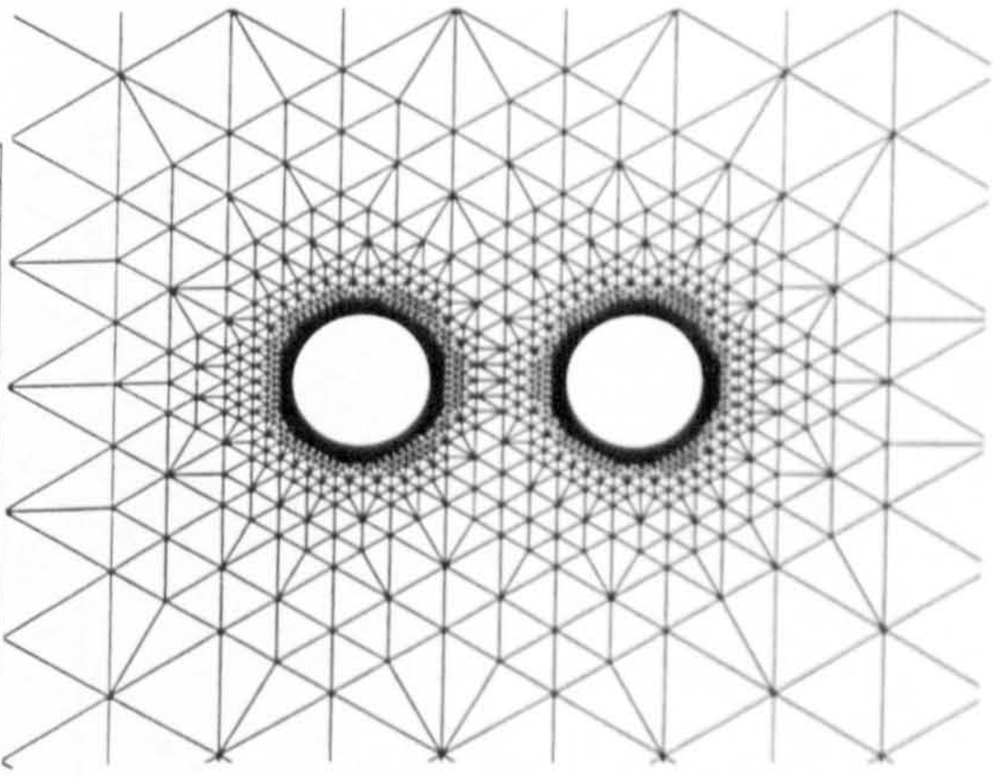


Figure 7. 50 Detail of base grid at P/d_c=2.0

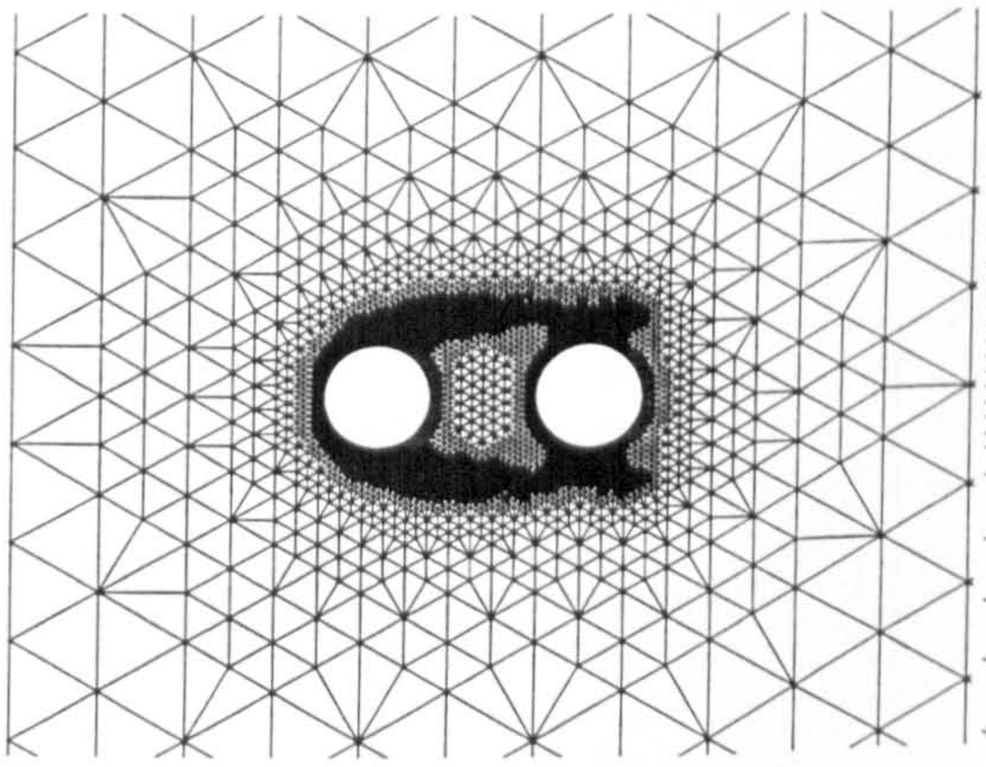


Figure 7. 51 Detail of adapted grid at P/d_c=2.0 (cells=13263,nodes=6911)

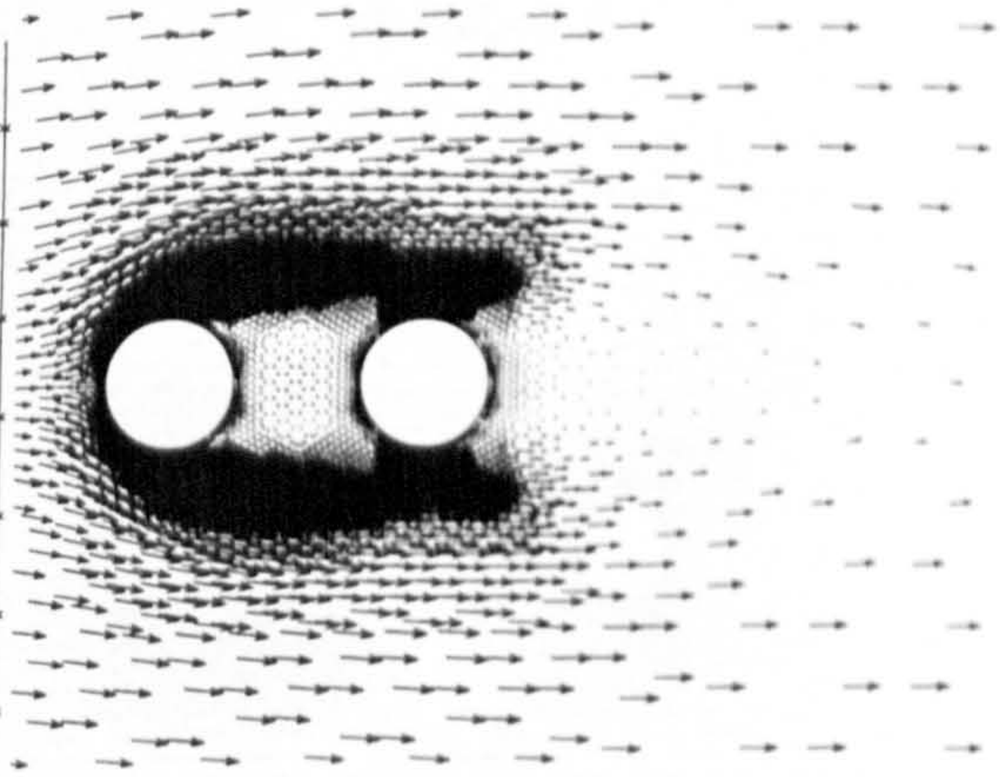


Figure 7.52 Velocity vectors at P/d_c=2.0

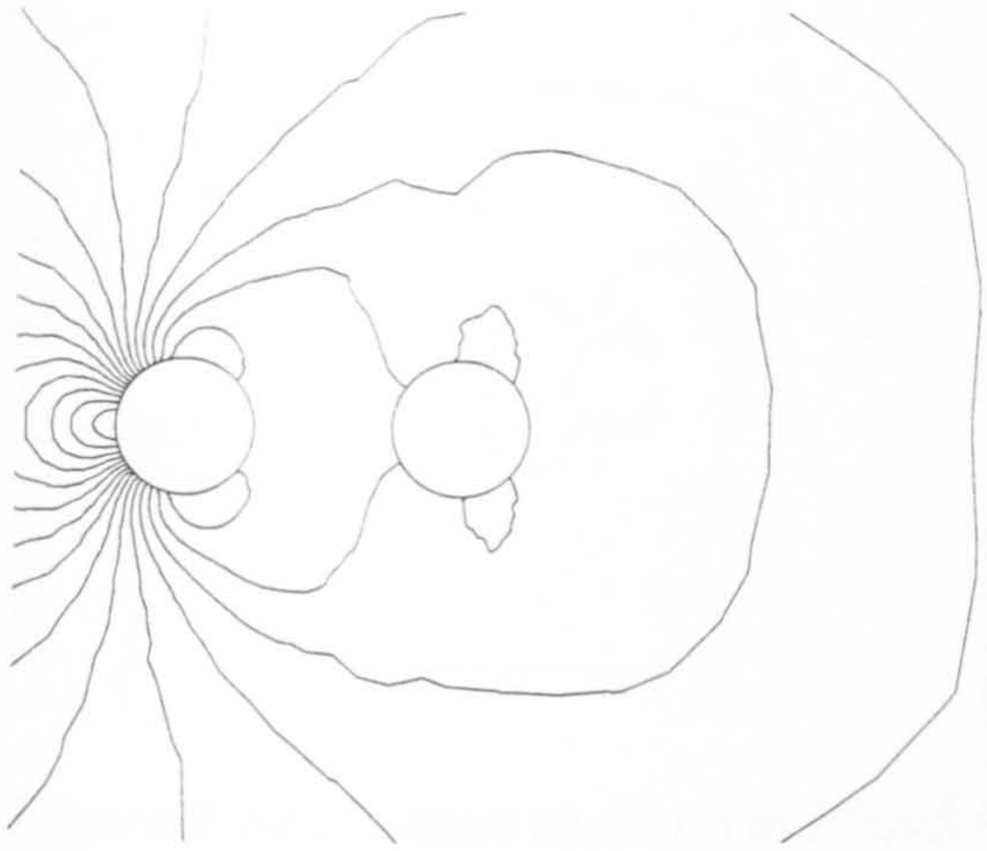


Figure 7.53 Pressure contours at P/d_c=2.0

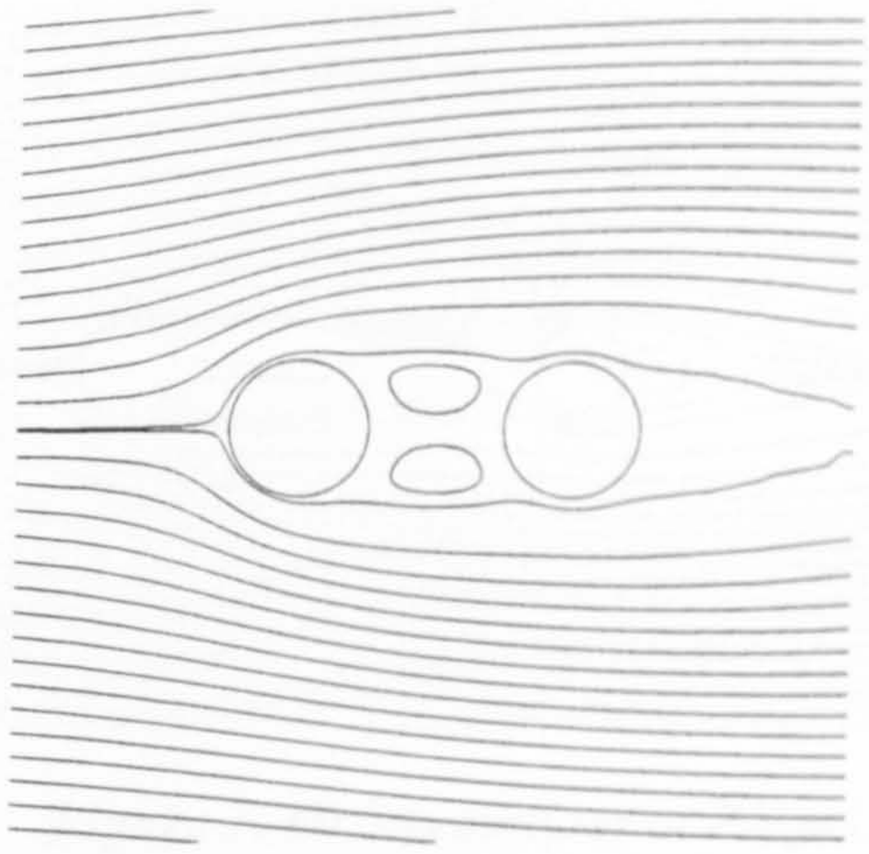


Figure 7.54 Streamlines at P/d_c=2.0

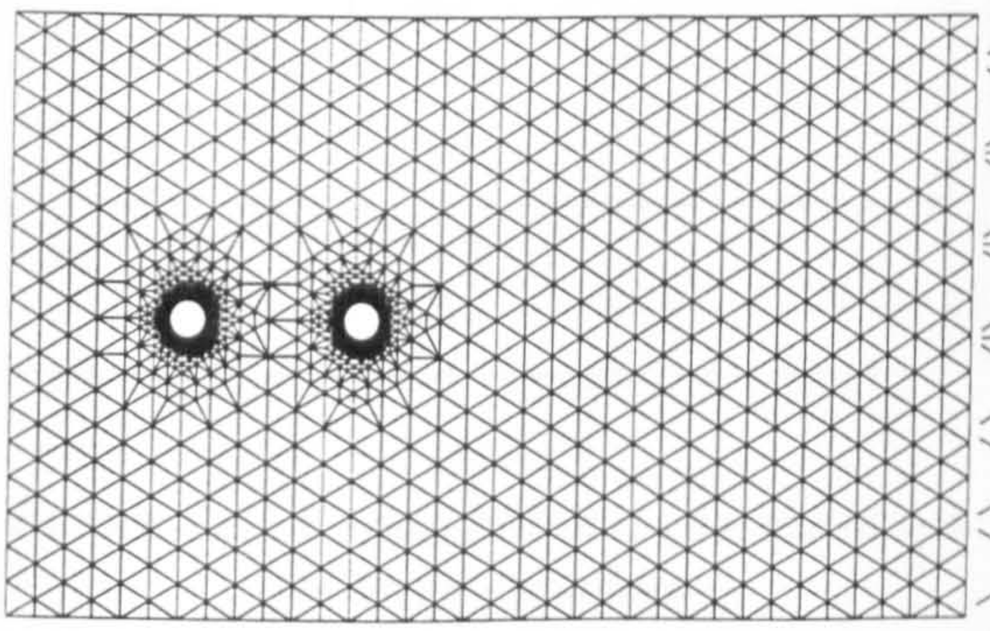


Figure 7. 55 Two cylinders in tandem at P/d_c=5.0 (cells=7187,nodes=3877)

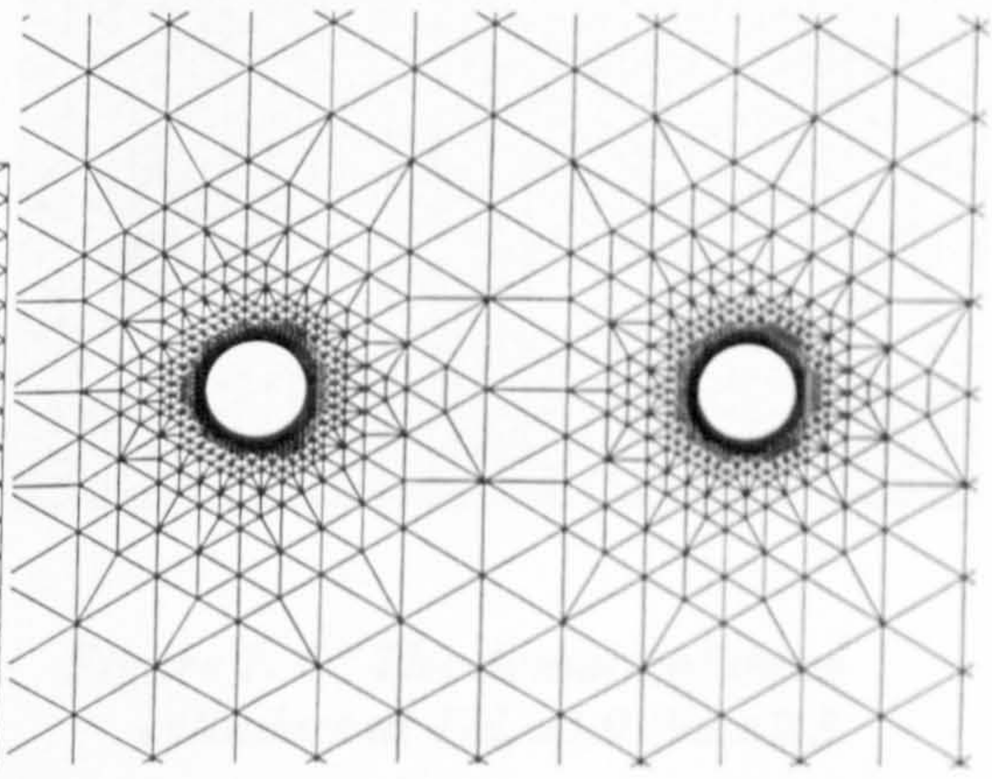


Figure 7. 56 Detail of base grid at P/d_c=5.0

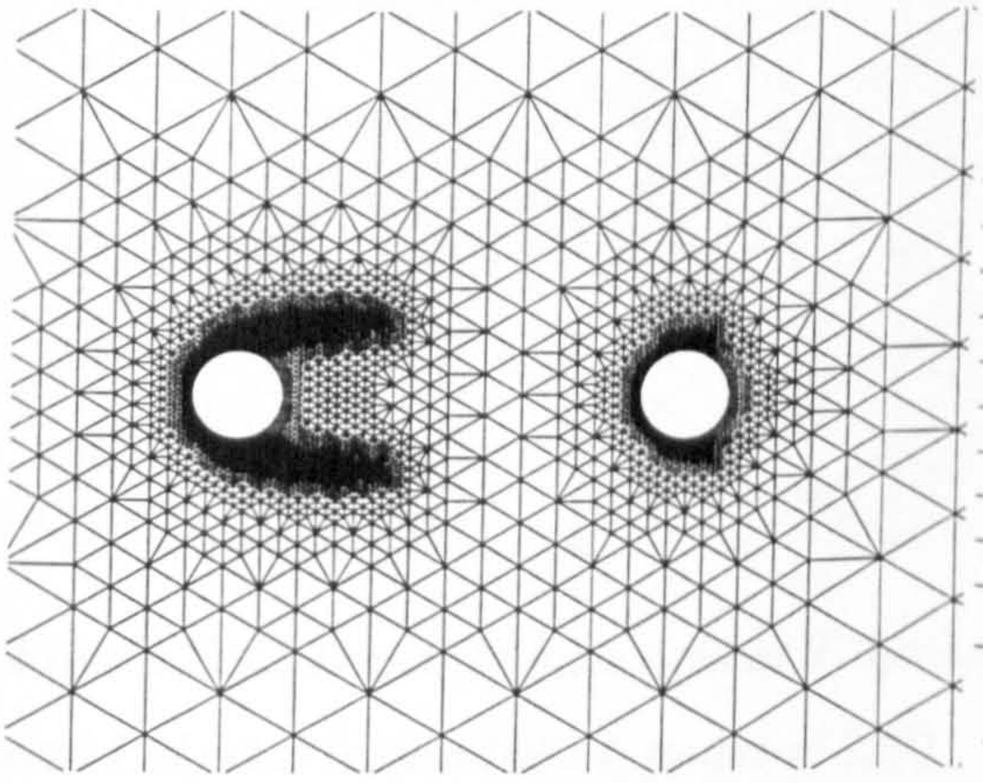


Figure 7. 57 Detail of adapted grid at P/d_c=5.0 (cells=12671,nodes=6619)

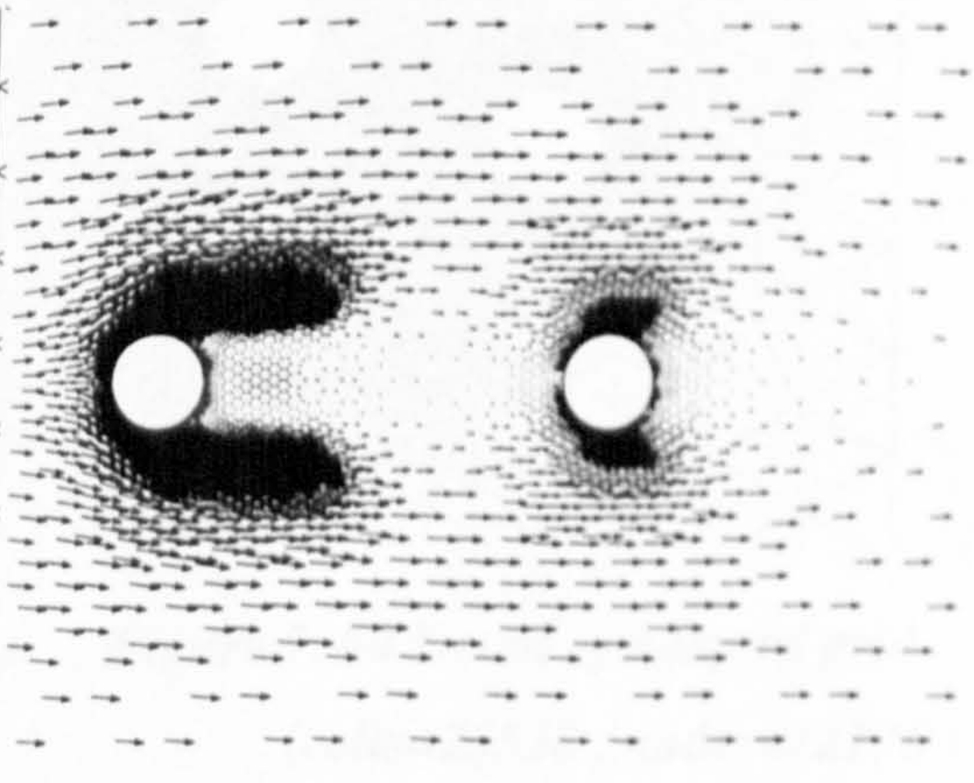


Figure 7.58 Velocity vectors at P/d_c=5.0

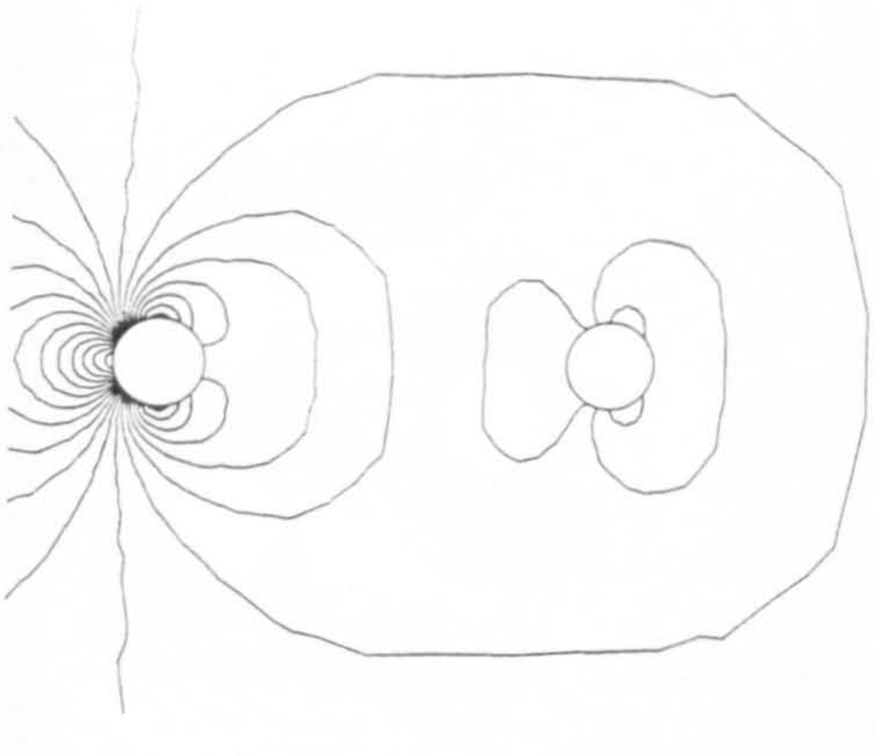


Figure 7. 59 Pressure contours at P/d_c=5.0

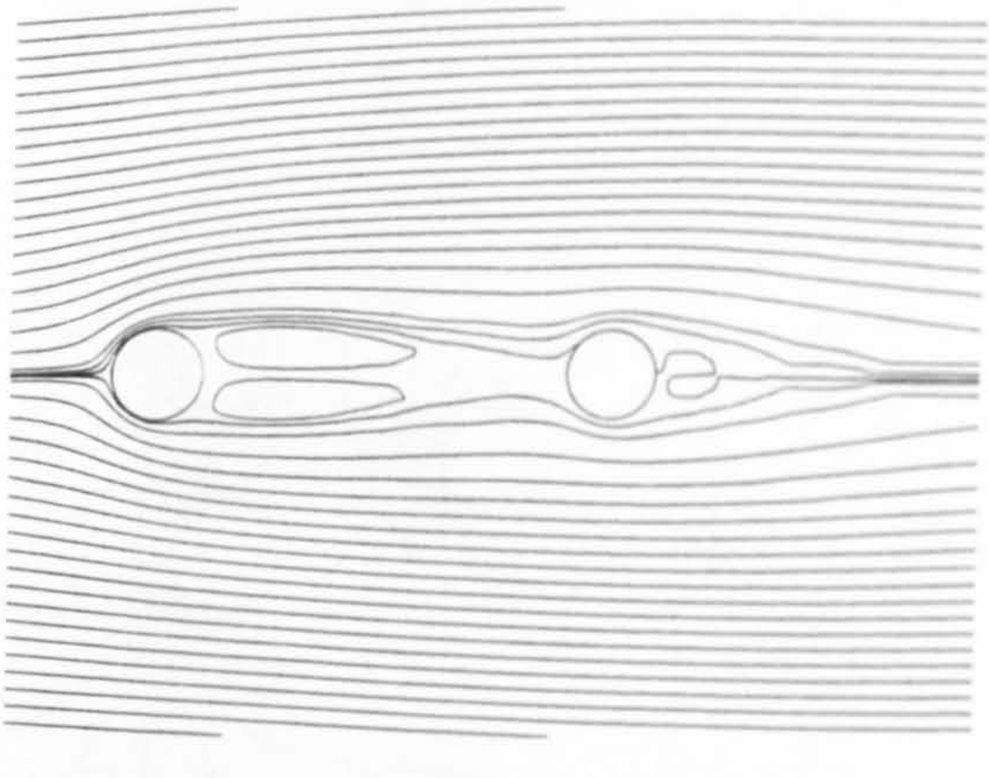


Figure 7. 60 Streamlines at P/d_c=5.0

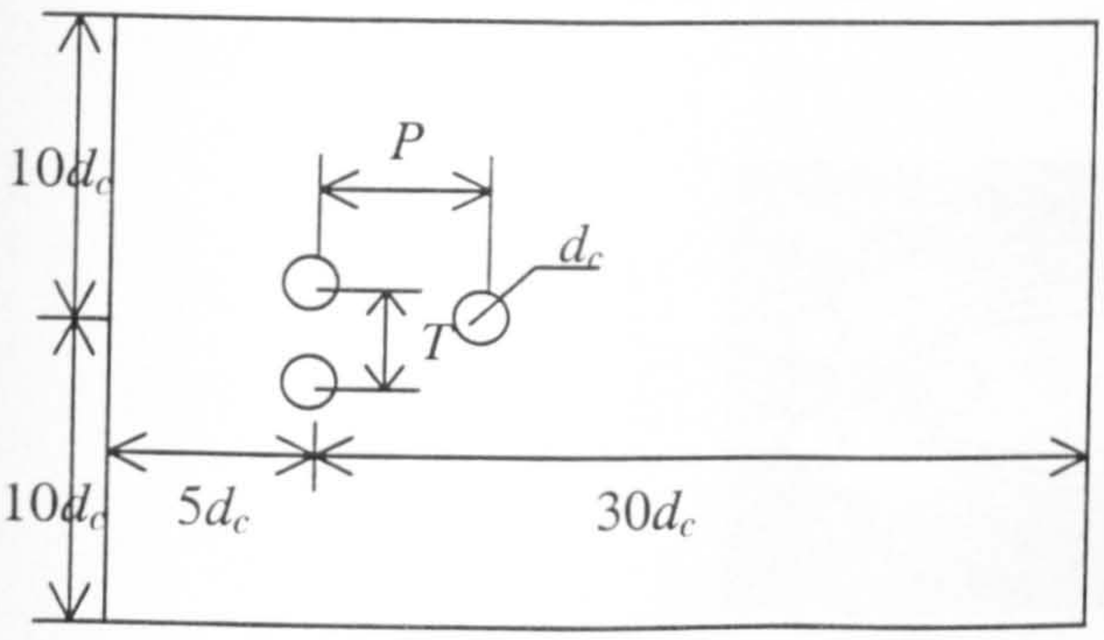


Figure 7.61 Computational configuration at three cylinders

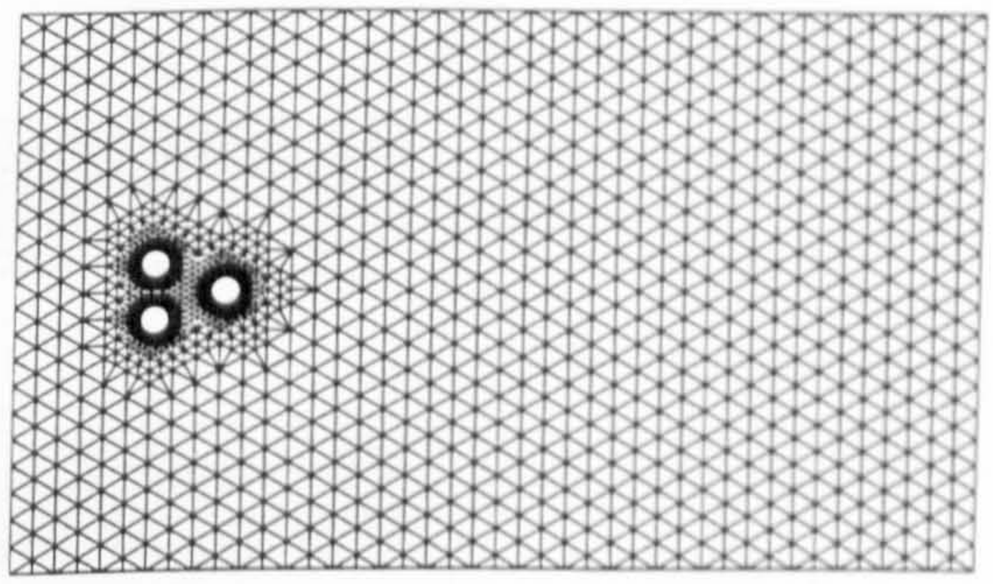


Figure 7. 62 The domain of three cylinders at T/d_c=2.0 P/d_c=2.5

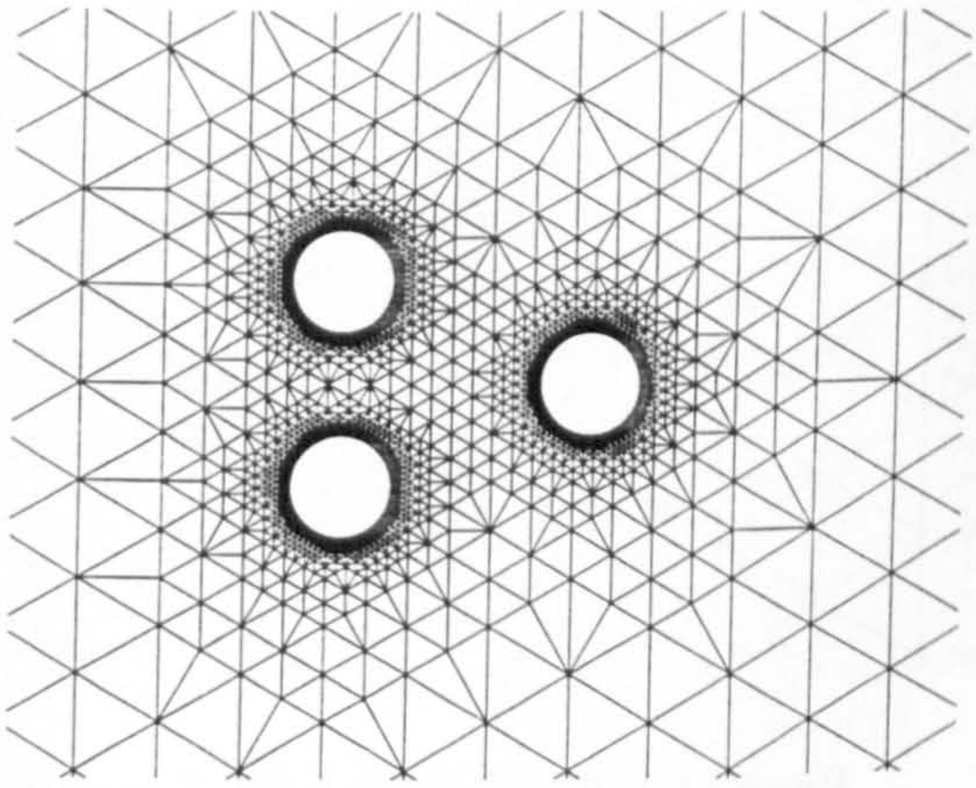


Figure 7. 63 Detail of base grid at T/d_c=2.0 and P/d_c=2.5 (cells=10834, nodes=5829)

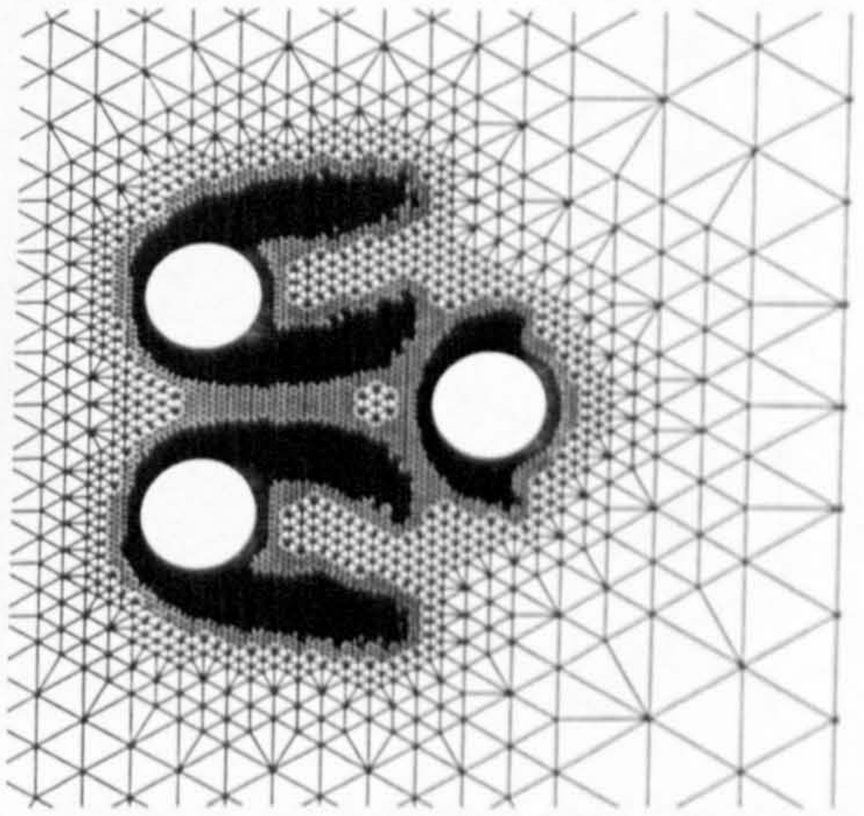


Figure 7. 64 Detail of adapted grid (cells=23528 , nodes=12176)

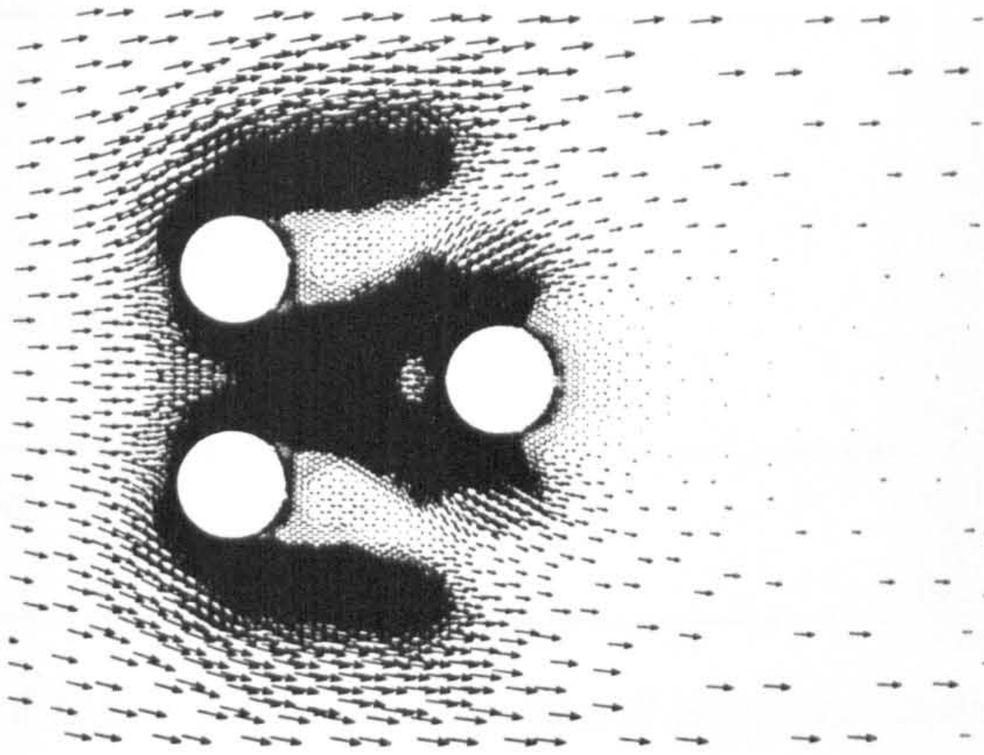


Figure 7.65 Velocity vectors at T/d_c=2.0 P/d_c=2.5

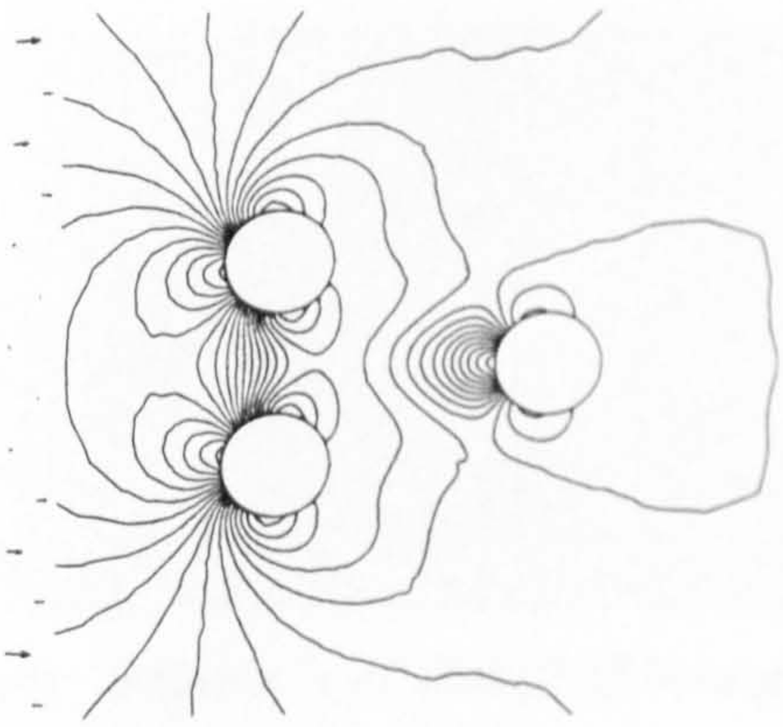


Figure 7. 66 Pressure contours at T/d_c=2.0 P/d_c=2.5

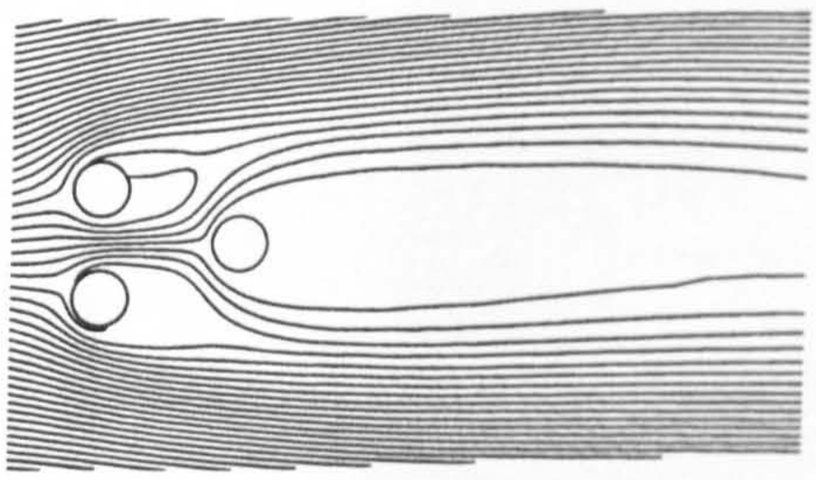


Figure 7. 67 Streamlines at T/d_c=2.0 P/d_c=2.5

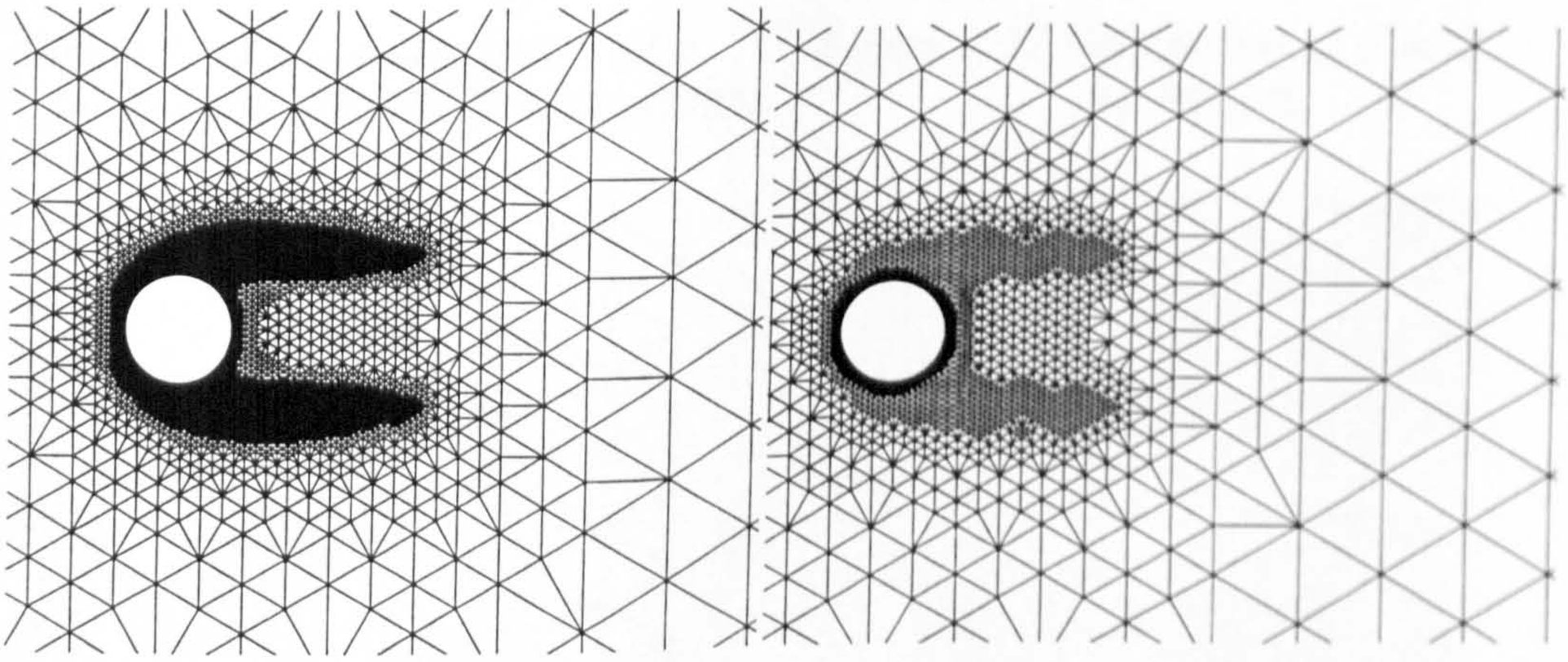


Figure 7. 68 Adapted grid for Re=40 Adapted $Level_{max.}$=11 and $Level_{min.}$=6

Figure 7. 69 Adapted grid for Re=40 Adapted $Level_{max}$=10 and $Level_{min.}$=6

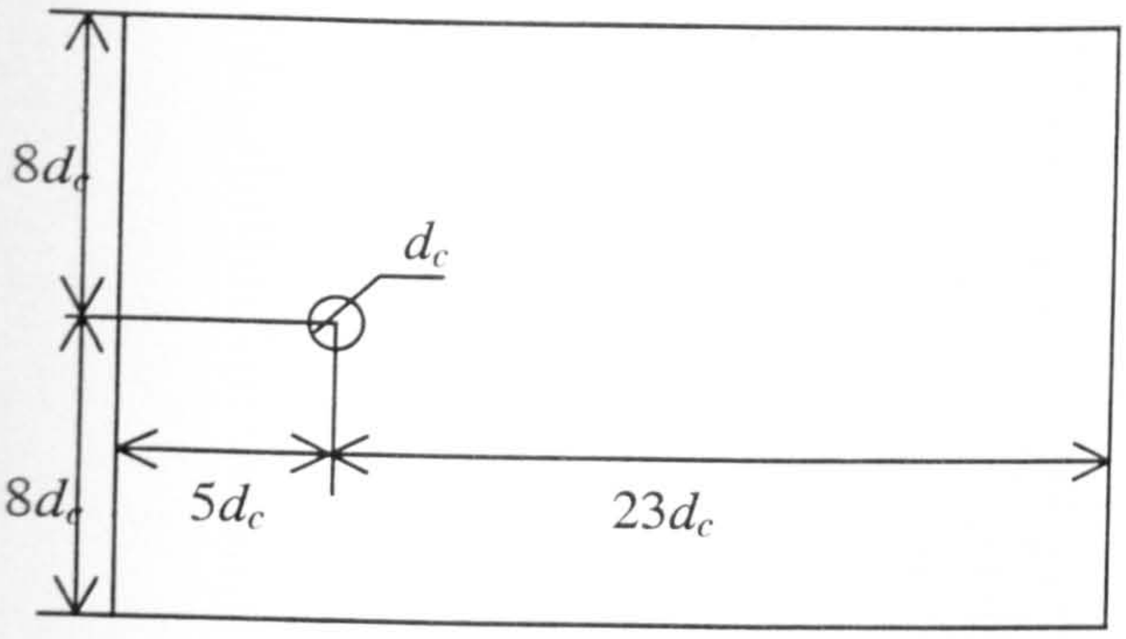


Figure 7.70 Computational configuration

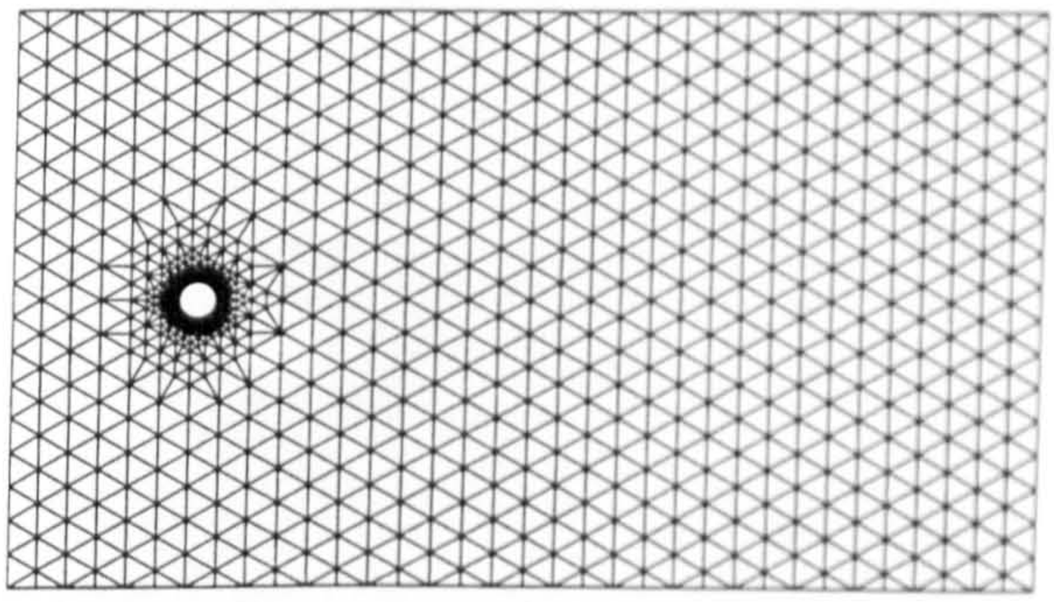


Figure 7. 71 Detail of base grid for Re=200 (cells=4115, nodes=2227)

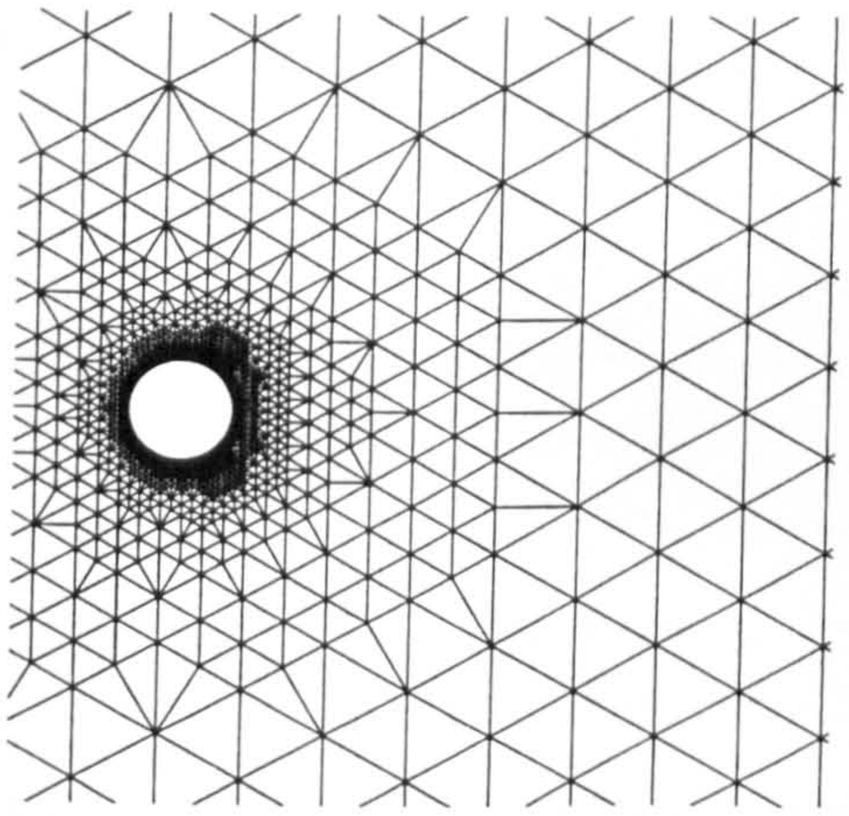


Figure 7. 72 Adapted grid at t^*=1.0 for Re=200 (cells=5067, nodes=2703)

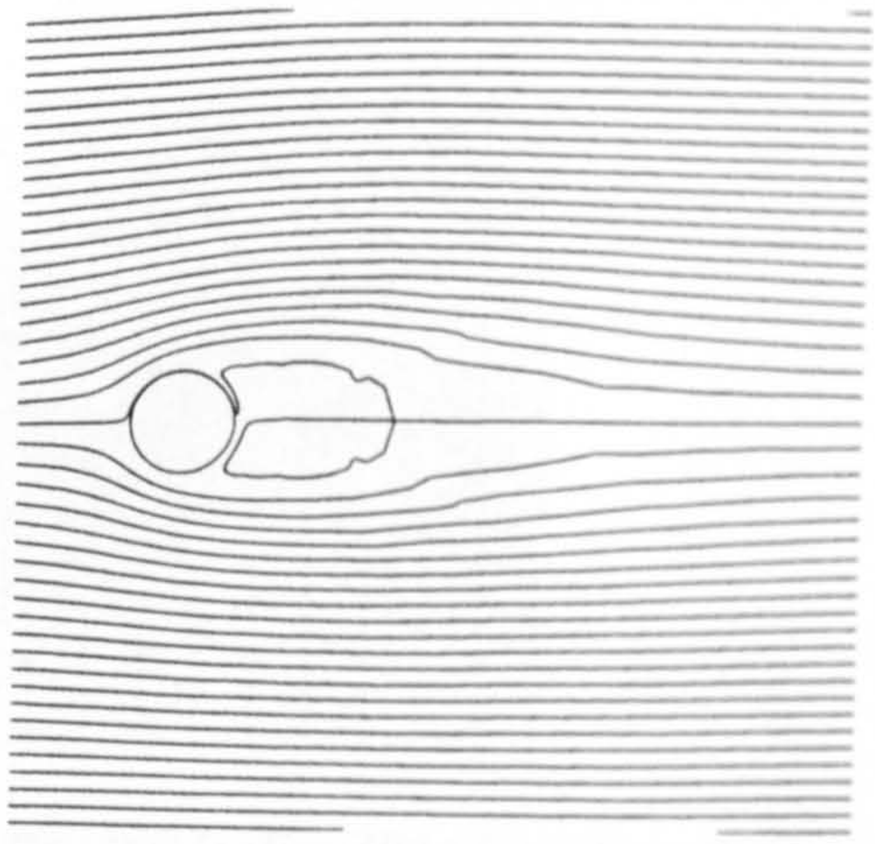


Figure 7. 73 Streamlines at t^*=1.0 for Re=200

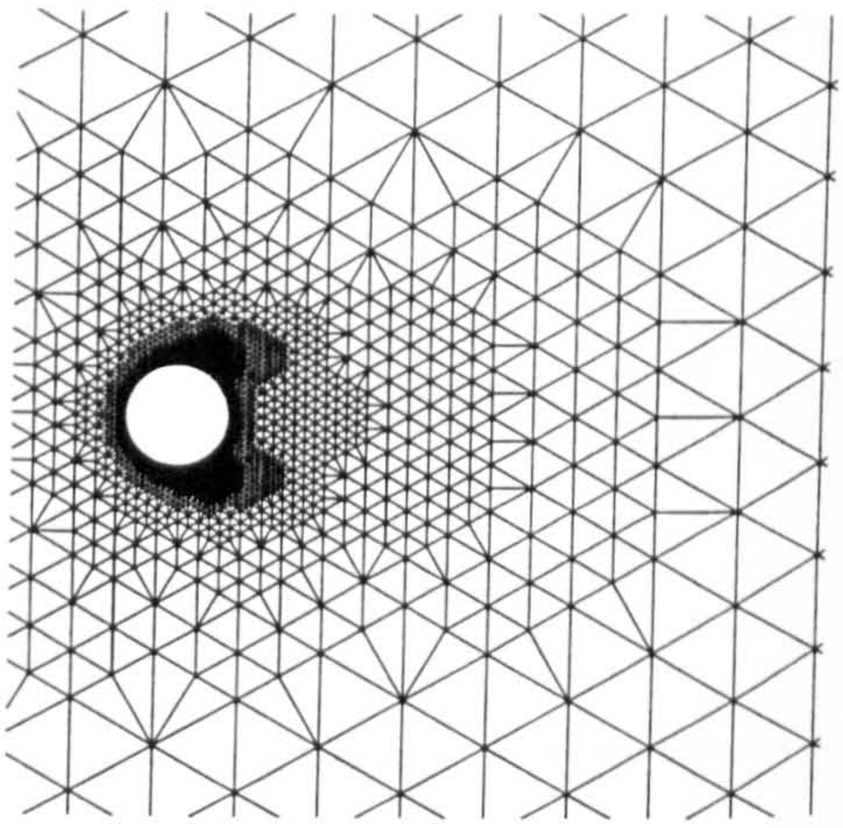


Figure 7. 74 Adapted grid at t^*=2.0 for Re=200 (cells=6319, nodes=3329)

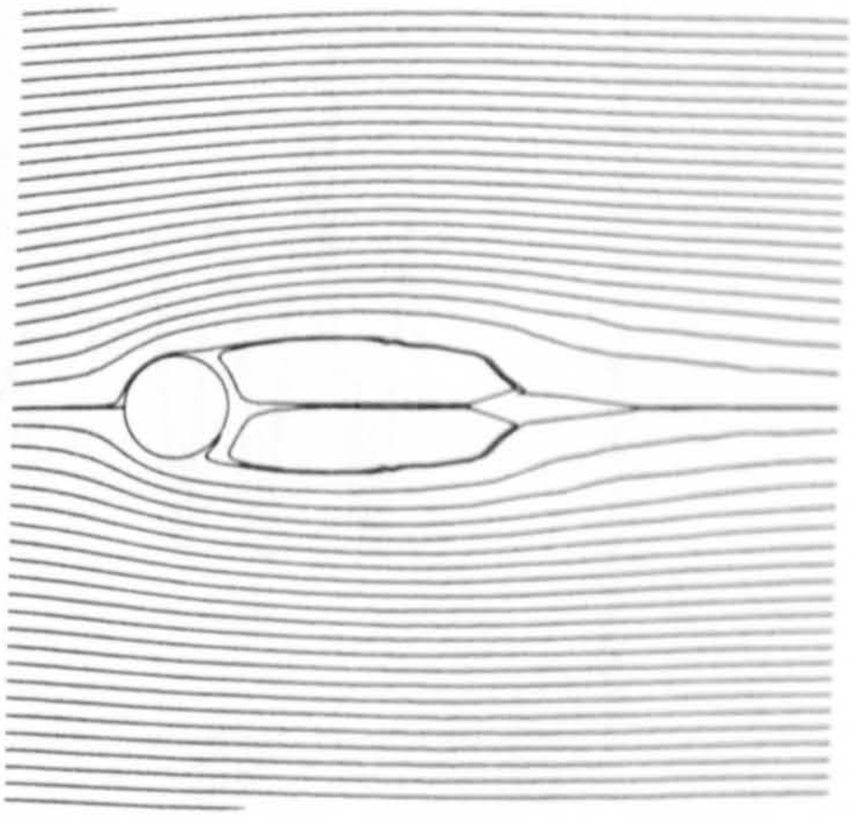


Figure 7. 75 Streamlines at t^*=2.0 for Re=200

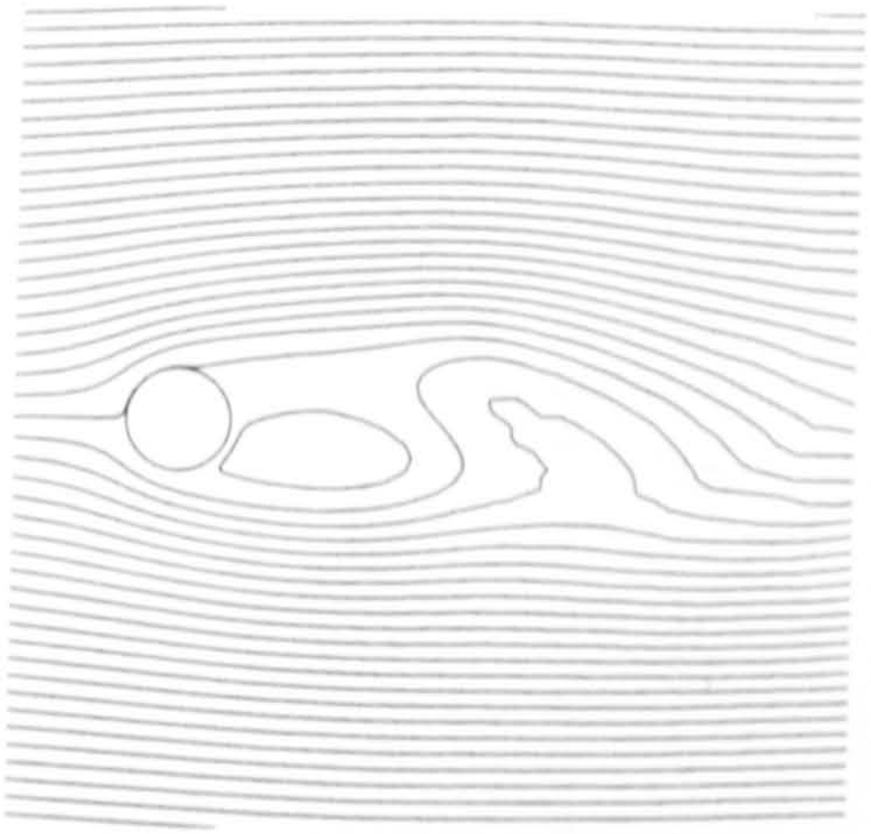


Figure 7. 76 Streamlines at t^*=2.25 for Re=200

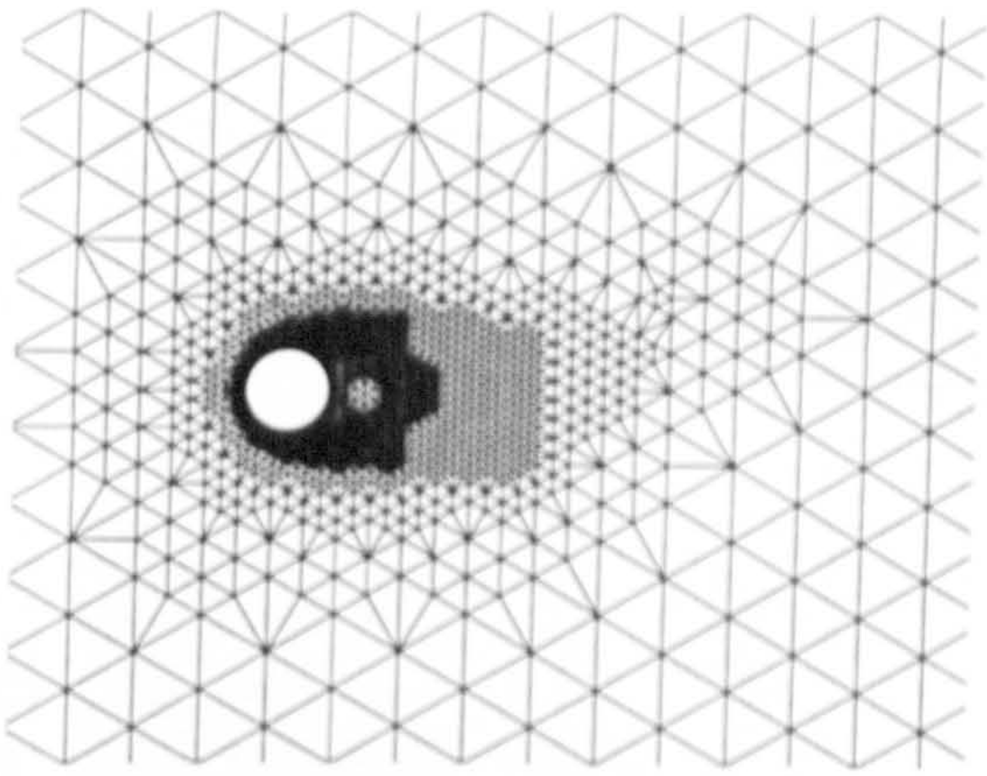


Figure 7. 77 adapted grid at t^*=3.0 for Re=200 (cells=8425, nodes=4382)

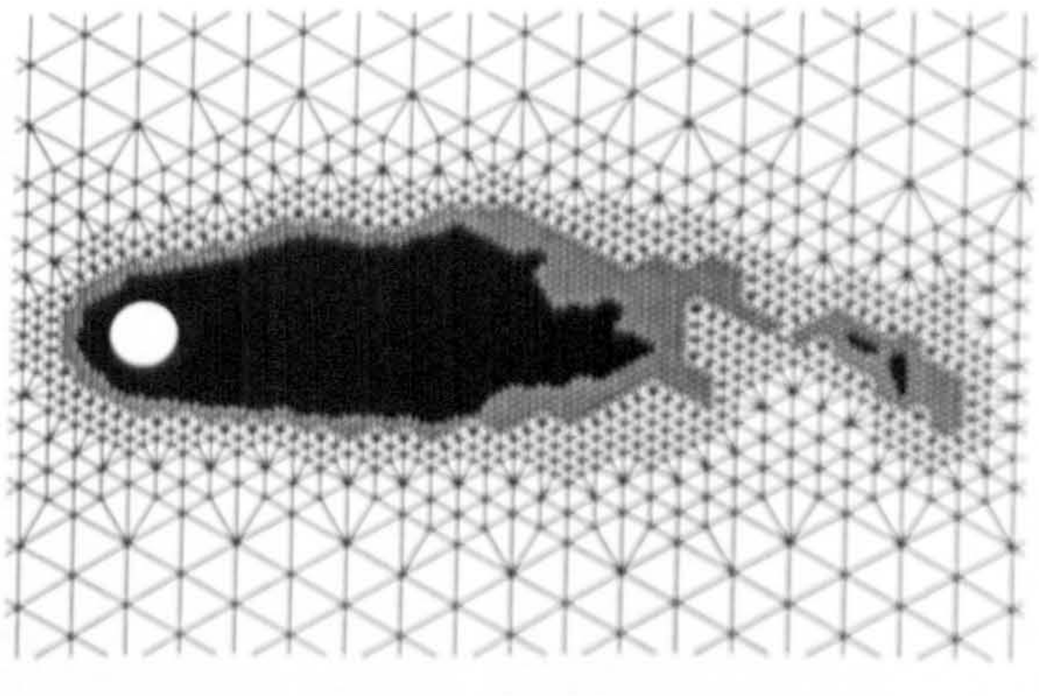


Figure 7. 78 adapted grid at t^*=66.0 for Re=200 (cells=39647, nodes=19993)

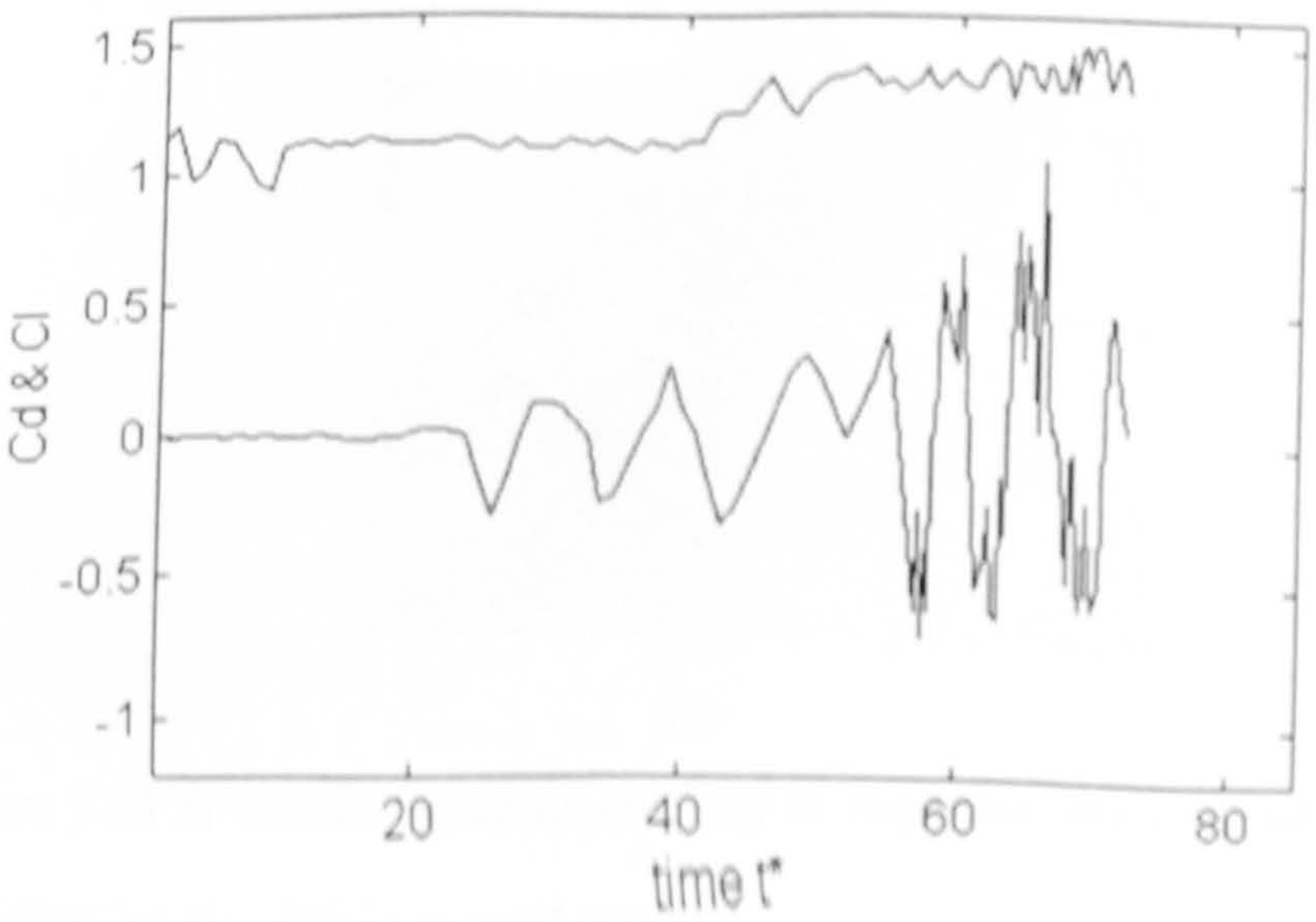
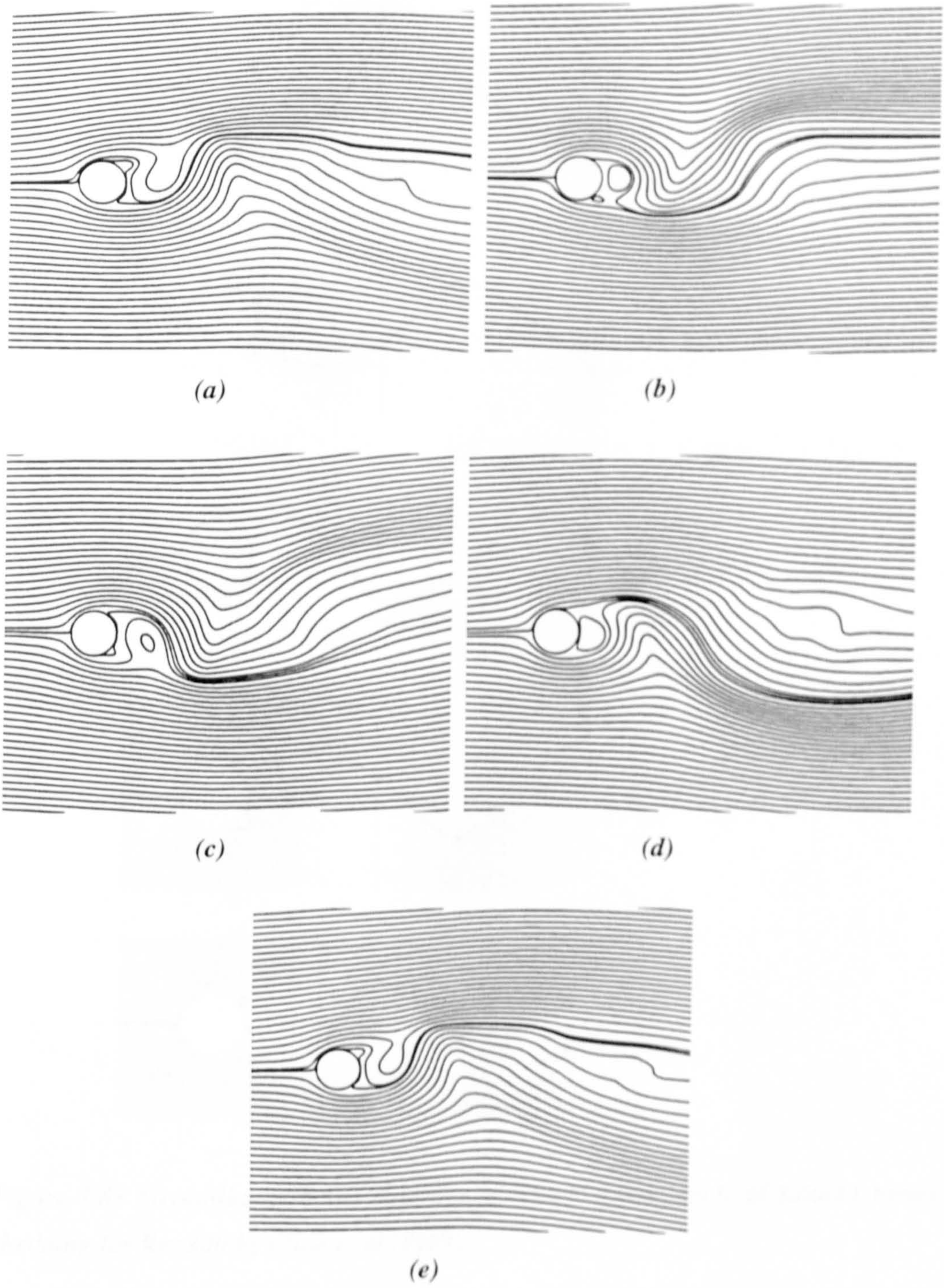


Figure 7. 79 Drag and lift coefficients with time for Re=200



(a) *(b)*

(c) *(d)*

(e)

Figure 7.80 Streamline patterns showing in sequence one cycle of Kàmàn vortex shedding for Re=200 by present results

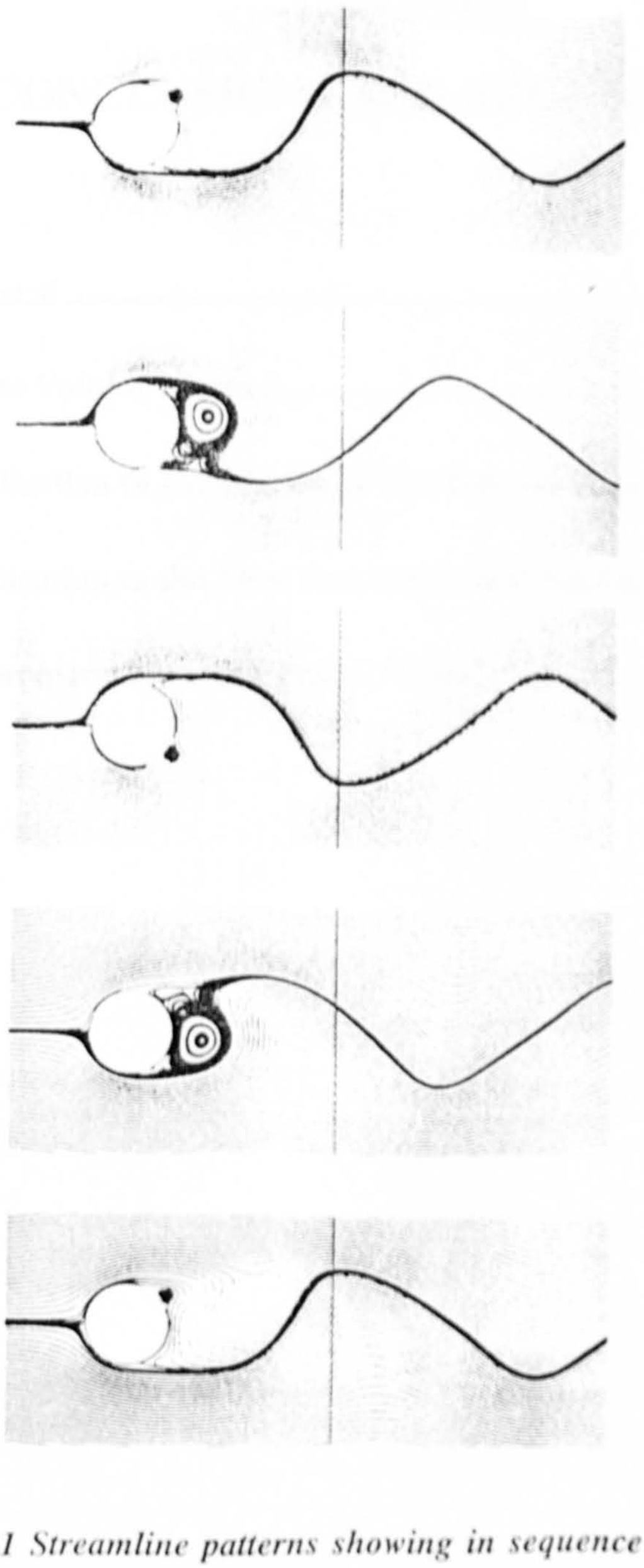


Figure 7.81 Streamline patterns showing in sequence one cycle of Kàmàn vortex shedding for Re=200 by Chen et al (1999)

CHAPTER 8

CONCLUSIONS AND RECOMMENDATIONS

8.1 General

The thesis has presented the study of the numerical simulations of laminar separated flow with tri-tree adaptive grids using the finite volume method. The work has developed a code which,

1) includes a new QUICK differencing scheme for unstructured grids, which satisfies the transportiveness requirement, is stable and possesses good accuracy associated with the non-linear convective term of the Navier-Stokes equations;

2) uses a special treatment to ensure good quality triangular elements around the cylinder boundaries;

3) adapts the grid during the solution in the time domain, in order to track the evolution of vortex shedding for the case of unidirectional flow past a circular cylinder at Re=200.

Considerations of the numerical algorithms and model investigation have been explored. Methodology and derivations have been presented for the analysis and understanding of the underlying physical phenomena, and good agreement with experimental and other numerical data for simulations at low Reynolds numbers has been achieved.

The studies of the unstructured adaptive tri-tree grid generation and the finite volume method have shown that accuracy, flexibility and efficiency demonstrated by some numerical examples appear to be quite favourable.

The studies to date suggest that the algorithm is a potential contender for future offshore engineering and naval applications. Although the selected cylinder configuration is still far from the actual situations encountered in the real world, it provided a good example of laminar flow separation and interaction. This is of great importance to further investigation of more complex flow configurations and the techniques can be extended to general unstructured three-dimensional simulations.

8.2 Finite Volume Method

Detailed considerations of the method have been described in Chapter 4 and three cases of investigation have been covered in Chapter 5, 6 and 7. To summarise, attention is drawn to the numerical formulation for the non-linearity of the discretised Navier-Stokes equations specifically in three aspects:

1) **Special interpolation schemes:** The velocity components and pressure are defined at cell-centres which simplifies the computational code and saves CPU time. However, implementation of the collocated grid arrangement requires calculation of the nodal value from its related cell-centre values. In this thesis, averaged linear interpolation (see equation (4-25)) is used for the channel flow simulations and Shepard-type scattered data interpolation (see equation (3-9)) is used for the simulations of flow past a backward facing-step and flow past cylinders. These interpolations have been successful in estimating the values of the velocity and pressure at vertices of triangular elements and crucial for the success of the collocated finite volume method.

2) **QUICK differencing:** A new formulation of the QUICK scheme for non-uniform grids has been derived, which is able to treat general cases. Versteeg and Malalasekera (1995) point out that the QUICK scheme of Leonard (1979) used an equal distance among the three-point upstream-weighted quadratic interpolation for cell face values, which could not be used in non-uniform grid calculations. The present scheme satisfies the transportiveness requirement, is stable and offers good accuracy associated with the non-linear convective term of the Navier-Stokes equations. The scheme has been successfully used with tri-tree grids with and without grid adaptation (backward-facing step and fluid flow past circular cylinders).

3) **Boundary conditions:** Consideration of boundary conditions has a significant effect in the application of any numerical algorithm. The application of a correct boundary condition is important because usually the stability and accuracy of the discrete condition affects the stability and accuracy of the difference scheme itself. Discussion about Dirichlet and the Neumann boundary conditions is

included in Section 5.3.2.4. The results show that using the Dirichlet boundary condition required less CPU time and achieved greater accuracy than the Neumann boundary condition. However, the Dirichlet boundary condition is not generally available. Therefore, the Neumann boundary condition is applied to the backward facing-step and flow past multiple cylinders. Four alternative methods for calculating the velocity gradient at the boundary have been considered in channel flow. The results show the application of the second-order extrapolation at the parallel wall (i.e. in the direction along which the velocity gradient changes most) is sufficient and so this is used for the other simulations. The no-slip conditions are imposed for the velocity on the cylinder boundaries. A Neumann boundary condition is used for the pressure gradient normal to the cylinder boundary ($\partial p / \partial n = 0$). A prescribed pressure value ($p = 0$) is utilised at the outlet for scaling the pressure values in the computational domain.

In this thesis, a code has been developed that solves the Navier-Stokes equations using a cell-centred finite volume method with unstructured tri-tree adaptive grids. Numerical simulations have been presented for incompressible two-dimensional laminar viscous flows both steady and unsteady. Grid convergence tests were carried out for channel flow and flow past a single cylinder. The results presented herein demonstrate the effectiveness of the method.

8.3 Application of Tri-Tree Grid Adaptation

The tri-tree grid algorithm can be used to generate either triangular grids that do not change with time in the domain or grids that adapt at specific time steps. In this work, fixed tri-tree grids are used for flow past a backward-facing step and adaptive grids for flow past circular cylinders. For the cases of unsteady simulation (Re=40 and Re=200) the calculation grid adapts automatically based on the solution at each time step.

The hierarchical data structure can be an efficient way to store grid information and related adaptation data. Hierarchical grid generation is conceptually easy, in that the grids are created about a set of discrete seeding points by recursive subdivision of a

geometric shape, which from the root level surrounds the flow domain. Element density can then be locally adapted according to the flow activity requirement. This can be readily achieved by *refinement* or *derefinement* of cell elements whilst maintaining the overall tree structure so that a base grid can be defined and adapted locally.without disturbing the tree data structure.

In this work, the grids are adapted in areas of high vorticity, in the boundary layer around cylinders and in the wake. These grids are adapted to provide local resolution by *refinement* and *derefinement* in order to improve the accuracy and efficiency of the finite volume method. The CPU time used for the generation of tri-tree adaptive grids (e.g. about five minutes at each time step for Re =40) was invariably found to be negligible in comparison with the CPU time needed to solve the finite volume method. The author believes that present algorithms of the tri-tree grid generation and its adaptation are more straightforward than the alternative algorithm suggested by Wille (1992), because the reference numbering adopted in this work requires the storage of a single integer, as opposed to Wille's (1992) nine. However, whilst the Wille's reference numbering can carry on more information, all the parameters have to be recalculated using the present method.

Here, a special treatment is developed for cylinder boundaries to achieve a smooth approximation to curved boundaries. The stretching technique is applied to ensure the nodes lie on cylinders and the elimination of obtuse angles is used to avoid error in the numerical solution. This treatment ensures good quality elements around the cylinder walls and the results have validated the significance of this innovation.

8.4 Application to the Flow Past Single and Multiple Cylinders

Investigation into the flow past single and multiple cylinders has demonstrated that the numerical results are in reasonable agreement with published data. This also means that the algorithms in this work work well, moreover with benchmark cases and low Reynolds number simulations. Furthermore, these numerical results have shown that not only the calculation of certain parameters is covered but also phenomena such as separated flow and vortex shedding are correctly predicted.

The grid convergence tests have been carried out for Re=2.04 simulations. It was found that the grid size has an important impact on the convergence and accuracy of numerical solutions.

The results for flow past a single cylinder at Re=10 and Re=40 have been shown with adaptation. The results for flow past a single cylinder at Re=40 were obtained for both steady and unsteady simulations. The evolutions in time of some parameters have been predicted and are in good agreement with published data. Furthermore, the flow past multiple cylinders at Re=40 using the unsteady simulation have been studied. The comparison between our numerical results and those of published papers showed similar characteristics and tendencies.

The force coefficients have been calculated for cases where one, two or three cylinders lie in the domain. The comparisons made in these cases confirm that the calculated forces agree reasonably with corresponding published solutions. The interaction of multiple cylinders is found to be dependent of their arrangement which affects the observed flow characteristics. For example, the obtained drag coefficient for one cylinder is larger than two cylinders in tandem arrangement and less than two cylinders in transverse arrangement or the three cylinder arrangement used in this thesis. This is caused by the flow interaction around the cylinders, which is strongly dependent on the arrangements. However, although the present numerical simulation can show trends, determining the optimal spacing between these multiple cylinders requires more investigation.

The adaptive grid was used for visualisation of vortex shedding for the Re=200 case. However, the point by point iteration scheme used here is expensive in terms of computational time. The reason is due to the high number of grid elements required. Therefore, fast iteration schemes should be investigated for use on non-uniform tri-tree grids before extending to higher Reynolds number flows.

8.5 Recommendations for Future Work

With regard to future work, the success of numerical simulation depends on two major factors, a good understanding of physical characteristics of flow dynamics, and

a good grasp of the numerical simulation procedure. It is believed that the thesis has made an initial contribution in these two aspects for the numerical simulation and will benefit the future development and extension of the numerical algorithms. The code that has been developed during the study provides preliminary tools for further development of practical and commercial packages. The recommendations for future work can be given as the following.

The work presented in this thesis uses a special treatment which achieved high quality triangular elements around cylinder boundaries. For further work, it is recommended that technology be developed for generating high quality elements in regions surrounding any complex surfaces. In addition, the distorted elements surrounding boundaries could be locally adapted with time to tackle moving boundary problems.

The drawback of the numerical solver is its CPU time consumption for finer grids at *Re*=200. The multigrid method for accelerating convergence has potential to be used with unstructured grids as shown by Löhner and Morgan (1987), Gáspár and Józsa (1991) and Francescatto and Dervieux (1998). The tri-tree grid algorithm is especially suited to multigrid application because of its hierarchical data structure. The tri-tree reference number of a given triangle points to the reference number of its parent triangle, which allows the algorithm to perform both up- and down-searches for multigrid applications.

Tri-tree adaptive grids have been successfully used in this thesis. They can provide good accuracy and can be efficiently generated. Extension of the tri-tree grid algorithm to a three-dimensional tetra-tree grid algorithm is conceptually straightforward. However, the finite volume simulation on three-dimensional configurations imposes great demands on hardware both in terms of memory and CPU time. Nevertheless, an adaptive tetra-tree grid is likely to form a good basis for a three-dimensional finite volume flow simulation.

It is recommended that future work be carried out to extend the model towards higher Reynolds number flows including turbulence simulation. To achieve that, the formulations of turbulent flows need to be approximated, with the governing Navier-Stokes equations replaced by the Reynolds Averaged Navier-Stokes (RANS)

equations. The RANS equations contain certain Reynolds stress terms which arise from turbulent relations through the averaging procedure. These terms can be dealt with using a suitable k-ε model, such as the Reynolds stress model (RSM) as used by Davidson (1993), or the algebraic stress model (ASM) as used by Demuren and Rodi (1984). A turbulence model will enable the code to deal with wider industrial applications.

REFERENCES

Abbott, M. B. and Basco, D. A. (1989), "Computational fluid dynamics, an introduction for engineers", *Longman Scientific & Technical, U.K.*

Acheson, D. J. (1990), "Elementary fluid dynamics", *Oxford Applied Maths and Computing Science series, Clarendon press, U.K.*

Allievi_A and Bermejo, R. (1997), "A generalized particle search-locate algorithm for arbitrary grids", *J. Compu. Phys.*, Vol.132, pp.157-166

Atkins, D. J., Maskell, S. J. and Patrick, M. A. (1980), "Numerical prediction of separated flows", *Int. J. for Num. Meth. Eng.*, Vol. 15, pp. 129-144.

Babuska, I. and Aziz, A. K. (1976), "On the angle condition in the finite element method", *SIAM J. Num. Analysis*, pp. 214-216.

Babuska, I., Strouboulis, T. and Gangaraj, S. K. (1997), "A posteriori estimation of the error in the recovered derivatives of the finite element solution", *Comput. Meth. App. Mech. Eng.*, Vol.150, pp.369-396

Banks, D., Mueller, J. D. and Vankeirsbilick, P. (1996), "An object oriented approach to hybrid structured/unstructured grid generation," *AIAA Paper* 96-0032.

Baker, T. J. (1986), "Three dimensional mesh generation by triangulation of arbitrary point sets", *AIAA Paper* 87-1124.

Baker, T. J. (1987), "Mesh generation by a sequence of transformations", *App. Num. Math.* Vol.2, pp.515-528.

Barragy, E. and Carey. G.F. (1988), "A partitioning scheme and iterative solution for sparse bordered systems" *Comput. Meth. Appl. Mech. Eng.*, Vol. 70, pp. 321-327.

Barton, I. E. (1994), "Laminar flow past an enclosed and open backward-facing step" *J. Phys. Fluid*, Vol. 6, pp. 4054-4056.

Baruzzi, G. S., Habashi, W., Guevremont, J.G. and Hafez, M M. (1995), "A 2nd-order finite element method method for the solution of the transonic Euler and Navier-Stokes equations", *Int. J. for Num. Methods in Fluids*, Vol. 20, pp. 671-693.

Batchelor, G. K. (1994), "An introduction to fluid dynamics", *Cambridge University Press, U.K.*

Benek, J. A., Buning, P. G. and Steger, J. L. (1985), "A 3-D Chimera grid embedding technique" *AIAA Paper* 85-1523.

Berzins, M., Ware, J. and Lawson, J. (1992), "Spatial and temporal error control in the adaptive solution of systems of conservation laws" *in Advances in Computational Methods for PDEs, IMACS PDE VII, IMACS.*

Borthwick, A. G. L. and Kaar, E. T. (1993), "Shallow flow modelling using curvilinear depth-averaged stream function and vorticity transport equations", *Int. J. for Num. Methods in Fluids*, Vol. 17, pp. 417-445.

Bowyer, A. (1981), "Computing Dirichlet tesselation", *J. Comput.*, Vol. 24, pp.162-166.

Carey, G. F., Wang, K. C. and Joubert, W.D. (1989), "Performance of iterative methods for Newtonian and generalised Newtonian flows", *Int. J. for Num. Methods in Fluids*, Vol. 9, pp. 127-150.

Cendes, Z. J., Shenton, D. and Shahnasser, H. (1983), "Magnetic computation using Delaunay triangulation and complementary finite element", *IEEE. Transactions on Magnetics*, Vol. Mag-19, pp. 2551-2554.

Chambarel, J. L., Donneaud, M. and Martin, R. (1991), "Numerical study of laminar flow past one and two circular cylinders", *Comput. Fluids*, Vol. 19, pp.155-170.

Chan, C. T. and Anastasion, K. (1999), "Solution of incompressible flows with or without a free surface using the finite volume method on unstructured triangular meshes", *Int. J. for Num. Methods in Fluids*, Vol. 29, pp. 35-57.

Chen, Y.H., Yang, S. C. and Yang, J. Y. (1999), "Implicit weighted essentially non-oscillatory schemes for the incompressible Navier-Stokes equations", *Int. J. for Num. Methods in Fluids*, Vol. 31, pp. 747-765.

Collins, W. M. and Dennis, S. C. R. (1973), "Flow past an impulsively started circular cylinder", *J. Fluid mech.*, Vol. 60, pp. 105-127.

Connell, S. D. and Braaten, M. E. (1995), "Semi-structured mesh generation for 3-D Navier-Stokes calculations", *AIAA Paper*, 95-1679.

Coutanceau, M. and Bouard R. (1977), "Experimental determination of the main features of the viscous flow in the wake of a circular cylinder in uniform translation. Part 1. Steady flow", *J. Fluid Mech.*, Vol. 79, pp. 231-256.

Coutanceau, M. and Bouard, R. (1977), "Experimental determination of the main features of the viscous flow in the wake of a circular cylinder in uniform translation. Part 2. Unsteady flow", *J. Fluid Mech.*, Vol. 79, pp. 257-272.

Davidson, L. (1993), "Implementation of a k-ε model and a Reynolds stress model into a multiblock code", *Cothcuburg, Sweden.*

Davidson, L. (1996), "A pressure correction method for unstructured meshes with arbitrary control volumes", *Int. J. for Num. Methods in Fluids*, Vol. 22, pp. 265-281.

Demuren, A. O. and Rodi, W. (1984), "Calculation of turbulence-driven secondary motion in non-circular Ducts", *J. Fluid Mech.*, Vol. 140, pp. 189-222.

Denham, M. K. and Patrick, M. A. (1974), "Laminar flow over a downstream-facing step in a two-dimensional flow channel", *Trans. Ins. Chem. Eng.*, Vol. 52, pp. 361-367.

Dennis, S. C. R. and Chang, G. Z. (1970), "Numerical solution for steady flow past a circular cylinder at Reynolds numbers up to 100", *J. Fluids Mech.*, Vol. 42, pp. 471-489.

Dimopoulos, H. G. and Hanratty, T.J. (1968), "Velocity gradients at the wall for flow around a cylinder for Reynolds numbers between 60 and 360", *J. Fluid Mech.*, Vol. 33, pp. 303.

Ding, Y. and Kawahara, M. (1999), " Three-dimensional linear stability analysis of incompressible viscous flows using the finite element method", *Int. J. for Num. Methods in Fluids*, Vol. 31, pp. 451-479.

Eiseman, P. R. (1985), "Alternating direction adaptive grid generation", *AIAA J.*, Vol. 23. pp.551-556

Eriksson, L. E. (1982), "generation of boundary-conforming grids around wing-body configuration using transfinite interpolation", *AIAA J.*, Vol. 20. pp. 1313-1320.

Evans, A. (1993), "Mesh adaptivity in compressible flow", *PhD. thesis, University College of Swansea, Swansea.*

Flores, J., Reznick, S. G., Holst, T. L. and Gundy, K. (1987), "Transonic Navier-Stokes solutions for a Fighter-like configuration", *AIAA Paper* 87-0032.

Fortune, S. (1986), "A sweepline algorithm for Voronoi diagrams", *Proc. of the 2nd computational Geometry Conference, Yorktown Heights, NY, June.*

Francescatto, J. and Dervieux, A. (1998), "A semi-coarsening strategy for unstructured multigrid based on agglomeration", *Int. J. for Num. Methods in Fluids*, Vol. 26, pp. 927-957.

Franke, R. (1982), "Smooth interpolation of scattered data by local thin plate splines", *Comput. Math. App.*, Vol.8, pp.273-281

Franke, R., Rodi, W. and Schönung, B. (1990), "Numerical calculation of laminar vortex shedding flow past cylinders ", *J. of Wind Eng. and Ind. Aerodynamics*, Vol. 35, pp. 237-257.

George, P. L. (1991), "Automatic mesh generation", *Masson, Paris. ISBN 2-225-82564-5*

Giraldo, F. X. (1997), "Lagrange-Galerkin methods on Spherical geodesic grids", *J. Comput. Phys.*, Vol.136, pp. 197-213.

Greaves, D. M. (1995), "Numerical modelling of laminar separated flow and inviscid steep waves using adaptive hierarchical meshes", *DPhil thesis, University of Oxford.*

Greaves, D. M. and Borthwick, A. G. L. (1997), "Adaptive quadtree grids applied to the computation of flow past circular cylinders ", *Eighth International Conference on Offshore structures, Boss'97.*

Greaves, D. M. and Borthwick, A. G. L. (1998), "On the use of adaptive hierarchical meshes for numerical simulation of separated flows", *Int. J. for Num. Methods in Fluids*, Vol. 26, pp. 303-322.

Grove, A. S., Shair, F. H., Petersen, E. F. and Acrivos, A. (1963), "An experimental investigation of the steady separated flow past a circular cylinder", *J. Fluid Mech.*, Vol. 19, pp. 60-85.

Gustafsson, I. (1978), "A class of first order factorisation methods", *BIT. 8*, pp. 142-156.

Gáspár, C., Józsa, J. and Simbierowicz, P. (1991), "Lagrangian modelling of the convective diffusion problem using unstructured grids and multigrid technique", *Proc. 1st. Int. Conf. On Water Pollution*, 3-5 *Sep. Southampton U.K.*

Gáspár, C. and Simbierowicz, P. (1992), "Difference schemes in tree-structured multigrid context", *C.M.W.R. Denver.*

Hackbrusch, W. and Trottenberg, U. (1982), "Multigrid methods", *Springer-Verlag, berlin.*

Hackman, L. P., Raithby, G.D. and Strong, A. B. (1984), "Numerical predictions of flows over backward-facing steps", *Int. J. for Num. Methods in Fluids*, Vol. 4, pp. 711-724.

Harlow, F.H. and Welch, J. E. (1965), "Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface", *The Phys. of Fluid*, Vol. 8, pp. 2182-2189.

Hassan, O. and Morgan, K. (1996), "Unstructured tetrahedral mesh generation for three-dimensional viscous flows", *Int. J. for Num. Methods in Eng.*, Vol. 39, pp. 549-567.

Heinrich, J. C., Huyakorn, P.S., Zienkiewicz, O.C. and Mitchell, A. R. (1977), "An upwind finite element scheme for two-dimensional convective transport equation", *Int. J. Numer. Meth. Eng.*, Vol. 11, pp. 131-143.

Hell, D. L. and Baskharone, E. A. (1977), "A multiblock Navier-Stokes algorithm using equal-order quadratic finite elements", *Int. J. Numer. Meth. Fluids*, Vol. 20, pp. 169-185.

Holmes, D. G. and Lamson, S. H. (1986), "Adaptive triangular meshes for compressible flow solutions", *Int. Conference on Num. Grid generation in CFD, Landshut, FRG.*

Hufford, G. S. and Mitchell, C. R. (1995), "The generation of hybrid and unstructured grids using curve and area sources", *AIAA. Paper*, 95-0215.

Hétu, J. F. and Pelletier, D. (1992), "Adaptive remeshing for viscous incompressible flows", *AIAA. J.*, Vol. 30, pp. 1986-1992.

Ilinca, F. Pelletier, D. and Ignat, L. (1998), "Adaptive finite element solution of compressible turbulent flows", *AIAA J.*, Vol.36, pp.2187-2194

Jackson, C. P. (1987), "A finite element study of the onset of vortex shedding in low past variously shaped bodies", *J. Fluid Mech.*, Vol. 182, pp. 23-45.

Jameson, A., Baker, T. J. and Weatherhill, N. P. (1986), "Calculation of inviscid Transonic Flow Over a Complete Aircraft", *AIAA J.*, pp. 86-103.

Jameson, A. and Baker, T. J. (1987), "Improvements to the aircraft Euler method", *American Institute of Aeronautics and Astronautics, Inc.*, pp. 1-18.

Jaime S. Son and Thomas J. Hanratty (1969), "Numerical solution for the flow around a cylinder at Reynolds numbers of 40, 200 and 500", *J. Fluids Mech.* Vol. 35, pp. 369-386.

Jiang, Y. and Przekwas, A. J. (1994), "Implicit, pressure-based incompressible Navier-Stokes equations solver for unstructured meshes", *AIAA Paper*, 94-0303.

Józsa, J. and Gáspár, C. (1992), "Fast, adaptive approximation of wind-induced horizontal flow patterns in shallow lakes using quadtree-based multigrid method", *C.M.W.R. Denver.*

Kelkar, K. M. and Patankar, S. V. (1992), "Numerical prediction of vortex shedding behind a square cylinder", *Int. J. for Num. Methods in Fluids*, Vol. 14, pp. 327-341.

Kallinderis, Y. (1992), "Adaptive hybrid prismatic/tetrahedral grids", *Int. J. for Num. Methods in Fluids*, Vol. 20, pp. 1023-1037.

Kallinderis, Y. and Nakajima, K. (1994), "Finite element method for incompressible viscous flows with adaptive hybrid grids", *AIAA J.*, Vol. 32, pp. 1617-1625.

Kallinderis, Y. (1998), "Hybrid grids and their applications", *Chapter in 'Grid generation handbook' CRC Press*

Kawaguti, M. and Jain, P. C. (1966), "Numerical study of a viscous fluid flow past a circular cylinder", *J. Phys. Soc. Japan*, Vol. 21, pp. 2055-2065.

Kochavi, E. and Segev, R. (1991), "Numerical solution of field problems by non-conforming Taylor discretisation", *Appl. Math. Modelling*, Vol. 15, pp. 152-157.

Kondo, N., Tosaka, N. and Nishimura, T. (1988), "high Reynolds solutions of the Navier-stokes equations using the third-order upwind finite element method", *Proc. Compu. Methods in Flow Analysis, Okayama University of Science, Okayama*, pp. 984-991.

Langtangen. H. P., Rusten, T., Tveito, A. and Wille, S. Ø. (1986), "An element by element preconditioner for iterative equation solvers", *Proceedings of the VI International Conference on Finite Elements in Water Resources. Lisbon. Portugal.*

Lax, P. D. (1989), "The review and prospects of American applied mathematics", *Chinese Society of Theo. and Applied Mech. Press, Beijing, Chinese J. Beijing, China*, Vol. 5, pp. 591-597.

Le, H, Troff, B., Sagaut, P., DangTran, K. and Phuoc, LT. (1997), " A Navier-Stokes solver for direct numerical simulation of incompressible flows", *Int. J. for Num. Methods in Fluids*, Vol. 24, pp. 833-861.

Lecointe, Y. and Piquet, J. (1984), "On the use of several compact methods for the study of unsteady incompressible viscous flow around a circular cylinder", *Comput. Fluids*, Vol. 12, pp. 255-280.

Löhner, R. and Ambrosiano, J. (1990),"A vectorized particle tracer for unstructured grids", *J. Comput. Phys.*, Vol. 91, pp. 22-31.

Löhner, R. and Morgan, K. (1987),"An unstructured multigrid method for elliptic problems", *Int. J. for Num. Methods in Eng.*, Vol. 24, pp. 101-115.

Löhner, R. and Parikn, P. (1989),"Generation of three-dimensional unstructured grids by the advancing-front method", pp. 1-12.

Massey, B. S. (1980),"Mechanics of fluids ", *Van Nostrand Reinhold.*

Mavriplis, D. J. (1989), "Adaptive mesh generation for viscous flows using Delaunay triangulation", *J. Comput. Phys., in the press.*

Mavriplis, D. J. (1991), "Turbulent flow calculations using unstructured and adaptive meshes", *Int. J. for Num. Methods in Fluids*, Vol. 13, pp. 1131-1152.

Min, B. K. and Chang, K. S. (1998), "A momentum coupling method for the unsteady incompressible Navier-Stokes equations on the staggered grid", *Int. J. for Num. Methods in Fluids*, Vol. 28, pp. 443-460.

Minyard, T. and kallinderis, Y. (1998),"Octree partitioning of hybrid grids for parallel adaptive viscous flow simulations", *Int. J. for Num. Methods in Fluids*, Vol. 26, pp. 57-78.

Mittal, S., Kumar, V. and Raghuvanshi, A. (1997), "Unsteady incompressible flows past two cylinders in tandem and staggered arrangements ", *Int. J. for Num. Methods in Fluids*, Vol. 25, pp. 1315-1344.

Morgan, K., Hassan, O. and Peraire, J. (1994), "An unstructured grid algorithm for the solution of Well's equations in the time domain", *Int. J. for Num. Methods in Fluids*, Vol. 19, pp. 849-863.

Morzynski, M. and Thiele, F. (1991), "Numerical stability analysis of a flow about a cylinder", Z. *Angew. Mech.*, Vol. 71, pp. T424-T428.

Naruhn, J. A., Welton, T. A. and Wong, C. Y. (1976), "Remarks on the numerical solution of Poisson's equation for isolated charge distributions", *J. Comput. Phys.*, Vol. 20, pp. 326-335.

Ohmori, K. and Ushijima, T. (1984), "A technique of upstream type applied to a linear non-conforming finite element approximation of convective diffusion equations", *RAIRO Anal. Numer.*, Vol. 18, pp. 309-332.

Okajima, A., Ueno, H. and Sakai, H. (1992), "Numerical simulation of laminar and turbulent flows around rectangular cylinders", *Int. J. for Num. Methods in Fluids*, Vol. 15, pp. 999-1012.

O'Leary, R. A. and Mueller, T. J. (1969), "Correlation of physical and numerical experiments for incompressible laminar separated flow", *Project THEMIS Technical Report, UND-69-4, Univ. of Notre Dame, Ind. Coll. of Engineering.*

Parthasarathy, V. and Kallinderis, Y. (1996), "Adaptive prismatic-tetrahedral grid refinement and redistribution for viscous flows", *AIAA J.*, Vol. 34, pp. 707-716.

Patankar, S. V. and Spalding, D.B. (1970), "Heat and mass transfer in boundary layers", *Second edition, Intertext books, London.*

Patel, V. A. (1976), "Time dependent solutions of the viscous incompressible flow past a circular cylinder by the method of series truncation", *Computers and Fluids*, Vol. 4, pp. 13.

Pelletier, D., Ilinca, F. and Turgeon, É. (1997), "An adaptive finite element method for forced convection", *Int. J. for Num. Methods in Fluids*, Vol. 25, pp. 803-823.

Peraire, J., Vahdati, M., Morgan, K. and Zienkiewicz, O. C. (1987),"Adaptive remeshing for compressible flow computations", *J. Comput. Phys.*, Vol. 72, pp. 449-466.

Pirzadeh, S. (1994), "Viscous unstructured three-dimensional grids by the advancing-layers method ", *AIAA Paper*, 94-0417.

Rhie, B. M. and Chow, W. L. (1983), "Numerical study of the turbulent flow past airfoil with trailing edge separation", *AIAA J.*, Vol. 21, pp. 1525-1532.

Rizzi, A. (1982), "Damped Euler equation method to compute transonic flow around wing-body combinations", *AIAA J.*, Vol. 20, pp. 1321-1328.

Rogers, S. E. and Kwak, D. (1988), "An upwind differencing scheme for the time-accurate incompressible Navier-Stokes equations", *AIAA Paper*, 88-2583.

Roshko, A. (1954), "On the development of turbulent wakes from vortex streets", *NACA Report*, 1191.

Saalehi, A. (1995), "Quadtree-based finite element modelling of laminar separated flow past a cylinder", *DPhil thesis, University of Oxford.*

Samet, H. (1990a), "The design and analysis of spatial data structures", *Addison-Wesley.*

Samet, H. (1990b), "Applications of spatial data structure", *Addison-Wesley.*

Sampaio, P. A. B. de, Lyra, P. R. M., Morgan, K. and Eeatherill, N. P. (1993), "Petrov-Galerkin solution of the incompressible Navier-Stokes equations in primitive variables with adaptive remeshing", *Comput. Meth. App. Mech. Eng.*, Vol. 106, pp. 143-178.

Schmidit, R. J. (1991), "Adaptive discretization for fluid flow problems", *PhD. thesis, Rensselaer Polytechnic Institute, New York.*

Schumann, U. (1976), "A direct method for the solution of Poisson's equation with Neumann boundary conditions on a staggered grid of arbitrary size", *J. Comput. Phys.*, Vol. 20, pp. 171-182.

Shamos, M. I. (1975), "Geometric complexity", Proceedings of the Seventh of Computing, pp. 224-233.

Sivaloganathan, S. and Shaw, G. J. (1988), "A multigrid method for recirculating flows", *Int. J. for Num. Methods in Fluids,* Vol. 8, pp. 417-440.

Sloan, S. W. and Houlsby, T. (1984), "An implementation of Watson's algorithm for computing 2-dimensional Delaunay triangulation", *Adv. Eng. Softw.*, Vol. 6, pp. 192-197.

Sloan, S. W. (1987), "A fast algorithm for constructing Delaunay triangulations in the plane", *Adv. Eng. Softw.*, Vol. 9, pp. 34-55.

Son, J. S., and Hanratty, T. J. (1969), "Numerical solution for the flow around a cylinder Reynolds numbers of 40, 200 and 500", *J. Fluid Mech.*, Vol. 35, pp. 369

Sonneveld, P. (1987), "A fast Lanczos-type solver for non-symmetric linear systems", *SIAM. J. Stat. Comput.*, Vol. 10, pp. 36-52.

Speares, W. and Berzins, M. (1997), "A 3D unstructured mesh adaptation algorithm for time-dependent shock-dominated problems", *Int. J. for Num. Methods in Fluids*, Vol. 25, pp. 81-104.

Tabata, M. (1986), "A theoretical and computational study of upwind-type finite element methods", *Int. Nishida et al.(eds), Patterns and Waves, kinokuniya/North-Holland, Tokyo*, pp. 319-356.

Tabata, M. and Fujima, S. (1991), "An Upwind Finite element scheme for high-Reynolds number flows", *Int. J. for Num. Methods in Fluids*, Vol. 12, pp. 305-322.

Tanida, Y., Okajima, A. and Watanabe, Y. (1973), "Stability of a circular cylinder oscillating uniform flow or in a wake", *J. Fluid Mech.*, Vol. 61, pp. 769

Taniguchi, T., Holz, K. P. and Ohta, C. (1992), "Grid generation for 2D flow problems", *Int. J. for Num. Methods in Fluids*, Vol. 15, pp. 985-997.

Thom, A. (1928), *Aero. Res. Counc. R. & M. no. 1194.*

Thom, A. (1933), "The flow past circular cylinders at low speeds", *Proc. Roy. Soc. A 141, 65.*

Thomadakis, M. and Leschziner, M. (1996), "A pressure-correction method for the solution of incompressible viscous flows on unstructured grids", *Int. J. for Num. Methods in Fluids*, Vol. 22, pp. 581-601.

Thomasset, F. (1981), "Implementation of finite element methods for Navier-stokes equations" *Springer, New York.*

Thompson, J. F. (1982*b*), "Elliptic grid generation, numerical grid generation", *Elsevier Science, pp. 79-105.*

Tan, M., Farrant, T. and Price, W. G. (1999), "A cell boundary-element method for viscous laminar flow solutions ", *Proc. R. Soc. Lond. A 455*, pp. 4277-4304.

Thompson, J. F. (1987), "A general three dimensional elliptic grid generation system on a composite block-structure", *Comput. Meth. App. Mech. Eng.*, Vol. 64, pp. 377-411.

Tritton, D. J. (1959), "Experiments on the flow past a circular cylinder at low Reynolds numbers", pp. 547-561.

Van. Dommelen, L., and Rindensteiner, E. A. (1989), "Fast adaptive summation of point forces in the two-dimensional Poisson equation ", *J. Comput. Phys.*, Vol. 83, pp. 126-147.

Versteeg, H. K. and Malalasekera, W. (1995), "Computational Fluid Dynamics: The Finite Volume Method ", *Longman Scientific & Technical Press, U.K.*

Viecelli, J. A. (1969), "A method for arbitrary external boundaries in the MAC incompressible fluid computing technique", *J. Comput. Phys.*, Vol. 4, pp. 543-551.

Voronori, G. (1908), "Nouvelles applications des parametres continus a la theorie des formes quadratiques. Recherches sur les parallelloedres primitifs", *J. R. Angew. Math.* 134.

Wan, D.C., Liu, Y. Z. and Miao, G.P. (1996), "The interactions between wave-currents and offshore structures with consideration of fluid viscosity", *Chinese Society of Theo. and Chinese J. of Mech. Press, Beijing, China*, Vol. 12, No. 4, Nov.

Watson, D. F. (1981), "Computing non-dimensional Delaunay tesselation with application to Voronoi polytopes", *Comput. J.*, Vol. 24, pp. 167-172.

Weatherill, N. P. and Forsey, C. R. (1985), "Grid generation and flow calculations for aircraft geometry ", *J. Aircraft,* Vol. 22, pp. 855-860.

Weatherill, N. P. and Marchant, M. J. (1994), "Grid adaptation using a distribution of sources applied to inviscid compressible flow simulations", *Int. J. for Num. Methods in Fluids,* Vol. 19, pp. 739-764.

Westermann, T. (1992), "Localization schemes in 2D-boundary-fitted grids", *J. Comput. Phys.*, Vol.101, pp.307- 313.

Wille, R. (1960), "Kámán vortex streets", *Adv. Appl. Meth.*, Vol.6, pp.273- 281.

Wille, S. Ø. (1992), "A Structured tri-tree search method for generation of optimal unstructured finite element grids in two and three dimensions", *Int. J. for Num. Methods in Fluids*, Vol. 14, pp. 861-881.

Wille, S. Ø. (1995), "A tri-tree multigrid recoarsement algorithm for the finite element formulation of the Navier-Stokes equations", *Comput. Meth. App. Eng.*, Vol. 135, pp. 129-142.

Wille, S. Ø. (1996), "The prolonged adaptive multigrid method for finite element Navier-Stokes equations", *Comput. Meth. App. Eng.*, Vol. 138, pp. 227-271.

Wille, S. Ø. (1995), "A local predictive convection-diffusion refinement indicator for the tri-tree adapted finite element multigrid algorithm of the Navier-Stokes equations", *Comput. Meth. App. Eng.*, Vol. 134, pp. 181-196.

Wille, S. Ø. (1997), "A non-linear adaptive full tri-tree multigrid method for the mixed finite element formulation of the Navier-Stokes equations", *Int. J. for Num. Methods in Fluids,* Vol. 24, pp. 1037-1047.

Wille, S. Ø. (1998), "Nodal operator splitting adaptive finite element algorithms for the Navier-Stokes equations", *Int. J. for Num. Methods in Fluids,* Vol. 26, pp. 959-975.

Williamson, C. H. K. (1989), "Oblique and parallel modes of vortex shedding in the wake of a circular cylinder at low Reynolds numbers", *J. Fluid Mech.*, Vol. 206, pp. 579-627.

Wygnanski, L., Champagne, F. and Marasli, B. (1986), "On the large-scale structures in two-dimensional, small-deficit, turbulent wakes", *J. Fluid Mech.*, Vol. 168, pp. 31-71.

Yahia, D. A. A., Habashi, W. G. and Tam, A. (1996), "A directionally adaptive methodology using an edge-based errors estimate on quadrilateral grids", *Int. J. for Num. Methods in Fluids,* Vol. 23, pp. 673-690.

Yu, N. T., Kusunose, K., Chen, H. C. and Sommerfield, D. M. (1987), "Flow simulations for a complex aeroplane configuration using Euler equations", *AIAA Paper,* 87-0454.

Zdravkovich, M. M. (1977), "Review of flow interference between two circular cylinders in various arrangements", *J. Fluids Eng. Trans. ASME,* Vol. 99, pp. 618-633.

Zhang, J. and Dalton, C. (1998), "A three-dimensional simulation of a steady approach flow past a circular cylinder at low Reynolds number ", *Int. J. for Num. Methods in Fluids*, Vol. 26, pp. 1003-1022.

Zienkiewicz, O. C. and Taylor, R. L. (1991), "The finite element method", *McGraw-Hill Book Company*, Vol. 1 and Vol. 2.

Zienkiewicz, O. C. and Zhu, J. Z. (1987), "A simple error estimator and adaptive procedure for practical engineering analysis ", *Int. J. Num. Methods eng.* Vol. 24, pp. 337-357.

Zienkiewicz, O. C. and Zhu, J. Z. (1992), "The super-convergent patch recovery and a posteriori error estimates. Part1: The recovery technique", *Int. J. Num. Methods eng.* Vol. 33, pp. 1331-1364.

Zienkiewicz, O. C. and Zhu, J. Z. (1992), "The super-convergent patch recovery and a posteriori error estimates. Part2: Error estimates and adaptivity", *Int. J. Num. Methods eng.* Vol. 33, pp. 1365-1382.