A Coupled Eulerian/Lagrangian Method for the Solution of Three-Dimensional Vortical Flows

by

Helene M. Felici

GTL Report #211 June 1992

Research funding for this work was provided by NASA Ames Research Center with Mr. Paul Stremel as Technical Monitor, by the Office of Naval Research with Mr. James A. Fein as Technical Monitor, and by the NSF PYI program. Additional funding was provided by the Swiss National Foundation for Scientific Research.

A COUPLED EULERIAN/LAGRANGIAN METHOD FOR THE SOLUTION OF THREE-DIMENSIONAL VORTICAL FLOWS

by
HELENE MARIE FELICI

A coupled Eulerian/Lagrangian method is presented for the reduction of numerical diffusion observed in solutions of three-dimensional rotational flows using standard Eulerian finite-volume time-marching procedures. A Lagrangian particle tracking method using particle markers is added to the Eulerian time-marching procedure and provides a correction of the Eulerian solution. In turn, the Eulerian solution is used to integrate the Lagrangian state-vector along the particles trajectories. The Lagrangian correction technique does not require any a-priori information on the structure or position of the vortical regions. While the Eulerian solution ensures the conservation of mass and sets the pressure field, the particle markers, used as 'accuracy boosters', take advantage of the accurate convection description of the Lagrangian solution and enhance the vorticity and entropy capturing capabilities of standard Eulerian finite-volume methods.

The combined solution procedure is tested in several applications. The convection of a Lamb vortex in a straight channel is used as an unsteady compressible flow preservation test case. The other test cases concern steady incompressible flow calculations and include the preservation of a turbulent inlet velocity profile, the swirling flow in a pipe, the constant stagnation pressure flow and secondary flow calculations in bends. The last application deals with the external flow past a wing with emphasis on the trailing vortex solution.

The improvement due to the addition of the Lagrangian correction technique is measured by comparison with analytical solutions when available or with Eulerian solutions on finer grids. The use of the combined Eulerian/Lagrangian scheme results in substantially lower grid resolution requirements than the standard Eulerian scheme for a given solution accuracy.

Acknowledgments

First of all, many thanks to my advisor, Prof. Mark Drela, for the support he provided throughout this thesis. Being exposed to his innovative and practical spirit was quite an experience. Also, I would like to thank the other members of my committee, Prof Mike B. Giles for his availibility for 'very urgent questions', Prof. Alan H. Epstein for his pertinent suggestions and constructive criticism and Prof. Anthony T. Patera for his comments.

Helpful suggestions were also provided by the readers for this thesis, Profs. Mårten T. Landahl and Manuel Martinez-Sanchez.

To Prof I. L. Ryhming, who was the instigator of the move to MIT, my sincere thanks for the sound advice.

Thanks to everybody in the CFD Lab for generally creating a nice working atmosphere and putting up with the 'extensive' use of disk space and CPU time.

To my companion in all adventures and special person in my life, you and I both know how much your constant support contributed to this work.

Research funding for this work was provided by NASA Ames Research Center with Mr. Paul Stremel as Technical Monitor, by the Office of Naval Research with Mr. James A. Fein as Technical Monitor and by the NSF PYI program. Additional funding to support my education was provided by the Swiss National Foundation for Scientific Research.

Contents

List of Figures

Nomenclature

a	Lamb vortex core radius
c	speed of sound
c_h	airfoil chord
c_a^2	artificial compressibility parameter
C	cost function
Cp	pressure coefficient (Eq.12.1)
Cp_0	stagnation pressure coefficient (Eq.12.2)
Cp_v	pressure coefficient for Lamb vortex (Eq.7.6)
d	pipe diameter
D^2	pseudo-Laplacian operator
e_0	total energy per unit mass
F, G, H	fluxes in Cartesian coordinates
J	Jacobian
h_0	total enthalpy per unit mass
p	static pressure
p^*	reduced pressure for incompressible flows ($p^* = p/\rho$)
R	radius
Re	Reynolds number (Eq.8.1)
S	entropy related function (Eq.4.4)
S_x	projected area on the yz plane of a cell face
S_y	projected area on the xz plane of a cell face
S_z	projected area on the xy plane of a cell face
t	time
T_l	Lagrangian source term
$\vec{r}$	position vector in Cartesian coordinates
u, v, w	Cartesian velocity components
v_θ	tangential velocity
u^*	friction velocity (Eq.8.3)
u^+	velocity referenced to friction velocity (Eq.8.2)
U	state vector of Eulerian conservative variables
U_l	state vector of Lagrangian variables
U_p	vector of perturbed primitive variables
V	cell volume
V_n	node volume
x, y, z	Cartesian coordinates
y^+	Reynolds number based on u^* (Eq.8.4)

Greek:

β	pseudo-speed of sound (Eq.2.6)
ΔC_{p_0}	local error in stagnation pressure (Eq.3.75)
ϵ_{p_0}	L_2 norm of errors in stagnation pressure (Eq.3.74)
γ	ratio of specific heats
Γ	circulation
λ	friction coefficient
ν	kinematic viscosity
ν_2	second-difference numerical smoothing coefficient
ν_4	fourth-difference numerical smoothing coefficient
ν_l	Lagrangian pseudo-diffusion coefficient
ρ	static density
Δt	time-step
ψ	grid weight function
Φ	vector of linearized characteristic variables
$\vec{\omega}$	vorticity vector
ξ, η, ζ	local coordinates

Superscripts:

n	time index

Subscripts:

c	corrected quantity
e	Eulerian quantity
kc	cell averaged
in, out	inlet, outlet
i, n	node index
l	Lagrangian quantity
max	maximum quantity
min	minimum quantity
p	predicted quantity
x, y, z	x, y, z components
0	total (stagnation) conditions
∞	freestream conditions

Accents:

$\vec{}$	vector quantity

Chapter 1

Introduction

1.1 Statement of the problem

Over the last few years, the improvement in CPU and memory capabilities of modern supercomputers has rendered practical the solution of flow problems of more and more complex nature. However, the efficient numerical treatment of flow non-homogeneities, such as vortex wakes or tip vortex roll-up, embedded in an otherwise smooth background flow field remains a challenging field of study. In many practical applications, the prediction of the strength and the position of the vortical regions reveals to be of primary importance. For instance, the flow around an helicopter rotor blade presents a case of strong interaction between the shed vortices due to one blade and the following blade. The prediction of the resulting load variations requires the accurate solution of the shed vortices, of their trajectories and of the subsequent interaction phenomenon. Another example is the prediction of the secondary flow through a bend with a pump attached at the bend exit. The location and strength of the secondary vortex, created by the tilting and stretching of the inlet boundary-layer vorticity, must be solved accurately, as a possible noise source and a performance loss may result from the impingement of the secondary vortex on the rotating blades.

Vortex-sheets, secondary flows, or vortex roll-up phenomena are all characterized by transverse length scales differing by orders of magnitude from the length scale of the supporting flow field (the transverse length scale of trailing vortices has been found experimentally to be as low as 5% of the airfoil chord [57]). Since the solution of these vortical features has often to cover a convection length much higher than their intrinsic length scale, the global prediction of vortex-dominated flows proves to be highly sensitive to small local errors. This makes these flow features difficult to be captured by finite-difference methods.

1.2 Existing approaches

Incompressible vortex methods and potential methods with fitted vortex sheets are not susceptible to the numerical diffusion. Examples of incompressible vortex methods, using Biot-Savart law to compute the velocity field, include the method used by Leonard [41] where the flow vorticity is modeled as a collection of a few isolated vortex-tubes with a computational element assigned to each vortex tube and Knio's study [38] where a three-dimensional vortex scheme is based on the transport of vorticity and material elements. Additionally, gradients of the scalar field are transported and the scalar field itself is recovered using Biot-Savart law.

Potential methods presuppose some a priori knowledge (either from a known solution or from empirical data) on the vortex structure or position, since potential methods do not 'capture' embedded vorticity as part of the solution. This limitation becomes especially acute when solving complex flow problems where an a priori information is not always available. Scully [64] and Miller [44] have used a Biot-Savart formulation for an incompressible flow solution of an helicopter rotor wake. Hassan [25] used the Euler equations in an implicit scheme and modeled the blade-vortex interaction by computing the vortex-induced velocities following Biot-Savart law. Steinhoff [71, 72] presented an alternative method for an aircraft configuration, where the strength, position and shape of the vortex sheet were calculated as part of the solution. The internal structure of the wake was, however, still to be specified. Ramachandran [54] used a potential method with embedded vortex wakes for the compressible flow solution of a rotor wake. The body was included in the calculation similarly as the rotor wake as a vorticity sheet whose strength is determined iteratively.

The use of an Euler or a Navier-Stokes solver presents the advantage that the embedded vorticity is captured as part of the solution. Poor solution representation is, however, a common feature of Eulerian and Navier-Stokes solvers in regions of high gradients. More precisely, the errors introduced by discretizing the equations of motion can be expressed in terms of dissipation and dispersion phenomena. In addition, rounding errors are introduced randomly in the solution. The dissipation expresses the fact that the finite difference model loses energy as the time progresses. Because numerical dissipation can be advantageous by counteracting unwanted instabilities and oscillations (for example saw-tooth modes), it is added to non-dissipative formulae. The dispersion errors correspond to the decay of a wave form into separate spurious oscillations and always occur with finite-difference formulae since their dispersion relation is always non-linear.

Because most common second-order accurate finite-difference schemes will smear and distort regions of high gradients, corresponding grid clustering seems the obvious approach. However, depending on the flow topology, this procedure could reveal to be prohibitively expensive. For example, the solution of the flow around helicopter blades involves the prediction of the interaction between the tip vortex from a blade and the following blade. Because the resulting flow presents vortex regions of high extent and is highly non-homogeneous, standard clustering of the grid would lead to high computational cost. As Drela [20] reported, a suitably overall fine grid would imply $\sim$ 260 billion points for a rotor wake solution. The sensitivity of the solution to the grid coarseness has, therefore, prompted several studies.

Selective refinement of the grid is used by Lõhner [42] in an adaptive algorithm in two dimensions. In three dimensions, however, the algorithm would undoubtedly present much complexity. Nakahashi [46] uses a 2-D solution adaptive structured grid method based on variational principles and spring analogies. A multigrid solution of the Euler equations using an implicit scheme has been performed by Jameson [37] and has proven effective for CPU reduction in two-dimensional applications. The three-dimensional application would certainly prove to be more involved. Landsberg [40] has studied vortex capturing using adaptation and a three-dimensional finite-element solver. Also Schmatz [63] presented a two-dimensional zonal solution to model the weak or strong viscous/inviscid interactions in subsonic and transonic flows. Powell [49] used an adaptive mesh procedure working on an unstructured mesh for solving the conical Euler equations for leading-edge vortex flows. In this respect, unstructured grids present a strong advantage over structured grids because of the flexibility of adding new grid resolution in defined areas.

An alternative to grid refinement is to use a high-order accurate scheme, a method used by Rai [53] who presented a fifth-order upwind-biased scheme in a blade/vortex interaction problem. Steger [70] used an implicit fourth-order accurate scheme for the computation of vortex wakes. However, the advantage of using these high-order accurate schemes, with suitable clustering of points in the regions of high gradients, is still linked to grid smoothness (more difficult to obtain in three dimensions) and were demonstrated on grids prohibitively fine for complex three-dimensional applications.

Perturbation methods, instead, rely on a known[1] flow solution in some areas of the computational domain such as to correct the numerical diffusion encountered in the basic finite-difference flow solution. Roberts [57, 56] applied this methodology, first introduced by Chow [15] to the rotor wake and blade/vortex interaction problems by coupling the Euler equations to a free-wake model of the rotary wing wake. The main drawback here is the need for a known solution which limits the correction method to regions of simple behavior (the correction can not be performed where the vortex impinges on the airfoil for instance). In a similar approach, Srinivasan [69, 68] uses the 2-D thin-layer Navier-Stokes equations and a prescribed vortex for computing the flow

[1] This is done from an analytic solution if available or from a previously computed high resolution local solution of the flow field.

over rotor blades.

The 'Cloud in Cell' technique, first introduced by Christiansen [16] and then transformed by Baker [5] for a 2-D incompressible inviscid fluid, uses an area averaged vorticity distribution from markers in cells onto grid nodes where Poisson's equation is solved. The circulation distribution must, however, be known at each marker and a three-dimensional solution is not straightforward. Also Basuki [7] used the inviscid 'Cloud in Cell' technique with vortices tracked though the grid on which the velocity is found by a finite-difference method. Poor resolution of the velocity field was, however, reported.

The advantages of spectral methods are accuracy, ease of implementation and the low number of collocation points required for a computation when compared to the discretization used in finite-difference methods. However, they lead to large matrices for more than one non-periodical direction (which is the case for the flow cases treated here) and are difficult to apply in computational domains of complex shapes. Also, no discontinuity (as a shock, for example) is allowed as part of the solution. Furthermore they are more time restrictive than finite-difference methods for unsteady flow calculations.

1.3 Present approach

As mentioned in the previous section, the Euler or Navier-Stokes equations present the advantage of directly capturing embedded vorticity, when compared to potential methods with fitted sheets. For example, as reported by Murman [45] and others [50, 9], the use of the Euler equations provides a good tool for a study of flows around wings enabling the study of the leading-edge vortex. The capturing capabilities of Euler or Navier-Stokes solvers are, nevertheless, limited by grid resolution issues. As an alternative to the methods dealing with this problem, this thesis presents a technique for substantially improving the capturing capabilities of time-marching Eulerian solvers.

This method does not require the knowledge of an a priori solution, grid refinement or the use of higher-order schemes.

The objective of this research is then to construct an alternative solution procedure to reduce the numerical diffusion observed in standard Eulerian time-marching calculations and to demonstrate the feasibility, efficiency and flexibility of the method by application to different flow problems. These include three-dimensional steady, unsteady, internal as well as compressible and incompressible inviscid flow cases. Furthermore, the extension of the method to include the Navier-Stokes equations (not performed in the frame of this thesis) is judged to be straightforward.

The present method consists in the addition of a Lagrangian particle tracking solution to a standard Eulerian solution in order to enhance its vorticity and entropy capturing capabilities. This method is based on the approach of Drela [20] in two dimensions and is here extended to include three-dimensional flow cases. The combination of the Eulerian and Lagrangian solvers takes advantage of both the accurate convection description of the Lagrangian technique and the 'elliptic' representation of the Eulerian solution which enforces the mass conservation and sets the pressure field. Briefly, the Lagrangian solution is based on particle markers carrying vorticity and entropy, and convecting with the local flow through the Eulerian grid. The Eulerian solver is used to conserve mass and to provide the source terms required for the Lagrangian time integration. In turn, since the Lagrangian solution is immune to the numerical diffusion process occurring in the Eulerian solver, it accurately captures the convection of vorticity and entropy. This information is then used to locally correct the Eulerian solution and to reduce its numerical diffusion errors. Each Lagrangian marker influences the Eulerian solution only locally (as opposed to the 'Cloud in Cell' technique where each marker has an influence on the entire flow field), which makes this scheme well suited for three-dimensional flow solutions. Also, the Lagrangian solution needs to be computed only in regions of interest as markers can be located selectively in the flow. No a priori information is required on the flow structure since the Lagrangian solution includes inherently 'convective' capabilities.

1.4 Thesis outline

Chapters 2 and 3 present the equations, numerical procedure and accuracy study for the Eulerian solver for both the compressible and incompressible flow cases. The mesh generation technique is described in Appendix A. The Lagrangian equations are the object of Chapter 4, whereas the coupling of the two solvers in the time integration is described in Chapter 5 for different flow configurations. Finally, the correction procedure by which the Lagrangian equations influence the Eulerian solution is presented in Chapter 6.

The first test case, discussed in Chapter 7, is the convection of a Lamb vortex in a three-dimensional uniform background flow and is used as a preservation test for a compressible unsteady flow. An analogous test case is presented in Chapter 8 as the preservation of a turbulent inlet velocity profile in a straight pipe. The swirling flow in a straight pipe is the object of Chapter 9 with an emphasis on the development of a vorticity gradient augmentation phenomenon and the particular solution adopted with the combined Eulerian/Lagrangian solver. The vorticity errors and stagnation pressure losses encountered in the Eulerian solution of a 90^0 bend are reduced by the use of the Lagrangian correction method in Chapter 10.

Chapter 11 deals with the secondary flow in bent pipes. The secondary flow genesis is first described and Eulerian and Eulerian/Lagrangian solutions are computed and compared with experiments. The introduction of a simple 'law of the wall' model is attempted in order to deal with viscosity effects.

The last case, reported in Chapter 12, is the external flow over a three-dimensional wing. This chapter emphasizes the spurious numerical diffusion of the tip vortex behind the trailing edge and the correction obtained using the combined Eulerian/Lagrangian scheme. Comparison with experimental data is also performed.

Finally, Chapter 13 presents a summary, the contributions as well as the conclusions and the recommendations for future work.

Chapter 2

Eulerian Governing Equations

Provided the tangential forces applied on fluid particles are small compared to the pressure forces, the fluid can be treated as inviscid. The evolution of an inviscid flow in time and space is described by the Euler equations [60, 3]. Here, these equations are used for the solution of both steady incompressible and unsteady compressible flow fields. The numerical solution procedure using a Lax-Wendroff scheme is the object of the next chapter.

2.1 Euler equations for compressible flow

For the solution of unsteady compressible flows, the solution is marched forward in time from an initial condition. The Euler equations expressed in a (x, y, z) Cartesian coordinates system and in conservation form are

$$\frac{\partial U}{\partial t} = \frac{\partial F}{\partial x} + \frac{\partial G}{\partial y} + \frac{\partial H}{\partial z}, \tag{2.1}$$

where U, the state vector of the conservative variables and F, G, H, the fluxes of mass, momentum and energy are written as

$$U = \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ \rho e_0 \end{pmatrix}, F = \begin{pmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ \rho uw \\ \rho u(e_0 + p/\rho) \end{pmatrix}, G = \begin{pmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ \rho vw \\ \rho v(e_0 + p/\rho) \end{pmatrix}, H = \begin{pmatrix} \rho w \\ \rho uw \\ \rho vw \\ \rho w^2 + p \\ \rho w(e_0 + p/\rho) \end{pmatrix}.$$

ρ denotes the fluid density, u, v, w are the velocity components in the Cartesian coordinates, p is the pressure and e_0 is the total energy per unit mass. Additionally the perfect gas law is used to relate the total energy per unit mass to the pressure as

$$p = (\gamma - 1)\rho(e_0 - \frac{1}{2}(u^2 + v^2 + w^2)), \tag{2.2}$$

where γ is the ratio of the specific heats. The speed of sound c and the total enthalpy per unit mass h_0 are defined by

$$c = \sqrt{\frac{\gamma p}{\rho}}, \quad h_0 = e_0 + \frac{p}{\rho}. \tag{2.3}$$

2.2 Euler equations for incompressible flow

For an incompressible unsteady flow, the state vector U and the fluxes F, G and H are written as

$$U = \begin{pmatrix} 0 \\ u \\ v \\ w \end{pmatrix}, \quad F = \begin{pmatrix} u \\ u^2 + p^* \\ uv \\ uw \end{pmatrix}, \quad G = \begin{pmatrix} v \\ uv \\ v^2 + p^* \\ vw \end{pmatrix}, \quad H = \begin{pmatrix} w \\ uw \\ vw \\ w^2 + p^* \end{pmatrix}, \tag{2.4}$$

where ρ is a constant and the ratio p/ρ is denoted p^*.

The main problem in solving the Euler equations for an incompressible flow is to link the velocity changes to the pressure changes in a way that enforces the divergence-free condition. In two dimensions, the solution of steady or unsteady incompressible flows can be achieved through the stream function-vorticity formulation (therefore eliminating the pressure from the governing equations). In three-dimensional flow calculations, however, this solution technique becomes more complex and other solution procedures are usually sought. The vorticity-velocity formulation used by Dennis [17] replaces the two-dimensional stream function-vorticity formulation. The Poisson's equation method developed by Harlow [24] consists of iteratively adjusting the pressure field by solving a Poisson type equation for the pressure change. Poisson's equation is obtained from the requirement that the continuity equation must be satisfied. Since this method involves an iterative procedure it is, however, very time consuming for three-dimensional applications.

A well-known class of solution procedures for steady compressible flows is the time-marching method where the full unsteady Euler equations are used and the solution evolves through a pseudo-unsteady process from an initial guess to the final steady-state. Nevertheless, in the limit of an incompressible flow, sound waves with very large speed tend to make the system stiff and render this inefficient. A well-known solution to this problem, and the method used in this work, is the artificial compressibility concept introduced by Chorin [14]. The purpose of this technique is to transform the character of the Euler equations for an incompressible flow from elliptic to hyperbolic by adding a time-dependent term in the continuity equation. This particular method has been successfully tested on an extensive set of internal and external incompressible flow problems [58, 13, 55, 61, 79].

To introduce a time-derivative of the pressure in the continuity equation, the divergence term is multiplied by the "artificial compressibility" parameter c_a^2 so that the modified state and flux vectors are defined as

$$U = \begin{pmatrix} p^* \\ u \\ v \\ w \end{pmatrix}, \quad F = \begin{pmatrix} c_a^2 u \\ u^2 + p^* \\ uv \\ uw \end{pmatrix}, \quad G = \begin{pmatrix} c_a^2 v \\ uv \\ v^2 + p^* \\ vw \end{pmatrix}, \quad H = \begin{pmatrix} c_a^2 w \\ uw \\ vw \\ w^2 + p^* \end{pmatrix}. \qquad (2.5)$$

The value of the artificial compressibility parameter can be adjusted to increase the convergence rate of the time-marching procedure. When steady-state is reached, the modified system of equations reduces to the standard Euler equations for steady flow. Also pseudo-time stepping can be used since the unsteady process is of no interest here and only the steady-state solution is retained.

The introduction of the artificial compressibility parameter results in giving finite speed to the propagating waves, in contrast to truly incompressible flow where the waves move with an infinite speed. The pseudo-speed of sound β is computed in Section 3.4 by analyzing the linearized Euler equations with 1-D variations. β depends on the artificial compressibility parameter c_a^2 as

$$\beta = \sqrt{u^2 + c_a^2}. \qquad (2.6)$$

Chang [13] estimates the relation between the parameter c_a^2 and the rate of convergence by looking at the speed of the propagating waves. The time taken by a wave to travel from the inlet of the computational domain to the exit over a distance L and back is

$$t = \frac{L}{\beta + u} + \frac{L}{\beta - u} = \frac{2\beta L}{c_a^2} \qquad (2.7)$$

This value represents the minimum time needed for convergence. If the time-step allowed for stability is Δt, then N the number of time-steps required becomes

$$N = \frac{2\beta L}{c_a^2 \Delta t} = \left(\frac{\sqrt{u^2 + c_a^2}}{c_a^2} \right) \frac{2L}{\Delta t}, \qquad (2.8)$$

a decreasing function of c_a^2. Regarding the value of the artificial compressibility parameter, it is shown in Section 3.4 that the ratio of the largest to the smallest eigenvalues of the linearized Euler system of equations is dependent on β. β is therefore a measure of the condition of the system. Rizzi [55] has verified numerically that a ratio c_a^2/u^2 between 1 and 5 ensures the system to be well-conditioned. In this particular study a constant value of c_a^2/u^2 of 1 is used.

Another advantage of the artificial compressibility method is its natural extension to the handling of the Navier-Stokes equations [13]. Also the same concept has been used for the solution of unsteady flows problems [43].

2.3 Non-dimensionalization

The Euler equations are used in a non-dimensionalized form which allows for the flow values to fall within prescribed limits. The arbitrary reference values are given below for the different flow cases treated in this work.

For the compressible flow in a channel the reference quantities are the channel length L, the inlet stagnation speed of sound $c_{0_{in}}$ and the inlet stagnation density $\rho_{0_{in}}$ so that the non-dimensional variables are

$$\begin{aligned}
x_r &= \frac{x}{L}, & y_r &= \frac{y}{L}, & z_r &= \frac{z}{L}, \\
t_r &= \frac{t}{L/c_{0_{in}}}, \\
u_r &= \frac{u}{c_{0_{in}}}, & v_r &= \frac{v}{c_{0_{in}}}, & w_r &= \frac{w}{c_{0_{in}}}, \\
p_r &= \frac{p}{\rho_{0_{in}} c_{0_{in}}^2}, & \rho_r &= \frac{\rho}{\rho_{0_{in}}} & e_{0r} &= \frac{e_0}{c_{0_{in}}^2}.
\end{aligned}$$

The corresponding reference inlet stagnation pressure and enthalpy are

$$(p_{0_{in}})_r = \frac{p_{0_{in}}}{\rho_{0_{in}} c_{0_{in}}^2} = \frac{1}{\gamma}, \qquad (h_{0_{in}})_r = \frac{\gamma}{\gamma - 1} \frac{(p_{0_{in}})_r}{(\rho_{0_{in}})_r} = \frac{1}{\gamma - 1}. \tag{2.9}$$

For incompressible flow in pipes, the reference quantities are the pipe radius R, and the inlet mass-flow averaged velocity $\overline{u_{in}}$. For the incompressible flow around an airfoil the reference length has been chosen as the airfoil chord c_h and the reference velocity is the freestream velocity U_∞. If the reference length and the reference velocity are denoted by L_{ref} and U_{ref}, then

$$\begin{aligned}
x_r &= \frac{x}{L_{ref}}, \qquad y_r = \frac{y}{L_{ref}}, \quad z_r = \frac{z}{L_{ref}}, \\
t_r &= \frac{t}{L_{ref}/U_{ref}}, \\
u_r &= \frac{u}{U_{ref}}, \qquad v_r = \frac{v}{U_{ref}}, \quad w_r = \frac{w}{U_{ref}}, \\
p_r^* &= \frac{p^*}{U_{ref}^2},
\end{aligned}$$

Chapter 3

Euler Solver: Numerical Procedure

3.1 Lax-Wendroff algorithm

The Euler solver uses a Ni-Lax-Wendroff node-based scheme on an unstructured grid. An explicit time-marching procedure subject to appropriate boundary conditions is used to drive the solution from an initial guess to a steady-state or to an unsteady solution.

The numerical procedure has been introduced by Ni [47] for two dimensions and has been described later by Ni and Bogoian [48] for a three-dimensional application on a structured grid. The present chapter deals with the algorithm description for unstructured meshes.

The spatial discretization uses hexahedral cells and the change in time of the state vector U is expressed as a function of the fluxes across the cell faces. These are evaluated as the average of the fluxes at the corner nodes. The residual (found from summing the fluxes across the six faces of each cell) is used to determine the change in the state vector and is then distributed back to the nodes following the Lax-Wendroff algorithm.

The state vector U at time-step $n+1$ can be expanded in a Taylor series up to the second-order terms as

$$U^{n+1} - U^n = \Delta t \left(\frac{\partial U}{\partial t}\right)^n + \frac{\Delta t^2}{2}\left(\frac{\partial^2 U}{\partial t^2}\right)^n, \tag{3.1}$$

and the time derivative of the state vector U is related to the spatial derivatives of the fluxes F, G, H by Equation (2.1)

$$\frac{\partial U}{\partial t} = -\frac{\partial F}{\partial x} - \frac{\partial G}{\partial y} - \frac{\partial H}{\partial z}. \tag{3.2}$$

Hence, the change in the state vector between two iterations is

$$\begin{aligned}\delta U = U^{n+1} - U^n = & - \Delta t \left[\frac{\partial F}{\partial x} + \frac{\partial G}{\partial y} + \frac{\partial H}{\partial z}\right]^n \\ & - \frac{\Delta t}{2}\left[\frac{\partial}{\partial x}\left(\Delta t \frac{\partial F}{\partial t}\right)^n + \frac{\partial}{\partial y}\left(\Delta t\frac{\partial G}{\partial t}\right)^n + \frac{\partial}{\partial z}\left(\Delta t \frac{\partial H}{\partial t}\right)^n\right].\end{aligned} \tag{3.3}$$

The second-order changes are defined as

$$\Delta F = \Delta t \frac{\partial F}{\partial t}, \quad \Delta G = \Delta t \frac{\partial G}{\partial t}, \quad \Delta H = \Delta t \frac{\partial H}{\partial t}. \tag{3.4}$$

Integrating over a pseudo-cell P formed by joining the centers of the cells surrounding node 1, as sketched in Figure 3.1a), gives

$$\int_P \delta U_1 dV = \int_P \left[-\Delta t\left(\frac{\partial F}{\partial x} + \frac{\partial G}{\partial y} + \frac{\partial H}{\partial z}\right)^n - \frac{\Delta t}{2}\left(\frac{\partial}{\partial x}\Delta F^n + \frac{\partial}{\partial y}\Delta G^n + \frac{\partial}{\partial z}\Delta H^n\right)\right] dV. \tag{3.5}$$

Then by applying Gauss' theorem we get

$$\delta U_1 = -\frac{\Delta t_1}{V_1}\int_P (F, G, H)\cdot \vec{n}\, dS - \frac{\Delta t_1}{2V_1}\int_P (\Delta F, \Delta G, \Delta H)\cdot \vec{n}\, dS. \tag{3.6}$$

$\vec{n}$ denotes the unit normal to the cell surfaces pointing outwards, V_1 is the volume of P. Δt_1 is the time-step associated with node 1 and is defined by

$$\left(\frac{V_1}{\Delta t_1}\right) = \frac{1}{8}\sum_{8\ cells}\left(\frac{V_{cell}}{\Delta t_{cell}}\right), \tag{3.7}$$

where the sum operates over the eight cells surrounding node 1. In Figure 3.1a) node 1 is surrounded by mesh cells A, B, C, D, E, F, G, H (mesh cells D and H are not represented for clarity purposes). Figure 3.1b) and c) represent the pseudo-cell split into eight cells $A_P, B_P, C_P, D_P, E_P, F_P, G_P, H_P$. The integration of the first-order terms is found by

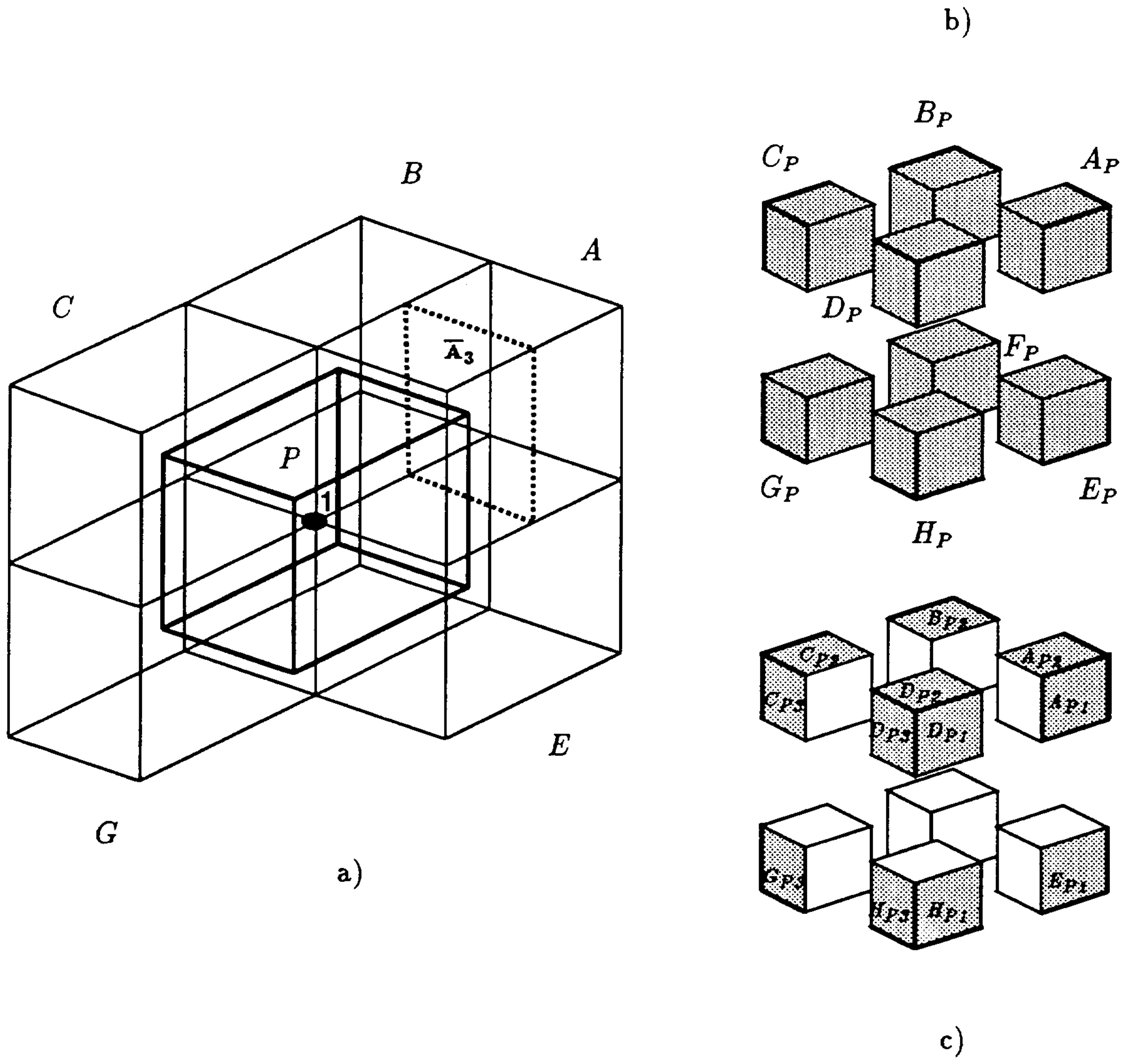


Figure 3.1: a) Mesh cells A to H surrounding node 1 and pseudo-cell P centered on node 1, b) and c) enlargement of pseudo-cell P split into eight cells A_P to H_P. Shaded surfaces indicate surfaces used for b) integration of first-order terms and c) integration of second-order terms.

integrating over *all* the surfaces of these eight volumes as shown by the dashed surfaces of Figure 3.1b)

$$-\frac{\Delta t_1}{V_1}\int_P (F,G,H)\cdot\vec{n}\,dS = -\frac{\Delta t_1}{V_1}\int_{A_P,\dots,H_P}(F,G,H)\cdot\vec{n}\,dS. \tag{3.8}$$

Each of these integrals is estimated as one eighth of the surface integral over the corresponding mesh cell surrounding node 1 so that

$$-\frac{\Delta t_1}{V_1}\int_P (F,G,H)\cdot\vec{n}\,dS = -\frac{\Delta t_1}{8V_1}\int_{A,\dots,H}(F,G,H)\cdot\vec{n}\,dS. \tag{3.9}$$

The integral of the second-order terms is performed over the *external* surfaces of the eight cells A_P to H_P as represented by the dashed surfaces in Figure 3.1c), so that

$$-\frac{\Delta t_1}{2V_1}\int_P (\Delta F,\Delta G,\Delta H)\cdot\vec{n}\,dS = -\frac{\Delta t_1}{2V_1}\int_{\substack{A_{P1},A_{P2},A_{P3}\\ \dots\\ H_{P1},H_{P2},H_{P3}}}(\Delta F,\Delta G,\Delta H)\cdot\vec{n}\,dS. \tag{3.10}$$

The integral over the surface A_{P1} is evaluated as one fourth of the integral over the 'mean surface' $\overline{A}_1$ defined as the average surface between two opposite faces of mesh cell A. The 'mean surface' $\overline{A}_3$ is represented on Figure 3.1a). A similar procedure applies for surfaces A_{P2} to H_{P3} so that

$$-\frac{\Delta t_1}{2V_1}\int_P (\Delta F,\Delta G,\Delta H)\cdot\vec{n}\,dS = -\frac{\Delta t_1}{8V_1}\int_{\substack{\overline{A}_1,\overline{A}_2,\overline{A}_3\\ \dots\\ \overline{H}_1,\overline{H}_2,\overline{H}_3}}(\Delta F,\Delta G,\Delta H)\cdot\vec{n}\,dS. \tag{3.11}$$

Hence,

$$\delta U_1 = -\frac{\Delta t_1}{8V_1}\int_{A,\dots,H}(F,G,H)\cdot\vec{n}\,dS - \frac{\Delta t_1}{8V_1}\int_{\substack{\overline{A}_1,\overline{A}_2,\overline{A}_3\\ \dots\\ \overline{H}_1,\overline{H}_2,\overline{H}_3}}(\Delta F,\Delta G,\Delta H)\cdot\vec{n}\,dS, \tag{3.12}$$

or formally

$$\delta U_1 = \delta U_{1A} + \delta U_{1B} \dots + \delta U_{1H}, \tag{3.13}$$

where δU_{1A} is the contribution from cell A to the change in the state vector at node 1 and so on for the contributions of cells B to H. δU_{1A} can then be written as

$$\delta U_{1A} = \frac{1}{8}\frac{\Delta t_1}{V_1}\left(\frac{V_A}{\Delta t_A}\Delta U_A - \sum_{\overline{A_1},\overline{A_2},\overline{A_3}} (\Delta F_A S_x + \Delta G_A S_y + \Delta H_A S_z)\right), \qquad (3.14)$$

where V_A is the volume of cell A. S_x, S_y and S_z are the components of the surface vector in the Cartesian coordinates, and ΔU_A is the average first-order change in the state vector in cell A defined as

$$\Delta U_A = -\frac{\Delta t_A}{V_A}\int_A (F, G, H)\cdot \vec{n}\, dS = -\frac{\Delta t_A}{V_A}\sum_A (\overline{F}S_x + \overline{G}S_y + \overline{H}S_z) \qquad (3.15)$$

and $\overline{F}, \overline{G}, \overline{H}$ are averages of F, G, H over the four nodes of each face.

The second-order terms $\Delta F_A, \Delta G_A, \Delta H_A$ are expressed as a function of the change ΔU_A as

$$\Delta F_A = \left(\frac{\partial F}{\partial U}\right)_A \Delta U_A, \quad \Delta G_A = \left(\frac{\partial G}{\partial U}\right)_A \Delta U_A, \quad \Delta H_A = \left(\frac{\partial H}{\partial U}\right)_A \Delta U_A. \qquad (3.16)$$

However, a more straight forward way to compute the second-orders terms is to use the changes in the conservative variables as follows.

For a compressible flow

$$\Delta U_A = \begin{bmatrix} \Delta\rho \\ \Delta(\rho u) \\ \Delta(\rho v) \\ \Delta(\rho w) \\ \Delta(\rho E) \end{bmatrix}_A, \qquad \Delta F_A = \begin{bmatrix} \Delta(\rho u) \\ u\Delta(\rho u) + \rho u\Delta u + \Delta p \\ u\Delta(\rho v) + \rho v\Delta u \\ u\Delta(\rho w) + \rho w\Delta u \\ u(\Delta(\rho E) + \Delta p) + (\rho E + p)\Delta u \end{bmatrix}_A,$$

$$\Delta G_A = \begin{bmatrix} \Delta(\rho v) \\ v\Delta(\rho u) + \rho u\Delta v \\ v\Delta(\rho v) + \rho v\Delta v + \Delta p \\ v\Delta(\rho w) + \rho w\Delta v \\ v(\Delta(\rho E) + \Delta p) + (\rho E + p)\Delta v \end{bmatrix}_A, \quad \Delta H_A = \begin{bmatrix} \Delta(\rho w) \\ w\Delta(\rho u) + \rho u\Delta w \\ w\Delta(\rho v) + \rho v\Delta w \\ w\Delta(\rho w) + \rho w\Delta w + \Delta p \\ w(\Delta(\rho E) + \Delta p) + (\rho E + p)\Delta w \end{bmatrix}_A$$

where u_A, v_A, w_A are obtained from averages over the nodes of cell A and

$$\begin{aligned}
(\Delta u)_A &= ((\Delta(\rho u) - u\Delta\rho)/\rho)_A \\
(\Delta v)_A &= ((\Delta(\rho v) - v\Delta\rho)/\rho)_A \\
(\Delta w)_A &= ((\Delta(\rho w) - w\Delta\rho)/\rho)_A \\
(\Delta p)_A &= (\gamma - 1)\left(\Delta(\rho E) - u\Delta(\rho u) - v\Delta(\rho v) - w\Delta(\rho w) + \frac{\Delta\rho}{2}(u^2 + v^2 + w^2)\right)_A .
\end{aligned}$$

For an incompressible flow

$$\Delta U_A = \begin{bmatrix} \Delta p^* \\ \Delta u \\ \Delta v \\ \Delta w \end{bmatrix}_A , \quad \Delta F_A = \begin{bmatrix} c_a^2 \Delta u \\ 2u\Delta u + \Delta p^* \\ u\Delta v + v\Delta u \\ u\Delta w + w\Delta u \end{bmatrix}_A ,$$

$$\Delta G_A = \begin{bmatrix} c_a^2 \Delta v \\ v\Delta u + u\Delta v \\ 2v\Delta v + \Delta p^* \\ v\Delta w + w\Delta v \end{bmatrix}_A , \quad \Delta H_A = \begin{bmatrix} c_a^2 \Delta w \\ w\Delta u + u\Delta w \\ w\Delta v + v\Delta w \\ 2w\Delta w + \Delta p^* \end{bmatrix}_A ,$$

where u_A, v_A, w_A are obtained from averages over the nodes of cell A.

In addition to Equation (3.14), the change in the state vector at node 1 receives contributions from the cells B to H written as

$$\delta U_{1B} = \frac{1}{8}\frac{\Delta t_1}{V_1}\left(\frac{V_B}{\Delta t_B}\Delta U_B - \sum_{\overline{B}_1,\overline{B}_2,\overline{B}_3} (\Delta F_B S_x + \Delta G_B S_y + \Delta H_B S_z)\right) , \quad (3.17)$$

$$\delta U_{1C} = \frac{1}{8}\frac{\Delta t_1}{V_1}\left(\frac{V_C}{\Delta t_C}\Delta U_C - \sum_{\overline{C}_1,\overline{C}_2,\overline{C}_3} (\Delta F_C S_x + \Delta G_C S_y + \Delta H_C S_z)\right) , \quad (3.18)$$

$$\delta U_{1D} = \frac{1}{8}\frac{\Delta t_1}{V_1}\left(\frac{V_D}{\Delta t_D}\Delta U_D - \sum_{\overline{D}_1,\overline{D}_2,\overline{D}_3} (\Delta F_D S_x + \Delta G_D S_y + \Delta H_D S_z)\right) , \quad (3.19)$$

$$\delta U_{1E} = \frac{1}{8}\frac{\Delta t_1}{V_1}\left(\frac{V_E}{\Delta t_E}\Delta U_E - \sum_{\overline{E}_1,\overline{E}_2,\overline{E}_3}(\Delta F_E S_x + \Delta G_E S_y + \Delta H_E S_z)\right), \quad (3.20)$$

$$\delta U_{1F} = \frac{1}{8}\frac{\Delta t_1}{V_1}\left(\frac{V_F}{\Delta t_F}\Delta U_F - \sum_{\overline{F}_1,\overline{F}_2,\overline{F}_3}(\Delta F_F S_x + \Delta G_F S_y + \Delta H_F S_z)\right), \quad (3.21)$$

$$\delta U_{1G} = \frac{1}{8}\frac{\Delta t_1}{V_1}\left(\frac{V_G}{\Delta t_G}\Delta U_G - \sum_{\overline{G}_1,\overline{G}_2,\overline{G}_3}(\Delta F_G S_x + \Delta G_G S_y + \Delta H_G S_z)\right), \quad (3.22)$$

$$\delta U_{1H} = \frac{1}{8}\frac{\Delta t_1}{V_1}\left(\frac{V_H}{\Delta t_H}\Delta U_H - \sum_{\overline{H}_1,\overline{H}_2,\overline{H}_3}(\Delta F_H S_x + \Delta G_H S_y + \Delta H_H S_z)\right). \quad (3.23)$$

The calculation of the cell volumes, face areas and volumes associated with cell nodes is described in Appendix B.

3.2 Numerical smoothing

By expanding the dispersion relation of the Lax-Wendroff scheme into Taylor series, it can be seen that the Lax-Wendroff scheme carries both an inherent dissipation term and a third-order dispersion error [73, 76]. The former term accounts for the scheme stability when operating on smooth flow fields. Nevertheless, the dispersion error is responsible for the introduction of oscillatory modes in the solution. In order to damp these background oscillations, an artificial dissipation term has to be added to the Euler equations.

The compressible flow version of the Euler solver uses a standard Laplacian second-difference smoothing. The Laplacian is obtained at node 1 by summing the difference between the state vector at node 1, U_1, and the cell-averaged state vector of the 8 surrounding cells U_A to U_H (the nomenclature is described in Section 3.1). The change in the state vector at node 1 is then the sum of Equation (3.13) and the contribution from the second-difference smoothing

$$\delta U_1 = \delta U_{1A} + ... + \delta U_{1H} + \nu_2 \frac{\Delta t_1}{V_1} \sum_{kc=A,...,H} \left(\frac{V_{kc}}{\Delta t_{kc}} (U_{kc} - U_1) \right), \qquad (3.24)$$

where ν_2 is an artificial viscosity coefficient, V_1 the volume associated with node 1 and V_{kc} the volume of the cell kc. Δt_1 and Δt_{kc} are the time-steps associated with the node 1 and the cell kc, respectively.

The incompressible flow version of the Lax-Wendroff algorithm uses a fourth-difference smoothing. The fourth-difference operator is constructed as a second-difference of a second-difference. Instead of using two standard Laplacian operators, the inner second-difference involves a "pseudo-Laplacian" operator defined by Holmes and Connell [31] and obtained at node n as

$$D_{2_n} = \sum_{i=1}^{i_{max}} w_i (U_i - U_n), \qquad (3.25)$$

where the sum operates on any number i_{max} of nodal points surrounding node n. As

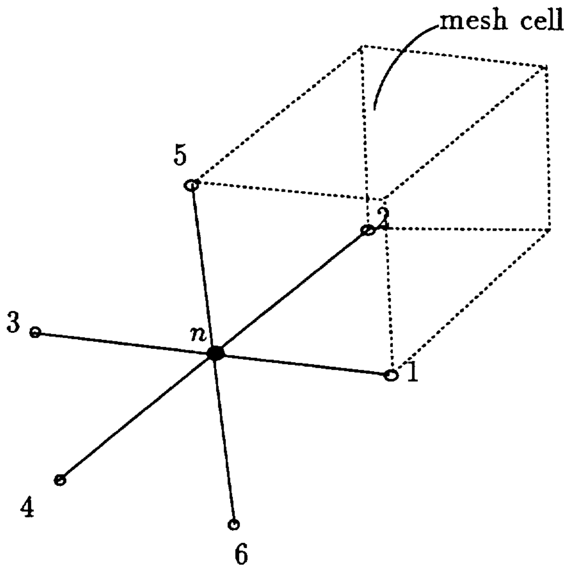


Figure 3.2: Nodal points 1 to 6 chosen for determination of fourth-difference smoothing at node n.

shown in Figure 3.2 the 6 surrounding nodes along the neighboring cell edges have been selected in the domain (less nodes are involved at the boundaries because of the missing neighbors). The w_i are the "weight" factors allowing the smoothing operator to be transparent when applied to any linear function in x, y or z and for any choice of i_{max}. Thus, the weight factors w_i must statisfy

$$\sum_{i=1}^{i_{max}} w_i(x_i - x_n) = 0, \tag{3.26}$$

$$\sum_{i=1}^{i_{max}} w_i(y_i - y_n) = 0, \tag{3.27}$$

$$\sum_{i=1}^{i_{max}} w_i(z_i - z_n) = 0. \tag{3.28}$$

To ensure stability, the weight factors are chosen close to unity by minimizing the cost function

$$C_n = \sum_{i=1}^{i_{max}} \Delta w_i^2, \tag{3.29}$$

where

$$\Delta w_i = 1 - w_i. \tag{3.30}$$

The fourth-difference smoothing is written as the standard second-difference operating on this "pseudo-Laplacian". The change in the state vector at node 1 is then formed by the sum of Equation (3.13) and the contribution from the fourth-difference smoothing

$$\delta U_1 = \delta U_{1A} + ... + \delta U_{1H} - \nu_4 \frac{\Delta t_1}{V_1} \sum_{kc=A,...H} \frac{V_{kc}}{\Delta t_{kc}} (D_{2_{kc}} - D_{2_1}). \tag{3.31}$$

ν_4 is the fourth-difference smoothing coefficient and $D_{2_{kc}}$ is the cell-average of D_2 at the eight nodes of cell kc.

The use of this pseudo-Laplacian ensures that the smoothing term does not distort any linear function in the domain (even the linear functions perpendicular to the boundaries). Since the choice of i_{max} is arbitrary, there is no need for a special treatment of the smoothing at boundaries of the domain (inlet, exit, wall,...). Also, the use of weights allows linear functions to be exactly represented on irregular grids.

In some instances, as in the case of a convex wall, some of the weights become negative. Holmes [31] elected to clip the weights in the range $(0, 2)$ because of stability problems. However, it has been found in this work that clipping the weights resulted in a degradation of the solution. The negative weights have not been clipped in this work without any significant stability problem. However, a slight decrease in the convergence rate was observed.

The actual algebra involved in the computation of the weight factors is presented below. The problem consists in minimizing the cost function C_n under the constraint that the pseudo-Laplacian operator must give zero value when applied to linear functions in x, y and z. Using the Lagrange multipliers, the function to minimize is

$$F = \sum_{i=1}^{imax} ((\Delta w_i)^2 + w_i(\lambda_x(x_i - x_n) + \lambda_y(y_i - y_n) + \lambda_z(z_i - z_n))) \tag{3.32}$$

where $\lambda_x, \lambda_y, \lambda_z$ are three unknowns. The 'Euler' equation of the problem is

$$\frac{\partial F}{\partial \Delta w_i} = 0, \tag{3.33}$$

or

$$\Delta w_i = -0.5(\lambda_x(x_i - x_n) + \lambda_y(y_i - y_n) + \lambda_z(z_i - z_n)) \tag{3.34}$$

By combining this equation with Equation (3.26),Eq (3.27) and Equation (3.28), $\lambda_x, \lambda_y, \lambda_z$ are found as

$$\begin{aligned}
\lambda_x &= \frac{-R_x(I_{yy}I_{zz}-I_{yz}^2)+R_y(I_{xy}I_{zz}-I_{xz}I_{yz})-R_z(I_{xy}I_{yz}-I_{yy}I_{xz})}{I_{xx}(I_{yy}I_{zz}-I_{yz}^2)-I_{xy}(I_{xy}I_{zz}-I_{xz}I_{yz})+I_{xz}(I_{xy}I_{yz}-I_{yy}I_{xz})}, \\
\lambda_y &= \frac{R_x(I_{xy}I_{zz}-I_{xz}I_{yz})-R_y(I_{xx}I_{zz}-I_{xz}^2)+R_z(I_{xx}I_{yz}-I_{xy}I_{xz})}{-I_{xy}(I_{xy}I_{zz}-I_{xz}I_{yz})+I_{yy}(I_{xx}I_{zz}-I_{xz}^2)-I_{yz}(I_{xx}I_{yz}-I_{xy}I_{xz})}, \\
\lambda_z &= \frac{-R_x(I_{xy}I_{yz}-I_{yy}I_{xz})+R_y(I_{xx}I_{yz}-I_{xy}I_{xz})-R_z(I_{xx}I_{yy}-I_{xy}^2)}{I_{xz}(I_{xy}I_{yz}-I_{yy}I_{xz})-I_{yz}(I_{xx}I_{yz}-I_{xy}I_{xz})+I_{zz}(I_{xx}I_{yy}-I_{xy}^2)},
\end{aligned} \tag{3.35}$$

where

$$\begin{aligned}
R_x &= \sum_{i=1}^{imax}(x_i - x_n), & I_{xx} &= \sum_{i=1}^{imax}(x_i - x_n)^2, & I_{xy} &= \sum_{i=1}^{imax}(x_i - x_n)(y_i - y_n), \\
R_y &= \sum_{i=1}^{imax}(y_i - y_n), & I_{yy} &= \sum_{i=1}^{imax}(y_i - y_n)^2, & I_{xz} &= \sum_{i=1}^{imax}(x_i - x_n)(z_i - z_n), \\
R_z &= \sum_{i=1}^{imax}(z_i - z_n)\ , & I_{zz} &= \sum_{i=1}^{imax}(z_i - z_n)^2, & I_{yz} &= \sum_{i=1}^{imax}(y_i - y_n)(z_i - z_n).
\end{aligned}$$

3.3 Mesh singularity treatment

The meshes used in this work for the computation of flow in pipes present singularities where a given node belongs to only six cells instead of eight. Figure 3.3 represents such a singularity (in a 2-D case for clarity). Node 1 is surrounded by cells A to F (faces of cells A to C are shown). The pseudo-cell P has volume V_1 defined as an average over 6 cells

$$V_1 = \frac{1}{6} \sum_{6\ cells} V_{cell}. \tag{3.36}$$

Δt_1 is the time step associated with node 1 defined by

$$\left(\frac{V_1}{\Delta t_1}\right) = \frac{1}{6} \sum_{6\ cells} \left(\frac{V_{cell}}{\Delta t_{cell}}\right). \tag{3.37}$$

The pseudo-cell P can be split into 6 smaller cells (faces of A_P to C_P are shown).

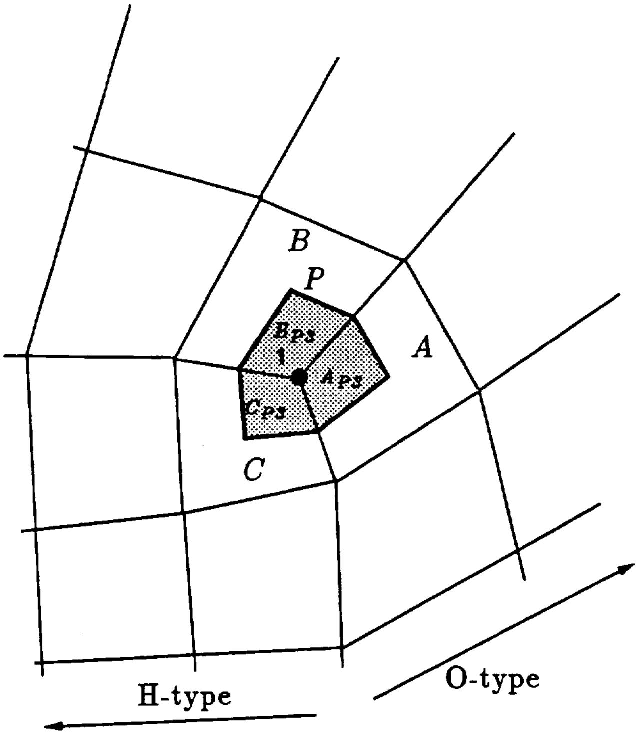


Figure 3.3: Mesh singularity at node 1 with pseudo-cell P.

The integration of the first-order terms is performed over all the surfaces of these 6 cells. As in Section 3.1, each of these integrals is in turn estimated as one eighth of the surface integral over the corresponding mesh cell surrounding node 1. Therefore

$$-\frac{\Delta t_1}{V_1}\int_P (F,G,H)\cdot\vec{n}\,dS = -\frac{\Delta t_1}{8V_1}\int_{A,\ldots,F}(F,G,H)\cdot\vec{n}\,dS. \tag{3.38}$$

The integral of the second-order terms is performed on the external surfaces of the 6 cells A_P to F_P (surfaces A_{P1}, A_{P2}, A_{P3} to F_{P1}, F_{P2}, F_{P3}). The integral over A_{P1} is evaluated as one fourth of the integral over $\overline{A}_1$ and so on for faces A_{P2} to F_{P3}. Hence,

$$-\frac{\Delta t_1}{2V_1}\int_P (\Delta F,\Delta G,\Delta H)\cdot\vec{n}\,dS = -\frac{\Delta t_1}{8V_1}\int_{\substack{\overline{A}_1,\overline{A}_2,\overline{A}_3\\ \cdots\\ \overline{F}_1,\overline{F}_2,\overline{F}_3}}(\Delta F,\Delta G,\Delta H)\cdot\vec{n}\,dS. \tag{3.39}$$

According to Section 3.2 the additive smoothing term can now be written as a sum over 6 cells instead of 8 and the final change in the state vector at the singular node 1 is

$$\delta U_1 = \delta U_{1A}+\delta U_{1B}+\delta U_{1C}+\delta U_{1D}+\delta U_{1E}+\delta U_{1F}+\nu_2\frac{\Delta t_1}{V_1}\sum_{kc=A,\ldots,F}\left(\frac{V_{kc}}{\Delta t_{kc}}(U_{kc}-U_1)\right), \tag{3.40}$$

when using a second-difference smoothing and

$$\delta U_1 = \delta U_{1A}+\delta U_{1B}+\delta U_{1C}+\delta U_{1D}+\delta U_{1E}+\delta U_{1F}-\nu_4\frac{\Delta t_1}{V_1}\sum_{kc=A,\ldots F}\frac{V_{kc}}{\Delta t_{kc}}(D_{2_{kc}}-D_{2_1}), \tag{3.41}$$

when using a fourth-difference smoothing. δU_{1A} to δU_{1F} are given by Equations (3.14) and (3.17) to (3.21).

In the unstructured code implementation the update of the state vector for the singular node is computed in a loop over the cells. Hence, the fact that only six cells are contributing to a singular node is directly dealt with by the connectivity table. Nevertheless, V_1 and Δt_1 associated with the singular node 1 need to be defined according to Equations (3.36) and (3.37).

3.4 Farfield boundary conditions

The farfield boundary conditions allow the use of truncated domains without affecting the numerical solution. These conditions follow from the 1-D linear characteristics theory whose theoretical background and implementation are described by Giles in [23] and [22]. The derivation of the Euler equations written in primitive form and computational coordinates is described in Appendix C. These equations are readily transformed back to physical coordinates as

$$\frac{\partial U_p}{\partial t} + A\frac{\partial U_p}{\partial x} + B\frac{\partial U_p}{\partial y} + C\frac{\partial U_p}{\partial z} = 0. \tag{3.42}$$

where

$$U_p = \begin{pmatrix} \rho \\ u \\ v \\ w \\ p \end{pmatrix}, \qquad A = \begin{pmatrix} u & \rho & 0 & 0 & 0 \\ 0 & u & 0 & 0 & \frac{1}{\rho} \\ 0 & 0 & u & 0 & 0 \\ 0 & 0 & 0 & u & 0 \\ 0 & \gamma p & 0 & 0 & u \end{pmatrix},$$

$$B = \begin{pmatrix} v & 0 & \rho & 0 & 0 \\ 0 & v & 0 & 0 & 0 \\ 0 & 0 & v & 0 & \frac{1}{\rho} \\ 0 & 0 & 0 & v & 0 \\ 0 & 0 & \gamma p & 0 & v \end{pmatrix}, \quad C = \begin{pmatrix} w & 0 & 0 & \rho & 0 \\ 0 & w & 0 & 0 & 0 \\ 0 & 0 & w & 0 & 0 \\ 0 & 0 & 0 & w & \frac{1}{\rho} \\ 0 & 0 & 0 & \gamma p & w \end{pmatrix},$$

for a compressible flow, and where

$$U_p = \begin{pmatrix} p^* \\ u \\ v \\ w \end{pmatrix}, \qquad A = \begin{pmatrix} 0 & c_a^2 & 0 & 0 \\ 1 & 2u & 0 & 0 \\ 0 & v & u & 0 \\ 0 & w & 0 & u \end{pmatrix},$$

$$B = \begin{pmatrix} 0 & 0 & c_a^2 & 0 \\ 0 & v & u & 0 \\ 1 & 0 & 2v & 0 \\ 0 & 0 & w & v \end{pmatrix}, \qquad C = \begin{pmatrix} 0 & 0 & 0 & c_a^2 \\ 0 & w & 0 & u \\ 0 & 0 & w & v \\ 1 & 0 & 0 & 2w \end{pmatrix}, .$$

for an incompressible flow (the artificial compressibility method has been used to modify the Euler equations).

Considering perturbations from a uniform and steady flow and retaining only the first-order terms gives a linear equation for $\tilde{U}_p$, the vector of the perturbations from a uniform flow

$$\frac{\partial \tilde{U}_p}{\partial t} + A\frac{\partial \tilde{U}_p}{\partial x} + B\frac{\partial \tilde{U}_p}{\partial y} + C\frac{\partial \tilde{U}_p}{\partial z} = 0, \tag{3.43}$$

where A, B and C are based on the uniform steady flow. Furthermore, if it is assumed that the perturbations travel normal to the considered boundary (say in the x direction), then the derivatives in the y and z directions can be ignored and the above system of equations reduces to the 1-D linear system

$$\frac{\partial \tilde{U}_p}{\partial t} + A\frac{\partial \tilde{U}_p}{\partial x} = 0. \tag{3.44}$$

For convenience, the $\tilde{}$ symbols will be omitted in the remainder of this section and $\tilde{U}_p$ is renamed U. The convention is that the matrices A, B and C are based on the uniform flow field and that U is now the vector of the perturbations from the uniform flow field. The matrix A can be diagonalized by the transformation

$$\Lambda = T^{-1}\, A\, T, \tag{3.45}$$

where the columns of T are the right eigenvectors and the lines of T^{-1} are the left eigenvectors associated with the five eigenvalues of A. Multiplying Equation (3.44) by T^{-1} gives

$$T^{-1}\frac{\partial U}{\partial t} + T^{-1}\, A\, T\, T^{-1}\frac{\partial U}{\partial x} = 0, \tag{3.46}$$

or

$$\frac{\partial \Phi}{\partial t} + \Lambda \frac{\partial \Phi}{\partial x} = 0, \tag{3.47}$$

where Φ is the vector of the linearized characteristic variables defined as

$$\Phi = T^{-1}\, U. \tag{3.48}$$

The eigenvalues of A (i.e. the diagonal elements of Λ) represent the speeds of different propagating waves. Each positive eigenvalue corresponds to a characteristic entering the domain and each negative eigenvalue corresponds to an outgoing characteristic. At any boundary the number of conditions to be imposed must correspond to the number of characteristics entering the domain.

In the following the superscript ^{s} stands for specified values. Superscript ^{n} refers to values at the previous time-step and $^{n+1}$ to predicted Lax-Wendroff values at the present time-step. Any characteristic leaving the domain will be superscripted $^{n+1}$ since it is a function of the Lax-Wendroff values in the domain.

3.4.1 Farfield boundary conditions for compressible flow

In the case of a compressible flow, the matrices T and T^{-1} are

$$T = \begin{pmatrix} -\frac{1}{c^2} & 0 & 0 & \frac{1}{2c^2} & \frac{1}{2c^2} \\ 0 & 0 & 0 & \frac{1}{2\rho c} & -\frac{1}{2\rho c} \\ 0 & \frac{1}{\rho c} & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{\rho c} & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{2} & \frac{1}{2} \end{pmatrix}, \quad T^{-1} = \begin{pmatrix} -c^2 & 0 & 0 & 0 & 1 \\ 0 & 0 & \rho c & 0 & 0 \\ 0 & 0 & 0 & \rho c & 0 \\ 0 & \rho c & 0 & 0 & 1 \\ 0 & -\rho c & 0 & 0 & 1 \end{pmatrix},$$

and the five eigenvalues of matrix A are

$$\lambda_1 = u, \quad \lambda_2 = u, \quad \lambda_3 = u, \quad \lambda_4 = u + c, \quad \lambda_5 = u - c. \tag{3.49}$$

For a subsonic compressible flow the first four eigenvalues are positive and the fifth is negative. Thus, at an inlet boundary the four first characteristics are propagating into the domain whereas the fifth propagates out of the domain. At an exit boundary the first four characteristics are exiting the domain and the fifth propagates into the domain.

Inlet boundary - compressible flow

At the inlet boundary, the total enthalpy h_0, the entropy related function s and two angles α_1, α_2 of the inlet velocity vector are specified. This is achieved by defining the four residuals R_1 to R_4 as differences between specified and Lax-Wendroff values as

$$\begin{aligned} R_1 &= h_0^n - h_0^s, \\ R_2 &= s^n - s^s, \\ R_3 &= (\tan\alpha_1)^n - (\tan\alpha_1)^s, \\ R_4 &= (\tan\alpha_2)^n - (\tan\alpha_2)^s, \end{aligned}$$

The entropy-related function s and the angles α_1, α_2 are defined as

$$s = \log(\gamma p) - \gamma \log \rho, \quad \tan\alpha_1 = \frac{v}{u}, \quad \tan\alpha_2 = \frac{w}{u}.$$

The necessary changes in the characteristics in order to drive the residuals to zero are found by performing one step of a Newton-Raphson procedure as

$$\begin{pmatrix} R_1 \\ R_2 \\ R_3 \\ R_4 \end{pmatrix} + \frac{\partial(R_1, R_2, R_3, R_4)}{\partial(\Phi_1, \Phi_2, \Phi_3, \Phi_4)} \begin{pmatrix} \delta\Phi_1 \\ \delta\Phi_2 \\ \delta\Phi_3 \\ \delta\Phi_4 \end{pmatrix} = 0. \tag{3.50}$$

The Jacobian matrix is computed as

$$J = \frac{\partial(R_1, R_2, R_3, R_4)}{\partial(\Phi_1, \Phi_2, \Phi_3, \Phi_4)} = \frac{\partial(R_1, R_2, R_3, R_4)}{\partial(\rho, u, v, w, p)} \frac{\partial(\rho, u, v, w, p)}{\partial(\Phi_1, \Phi_2, \Phi_3, \Phi_4)}$$

$$= \begin{pmatrix} \frac{-\gamma p}{(\gamma-1)\rho^2} & u & v & w & \frac{\gamma}{(\gamma-1)\rho} \\ \frac{-\gamma}{\rho} & 0 & 0 & 0 & \frac{1}{p} \\ 0 & \frac{-v}{u^2} & \frac{1}{u} & 0 & 0 \\ 0 & \frac{-w}{u^2} & 0 & \frac{1}{u} & 0 \end{pmatrix} \begin{pmatrix} -\frac{1}{c^2} & 0 & 0 & \frac{1}{2c^2} \\ 0 & 0 & 0 & \frac{1}{2\rho c} \\ 0 & \frac{1}{\rho c} & 0 & 0 \\ 0 & 0 & \frac{1}{\rho c} & 0 \\ 0 & 0 & 0 & \frac{1}{2} \end{pmatrix}$$

$$= \begin{pmatrix} \frac{\gamma}{\rho(\gamma-1)} & \frac{v}{\rho c} & \frac{w}{\rho c} & \frac{1}{\rho c}\frac{c+u}{2} \\ \frac{1}{p} & 0 & 0 & 0 \\ 0 & \frac{1}{\rho c u} & 0 & \frac{-v}{2\rho c u^2} \\ 0 & 0 & \frac{1}{\rho c u} & \frac{-w}{2\rho c u^2} \end{pmatrix} .$$

The required changes in the first four characteristics are then found from Equation (3.50) as

$$\begin{pmatrix} \delta\Phi_1 \\ \delta\Phi_2 \\ \delta\Phi_3 \\ \delta\Phi_4 \end{pmatrix} = -J^{-1} \begin{pmatrix} R_1 \\ R_2 \\ R_3 \\ R_4 \end{pmatrix} , \tag{3.51}$$

whereas the change in the fifth characteristic is found from

$$(\delta\Phi_5)^{n+1} = T_5^{-1}\,\delta U = -\rho c(\delta u)^{n+1} + (\delta p)^{n+1}, \tag{3.52}$$

where T_5^{-1} is the fifth line of T^{-1} and $(\delta p)^{n+1}$ and $(\delta u)^{n+1}$ are the Lax-Wendroff changes in pressure and inlet normal velocity at the inlet nodes. The changes in the primitive variables are then found by

$$\begin{pmatrix} \delta\rho \\ \delta u \\ \delta v \\ \delta w \\ \delta p \end{pmatrix} = T \begin{pmatrix} \delta\Phi_1 \\ \delta\Phi_2 \\ \delta\Phi_3 \\ \delta\Phi_4 \\ (\delta\Phi_5)^{n+1} \end{pmatrix} .$$

Exit boundary - compressible flow

At the exit boundary the pressure is set corresponding to the one necessary condition defining the residual

$$R = p^n - p^s. \tag{3.53}$$

The necessary changes in the characteristics in order to drive this residual to zero is found by

$$R + \frac{\partial R}{\partial \Phi_5} \delta \Phi_5 = 0, \tag{3.54}$$

and

$$\frac{\partial R}{\partial \Phi_5} = \frac{\partial R}{\partial (\rho, u, v, w, p)} \frac{\partial (\rho, u, v, w, p)}{\partial \Phi_5} = 1 \cdot \frac{1}{2}.$$

The required change in the fifth characteristic is then found as

$$\delta \Phi_5 = -2R_1 = -2(p^n - p^s),$$

whereas the change in the first four characteristics are calculated using the Lax-Wendroff changes in the primitive state vector as

$$\begin{pmatrix} \delta\Phi_1 \\ \delta\Phi_2 \\ \delta\Phi_3 \\ \delta\Phi_4 \end{pmatrix}^{n+1} = \begin{pmatrix} -c^2 & 0 & 0 & 0 & 1 \\ 0 & 0 & \rho c & 0 & 0 \\ 0 & 0 & 0 & \rho c & 0 \\ 0 & \rho c & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \delta\rho \\ \delta u \\ \delta v \\ \delta w \\ \delta p \end{pmatrix}^{n+1} . \tag{3.55}$$

The changes in the primitive variables are then found by

$$\begin{pmatrix} \delta\rho \\ \delta u \\ \delta v \\ \delta w \\ \delta p \end{pmatrix} = T \begin{pmatrix} (\delta\Phi_1)^{n+1} \\ (\delta\Phi_2)^{n+1} \\ (\delta\Phi_3)^{n+1} \\ (\delta\Phi_4)^{n+1} \\ \delta\Phi_5 \end{pmatrix} .$$

The changes in conservative variables are found for an inlet or exit boundary by using the changes in the primitive variables as

$$\begin{aligned}
\delta(\rho u) &= u\delta\rho + \rho\delta u, \\
\delta(\rho v) &= v\delta\rho + \rho\delta v, \\
\delta(\rho w) &= w\delta\rho + \rho\delta w, \\
\delta(\rho e_0) &= \frac{1}{\gamma-1}\delta p + \rho u\delta u + \rho v\delta v + \rho w\delta w + \frac{1}{2}(u^2+v^2+w^2)\delta\rho.
\end{aligned}$$

3.4.2 Farfield boundary conditions for incompressible flow

In the case of an incompressible flow, the matrices T and T^{-1} are

$$T = \begin{pmatrix} 0 & 0 & c_a^2\beta & -c_a^2\beta \\ 0 & 0 & u(u+\beta)+c_a^2 & u(u-\beta)+c_a^2 \\ 0 & 1 & v(u+\beta) & v(u-\beta) \\ 1 & 0 & w(u+\beta) & w(u-\beta) \end{pmatrix}, \quad T^{-1} = \begin{pmatrix} -\frac{w}{\beta^2} & -\frac{wu}{\beta^2} & 0 & 1 \\ -\frac{v}{\beta^2} & -\frac{vu}{\beta^2} & 1 & 0 \\ -\frac{u-\beta}{2c_a^2\beta^2} & \frac{1}{2\beta^2} & 0 & 0 \\ -\frac{u+\beta}{2c_a^2\beta^2} & \frac{1}{2\beta^2} & 0 & 0 \end{pmatrix},$$

and the four eigenvalues of matrix A are

$$\lambda_1 = u, \quad \lambda_2 = u, \quad \lambda_3 = u+\beta, \quad \lambda_4 = u-\beta, \tag{3.56}$$

where $\beta = \sqrt{u^2+c_a^2}$.

The first three eigenvalues are positive and the fourth is negative. Thus, at an inlet boundary the first three characteristics are propagating into the domain whereas the fourth propagates out of the domain. At an exit boundary the first three characteristics are exiting the domain and the fourth propagates into the domain.

Inlet boundary - incompressible flow (internal flow case)

For the internal incompressible flow cases, namely the flow in pipes, the inlet velocity profile is known through measurements or can be approximated by computing a fully developed velocity profile. In these cases, the three components of the inlet velocity are specified so that

$$\begin{aligned} R_1 &= u^n - u^s \\ R_2 &= v^n - v^s \\ R_3 &= w^n - w^s. \end{aligned}$$

The necessary changes in the characteristics in order to drive the residuals to zero are found by

$$\begin{pmatrix} R_1 \\ R_2 \\ R_3 \end{pmatrix} + \frac{\partial(R_1, R_2, R_3)}{\partial(\Phi_1, \Phi_2, \Phi_3)} \begin{pmatrix} \delta\Phi_1 \\ \delta\Phi_2 \\ \delta\Phi_3 \end{pmatrix} = 0. \tag{3.57}$$

The Jacobian matrix is computed as

$$J = \frac{\partial(R_1, R_2, R_3)}{\partial(\Phi_1, \Phi_2, \Phi_3)} = \frac{\partial(R_1, R_2, R_3)}{\partial(p^*, u, v, w)} \frac{\partial(p^*, u, v, w)}{\partial(\Phi_1, \Phi_2, \Phi_3)}$$

$$= \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0 & 0 & c_a^2\beta \\ 0 & 0 & u(u+\beta) + c_a^2 \\ 0 & 1 & v(u+\beta) \\ 1 & 0 & w(u+\beta) \end{pmatrix} = \begin{pmatrix} 0 & 0 & u(u+\beta) + c_a^2 \\ 0 & 1 & v(u+\beta) \\ 1 & 0 & w(u+\beta) \end{pmatrix}.$$

The required changes in the three first characteristics are then found from Equation (3.57) as

$$\begin{pmatrix} \delta\Phi_1 \\ \delta\Phi_2 \\ \delta\Phi_3 \end{pmatrix} = -J^{-1} \begin{pmatrix} R_1 \\ R_2 \\ R_3 \end{pmatrix}, \tag{3.58}$$

whereas the change in the fourth characteristic is found from

$$(\delta\Phi_4)^{n+1} = T_4^{-1}\ \delta U = -\frac{u+\beta}{2\beta^2 c_a^2}(\delta p^*)^{n+1} + \frac{1}{2\beta^2}(\delta u)^{n+1}. \qquad (3.59)$$

T_4^{-1} is the fourth line of T^{-1} and $(\delta p^*)^{n+1}$ and $(\delta u)^{n+1}$ are the Lax-Wendroff changes in pressure and inlet normal velocity at the inlet nodes. The changes in the primitive variables are then found from

$$\begin{pmatrix} \delta p^* \\ \delta u \\ \delta v \\ \delta w \end{pmatrix} = T \begin{pmatrix} \delta\Phi_1 \\ \delta\Phi_2 \\ \delta\Phi_3 \\ (\delta\Phi_4)^{n+1} \end{pmatrix}.$$

Inlet boundary - incompressible flow (external flow case)

For the external incompressible flow cases, namely the flow around a wing, better results were obtained by not specifying the three components of the inlet velocity but the stagnation pressure and two angles of the velocity vector instead. In this case the residuals are

$$\begin{aligned} R_1 &= p_0^n - p_0^s, \\ R_2 &= (\tan\alpha_1)^n - (\tan\alpha_1)^s, \\ R_3 &= (\tan\alpha_2)^n - (\tan\alpha_2)^s, \end{aligned}$$

where the stagnation pressure and the two angles of the inlet velocity vector are defined as

$$p_0 = p^* + \frac{1}{2}(u^2 + v^2 + w^2), \quad \tan\alpha_1 = \frac{v}{u}, \quad \tan\alpha_2 = \frac{w}{u}.$$

The necessary changes in the characteristics in order to drive the residuals to zero are found by

$$\begin{pmatrix} R_1 \\ R_2 \\ R_3 \end{pmatrix} + \frac{\partial(R_1, R_2, R_3)}{\partial(\Phi_1, \Phi_2, \Phi_3)} \begin{pmatrix} \delta\Phi_1 \\ \delta\Phi_2 \\ \delta\Phi_3 \end{pmatrix} = 0. \tag{3.60}$$

The Jacobian matrix is computed as

$$J = \frac{\partial(R_1, R_2, R_3)}{\partial(\Phi_1, \Phi_2, \Phi_3)} = \frac{\partial(R_1, R_2, R_3)}{\partial(p^*, u, v, w)} \frac{\partial(p^*, u, v, w)}{\partial(\Phi_1, \Phi_2, \Phi_3)}$$

$$= \begin{pmatrix} 1 & u & v & w \\ 0 & -\frac{\tan\alpha_1}{u} & \frac{1}{u} & 0 \\ 0 & -\frac{\tan\alpha_2}{u} & 0 & -\frac{1}{u} \end{pmatrix} \begin{pmatrix} 0 & 0 & c_a^2\beta \\ 0 & 0 & u(u+\beta)+c_a^2 \\ 0 & 1 & v(u+\beta) \\ 1 & 0 & w(u+\beta) \end{pmatrix},$$

$$= \begin{pmatrix} w & v & (u+\beta)(c_a^2 + u^2(1+\tan^2\alpha_1 + \tan^2\alpha_2)) \\ 0 & \frac{1}{u} & -\frac{\tan\alpha_1 c_a^2}{u} \\ \frac{1}{u} & 0 & -\frac{\tan\alpha_1 c_a^2}{u} \end{pmatrix}.$$

The required changes in the first three characteristics are then found from Equation (3.60) as

$$\begin{pmatrix} \delta\Phi_1 \\ \delta\Phi_2 \\ \delta\Phi_3 \end{pmatrix} = -J^{-1} \begin{pmatrix} R_1 \\ R_2 \\ R_3 \end{pmatrix}, \tag{3.61}$$

whereas the change in the fourth characteristic is found from

$$(\delta\Phi_4)^{n+1} = T_4^{-1}\,\delta U = -\frac{u+\beta}{2\beta^2 c_a^2}(\delta p^*)^{n+1} + \frac{1}{2\beta^2}(\delta u)^{n+1}. \tag{3.62}$$

T_4^{-1} is the fourth line of T^{-1} and $(\delta p^*)^{n+1}$ and $(\delta u)^{n+1}$ are the Lax-Wendroff changes in pressure and inlet normal velocity at the inlet nodes. The changes in the primitive variables are then found from

$$\begin{pmatrix} \delta p^* \\ \delta u \\ \delta v \\ \delta w \end{pmatrix} = T \begin{pmatrix} \delta\Phi_1 \\ \delta\Phi_2 \\ \delta\Phi_3 \\ (\delta\Phi_4)^{n+1} \end{pmatrix}.$$

Exit boundary - incompressible flow

In the pipe flow cases, the vortex created by the secondary flow impinges on the exit boundary. Also in the incompressible flow around a wing, the trailing vortex sheet rolls up into a vortex which crosses the exit boundary. Imposing a constant pressure at the exit boundary would not be consistent with the presence of the vortex. Instead, good results were obtained by setting the first derivative of pressure to zero in the direction normal to the exit boundary. Using the node $i-1$ adjacent to the exit boundary as depicted in Figure 3.4, this gives

$$p_i^* = p_{i-1}^*$$

Approximating the pressure in the direction perpendicular to the exit boundary by a linear function (by setting the second derivative of the pressure to zero) gave similar results.

Using the pressures computed at each node of the exit boundary as specified pressures, the following residual is defined

$$R = p^{*n} - p^{*s}. \tag{3.63}$$

The necessary changes in the characteristics in order to drive this residual to zero is found from

$$R + \frac{\partial R}{\partial \Phi_4}\delta\Phi_4 = 0, \tag{3.64}$$

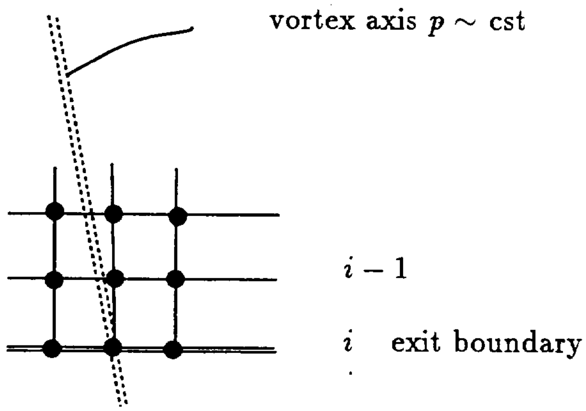


Figure 3.4: Exit boundary with impinging vortex.

and

$$\frac{\partial R}{\partial \Phi_4} = \frac{\partial R}{\partial (p^*, u, v, w)} \frac{\partial (p^*, u, v, w)}{\partial \Phi_4} = 1 \cdot (-c_a^2 \beta).$$

The required change in the fourth characteristic is then found as

$$\delta \Phi_4 = \frac{1}{c_a^2 \beta} R = \frac{1}{c_a^2 \beta} (p^{*n} - p^{*s}),$$

whereas the change in the first three characteristics are

$$\begin{array}{llllll} (\delta \Phi_1)^{n+1} & = T_1^{-1}\, \delta U & = -\frac{w}{\beta^2} & (\delta p^*)^{n+1} - \frac{wu}{\beta^2} & (\delta u)^{n+1} & & +(\delta w)^{n+1}, \\ (\delta \Phi_2)^{n+1} & = T_2^{-1}\, \delta U & = -\frac{v}{\beta^2} & (\delta p^*)^{n+1} - \frac{vu}{\beta^2} & (\delta u)^{n+1} & +(\delta v)^{n+1}, & \\ (\delta \Phi_3)^{n+1} & = T_3^{-1}\, \delta U & = \frac{-u+\beta}{2\beta^2 c_a^2} & (\delta p^*)^{n+1} + \frac{1}{2\beta^2} & (\delta u)^{n+1}. & & \end{array}$$

$T_1^{-1}, T_2^{-1}, T_3^{-1}$ are the first, second and third lines of T^{-1}, respectively and $(\delta p^*)^{n+1}$, $(\delta u)^{n+1}$, $(\delta v)^{n+1}$, $(\delta w)^{n+1}$ are the predicted Lax-Wendroff changes in pressure over density and velocity components at the exit nodes. The changes in the primitive variables are then found from

$$\begin{pmatrix} \delta p^* \\ \delta u \\ \delta v \\ \delta w \end{pmatrix} = T \begin{pmatrix} (\delta\Phi_1)^{n+1} \\ (\delta\Phi_2)^{n+1} \\ (\delta\Phi_3)^{n+1} \\ \delta\Phi_4 \end{pmatrix} .$$

3.5 Wall boundary condition

A no flux boundary condition is imposed at the wall boundaries, i.e.

$$\vec{v} \cdot \vec{n} = 0. \tag{3.65}$$

This condition implies that the fluxes F, G, H through the wall faces are only a function of the pressure.

The wall boundary condition is imposed once the Lax-Wendroff fluxes have been computed and the changes in the state vector have been determined for each node. Since the boundary condition deals with a flux condition on a face, the cell residual computed before in the main Lax-Wendroff routine is erroneous. The wall boundary condition corrects the residual for these particular cells. The corrected cell residual is then distributed to the eight nodes as before. The cell residual for cell A is (taking into account only the first-order terms)

$$\Delta U_A = -\frac{\Delta t_A}{V_A} \sum_{6\ faces} \left(\overline{F} S_x + \overline{G} S_y + \overline{H} S_z \right), \tag{3.66}$$

where $\overline{F}, \overline{G}, \overline{H}$ stands for the average values over the four nodes of F, G, H. The residual at the wall node 1 of cell A is defined as

$$\delta U_{1A} = \frac{1}{8} \frac{\Delta t_1}{V_1} \left(\frac{V_A}{\Delta t_A} \Delta U_A \right). \tag{3.67}$$

For a wall node, the node residual receives contribution from only four cells instead of eight but also takes into account a smaller node volume. By adding to the cell residual ΔU_A the term

$$\frac{\Delta t_A}{V_A} \left[\overline{F} S_x + \overline{G} S_y + \overline{H} S_z - \begin{pmatrix} 0 \\ \overline{p} S_x \\ \overline{p} S_y \\ \overline{p} S_z \\ 0 \end{pmatrix} \right]$$

computed on the wall face for the compressible flow cases and the term

$$\frac{\Delta t_A}{V_A}\left[\overline{F}S_x + \overline{G}S_y + \overline{H}S_z - \begin{pmatrix} 0 \\ \overline{p^*}S_x \\ \overline{p^*}S_y \\ \overline{p^*}S_z \end{pmatrix}\right]$$

computed on the wall face for the incompressible flow cases, the contribution of a normal mass flux on the wall face is eliminated. The new cell residual is then distributed to the eight nodes as in Section 3.1.

3.6 Symmetry boundary condition

In the incompressible flow cases analyzed in this work, the symmetry of the problem allows the solution to be performed on only half of the domain. A symmetry-plane boundary condition is enforced on the nodes lying in the symmetry surface.

The symmetry surface is defined by one computational plane. Due to the present choice of Cartesian coordinates, this surface corresponds also to the $(y = 0.)$ physical plane. Then for any existing node at a position (x, y, z), a pseudo-node located at a symmetrical position $(x, -y, z)$ is characterized by the same primitive variables but an opposite velocity component in the direction perpendicular to the symmetry surface (here the v component). For a given face with vector surface $\vec{S} = (S_x, S_y, S_z)$, the surface vector of the symmetrical face is $\vec{S_s} = (S_x, -S_y, S_z)$. From Equation (3.15) the average first-order change in the state vector ΔU_{A_s} for the pseudo-cell A_s symmetrical to cell A is then

$$\begin{pmatrix} \Delta p^* \\ \Delta u \\ \Delta v \\ \Delta w \end{pmatrix}_{A_s} = \begin{pmatrix} \Delta p^* \\ \Delta u \\ -\Delta v \\ \Delta w \end{pmatrix}_{A} .$$

From Section 3.1, it can be seen that the same kind of relations between cell A and cell A_s holds for the second-order terms. The time-steps and the volumes of cell A and A_s are identical, so that the contribution from cell A_s to the change in state vector at node 1 lying on the symmetry surface is

$$\begin{pmatrix} \delta p^* \\ \delta u \\ \delta v \\ \delta w \end{pmatrix}_{1A_s} = \begin{pmatrix} \delta p^* \\ \delta u \\ -\delta v \\ \delta w \end{pmatrix}_{1A} .$$

Therefore, the contribution from symmetric pseudo-cells is taken into account by doubling the contribution from the existing cells for the pressure and the components of velocity parallel to the symmetry surface. The change to the component of velocity perpendicular to the symmetry surface is set to zero.

Because the initialization of the flow may not satisfy the symmetry condition, a 'no through velocity' condition is specifically applied each time the state vector is updated, that is $v_1 = 0$.

The smoothing stencil at the symmetry surface is completed by taking into account the pseudo-node at location $(x, -y, z)$ characterized by the primitive variables $p^*, u, -v, w$.

3.7 Numerical implementation on unstructured grids

The actual implementation of the Lax-Wendroff algorithm is done in two steps. First the fluxes F, G, H at the nodes are computed in a loop over the nodes. Secondly, in a loop over the cells, the cell average state vector is computed. In the same loop the changes due to the first-order and second-order terms are determined by summation over cell faces, and the first-order and second-order terms are then added to the smoothing term to give the contribution of each cell. A second-difference operator is computed in the same loop and is applied on the state vector U in the case of a second-difference smoothing operator. In the case of a fourth difference smoothing, this operator is applied on a pseudo-Laplacian computed beforehand in a separate routine. The change in the state vector for the 8 nodes is then obtained by summing the contributions from the eight surrounding cells.

The subsequent boundary conditions imposition modifies the changes in the state vector for the nodes lying on boundary surfaces. Finally, in a loop over the nodes, the state vector changes are added to the current state vector value for each node.

The Lax-Wendroff algorithm is implemented on an unstructured grid where an array over the cells points to the 8 nodes of each cell. When looping over the cells, this array allows one to readily add the contributions from each cell to a node. 6 arrays over the cells are initialized as the 6 surface vectors for each cell. The numbering of the surface vectors is described in Figure B.1 of Appendix B. The inlet, exit, wall and symmetry nodes have associated arrays to allow the differentiation of these particular nodes from the rest of the field when handling the boundary conditions. The fourth-difference smoothing term calculation requires 'neighbor' information. According to the stencil defined in Figure 3.2, an additional array over the nodes contains the 6 neighbouring nodes along cell edges. Another array contains the 6 weight factors used in the computation of the pseudo-Laplacian.

During the calculation each node gets associated with the fluxes F, G, H, a volume, a time-step, a change in the state vector and the state vector itself. Each cell is associated with a volume and a time-step.

In order to take advantage of vectorization capabilities, each cell is 'colored'. The loops over cells are replaced by loops over colors. The maximum number of colors is determined by the requirement that any two cells of one color must have no common nodes, so that dependencies can be avoided. Boundary cells are associated with separate colors in order to differentiate them from the field cells.

3.8 Time-step restriction

According to the stability analysis given in Appendix C, the allowable time-step is given by the CFL condition

$$\Delta t \leq \frac{J}{\sqrt{r_1^2 + r_2^2 + r_3^2} + c\sqrt{a^2 + b^2 + d^2}} \tag{3.68}$$

for a compressible flow and

$$\Delta t \leq \frac{J}{\sqrt{r_1^2 + r_2^2 + r_3^2} + \sqrt{r_1^2 + r_2^2 + r_3^2 + c_a^2(\overline{a}^2 + \overline{b}^2 + \overline{d}^2)}}, \tag{3.69}$$

for an incompressible flow, where

$$J = x_\xi(y_\eta z_\zeta - y_\zeta z_\eta) - x_\eta(y_\xi z_\zeta - y_\zeta z_\xi) + x_\zeta(y_\xi z_\eta - y_\eta z_\xi), \tag{3.70}$$

$$\begin{aligned}
r_1 &= \quad u(y_\eta z_\zeta - y_\zeta z_\eta) - v(x_\eta z_\zeta - x_\zeta z_\eta) + w(x_\eta y_\zeta - x_\zeta y_\eta),\\
r_2 &= -u(y_\xi z_\zeta - y_\zeta z_\xi) + v(x_\xi z_\zeta - x_\zeta z_\xi) - w(x_\xi y_\zeta - x_\zeta y_\xi),\\
r_3 &= \quad u(y_\xi z_\eta - y_\eta z_\xi) - v(x_\xi z_\eta - x_\eta z_\xi) + w(x_\xi y_\eta - x_\eta y_\xi),
\end{aligned}$$

$$\overline{a}^2 = (y_\eta z_\zeta - y_\zeta z_\eta)^2 + (y_\xi z_\zeta - y_\zeta z_\xi)^2 + (y_\xi z_\eta - y_\eta z_\xi)^2, \tag{3.71}$$

$$\overline{b}^2 = (x_\eta z_\zeta - x_\zeta z_\eta)^2 + (x_\xi z_\zeta - x_\zeta z_\xi)^2 + (x_\xi z_\eta - x_\eta z_\xi)^2, \tag{3.72}$$

$$\overline{d}^2 = (x_\eta y_\zeta - x_\zeta y_\xi)^2 + (x_\xi y_\zeta - x_\zeta y_\xi)^2 + (x_\xi y_\eta - x_\eta y_\xi)^2. \tag{3.73}$$

The CFL condition is satisfied with a margin set by the CFL number multiplying the maximum time-step. This number is set to a value of 0.9. When computing steady-state flow problems with the time-marching procedure, local time-stepping is used in order to accelerate the convergence, i.e. the CFL condition is satisfied independently for each cell in the flow field. When computing unsteady flow cases, the time-step used in each cell is set equal to the minimum time-step in the field.

3.9 Accuracy study

An accuracy study is performed to confirm the second-order accuracy of the Eulerian Lax-Wendroff scheme. Since the incompressible flow through bent pipes is one of the topics of this thesis, the constant stagnation pressure flow though a 90^0 bend is chosen for this accuracy study. The Euler equations are transformed using the pseudo-compressibility concept described in Chapter 2 and since the flow is symmetrical, the calculations are performed on only one half of the pipe.

The test geometry is the 90^0 bend of circular cross-section, taken from the Enayet et al. data set [21]. It will also be used in Chapters 10 and 11 of the results. The pipe diameter is 0.048 m and the ratio of radius of curvature to pipe diameter is 2.8. The geometry extends two diameters upstream and downstream of the bend. The accuracy study is performed by using three grid densities with a mixed O-H type grid on each pipe cross-section and a H type grid along the bend. The three grids are composed of 53×22 nodes, 189×43 nodes and 713×85 nodes, respectively. Each grid is composed of four times as many cells as the previous grid in a cross-section and two times as many cells in the streamwise direction. Side and front views of the grids are represented in Figure 3.5.

The inlet conditions are defined to have constant stagnation pressure. A constant inlet velocity is set through the inlet boundary condition. The inlet cross-section is placed two diameters upstream of the bend so that the upstream influence of the bend is negligible at this location and the constant inlet pressure condition is obtained.

The exact solution for the flow field has constant stagnation pressure everywhere. Therefore, a global error in the Eulerian solver is quantified in terms of the L_2 norm of the errors in stagnation pressure ϵ_{p_0}

$$\epsilon_{p_0} = \sqrt{\frac{\sum_{i=1}^{N}(\Delta C_{p_{0_i}})^2}{N}}, \tag{3.74}$$

where N is the total number of grid nodes, and ΔC_{p_0} is the local error in stagnation

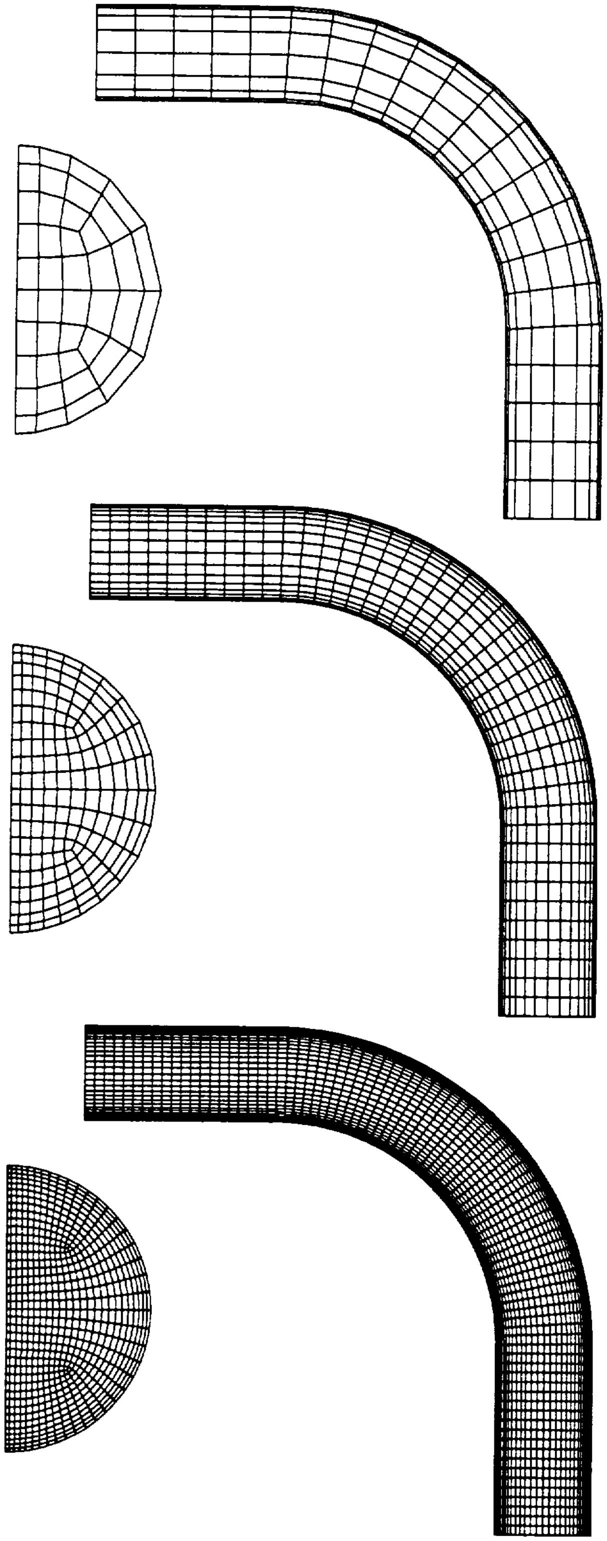


Figure 3.5: Front and side views of the grids used in the accuracy study (53 × 22 nodes, 189 × 43 nodes and 713 × 85 nodes).

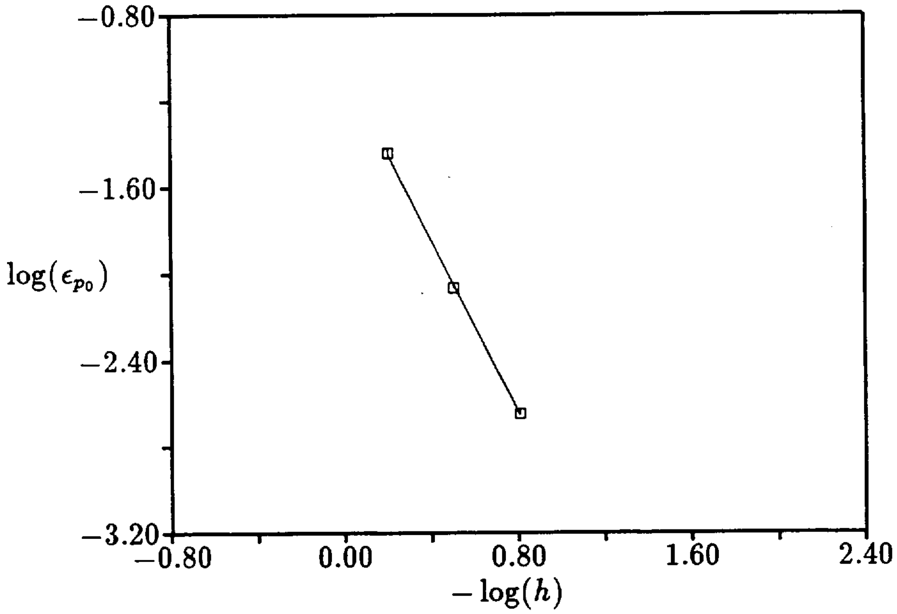


Figure 3.6: L_2 norm of errors in stagnation pressure as a function of the grid spacing for the three grid densities.

pressure coefficient defined at a node by

$$\Delta C_{p_0} = \frac{p_{0_{in}} - p_0}{\frac{1}{2}\rho_c U_{in}^2}. \tag{3.75}$$

The denominator represents the inlet dynamic head and $p_{0_{in}}$ is the inlet stagnation pressure. Referencing to the inlet dynamic head instead of the inlet stagnation pressure eliminates the dependency on the background pressure level which is arbitrary for incompressible flows.

The results are represented in Figure 3.6 where the L_2 norm of the errors in stagnation pressure coefficient is plotted as a function of the grid spacing h in a logarithmic scale for the three grids. The grid spacing is not uniform for this test case, but since the value of h is just a reference value, it is determined as an average over the cells. A line is drawn through the three points using a least-square fit. The slope of the line determining the order of accuracy of the Lax-Wendroff scheme is -2.0. The theoretical value which ensures second-order accuracy is -2.

Chapter 4

Lagrangian Governing Equations

4.1 Lagrange equations for compressible flow

The Lagrangian equations for the convection of a particle in an inviscid compressible and isentropic flow subjected to a conservative force field are

$$\frac{DU_l}{Dt} = T_l, \tag{4.1}$$

where

$$\frac{D}{Dt} \equiv \frac{\partial}{\partial t} + (\vec{v} \cdot \nabla) \tag{4.2}$$

defines the material derivative. When applied to a fluid quantity, this operator expresses the 'convective change' of the quantity due to the displacement of a given particle in time as the sum of the unsteady and spatial changes as the particle moves from one location to another. U_l is the state vector of the Lagrangian variables and T_l is a source-term vector governing the change during time of the Lagrangian state vector for a given particle. The Lagrangian state vector U_l and the source term T_l are defined by

$$U_l = \begin{bmatrix} \vec{r} \\ \vec{\omega} \\ S \end{bmatrix}, \quad T_l = \begin{bmatrix} \vec{v} \\ (\vec{\omega} \cdot \nabla)\vec{v} - \vec{\omega}(\nabla \cdot \vec{v}) - \nabla(\frac{1}{\rho}) \times \nabla p \\ 0 \end{bmatrix}, \tag{4.3}$$

where $\vec{r} = (x, y, z)$ is the position vector. The vorticity $\vec{\omega}$ and the entropy-related function S are defined by

$$\vec{\omega} = \nabla \times \vec{v}, \qquad S = \frac{p}{\rho^{\gamma}}. \tag{4.4}$$

By using the continuity equation, the change of vorticity for a given particle during time can also be written as

$$\frac{D(\frac{\vec{\omega}}{\rho})}{Dt} = (\frac{\vec{\omega}}{\rho} \cdot \nabla)\vec{v} - \frac{1}{\rho}\nabla(\frac{1}{\rho}) \times \nabla p. \tag{4.5}$$

The Lagrangian view of the momentum and energy equations given by Equations (4.1) and (4.3) expresses the convective change of vorticity and entropy for a particle.

4.1.1 Source-term contribution in compressible flow

The convective change for $\frac{\vec{\omega}}{\rho}$ in a compressible flow is the sum of the contributions from a tilting/stretching term and a non-barotropic term. The first term $(\frac{\vec{\omega}}{\rho} \cdot \nabla)\vec{v}$ results from the tilting and the stretching of the vortex lines during time. The non-barotropic term $-\frac{1}{\rho}\nabla(\frac{1}{\rho}) \times \nabla p$ represents the contribution from the moment of the pressure forces about the center of mass. The entropy-related function S is said to be passive, since it does not undergo any convective change.

4.2 Lagrange equations for incompressible flow

For an incompressible flow subjected to a conservative flow field, U_l and T_l are defined by

$$U_l = \begin{bmatrix} \vec{r} \\ \vec{\omega} \end{bmatrix}, \qquad T_l = \begin{bmatrix} \vec{v} \\ (\vec{\omega} \cdot \nabla)\vec{v} \end{bmatrix}. \tag{4.6}$$

The use of the total pressure p_0 as an additional convected quantity has not been elected since the vorticity and the Eulerian velocity field already determine the total pressure gradient according to Crocco's equation for incompressible flow

$$\frac{\partial \vec{v}}{\partial t} - \vec{v} \times \vec{\omega} = -\nabla \left(\frac{p_0}{\rho} \right). \tag{4.7}$$

4.2.1 Source-term contribution in incompressible flow

In the case of an incompressible flow, the source-term reduces to the tilting/stretching contribution $(\vec{\omega} \cdot \nabla)\vec{v}$. By decomposing $(\vec{\omega} \cdot \nabla)\vec{v}$, the stretching terms are identified as the terms

$$\omega_x \frac{\partial u}{\partial x}, \quad \omega_y \frac{\partial v}{\partial y}, \quad \omega_z \frac{\partial w}{\partial z}. \tag{4.8}$$

The fluid elements in the x, y, z directions with associated vorticity components ω_x,ω_y,ω_z, are stretched due to the strain fields $\frac{\partial u}{\partial x}, \frac{\partial v}{\partial y}, \frac{\partial w}{\partial z}$, respectively. The stretching of a fluid element of length dy with associated vorticity ω_y under the strain $\frac{\partial v}{\partial y}$ is illustrated in Figure 4.1. The resulting change in vorticity is $(\omega_y dv)$ for a fluid element of length dy. Each component of $\vec{\omega}$ is intensified or reduced depending on the the sign of the strain field. A positive strain corresponds to a vortex line intensification and inversely. The remainder of the terms

$$\omega_y \frac{\partial u}{\partial y}, \quad \omega_z \frac{\partial u}{\partial z}, \quad \omega_x \frac{\partial v}{\partial x}, \quad \omega_z \frac{\partial v}{\partial z}, \quad \omega_x \frac{\partial w}{\partial x}, \quad \omega_y \frac{\partial w}{\partial y} \tag{4.9}$$

are due to tilting. An example of vortex line tilting is illustrated in Figure 4.1 where a fluid element of length dx with associated vorticity ω_x is tilted an amount dw about the y axis under the influence of the strain field $\partial w/\partial x$. The resulting change in vorticity for a fluid element of length dx is $\omega_x dw$.

As can be seen in Figure 4.1, the final directions of the fluid element and vorticity vector are identical. Therefore, in an inviscid incompressible fluid subjected to a conservative force field, the vorticity $\vec{\omega}$ is said to be frozen to the fluid element. The fact

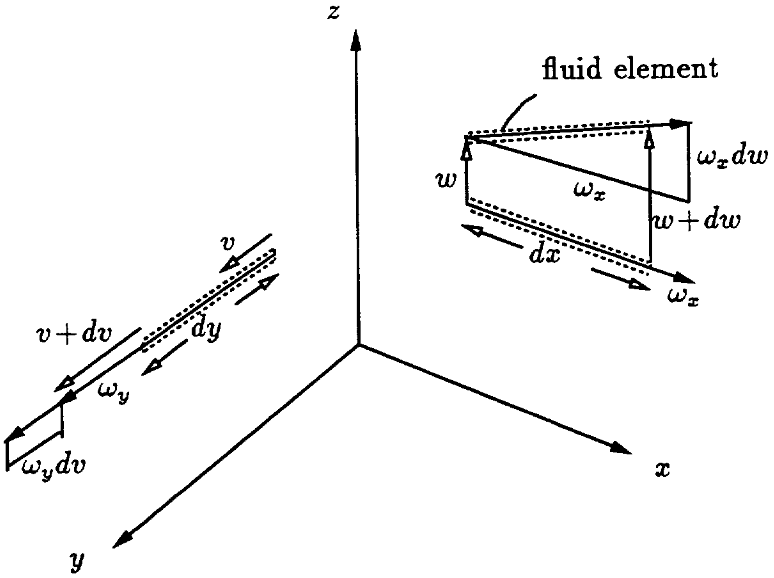


Figure 4.1: Stretching of vortex line in y direction and tilting of vortex line in x direction.

that vortex lines move as material lines, in an incompressible and inviscid flow, can also be deduced from the comparison of the Helmholtz equation

$$\frac{D\vec{\omega}}{Dt} = \vec{\omega} \cdot \nabla \vec{v}, \tag{4.10}$$

to the equation for the evolution of an infinitesimal line element $d\vec{l}$ [8]

$$\frac{D\vec{dl}}{Dt} = d\vec{l} \cdot \nabla \vec{v}. \tag{4.11}$$

In a compressible flow, the quantity $\frac{\vec{\omega}}{\rho}$ is attached to the fluid element.

Chapter 5

Eulerian/Lagrangian Integration

The Lagrangian solution is based on particle markers convecting through the flow. Each marker position, vorticity (and additionally entropy for compressible flow) at the marker's position form the Lagrangian state vector U_l. The convective change in the Lagrangian state vector, as the marker moves through the Eulerian grid, is obtained by the evaluation of the source term T_l.

The Eulerian solution provides the information required for the computation of the markers' trajectories and for the integration of the Lagrangian source-terms. In turn the Lagrangian solution is used to correct the Eulerian solution and reduce its numerical diffusion error. This process allows the Lagrangian solution to accurately capture the convection of vorticity (and entropy) while the Eulerian solution conserves mass, momentum (and energy). The standard time-integration of the Eulerian solver supports the iterative interaction of the Eulerian solution and the Lagrangian particle tracking solution.

First the computation of the markers' convection and the integration of the source-terms are described. The correction of the Eulerian solution using the Lagrangian state vector is presented in the next chapter.

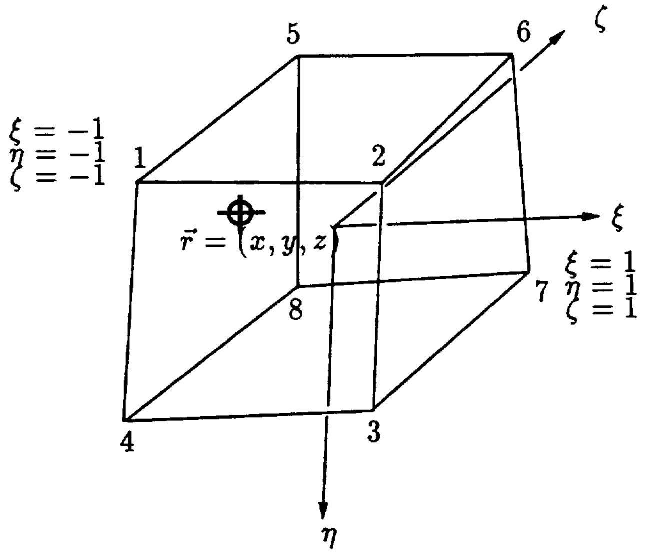


Figure 5.1: Local coordinate system in cell with node numbering and marker location at $\vec{r} = (x, y, z)$.

5.1 Convection of the markers and integration of the source-terms

At a given time-step in the Eulerian solution a set of markers is initialized at a chosen location in the Eulerian grid. Associated with the markers are arrays indicating in which cell each marker is located. The initialization of these arrays is provided by a brute force search through the whole domain as described in Appendix D. As the markers move through the Eulerian grid, these arrays are reset with the proper cell values by using a system of 'neighbouring cells' which will be described later.

A cell-centered local coordinate system (ξ, η, ζ) is set up in each cell of the computational domain as shown in Figure 5.1. The eight tri-linear interpolating functions N_1

to N_8 are defined from cell corners to marker in the cell as

$$\begin{aligned}
N_1 &= \tfrac{1}{8}(1-\xi)(1-\eta)(1-\zeta), & N_5 &= \tfrac{1}{8}(1-\xi)(1-\eta)(1+\zeta),\\
N_2 &= \tfrac{1}{8}(1+\xi)(1-\eta)(1-\zeta), & N_6 &= \tfrac{1}{8}(1+\xi)(1-\eta)(1+\zeta),\\
N_3 &= \tfrac{1}{8}(1+\xi)(1+\eta)(1-\zeta), & N_7 &= \tfrac{1}{8}(1+\xi)(1+\eta)(1+\zeta),\\
N_4 &= \tfrac{1}{8}(1-\xi)(1+\eta)(1-\zeta), & N_8 &= \tfrac{1}{8}(1-\xi)(1+\eta)(1+\zeta).
\end{aligned} \tag{5.1}$$

At each iteration, the local cell coordinates (ξ, η, ζ) for a marker are determined by solving the implicit system

$$\vec{r} = \sum_{k=1}^{8} N_k(\xi, \eta, \zeta)\, \vec{r}_k, \tag{5.2}$$

by a few Newton-Raphson iterations as described in Appendix E. $\vec{r}$ and $\vec{r}_k$ are the marker and the nodes positions, respectively. Usually, only three Newton-Raphson iterations are required.

Any Eulerian function f (say the velocity or the entropy) defined at the cell nodes is transferred to the marker location in the cell by the tri-linear interpolation as

$$f(\xi, \eta, \zeta) = \sum_{k=1}^{8} N_k(\xi, \eta, \zeta) f_k. \tag{5.3}$$

For steady-state solutions, the Eulerian and the Lagrangian solutions do not need to be integrated in time using the same time-steps during the pseudo-unsteady convergence. A local time-step can be used for the markers integration since, for steady flows, the markers can convect at any speed along the streamlines. In unsteady flows, however, the Eulerian and Lagrangian time-steps have to be the same, since at each instant in time, the markers represent the current location and Lagrangian state vector of a given particle. In the following, the Eulerian and the Lagrangian time-steps will be denoted Δt and Δt_l, respectively.

When a time-step Δt_l is taken in the Lagrangian solution in order to advance the marker position and integrate the source-terms, the new marker position is found by using a predictor-corrector integration scheme as

$$\vec{r}_p \;=\; \vec{r}(t_l) + \Delta t_l\, \vec{v}_{\vec{r}}^{\,n},$$

$$\begin{aligned} \vec{r}_c &= \vec{r}(t_l) + \Delta t_l \, \vec{v}^{n+1}_{\vec{r}_p}, \\ \vec{r}(t_l + \Delta t_l) &= \frac{1}{2}(\vec{r}_p + \vec{r}_c), \end{aligned} \tag{5.4}$$

where $\vec{r}_p$ and $\vec{r}_c$ are the predicted and corrected marker positions, respectively. $\vec{v}^n_{\vec{r}}$ is the Eulerian velocity at time t interpolated from the cell nodes to the old marker position $\vec{r}$, and $\vec{v}^{n+1}_{\vec{r}_p}$ is the Eulerian velocity at time $t + \Delta t$ interpolated from the cell nodes to the predicted marker position $\vec{r}_p$. The interpolation from nodes to marker position is performed using Equation (5.3). In System (5.4), the Eulerian velocity fields at time t and $t + \Delta t$ are used to find the new marker position. The predictor and the corrector step can also be based only on the Eulerian velocity field at time t. Both options are used depending on the Eulerian/Lagrangian interaction procedure as will be described in Section 5.2.

The marker's local coordinates have to be recalculated for the updated position of the marker $\vec{r}(t_l + \Delta t_l)$ by again solving System (5.2). If any of the ξ, η, ζ values falls out of the range $(-1, 1)$ but in the range $(-2, 2)$, the marker has moved out of the current cell but is located in one of the neighbouring cells. As mentioned before, associated with each cell is an array containing its 26 neighbouring cells so that when a marker moves out of a cell it can be readily relocated. Moreover, the locations of the 26 neighbouring cells relative to the central cell are known, so that the values of ξ,η and ζ directly indicate in which of the 26 cells the marker is located. (In the case of the grid singularities mentioned in Section 3.3 and encountered in pipe flow calculations, the missing neighbouring cells are accounted for by assigning two pointers to the same cell). The marker local coordinates are now recomputed by again solving System (5.2) for the cell where the marker has been found.

If the marker has moved more than one cell (any of the ξ, η, ζ values are out of the range $(-2, 2)$), a brute force search through the whole domain is required. This time-consuming procedure is avoided by specifying a Lagrangian time-step smaller than the time-step allowed by the CFL criterion in the Eulerian solution. The marker will then move over a distance of less than one cell in one Lagrangian time-step Δt_l. However, in steady flows with strong streamline curvature, as shown by the results from the swirling-flow cases, the CFL restriction does not allow for accurate streamline integrations. In these flows, the predicted marker position is used to scale the Lagrangian time-step such that the marker cannot cross the cell in less than two Lagrangian time-steps in any local cell coordinate direction.

The domain boundaries are bordered by 'wastebasket cells', so that any marker exiting the domain is either eliminated or can be reinitialized somewhere else in the domain. In order to minimize the array sizes, a marker renumbering system is used to eliminate any exiting marker from the arrays and introduce new markers in the domain.

In parallel with the trajectory integration described by System (5.4), the Lagrangian values of vorticity are computed at each marker by a predictor-corrector integration of the source-terms for the vorticity from Equation (4.3) as

$$\begin{aligned}\Delta\vec{\omega_p} &= \Delta t_l \left[(\vec{\omega}\cdot\nabla)\vec{v} - \vec{\omega}(\nabla\cdot\vec{v}) - \nabla\left(\frac{1}{\rho}\right)\times\nabla p\right]^{n}_{\vec{r}}, \\ \Delta\vec{\omega_c} &= \Delta t_l \left[(\vec{\omega}\cdot\nabla)\vec{v} - \vec{\omega}(\nabla\cdot\vec{v}) - \nabla\left(\frac{1}{\rho}\right)\times\nabla p\right]^{n+1}_{\vec{r_p}}, \qquad (5.5)\end{aligned}$$

for a compressible flow case. In turn, from Equation (4.6) for an incompressible flow

$$\begin{aligned}\Delta\vec{\omega_p} &= \Delta t_l \left[(\vec{\omega}\cdot\nabla)\vec{v}\right]^{n}_{\vec{r}}, \\ \Delta\vec{\omega_c} &= \Delta t_l \left[(\vec{\omega}\cdot\nabla)\vec{v}\right]^{n+1}_{\vec{r_p}}. \qquad (5.6)\end{aligned}$$

Then the Lagrangian value of vorticity at the new marker location is found by

$$\vec{\omega}(t_l + \Delta t_l) = \vec{\omega}(t_l) + \frac{1}{2}(\,\Delta\vec{\omega}_p + \Delta\vec{\omega}_c). \qquad (5.7)$$

In Systems (5.5) and (5.6), the quantities within brackets are determined at the old marker location $\vec{r}$ by using the Eulerian state vector at time t for the predictor step and at predicted marker location $\vec{r}_p$ by using the Eulerian state vector at time $t + \Delta t$. As for the trajectory integration, depending on the Eulerian/Lagrangian interaction procedure, predicted and corrected values can also be dependent on the Eulerian state vector at time t only.

The determination of the vorticity and of the non-barotropic term $\nabla(1/\rho) \times \nabla p$ at the marker location (old or predicted) on the right-hand side of Systems (5.5) and (5.6) requires the definition of spatial derivatives. These are found directly for any Eulerian function f defined at the grid nodes by differentiating Equation (5.3)

$$\frac{\partial f}{\partial \vec{r}} = \sum_{k=1}^{8} \left(\frac{\partial N_k}{\partial \xi} \frac{\partial \xi}{\partial \vec{r}} + \frac{\partial N_k}{\partial \eta} \frac{\partial \eta}{\partial \vec{r}} + \frac{\partial N_k}{\partial \zeta} \frac{\partial \zeta}{\partial \vec{r}} \right) f_k. \tag{5.8}$$

The metrics' derivatives $(\partial \xi / \partial \vec{r}), (\partial \eta / \partial \vec{r}), (\partial \zeta / \partial \vec{r})$ are directly found when using the Newton-Raphson procedure as described in Appendix E. The velocity derivatives computed by Equation (5.8) are required for the integration of the source-terms. In a compressible flow, the spatial derivatives of density and pressure have also to be computed.

In addition to Equation (5.5), for a compressible flow, the entropy-related function S is integrated along trajectories as

$$S(t_l + \Delta t_l) = S(t_l). \tag{5.9}$$

5.2 Eulerian/Lagrangian interaction

In order to achieve an efficient interaction between the Eulerian and the Lagrangian solutions, two different strategies have been implemented for the different flow cases treated here.

5.2.1 Downstream integration of trajectories

The first and more straight forward option consists in positioning the markers at the inlet of the computational domain and tracing them as they convect downstream with the local Eulerian flow. At each time-step along their respective trajectories, the new

Lagrangian state vectors are obtained by integration of the source-terms and the markers are used to locally correct the Eulerian flow. The initial values for the Lagrangian state vectors are provided by linear interpolation from the Eulerian flow at the inlet of the computational domain where the inlet boundary conditions are known. At this location, the Eulerian solution does not significantly suffer from numerical diffusion due to the proximity of the boundary conditions. For example, the vorticity field at the inlet of a pipe will be function of the specified inlet velocity profile.

Figure 5.2b) shows a vortex convecting (perpendicular to its axis of rotation) through a contraction. For this unsteady calculation, the markers are initially positioned in the vortical region of the flow (near the vortex core) and then convect downstream together with the vortex. In this unsteady test case, the time-step Δt_l used for the integration of the marker trajectories must be the identical to the Eulerian Δt. Figure 5.2a) shows the sequence of interactions between the Eulerian and the Lagrangian solutions. First the Eulerian solution at time t and the Eulerian solution at time $t + \Delta t$ are used to compute the new marker position following Equations (5.4). The new values of vorticity and entropy are computed using Equations (5.7) and (5.9) for a compressible flow and Equation (5.7) for an incompressible flow. The new Lagrangian state vector is then used to correct the Eulerian state vector for each cell containing a marker at time $t + \Delta t$.

Figure 5.2c) represents a schematic for a steady swirling flow through a pipe. The markers are injected from the inlet of the domain at regular time intervals in such a way as to have approximately one marker per cell in the vortical regions of the domain. As markers are convected downstream and exit the domain, it is necessary to reinject them at the inlet. The trajectories and the source-term integration are computed in the same manner as for the unsteady process but now since the Eulerian solver uses a pseudo-time marching, the Lagrangian time-step does not need to equal the Eulerian time-step. The Lagrangian time-step is now fixed by accuracy considerations in the trajectory integration. The interaction between the Eulerian and the Lagrangian solvers is again described by Figure 5.2a).

The downstream integration of the trajectories presents the advantage that no a priori knowledge of the location or structure of the flow features is necessary. For an unsteady flow, the markers are simply positioned in the features area at the initial time. The use of identical Lagrangian and Eulerian time-steps ensure the presence of markers in the vortex area for each time. For the steady swirling flow in a pipe, the markers can be positioned in the entire inlet cross-section. In this case, however, it can be inferred that placing markers only near the wall region of the pipe at the inlet is sufficient to trace the strong vorticity regions downstream.

On the other hand, for this last flow example, the markers are subjected to a strong redistribution during their convection, such that an even correction of the Eulerian solution can not be ensured. In order get a more even distribution of markers in the areas where a correction is required, a second Eulerian/Lagrangian interaction strategy is used.

5.2.2 Upstream integration of trajectories

The second interaction strategy is illustrated in Figure 5.2d) and (5.2e). For the same steady swirling flow through a pipe, the markers are initially placed at the center of each cell and the trajectories and source-terms are integrated upstream until the markers reach the inlet. The time-step for the integration is again fixed for each marker by accuracy considerations in the trajectory integration. At the inlet, the Lagrangian state vectors are found by linearly interpolating the inlet (non-diffused) Eulerian solution to the markers location. By adding the integrated source-terms to the inlet state vectors the marker state vector at the center of each cell is determined. The correction procedure can then take place evenly from the center of each cell. Figure 5.2d) describes the interaction between the Lagrangian and the Eulerian solvers. The integration of the source-terms requires interpolations from the Eulerian state vector at time t. Once the value at the center of each cell is determined, the correction of the Eulerian solution is performed. Then the Lax-Wendroff algorithm is applied on the corrected solution. At this point, the trajectories could be computed again, but it has been found that iterating the correction step and the Lax-Wendroff algorithm, while keeping the Lagrangian state vector and the marker trajectories fixed, leads to a faster convergence. On the other hand, to ensure stability, the trajectories should be recomputed before the Eulerian solution changes are too large. Also, the trajectories do not have to be traced backwards to the inlet, but can be stopped at any previous upstream location. If the trajectories are stopped before the inlet, the corrections will be smaller and the convergence will take more time since the full correction is obtained only when the cells where the markers stop have been fully corrected. Again, in order to minimize CPU time, the markers can be placed only in the flow regions where corrections are required.

For an unsteady flow solution using the upstream integration of the trajectories means that the trajectories have to be recomputed at each time-step. Again, the trajectories can be integrated backwards until they reach the inlet or be integrated over only a few time-steps. At any backward position, though, the Eulerian solution at that particular time is required to compute the source-terms and the trajectories. Therefore, in order to minimize the number of Eulerian solutions to store, the trajectories can be integrated backwards over only one time-step. For unsteady flows, the downstream integration of trajectories is, therefore, simpler and much more efficient.

In summary, the downstream integration is the more straight forward method, especially for unsteady flows. For steady flows, the use of the upstream integration of the trajectories allows a spatially-even correction of the Eulerian solution. This has met with more success for the calculation of swirling flows and secondary flows. Both trajectory integrations are further discussed in relation with the markers positioning in the flow.

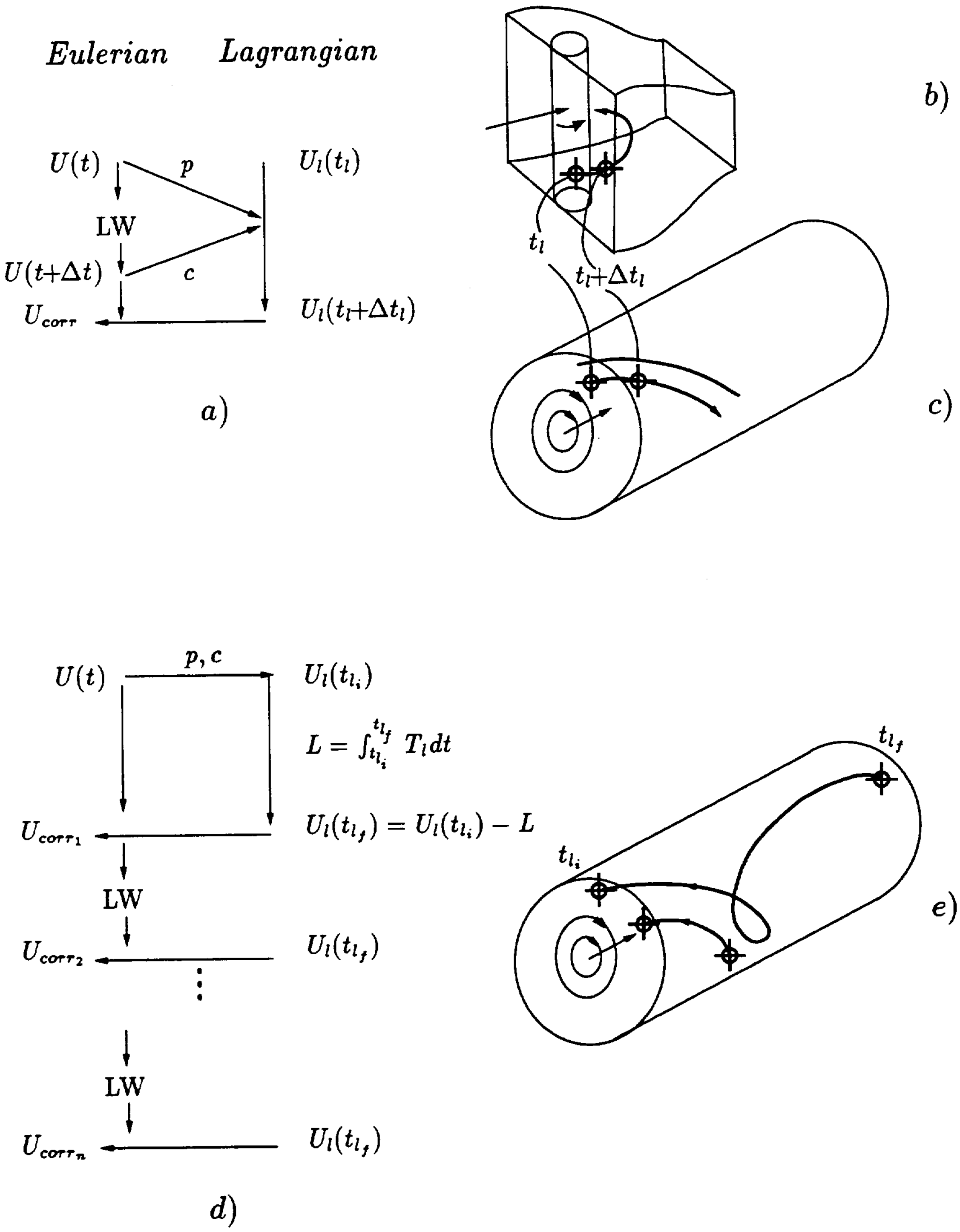


Figure 5.2: Eulerian/Lagrangian interaction procedures *for downstream integration of the trajectories:* a) schematic of Eulerian/Lagrangian interaction, b) unsteady vortex convection in contraction, c) steady swirling flow in pipe, *for upstream integration of the trajectories:* d) schematic of Eulerian/Lagrangian interaction, e) steady swirling flow in a pipe.

5.3 Positioning of the markers in the flow

Since each marker corrects the Eulerian solution locally, the markers need to be located only in the regions of the flow where a correction is required. This property is particularly useful in terms of CPU time reduction when the flow features are concentrated in an otherwise smooth background flowfield. In contrast, the 'cloud-in-cell' technique of Baker [5] uses finite-size vortices which influence the entire flowfield and whose geometry and circulation must be known.

When using a downstream integration, the built-in convection properties of the Lagrangian technique allow for the steady or unsteady tracing of flow features without an a priori knowledge of their placement or structure. In the upstream integration of the trajectories, the area covered by the flow features must be broadly known before. It is usually possible to determine the general area where corrections are required from a basic knowledge of the flow. However, in some cases like the secondary flow in a pipe, the features of interest are so dispersed in the flow that markers have to cover almost the entire flow area.

If the markers are convecting downstream and locally correcting the Eulerian flow, weighting factors are required when distributing corrections to the grid nodes because at any time-step the markers are not distributed uniformly with respect to the Eulerian grid, (each node may not be influenced by the same number of markers and the distances from the nodes to the markers are different). Moreover the divergence/convergence of the trajectories can create holes in the markers distribution. In the case illustrated in Figure 5.3a), because of the divergence of the streamlines, the markers are redistributed as they convect from the inlet downstream. This results in some nodes in the field not being influenced by any markers, whereas other nodes are influenced by several markers. When the correction is large enough, this uneven correction applied to the Eulerian scheme can lead to numerical instability.

The use of the second interaction option where the markers are placed at the center of each cell and convect upstream eliminates the need for weighting functions. Since the distribution of the corrections is done from the cell centers, each grid node is influenced by the same number of markers disposed uniformly. The problem of redistribution of the markers during their convection is also eliminated as presented in Figure 5.3b). Nevertheless, as pointed out before, the choice of upstream integration is less suitable for unsteady flow calculations.

The assumption that the Lagrangian state vector is piecewise constant in each cell makes the representation of strong gradients inaccurate on very coarse grids. This case is illustrated in Figure 5.3c) and 5.3d) for the downstream and upstream integration of the trajectories, respectively. In Figure 5.3c) where the markers are convecting downstream, using the weighting factors results in an average correction for the field cells. This is clearly inaccurate if the gradient between the Lagrangian state vectors is high. In the case of upstream integration represented in Figure 5.3d), the presence of only one marker in each field cell is ensured. However, because of the lack of grid resolution and the presence of only one marker per cell in the field region, when the markers trajectories are diverging (in an upstream integration), the final distribution of markers will be sparse in the inlet region as shown in Figure 5.3d). The inlet information between the two streamlines is not 'seen' or transported by the markers. Both problems illustrated in Figures 5.3c) and 5.3d) are linked to a lack of grid resolution. The problem of high vorticity gradients does not appear in the Eulerian solution alone, since numerical diffusion results in lower vorticity gradients in the field. The Lagrangian solution, however, is immune to numerical diffusion and presents stronger vorticity gradients.

The different flow cases treated here reflect the need for distinct marker initial locations and trajectory integrations. For the unsteady vortex convection, few markers are required in the flow and the markers are attached to the flow feature as it convects downstream. For the steady swirling flow in a pipe, many more markers are required and are initialized near the inlet cross-section. However, since a strong redistribution occurs in a downstream integration, one can also attempt an upstream integration of

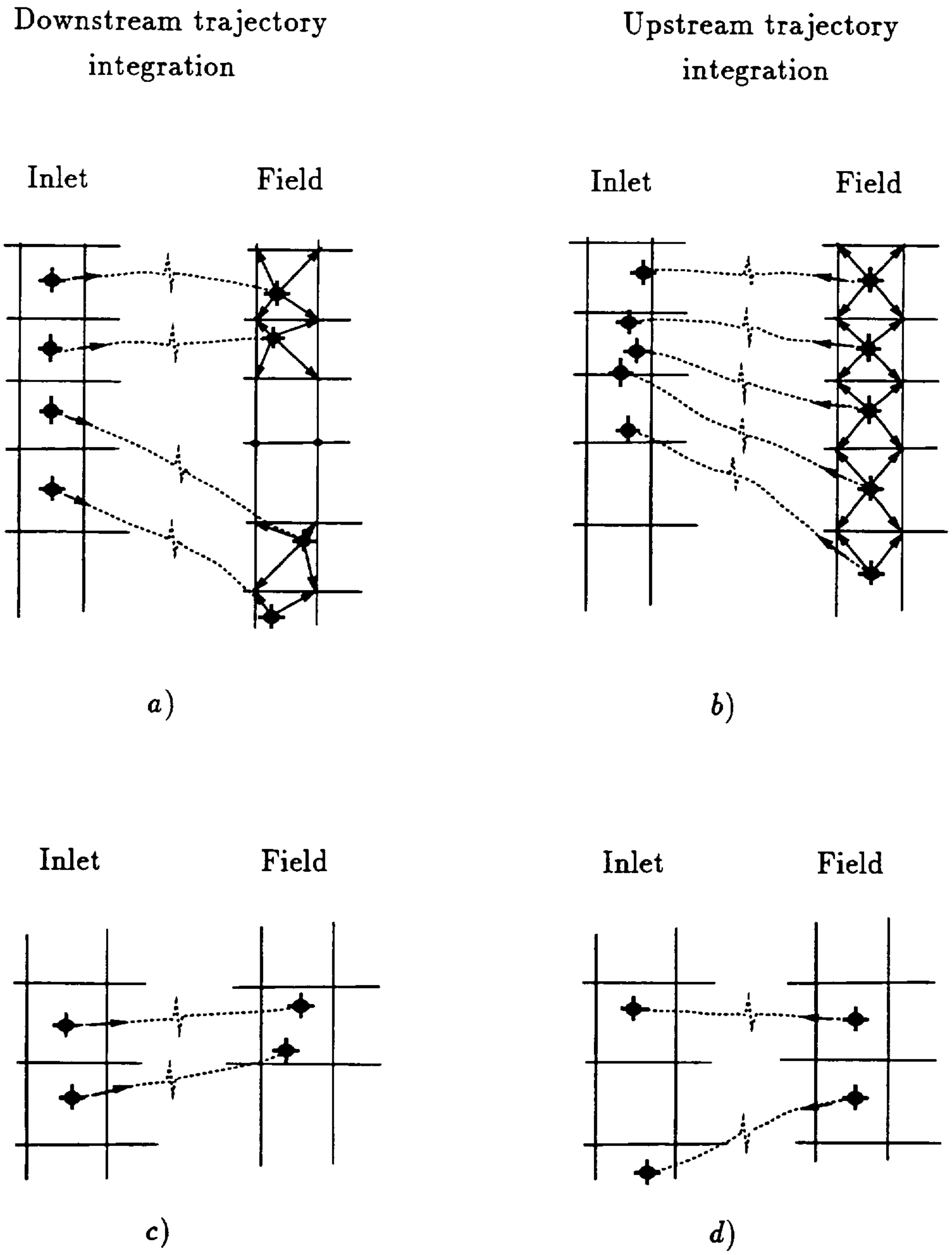

Downstream trajectory integration
Upstream trajectory integration
Inlet
Field
Inlet
Field
a)
b)
Inlet
Field
Inlet
Field
c)
d)

Figure 5.3: Markers redistribution during downstream or upstream convection leads to a) lack of correction of the Eulerian solution, b) even correction, c) average correction in cell, and d) sparse distribution of markers in the inlet region leading to inaccurate representation of inlet flow values by the Lagrangian markers.

the markers by initially locating a marker at the center of each cell. This last procedure ensures an even correction distribution, but the downstream integration of the trajectories remains much simpler for unsteady flow calculations. The lack of grid resolution can create problems when tracking the Lagrangian vorticity upstream or downstream. Section 9.2 addresses this problem by using a pseudo-viscous term in the Helmholtz equation in order to smooth out the strong vorticity gradients of the flow.

Chapter 6

Correction Procedure

This section describes the corrections by which the Lagrangian solution induces changes in the Eulerian state vector. Since the vorticity and entropy are accurately described by the Lagrangian equations, these quantities are used to correct locally the Eulerian solution. For an incompressible flow the vorticity at the marker location is used to locally correct the Eulerian solution. An entropy correction is used in the compressible flow cases in addition to the vorticity correction.

The main difference in terms of correction when using the downstream or upstream trajectory integrations described in the previous section is the location of the markers in the field. With a downstream integration the markers are not placed uniformly in the flow. Therefore, the use of weighting factors is required when interpolating the flowfield values from the grid nodes or from the cell centers to the location of the marker and vice-versa. When using an upstream trajectory integration the weighting functions are not required in the correction procedure since each correction occurs at the initial location of the markers, i.e. from the center of each cell where a marker has been placed. The correction procedure is described here after for a non-uniform marker distribution (or a downstream trajectory integration). The simplifications for a uniform marker distribution (upstream trajectory integration) are also mentioned.

6.1 Entropy correction

For a non-uniform marker distribution in the field, the Eulerian value of the entropy-related function S_e defined by Equation (4.4) is found at the marker location by linear interpolation from the nodes according to Equation (5.3) as

$$S_e = \sum_{k=1}^{8} N_k S_k. \tag{6.1}$$

This value is then compared to the Lagrangian value of the entropy-related function S_l at the marker location. The difference

$$\Delta S = S_l - S_e \tag{6.2}$$

is considered as an error in the Eulerian solution at the location of the marker. This error is then distributed back to the nodes as illustrated in Figure 6.1. When using a downstream trajectory integration, each node may be influenced by a different number of markers, so that the resulting change in S at a node is found by a weighted average over all the markers influencing that node (i.e. all the markers located in cells to which that node belongs)

$$\delta S = \frac{\sum_m ((N_k^2)_m \Delta S_m)}{\sum_m (N_k)_m}, \tag{6.3}$$

If the markers are located at the cell centers, Equations (6.1) and (6.3) simply become

$$S_e = \frac{1}{8} \sum_{k=1}^{8} S_k, \tag{6.4}$$

and

$$\delta S = \frac{1}{8} \sum_{m=1}^{8} \Delta S_m, \tag{6.5}$$

respectively. The change in the S value at a node is then translated into changes in the Eulerian flow field variables.

It is necessary to fix four of the five flow variables in order to induce a change in entropy in the Eulerian solution. If the conservation variables $\rho u, \rho v, \rho w$ and the pressure

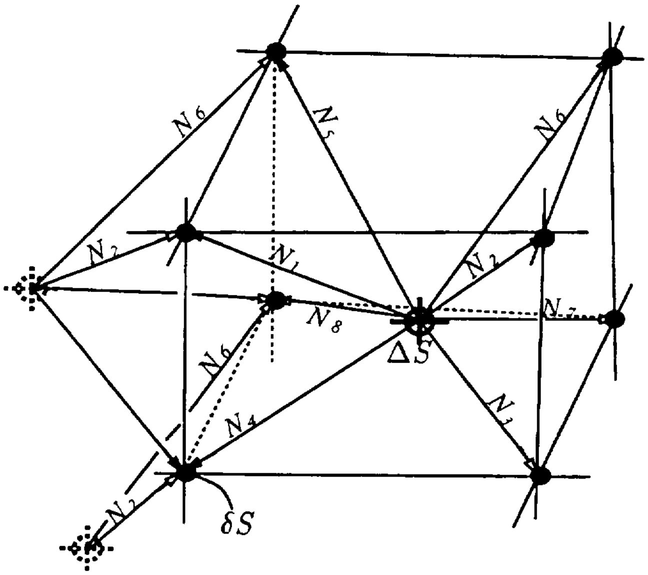


Figure 6.1: Distribution of the error in entropy-related function ΔS from markers to nodes using weighted average.

are kept constant on the grounds that these elliptic quantities are well represented on the Eulerian grid, then the change in S is translated into changes in the conservative variables as

$$\delta\rho = -\frac{\rho}{\gamma S}\delta S \tag{6.6}$$

$$\delta(\rho u) = 0 \tag{6.7}$$

$$\delta(\rho v) = 0 \tag{6.8}$$

$$\delta(\rho w) = 0 \tag{6.9}$$

$$\delta(\rho e_0) = -\frac{u^2+v^2+w^2}{2}\delta\rho \tag{6.10}$$

Another option, similar to the approach used by Giles for his entropy smoothing in [23], consists in keeping the vorticity constant during the entropy correction since the correction in vorticity is the object of a separate treatment. This is achieved by updating the density and energy and keeping the velocity and pressure constant. Then

$$\begin{aligned}
\delta\rho &= -\frac{\rho}{\gamma S}\delta S && (6.11)\\
\delta(\rho u) &= u\delta\rho && (6.12)\\
\delta(\rho v) &= v\delta\rho && (6.13)\\
\delta(\rho w) &= w\delta\rho && (6.14)\\
\delta(\rho e_0) &= \frac{u^2+v^2+w^2}{2}\delta\rho && (6.15)
\end{aligned}$$

In practice, both distribution methods give the same results.

6.2 Vorticity correction

6.2.1 Vorticity error at cell centers

The Eulerian vorticity $\vec{\omega}$ is first computed at the center of each cell by assuming a cellwise constant vorticity. Then

$$\int_V \vec{\omega} dV = \vec{\omega} V_c \qquad (6.16)$$

where the integral is performed over the cell volume V_c. Using Stokes theorem the vorticity is

$$\vec{\omega} = \frac{1}{V_c}\int_V \vec{\omega} dV = \frac{1}{V_c}\int_V (\nabla \times \vec{v}) dV = \frac{1}{V_c}\int_S (\vec{n} \times \vec{v}) dS \qquad (6.17)$$

where the surface integral is performed over the cell faces and $\vec{n}$ is the surface unit normal vector. Thus, summing over the six cell faces, the components of vorticity in Cartesian coordinates are

$$\omega_x = \frac{1}{V_c}\sum_{i=1}^{6}(\overline{w}\, S_y - \overline{v}\, S_z)_i$$

$$
\begin{aligned}
\omega_y &= \frac{1}{V_c}\sum_{i=1}^{6}(\overline{u}\ S_z - \overline{w}\ S_x)_i, \qquad (6.18)\\
\omega_z &= \frac{1}{V_c}\sum_{i=1}^{6}(\overline{v}\ S_x - \overline{u}\ S_y)_i
\end{aligned}
$$

On each face, the values for $\overline{u}, \overline{v}$ and $\overline{w}$ are obtained from averages over the velocities at the four corner nodes.

In the case of a non-uniform marker distribution, the Eulerian vorticity is then interpolated to the location of the marker by using a second local coordinate system (ξ^*, η^*, ζ^*) based on the eight cell centers that the marker is the nearest to (these 8 cells are easily determined by using the array containing the 26 cell neighbours and the marker local coordinates (ξ, η, ζ)). The ξ, η, ζ directly define the (ξ^*, η^*, ζ^*) coordinates as shown by the example of Figure 6.2 (represented in a 2-D case for clarity).

The 8 tri-linear interpolating functions $N_1^*(\xi^*, \eta^*, \zeta^*)$ to $N_8^*(\xi^*, \eta^*, \zeta^*)$ are used to interpolate the Eulerian vorticity from the eight cell centers to the marker location as

$$\vec{\omega}_e = \sum_{c=1}^{8} N_c^* \vec{\omega}_c. \qquad (6.19)$$

This value is then compared to the Lagrangian value of vorticity at the marker location $\vec{\omega}_l$. The difference

$$\Delta\vec{\omega} = \vec{\omega}_l - \vec{\omega}_e \qquad (6.20)$$

is taken as an error in the Eulerian solution at the location of the marker. This error is first distributed back to the center of the eight cells by using a weighted average over all the markers influencing this cell center.

$$\delta\vec{\omega}_c = \frac{\sum_m (N_k^{*2})_m \Delta\vec{\omega}_m}{\sum_m (N_k^*)_m}. \qquad (6.21)$$

All the markers influencing a cell center are contained in a volume based on the eight neighbouring cell centers as illustrated by the dashed area in Figure 6.2. If the markers

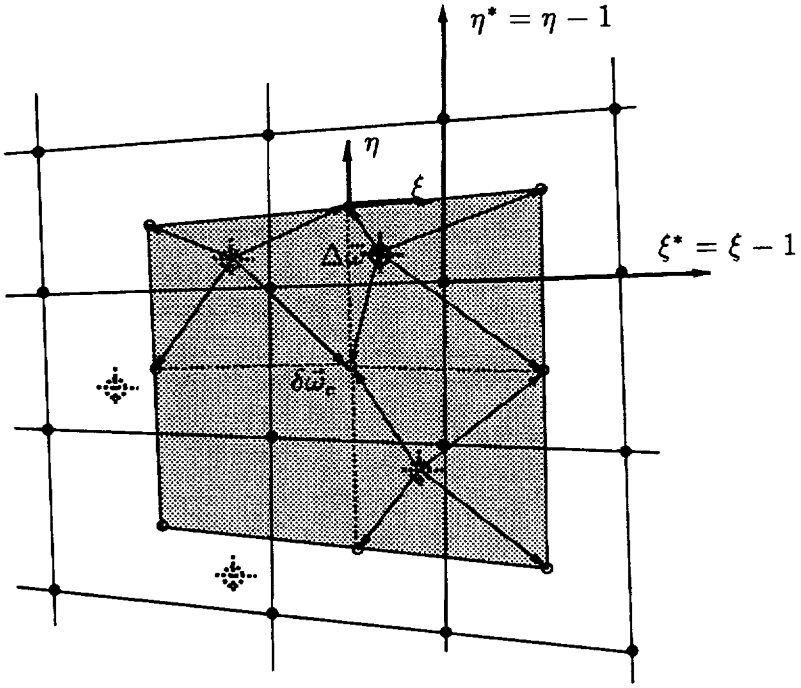


Figure 6.2: Local coordinate system (ξ^*, η^*, ζ^*) centered on nodes (represented for the solid line marker) and vorticity error distribution from markers to cell centers.

are located at the cell centers, Equations (6.19) and (6.21) become

$$\vec{\omega}_e = \vec{\omega}_c, \tag{6.22}$$

and

$$\delta\vec{\omega}_c = \Delta\vec{\omega}, \tag{6.23}$$

respectively. Once the change in vorticity at a cell center is determined, it is translated into changes in the Eulerian velocity vectors at the corner nodes of the cell as described in the next section.

6.2.2 Distribution of vorticity error

The error in vorticity in a cell is translated into changes in the velocity components at the cell nodes. In a 2-D problem, the error in vorticity implies a circulation error around the cell perimeter. The correction is then distributed evenly between the eight velocity components at the four cell nodes as described in [20]. In three dimensions, the vorticity error is a vector, but a similar procedure can be applied for each of the six cell surfaces.

The error in vorticity $\Delta\vec{\omega}$ at the cell center is used to compute an error in circulation on each face of the cell as

$$\Delta\Gamma = \Delta\vec{\omega} \cdot \vec{S}, \tag{6.24}$$

where $\vec{S}$ is the surface vector of the face. Figure 6.3 shows the errors in circulation for the faces 1 4 and 6 (the nomenclature for the face numbering is described in Appendix B). On each face, the error in circulation is then used to find velocity corrections at the four corner nodes as shown in Figure 6.3a). A local coordinate system (σ, τ) and a local node numbering from $1f$ to $4f$ is defined on each face as sketched in Figure 6.3b) for face number 1. The location vector $\vec{r} = (x, y, z)$ is used to define the unit vectors along the local coordinates as

$$\hat{\sigma} = \frac{\vec{r}_{2f} - \vec{r}_{1f} + \vec{r}_{3f} - \vec{r}_{4f}}{||\vec{r}_{2f} - \vec{r}_{1f} + \vec{r}_{3f} - \vec{r}_{4f}||}, \qquad \hat{\tau} = \frac{\vec{r}_{4f} - \vec{r}_{1f} + \vec{r}_{3f} - \vec{r}_{2f}}{||\vec{r}_{4f} - \vec{r}_{1f} + \vec{r}_{3f} - \vec{r}_{2f}||}. \tag{6.25}$$

The local coordinates σ and τ are then written as

$$\sigma = \hat{\sigma} \cdot (\vec{r} - \vec{r}_c), \qquad \tau = \hat{\tau} \cdot (\vec{r} - \vec{r}_c), \tag{6.26}$$

where $\vec{r}_c$ refers to the location vector of the geometric center of the face.

The error in circulation can be written as an integral over the face perimeter as

$$\Delta\Gamma = \oint \Delta u dx + \Delta v dy + \Delta w dz = \oint \Delta u_\sigma d\sigma + \Delta u_\tau d\tau, \tag{6.27}$$

where Δu, Δv, Δw are corrections to the u, v and w components of velocity, respectively.

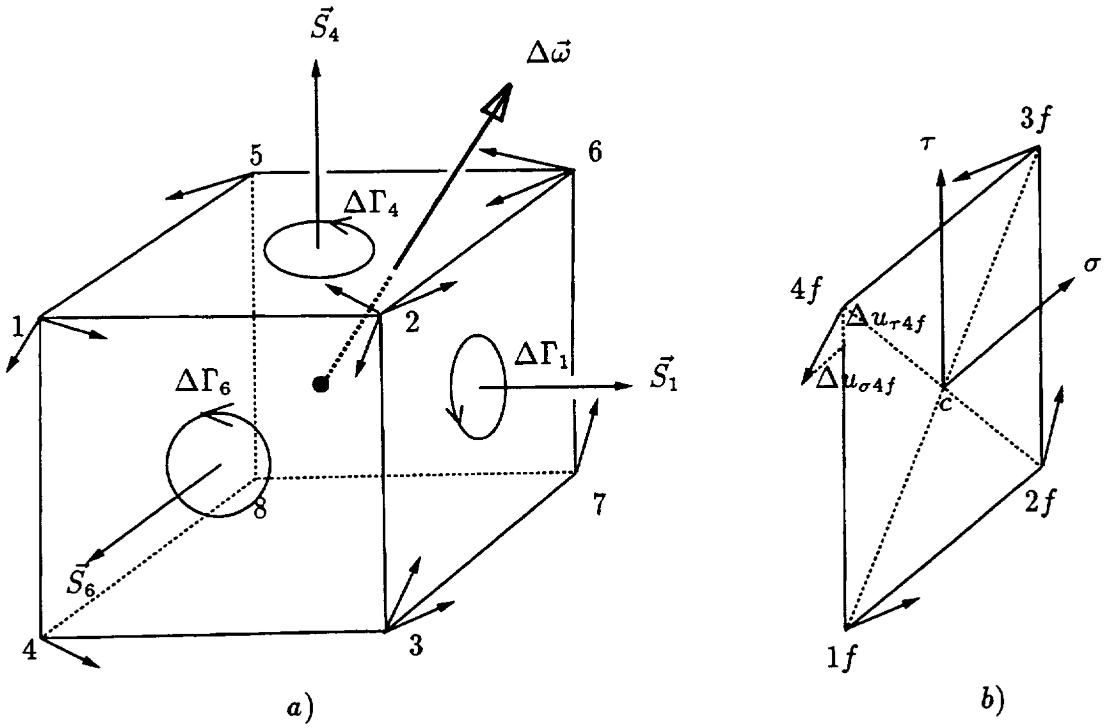


Figure 6.3: a) Velocity correction (contribution from faces 1 4 and 6 only), and b) local coordinate system σ, τ and local node numbering for face number 1.

$\Delta u_\sigma, \Delta u_\tau$ are corrections to the u_σ, u_τ components of velocity expressed in the local coordinate system. Using trapezoidal integration, the error in circulation is written as

$$\begin{aligned}
2\Delta\Gamma &= (\Delta u_{\sigma 1f} + \Delta u_{\sigma 2f})(\sigma_{2f} - \sigma_{1f}) + (\Delta u_{\tau 1f} + \Delta u_{\tau 2f})(\tau_{2f} - \tau_{1f}) + \\
&\quad (\Delta u_{\sigma 2f} + \Delta u_{\sigma 3f})(\sigma_{3f} - \sigma_{2f}) + (\Delta u_{\tau 2f} + \Delta u_{\tau 3f})(\tau_{3f} - \tau_{2f}) + \\
&\quad (\Delta u_{\sigma 3f} + \Delta u_{\sigma 4f})(\sigma_{4f} - \sigma_{3f}) + (\Delta u_{\tau 3f} + \Delta u_{\tau 4f})(\tau_{4f} - \tau_{3f}) + \\
&\quad (\Delta u_{\sigma 4f} + \Delta u_{\sigma 1f})(\sigma_{1f} - \sigma_{4f}) + (\Delta u_{\tau 4f} + \Delta u_{\tau 1f})(\tau_{1f} - \tau_{4f}) \\
&= (\Delta u_{\sigma 1f} - \Delta u_{\sigma 3f})(\sigma_{2f} - \sigma_{4f}) + (\Delta u_{\sigma 2f} - \Delta u_{\sigma 4f})(\sigma_{3f} - \sigma_{1f}) + \\
&\quad (\Delta u_{\tau 1f} - \Delta u_{\tau 3f})(\tau_{2f} - \tau_{4f}) + (\Delta u_{\tau 2f} - \Delta u_{\tau 4f})(\tau_{3f} - \tau_{1f}). \qquad (6.28)
\end{aligned}$$

By choice, the error in circulation is translated into homogeneous velocity corrections for each of the eight velocity components at the four corner nodes. Then

$$\Delta u_{\sigma,\tau} = \Delta u_{\sigma 1f} = \Delta u_{\sigma 2f} = -\Delta u_{\sigma 3f} = -\Delta u_{\sigma 4f}$$

$$= \quad -\Delta u_{\tau 1f} = \Delta u_{\tau 2f} = \Delta u_{\tau 3f} = -\Delta u_{\tau 4f}. \qquad (6.29)$$

Thus, the correction to the components u_σ, u_τ is found as

$$\Delta u_{\sigma,\tau} = \frac{\Delta\Gamma}{P}, \qquad \text{where } P = \sigma_{3f} - \sigma_{1f} + \sigma_{2f} - \sigma_{4f} + \tau_{4f} - \tau_{2f} + \tau_{3f} - \tau_{1f}. \quad (6.30)$$

The transformation from the local face coordinates to the Cartesian coordinates in Equation (6.27) gives the velocity corrections $\Delta u, \Delta v, \Delta w$

$$\begin{aligned}
\Delta\Gamma &= \oint \Delta u_\sigma d\sigma + \Delta u_\tau d\tau \\
&= \oint \Delta u_\sigma \left(\frac{\partial\sigma}{\partial x}dx + \frac{\partial\sigma}{\partial y}dy + \frac{\partial\sigma}{\partial z}dz\right) + \oint \Delta u_\tau \left(\frac{\partial\tau}{\partial x}dx + \frac{\partial\tau}{\partial y}dy + \frac{\partial\tau}{\partial z}dz\right) \\
&= \oint \left(\Delta u_\sigma \frac{\partial\sigma}{\partial x} + \Delta u_\tau \frac{\partial\tau}{\partial x}\right)dx + \oint \left(\Delta u_\sigma \frac{\partial\sigma}{\partial y} + \Delta u_\tau \frac{\partial\tau}{\partial y}\right)dy + \oint \left(\Delta u_\sigma \frac{\partial\sigma}{\partial z} + \Delta u_\tau \frac{\partial\tau}{\partial z}\right)dz \\
&= \oint \Delta u \; dx + \Delta v \; dy + \Delta w \; dz,
\end{aligned}$$

so that

$$\begin{aligned}
\Delta u &= \Delta u_\sigma \frac{\partial\sigma}{\partial x} + \Delta u_\tau \frac{\partial\tau}{\partial x}, \\
\Delta v &= \Delta u_\sigma \frac{\partial\sigma}{\partial y} + \Delta u_\tau \frac{\partial\tau}{\partial y}, \qquad (6.31) \\
\Delta w &= \Delta u_\sigma \frac{\partial\sigma}{\partial z} + \Delta u_\tau \frac{\partial\tau}{\partial z},
\end{aligned}$$

where

$$\begin{aligned}
\frac{\partial\sigma}{\partial x} &= \frac{x_{2f} - x_{1f} + x_{3f} - x_{4f}}{\|\vec{r}_{2f} - \vec{r}_{1f} + \vec{r}_{3f} - \vec{r}_{4f}\|}, & \frac{\partial\tau}{\partial x} &= \frac{x_{4f} - x_{1f} + x_{3f} - x_{2f}}{\|\vec{r}_{4f} - \vec{r}_{1f} + \vec{r}_{3f} - \vec{r}_{2f}\|}, \\
\frac{\partial\sigma}{\partial y} &= \frac{y_{2f} - y_{1f} + y_{3f} - y_{4f}}{\|\vec{r}_{2f} - \vec{r}_{1f} + \vec{r}_{3f} - \vec{r}_{4f}\|}, & \frac{\partial\tau}{\partial y} &= \frac{y_{4f} - y_{1f} + y_{3f} - y_{2f}}{\|\vec{r}_{4f} - \vec{r}_{1f} + \vec{r}_{3f} - \vec{r}_{2f}\|}, \\
\frac{\partial\sigma}{\partial z} &= \frac{z_{2f} - z_{1f} + z_{3f} - z_{4f}}{\|\vec{r}_{2f} - \vec{r}_{1f} + \vec{r}_{3f} - \vec{r}_{4f}\|}, & \frac{\partial\tau}{\partial z} &= \frac{z_{4f} - z_{1f} + z_{3f} - z_{2f}}{\|\vec{r}_{4f} - \vec{r}_{1f} + \vec{r}_{3f} - \vec{r}_{2f}\|}.
\end{aligned}$$

At a node, the contributions to the velocity change from each face are added to give the final velocity correction since, if the faces are perpendicular to each other, the velocity correction on one face does not affect the circulation on another face. Each node receives

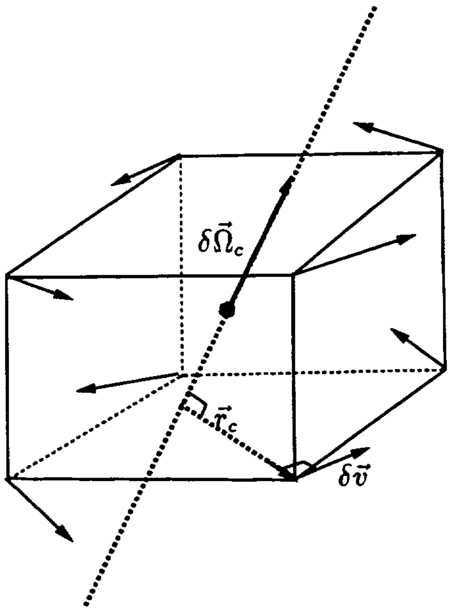


Figure 6.4: Solid-body rotation distribution of error in vorticity to velocity components.

contributions from three faces for a given cell. The final node velocity correction is then found by adding the contributions from the eight cells surrounding the node.

This velocity correction method is analogous to the use of a solid-body rotation assumption where the change in vorticity at a cell center induces changes in velocity components at the 8 nodes of the cell as

$$\delta\vec{v} = \delta\vec{\Omega}_c \times \vec{r}_c, \tag{6.32}$$

where the change in angular speed $\delta\vec{\Omega}_c$ in the cell is defined as half the vorticity error in the cell

$$\delta\vec{\Omega}_c = \frac{1}{2}\delta\vec{\omega}_c, \tag{6.33}$$

and $\vec{r}_c$ is the vector connecting the node to the center of the cell as illustrated in Figure 6.4. The analogy between the previous distribution procedure and the solid-body correction does not hold if the cell faces are of different shapes. Using the previously described correction distribution ensures identical corrections for faces of same (area/perimeter) ratio, independently of the face shape as illustrated in Figure 6.5.

C-2.

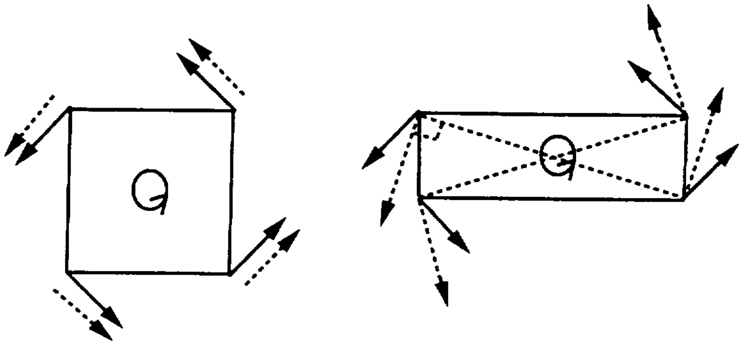


Figure 6.5: Distribution procedure ensures identical velocity correction for faces of same (area/perimeter) ratio, independently of face shape as opposed to the solid-body rotation distribution (dotted line vectors).

Instead, the correction obtained through a solid-body correction does not lead to an identical correction in every velocity component as indicated by the dotted arrows. In the case of a sheared cell, since the local coordinate system is not orthogonal, the corrections are different from the orthogonal case. However, the right correction in circulation and vorticity are still enforced.

For an incompressible flow, summing the contributions from each face described by Equation (6.31) and averaging over the eight surrounding cells, together with

$$\delta p^* = 0, \tag{6.34}$$

give the changes in the Eulerian conservative variables due to the vorticity correction. For a compressible flow, the resulting changes in the conservative Eulerian variables are found by

$$\delta\rho = 0, \tag{6.35}$$

$$\delta(\rho u) = \rho\,\delta u, \tag{6.36}$$

$$\delta(\rho v) = \rho\,\delta v, \tag{6.37}$$

$$\delta(\rho w) = \rho\,\delta w, \tag{6.38}$$

$$\delta(\rho e_0) \;=\; 0. \tag{6.39}$$

The actual implementation of the correction procedure is performed in two steps. First, the contribution from each surrounding face to a node is computed for a unit value of $\Delta\Gamma$ on the face (using Equations (6.30) and (6.31) with $\Delta\Gamma = 1$). For example, in Figure 6.3 node 2 receives contributions from faces 1, 4 and 6. By using a unit value for $\Delta\Gamma$, this step reduces to a simple metrics calculation required only once at the beginning of the computation. Then, at each iteration in time, the error in vorticity at the cell center is determined by Equation (6.21) or (6.23) and the the contributions for a unit circulation are multiplied by the actual value of $\Delta\Gamma$ for each face. Then the contributions of the three faces are summed for each node. The velocity correction at a node is then found from by adding the corrections due to the eight surrounding cells. Since this last step involves a loop over the cells, it is fully vectorizable.

6.3 Boundary conditions for velocity correction

The nodes located on the limits of the domain receive contributions from less than eight cells and, therefore, require special treatment. The imposition of boundary conditions is performed during the first step process for nodes located on domain boundaries and results in modifications of the contributions to these particular nodes.

In this work, four types of domain boundaries are present : inlet, exit, wall and symmetry-plane. The correction of velocity for the nodes located on an inlet boundary is set to zero, since the Eulerian values are exactly set through the inlet boundary condition. The nodes located on the exit surface miss the contributions from the cells placed downstream of the exit surface. Pseudo-cells are defined with the same error in vorticity at their center than the one for cells lying inside the domain as illustrated in Figure 6.6a). This is the method used for the flow over a three-dimensional wing where the trailing vortex axis crosses the exit boundary almost perpendicularly. For the secondary flow calculation in a circular pipe, the axis of the secondary vortex crosses

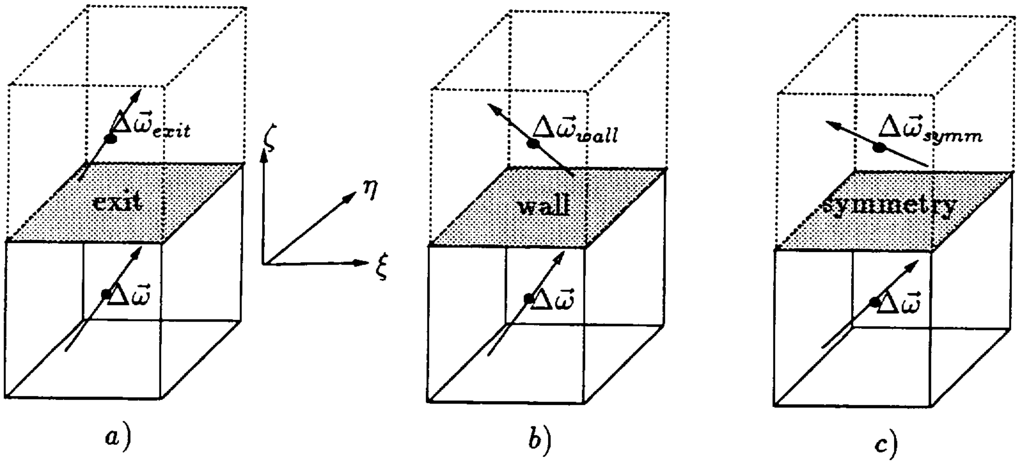


Figure 6.6: Correction in vorticity for pseudo-cells placed at a) an exit surface, b) a wall surface, and c) a symmetry surface.

the exit boundary with a smaller angle and a better approach is to use the same type of exit boundary condition as the Eulerian solver. The 1-D characteristics theory is then applied to a vector formed by the corrections in velocity components.

The imposition of wall and symmetry-plane boundary conditions uses pseudo-cells as sketched in Figures 6.6b) and 6.6c), respectively.

If ζ is the local coordinate normal to the exit,wall or symmetry-plane, then

$$\Delta\vec{\omega}_{exit} = (\Delta\omega_\xi, \Delta\omega_\eta, \Delta\omega_\zeta), \tag{6.40}$$

$$\Delta\vec{\omega}_{wall} = (-\Delta\omega_\xi, -\Delta\omega_\eta, \Delta\omega_\zeta), \tag{6.41}$$

$$\Delta\vec{\omega}_{symm} = (-\Delta\omega_\xi, -\Delta\omega_\eta, \Delta\omega_\zeta). \tag{6.42}$$

The corresponding velocity corrections are sketched in Figure 6.7 for faces 6 , 4 and 1, respectively. The additional velocity corrections due to the presence of the pseudo-cells are represented as dashed vectors. Figure 6.7a) shows the case of an exit boundary condition whereas the case of a wall or symmetry-plane boundary condition is presented in Figure 6.7b).

6.4 Discretisation error for velocity correction

The discretisation error obtained by using the Lagrangian correction on a finite-size grid is estimated here for the case of a linear vorticity profile (say a Poiseuille flow in a channel). The vorticity is written as a linear function of channel height z as

$$\omega = k\ z, \tag{6.43}$$

where k is a proportionality factor. In the Lagrangian technique, the vorticity is assumed to be a constant within each cell and takes the value at the cell center. By expanding the vorticity as a function of the channel height into a Taylor expansion to the first order, the discretisation error e_ω in a cell is

$$e_\omega = \frac{\partial \omega}{\partial z}\Delta z \sim k\ l, \tag{6.44}$$

where Δz is the distance to the cell center and l is a representative cell length. Using Equation (6.30), the velocity correction Δu is related to the vorticity correction as

$$\Delta u = \frac{\Delta \Gamma}{P} = \frac{\Delta \omega\ S}{P} \sim \Delta \omega\ l. \tag{6.45}$$

Thus, the discretisation error in velocity correction e_u is

$$e_u \sim e_\omega\ l \tag{6.46}$$

and using Equation (6.44), the error in velocity is shown to be second-order on the finite-size grid

$$e_u \sim k\ l^2. \tag{6.47}$$

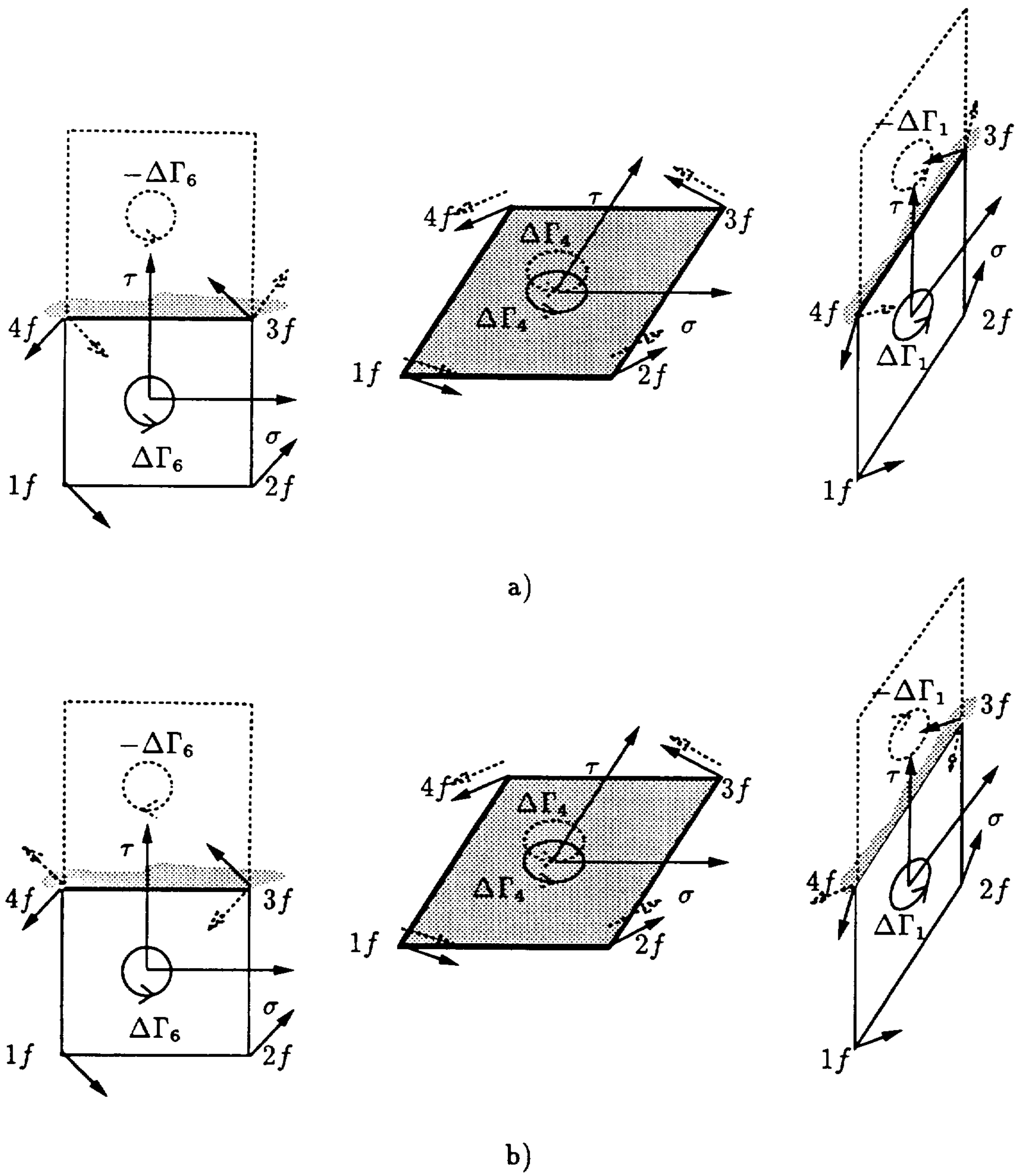


Figure 6.7: Contribution to velocity corrections from faces 6, 4 and 1 in the case of a) exit boundary condition, and b) wall or symmetry-plane boundary condition (additional contributions from pseudo-cells are represented as dashed line vectors).

6.5 Iterative procedure for the correction of vorticity

The previous section's subject was the translation of an error in vorticity, defined at the cell center, into errors in circulation around each face of the cell, and finally into velocity corrections at the corner nodes. This section shows why a 'recursive' correction process is required in order to obtain the correct circulation around each cell. The different steps of the iterative correction procedure are presented as well as a numerical study of the process's convergence rate for a simple two-dimensional case.

As will be shown, the vorticity correction requires an iterative procedure. For steady flows, the vorticity correction is converging along with the pseudo time-marching Eulerian scheme and only one step is usually required for the vorticity correction at each Eulerian iteration. In the case of an unsteady flow, the vorticity correction should be iterated until convergence for each Eulerian time-step. However, using only one step of the vorticity correction at each Eulerian iteration has given satisfactory results.

In this study, the correction is applied recursively on an initial solution without taking any Eulerian steps. However, the accuracy of the Eulerian solution enters into the analysis of the convergence rate of the combined Eulerian/Lagrangian scheme.

Figure 6.8 shows a two-dimensional numerical domain with three cells in each direction. The initial solution is such that the circulation in each cell is zero. The desired circulation in each cell is assumed to be unity. Therefore, a uniform circulation correction is imposed for each cell ($\Delta\Gamma = 1$). In the first step, velocity corrections are determined for each cell such as to satisfy the circulation correction according to Section 6.2.2. The resulting velocity correction at a node is then found by adding the contributions from the surrounding cells. When all the contributions are added, the circulation in each cell is modified. Hence, a recursive correction is required. In the second step, the modified circulation in each cell is compared to the desired circulation in order to define a new circulation error. This is in turn translated into new velocity corrections at the corner nodes. The procedure is iterated until the correct circulation in each cell is obtained.

Figure 6.9a) shows the maximum error in circulation $\Delta\Gamma_{max}$ in the domain as a function of the iteration number n for six grid sizes (2×2 to 33×33 nodes), each grid having twice as many cells in each direction than the previous one. $\Delta\Gamma_{max}$ is normalized by the initial constant error in the domain $\Delta\Gamma_{initial}$. The number of iterations that the information takes to travel from the boundaries to the center cell is shown by a constant error in circulation during the first iterations (the number of iterations with constant error corresponds to half the domain size in one computational direction). The convergence rate is slower with increased grid size. However, each time the grid size is increased, the initial error in the domain is reduced by a factor 4 since the Lax-Wendroff scheme is second-order accurate.

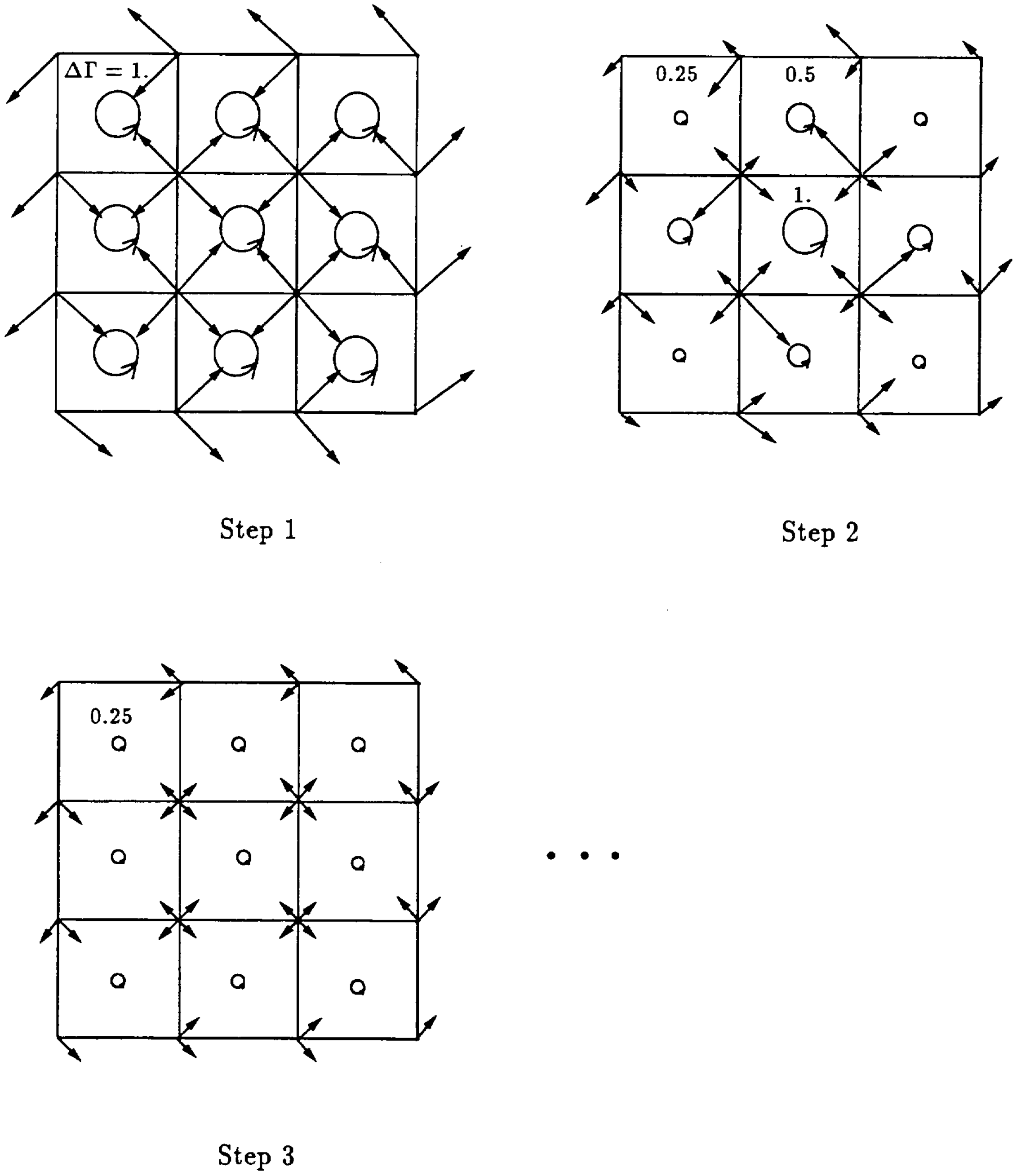
In Figure 6.9b), the ratio of the maximum error in circulation between two consecutive iterations is shown to converge to a value dependent only on the grid size. Figure 6.10 displays the behavior of the difference between these values and unity (1 represents no convergence) as a function of the mesh size h, plotted in a logarithmic scale. This measure of 'how slowly the correction procedure is converging', is shown to vary as a second-order function of the mesh size. However, the initial error given by the Eulerian scheme is also a second-order function of the grid size. Thus, as the convergence rate of the iterative procedure goes to zero at the limit of an infinite grid resolution, the truncation errors due to the Lax-Wendroff scheme vanish too. Thus, the resulting solution is still consistent.

The extension to three dimensions is straight forward. As mentioned earlier, the velocity corrections on one face do not induce any circulation on another face if the faces are perpendicular to each other. Thus, the results for a two-dimensional case remain unchanged in three dimensions. (However, the velocity corrections have to be divided by a factor 2 for the 'interior' faces since when performing the three-dimensional vorticity correction in a unstructured fashion by a loop over the cells, each face (except the boundary faces) is taken twice into account). Also, the convergence rate remains unchanged if the cells faces are rectangular instead of square.

In the steady-state calculations, at each iteration the corrections can be multiplied by a factor smaller than 1 in order to limit the perturbations to the Eulerian solution. In the flow over a 3-D wing, the magnitude of the correction in vorticity for the trailing vortex is of the same order as the basic vorticity in the flow. If the corrections are limited, the convergence rate to the correct circulation around each cell is slower and more iterations are required between the Eulerian and the Lagrangian solutions. Another option consists of under-relaxing the circulation corrections by using the 'old' circulation correction

$$\Delta\Gamma^{n+1} = \Delta\Gamma^{n} + R_f(\Delta\Gamma^{n+1} - \Delta\Gamma^{n}), \tag{6.48}$$

where R_f is a relaxation factor smaller than 1 and n and $n+1$ refer to the old and present iterations of the correction procedure, respectively.



Step 1

Step 2

Step 3

Figure 6.8: Recursive vorticity correction: error in circulation in cells and corresponding velocity corrections.

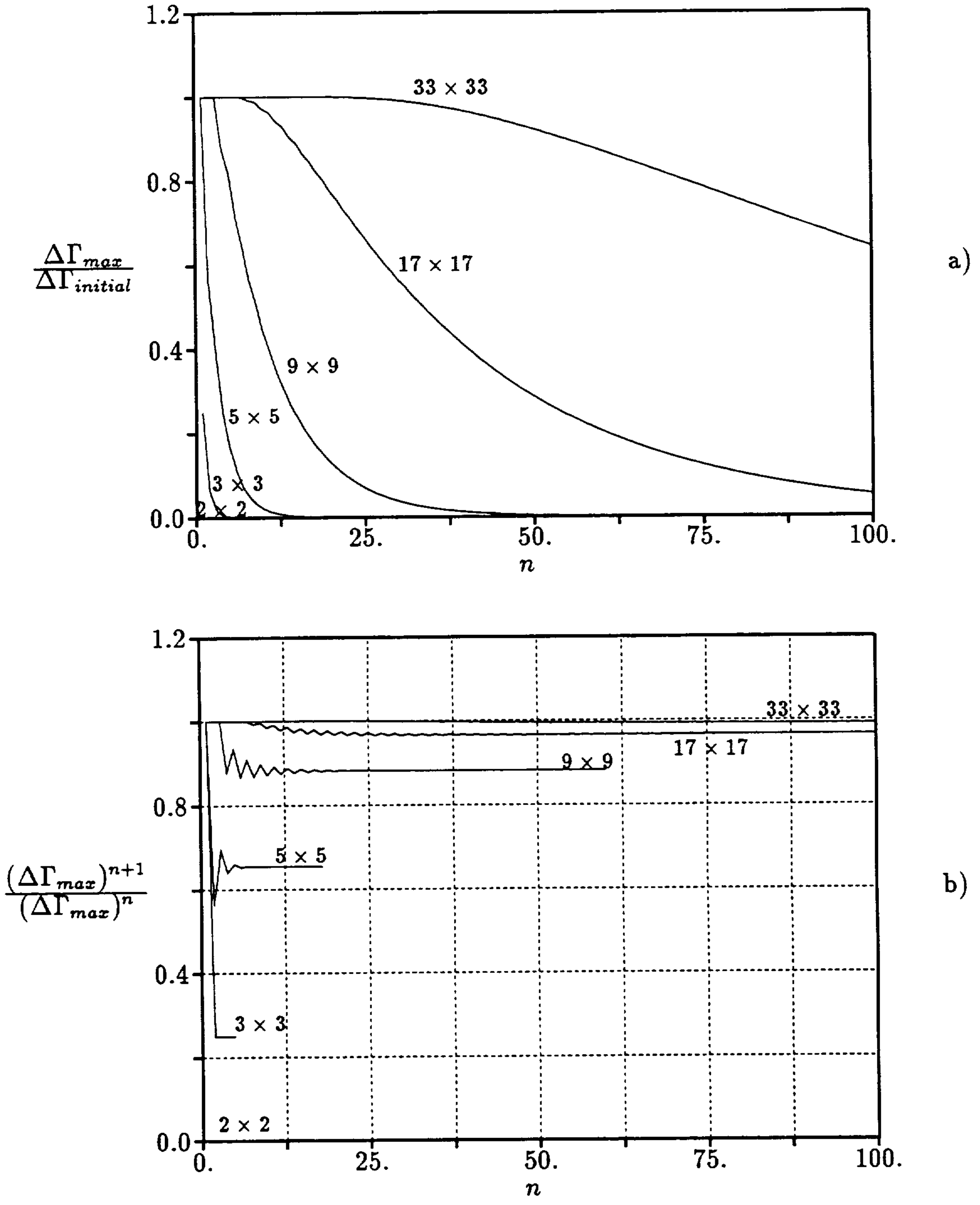


Figure 6.9: Convergence of recursive vorticity correction: a) maximum error in circulation in domain (referenced to initial error in circulation) as a function of the iteration number n and b) ratio of the maximum circulation error between two consecutive iterations.

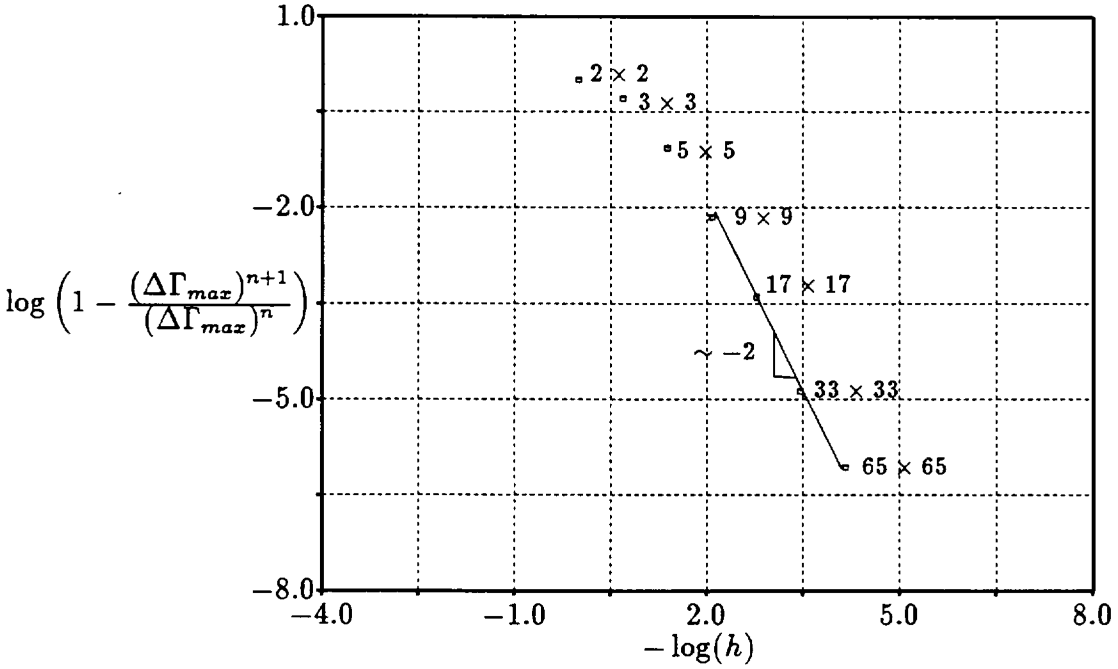


Figure 6.10: Measure of 'how slowly the iterative procedure is converging' as a function of the grid size.

Chapter 7

Vortex Preservation Test Case

The convection of a Lamb vortex in a three-dimensional uniform flow is computed as a vortex preservation test case in a compressible unsteady flow field. The vortex convects perpendicularly to its axis of rotation along a straight channel over a distance of approximately 50 core radii. The tangential velocity v_θ in the frame of reference of the Lamb vortex is given by

$$v_\theta = \frac{\Gamma}{2\pi r}\left(1 - e^{-\left(\frac{r}{a}\right)^2}\right), \tag{7.1}$$

where Γ is the vortex circulation, a is the vortex core radius and r is the distance from the center of the vortex. The initial Eulerian and Lagrangian state vectors are obtained by superimposing the vortex on the uniform background flow. The entropy is found as a function of the radius by numerically integrating the radial entropy gradient given by Crocco's relation

$$\left(h_{0v} - \frac{1}{2}v_\theta^2\right)\frac{d(ln(S))}{dr} = -\gamma\,\frac{v_\theta}{r}\,\frac{d(rv_\theta)}{r}, \tag{7.2}$$

where v_θ is given by Equation (7.1) and the stagnation enthalpy h_{0v} is assumed constant in the vortex frame of reference. Once the entropy-related function S defined by Equation (4.4) is found, the density and pressure follow from

$$\rho \;=\; \left(\frac{\gamma - 1}{\gamma S}\,(h_{0v} - \frac{1}{2}v_\theta^2)\right)^{\frac{1}{\gamma - 1}} \tag{7.3}$$

$$p \;=\; \rho\frac{\gamma - 1}{\gamma}\,(h_{0v} - \frac{1}{2}v_\theta^2) \tag{7.4}$$

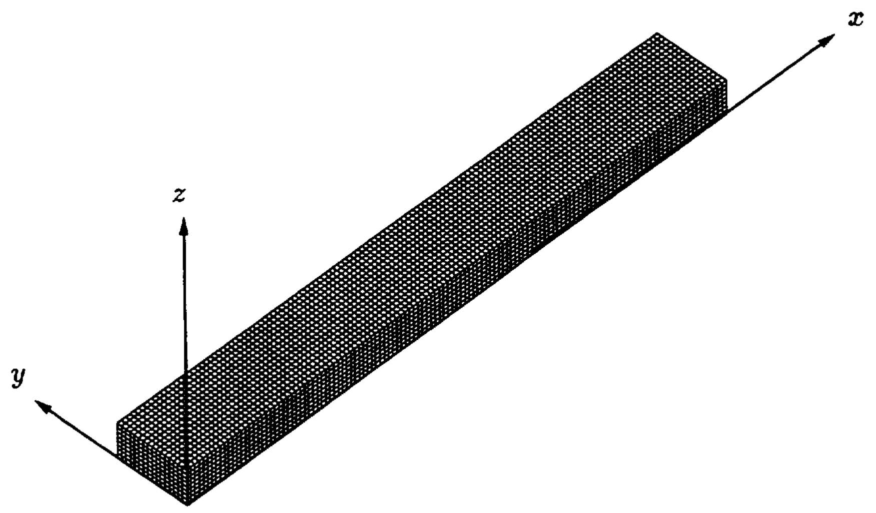


Figure 7.1: Computational grid for vortex preservation case (129 × 17 × 9 nodes).

A uniform flow Mach number of 0.7 and a value for Γ/a of twice the freestream stagnation speed of sound are used in this calculation. The resulting maximum tangential Mach number in the frame of reference of the vortex is approximately 0.2.

The grid used for the calculation is composed of 129 × 17 × 9 nodes and is shown in Figure 7.1. The channel width and height are 1/8 and 1/16 of the channel length, respectively. The mesh spacing is uniform and equal in each direction and chosen such as to have four cells across the vortex core diameter, hence

$$\frac{\Delta x}{a} = \frac{\Delta y}{a} = \frac{\Delta z}{a} = 2. \tag{7.5}$$

A 'no-flux' boundary condition is imposed on the top and bottom walls of the channel. On the inlet, exit and side surfaces of the channel, the exact solution of the convecting vortex is imposed. This allows a minimization of the size of the computational domain (the half-width of the channel measures only four vortex core radii).

The results are presented on a mesh surface at channel mid-height (the solution is independent of the channel height). The initial and final marker distributions for

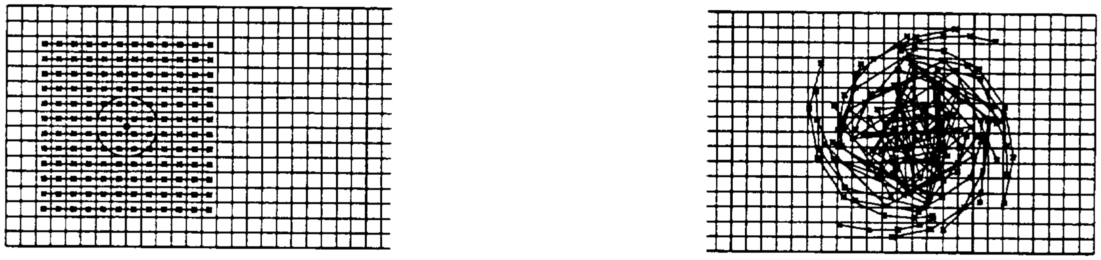


Figure 7.2: a) Initial distribution of markers with core size and b) final distribution.

this calculation are represented in Figures 7.2a) and 7.2b), respectively. The markers are initially placed in the vortex neighbourhood with one marker per cell. Additional markers are placed on the vortex axis (only 17 markers are located in the vortex core in a 2-D plane). The total number of markers for this calculation is 1160. The Eulerian solution and the Eulerian/Lagrangian solution are performed on the same grid. For this unsteady flow case, the same time-steps are used in the Eulerian and Lagrangian integration scheme, hence the markers convect at the same speed as the vortex. The Lagrangian state vector consists of the position, the vorticity and the entropy-related function S. The trajectories are integrated downstream and the computation of the source-terms for the vorticity gives no contribution due to the absence of three-dimensional effects. The interaction sequence between the Eulerian and the Lagrangian solvers follows the procedure described in Section 5.2.1. Only one correction step is performed at each Eulerian iteration at the current location of the markers.

Figures 7.3a), 7.4a) and 7.5a) show the initial pressure, vorticity and S function contours, respectively, when the vortex is located at the inlet of the channel (all flow values are referenced as described in Section 2.3). After the vortex has traveled approximately 53 core radii, the final pressure, vorticity and S function contours are represented in Figures 7.3b), 7.4b) and 7.5b) when using the Eulerian formulation alone and in Figures 7.3c), 7.4c) and 7.5c) when using the Eulerian/Lagrangian scheme. The initial velocity vectors (after subtraction of the convection speed) are plotted in Figure 7.6a).

The vortex at the end of the channel is represented in Figures 7.6b) and 7.6c) for the Eulerian solution and the Eulerian/Lagrangian solution, respectively.

A strong diffusion is observed in the Eulerian solution, whereas the addition of the Lagrangian correction is successful in preserving the vortex structure and intensity.

This is substantiated by looking at the velocity (minus convection speed) and pressure profiles across the vortex (after traveling 53 core radii) represented in Figures 7.7 and 7.8. Whereas the diffusion of the Eulerian solution is obvious, the vortex core radius and vortex strength remain almost unchanged when using the Lagrangian correction procedure.

Figure 7.9 shows a measure of the numerical diffusion as the decay rate of the pressure coefficient Cp_v at the center of the vortex

$$Cp_v = \frac{p_{min} - p_\infty}{\frac{1}{2}\rho_\infty(\frac{\Gamma}{a})^2}, \tag{7.6}$$

where p_{min} is the pressure at the center of the vortex and p_∞ is the pressure of the background potential flow. The exact solution corresponds to a constant Cp_v coefficient. For the same grid density, the standard Eulerian scheme leads to a strong diffusion of the vortex during its convection whereas a substantial improvement in the vortex preservation is obtained when using the combined Eulerian/Lagrangian scheme. The ragged aspect of the Cp_v curve (computed as a cell-average value) when using the Lagrangian procedure is due to the vortex passing over the Eulerian grid. The difference between the Eulerian/Lagrangian curve and the exact curve is due to discretization errors when representing the vorticity and entropy of the vortex by a finite amount of markers placed at the cell centers.

If the accuracy of the solution is measured by monitoring the pressure losses at the center of the vortex, the Eulerian/Lagrangian scheme leads to a solution approximately 4 times more accurate than the Eulerian solution alone after the vortex has traveled 53 core radii. In order to gain a factor 4 in accuracy a second-order accurate scheme such as the Ni-Lax-Wendroff scheme would require a grid twice as fine in every direction. This would lead to an increase in CPU of a factor 16 (a factor 2 for every spatial direction times a factor 2 for the corresponding decrease in time-step). In comparison, the present Eulerian/Lagrangian scheme requires only $\sim 30\%$ increase in CPU over the basic Eulerian solution.

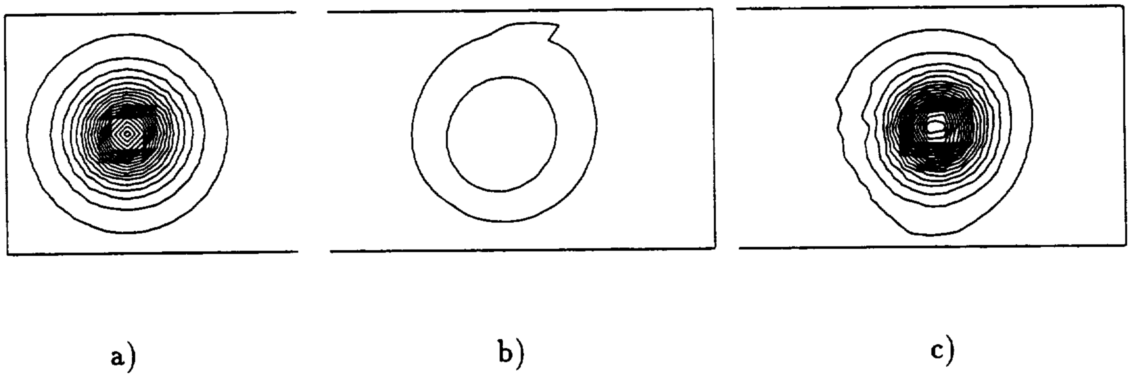


Figure 7.3: Pressure contours at channel mid-height, a) initial, b) final with Eulerian scheme and c) final with Eulerian/Lagrangian scheme (increment = 0.0025).

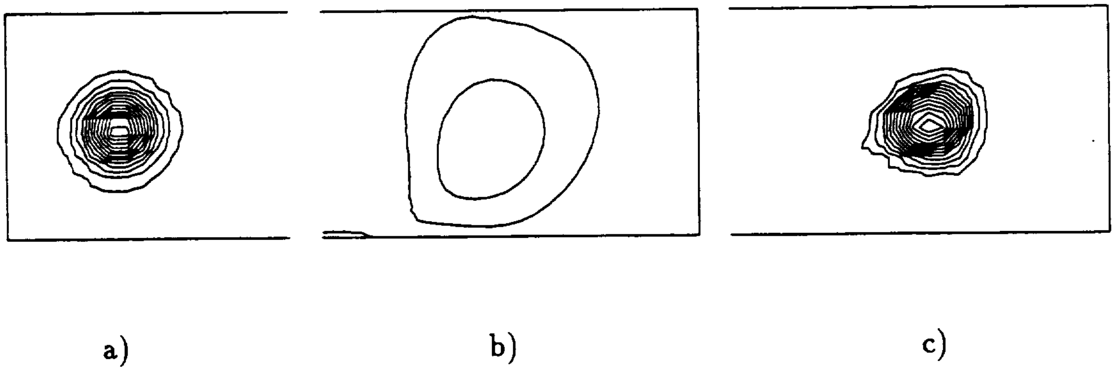


Figure 7.4: Vorticity contours at channel mid-height, a) initial, b) final with Eulerian scheme and c) final with Eulerian/Lagrangian scheme (increment = 2.0).

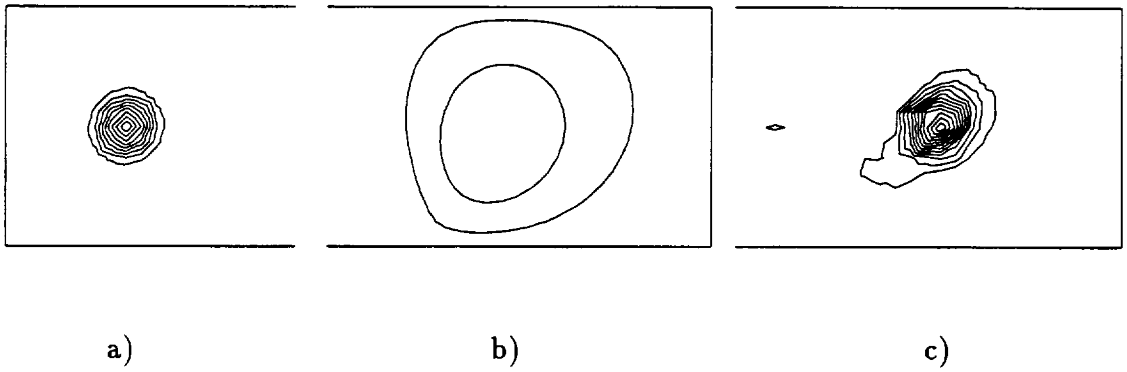


Figure 7.5: Contours in S (entropy related function) at channel mid-height, a) initial, b) final with Eulerian scheme and c) final with Eulerian/Lagrangian scheme (increment = 0.003).

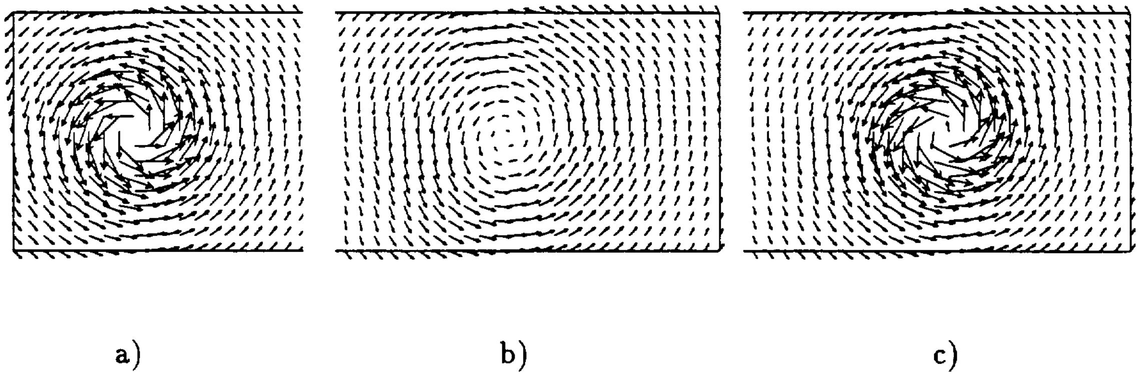


Figure 7.6: Velocity vectors at channel mid-height, a) initial, b) final with Eulerian scheme and c) final with Eulerian/Lagrangian scheme.

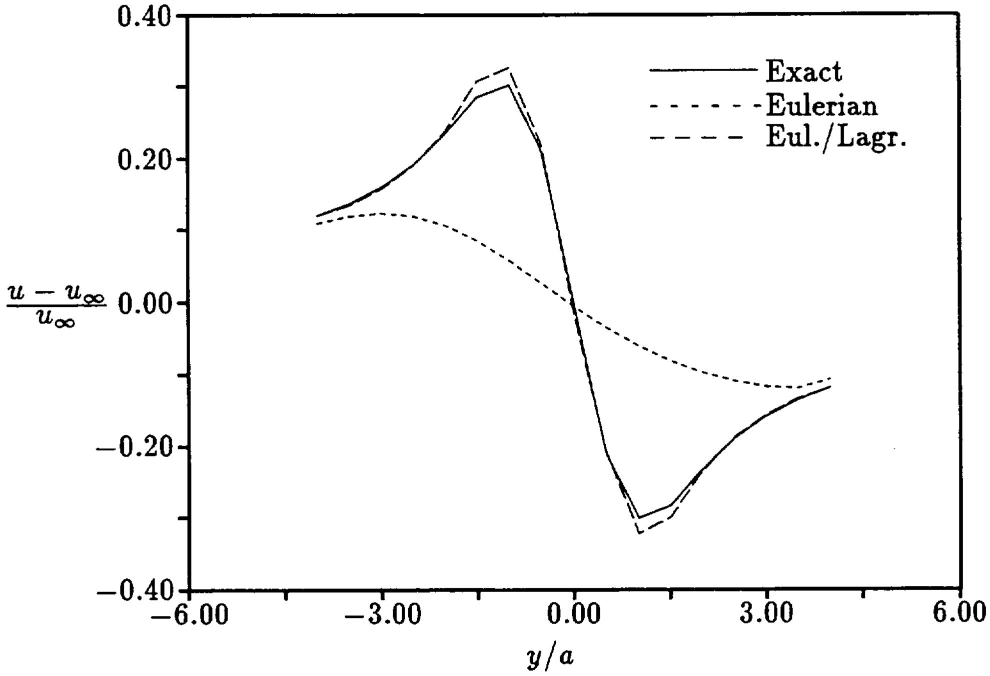


Figure 7.7: Velocity profiles across vortex for exact, Eulerian and Eulerian/Lagrangian solutions.

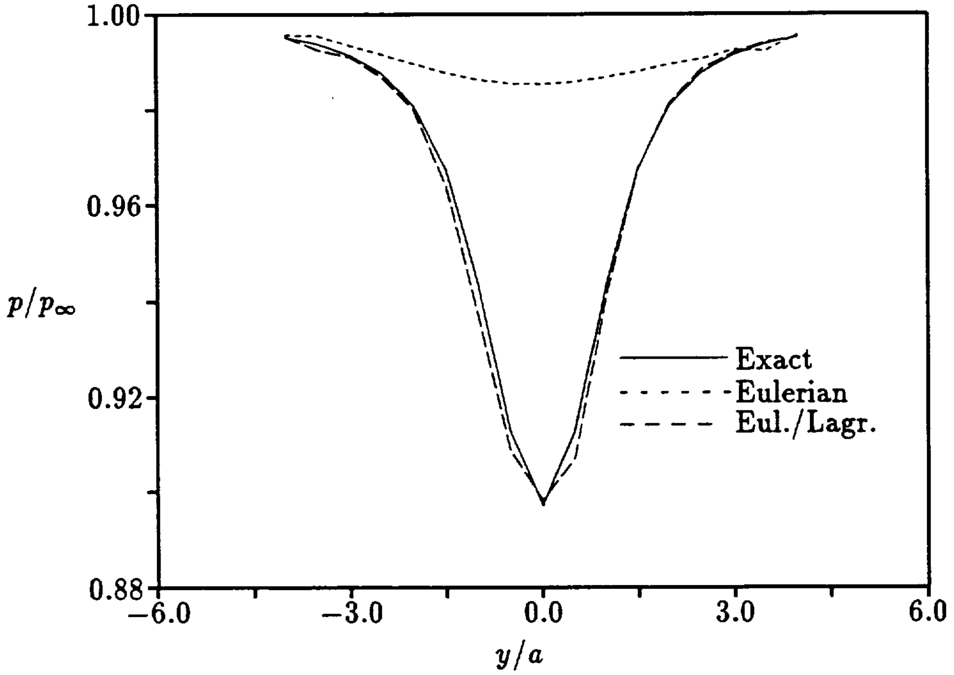


Figure 7.8: Pressure profiles across vortex for exact, Eulerian and Eulerian/Lagrangian solutions.

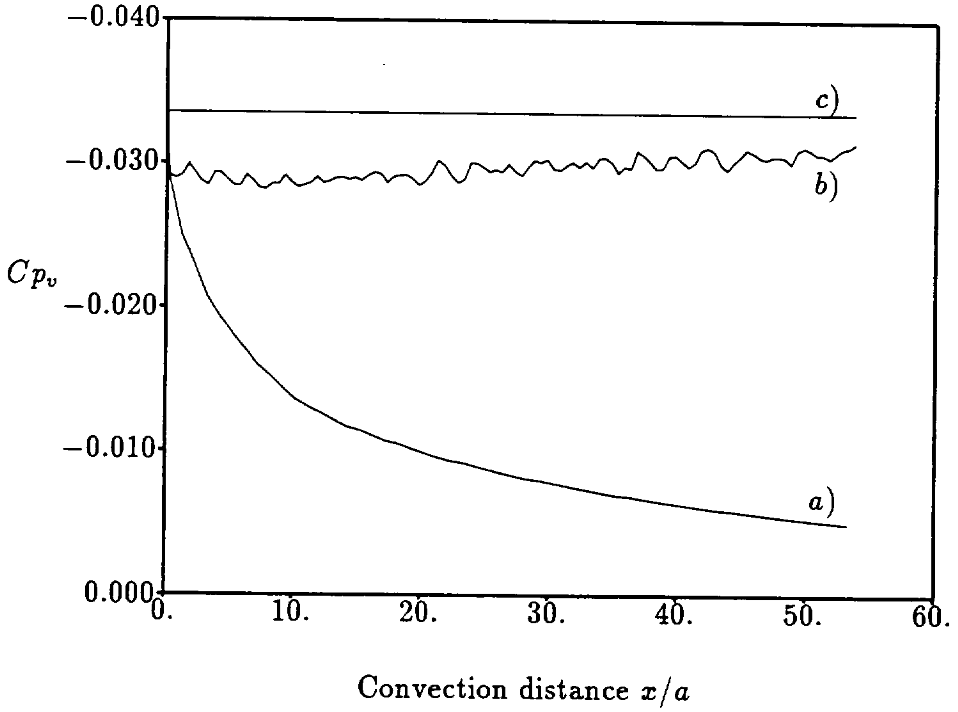


Figure 7.9: Pressure coefficient as a function of the convection distance along the channel, a) Eulerian solution, b) Eulerian/Lagrangian solution and c) exact solution.

Chapter 8

Preservation of a Turbulent Inlet Velocity Profile in a Pipe

The preservation of a turbulent inlet velocity profile in a straight pipe is analogous to the preservation of the Lamb vortex in a straight channel described in Chapter 7. Both the Lamb vortex and the turbulent velocity profile are characteristics of a viscous flow. The exact inviscid solution of the flow problem states that the velocity profile is not a function of the convection distance. As in the case of the Lamb vortex, the interest here is to see how these viscous flow features represented though the velocity profile are affected by the numerical diffusion of the Eulerian scheme. The turbulent velocity profile is provided as the inlet boundary condition of the calculation. As the viscous effects are not taken into account in the Euler equations, each flow particle should conserve its speed as it convects downstream. When using an Eulerian solver, the changes in the velocity profile are the result of numerical diffusion. The Lagrangian correction technique is then applied in order to recover the exact velocity profile at any station along the pipe. This case bears similarity with the convection of a fully developed turbulent flow in that the velocity profile is not a function of the convection distance.

This test case is a preamble to the computation of secondary flows in bent pipes of Chapter 11 where the source for the secondary flow creation is introduced as a non-uniform inlet stagnation pressure (obtained through a non-uniform inlet velocity profile), and where the Euler equations are used to predict the subsequent secondary flow

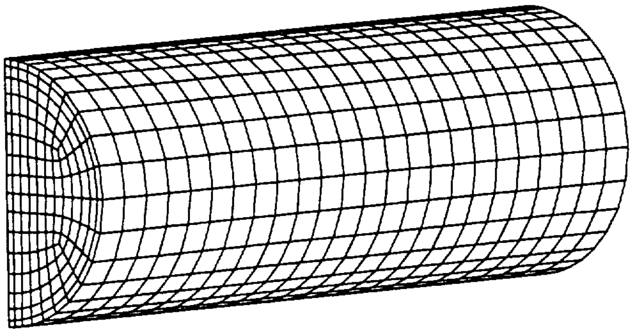


Figure 8.1: Straight circular pipe computational grid (161x25 nodes).

development.

Since the flow is incompressible and steady, the Euler equations are modified by the artificial compressibility method as described in Section 2.2. The Eulerian and the Eulerian/Lagrangian solutions are computed using the same grid, shown in Figure 8.1, with 161 nodes on an axial cross-section and 25 cross-sections equally spaced along the pipe length. Since the flow is symmetric, the computations are performed on only half of the pipe. The ratio of pipe length to pipe radius is 5.52. The inlet cross-section shows the unstructured grid with an O-type grid near the pipe wall and an H-type grid near the center of the pipe. This type of combined grid is chosen for the pipe geometries over the standard O-grid which presents a singularity at the pipe center.

In the case of swirling flow in pipes or secondary flow calculations, the Lagrangian solution technique, using an upstream integration of streamlines and markers placed at the center of each cell, is preferred over the downstream integration technique, since the vorticity is diffused in the whole domain.

The streamlines are also traced backward in the present test case, with two integration steps per cell and are recomputed each 25 iterations of the Eulerian solver. The correction procedure is applied only once every Eulerian iteration and the corrections are multiplied by a factor 1/4 in order to limit the perturbations imposed on the Eulerian solution.

A fully developed turbulent velocity profile is assumed at the pipe inlet cross-section. The inlet velocity profile corresponds to a Reynolds number of

$$Re = \frac{\overline{u}d}{\nu} = 2.48 \times 10^5 \tag{8.1}$$

based on a mass-flow averaged velocity $\overline{u}$ of 2.31 m/s (chosen here as the reference velocity) and on the pipe diameter d of 0.1023 m. The kinematic viscosity ν is 0.952×10^{-6} m$^2/s$. The pipe radius R is chosen as the reference length. The inlet velocity profile is determined by the 'universal velocity distribution law' for smooth pipes and very large Reynolds numbers given in [62]

$$u^+ = \frac{u}{u^*} = 5.75 \log_{10}(y^+) + 5.5, \tag{8.2}$$

where u^* is the friction velocity defined as

$$u^* = \left(\frac{\tau_w}{\rho}\right)^{\frac{1}{2}}, \tag{8.3}$$

and y^+ is a Reynolds number based on u^* as

$$y^+ = \frac{yu^*}{\nu}, \tag{8.4}$$

where y is the distance from the wall of the pipe.

Since the flow computation is inviscid, the velocity at the wall must take a finite value called the 'slip velocity'. Imposing a zero velocity value at the wall would tend to produce non-physical reverse flow under any small positive pressure gradient perturbation. In order to define the slip velocity, Prandtl's universal law of friction for smooth pipes given in [62] is used to provide a relationship between the coefficient of friction λ

$$\lambda = 8\left(\frac{u^*}{u}\right)^2 \tag{8.5}$$

and the Reynolds number as

$$\frac{1}{\sqrt{\lambda}} = 2.0 \log_{10}(Re\sqrt{\lambda}) - 0.8, \tag{8.6}$$

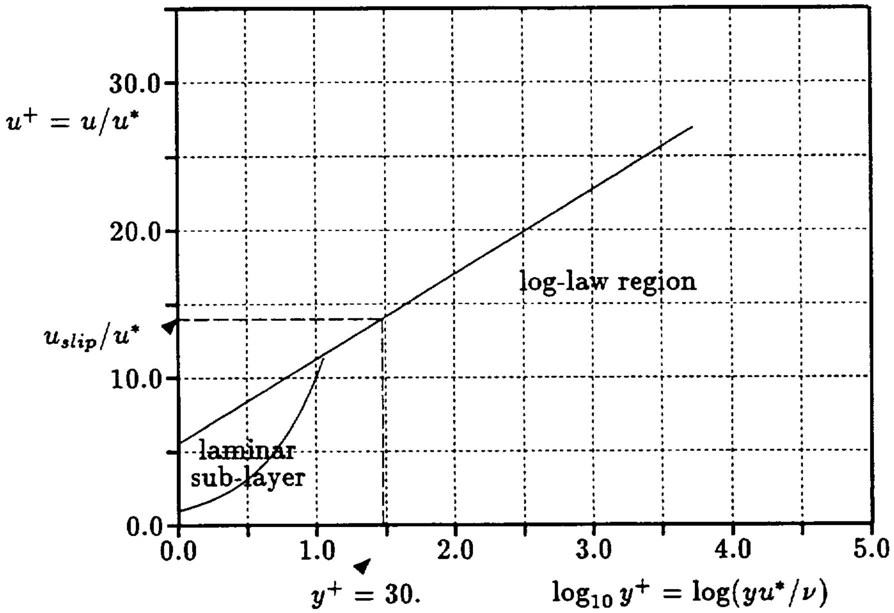


Figure 8.2: Universal velocity distribution law for smooth pipes.

For a Reynolds number of 2.48×10^5, this relation gives $\lambda = 0.01503$. The shear stress at the wall τ_w and the friction velocity u^* are now found by

$$\tau_w = \frac{\lambda}{8}\rho\bar{u}^2, \qquad (8.7)$$

and Equation (8.3). The slip velocity is chosen as the velocity just outside of the viscous sub-layer at a y^+ value of 30.

Using the values for y^+ and u^*, the slip velocity is found from the 'universal velocity distribution law' given in Equation (8.2). The resulting slip velocity u_{slip} is equal to 60.7% of the flow averaged velocity at a distance from the wall of 0.0286% of the pipe radius. The universal velocity distribution law for smooth pipes is represented in Figure 8.2 as well as the slip velocity location.

In Figure 8.3, the velocity profiles on the exit cross-section are plotted as a function of the normalized radius r/R for the exact solution (inlet velocity profile), the Eulerian solution and the Eulerian/Lagrangian solution. Due to the numerical diffusion

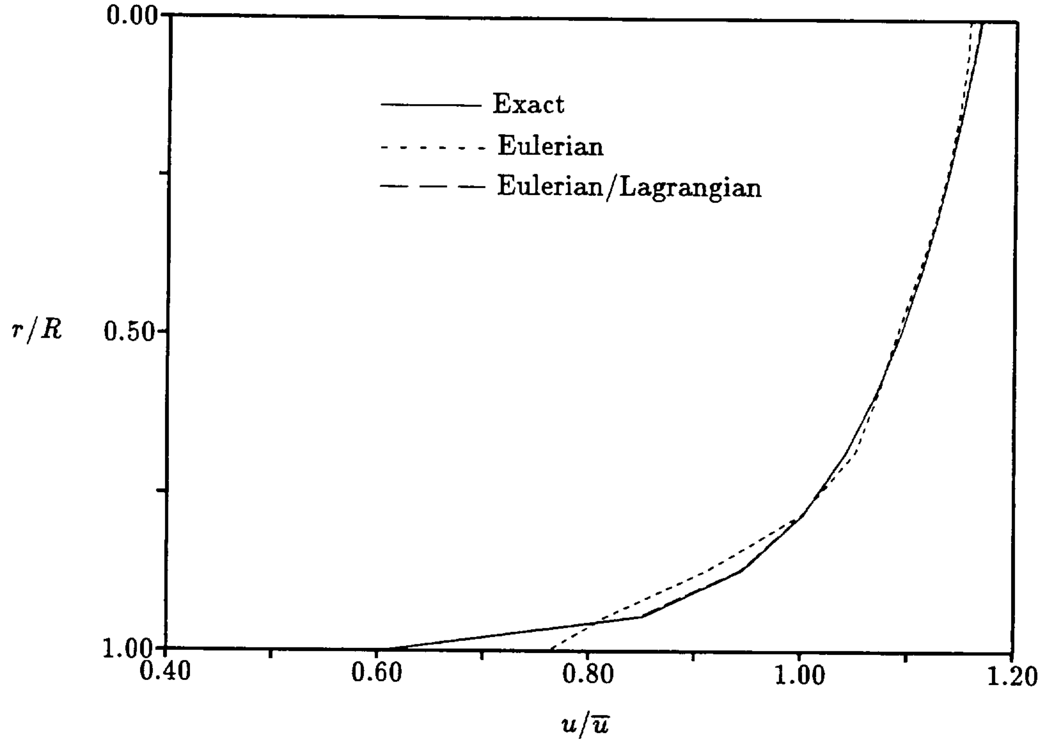


Figure 8.3: Velocity profiles at outflow cross-section.

of the scheme, the sharp gradients near the pipe wall are smoothed out in the Eulerian solution. With the Lagrangian correction scheme, this effect is canceled and the Eulerian/Lagrangian solution and the exact solution are within drawing accuracy.

As for the unsteady flow case, the Lagrangian correction could be applied during the pseudo-unsteady convergence of the Eulerian solution. However, for the steady flow cases, it is more effective in terms of CPU reduction to apply the Lagrangian correction once the Eulerian solution has converged (or nearly converged) since the convergence of the Eulerian solution takes more iterations than the Lagrangian correction convergence.

For this particular test case, the combined Eulerian/Lagrangian solution requires only $\sim$ 60 iterations to converge (from the converged Eulerian solution), whereas the Eulerian solution requires $\sim$ 1000 iterations to converge (from a constant flow initial solution) to a maximum residual of 1×10^{-5}. The Eulerian solution, based on the Lax-Wendroff scheme and the artificial compressibility concept, takes $\sim$ 274×10^{-6}

seconds/iteration/grid node. In comparison, the combined Eulerian/Lagrangian scheme takes ~ 8.0 seconds/iteration for this particular frequency of trajectory integration and marker number. The CPU increase due to the Lagrangian correction for this test case is $\sim 30\%$ of the basic CPU required for the Eulerian solution.

Chapter 9

Swirling Flow in a Pipe

Another preamble case to the secondary flow computation in bent pipes of Chapter 11 is the analysis of a swirling flow in a straight circular pipe. A swirling flow in a straight pipe superimposed on a uniform axial velocity is chosen as a representation of the convection of the secondary flow in the straight section of pipe downstream of the bend exit cross-section.

The model for the swirling flow and the Eulerian and Eulerian/Lagrangian solutions are presented in Section 9.1. In Section 9.2, the strength of the swirling flow is increased and the vorticity field is shown to concentrate by the phenomenon of vorticity gradient augmentation to a point where the vorticity field becomes inaccurately represented when using the fixed Eulerian or Lagrangian spatial discretization. As a result, the convergence of the combined Eulerian/Lagrangian solution is affected. The sources for the vorticity concentration are identified in Section 9.2.1 and a solution to the problem is proposed in Section 9.2.2 as the introduction of a pseudo-diffusion term in the Helmholtz equation.

9.1 Swirling flow model and solution

As in the case of the secondary flow, the swirling flow of this example is composed of two counter-rotating vortices and, since the solution is symmetrical, it can be performed

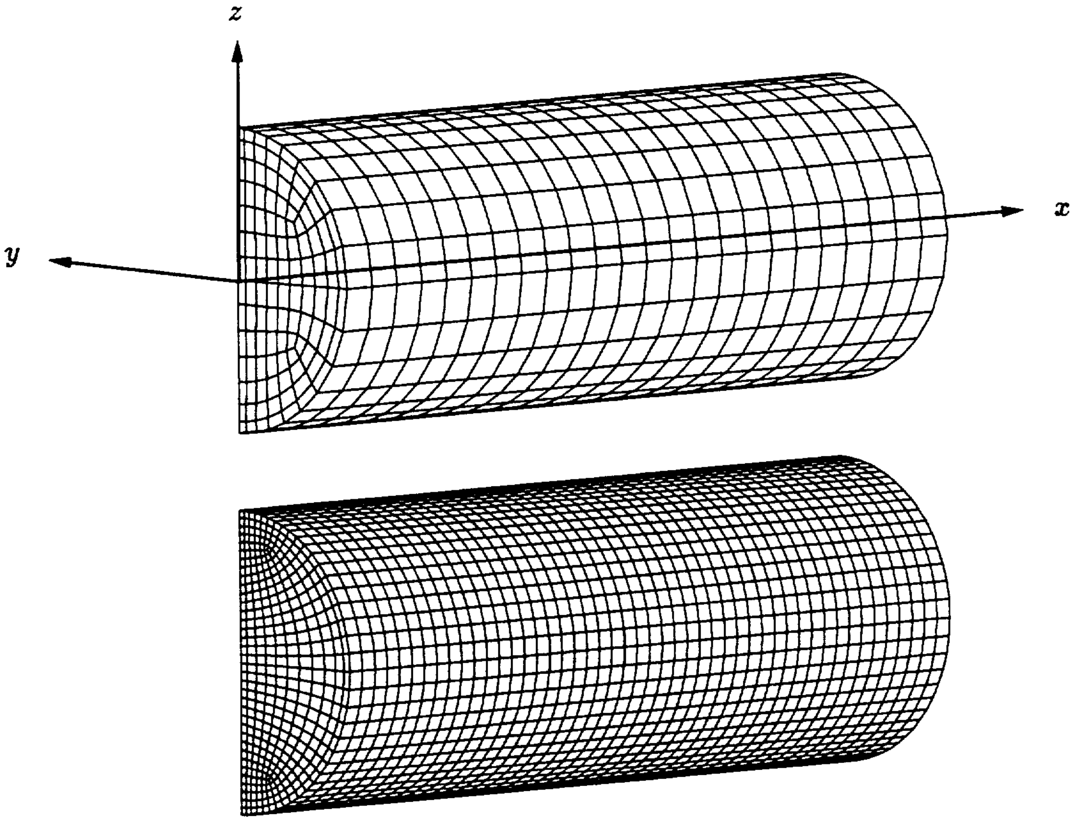


Figure 9.1: Straight circular pipe computational grids: coarse grid with 125 × 25 nodes and fine grid with 384 × 49 nodes.

on a half-pipe geometry. The ratio of pipe length to pipe radius is $L/R = 5.52$.

The calculations are performed on a coarse grid with 25 cross-sections equally spaced along the pipe length and 125 nodes on an axial cross-section and on a fine grid with 49×384 nodes. Both grids are shown in Figure 9.1. The present model for the inlet velocity profile is characterized by high cross-flow velocities near the pipe walls and the half-pipe symmetry surface according to the results found in bent pipes in Chapter 11. The inlet cross-flow velocities are found from the Poisson equation relating the streamfunction Ψ to the axial vorticity ω_x

$$\nabla^2 \Psi = -\omega_x, \tag{9.1}$$

where the streamfunction is defined by

$$\Psi = -Cy(1 - \frac{y^2 + z^2}{R^2}). \tag{9.2}$$

R is the pipe radius and C is a constant characterizing the strength of the swirling flow. The cross-flow velocities v and w are found by

$$v = \frac{\partial \Psi}{\partial z} = \frac{2Cyz}{R^2}, \qquad w = -\frac{\partial \Psi}{\partial y} = C\left(1 - \frac{3y^2 + z^2}{R^2}\right). \tag{9.3}$$

The resulting axial vorticity distribution is linear with respect to y

$$\omega_x = \frac{-8Cy}{R^2}. \tag{9.4}$$

The first chosen cross-flow strength is such that the maximum inlet cross-flow velocity is 40% of the uniform convection velocity, i.e. $C = 0.2$. The inlet cross-flow velocities and axial vorticity distribution of the coarse grid solution are shown in Figure 9.2. Since the vorticity is obtained as a cell-averaged quantity, the minimum and maximum inlet axial vorticity levels are different on the coarse and the fine grids. Nevertheless, the differences in minimum and maximum axial vorticity levels between the two grids are less than 3% of the maximum axial vorticity.

The Eulerian and the Eulerian/Lagrangian calculations are performed on both grids. In the combined scheme, the markers are initially placed at the center of each cell and

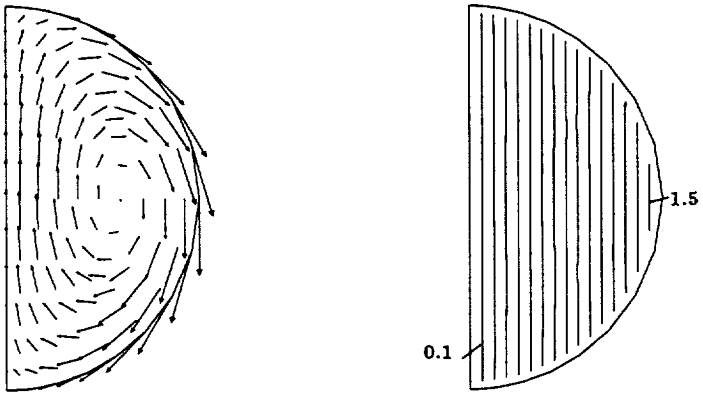


Figure 9.2: Case $C = 0.2$: Cross-flow velocity vectors and axial vorticity on inlet cross-section for coarse grid calculations.

the streamlines are integrated upstream with approximately two steps per cell. The streamlines are recomputed every 50 iterations of the Eulerian solver. The correction step is applied only once at each Eulerian iteration and the corrections are multiplied by a factor 1/4 in order to limit the perturbations to the solution during the pseudo-time integration process.

Because of the induced velocities of one vortex on the other and the presence of the pipe wall, the center of each vortex is moving in an helicoidal pattern resulting in an alteration of the initially linear vorticity distribution. Figure 9.3 presents the axial vorticity contours on three cross-sections along the pipe for the Eulerian solution and the Eulerian/Lagrangian solution on the coarse grid and the fine grid. As the distance along the pipe increases, the constant axial vorticity contours undergo a rotation (stations $x/L = 0.08$ and $x/L = 0.52$). At the symmetry surface, the axial vorticity must be zero since $\partial w/\partial y = 0$ and $v = 0$ in the symmetry surface. This condition forces the axial vorticity contours to concentrate near the symmetry surface (stations $x/L = 0.52$ and $x/L = 0.98$). The effect of numerical diffusion near the symmetry surface spreads the gradient of vorticity on approximately two cells in the Eulerian solutions. With the Lagrangian solution, the gradient of vorticity is concentrated on approximately only one cell and does not appear in the plots since the solution for the vorticity is

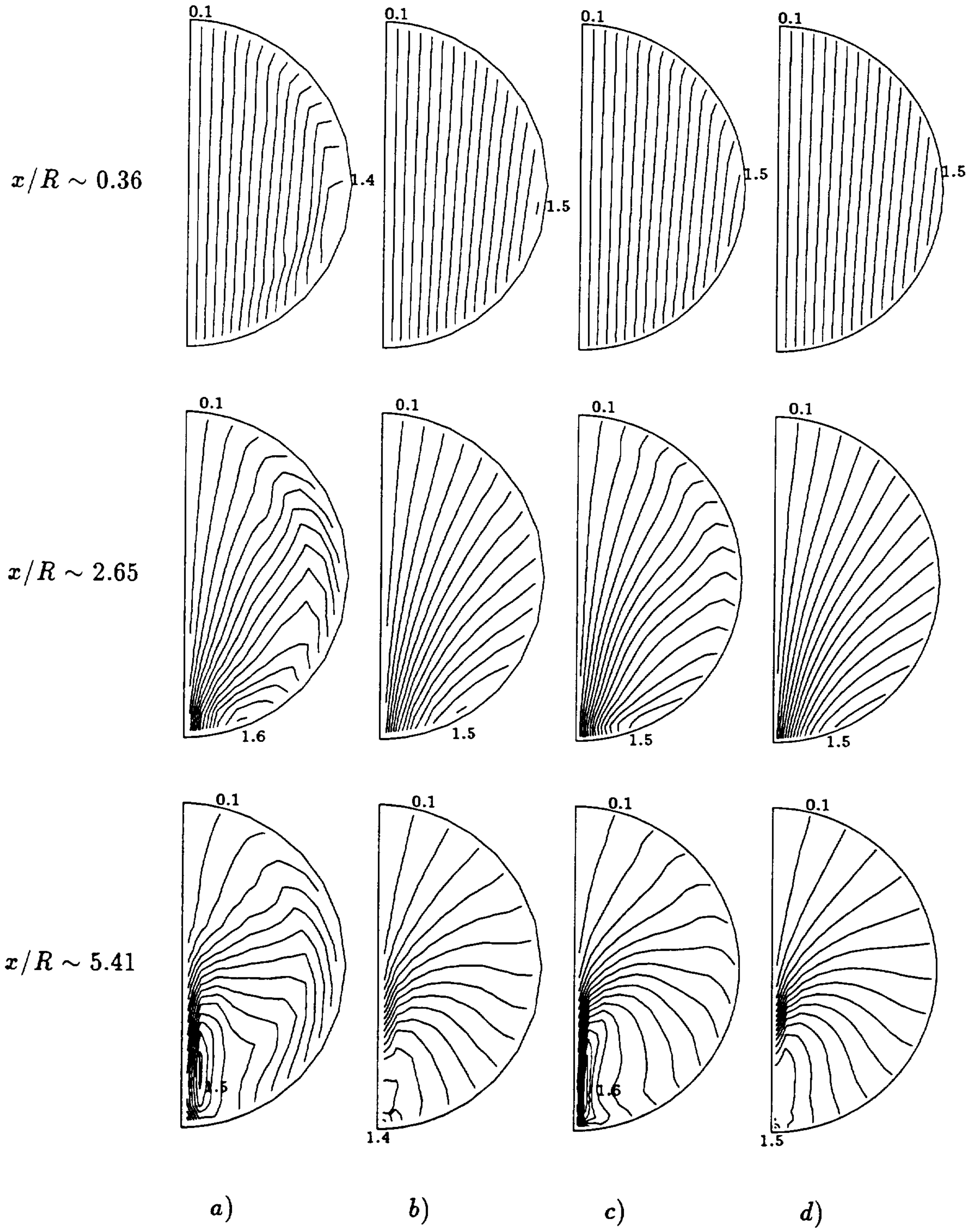


Figure 9.3: Case $C = 0.2$: Axial vorticity contours at diverse cross-sections along the pipe, a) coarse grid Eulerian solution, b) coarse grid Eulerian/Lagrangian solution, c) fine grid Eulerian solution, d) fine grid Eulerian/Lagrangian solution (increment=0.1).

drawn only until the last cell center near the symmetry. The effect of the numerical diffusion is clearly seen in the coarse grid Eulerian solution as a spurious curving of the axial vorticity contours near the pipe wall (inducing smaller velocities on the pipe wall and therefore smaller circulation around the pipe). Using the Lagrangian correction, the axial vorticity contours are in a very good agreement with the Eulerian solution on the fine grid, especially near the pipe wall. The fine grid Eulerian solution still exhibits a small numerical diffusion effect near the pipe wall. When using the Lagrangian correction on the fine grid, this effect is eliminated.

The circulation Γ around a closed curve is an integral value used to quantify the numerical diffusion as a function of the distance along the pipe. In a barotropic flow field where viscous effects are non-existent or can be neglected and if the forces acting on the fluid are conservative, Kelvin's theorem states that the circulation around an arbitrary closed curve moving with the fluid should remain constant. Using the material derivative, this is expressed as

$$\frac{D\Gamma}{Dt} = 0. \tag{9.5}$$

The convection of a material curve is obtained by setting 'convective' markers in a closed curve pattern at some location in the flow field once the solution has reached its steady-state. The material curve is deformed as each marker convects downstream with the local flow. Since these markers are used only for particle tracing purposes, they do not induce any correction in the Eulerian flow field. The tracing of the 'convective' markers is similar to the technique used for the 'corrective' markers trajectories integration. However, pseudo-time steps can not be used since the time-steps used for the integration of the streamlines have to be identical for every convective marker. The circulation is found by integration around the material curve as

$$\Gamma = \oint \vec{v} \cdot d\vec{r}, \tag{9.6}$$

where the integral is obtained by trapezoidal integration over the total number of convective markers. The material curve can be traced downstream or upstream and additional convective markers are added to the material curve if the distance between two

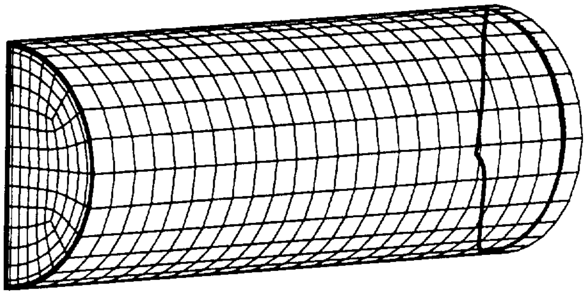


Figure 9.4: Initial and final circulation contours.

consecutive markers becomes too large due to the divergence of the streamlines.

The chosen closed curve is initially placed around the inlet cross-section and the convective markers are traced downstream. Figure 9.4 shows the position of the initial and final locations of the material curve. The curve is initially located near the pipe wall where the largest numerical diffusion occurs. Figure 9.5 shows the circulation around the closed curve as a function of the convection distance along the pipe (the average location over all the convective markers). The Eulerian solution on the coarse grid exhibits the largest circulation change. By using the combined Eulerian/Lagrangian scheme on the coarse grid, the changes in circulation are reduced by a factor $\sim$ 4 near the pipe exit. The circulations for the fine grid Eulerian solution and the fine grid Eulerian/Lagrangian solution are comparable since the correction is much smaller than in the case of the coarse grid solution. In the case of the fine grid solutions, the amount of diffusion is too small to judge the effectiveness of the Lagrangian correction (the comparison would have to be performed on a longer pipe in order to get a more diffused Eulerian solution).

The coarse grid Euler solution requires $\sim$ 2000 iterations to converge (from a constant flow initial condition) to a maximum residual of $\sim 1 \times 10^{-5}$. The Eulerian/Lagrangian solution takes only $\sim$ 275 iterations to converge (starting from the converged Eulerian solution). The computations are performed on a Stardent GS-2000 in vector mode. The

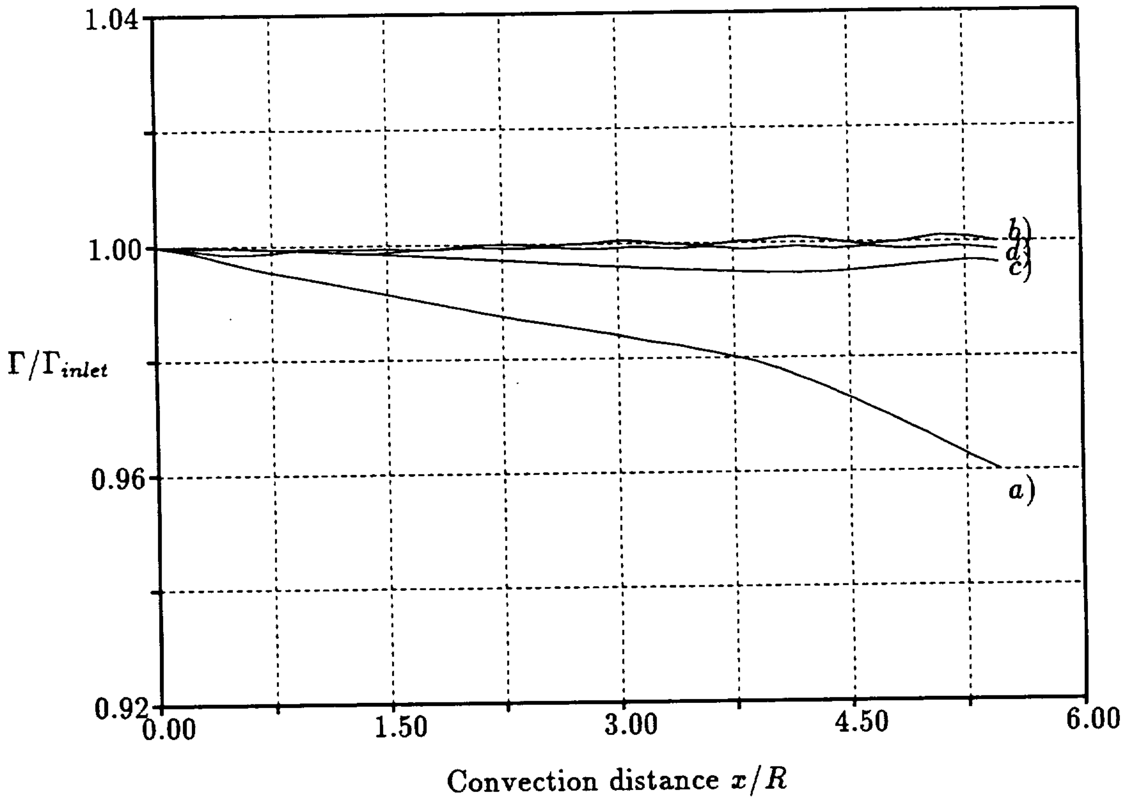


Figure 9.5: Case $C = 0.2$: Circulation on a closed curve as a function of the distance along the pipe, a) coarse grid Eulerian solution, b) coarse grid Eulerian/Lagrangian solution, c)fine grid Eulerian solution, d) fine grid Eulerian/Lagrangian solution.

combined scheme takes 2.8 seconds/iteration/node on average (for the marker number and frequency of streamline integration described above). Thus, in the coarse grid case, the increase in CPU due to the introduction of the Lagrangian correction is $\sim$ 33% of the basic Eulerian solution. In comparison, the fine grid Eulerian solution requires $\sim$ 16 times more CPU than the basic coarse grid Eulerian calculation (starting from a uniform flow solution).

9.2 Vorticity gradient augmentation

As the strength of the swirling flow increases, the gradients of vorticity shown in Figure 9.3 near the exit of the pipe become stronger. The evolution of the scalar field in a flow with high vorticity concentration has been previously studied by Knio in [38].

The phenomenon of vorticity gradient augmentation is intensified when the Lagrangian correction is applied. Indeed, due to numerical diffusion, the Eulerian solution tends to smooth out strong gradients, which is not the case when using the Lagrangian correction.

The intensification of the vorticity gradient is illustrated in Figure 9.6 where the strength of the swirling flow has been increased by a factor 1.5 over the previous test case (i.e. $C = 0.3$). The axial vorticity contours are plotted at stations along the pipe in Figures 9.6a) and 9.6b) for the coarse grid Eulerian and Eulerian/Lagrangian solutions and in Figures 9.6c) and 9.6d) for the fine grid Eulerian and Eulerian/Lagrangian solutions, respectively. When using the Lagrangian correction on the coarse grid, the gradient of vorticity at the exit of the pipe is increased compared to the Eulerian solution alone. A strong vorticity gradient region of a few cells width is created in the Eulerian/Lagrangian solution separating a high vorticity region from a low vorticity region. The same behavior is seen in the fine grid solutions, but the gradient of vorticity is more intense and spread over one cell only in the Eulerian/Lagrangian solution.

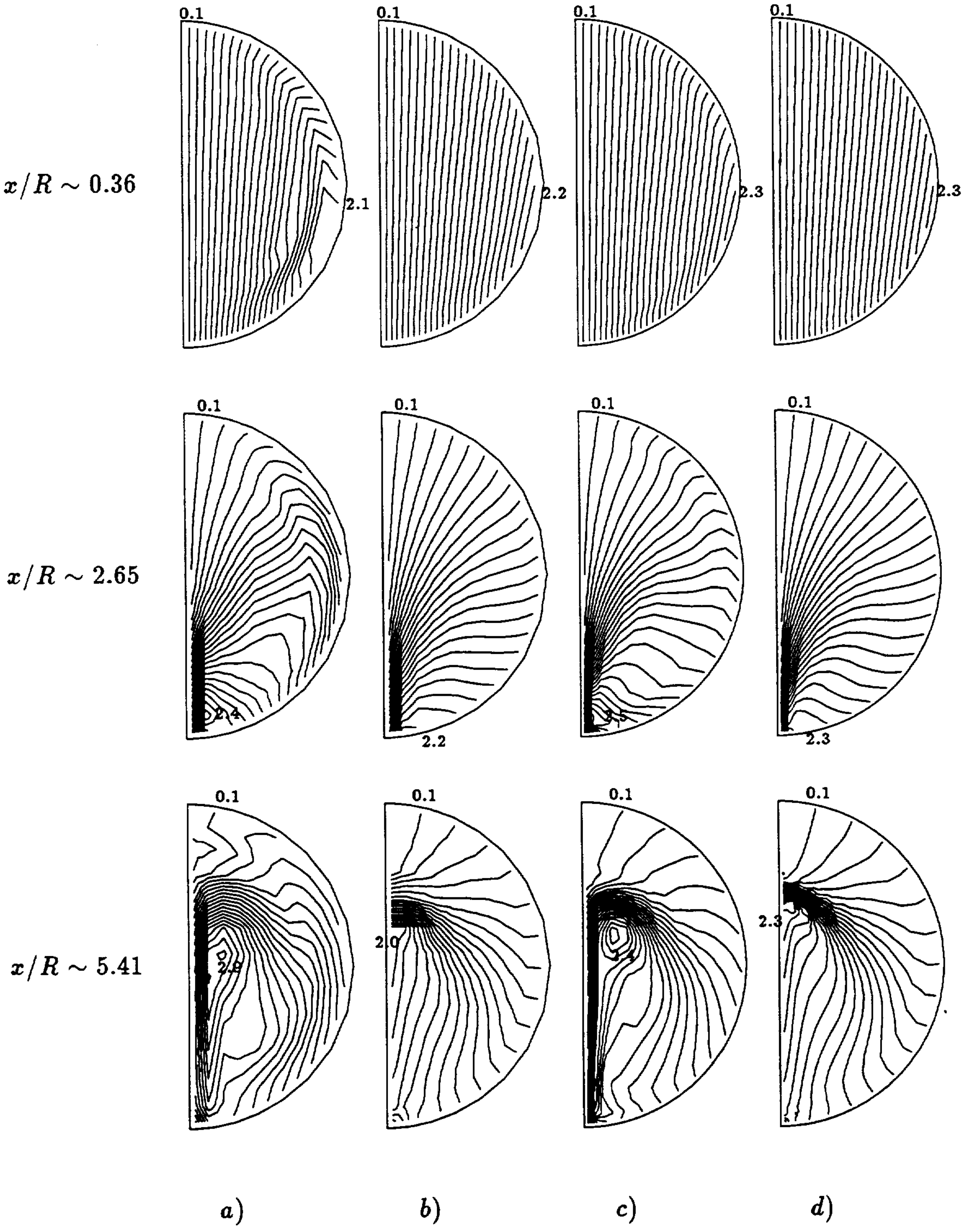


Figure 9.6: Case $C = 0.3$: Axial vorticity contours at diverse cross-sections along the pipe, a) Eulerian solution on coarse grid, b) Eulerian/Lagrangian solution on coarse grid, c) Eulerian solution on fine grid, d) Eulerian/Lagrangian solution on fine grid (increment=0.1).

In Section 9.2.2, the strength of the swirling flow is further increased (to $C = 0.6$) and the width of the vorticity gradient region decreases. The vorticity field becomes poorly represented (either on the fixed size grid in the Eulerian solution or by markers located at cell centers in the Lagrangian scheme). In turn, the poor sampling rate of the vorticity field leads to inaccurate corrections of the Eulerian solution and affects the convergence of the combined Eulerian/Lagrangian scheme.

In Section 9.2.1, the strain field and the vorticity field are first identified as the sources for the vorticity gradient augmentation. In Section 9.2.2, a solution to the problem is proposed as the introduction of a pseudo-diffusion term in the Helmholtz equation.

9.2.1 Sources for vorticity concentration

In Figures 9.7a) and 9.7b), the convergence of three streamlines from inlet to exit is shown for the coarse grid Eulerian and Eulerian/Lagrangian solutions in the case of the swirling flow strength corresponding to $C = 0.3$. The streamlines are chosen such as to end up in the strong vorticity gradient region at the exit of the pipe. Since for this test case the source-terms for the vorticity are small (as will be shown later), the vorticity carried by the material particles along the streamlines is essentially constant and equal to the inlet vorticity at the inlet location of the markers. Because of the merging of the streamlines, the inlet vorticity assigned to each of the three markers differs by a large amount. When the streamlines merge along the pipe, a strong gradient in vorticity is created. When using the Eulerian scheme, the vorticity along the streamlines does not remain constant because of numerical diffusion and the creation of the strong vorticity gradient is inhibited. When the Lagrangian correction is used, the diffusion is reduced, and consequently the solution presents stronger gradients of vorticity. This effect is accentuated when using the Lagrangian correction since, the three streamlines show a stronger convergence in the Eulerian/Lagrangian solution than in the Eulerian solution alone due to the stronger swirl effect.

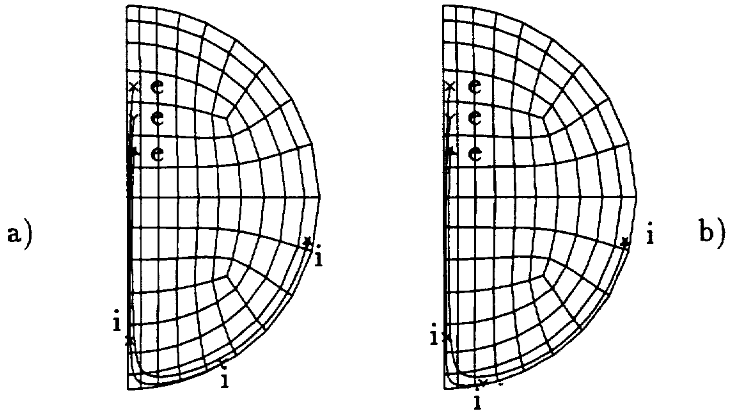


Figure 9.7: Case $C = 0.3$: Frontal view of 3 streamlines drawn from inlet to exit of the pipe, a) Eulerian solution, b) Eulerian/Lagrangian solution (i=inlet, e=exit).

The creation of strong vorticity gradients along the pipe can be explained from the action of the strain and vorticity field by using the Helmholtz equation giving the convective change in vorticity for an incompressible flow as

$$\frac{D\vec{\omega}}{Dt} = (\vec{\omega}\cdot\nabla)\vec{v}. \tag{9.7}$$

This equation states that the evolution of the vorticity attached to a particle convecting in an incompressible flow is governed by the generation of tilting and stretching source-terms along the particle trajectory. If the right-hand side terms (source-terms for the vorticity) are small, each component of $\vec{\omega}$ will approximately behave as a non-diffusive quantity as

$$\frac{D\vec{\omega}}{Dt} \simeq \vec{0}. \tag{9.8}$$

By taking the gradient of Equation (9.8), relations are found which describe the behavior of the vorticity gradient for a convecting particle in a steady flow.

$$\frac{D(\nabla\omega_x)}{Dt} = (\vec{v}\cdot\nabla)\nabla\omega_x = \nabla\underbrace{\frac{D\omega_x}{Dt}}_{\simeq 0} - \frac{\partial u}{\partial\vec{r}}\frac{\partial\omega_x}{\partial x} - \frac{\partial v}{\partial\vec{r}}\frac{\partial\omega_x}{\partial y} - \frac{\partial w}{\partial\vec{r}}\frac{\partial\omega_x}{\partial z}, \tag{9.9}$$

$$\frac{D(\nabla\omega_y)}{Dt} = (\vec{v}\cdot\nabla)\nabla\omega_y = \nabla\underbrace{\frac{D\omega_y}{Dt}}_{\simeq 0} - \frac{\partial u}{\partial\vec{r}}\frac{\partial\omega_y}{\partial x} - \frac{\partial v}{\partial\vec{r}}\frac{\partial\omega_y}{\partial y} - \frac{\partial w}{\partial\vec{r}}\frac{\partial\omega_y}{\partial z}, \tag{9.10}$$

$$\frac{D(\nabla\omega_z)}{Dt} = (\vec{v}\cdot\nabla)\nabla\omega_z = \nabla\underbrace{\frac{D\omega_z}{Dt}}_{\simeq 0} - \frac{\partial u}{\partial\vec{r}}\frac{\partial\omega_z}{\partial x} - \frac{\partial v}{\partial\vec{r}}\frac{\partial\omega_z}{\partial y} - \frac{\partial w}{\partial\vec{r}}\frac{\partial\omega_z}{\partial z}. \tag{9.11}$$

Or

$$\frac{D(\nabla\omega_x)}{Dt} = -(\nabla\omega_x\cdot\nabla)\vec{v} - \nabla\omega_x\times\vec{\omega}, \tag{9.12}$$

$$\frac{D(\nabla\omega_y)}{Dt} = -(\nabla\omega_y\cdot\nabla)\vec{v} - \nabla\omega_y\times\vec{\omega}, \tag{9.13}$$

$$\frac{D(\nabla\omega_z)}{Dt} = -(\nabla\omega_z\cdot\nabla)\vec{v} - \nabla\omega_z\times\vec{\omega}. \tag{9.14}$$

Hence, if a scalar quantity (here any component of vorticity) remains approximately constant along a streamline, its gradient is governed along a streamline by both the local strain field $\nabla\vec{v}$ and the local vorticity (rotation) field $\vec{\omega}$.

In the case of a swirling flow through a straight pipe, the main component of vorticity is in the axial direction (x). The axial vorticity and the gradient of axial vorticity along a streamline can be written as a function of convection time t. From Equation (9.7)

$$\omega_x(t) - \omega_x(0) = \int_0^t ((\vec{\omega}\cdot\nabla)u)dt = S_{\omega_x}, \tag{9.15}$$

and from Equation 9.12

$$\nabla\omega_x(t) - \nabla\omega_x(0) = \int_0^t (-(\nabla\omega_x\cdot\nabla)\vec{v} - \nabla\omega_x\times\vec{\omega})dt = S_{\nabla\omega_x}. \tag{9.16}$$

These equations represent the increase in axial vorticity and axial vorticity gradient along a streamline. Next, the right-hand side terms of Equation (9.15) and of the z-component of Equation (9.16) are integrated along two particular streamlines from inlet to exit. For these two streamlines, Equation (9.16) is verified by comparing the increase in the z-derivative of the vorticity at the exit of the pipe to the vorticity gradient computed from the solution at the pipe exit. Moreover, the source-term for the axial vorticity is shown to be indeed small compared to the source-term for the axial vorticity

z-derivative as declared by hypothesis.

In Figure 9.8a) two streamlines are shown in front and side views. The axial vorticity derivative $\partial\omega_x/\partial z$ computed from the Eulerian/Lagrangian solution is plotted on the pipe exit cross-section with indication of the location of the end-points of the two streamlines. Figure 9.8c) represents the values of the right-hand sides of Equation (9.15) (S_{ω_x}) and of the third component of Equation (9.16) ($S_{\partial\omega_x/\partial z}$) (i.e. the source-terms for the axial vorticity and the source-terms for the z-derivative of the axial vorticity) integrated on the two streamlines from the pipe inlet to the pipe exit. The source-terms for the axial vorticity are indeed small for both streamlines compared to axial vorticity magnitude, therefore justifying the use of Equation (9.8). The magnitude of the source-terms for the z-derivative of the axial vorticity is larger for both streamlines.

These terms, taken at the exit of the pipe, correspond to the values of vorticity derivative displayed on the pipe exit cross-section in Figure 9.8b) at the indicated end-point location of the two streamlines (the inlet values for $\partial\omega_x/\partial z$ is 0). Thus, Equation (9.16) is verified for the particular case of these two streamlines and show that the strain and rotation fields combination integrated on the right-hand side of the equation is indeed the mechanism by which the large vorticity gradients are created along the pipe. Furthermore, these vorticity gradients appear in the absence of strong source-terms for the vorticity. For this test case, the vorticity is essentially convected passively with the material particles.

9.2.2 Introduction of a pseudo-diffusion term

Examining Figures 9.7a) and 9.7b), it can be seen that the information between two markers at the inlet is lost because of the lack of grid resolution. In order to transport information from approximately one marker per cell at the inlet, the grid resolution in the cross-flow plane would have to be increased by a factor ~ 4 near the pipe exit, see Figure 9.7a). As the strength of the swirling flow increases, so does the required grid

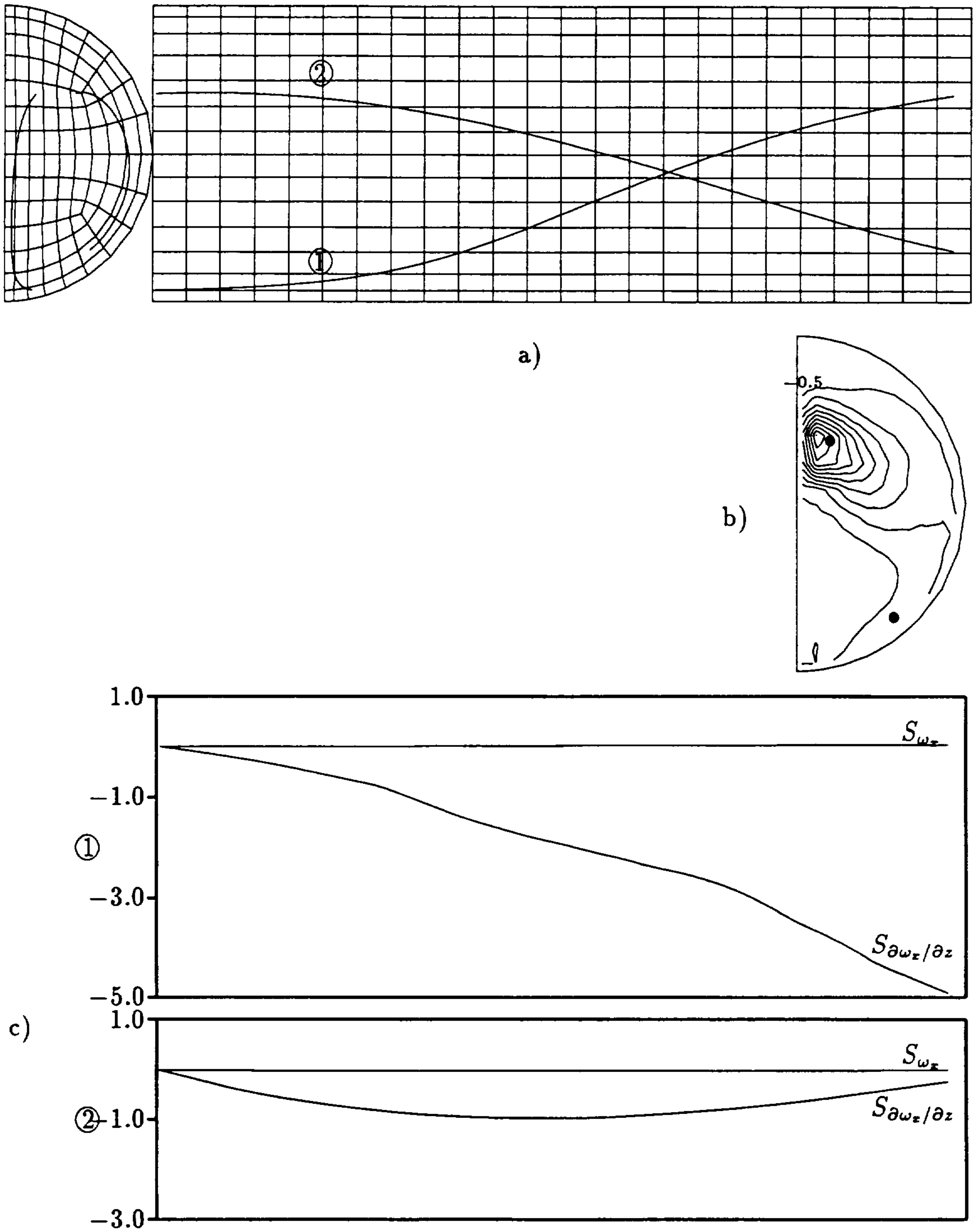


Figure 9.8: Case $C = 0.3$: a) Front and side views of pipe with 2 streamlines, b) axial vorticity derivative $\partial\omega_x/\partial z$ on exit cross-section with end-points of streamlines (inc.=0.5), c) source-term for axial vorticity: S_{ω_x} and source-term for $\partial\omega_x/\partial z$: $S_{\partial\omega_x/\partial z}$ along the 2 streamlines.

resolution. However, because the representation of vorticity gradient augmentation is inviscid, as the grid resolution increases so do the gradients of vorticity as shown in Figure 9.6.

When the strength of the swirling flow is increased by a factor 2, i.e. $C = 0.6$, the vorticity gradients are further intensified to a point where the high vorticity gradient region covers less than one cell. (This region separates a high vorticity from a low vorticity). This phenomenon is shown in Figure 9.9 where contours in axial vorticity are drawn for diverse stations along the pipe. The vorticity gradient is intensifying along the pipe. When using the Eulerian scheme alone, the numerical diffusion spreads the high gradient of vorticity region over a few cells, whereas when using the Lagrangian correction, wiggles in the high vorticity gradient region appear because the representation of this region by markers placed at cell centers (or by state-vectors at the grid nodes) becomes inadequate near the pipe exit (as seen when superimposing the grid on the exit-cross-section in Figure 9.9b)). As mentioned before, the insufficient spatial representation of the flow destabilizes the Eulerian solution through the correction procedure. For example, the inaccurate velocity corrections lead to erroneous perturbations of the axial velocity through the solution of the continuity equation.

As shown before, the use of larger size grids does not help to overcome the problem since the gradient region concentrates. Therefore, the gradients have to be 'controlled' so as to be supported on the fixed size grid. The goal of the Lagrangian correction technique is still the reduction of the numerical diffusion of the Eulerian solver without resorting to larger size grids. However, in the present flow cases, because of the vorticity gradient augmentation phenomenon, the additional control of vorticity gradients is required in order to get a stable Eulerian/Lagrangian solution.

The present section shows that the introduction of a pseudo-diffusion term in the Helmholtz equation allows to control the strength of the vorticity gradients in the flow. If the diffusion level introduced by the Helmholtz pseudo-diffusion is maintained lower than the numerical diffusion of the Eulerian scheme, the resulting solution will still show

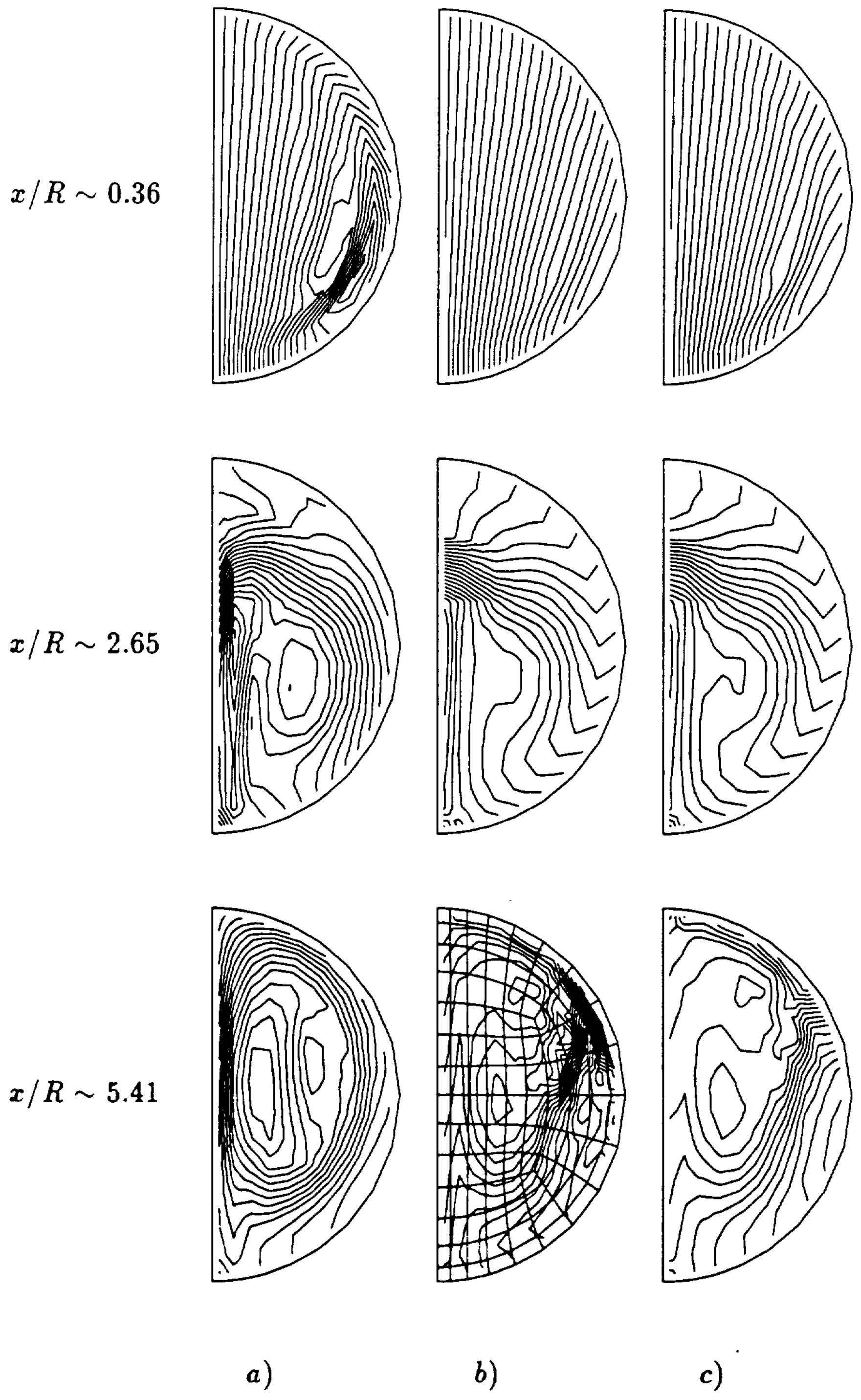


Figure 9.9: Case $C = 0.6$: Axial vorticity contours at diverse cross-sections along the pipe, a) Eulerian solution, b) Eulerian/Lagrangian solution, c) Eulerian/Lagrangian solution including Helmholtz smoothing term (increment=0.2).

a substantial improvement over the Eulerian solution alone.

The pseudo-diffusion term is chosen such as to take the same form as the real diffusion term arising in a viscous fluid, hence the Helmholtz equation is now written as

$$\frac{D\vec{\omega}}{Dt} = (\vec{\omega} \cdot \nabla)\vec{v} + \nu_l \nabla^2 \vec{\omega}, \tag{9.17}$$

where the coefficient ν_l is the Lagrangian pseudo-diffusion coefficient proportional to the mesh size h. The ratio of the right-hand side terms is of the order of the pseudo-Reynolds number Re_L and is a measure of the tendency of the vortex lines to be frozen to the fluid compared to the tendency of the vorticity to diffuse in the flow. Using V and L to represent the reference velocity and length, then

$$\frac{(\vec{\omega} \cdot \nabla)\vec{v}}{\nu_l \nabla^2 \vec{\omega}} \sim \frac{V\, L}{\nu_l} \sim Re_L. \tag{9.18}$$

The pseudo-diffusion term is added to the tilting/stretching term and integrated along the marker trajectory using a predictor-corrector scheme. The evaluation of the Lagrangian pseudo-diffusion term in the Helmholtz equation is similar to the evaluation of the smoothing term in the Lax-Wendroff algorithm of Section 3.2. However, the vector $\vec{\omega}$ is known at the cell centers instead of the grid nodes. Therefore, the average values $\bar{\vec{\omega}}$ are first defined at the cell nodes by averaging the values of $\vec{\omega}$ at the eight surrounding cell centers. The Lagrangian pseudo-diffusion term is then evaluated at a cell center by summing the differences between the eight nodes average values and the cell center value. The vorticity Laplacian is then interpolated from the cell centers to the marker location for each Lagrangian time-step.

Definition of vorticity at a wall node

When defining the node average values of $\vec{\omega}$, the wall nodes receive contribution from only four cell centers. The vorticity value at the wall nodes is determined through the following relations defining the vorticity gradients at a node by a surface integral over

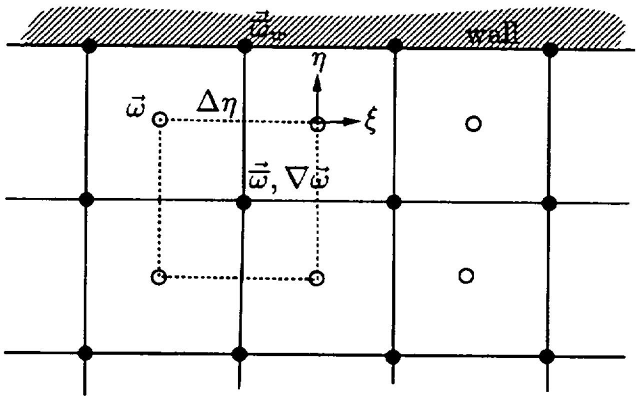


Figure 9.10: Estimation of the vorticity at a wall node.

a pseudo-cell based on the cell centers as sketched in Figure 9.10

$$\nabla\omega_x\, V = \int_V \nabla\omega_x dV \quad = \quad \int_S \omega_x \vec{n} dS, \tag{9.19}$$

$$\nabla\omega_y\, V = \int_V \nabla\omega_y dV \quad = \quad \int_S \omega_y \vec{n} dS, \tag{9.20}$$

$$\nabla\omega_z\, V = \int_V \nabla\omega_z dV \quad = \quad \int_S \omega_z \vec{n} dS, \tag{9.21}$$

where V is the volume of the pseudo-cell. Using the nomenclature defined in Section 3.1, the vorticity gradient at node 1 of Figure 3.1 is defined as the sum from the contributions from the eight surrounding cells A to H. The contribution of cell A is given by

$$\nabla\omega_{x1} \quad = \quad \frac{1}{4V_1}\int_{\overline{A_1},\overline{A_2},\overline{A_3}} \omega_x \vec{n} dS, \tag{9.22}$$

$$\nabla\omega_{y1} \quad = \quad \frac{1}{4V_1}\int_{\overline{A_1},\overline{A_2},\overline{A_3}} \omega_y \vec{n} dS, \tag{9.23}$$

$$\nabla\omega_{z1} \quad = \quad \frac{1}{4V_1}\int_{\overline{A_1},\overline{A_2},\overline{A_3}} \omega_z \vec{n} dS, \tag{9.24}$$

and so on for the contributions from cells B to F. The value of $\vec{\omega}_w$ at the wall node is then found from the nearest node in the direction perpendicular to the wall as sketched in Figure 9.10 by using the values of $\vec{\omega}$ and $\nabla\vec{\omega}$ for this node as

$$\vec{\omega}_w = \vec{\omega} + \frac{\partial\vec{\omega}}{\partial\eta}\Delta\eta, \qquad \text{where} \quad \frac{\partial}{\partial\eta} = \frac{\partial}{\partial x}\frac{\partial x}{\partial\eta} + \frac{\partial}{\partial y}\frac{\partial y}{\partial\eta} + \frac{\partial}{\partial z}\frac{\partial z}{\partial\eta}. \qquad (9.25)$$

Figure 9.9c) shows the axial vorticity contours for the same cross-sections as Figure 9.9a) and 9.9b) for the Eulerian/Lagrangian solution with a Lagrangian pseudo-diffusion term added to the Helmholtz equation. The axial vorticity contours are smoother showing that the the addition of the pseudo-diffusion terms allows to control the vorticity gradients. The resulting vorticity field is more accurately represented on the fixed-size grid and the Eulerian/Lagrangian scheme is stable. As designed, the Lagrangian pseudo-diffusion term is shown to be very effective at smoothing out the strong vorticity gradients in the flow without affecting the smooth vorticity regions. The numerical value of ν_l used for this test case is 0.05.

The amount of diffusion introduced in the solution is quantified by looking at the changes in circulation around a closed curve moving with the flow. The circulation around the closed curve initially placed around the inlet cross-section is shown in Figure 9.11 as a function of the convection distance along the pipe (measured as a mean value of all the convective markers positions) for the Eulerian solution and the Eulerian/Lagrangian solution with the addition of the Lagrangian pseudo-diffusion term in the Helmholtz equation. In Figure 9.11, the change in circulation when using an Eulerian/Lagrangian scheme and a Lagrangian pseudo-diffusion term is shown to be small compared to the change observed in the circulation using the basic Eulerian solution.

The convective change in circulation around a closed curve due to the introduction of a diffusion term in the Helmholtz equation can also be expressed as a contour integral of the curl of the vorticity as given in Appendix F.

As the strength of the swirling flow increases, the phenomenon of vorticity gradient augmentation is responsible for the inaccurate representation of the vorticity in the flow which is the cause for a destabilization of the combined Eulerian/Lagrangian scheme.

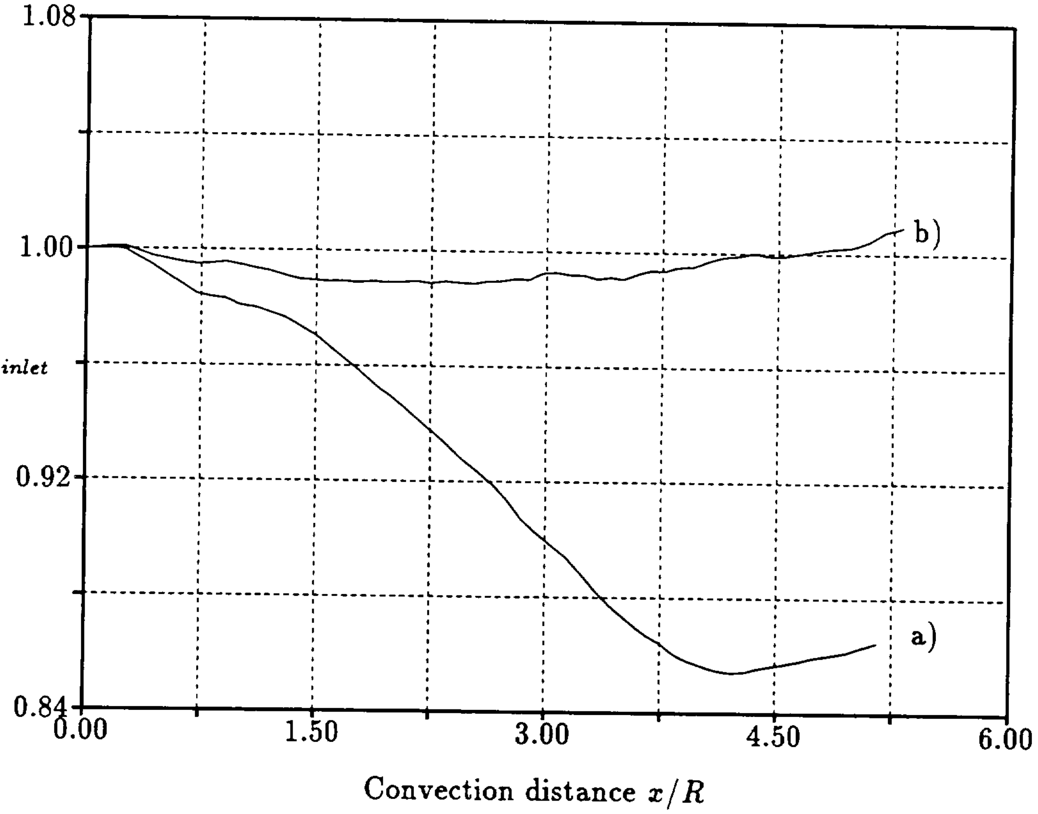


Figure 9.11: Case $C = 0.6$: Circulation around a closed curve, a) Eulerian solution, b) Eulerian/Lagrangian with smoothing term in the Helmholtz equation.

The introduction of a pseudo-smoothing term in the Helmholtz equation has been shown to successfully smooth out the strong vorticity gradients and allows one to obtain a stable Eulerian/Lagrangian solution. By comparing circulations around a closed convecting material curve, the Eulerian/Lagrangian solution gives still a substantial improvement over the Eulerian solution alone. The computation of the pseudo-smoothing term results in a $\sim$ 5% increase of CPU over the Eulerian/Lagrangian basic solution.

Chapter 10

Constant Stagnation Pressure Flow in a 90^0 Bend

The incompressible flow in a 90^0 bend of circular cross-section with constant stagnation pressure inlet conditions is examined. The constant stagnation pressure is obtained at the inlet through constant pressure and constant velocity conditions. The inlet velocity is set as part of the inlet boundary conditions. The inlet surface is located far enough from the bend so that the upstream influence of the bend is negligible and the constant pressure condition is also obtained.

The exact solution for the flow field is at constant stagnation pressure since the stagnation pressure remains constant along a streamline and each streamline is characterized by the same stagnation pressure. In turn, the vorticity in the domain is zero everywhere because the flow is irrotational upstream of the inlet cross-section. From [4], if at any instant in time before $t = 0$ (time at which particles cross the inlet section) the flow is irrotational, then the flow remains irrotational for any subsequent instant. That is if all derivatives of $\vec{\omega}$ are zero for any time previous to $t = 0$, then

$$\frac{D\vec{\omega}}{Dt} = \frac{D^2\vec{\omega}}{Dt^2} = ... = \frac{D^n\vec{\omega}}{Dt^n} = 0, \tag{10.1}$$

and if all derivatives are defined, Taylor's theorem shows that the quantity $\vec{\omega}$ vanishes for all subsequent instants in time. Alternatively, Hawthorne showed in [26] that the growth of streamwise vorticity is a function of the stagnation pressure field only, so that no streamwise vorticity can be created in a constant stagnation pressure field.

Because of truncation errors, the Eulerian solution does not exactly satisfy the constant stagnation pressure and zero vorticity conditions. By computing Eulerian solutions on a coarse and a fine grid, this test case provides for a calibration of local and global errors in stagnation pressure and vorticity. Also, by using the Eulerian/Lagrangian scheme, the errors in vorticity and stagnation pressure are shown to be less than for the Eulerian solution alone. This test case also provides a check on the numerical integration of the vorticity source-terms $(\vec{\omega} \cdot \nabla)\vec{v}$. These must be driven towards zero (starting from an Eulerian solution with errors in vorticity and stagnation pressure) since the exact solution for the flow is irrotational. The improvement due to the addition of the Lagrangian correction is compared to the improvement obtained when using the Eulerian scheme on a finer grid.

Again, since the flow is symmetrical, the computations are performed only one half of the pipe. The geometry used for this test case is taken from the Enayet et al. [21] flow data and will also be used in Chapter 11 for secondary flow calculations. The geometry and grids are identical to the ones used for the accuracy study of Section 3.9.

The pipe diameter is 0.048 m and the ratio of radius of curvature to pipe diameter is 2.8. The computational domain extends two diameters upstream of the bend inlet cross-section and two diameters downstream of the bend exit cross-section. As mentioned before, the distance of two diameters upstream is required to satisfy the constant stagnation pressure condition at the pipe inlet cross-section. The calculations are performed on a coarse grid with 43 stations along the bend total length and 189 nodes in a cross-section. The fine grid is formed by 85 cross-sections with 713 nodes in a cross-section. The fine grid is composed of eight times as many cells as the coarse grid. The two grids are shown in Figure 10.1 in side and frontal views. The cross-sections at which the results are presented are indicated. Eulerian solutions are computed on both grids, whereas an Eulerian/Lagrangian solution is computed on the coarse grid only.

The Lagrangian correction uses the upstream streamline integration scheme where a marker is placed at the center of every cell and the streamlines are integrated backwards

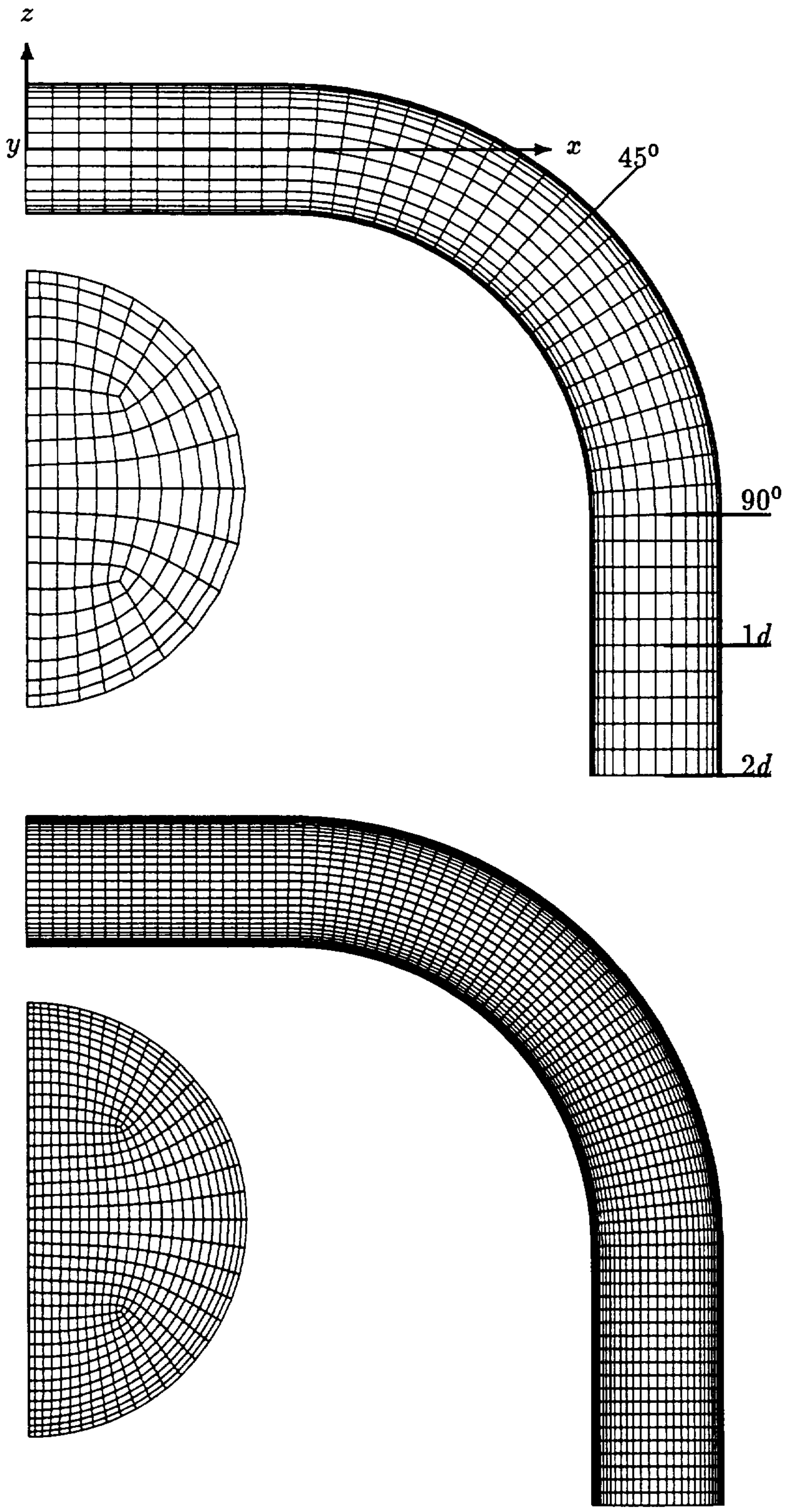


Figure 10.1: Coarse and fine grids front and side views (189 × 43 nodes and 713 × 85 nodes) with particular cross-sections.

with approximately two steps per cell until they reach the inlet. The streamlines are recomputed every 50 iterations of the Eulerian solver. The correction step is applied once at each Eulerian iteration. An under-relaxation on the vorticity correction is used according to Equation (6.48) in order to limit the perturbations to the Eulerian solver. The under-relaxation factor R_f is 0.5%.

Since the Lagrangian correction technique is based on vorticity corrections, the local errors are first shown in terms of vorticity errors. Contours in the z-component of vorticity are shown in Figure 10.3 for a cross-section located at 45^0 in the bend. The coarse grid Eulerian solution and the coarse Eulerian/Lagrangian solution are shown in Figures 10.3a) and 10.3b). The fine grid Eulerian solution is represented in Figure 10.3c). Similarly, in Figures 10.4, 10.5 and 10.6, the contours of the z-component of vorticity are drawn for three cross-sections located at 90^0 (bend exit cross-section) and at 1 and 2 diameters downstream of the bend exit cross-section (the z-component of vorticity is streamwise for these stations). The maximum amount of error in the cross-section is indicated in each case. The vorticity is normalized with respect to the group U_{in}/R representing the inlet velocity divided by the pipe radius.

The errors in vorticity are concentrated near the wall of the pipe. By comparing the three solutions, it is clear that the combined Eulerian/Lagrangian scheme is very effective at reducing the errors in vorticity in the flow field. Indeed, in the Eulerian/Lagrangian solution, the error in the z-component of vorticity is reduced to nearly zero, whereas using the Eulerian scheme on a finer mesh leads to very little reduction of the error in vorticity. In the coarse grid Eulerian solution, the errors in vorticity are more diffused in the domain, but the level of error remains approximately the same for the coarse and the fine grid.

The results are also calibrated in terms of the local error in stagnation pressure coefficient ΔC_{p_0} defined at any node by Equation (3.75).

Figure 10.7 shows contours in ΔC_{p_0} on the 45^0 cross-section for the coarse grid Eulerian and Eulerian/Lagrangian solutions and the fine grid Eulerian solution. The 90^0, $1d$ and $2d$ cross-section results are represented in Figures 10.8, 10.9 and 10.10, respectively. The maximum error is indicated for each cross-section. The error in stagnation pressure is the largest at the inside of the bend near the 90^0 station where the pressure recovery occurs. By using the Eulerian/Lagrangian scheme, the error in stagnation pressure is reduced locally when compared to the reference case of the coarse grid Eulerian solution. The maximum error in stagnation pressure for the Eulerian/Lagrangian solution is even lower than for the fine grid Eulerian solution. Since the stagnation pressure is not transported along the streamlines in the Lagrangian scheme, the correction of the stagnation pressure occurs indirectly through the vorticity correction (Crocco's equation relates vorticity and stagnation pressure gradient) as mentioned in Section 4.2. The corrected velocities alter the flux balance around each cell in the Lax-Wendroff algorithm so that the pressure is corrected too.

A global indicator of stagnation pressure losses is the L_2 norm of the stagnation pressure losses for the domain defined as ϵ_{p_0} by Equation (3.74).

Even if the maximum local error in stagnation pressure is smaller with the Eulerian/Lagrangian scheme than with the fine grid Eulerian solution, the errors are less diffused with the latter so that an average measure over the nodes like the L_2 norm gives a more unfavorable result for the Eulerian/Lagrangian solution. The L_2 norm of the stagnation pressure errors decreases by a factor ~ 2.3 when using the Eulerian/Lagrangian solution, whereas the Eulerian solution on the finer grid leads to a reduction factor of ~ 4.0 (which is in accordance with the Lax-Wendroff scheme second-order accuracy).

In an inviscid flow, by expressing the gradient of pressure in terms of the local coordinates system (s, n, b) where s, n and b stand for the streamwise, normal and binormal directions as shown in Figure 10.2, the gradient of pressure along the binormal direction can be shown to be zero [39]. This also implies that the norm of the velocity presents no dependence on the binormal direction since the flow is at constant stagnation

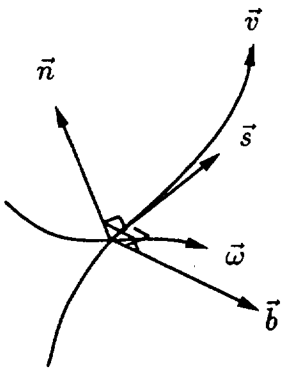
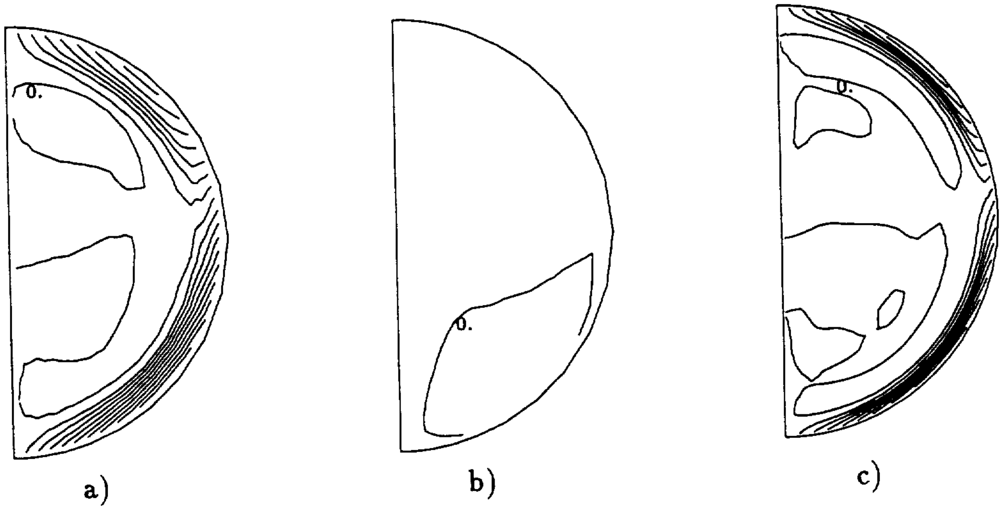


Figure 10.2: Local coordinate system (s, n, b).

pressure for this test case. Also, because the secondary vorticity is zero (as is the case for the other components of vorticity), the streamlines do not present any torsion, and cross the grid cross-sections perpendicularly. The contours in velocity norm are represented in Figures 10.11 and 10.12 for the 45^0 and 90^0 stations, respectively. These plots provide an indication of the vorticity correction near the pipe wall. The exact solution presents no dependence on the binormal direction (here the direction perpendicular to the symmetry surface, i.e. the y-direction). The reduction of the spurious deformation of the contours of velocity norm is larger when using the Eulerian/Lagrangian scheme than when using the Eulerian scheme on the fine grid.

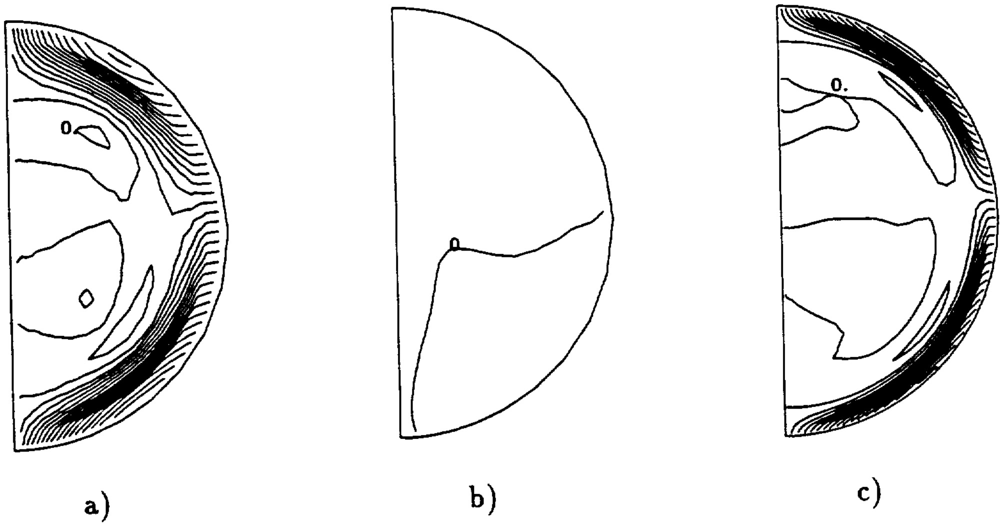
The Eulerian/Lagrangian scheme is clearly more efficient than the fine grid Eulerian solution at reducing the errors in vorticity in the flow. Indeed, while the vorticity errors are more concentrated when using a finer grid, the level of vorticity remains approximately the same in the coarse or the fine grid Eulerian solution. Also, it has been shown that the vorticity correction of the Lagrangian scheme results in a more accurate solution of the velocity field than the fine grid Eulerian solution. The maximum error in stagnation pressure in the field is lower when using the Eulerian/Lagrangian scheme than the fine grid Eulerian scheme. However, since the stagnation pressure errors are more diffused in the flow with the Eulerian/Lagrangian scheme, the reduction in the L_2 norm of the stagnation pressure losses is lower than with the Eulerian solution on the fine grid (a factor 2.3 compared to a factor 4.0).

The Eulerian/Lagrangian solution is substantially less expensive than the fine grid Eulerian solution. The addition of the Lagrangian correction leads to an increase of $\sim 75\%$ of the basic coarse Eulerian solution. In comparison the fine grid Eulerian solution requires 16 times more CPU than the coarse grid Eulerian solution (the Eulerian solution on the coarse grid requires ~ 2000 iterations to converge to a maximum residual of $\sim 1 \times 10^{-5}$ (the initial flow is interpolated from a coarser grid solution), the Eulerian/Lagrangian solution requires ~ 390 iterations to converge starting from the converged Eulerian solution and takes ~ 10.6 seconds/iteration in average).



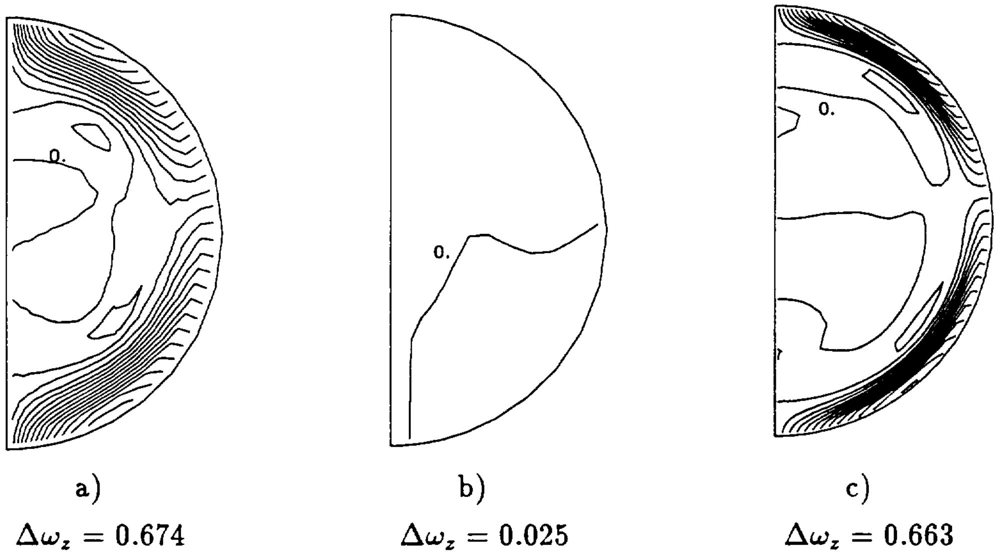
$\Delta\omega_z = 0.480$ $\Delta\omega_z = 0.015$ $\Delta\omega_z = 0.425$

Figure 10.3: Contours of z-component of vorticity at 45^0 station, a) Eulerian solution, b) Eulerian/Lagrangian solution, c) Eulerian solution on fine grid (increment = 0.05).



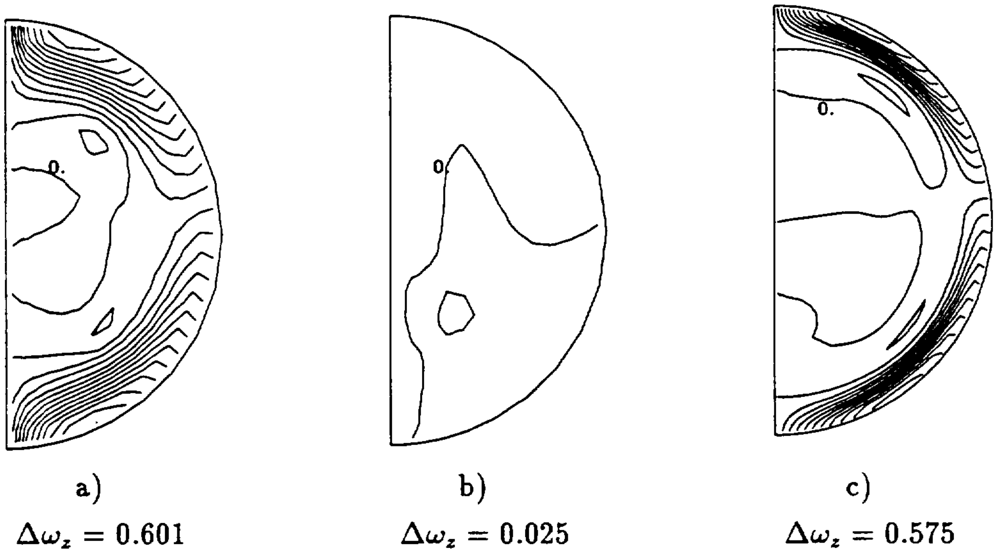
$\Delta\omega_z = 0.845$ $\Delta\omega_z = 0.027$ $\Delta\omega_z = 0.809$

Figure 10.4: Contours of streamwise vorticity at 90^0 station, a) Eulerian solution, b) Eulerian/Lagrangian solution, c) Eulerian solution on fine grid (increment = 0.05).



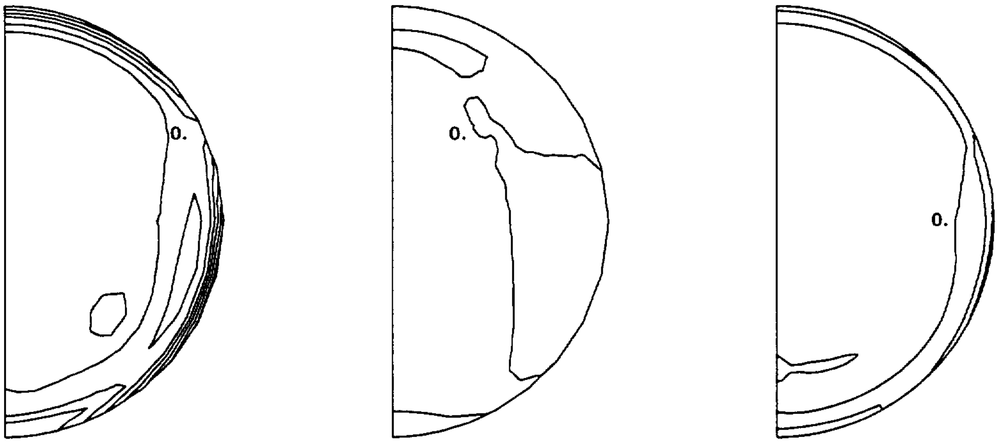
a) $\Delta\omega_z = 0.674$ b) $\Delta\omega_z = 0.025$ c) $\Delta\omega_z = 0.663$

Figure 10.5: Contours of streamwise vorticity at 1d station, a) Eulerian solution, b) Eulerian/Lagrangian solution, c) Eulerian solution on fine grid (increment = 0.05).



a) $\Delta\omega_z = 0.601$ b) $\Delta\omega_z = 0.025$ c) $\Delta\omega_z = 0.575$

Figure 10.6: Contours of streamwise vorticity at 2d station, a) Eulerian solution, b) Eulerian/Lagrangian solution, c) Eulerian solution on fine grid (increment = 0.05).

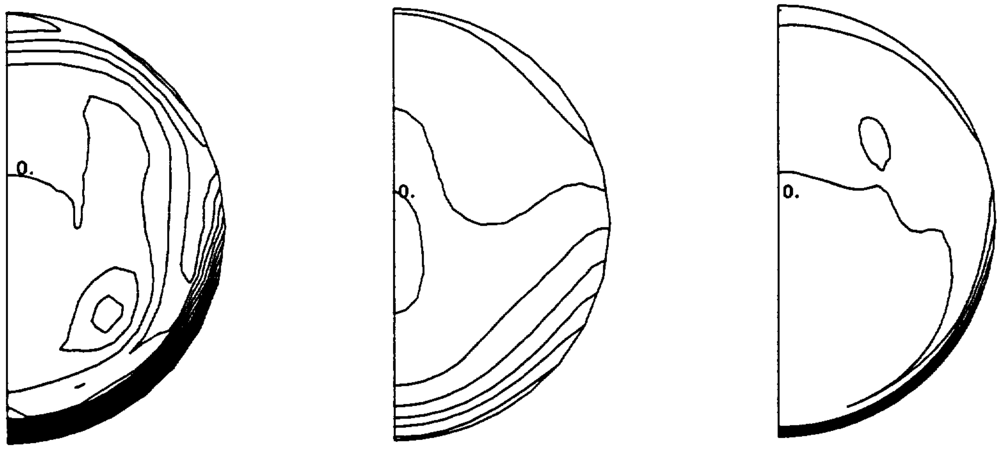


a) $(\Delta C_{p_0})_{max} = 2.5\%$

b) $(\Delta C_{p_0})_{max} = 0.9\%$

c) $(\Delta C_{p_0})_{max} = 1.0\%$

Figure 10.7: Contours for the local error in stagnation pressure coefficient ΔC_{p_0} at 45^0 station, a) Eulerian solution, b) Eulerian/Lagrangian solution, c) Eulerian solution on fine grid (increment = 0.005).

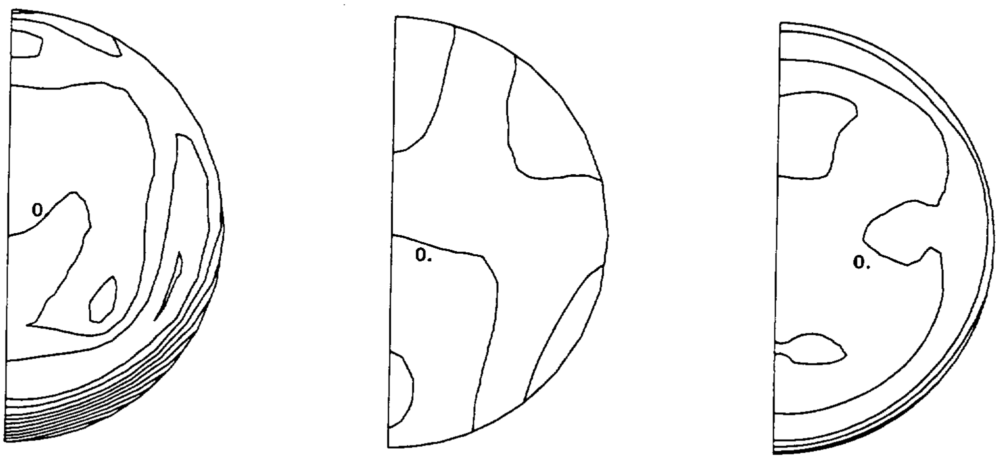


a) $(\Delta C_{p_0})_{max} = 10.6\%$

b) $(\Delta C_{p_0})_{max} = 2.7\%$

c) $(\Delta C_{p_0})_{max} = 4.9\%$

Figure 10.8: Contours for the local error in stagnation pressure coefficient ΔC_{p_0} at 90^0 station, a) Eulerian solution, b) Eulerian/Lagrangian solution, c) Eulerian solution on fine grid (increment = 0.005).

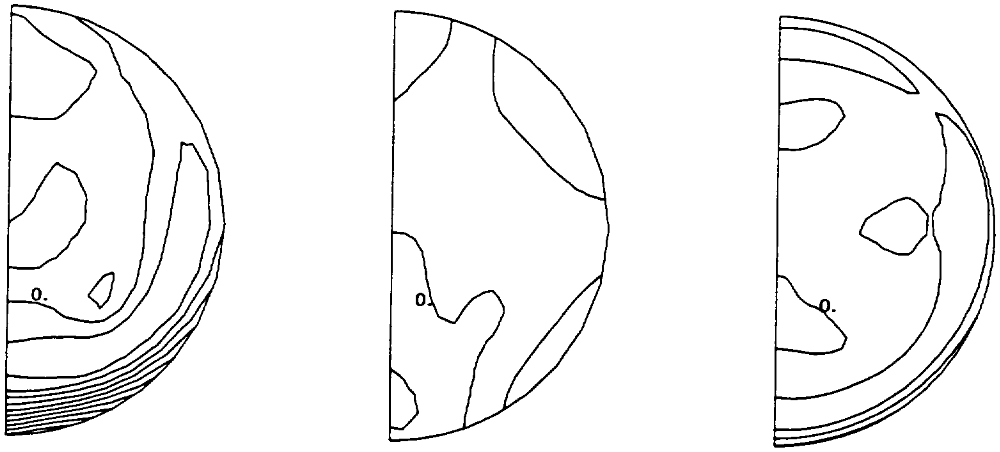


a) $(\Delta C_{p_0})_{max} = 4.6\%$

b) $(\Delta C_{p_0})_{max} = 0.8\%$

c) $(\Delta C_{p_0})_{max} = 1.3\%$

Figure 10.9: Contours for the local error in stagnation pressure coefficient ΔC_{p_0} at 1d station, a) Eulerian solution, b) Eulerian/Lagrangian solution, c) Eulerian solution on fine grid (increment = 0.005).



a) $(\Delta C_{p_0})_{max} = 4.4\%$

b) $(\Delta C_{p_0})_{max} = 0.9\%$

c) $(\Delta C_{p_0})_{max} = 1.1\%$

Figure 10.10: Contours for the local error in stagnation pressure coefficient ΔC_{p_0} at 2d station, a) Eulerian solution, b) Eulerian/Lagrangian solution, c) Eulerian solution on fine grid (increment = 0.005).

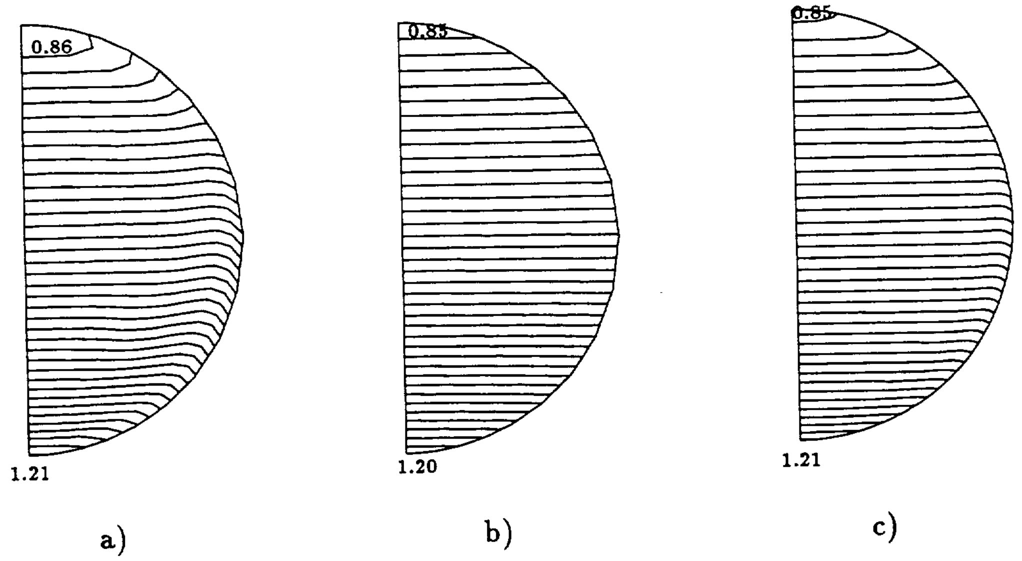


Figure 10.11: Contours in velocity norm at 45^0 station, a) Eulerian solution, b) Eulerian/Lagrangian solution, c) Eulerian solution on fine grid (increment = 0.01).

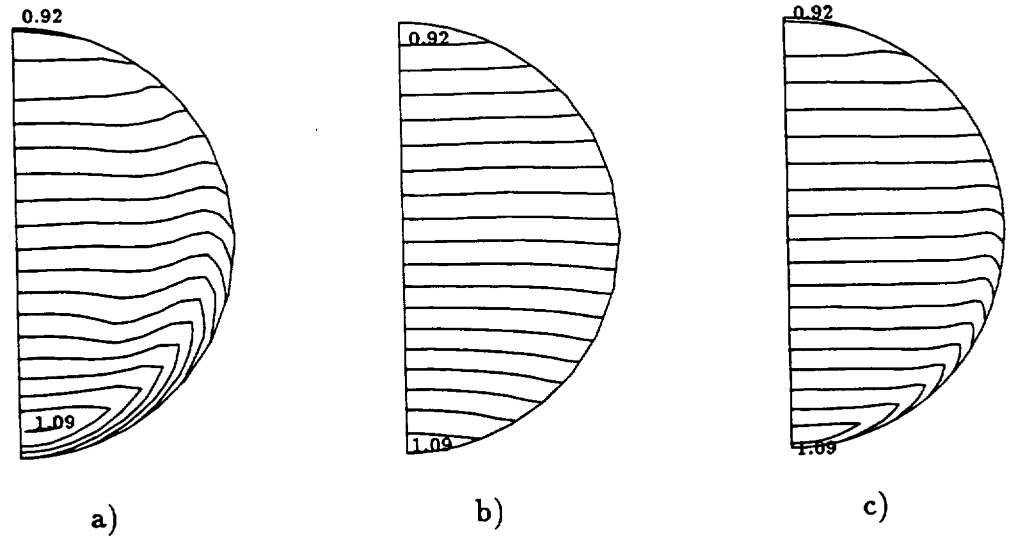


Figure 10.12: Contours in velocity norm at 90^0 station, a) Eulerian solution, b) Eulerian/Lagrangian solution, c) Eulerian solution on fine grid (increment = 0.01).

Chapter 11

Secondary Flow in Bent Pipes

The secondary flow phenomenon results from the existence of a vorticity component in the 'throughflow' direction. As the flow moves along a bend, each fluid particle is subjected to a centrifugal force acting in a direction of a line going through the center of curvature of the streamline. If the velocity profile is non-uniform, the centrifugal force acting on faster moving particles will be larger than the one acting on slower moving particles. This induces a motion of the particles in a cross-flow plane with non-uniform cross-flow velocities. For example, a fully-developed pipe flow profile presenting higher velocities at the pipe center and lower velocities near the pipe wall will lead to secondary flow when taken through a bent pipe. Under the action of the centrifugal forces, the slow moving fluid is pushed inwards whereas the fast moving fluid moves towards the bend outer wall. Because of continuity, the fast moving fluid displacement towards the outer wall forces slow moving fluid to convect along the pipe wall towards the pipe inner wall. A circular motion constituting the secondary flow phenomenon is then established in a cross-flow plane. This is illustrated in Figure 11.1 taken from [19]. The effect of the secondary flow in bent pipes of circular cross-section is then to displace the high velocity regions towards the outer wall. But if the inlet boundary-layer is thin, the main core flow will behave approximately as a free-vortex flow through the bend and the displacement of particles in cross-flow planes will be small compared to the displacement along the streamwise direction.

As a result of the Helmholtz equation, the secondary flow in bent pipes can be explained in terms of the turning, stretching and diffusion of the vorticity attached to a

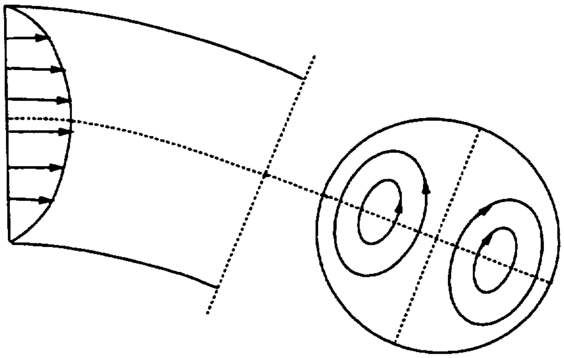


Figure 11.1: Secondary flow in a bent pipe due to non-uniform streamwise velocity.

given fluid particle. For the flow in a bent pipe, the vorticity may be initially present in the inlet velocity profile. In the case where the flow is fully developed at the inlet of a pipe of circular cross-section, the inlet vortex lines form concentric rings. If the diffusion term is ignored, the secondary vorticity is due only to the turning and stretching of the legs of the inlet vortex lines which are parallel to the pipe symmetry-plane, as two particles on the inner and outer radii convect through the bend with different speeds. This is illustrated in Figure 11.2 where a few vortex lines are drawn from an Eulerian solution of a flow in a 90^0 bend. At the inlet of the computational domain, the flow is fully developed and the vortex lines present no throughflow component. Near the bend inlet, the few vortex lines drawn near the wall surface indicate the presence of secondary vorticity due to tilting and stretching. The phenomenon intensifies as the flow moves down the bend as shown by the vortex lines drawn near the bend exit.

The creation of the secondary flow in the bend has been described so far as an 'inviscid' flow phenomenon. The additional effect of the developing boundary-layers or of the wall log region in the case of a fully developed flow is to counteract the development of the inviscid secondary flow as explained in [21]. The boundary-layer on the inner wall of the pipe is subjected to a favorable pressure gradient, so that it grows slowly. On the outer wall the adverse pressure gradient causes the boundary-layer to thicken. The result of taking the boundary-layers into account is then to displace the high velocity regions

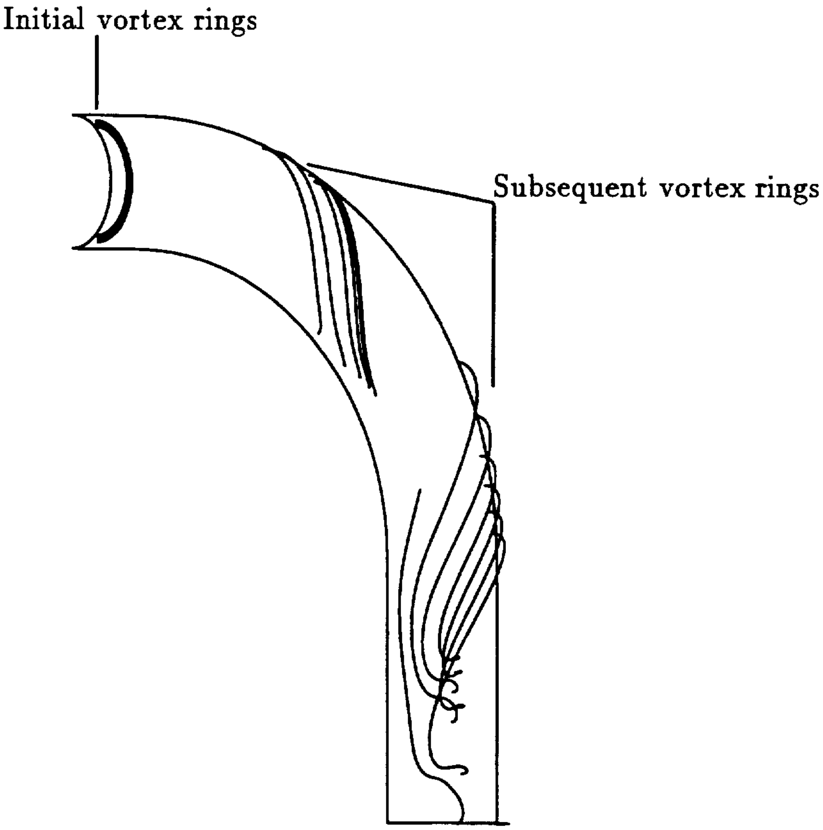


Figure 11.2: Secondary flow generation by tilting and stretching of vortex lines in a 90^0 bend.

more towards the inner wall, an effect inverse to the prediction of the inviscid secondary flow theory. If the bend exit is under constant pressure conditions (for example due to a straight section or an exit at atmospheric pressure) the streamwise pressure gradients have to be reversed near the exit in order to readjust to the downstream pressure. The evolution of the boundary-layer is therefore complex to predict.

The secondary flow phenomenon is a feature common to many flow problems and has been the object of extensive studies. Both approximate solutions and numerical computations have been tested against experiments. Berger et al. [10] review article provides an extensive list of references for the flow in curved pipes. Hawthorne [28] studied the applicability of secondary flow analyses for the solution of internal flow cases. Ackeret [2] also described peculiarities linked to the internal flow behavior.

A well-known approximate solution for the secondary vorticity in a bend or a cascade of airfoils is the Squire and Winter relation [67], which links the secondary vorticity generation to the bend angle and the inlet vorticity. Rowe [59] compared experiments and computations based on the Squire and Winter formulation for a 180^0 bend and found reasonable agreement up to an angle of about 75^0. Detra [19] also derived an approximate solution procedure based on small perturbations of the throughflow in good agreement with experiments in pipes of 21^0 and 42^0 bend angle. However, the pipe radius has to be small compared to the radius of curvature for this approximate solution to be valid.

More recently, Lakshminarayana [39] derived generalized expressions for the secondary vorticity using intrinsic coordinates. This work has been extended by Hawthorne [29] for stratified fluids in rotating systems.

An analytical formulation based on a streamlike function formulation of the inviscid flow in a curved duct was proposed by Abdallah [1] and tested on a 90^0 duct of rectangular cross-section. The turning of the velocity contours is predicted in good agreement with experiments. Briley [12] investigated three-dimensional viscous flows with large secondary vorticity by deriving a system of approximating equations for flows where a primary direction is present, but in which the transverse velocity components can be large. Also, an extensive series of laminar and turbulent Navier-Stokes solutions in bent pipes of circular cross-section and square ducts have been compared with experiments. A few of these are [35], [34], [33], [74], [79], [32], [52].

11.1 Motivation

The present work uses the Euler equations for the prediction of secondary flows in bent pipes. The motivation for the use of an inviscid formulation as reported by Hawthorne in [28] is based on the fact that large regions of the flow, although non-uniform, may be assumed frictionless since the influence of the walls extends only gradually inwards as the flow passes around the bend. Also, the presence of viscosity is not required for the generation of secondary vorticity even if the initial presence of vorticity at the inlet of the pipe is due to a viscous effect. The fundamental behavior of the flow can be predicted using the Euler equations since the development of the secondary flow is mainly the result of an inviscid process of vortex stretching and tilting. This applies for high Reynolds number flows since the ratio of time scales between the transverse viscous momentum diffusion and the convection time scale is proportional to the inverse of the Reynolds number for laminar flows. In turbulent flows, the ratio is roughly constant but still small.

Also, the Euler equations are not restricted by the small shear assumption used in the approximate solutions of [67] and [19] where the distortion of the vorticity by the secondary vorticity is neglected as a second-order effect. This is particularly important here, where high turning angles and high Reynolds numbers are of interest.

An underlying motivation is the study of the numerical diffusion effects on the solution. By comparing Eulerian solutions on coarse and fine grids, it is shown that the numerical diffusion inherent to the Eulerian solver has an important effect on the solution both in terms of strength and location of the secondary flow. The fact that the general effect of the numerical diffusion is shown to be qualitatively similar to the real viscous effects is a misleading result since the solution remains very sensitive to the amount of numerical diffusion. Therefore, very fine grids are required for secondary flow computations in order to obtain a solution in which the numerical diffusion effect is small. By computing the circulation around a closed curve moving with the fluid, it is shown that even the fine grid Eulerian solution still suffers from numerical diffusion, indicating the limitation of standard Eulerian solvers for the computation of secondary flows. The motivation for the introduction of the Lagrangian correction technique is to find an alternative to expensive large size grids Eulerian calculations of secondary flow phenomena. With the addition of the Lagrangian correction technique, the amount of numerical diffusion observed in the Eulerian solution is shown to be reduced for a given grid size when compared to the standard Eulerian solution, or equivalently the grid requirements are reduced to converge to a solution where the effects of the numerical diffusion are small.

The reduction of the spurious numerical diffusion to an unimportant level allows one to address the question of how much the secondary flow is influenced by the real fluid viscosity by comparing the truly inviscid numerical solution with the existing experimental data.

Earlier, Navier-Stokes solutions have been reported to suffer from numerical diffusion more than from the turbulence modeling uncertainties [35]. This was due to the coarse discretisation imposed by large memory size requirements. Today, the turbulence modeling together with near wall boundary conditions is likely to be the predominant uncertainty factor in the viscous solution of the secondary flow in bends as seen in [32]. In comparison, Eulerian/Lagrangian computations offer a computationally cheaper approach and, as seen in the result sections, still provide with a description of the basic flow behavior. However, the limitations of the present method are listed below.

11.2 Limitations

As mentioned earlier, the viscosity effects tend to reduce the development of the secondary flow. Therefore, inviscid numerical solutions tend to overpredict the secondary flow generation, a result consistent with the findings of this study. The effect of the growing boundary-layers is to narrow the channel and change the pressure distribution. The shear stress for a laminar boundary-layer subjected to a streamline curvature can be written as a function of δ/R_c where δ is the boundary-layer thickness and R_c is the radius of curvature of the pipe. For a turbulent boundary-layer, however, the effect of the streamline curvature on the boundary-layer thickness are an order of magnitude larger [11]. In the turbulent flow case, the curvature of the bend acts as a flow destabilizer near the outer wall whereas the flow is stabilized near the pipe inner wall [2]. At the same time, the fluid is transported from inner to outer wall by the secondary flow motion.

Another limitation for the use of the Euler equations is the requirement that a slip velocity has to be defined at the wall for the inlet velocity profile (if not reverse flow would immediately occur on the outside of the bend under the influence of the adverse pressure gradient). However, the problem is not singular to the Euler equations solutions, since the requirement of a 'cut-off' velocity at the wall also arises in the approximate solutions methods because the small shear assumption becomes invalid near the wall. Squire and Winter [67] used a 'cut-off' velocity of 45% of the mean velocity and Detra [19] tried both 65% and 80% cut-off velocities on the inlet velocity profile. As reported in [28], the limitation is important since the secondary vorticity depends on the incoming velocity profile vorticity. Hawthorne [27] found a cut-off velocity by canceling the overpredicting effect of the secondary vorticity of Squire and Winter by a large negative vorticity at the wall. However, this analysis is limited to small bend angles.

The present approach to the slip velocity definition for a turbulent inlet mean velocity profile is to neglect the laminar sublayer. The slip velocity is then chosen as the velocity just outside the laminar sublayer as mentioned in Chapter 8 where this approach has been used for the preservation of a turbulent inlet velocity profile in a pipe. The model is valid for turbulent boundary-layers where the thickness of the boundary-layer is much larger than the laminar sublayer. The present choice of $y^+ = 30$ as the upper limit of the laminar sublayer is arbitrary. However, this value is consistently used to determine the slip velocity for the two flow cases computed in this Section. Still, the best way to determine this slip velocity would be to calibrate the y^+ value by fitting numerical computations to measurements. This was not attempted here, since a wider study would be required in order to determine the best value of y^+ valid for a range of flow cases. Instead, as mentioned in Section 11.1, the present chapter deals with the reduction of the spurious numerical diffusion phenomenon.

Denton [18] uses a similar concept in his inviscid calculations of turbomachinery in order to introduce viscous effects in the momentum equation by the use of a distributed body force. The wall shear stress is obtained by neglecting the displacement thickness of the laminar sublayer, that is the surface streamline lies at the edge of the laminar sublayer. A loss function is derived as a power law distribution from the wall into the field. A value of y^+ of 10 for the edge of the laminar sublayer is recommended by [18] only if there are enough nodes to describe accurately the boundary-layer. In practice, a value between 10 and 40 produced the best fit with measurements.

A first attempt to counteract the overprediction of the secondary-flow is performed in this study by introducing a simple 'wall correction' where the velocity is adjusted at every node on the wall, using the universal log-law distribution between the nodes adjacent to the wall and the nodes on the wall. This method is described in more detail in Section 11.4.3.

The comparison of numerical solutions with experiments is, however, suffering from a lack of experimental data on the cross-flow itself for the cases treated here. Refraction of the laser beam at the water-plexiglass interface prevented the measurements of cross-section velocity components in the pipe of circular cross-section of the Enayet data set [21]. The data available is essentially composed of streamwise velocity contours for diverse cross-sections along the bend supplemented by pressure measurements along the wall for diverse angles around the bend. The absence of experimental measurements of the cross-flow (necessary to quantify directly the strength and position of the secondary flow) makes the comparison with the calculations incomplete. However, calculations are helpful since they indicate a correlation between streamwise velocity contours and position and strength of the cross-flow regions.

11.3 Outline

The first pipe geometry is taken from the Enayet et al. [21] data set while the second corresponds to a bend tested at the Gas Turbine Laboratory of MIT.

First, coarse and fine grid Euler solutions of the flow are compared to indicate the effect of numerical diffusion on the strength and position of the secondary flow. The Lagrangian correction technique is then introduced in order to reduce the level of numerical diffusion, allowing the identification of the real viscous effects by comparison with experiments. The numerical diffusion reduction is demonstrated by computing the circulation around bend cross-sections along the pipe and by tracing the circulation on a closed curve moving with the fluid. Then, results are shown using the wall correction method which attempts to counteract the overprediction of the secondary flow. Pressure measurements along the bend are also compared with experiments for the Enayet et al. geometry.

11.4 Enayet 90^0 bend case

The secondary flow development for an incompressible flow (water) through a 90^0 bend of constant circular cross-section is investigated. The results are compared to experimental data taken from the Enayet et al. [21] data set providing Laser-Doppler Velocimeter measurements of throughflow or 'streamwise' velocities at different cross-stations along the bend and pressure measurements along the wall for diverse angles around the bend.

The geometry is identical to the one used for the constant stagnation pressure flow in Chapter 10. The diameter of the pipe is 0.048 m. and the ratio of radius of curvature to pipe diameter is 2.8. The computations were performed on a coarse and a fine grid with 320×36 nodes and 1223×71 nodes, respectively. The grids front and side views are represented on Figure 11.3. The grid cross-sections are spaced evenly in the streamwise direction along the bend centerline. The test geometry extends three diameters downstream of the bend exit cross-section. In order to accelerate the calculations, the computational grid reaches only two diameters downstream. It has been verified by numerical experiments that the computational results are not affected by applying the exit boundary condition at such a reduced distance from the exit of the bend. The grid extends 0.58 diameters upstream of the bend inlet cross-section where the inlet velocity is known from measurements. Eulerian solutions are performed on both grids, whereas the Eulerian/Lagrangian solution is computed on the coarse grid only.

A particularity of this secondary flow problem is that the vorticity (either inlet boundary-layer vorticity or secondary vorticity) presents concentrated regions but is also present in a more diffused form in a large portion of the flowfield. This implies that the Lagrangian correction technique has to operate on a large extent of the flow therefore requiring a large amount of markers. The option to place a marker at the center of each cell has been chosen here for simplicity. The upstream integration of the streamlines is found to be more successful than the downstream integration since it implies an even distribution of the markers in the flow. The correction step is applied only once at each Eulerian iteration. Both an under-relaxation of the correction ($R_f =$

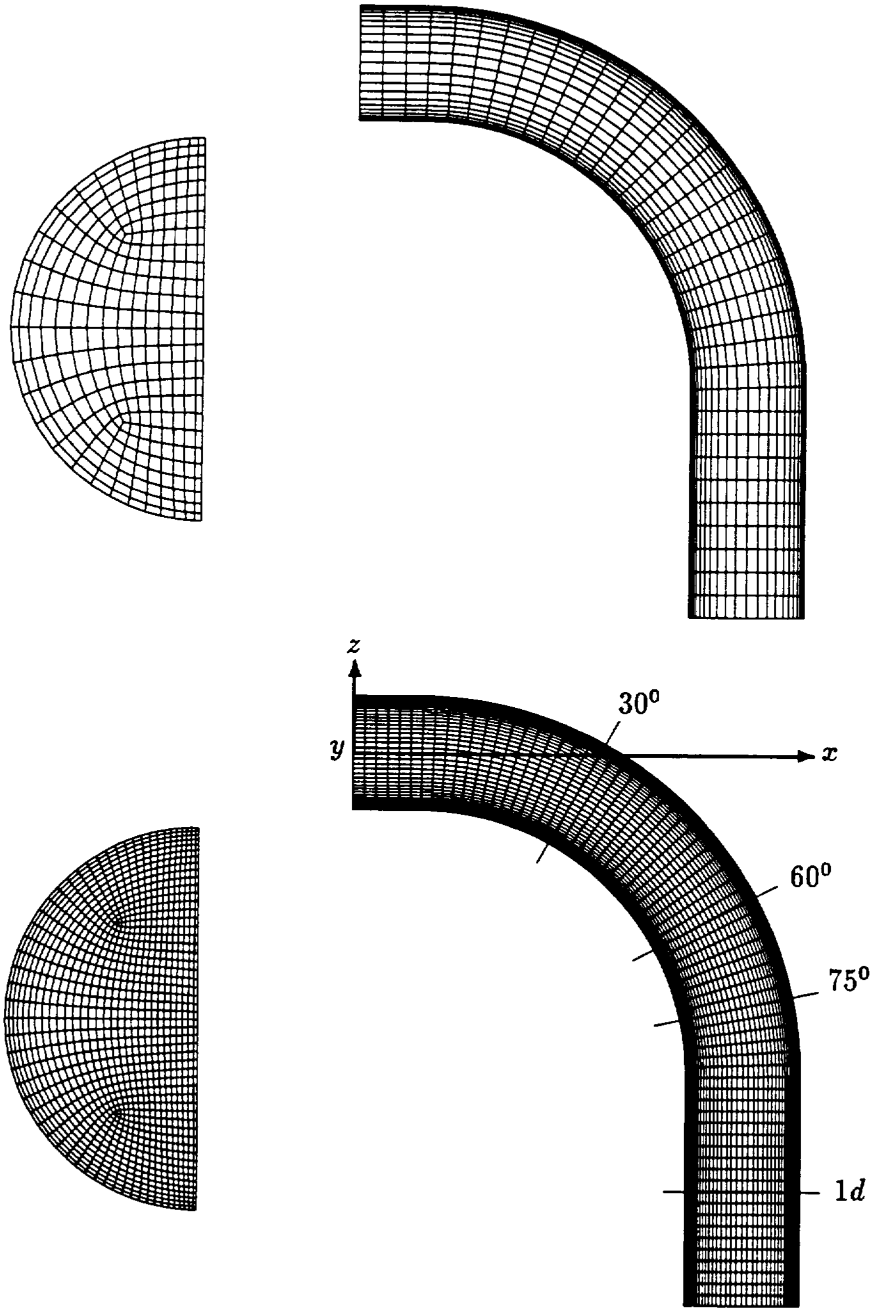


Figure 11.3: Coarse and fine grids front and side views (320 × 36 nodes and 1223 × 71 nodes) with particular cross-sections.

0.1%) and a multiplication of the corrections by a factor 1/4 are used in order to limit the perturbations to the Eulerian solution. The streamlines are recomputed each 50 iterations of the Eulerian solver. In order to prevent the formation of strong gradients in vorticity, a Lagrangian pseudo-diffusion coefficient of 2% is used.

The Enayet data set provides streamwise velocity measurements for a turbulent flow case at a Reynolds number of 43000 and a flow averaged velocity $\overline{u}$ of 0.92 m/s. The fluid kinematic viscosity is 0.804×10^{-6} m^2/s. The stations at which the experimental data are available are located at 30^0, 60^0 and 75^0 along the bend and at 1 diameter downstream of the bend exit section as indicated in Figure 11.3.

11.4.1 Inlet velocity profile definition

The measured inlet velocities on horizontal and vertical traverses at a distance of 0.58 diameter upstream of the bend inlet section are splined in the circumferential and radial directions to give an inlet boundary condition for the calculation. The slight asymmetry in the inlet velocity measurements has been averaged out for this calculation. The inlet velocity profile is not a fully developed profile due to the proximity of the bend but the assumption of a fully developed turbulent flow is used here for the determination of the slip velocity at the wall. The procedure is identical to the one described in Chapter 8. The coefficient of friction, shear stress and friction velocity take the values

$$\lambda = 0.022, \quad \tau_w = 0.00233, \quad u_\tau = 0.0482. \tag{11.1}$$

Using a value of $y^+ = 30$ or $u^+ = 14$ at the edge of the laminar sublayer, the slip velocity u_{slip} is

$$u_{slip} = u^+ \, u_\tau = 0.675, \quad \text{or} \quad \frac{u_{slip}}{\overline{u}} = 0.730. \tag{11.2}$$

The distance from the wall at which the last inlet velocity measurement is taken is equal to 10% of the pipe radius. Since the first grid point out from from the wall is located in between the last experimental value and the wall, its velocity has to be fitted

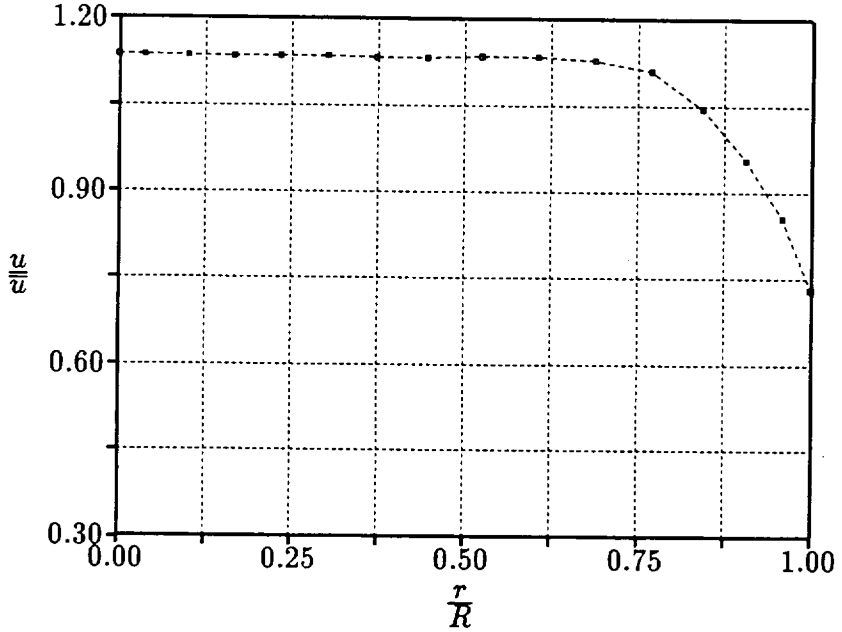


Figure 11.4: Inlet velocity profile at grid nodes.

on the velocity profile. The velocity profile is represented at the grid nodes location in Figure 11.4.

In order to be able to compare the Eulerian solution on a coarse and on a fine grid, both calculations must have the same inlet boundary condition. When discretizing an analytical velocity profile onto a coarse grid, the inlet vorticity is lower than the vorticity obtained on a fine grid. This problem is illustrated in Figure 11.5 for the case of an inlet velocity profile of boundary-layer type. To ensure that the vorticity values between grid nodes is identical between the two cases, the velocity profile is modified when using the fine grid according to the arrows in Figure 11.5. The inlet vorticity value is particularly important here since it is used as a component of the state-vector in the Eulerian/Lagrangian solution and also as a measure of the numerical diffusion of the solution when comparing Eulerian or Eulerian/Lagrangian results on a coarse grid to an Eulerian result on a fine grid.

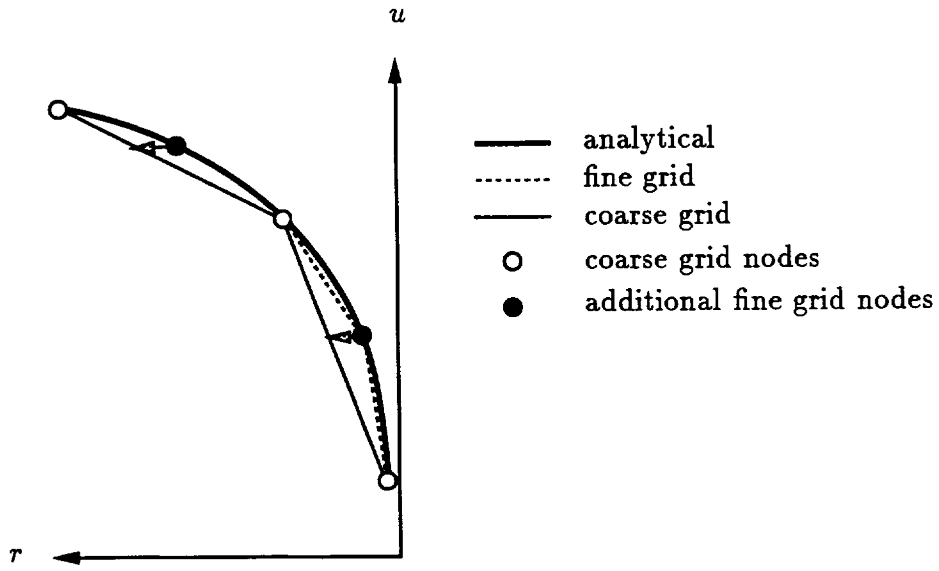


Figure 11.5: Velocity profile defined on coarse and fine grids and adjustment of the velocity profile for the fine grid case.

11.4.2 Enayet case 90^0 bend: Eulerian and Eulerian/Lagrangian results

This section presents the secondary flow solution for the Eulerian solution computed on the coarse grid, the Eulerian/Lagrangian solution on the coarse grid and an Eulerian solution on the fine grid. The three solutions are compared with experiments.

The streamwise velocity contours and the cross-flow velocity vectors are shown for the four stations along the bend on Figures 11.6, 11.7, 11.8 for the Eulerian solution on the coarse grid, the Eulerian/Lagrangian solution and the Eulerian solution on the fine grid, respectively. The experimental streamwise velocity contours taken from [21] are represented on Figure 11.9.

The cross-flow velocity vectors indicate a weak secondary flow region is present along the pipe wall at the 30^0 station. On further stations, this region intensifies, moves towards the inside of the bend, and then moves along the pipe symmetry surface as the vortical region is entrained due to the the proximity of the wall and the symmetry surface.

Generally, the Eulerian solution on the fine grid and the Eulerian/Lagrangian solution on the coarse grid lead to a stronger secondary flow than the Eulerian solution on the coarse grid. The presence of the pipe wall and of the counter-rotating vortex entrains the vorticity around the pipe wall and then along the symmetry surface. Because of the difference in the secondary flow intensity, the entrainment speed and the location of the vortical regions is also different in the three cases. Therefore, the numerical diffusion affects both the strength and the location of the secondary flow regions. Using the Eulerian solution on the coarse or the fine grid results in very different vortical flow location at the $1d$ station as can be seen on Figure 11.6 and 11.8. With the Eulerian solution on the coarse grid, the secondary flow region is weaker due to numerical diffusion and the vortical region moves more slowly along the wall and the symmetry plane. When using the Eulerian/Lagrangian solution on the coarse grid, the vortical regions move faster than with the Eulerian solution alone and are closer to the Eulerian solution on the fine grid predictions showing that the combined scheme is successfully at reducing diffusion errors.

Not only is the strength and location of the vorticity field affected by the numerical diffusion for the Eulerian solution on the coarse grid, but the structure of the vortical field is also affected. With the Eulerian solution on the fine grid, the main secondary vortex present on the $1d$ station is accompanied by two smaller counter-rotating vortices located near the inside of the bend. With the Eulerian solution on the coarse grid, the two small vortices are smeared out due to numerical diffusion. With the Lagrangian correction, the two small vortices are recognizable at the $1d$ station indicating that the structure of the secondary flow is closer to the one predicted by the fine grid Eulerian solution.

For each of the three solutions, the low streamwise velocity regions are associated with the strong cross-flow regions so that the location of the cross-flow in the experiments can be estimated by looking at the experimental velocity contours on Figure 11.9. For both the Eulerian/Lagrangian solution on the coarse grid and the Eulerian solution on the fine grid, the predicted speed of entrainment of the cross-flow seems to be too large indicating that the neglected viscous effects near the pipe wall are important. The Eulerian/Lagrangian solution shows a good agreement with the Eulerian solution on the fine grid. In particular, Figure 11.7 shows how the gradients in axial and cross-flow velocities near the pipe wall and the symmetry plane are better represented when using the Eulerian/Lagrangian solution.

Nevertheless, in the Eulerian solution on the coarse grid, the location of the cross-flow is in better agreement with the experiments because the numerical diffusion plays a role in decreasing the strength of the cross-flow and hence its entrainment speed. The numerical diffusion helps the solution when comparing the solution with experiments in terms of cross-flow strength and location. However, this is a misleading result since the numerical diffusion and the real viscous diffusion are two distinct phenomena. Indeed the

Eulerian solution shows strongly diffused streamwise velocity contours, a result which does not agree with the experimental streamwise velocity. Therefore the numerical diffusion of the flow is a spurious computational effect which has to be minimized.

Figures 11.10, 11.11 and 11.12 show the streamwise vorticity contours on the four stations for the Eulerian solution on the coarse grid, the Eulerian/Lagrangian solution and the Eulerian solution on the fine grid, respectively. The effect of the numerical smoothing is clearly visible in Figure 11.10 (Eulerian solution on the coarse grid) by the curving of the streamwise vorticity contours near the wall of the pipe. This effect is reduced for the solution on the fine grid in Figure 11.12. The use of the combined Eulerian/Lagrangian scheme also results in the correction of the diffusion effect near the pipe wall as seen in Figure 11.11.

Nevertheless, the Eulerian/Lagrangian solution on the coarse grid still suffers from lack of grid resolution when compared to the Eulerian solution on the fine grid. The effectiveness of the Eulerian/Lagrangian technique is limited because the flow features are too small to be accurately captured on the coarse grid. If the Eulerian solution on the fine mesh is taken as the representation of the inviscid non-diffused solution one can see that the secondary vorticity gradients to be captured are very high and that the effectiveness of a solution on a grid twice as coarse is limited by a grid resolution issue. This is the case when the formed secondary vorticity zone moves along the pipe symmetry line.

Figure 11.13 shows the streamlines emerging from the inlet boundary-layer near the wall and wrapping around the vortex for the Eulerian solution on the coarse grid (a), the Eulerian/Lagrangian solution (b) and the Eulerian solution on the fine grid (c). In the Eulerian solution on the coarse grid, because of the numerical diffusion, the vortex is formed further downstream compared to the two other solutions and the rate of rotation around the vortex is lower. The Eulerian/Lagrangian solution predicts the vortex formation even earlier than the Eulerian solution on the fine grid. This is because in the first part of the pipe, the Lagrangian correction scheme is more effective than the fine grid Eulerian solution in preventing numerical diffusion. This is not true, however, in the second part of the pipe where the vorticity is very concentrated and where the Lagrangian correction effectiveness is limited by the grid resolution issue mentioned earlier.

Figure 11.14 compares the pressure contours for the three solutions on the half-pipe symmetry surface. The negative streamwise pressure gradient on the inner wall is followed by a pressure recovery. The trace of the secondary vortex on the symmetry surface is clearly visible in the fine grid Eulerian solution. Up to 2/3 of the pipe length, the Eulerian/Lagrangian solution shows a good agreement with the fine grid Eulerian solution.

Figure 11.15 shows the circulation around pipe cross-sections as a function of their axial distance downstream s (measured on the pipe axis and normalized by the pipe radius R) for the Eulerian solution on the coarse grid (a), the Eulerian/Lagrangian solution (b), and the Eulerian solution on the fine grid (c). The condition of constant circulation around pipe cross-sections does not apply since the contours defined around pipe cross-sections do not correspond to the convection of a material curve initially placed around the inlet cross-section. However, this allows for an integral value comparison between coarse and fine grid calculations. The circulation (normalized by the mass-flow averaged velocity $\overline{u}$ and the pipe radius) increases from the inlet until approximately the 70^0 station due to the creation of the secondary flow. With the solution on the coarse grid, the circulation increases more slowly due to the strong presence of numerical diffusion damping the strength of the secondary flow. The circulation computed with the Eulerian/Lagrangian scheme, however, agrees better with the fine grid Eulerian solution. The departure between the Eulerian/Lagrangian solution and the Eulerian solution on the fine grid increases after the 60^0 station, approximately. This is an indication of the numerical diffusion occurring on the finer mesh. The upstream influence of the bend is shown by the increase in circulation for the straight pipe segment placed before the bend.

An interesting circulation contour to assess the numerical diffusion of the secondary flow is a material curve surrounding the exit cross-section and convecting 'upstream' until approximately the 90^0 cross-section location. The value of the circulation should remain constant while the material curve is convecting. Again, the circulations for the three types of calculations are plotted in Figure 11.16. The symbol s now stands for the average convection distance from the exit cross-section. The numerical diffusion is seen as an increase of the circulation when the material curve convects 'upstream'. Also because of the numerical diffusion the Eulerian solutions start from lower values of circulation and the two circulations increase as the material curves convect upstream. In comparison, the Eulerian/Lagrangian solution shows little change in the circulation. Still the curve for the Eulerian solution on the fine grid shows a substantial amount of loss in circulation. This integral value plot indicates that very fine grid Eulerian calculations are indeed required for these particular vortical flows in order to get a truly non-diffused solution.

The convergence towards a solution where the numerical diffusion is small results in an increase of the entrainement speed of the secondary vortex. The predicted speed of the vortical regions for the Eulerian/Lagrangian solution and the Eulerian solution on the fine grid are too large when comparing the computed results to the streamwise velocity measurements. The difference is accounted for by the effects of the real fluid viscosity effects. An attempt to take these effects into account is described in the next section.

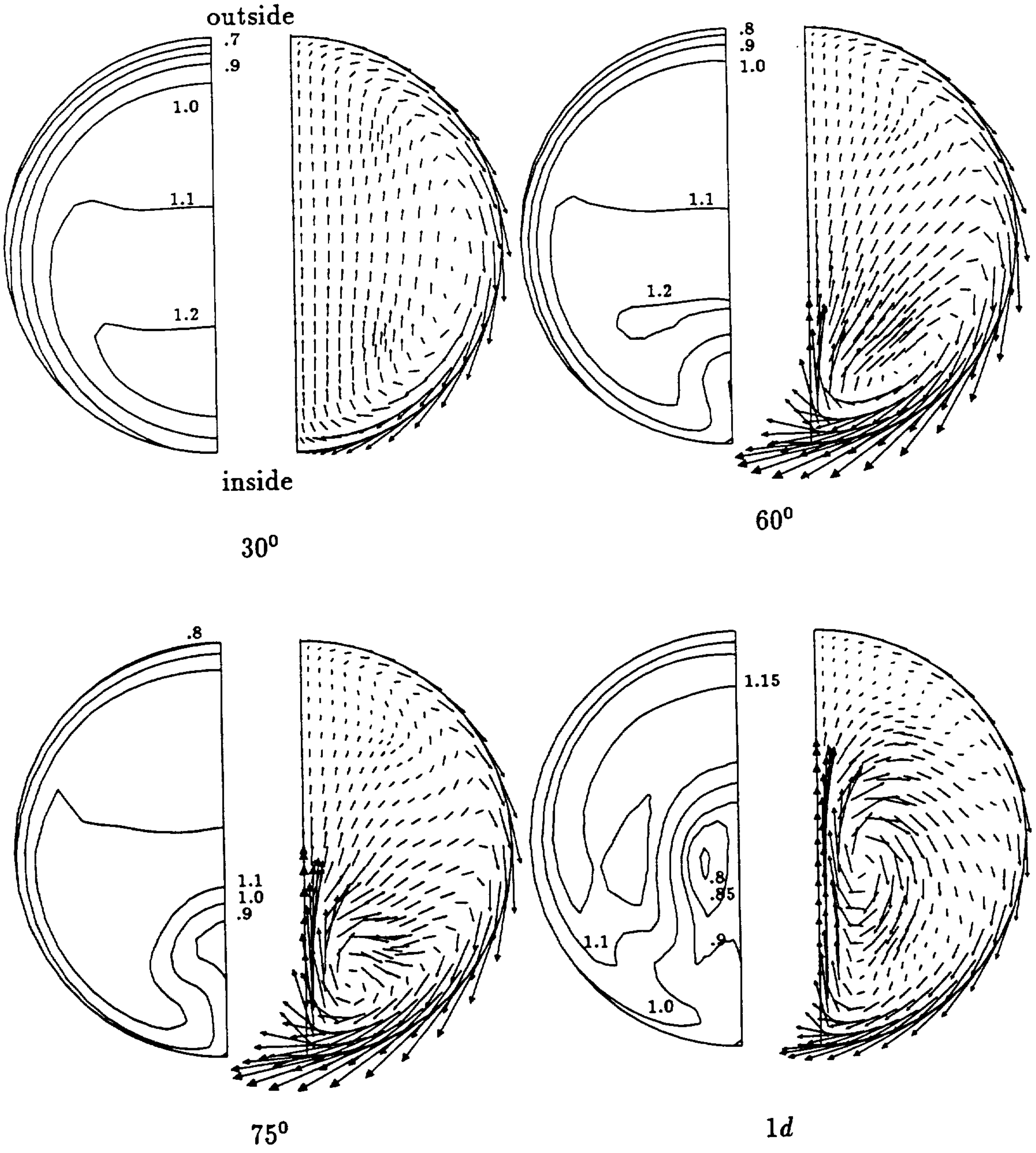


Figure 11.6: Streamwise velocity contours and cross-flow velocity vectors for four stations along the bend using Eulerian scheme on coarse grid.

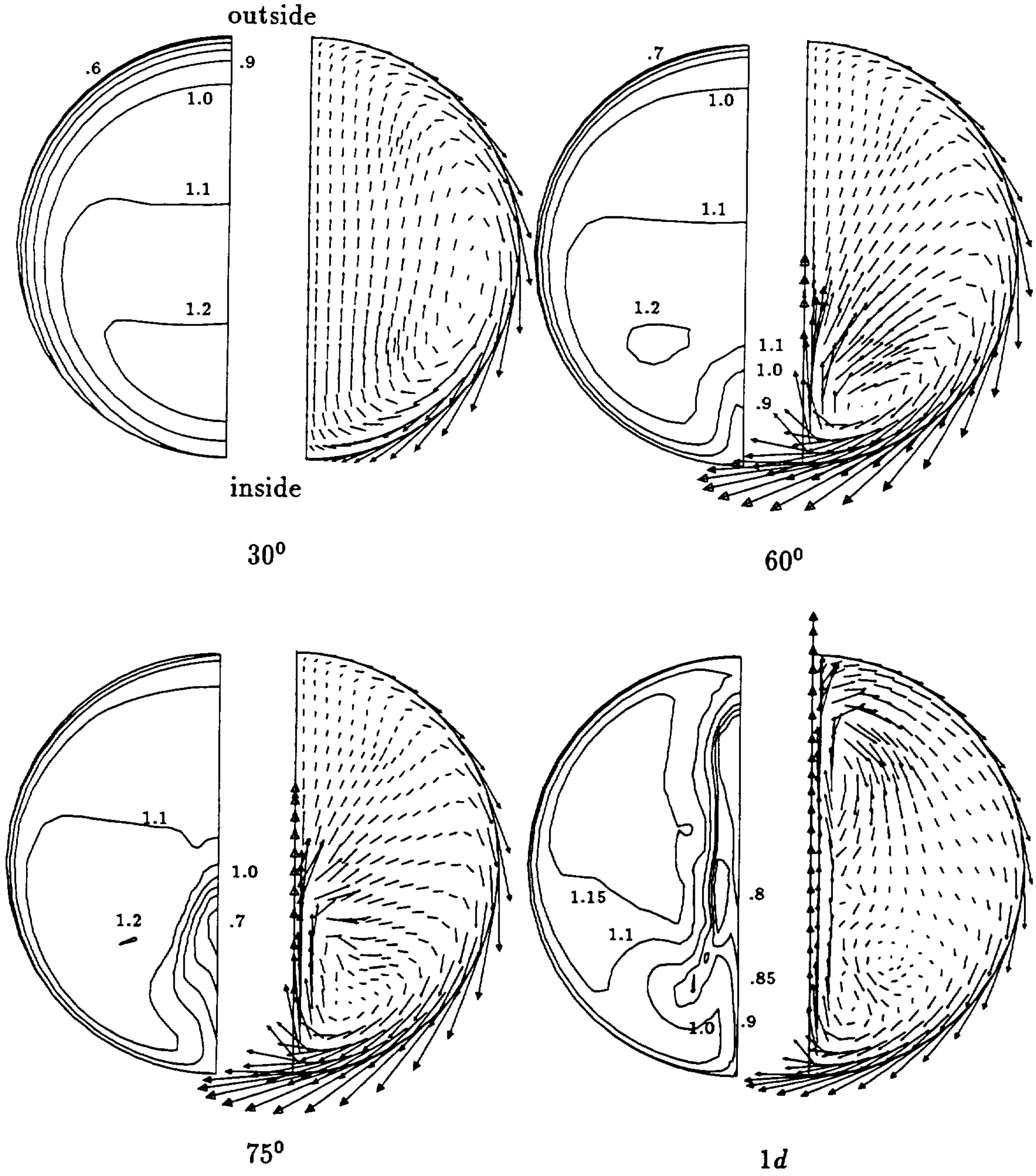


Figure 11.7: Streamwise velocity contours and cross-flow velocity vectors for four stations along the bend using Eulerian/Lagrangian scheme on coarse grid.

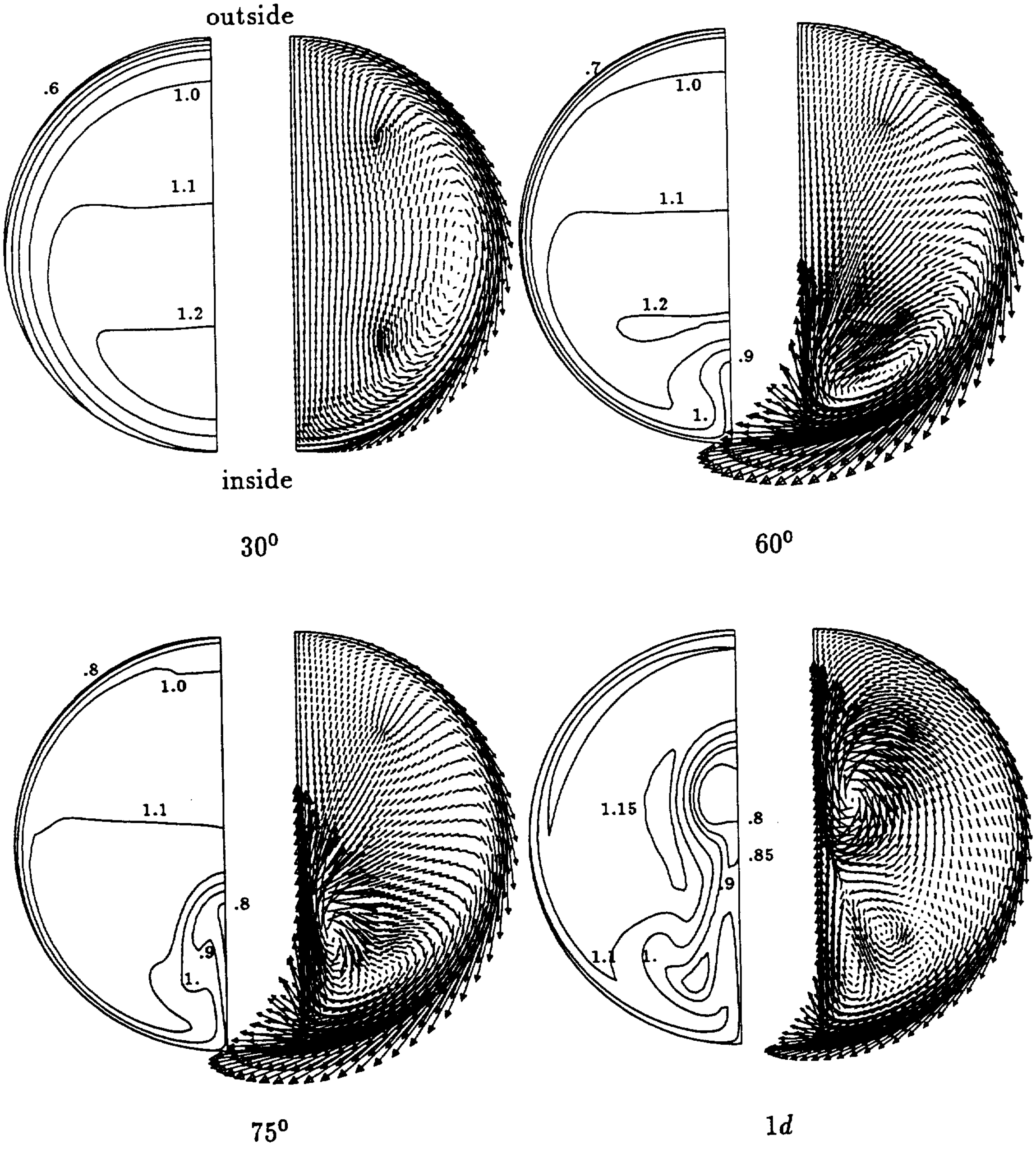


Figure 11.8: Streamwise velocity contours and cross-flow velocity vectors for four stations along the bend using Eulerian scheme on fine grid.

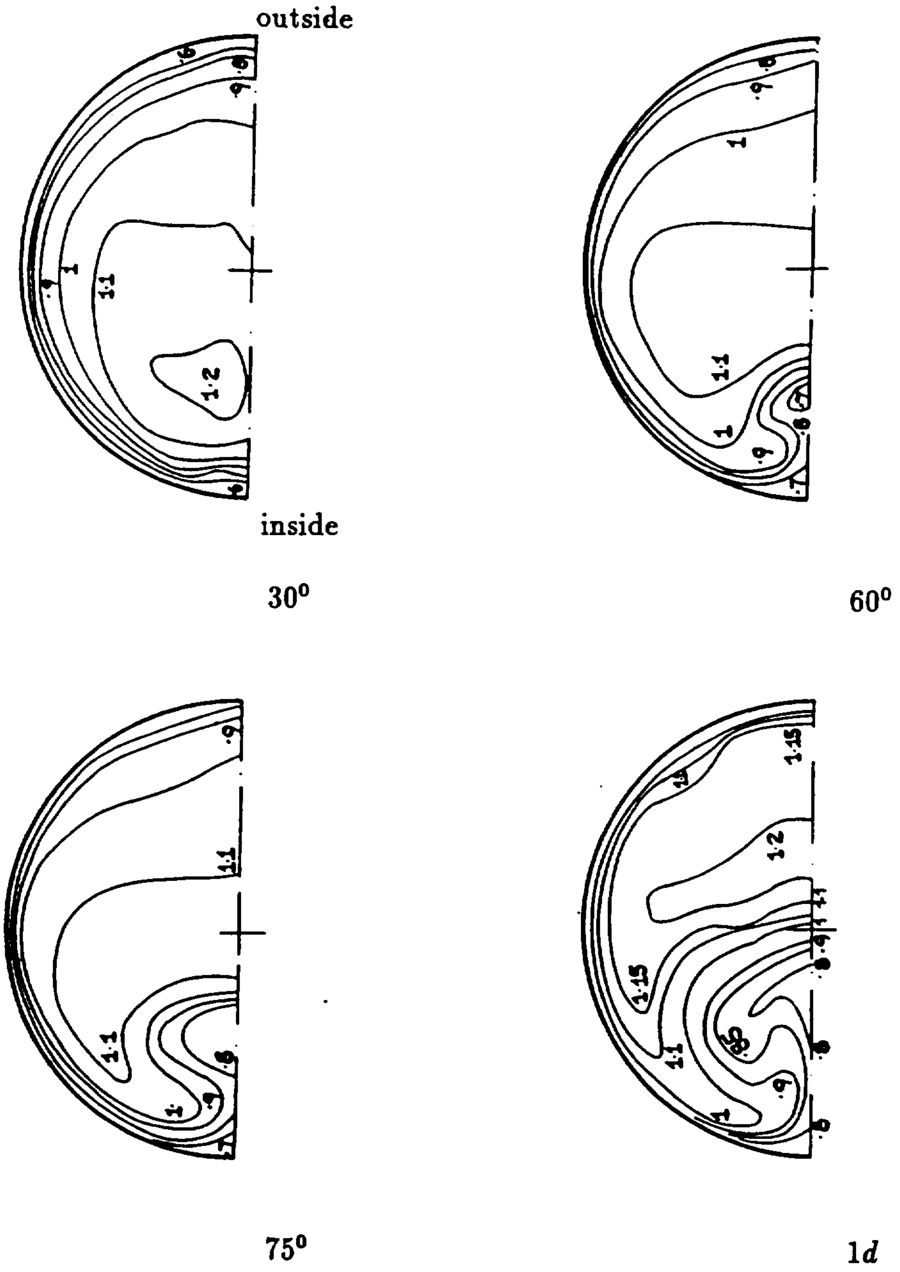


Figure 11.9: Experimental streamwise velocity contours for four stations along the bend.

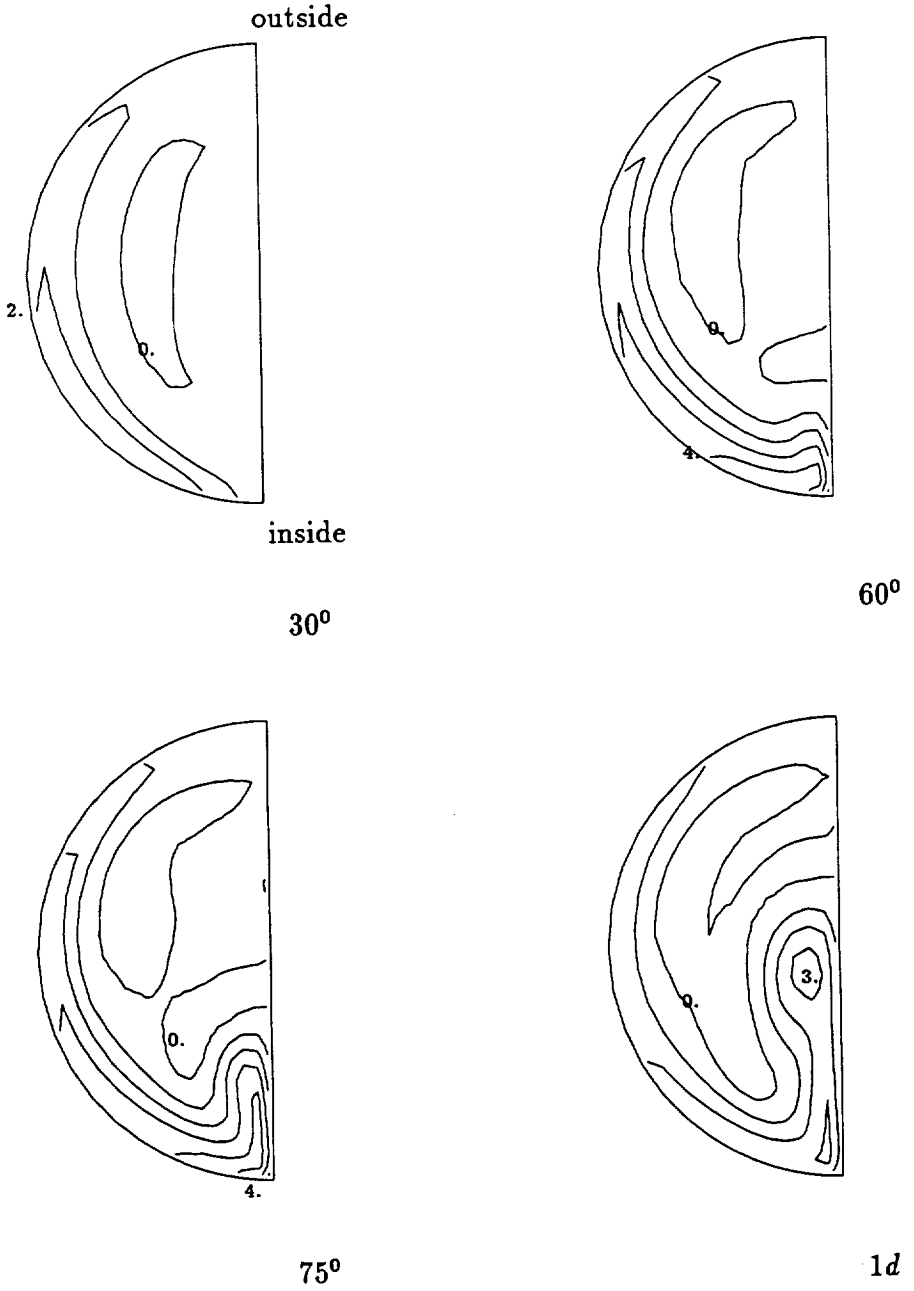


Figure 11.10: Streamwise vorticity contours (increment = 1.0) for four stations along the bend using Eulerian scheme on coarse grid.

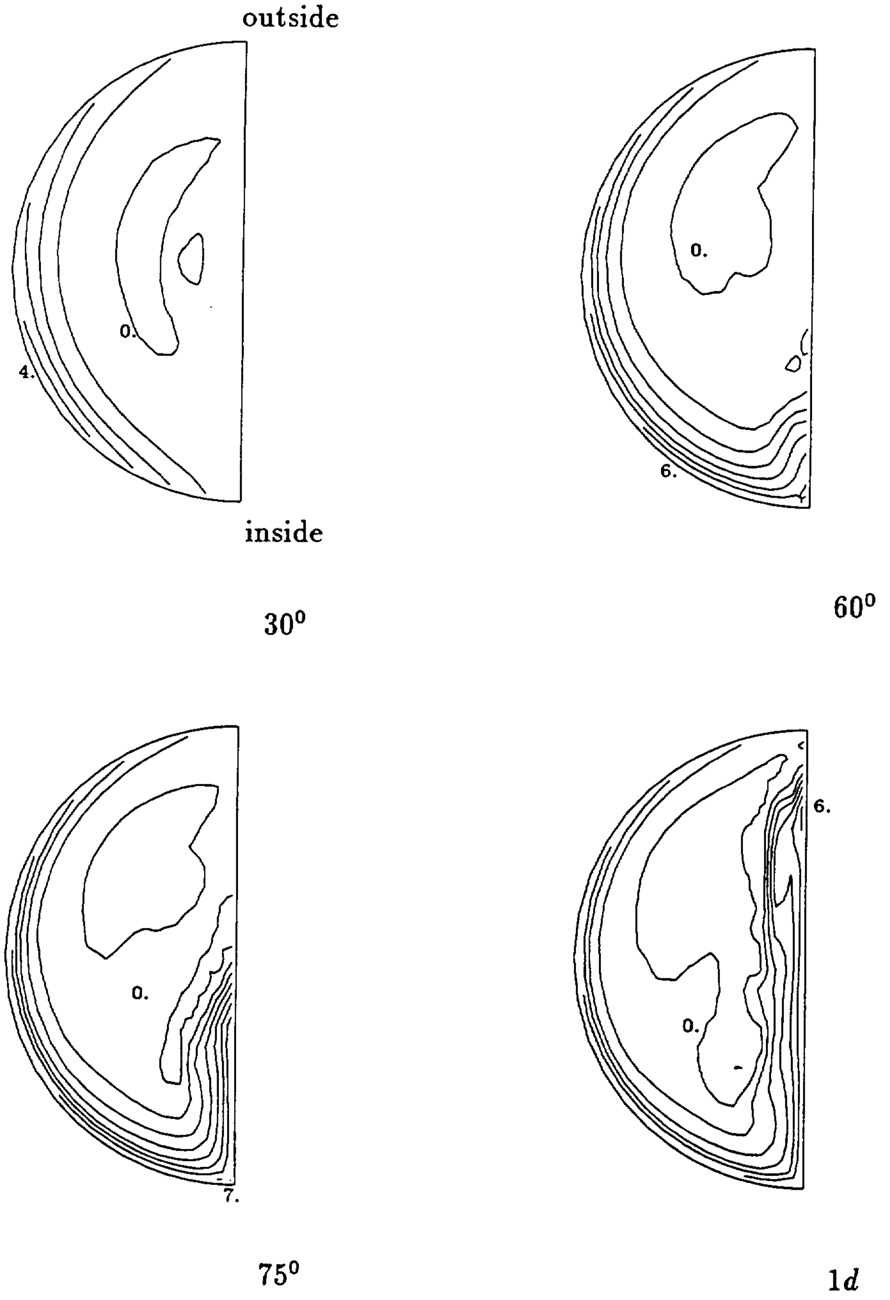
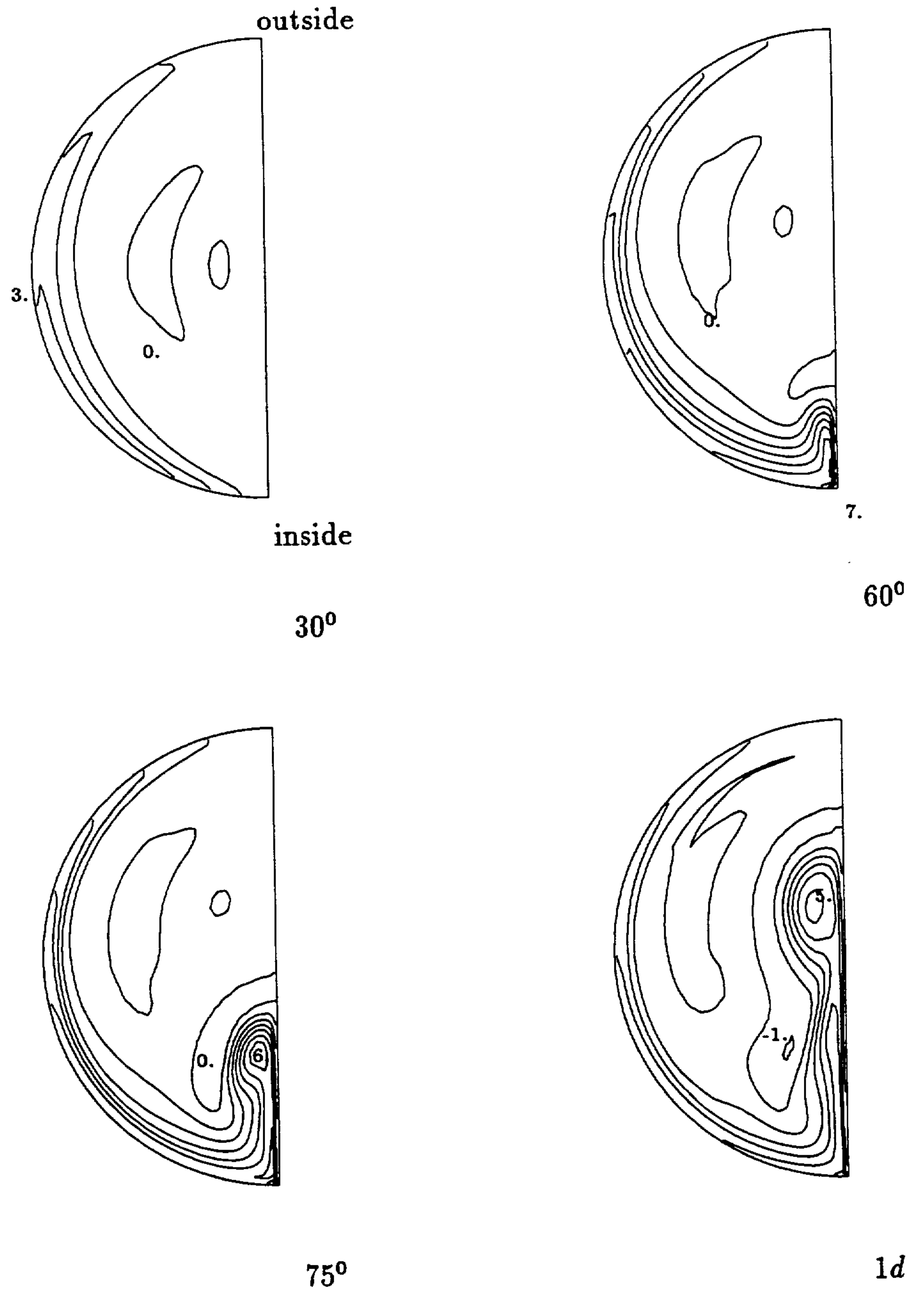


Figure 11.11: Streamwise vorticity contours (increment = 1.0) for four stations along the bend using Eulerian/Lagrangian scheme on coarse grid.



30^0

60^0

75^0

1d

Figure 11.12: Streamwise vorticity contours (increment = 1.0) for four stations along the bend using Eulerian scheme on fine grid.

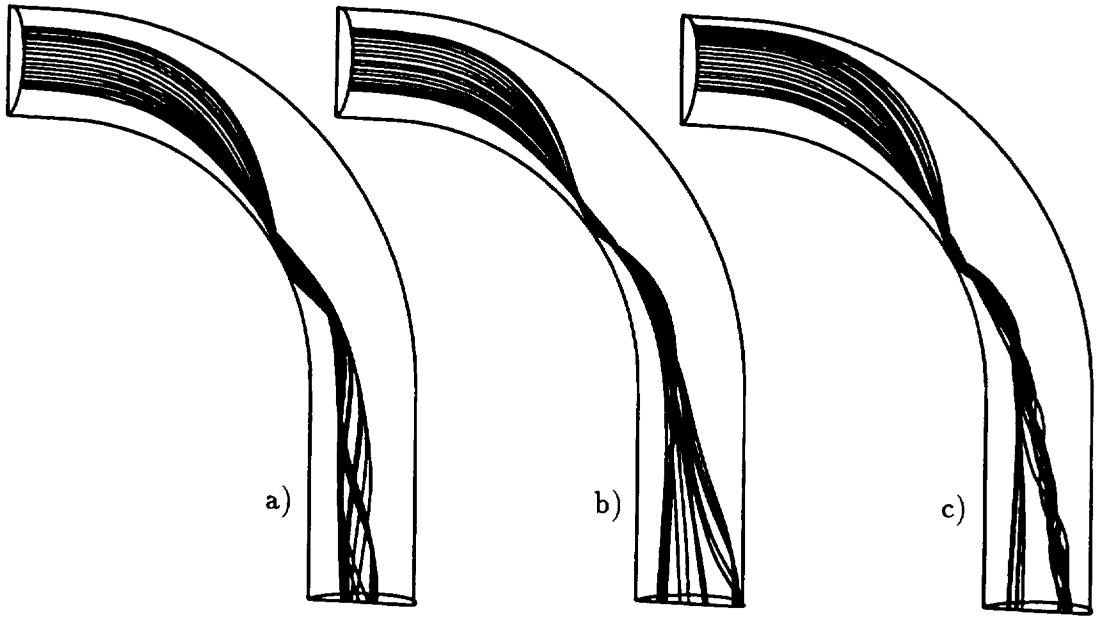


Figure 11.13: Streamlines emerging from the near wall region at the inlet of the pipe and forming the secondary flow: a) Eulerian solution, b) Eulerian/Lagrangian solution, c) Eulerian solution on fine grid.

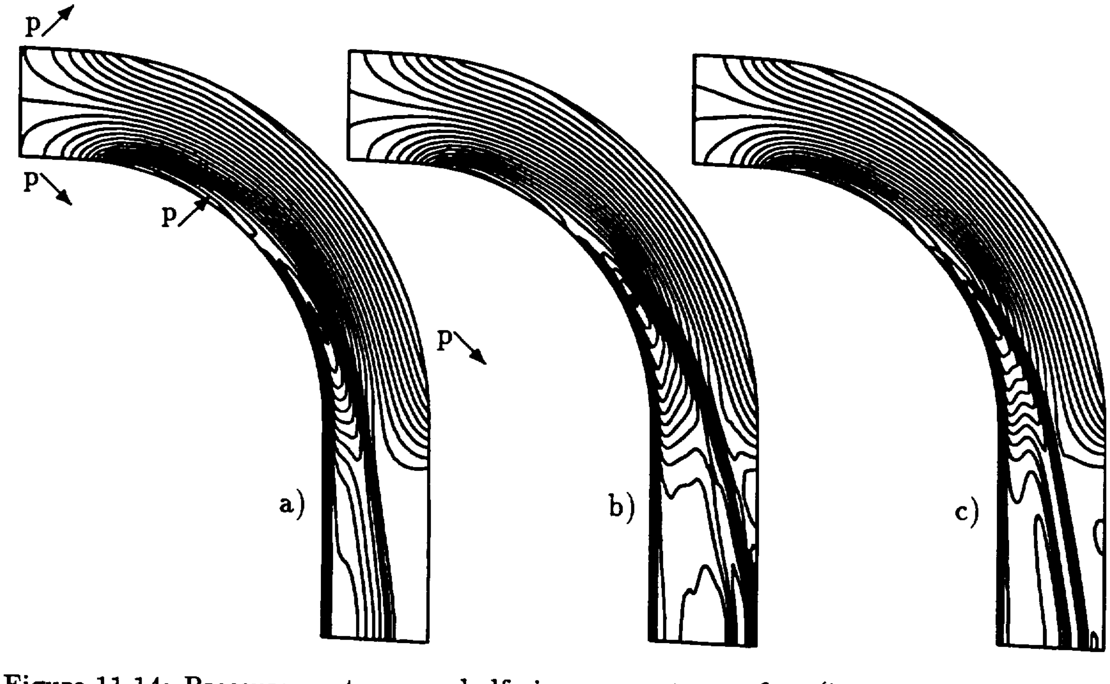


Figure 11.14: Pressure contours on half-pipe symmetry surface (increment = 0.02): a) Eulerian solution, b) Eulerian/Lagrangian solution, c) Eulerian solution on fine grid.

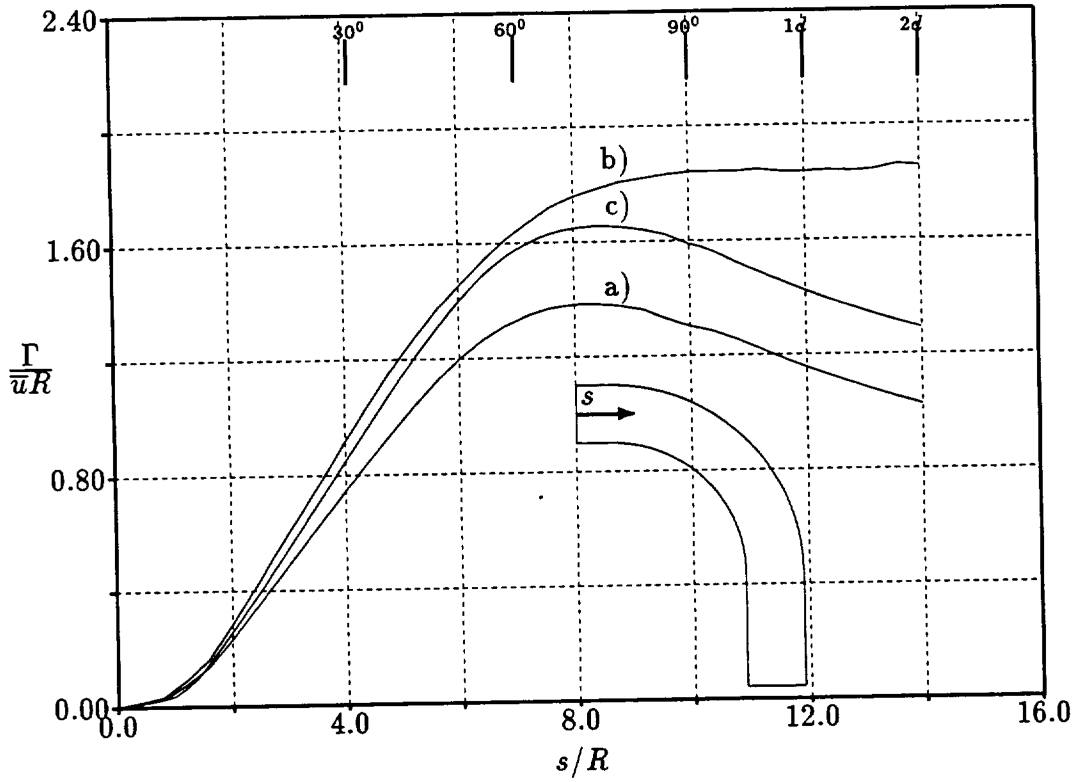


Figure 11.15: Circulation around pipe cross-sections, a) Eulerian solution, b) Eulerian/Lagrangian solution, c) Eulerian solution on fine grid.

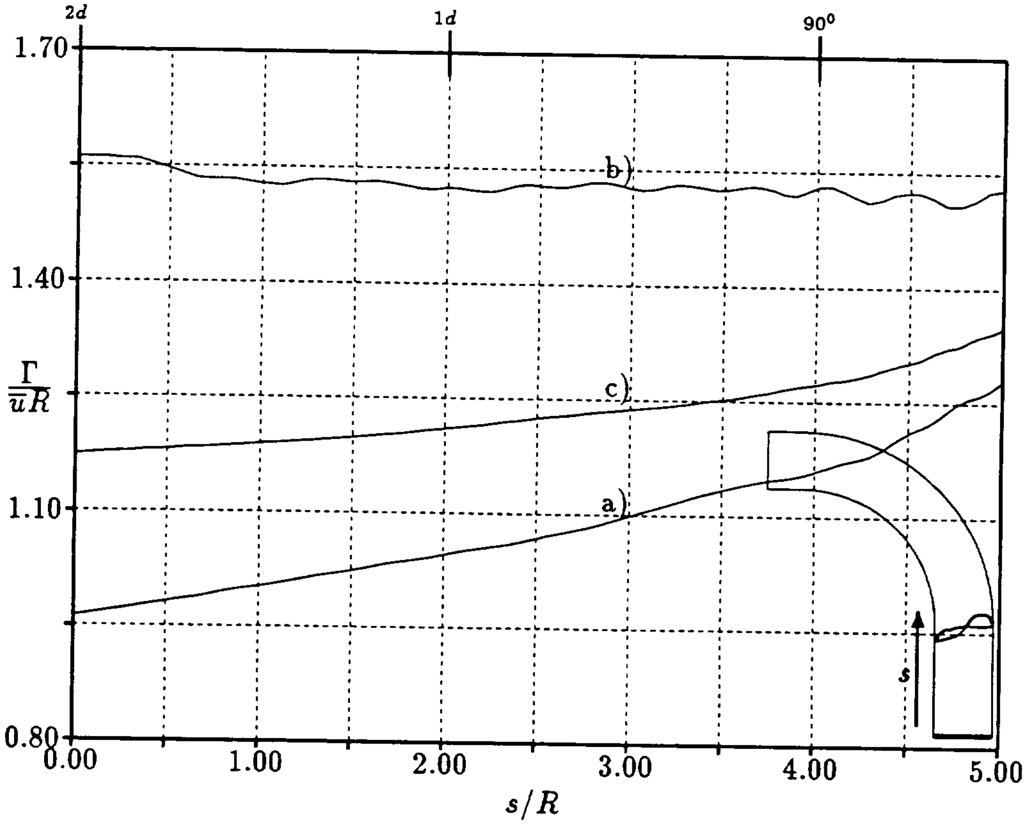


Figure 11.16: Circulation around a closed convecting curve: a) Eulerian solution, b) Eulerian/Lagrangian solution, c) Eulerian solution on fine grid.

11.4.3 Law of the wall correction

As mentioned in the previous section, the streamwise velocity measurements are indicative that the inviscid solution of the flow predicts a speed of entrainment of the secondary flow which is too large. While the Eulerian solution predicts the fundamental flow features, the effect of the fluid viscosity is to decrease substantially the strength of the secondary flow and therefore its entrainment speed. The effects of the fluid viscosity were briefly described in Chapter 11. This section presents an attempt at accounting the shear effects near the wall of the pipe as part of the flow solution.

In order to account for the viscous phenomenon occurring at the wall a simple 'law of the wall' is imposed on each wall node. The log-layer region of the boundary-layer is in equilibrium, i.e. it does not depend on the pressure gradient imposed upon it. Under the assumption that the first grid point out from the wall is embedded in the log-layer, the wall velocity depends only of the velocity value at the first neighboring node since the log-layer is described by a universal distribution law. The correction at a wall node consists then in scaling the velocity by using the same log-law relation than for the inlet velocity profile described in Chapter 8. For each wall node, the magnitude of the velocity at the first grid point out from the wall is used to determine the value of the friction velocity u^* by implicitly solving the universal velocity distribution law given in Equation (8.2). Each component of the slip velocity at the wall is then scaled so that the resulting velocity magnitude is equal to the velocity magnitude at the end of the laminar sublayer at a u^+ value of 14 (this value is identical to the one used for the inlet velocity profile definition).

Figure 11.17 shows the streamwise velocity contours and the cross-flow velocity vectors on the four stations along the pipe for the Eulerian/Lagrangian solution on the coarse grid with the addition of the simple wall velocity correction. When using the Eulerian/Lagrangian scheme, the markers in the cells near the pipe walls are omitted since the wall velocity correction procedure already determines a vorticity value in the cell.

The streamwise velocity contours are now in better agreement with the experimental values drawn in Figure 11.9. The addition of the wall correction leads to a reduction of cross-flow velocities near the pipe wall and symmetry surface. This means that the mechanism of near wall diffusion is indeed a factor to take into account when simulating the flow field. The agreement is particularly good at the 75^0 station. After this station, the main vortex departs from the near wall region and travels along the symmetry surface. At the 1d station, the vortex seems to be located higher than in the experiments showing the absence of numerical diffusion modeling in the interior of the flow field.

In general, the agreement is not as good when the viscous effects are not taken into account as shown in the previous section. However, the simple 'law of the wall correction' suffers from the absence of viscous diffusion out from the wall regions. Also, the assessment of the quality of the solutions proves to be difficult since the present experiments do not provide any direct information on the cross-flow itself.

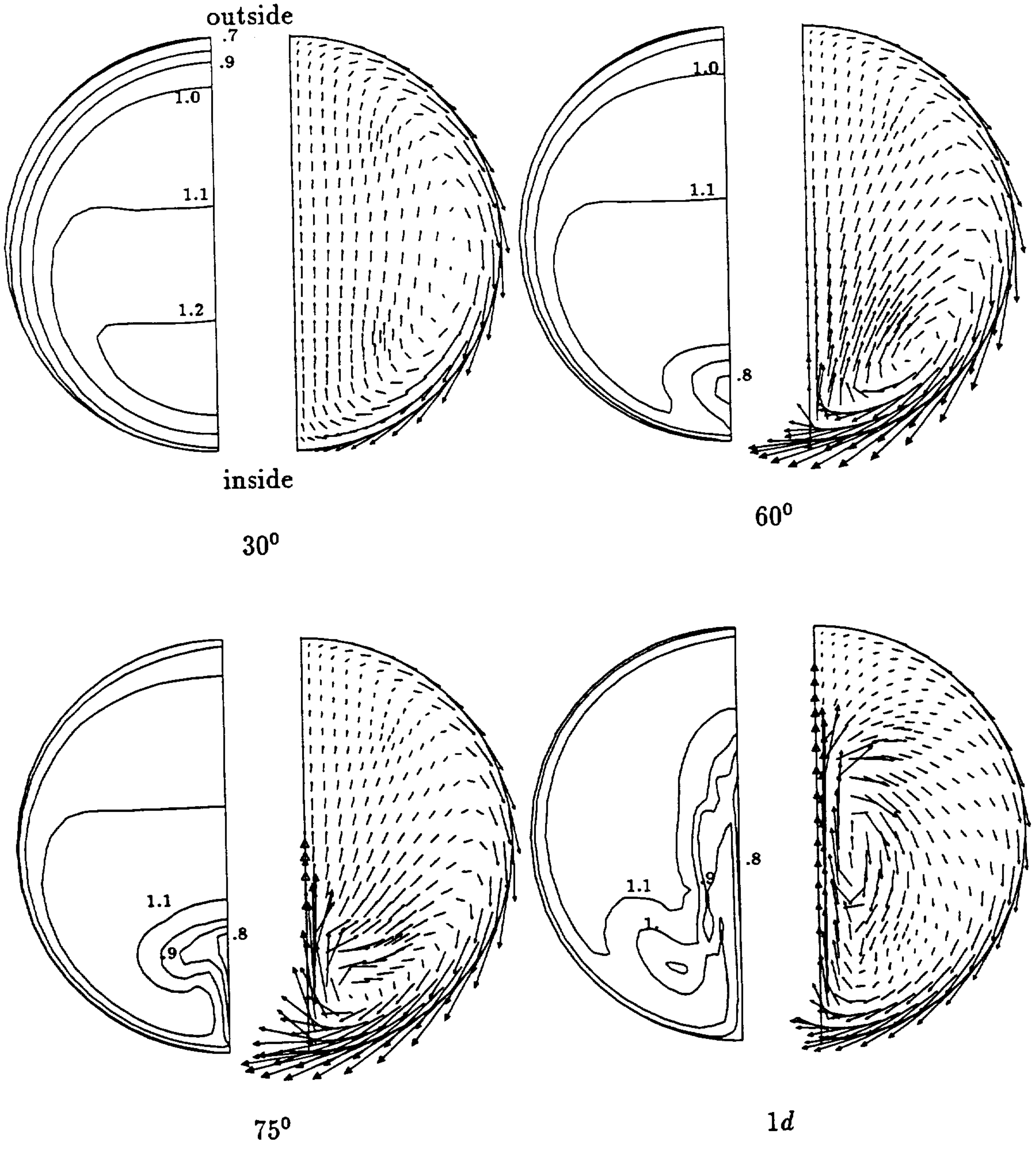


Figure 11.17: Streamwise velocity contours and cross-flow velocity vectors for four stations along the bend using Eulerian/Lagrangian scheme on coarse grid and wall correction.

The Enayet data set also provides pressure measurements along the wall of the pipe for 5 angles around the bend from the bend inside at 0^0 to the bend outside at 180^0 and for 6 stations along the bend located at $15^0, 30^0$, $60^0, 75^0$ and at 1 and 2 diameters downstream of the bend exit cross-section.

Figures 11.18 shows the computed pressure along the wall for angles around the bend at $0^0, 45^0$, $90^0, 135^0$ and 180^0 for the Eulerian solution on the coarse grid (a) and the Eulerian/Lagrangian solution (b). The pressure is non-dimensionalized by the use of a reference pressure p_{ref} (computed as the average pressure on the inlet cross-section) and a dynamic head based on the flow averaged velocity $\overline{u}$. The abscissa represents the curvilinear coordinate along the bend measured on the pipe axis. The measurements are indicated by different symbols for each angle around the bend. The measurements show a pressure loss of approximately $0.3\rho\overline{u}^2$ from inlet to exit. Since the Eulerian solver does not take into account any physical loss mechanism, the pressure is entirely recovered at a distance of 2 diameters downstream of the bend exit. The agreement between measurements and calculations is only qualitative. Both calculations overpredict the pressure for the $0^0, 45^0$ and 90^0 angles. The agreement is better at 135^0 but on the bend inside (at 180^0) the pressure recovery is much larger than in the experiments. In the experiments, the pressure recovery region on the bend inside corresponds to an increase in the boundary-layer thickness neglected in the Eulerian calculations.

Figure 11.19 shows the computed pressure along the wall when the wall velocity correction is used in the Eulerian/Lagrangian solution. The loss mechanism at the wall has generally little effect on the pressure distribution. While the agreement has improved from a solution without wall correction, the coincidence with experiments remains only qualitative.

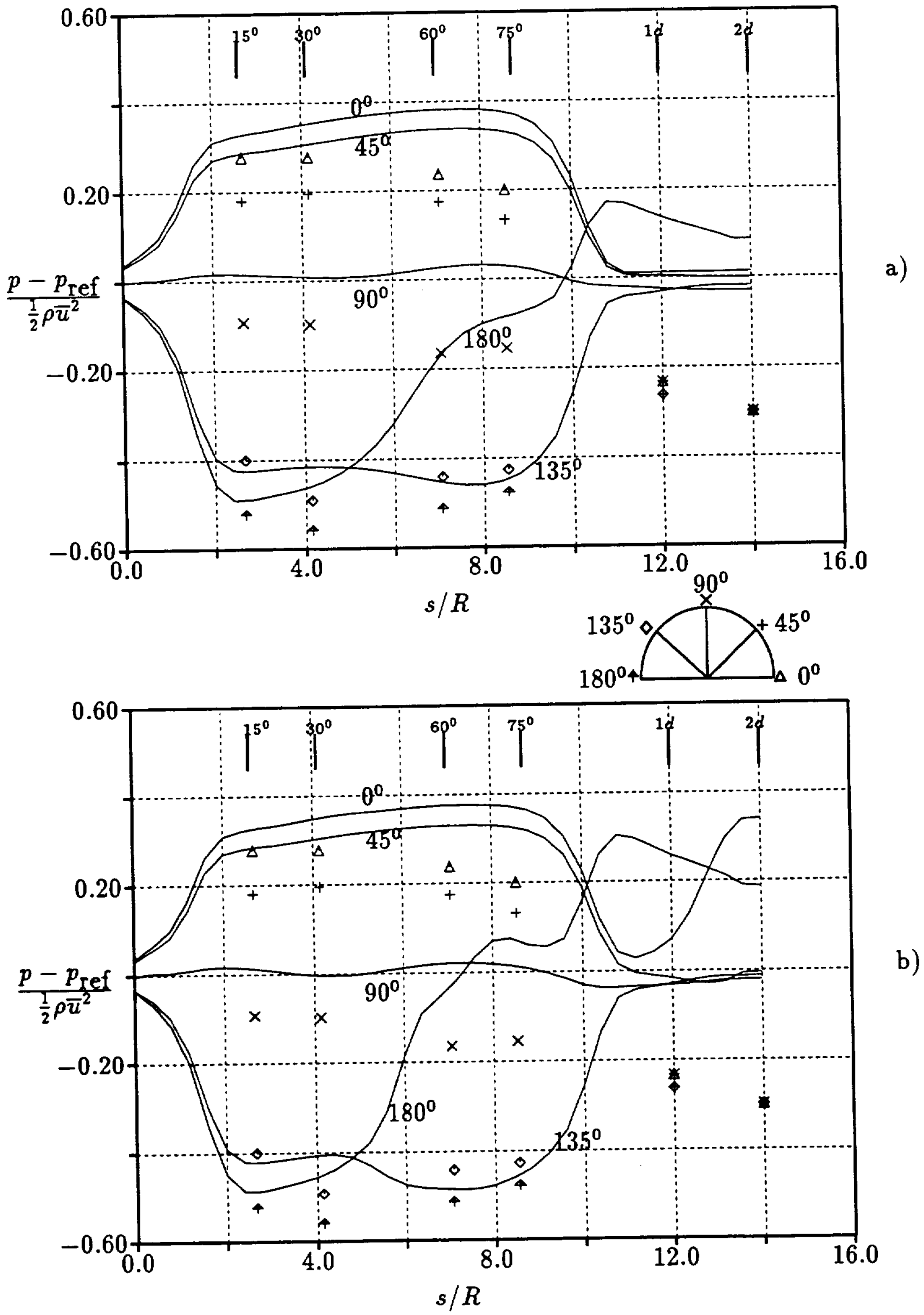


Figure 11.18: Wall static pressure variation at four angles around the bend: a) Eulerian solution, b) Eulerian/Lagrangian solution (symbols indicate experimental values).

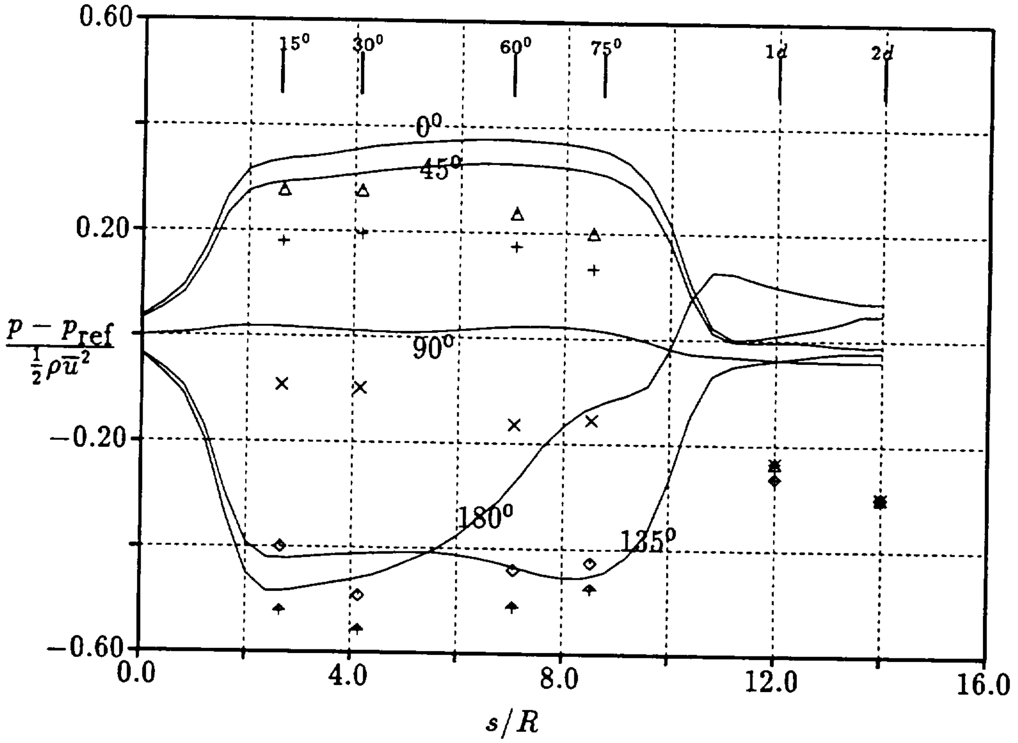


Figure 11.19: Wall static pressure variation at four angles around the bend: Eulerian/Lagrangian solution with wall velocity correction (symbols indicate experimental values).

11.5 GTL 90^0 bend case

The second test geometry has been tested at the Gas Turbine Laboratory of MIT [6]. Streamwise velocity measurements are provided at a station located at 1.61 diameters downstream of the bend exit cross-section.

The diameter of the pipe is 0.1023 m. and the ratio of radius of curvature to pipe diameter is 1.61. The computations were performed on a coarse and a fine grid with 320×51 nodes and 1223×101 nodes, respectively. The grids front and side views are represented on Figure 11.20. The grid cross-sections are spaced evenly in the streamwise direction along the bend centerline. The test geometry extends 1.61 diameters upstream and downstream of the bend exit cross-section. The upstream distance is required for the influence of the bend to be negligible at the inlet cross-section since the inlet velocity profile is specified as a fully developed flow.

The flow Reynolds number is 3.32×10^5 and the flow averaged velocity is 2.61 m/s. Using the procedure described in Chapter 8, the coefficient of friction, wall shear stress and friction velocity take the values

$$\lambda = 0.0142, \quad \tau_w = 0.0121, \quad u_\tau = 0.1101. \tag{11.3}$$

Using a value of $y^+ = 30$ or $u^+ = 14$ at the edge of the laminar sublayer, the slip velocity u_{slip} is

$$u_{slip} = u^+ \, u_\tau = 1.541, \quad \text{or} \quad \frac{u_{slip}}{\overline{u}} = 0.589. \tag{11.4}$$

The upstream integration of the streamlines is used in the Eulerian/Lagrangian technique. The correction step is applied only once at each Eulerian iteration. Both an under-relaxation of the correction ($R_f = 0.1\%$) and a multiplication of the corrections by a factor 1/4 are used in order to limit the perturbations to the Eulerian solution. The streamlines are recomputed every 50 iterations of the Eulerian solver. In order to prevent the formation of strong gradients in vorticity, a Lagrangian pseudo-diffusion

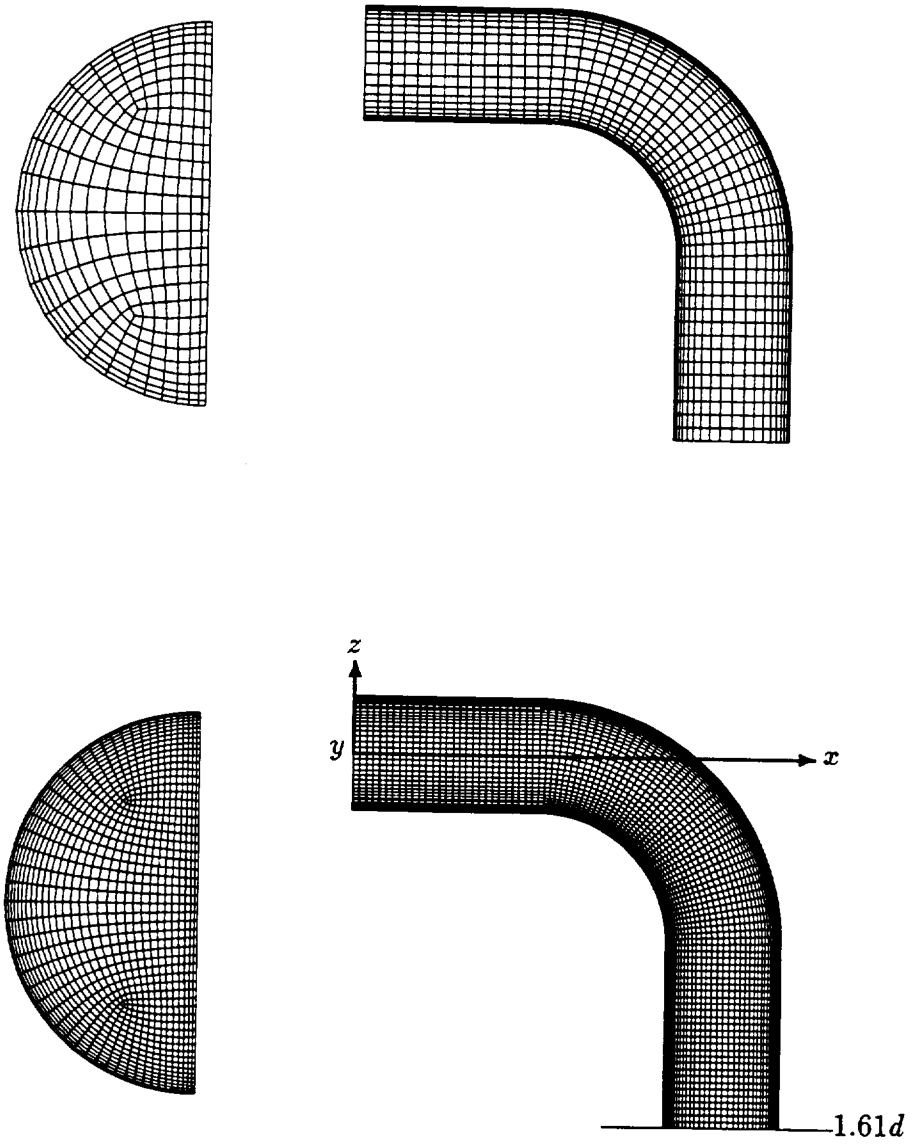


Figure 11.20: Coarse and fine grids front and side views (320 × 51 nodes and 1223 × 101 nodes) with measurement cross-section.

coefficient of 5% is used.

11.5.1 GTL case 90^0 bend: Eulerian and Eulerian/Lagrangian results

The streamwise velocity contours are shown for the station located at 1.61 diameter downstream of the bend exit in Figure 11.21 for the Eulerian solution on the coarse grid, the Eulerian/Lagrangian solution and the Eulerian solution on the fine grid, respectively. The experimental streamwise velocity contours are represented on Figure 11.21 d).

The three numerical solutions again predict an entrainment speed of the secondary flow which is too large. This is especially true of the fine grid Eulerian solution and the Eulerian/Lagrangian solution. The latter is overpredicting the secondary flow entrainment by the largest amount indicating that this solution is the one with the least numerical diffusion.

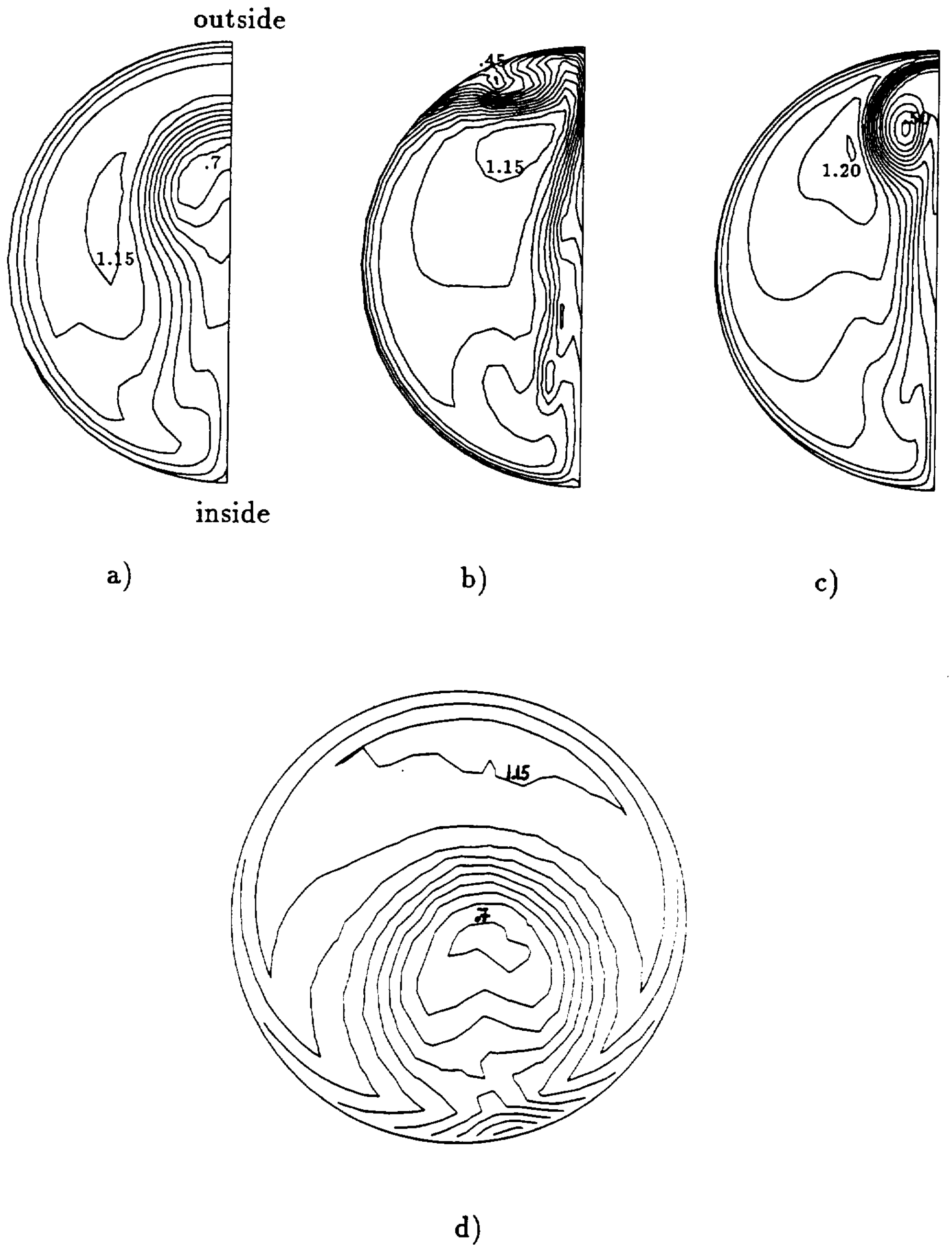


Figure 11.21: Contours of axial velocity at station located at 1.61 diameters downstream of bend exit, a) Eulerian solution, b) Eulerian/Lagrangian solution, c) Eulerian solution on fine grid, d) experiment (increment = 0.05).

11.6 Conclusions for the secondary flow in bends

The use of the Euler equations for the solution of the secondary flow in bent pipes has been motivated by the fact that viscous effects should appear in only limited regions of the flow. Indeed, the Euler solution of the problem has been shown to predict the fundamental features of the flow for the two bend cases treated here. However, the solution is largely dependent on the amount of numerical diffusion, even with the fine grid solution. By using the Lagrangian correction technique, the numerical diffusion on the coarse grid was reduced to a level below the one observed on the fine grid. Based on this 'true inviscid' solution, the assessment of the fluid viscosity effects can be assessed by comparing with experiments. One of the limitations of the method is the definition of a slip velocity at the wall.

Consistently with the findings of [19], [67], and [28], the inviscid solution of the flow has been shown to overpredict the development of the secondary flow. A law of the wall correction has been implemented to take the near wall viscous effects into account. The resulting streamwise velocity contours and cross-flow position compared better with the measurements. However, the pressure distribution along the wall was only in qualitative agreement with measurements. The concordance with the experiments also deteriorated when the secondary flow moved away from the wall since no viscous effects are taken into account apart from the wall regions. The law of the wall correction is also dependent on the chosen value of y^+ at the edge of the laminar sublayer. Clearly, this area of research requires more study. For example, combining an expression like the power law of Denton to a calibration of the value of y^+ over several experiments would yield more promising results.

Chapter 12

Weston Wing Case

The external flow over a three-dimensional wing tested by Weston [78] at the Langley Research Center is the object of this section. The wing geometry is characterized by a rectangular untwisted planform, a NACA0012 cross-section, a semi-span to chord ratio b/c of 3 and a body of revolution tip.

The Euler equations have been previously used by many authors to describe the flow around wings [45, 55, 36, 50, 51, 30, 9, 77]. These equations provide realistic solutions for these flows, since vorticity is captured as part of the solution and since the dynamics of the wake roll-up and convection is essentially inviscid. It is generally accepted that the artificial dissipation is the cause for the flow separation at sharp trailing edges. Moreover, as reported by Roberts [56], separation has been observed also on rounded wing tips, a phenomenon believed to be linked to grid resolution.

The freestream Mach number and angle of attack are 0.1425 and 8^0, respectively. The experimental data consists of wake and pressure coefficient measurements. The flow around the identical geometry has been computed by Roberts [56], on two grids of different topology, with a pressure coefficient on the wing in good agreement with the experiments. However, his calculations showed a spurious diffusion of the tip vortex behind the trailing edge when compared with experiments. This effect was identified as a numerical diffusion artifact.

Because of the low Mach number, the approximation of incompressible flow is used for the present computations. The Eulerian solution of the flow is first used for a validation of the incompressible flow solver by comparing the calculation to experiments. The Lagrangian correction is then applied here in order to counteract the numerical diffusion observed in the trailing vortex region.

The grid with a C-H structure is generated as described in Appendix A. The grid has a C structure at each spanwise cross-section and the wing surface is discretized by a H grid structure as shown in Figure 12.1. Figure 12.2 shows a detail of the definition of the wing tip body of revolution surface. Each streamwise grid surface in the wake has a H structure and presents a strong clustering in the wing tip vortex region as shown in Figure 12.3. This type of grid in the wake has been selected instead of the O-O type of grid used by Roberts because of the higher resolution of the wake. The mesh is slanted in the wake so that the high clustering region follows approximately the trailing vortex upward movement behind the trailing edge. The grid extends 2.3 chords downstream of the trailing edge and 2.5 chords away from the wing tip in the spanwise direction. The minimum distance from the wing to the grid outer surface is 2.5 chords. Since the farfield boundary conditions are based on the normal velocity component through the boundary, the external surface of the grid was inclined at an angle with respect to the freestream in order to get a non-zero normal velocity component.

The farfield boundary conditions use the theory developed in Section 3.4. On the farfield boundary, the freestream velocity at an incidence angle of 8^0 is corrected by the amount of velocity induced by the horseshoe vortex system.

The computed pressure coefficient on the wing surface is presented at five spanwise locations and compared with the experimental values of [78] in Figure 12.4. The computed pressure coefficient is defined as

$$Cp = \frac{p - p_\infty}{\frac{1}{2}\rho U_\infty^2}, \tag{12.1}$$

where the subscript $_\infty$ refers to the freestream values, is interpolated linearly from the grid nodes to each chosen spanwise location. Computed results and experiments are in

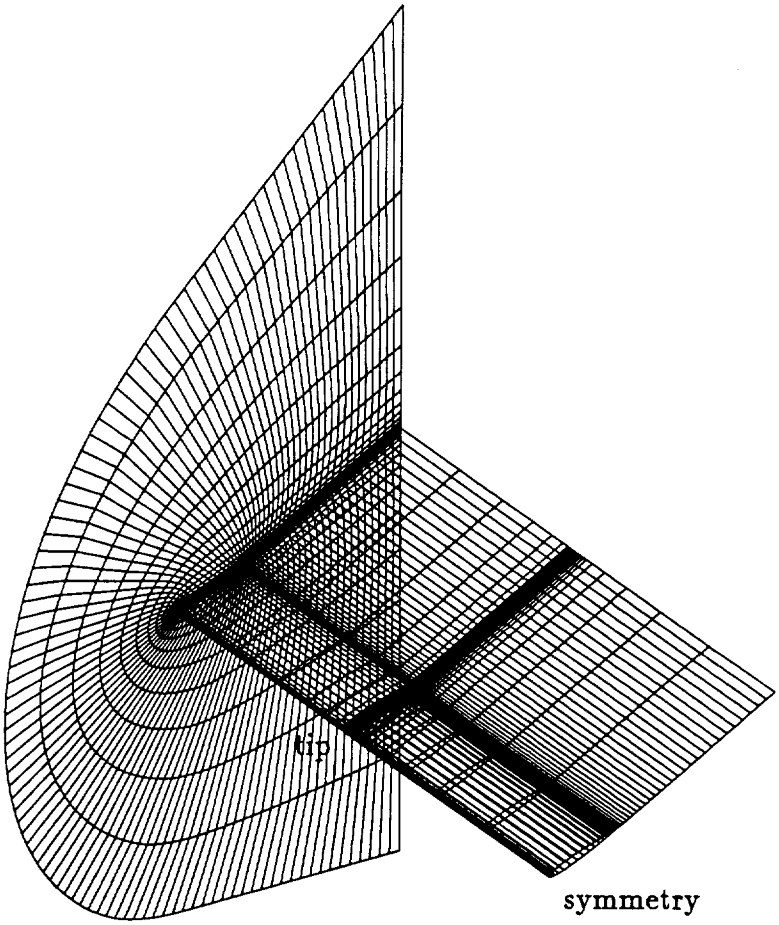


Figure 12.1: Weston wing grid 101 × 26 × 17 nodes with C-H structure shown by 2 mesh surfaces.

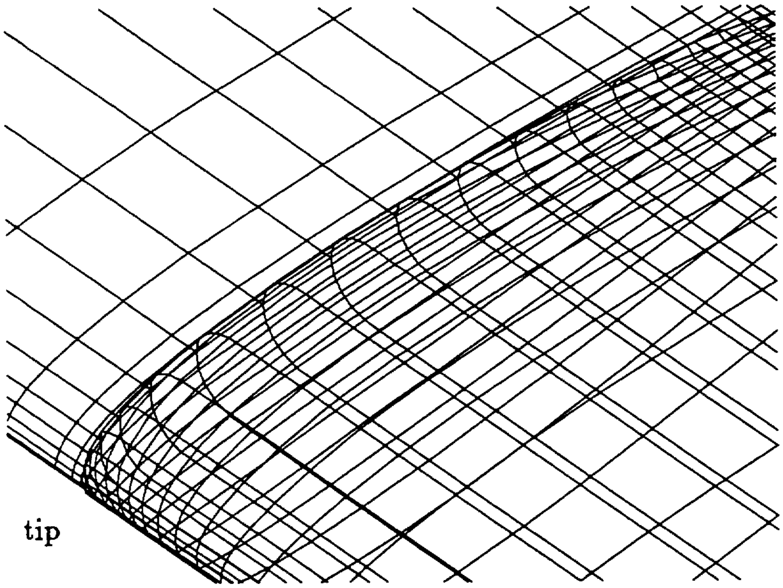


Figure 12.2: Weston wing grid detail of leading edge-tip region.

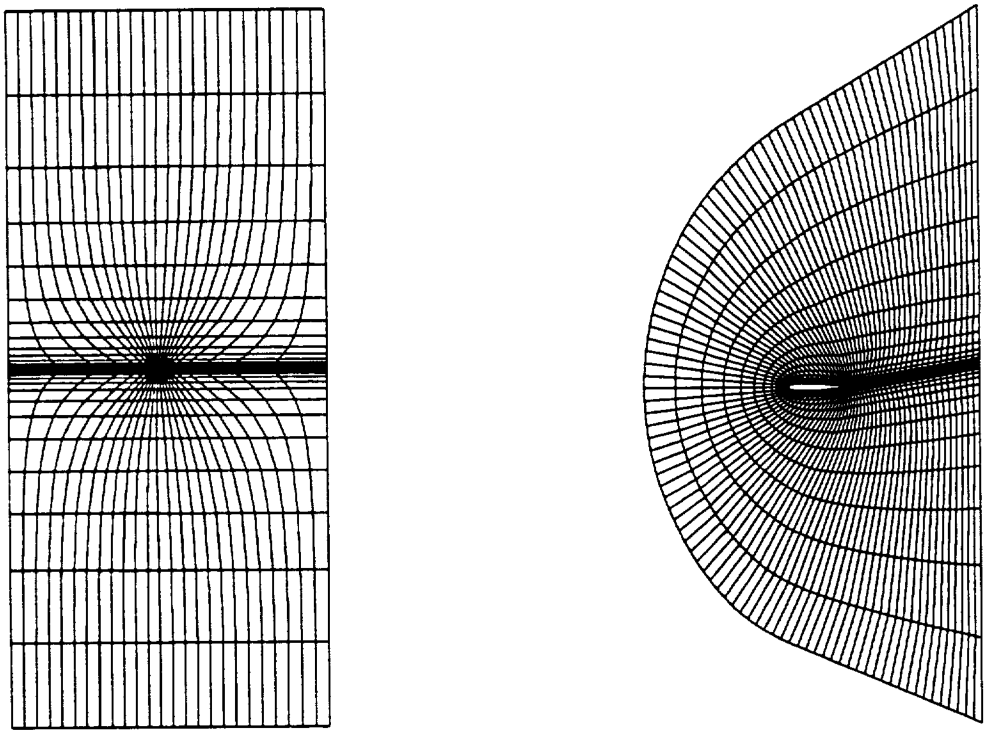


Figure 12.3: Exit surface with clustering near trailing vortex region and symmetry mesh surface showing wake surface angle behind trailing edge.

good agreement. In this calculation as in Roberts', the predicted leading-edge suction peak is higher near the wing root than over the rest of the wing. Roberts related this effect to the considerable tunnel flow angularity near the wing root. Also, the present solution gives lower leading-edge suction peaks near the wing root and higher near the wing tip compared to Roberts' solution. This can be attributed to a coarser grid resolution near the wing root and a better resolution of the tip geometry. Generally, Roberts' calculation and this calculation both lead to an underprediction of the suction peak over most of the wing, when comparing with the experiments. Again, this can be related to the flow angularity in the tunnel. However, the underprediction near the wing root is more pronounced in the present calculation since the wing resolution is approximately four times coarser than in the solution of Roberts. The grid resolution on the wing was traded for more grid resolution in the wake when the C-H grid type was selected instead of the O-O type.

A singular grid line extends from the wing tip to the outer surface corresponding to a locally lower solution accuracy and accounting partially for the larger discrepancies between experiments and calculations in this region. The deterioration of the computed solution near the wing tip was also reported by Roberts and the local solution in the tip region was shown to be very sensitive to the details of the grid and wing geometry.

The Lagrangian correction technique is applied on the Eulerian solution by placing a marker at the center of the cells in the vicinity of trailing vortex and by integrating the streamlines backward towards the trailing edge. An under-relaxation factor $R_f = 1\%$ and a multiplication of the corrections by a factor 1% are used in order to limit the perturbations to the Eulerian solver. The correction step is applied once each Eulerian iteration. The integration of the streamlines is repeated each 5 iterations of the Eulerian solver. In contrast with the computation of the flow in pipes of Chapter 11 where the vorticity is spread over a large portion of the computational domain, the Lagrangian correction technique shows its potential in this case since the markers need to be located in only a small region of the flow and the CPU time allotted to the Lagrangian correction remains correspondingly low.

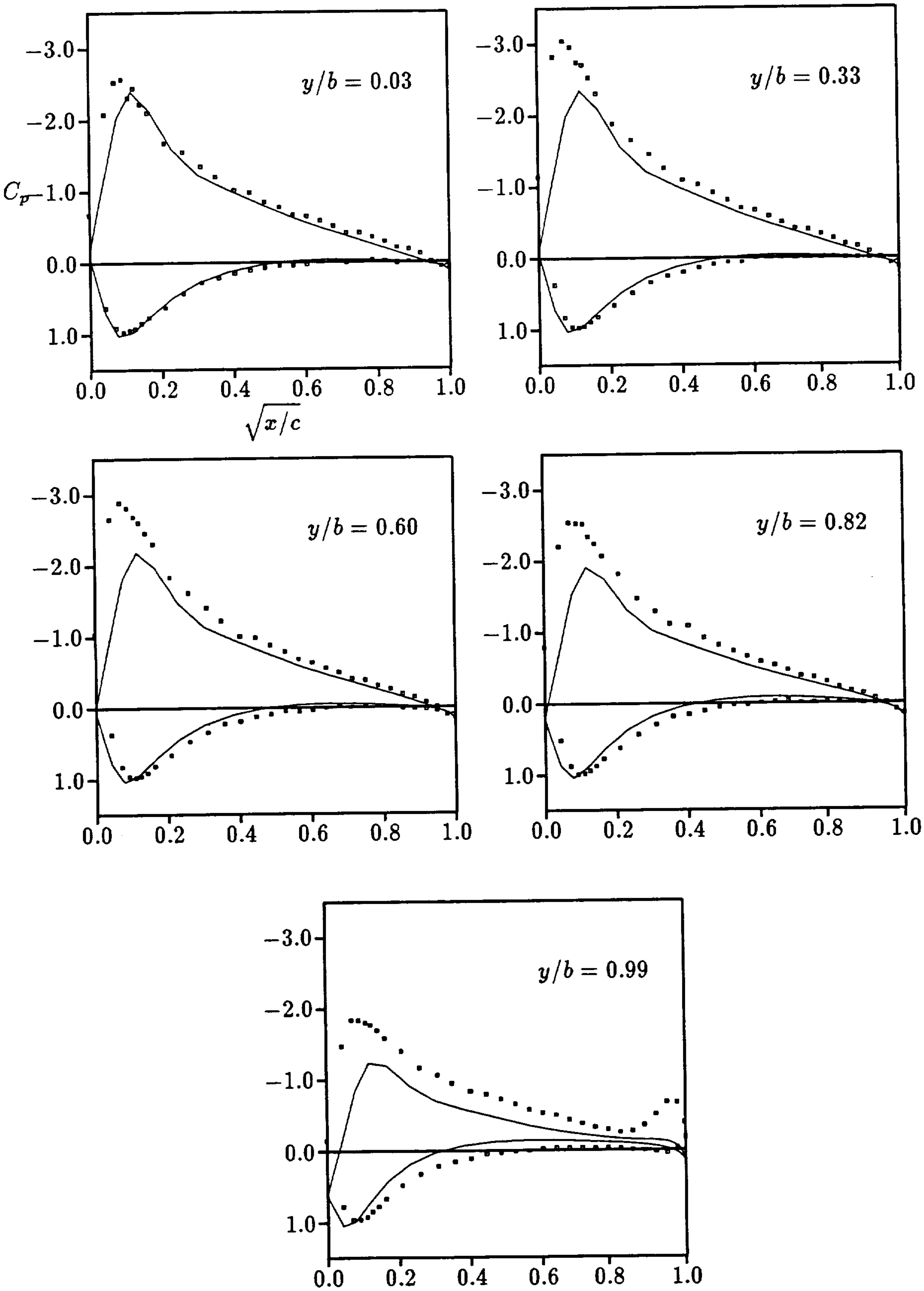


Figure 12.4: Computed pressure coefficient on wing surface compared with experiments at five spanwise locations.

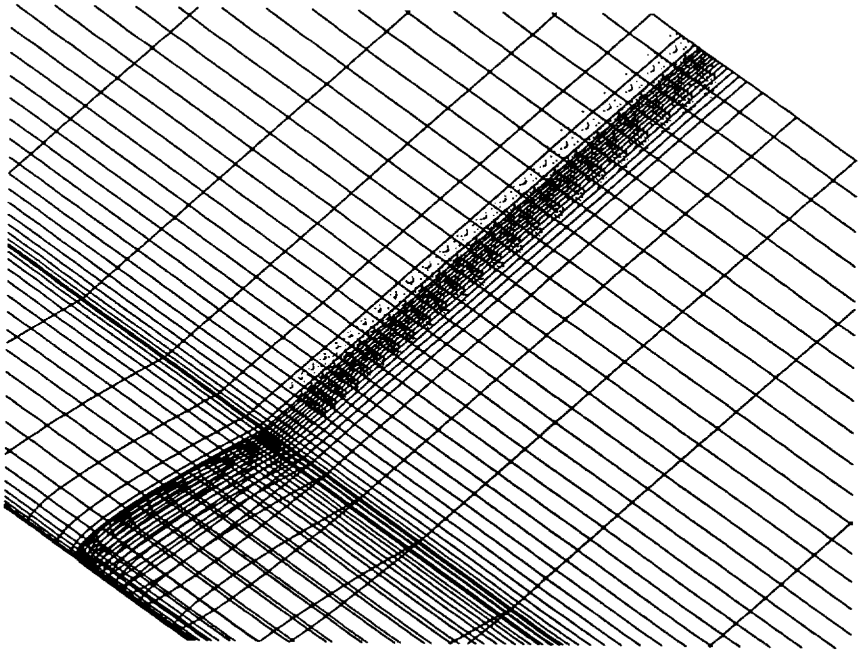


Figure 12.5: Initial location of markers in the trailing vortex region.

Figure 12.5 shows the initial patch of markers before the integration of the streamlines. Each marker is indicated by a dot. The markers are placed within a 'conical' region roughly accounting for the trailing vortex expansion downstream of the trailing edge. Because the immediate region behind the trailing edge corresponds to very high gradients, attempts to correct this zone with the Lagrangian method (where vorticity is constant over a cell) were unsuccessful. The vortex is spread over very few cells and discretizing it by markers placed at cell centers is very inaccurate. Thus, the markers were placed only from 0.2 chords downstream of the trailing edge.

The experimental survey of the wake provides axial vorticity, pressure coefficient and stagnation pressure coefficient defined as

$$Cp_0 = \frac{p_0 - p_{0\infty}}{\frac{1}{2}\rho U_\infty^2}, \qquad (12.2)$$

for two stations in the wake located at 0.5 chords and 2.0 chords downstream of the trailing edge, respectively. Figure 12.6a) shows axial vorticity contours predicted by the Eulerian solver for the two stations downstream of the trailing edge. Figure 12.6b) shows the axial vorticity contours for the same stations when the Lagrangian correction is used. The experimental values of axial vorticity are reported in Figure 12.7. The maximum computed level of vorticity is generally lower than the experimental values, indicating that a finer grid around the wing and in the wake would be required to get the vorticity level encountered in the experiments. However, the reduction of the wake diffusion using the Lagrangian technique is still relevant. The experimental data show a decrease of vorticity by a factor 1.2 between the two stations, whereas the maximum vorticity decreases by a factor 1.5 between the two stations with the Eulerian solution alone, an effect of numerical diffusion. With the Lagrangian correction, the maximum vorticity level is higher at 0.5 chords downstream of the trailing edge indicating that the numerical diffusion is reduced. This level remains approximately unchanged in the wake. At the 2 chords station, the corrected and uncorrected maximum vorticity levels vary by a factor 2, indicating that the Lagrangian correction is capable of handling large corrections of vorticity. Also the vortex core is tighter when using the Lagrangian correction.

Figure 12.8 shows the computed pressure coefficient contours on the two stations for the Eulerian solution and the Eulerian/Lagrangian solution. The experimental values are shown in Figure 12.9. The diffusion of the vortex is clearly seen at the 2 chords cross-section with the Eulerian solution. The experimental minimum pressure coefficient remains approximately unchanged between the two stations. With the Lagrangian correction, the minimum pressure coefficient is closer to the experimental value and decreases between the two stations, an effect which will be explained later. The computed stagnation pressure coefficient is shown in Figure 12.10 and the experimental values are reported in Figure 12.11. Again, the minimum stagnation pressure coefficient increases in the wake with the Eulerian solution due to numerical diffusion. The same decreasing behavior seen for the pressure is also valid for the stagnation pressure in the Eulerian/Lagrangian solution.

The maximum vorticity in the wake is traced as a function of the distance downstream of the trailing edge for the Eulerian and the Eulerian/Lagrangian solutions in Figure 12.12. The diffusion of the Eulerian solution is seen between the trailing edge ($x/c = 1.$) and the exit cross-section ($x/c = 3$). With the Lagrangian correction, the maximum vorticity remains approximately constant. The roughness of the curve is due to the vortex convecting through the grid. As mentioned earlier, the vorticity correction is not performed near the trailing edge but begins at 0.2 chords downstream of the trailing edge.

In general the present solution shows pressure coefficient and stagnation pressure coefficients in better agreement with experiments than Robert's solution. This is believed to be due to the higher grid resolution in the vortex vicinity used in the present work. Roberts' Eulerian solution also showed excessive numerical diffusion in the wake with an increase in stagnation pressure coefficient by a factor 3 between the two stations.

The minimum pressure coefficient traced as a function of the distance downstream of the trailing edge is shown in Figure 12.13. The diffusion of the vortex leads to a increase of the minimum value of the pressure coefficient with the Eulerian solution. The Lagrangian correction leads to a very different behavior. When the correction begins at 0.2 chords downstream of the trailing edge, the minimum pressure coefficient first decreases because more vorticity is entrained in the vortex and the larger vorticity region leads to a lower pressure at the vortex center. After 1 chord downstream of the trailing edge ($x/c = 2.0$), most of the trailing wake vorticity has been pulled into the vortex, and hence the minimum pressure in the vortex remains approximately constant.

The numerical diffusion of a trailing tip vortex is shown by an Eulerian calculation. The Lagrangian correction is successful at preventing the vortex diffusion downstream of the trailing edge. However, the loss of vorticity is the highest in the trailing edge region and the lack of resolution impeded the use of the Lagrangian correction at this location. A more effective correction procedure could be obtained by using the Lagrangian correction technique in conjunction with an adaptive grid refinement procedure, so that the vortex resolution remains approximately the same even near the trailing edge. The location of regions to be adapted would be indicated simply by the presence of the Lagrangian markers. Searching for 'features' in the Eulerian solution woud likely be unnecessary.

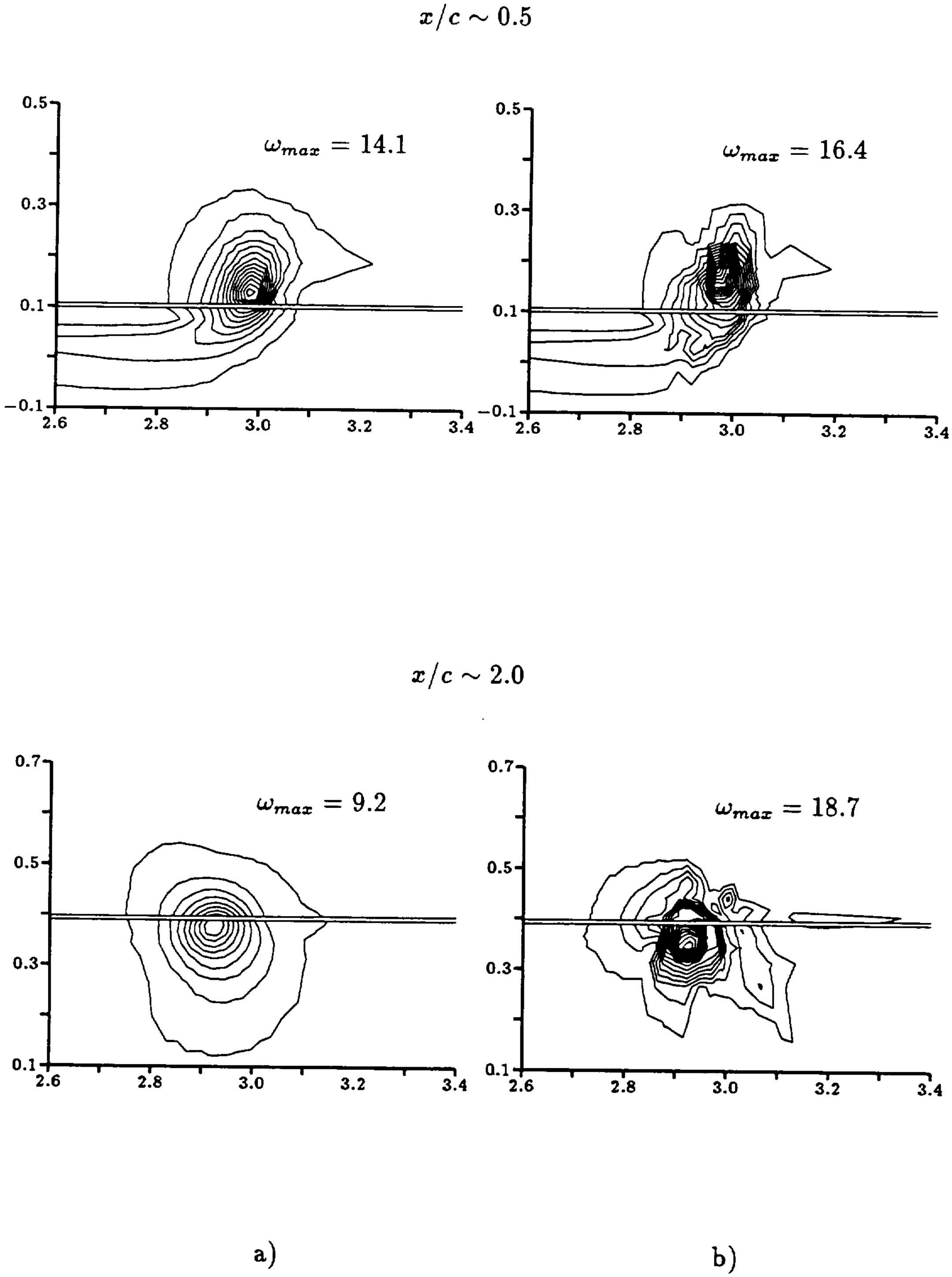
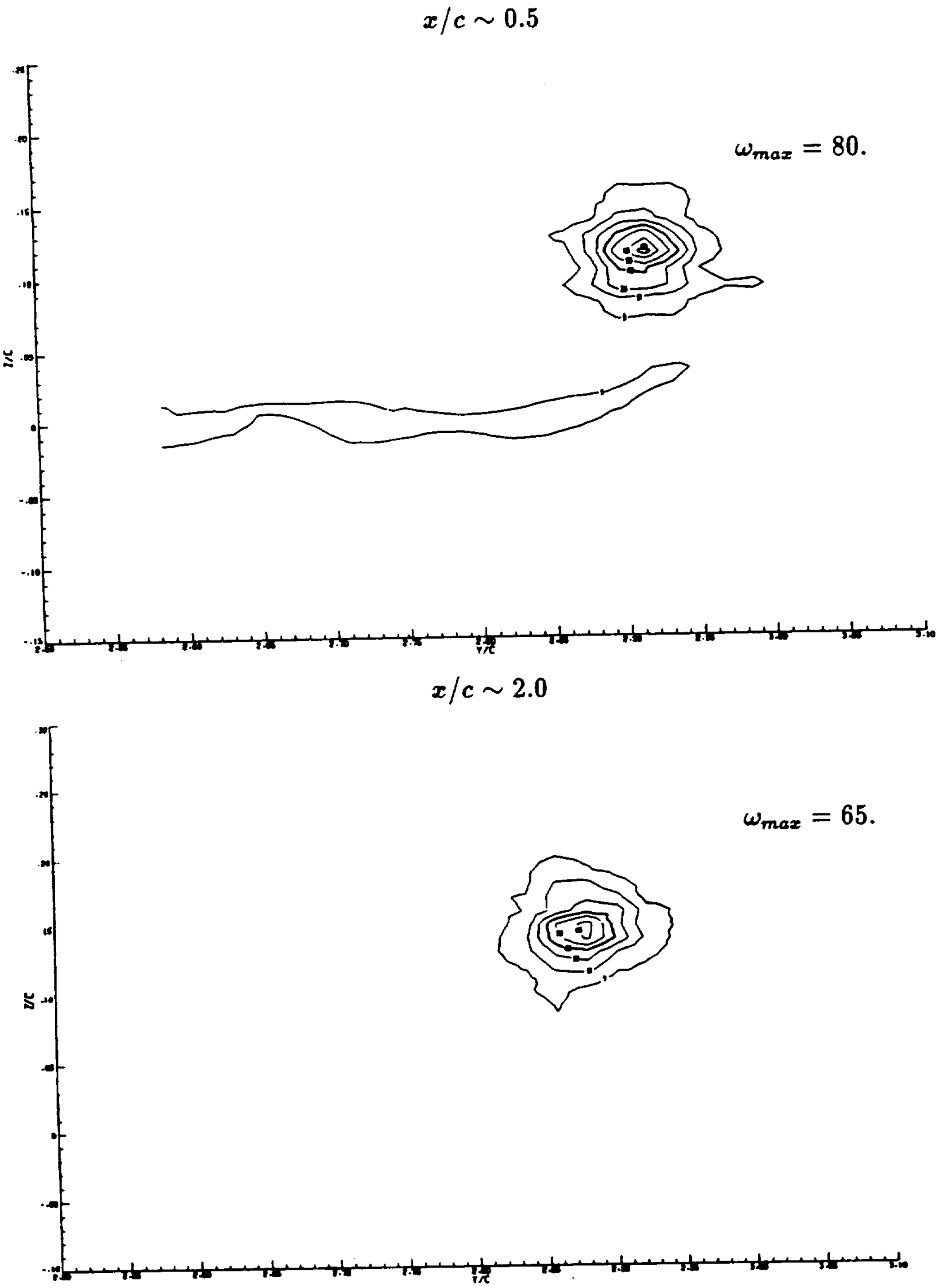


Figure 12.6: Axial vorticity in wake for stations located at 0.5 and 2.0 chords downstream of the trailing edge, a) Eulerian solution, b) Eulerian/Lagrangian solution (inc. = 1.).



$x/c \sim 0.5$

$x/c \sim 2.0$

Figure 12.7: Experimental axial vorticity in wake for stations located at 0.5 and 2.0 chords downstream of the trailing edge.

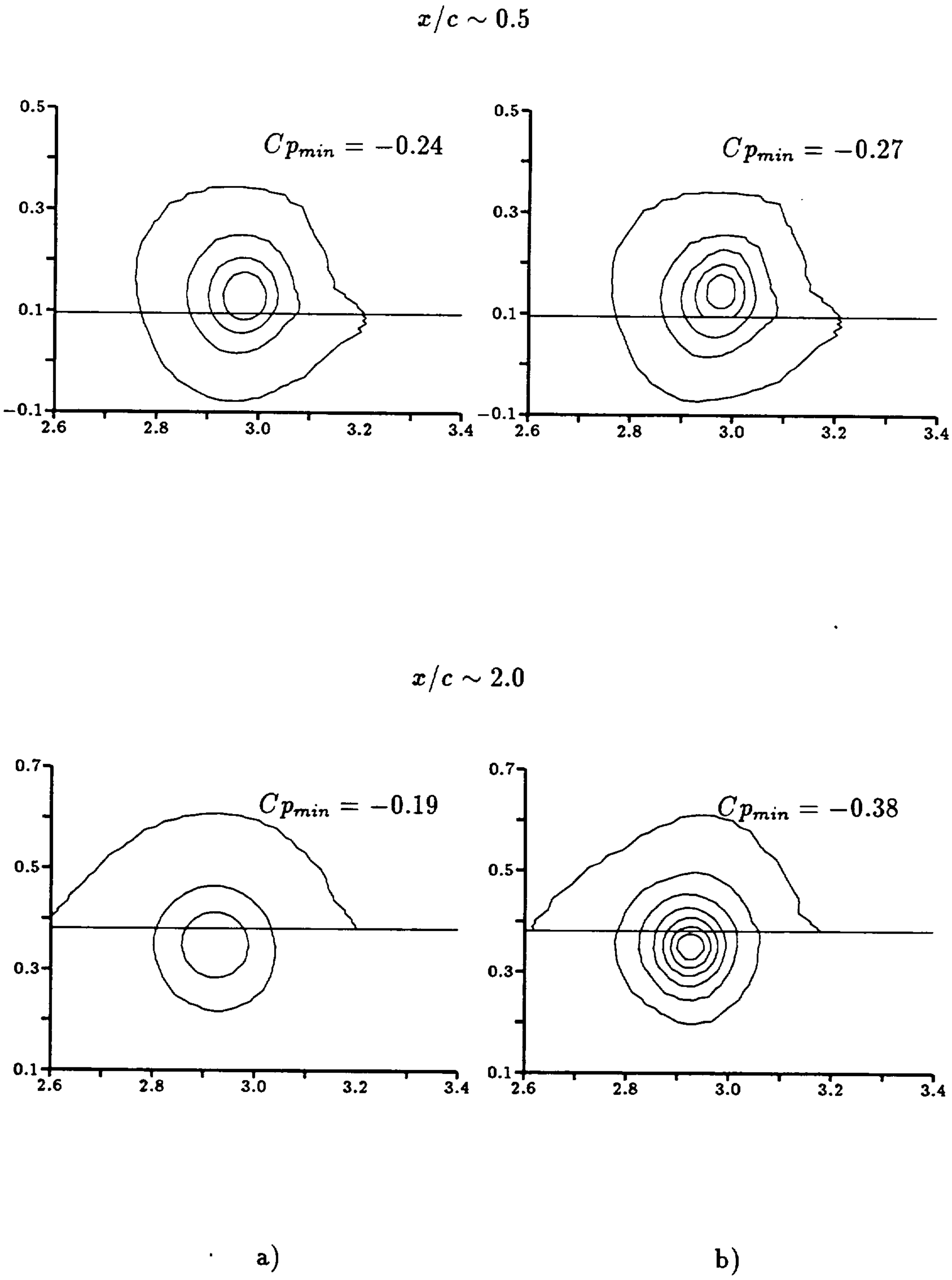


Figure 12.8: Pressure coefficient in wake for stations located at 0.5 and 2.0 chords downstream of the trailing edge, a) Eulerian solution, b) Eulerian/Lagrangian solution (inc. = 0.05).

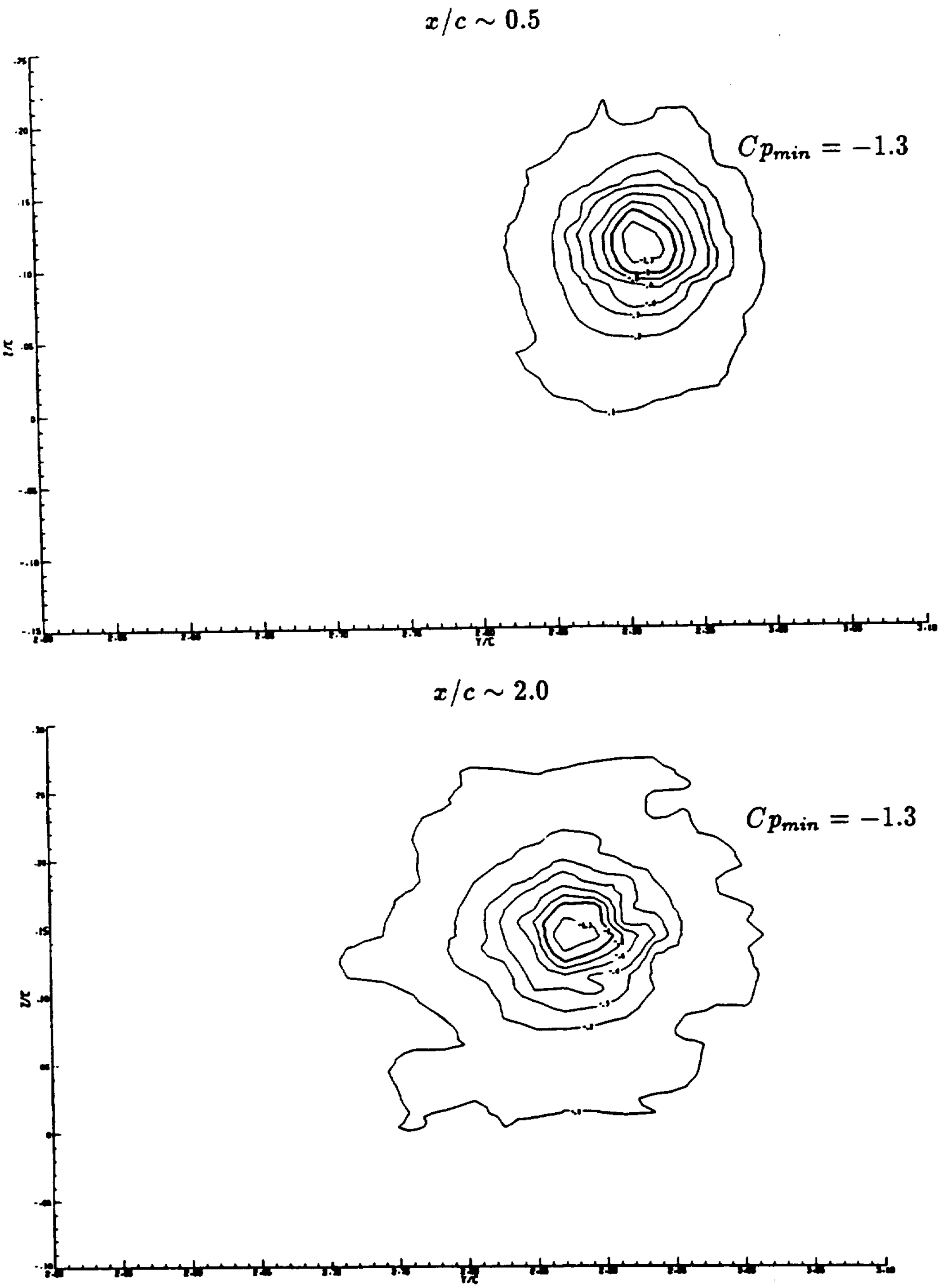


Figure 12.9: Experimental pressure coefficient in wake for stations located at 0.5 and 2.0 chords downstream of the trailing edge.

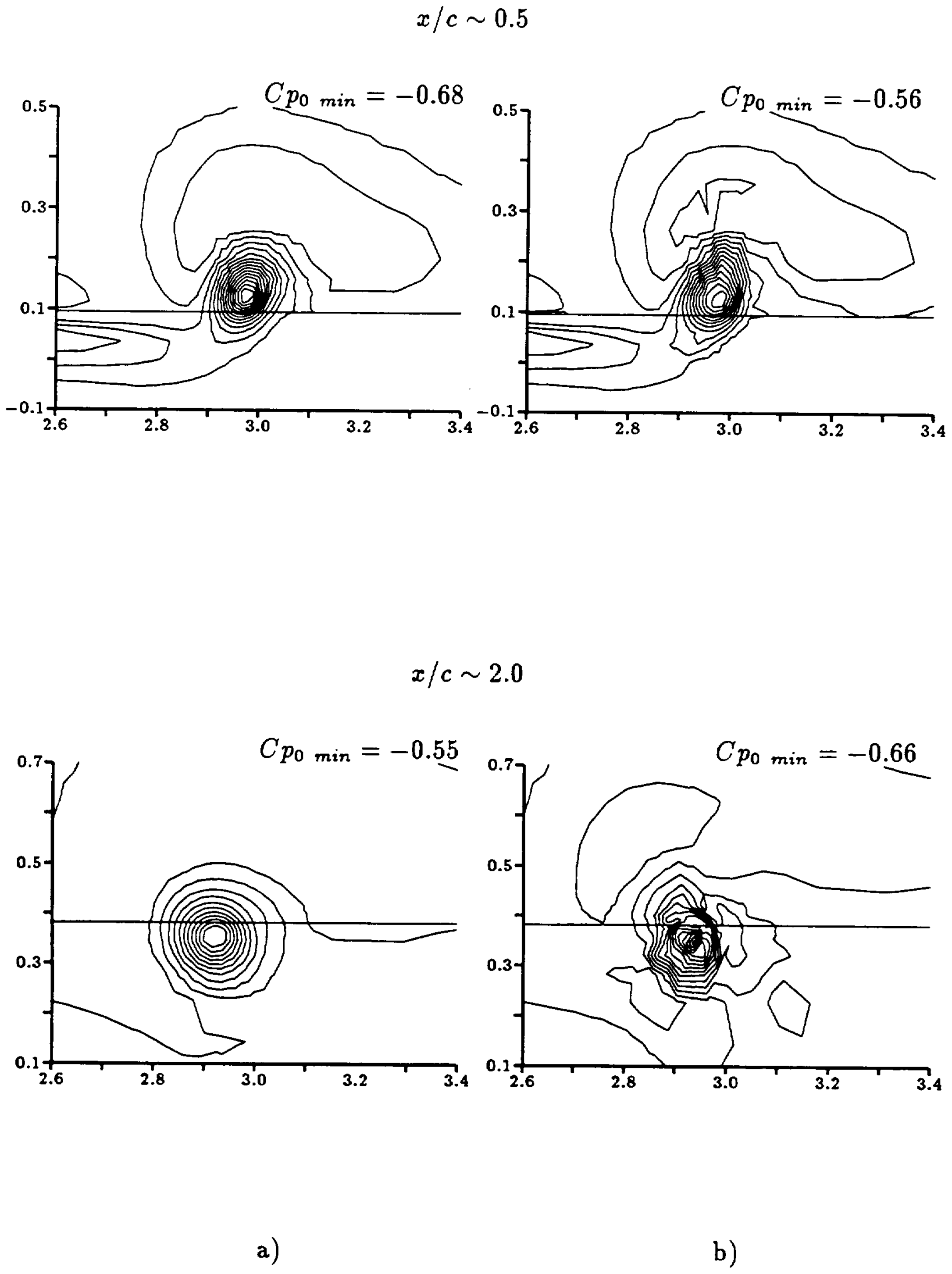


Figure 12.10: Stagnation pressure coefficient in wake for stations located at 0.5 and 2.0 chords downstream of the trailing edge, a) Eulerian solution, b) Eulerian/Lagrangian solution (inc. = 0.05).

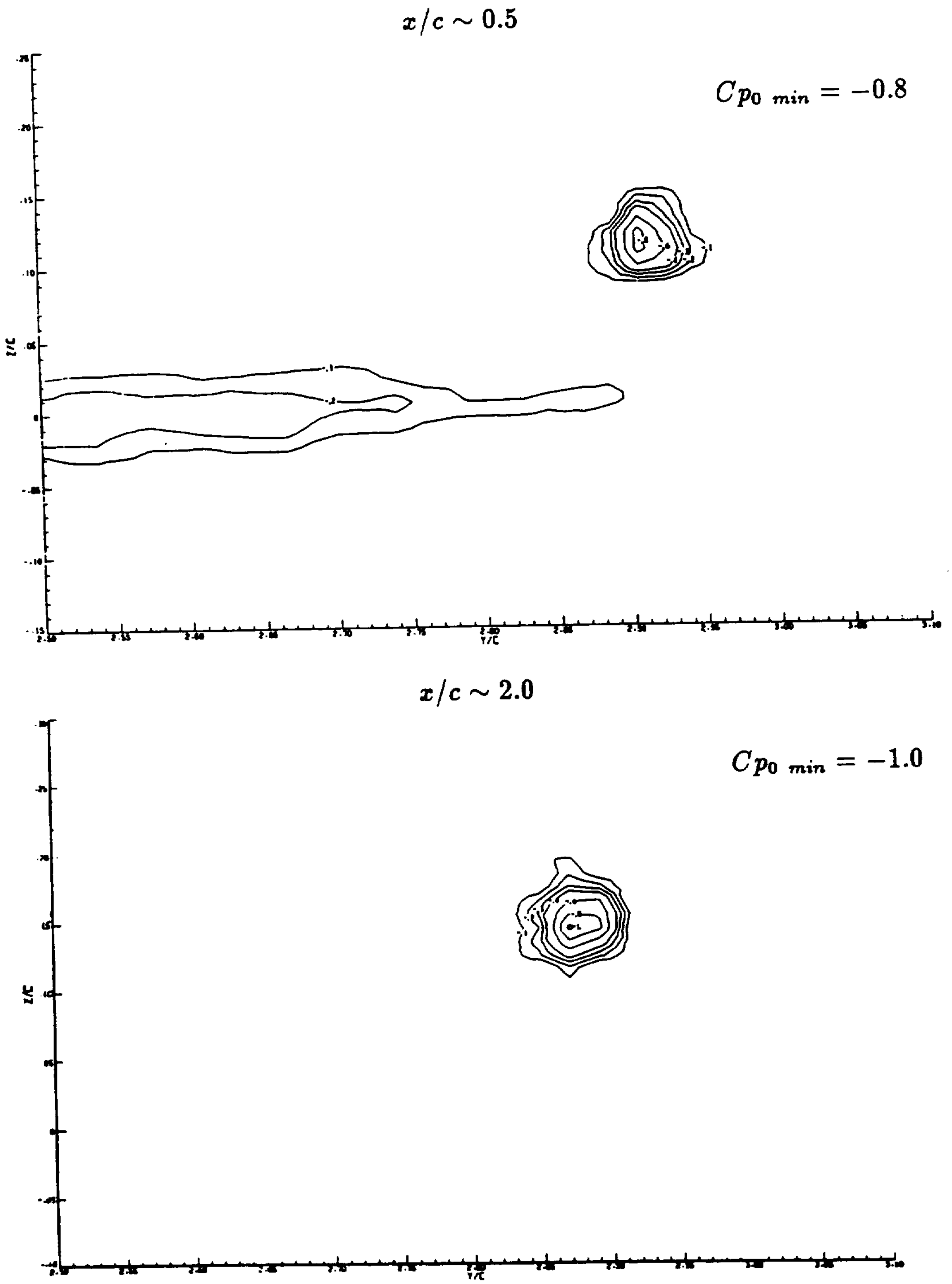


Figure 12.11: Experimental stagnation pressure coefficient in wake for stations located at 0.5 and 2.0 chords downstream of the trailing edge.

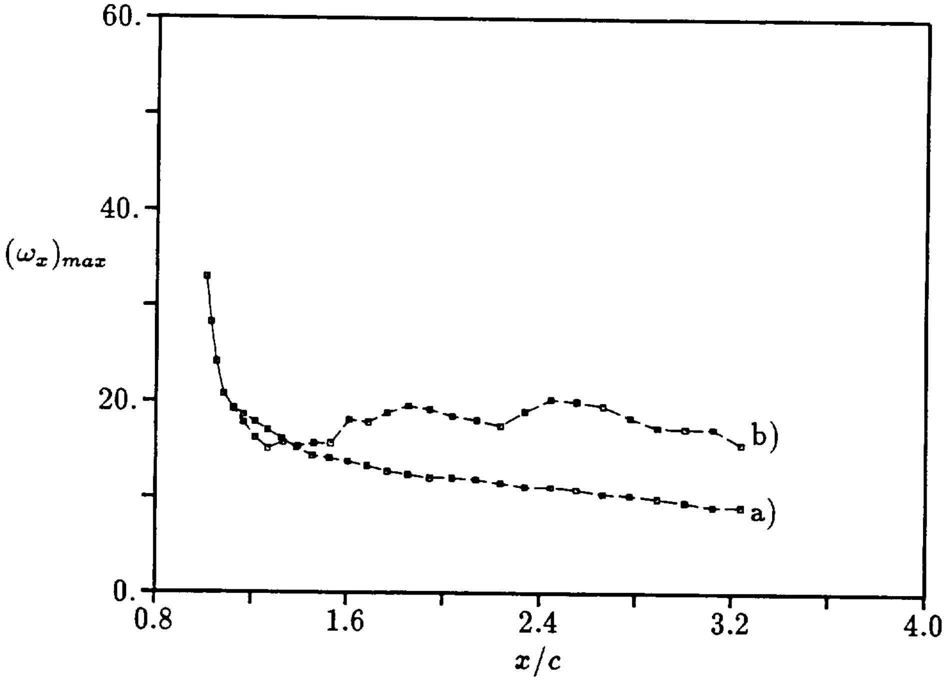


Figure 12.12: Maximum axial vorticity in wake as a function of the distance behind the trailing edge, a) Eulerian solution, b) Eulerian/Lagrangian solution.

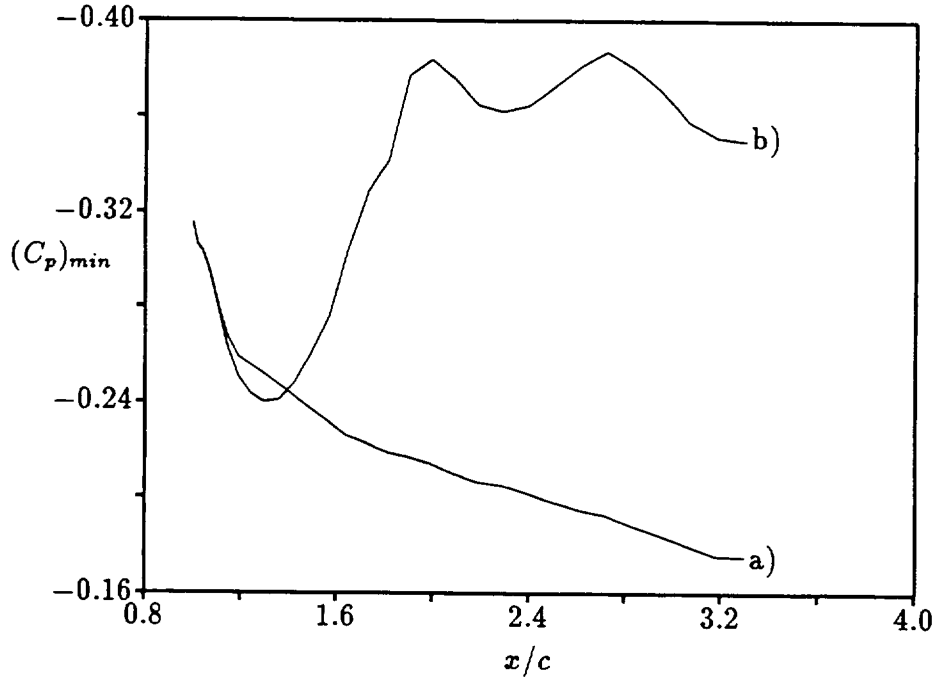


Figure 12.13: Minimum pressure coefficient in wake as a function of the distance behind the trailing edge, a) Eulerian solution, b) Eulerian/Lagrangian solution.

Chapter 13

Conclusions

13.1 Summary

A new approach is proposed as the coupling of an Eulerian and a Lagrangian solution procedures for the reduction of numerical diffusion encountered in finite-difference time-marching Eulerian calculations. The motivation behind this work is the efficient numerical treatment of flow non-homogeneities, such as vortex wakes, embedded in an otherwise smooth background flow field. The coupling of the Eulerian and the Lagrangian solution techniques is intended to enhance the poor vorticity and entropy capturing capabilities of standard Eulerian solvers.

The first part of this thesis deals with the numerical methodology used for the solution of the Euler and Lagrange equations. The Euler equations are solved using a Lax-Wendroff algorithm on an unstructured grid, and are used for compressible and incompressible (through the artificial compressibility concept) flow situations. The numerical smoothing is based on a second-difference for compressible flows and on a fourth-difference (second-order accurate on distorted grids) formulation for incompressible flows. The far-field, wall and symmetry boundary conditions are described in both cases. A numerical study is performed to prove the second-order accuracy of the scheme.

The Lagrangian solution method, based on particle markers, is then described. The integration in time of the position, vorticity and entropy attached to the markers flowing at the local speed through the Eulerian grid is performed using a predictor/corrector scheme. The flow quantities required at a marker location are tri-linearly interpolated from the cell containing the marker to the marker location. The comparative advantages of two strategies of trajectory integration and/or marker positioning in the flow are described for representative flow situations. The coupling sequence between Eulerian and Lagrangian solutions is completed (and this is the key point of the procedure) by a 'correction step' in which the Lagrangian markers provide information on how to 'correct' the spurious numerical diffusion of the Eulerian solution. As mentioned in this work, the correction step takes place only locally, each marker influencing only the cell where it is located. The correction procedure is described as a 'vorticity correction' where the vorticity attached to the marker is used to alter the velocity components at the nodes of a cell. Additionally, an 'entropy correction' is described for the compressible flow cases. The corrections are implemented as iterative procedures. The convergence speed of the vorticity correction process is reported.

In all the test cases, the improvement due to the combined Eulerian/Lagrangian solver is tested by comparison to the standard Eulerian solution and to the Eulerian solution on a finer grid (where the numerical diffusion effects are lower). All combined Eulerian/Lagrangian solutions are computed on the basic coarse Eulerian grid. Also, in some flow cases, a comparison with experiments is performed. The CPU requirements for the combined scheme are also compared to the CPU needed for coarse and fine grid Eulerian calculations.

The first test case is a compressible unsteady calculation, namely a Lamb vortex convection in a straight channel. The numerical diffusion effects are assessed by comparing the vortex solutions at the beginning and at the end of the channel.

The preservation of a turbulent inlet velocity profile in a straight pipe shows the numerical diffusion occurring near the pipe wall (where the gradients are high) with the standard Eulerian solver. The use of the combined Eulerian/Lagrangian solver results, instead, in a solution indistinguishable from the exact solution.

The calculation of a swirling flow through a straight pipe, superimposed on a uniform axial velocity, is used as a preamble to a secondary flow calculation in a bent pipe. Again, numerical diffusion is identified in the Eulerian solution, especially near the pipe wall. By comparison with an Eulerian solution on a finer grid, the combined scheme is shown to be successful at reducing numerical diffusion errors. The circulation around a closed convecting curve (which should remain constant in an incompressible inviscid flow) is also used as an integral measure of numerical diffusion and shows the substantial improvement obtained with the combined scheme. A basic problem of inviscid flow calculations is also identified as a vorticity gradient augmentation, encountered in this swirling flow case, but also present when computing secondary flows in bent pipes. This concentration leads to a poor representation of the vorticity on the fixed size grid and destabilizes the combined scheme. The source for the phenomenon is identified as a 'vorticity convection process' along streamlines, and appears in the absence of strong source-terms for the vorticity. The intensification process is shown to worsen when using the combined scheme because the reduction of numerical diffusion leads to a gradient definition over fewer cells. A solution is proposed as the introduction of a Lagrangian pseudo-diffusion term, similar in form to the true viscous diffusion term, introduced in the right-hand-side of the Helmholtz equation. This procedure is shown to result in a solution accurately described on the fixed grid. The change in circulation around a closed convecting curve is shown to be still much lower for the combined scheme with the Lagrangian pseudo-diffusion term than for the basic Eulerian solution.

A constant stagnation pressure flow in a 90^0 bend is investigated. The errors in vorticity and stagnation pressure are reduced when using the combined scheme. The errors in vorticity are shown to reduce far below the errors encountered in an Eulerian calculation with a grid twice as fine in each direction. The errors in stagnation pressure are shown to be corrected implicitly by the correction of vorticity and velocity (through Crocco's equation).

Two secondary flow calculations in bent pipes are performed. The secondary flow results from the imposition of a turbulent inlet velocity profile. Since this work deals with Eulerian calculations, only the tilting/stretching of the vortex lines are taken into account. The production of secondary flow is described, and the use of the Euler equations (instead of the Navier-Stokes equations) is motivated for this particular flow problem. The Lagrangian correction is employed as an alternative to expensive fine grid Eulerian calculations in order to reduce the numerical diffusion effects. As shown in the calculations, these effects have a strong influence on the strength and position of the secondary flow. Because numerical diffusion and real viscous diffusion are shown to result in different flow behaviors, the use of a combined Eulerian/Lagrangian scheme with lower numerical diffusion is required in order to assess the fluid real viscous effects when comparing with experiments. The limitations implied by the choice of an Eulerian solver are also addressed. A first attempt at including the near wall viscous effects is then reported as the introduction of a 'law of the wall' correction. Two bend geometries and flow cases are tested and the numerical results compared with available experiments.

The external flow over a three-dimensional wing is the object of the last flow case. The basic Eulerian and combined Eulerian/Lagrangian solutions are compared in terms of numerical diffusion of the tip vortex behind the trailing edge. The numerical diffusion is shown to be excessive for the Eulerian solution and reduced for the combined scheme. Comparison with experiments is performed. The flexibility of the proposed method is shown by correcting the solution selectively in the tip-vortex region, and by minimizing the CPU required for the calculation.

13.2 Contributions

To the author's knowledge, the coupling of an Eulerian and a Lagrangian solution procedures in three dimensions, enabling a correction of the Eulerian state vector based on Lagrangian values and aimed at reducing numerical diffusion errors, represents an original contribution. The present 3-D version is based on the 2-D work of [20], but includes the computation of the source-terms for the vorticity not required in two dimensions. The alternate upstream integration technique of the streamlines is also an original contribution of this work.

The efficiency and flexibility of this new approach has been demonstrated by application to flow cases of different characteristics. The treated examples included steady, unsteady, compressible, incompressible as well as internal and external flow applications. The combined Eulerian/Lagrangian scheme takes advantage of both the accurate 'elliptic' representation of the Eulerian solution enforcing the mass requirements and setting the pressure field and the convection capturing capabilities of the Lagrangian solution. The contribution of this work is, therefore, the addition of built-in convection properties to a standard Eulerian solver.

The numerical diffusion effects are quantified by comparing Eulerian solutions on coarse and fine grids to coarse grid Eulerian/Lagrangian solutions. Using the same solution comparisons, the CPU requirements are shown to be substantially lower, for a given accuracy, when using the combined Eulerian/Lagrangian scheme. Also, reducing the numerical diffusion allows one to identify the true inviscid behavior of the flow and to assess the real viscosity effects when comparing with experiments.

In the course of this work, a vorticity gradient augmentation phenomenon has been identified which resulted in poor vorticity representation on the Eulerian grid and destabilization of the combined scheme. In order to remediate to this problem, a Lagrangian pseudo-diffusion term has been added to the Helmholtz equation, without substantially compromising the reduction of numerical diffusion.

In its present form, the Lagrangian correction technique consists in a set of subroutines which can be added a posteriori, not only to the present Lax-Wendroff time-marching technique, but to any time-marching finite-volume algorithm for the solution of the Euler equations on hexahedral cells. The application of the Lagrangian algorithm to tetrahedral cells, for example, would require the redefinition of the linear functions within the cells.

13.3 Conclusions and recommendations for future work

As mentioned in the introduction, the different approaches taken to efficiently resolve small size flow non-homogeneities result from a compromise between CPU/memory requirements and solution accuracy. Any attempt at improving the solution accuracy always ends up increasing the requirements in CPU/memory. The accuracy of the flow solution is linked either with grid resolution issues leading to grid refinement strategies or with the use of a priori known solutions and analytical flow models correcting the basic flow solution, or also with the use of more accurate discretization/integration algorithms. All of these are used separately or in conjunction with each other in order to cope with the poor solution accuracy encountered in finite-difference solutions of the Euler (or Navier-Stokes) equations in regions of high gradients and small length scales with respect to the background flow. All these studies were prompted by the important influence of these regions on the overall flow solution.

The objective of this work was to obtain a more accurate solution on a fixed grid and can also be viewed as a lowering of the CPU/memory (grid size) requirements for a given accuracy. The comparison of the proposed approach against existing methods has been expanded upon in the introduction and will not be repeated here.

By increasing the solution accuracy on the given grid, the proposed combined Eulerian/Lagrangian scheme allows for a more accurate solution of the high gradients of the flow. Two of the main advantages of the proposed method are the flexibility by which corrections to the Eulerian solution can be performed in only chosen areas of the computational domain, and the fact that the markers have an effect limited to the cells where they are located. This makes the combined Eulerian/Lagrangian scheme perform better in situations where the flow non-homogeneities are concentrated, and less suited where the vorticity and entropy corrections have to cover the entire domain. For example, the correction of the relatively small non-homogeneity represented by the unsteady convection of a Lamb vortex in a channel (Chapter 7) leads to $\sim$ 30% CPU increase over the basic Eulerian solution. In the case of steady pipe flows, since markers have been placed at the center of each cell, the CPU increase due to the introduction of the Lagrangian correction is directly linked to the size of the computational domain. The introduction of the Lagrangian correction for the preservation of a turbulent inlet velocity profile in a straight pipe (Chapter 8) results in a 30% CPU increase. The correction of the swirling flow of Chapter 9, in the identical computational domain, required a 33% CPU increase. In the case of the constant stagnation pressure flow in the 90^0 bend of Chapter 10, with twice as many nodes as the previous pipe cases, the increase in CPU was $\sim$ 75%. Of course, the particular CPU increase depends also on the flow characteristics (a high turning of streamlines requiring more Lagrangian integration steps).

An advantage of the combined scheme is the presence of built-in convective properties, so that an a priori knowledge of the position or strength of the flow non-homogeneities is not required.

Another conclusion to this work is that the use of the combined Eulerian/Lagrangian scheme does not quite eliminate the need for extra grid resolution. In reducing the numerical diffusion phenomenon, the gradients of the flow become larger and, in some flow situations, the grid becomes too coarse to accurately support the new gradients, as mentioned in Section 9.2. The same grid resolution issue limits the correction of the wing tip vortex from only 20% chord downstream of the trailing edge (upstream of this distance, the vortex was defined on too few cells to attempt a Lagrangian correction procedure). This phenomenon is particularly acute in inviscid flow computations where no viscous diffusion occurs to spread the non-homogeneity on more grid cells.

The use of the proposed Eulerian/Lagrangian technique in combination with a grid adaptation should prove to be a possible solution to this problem. In addition, the refinement of the grid could be linked to the markers trajectories. For example, when placing markers at the inlet of a bent pipe in the high vorticity regions near the wall, the grid could be refined along the markers trajectories as the markers move downstream and gather into a tight secondary vortex. The level of grid refinement could be based on the vorticity and entropy carried by the markers. Also, this combination technique would be advantageous in terms of CPU requirements for pipe flows where, instead of placing markers in each cell, the selective addition of markers could be realized since the refinement technique would prohibit many markers with largely different state vectors from being located in the same cell during the tightening of the secondary vortex.

The use of a Lagrangian particle tracking solver in conjunction with the Navier-Stokes equations instead of the Euler equations requires the addition of the viscous dissipation and diffusion term in the Helmholtz equation. This is not viewed as a main roadblock since a similar term has already been implemented in this equation when using a Lagrangian pseudo-diffusion along the markers streamlines in Section 9.2. As mentioned by Drela in [20], the main problem here is to account for turbulence (as suitable models for turbulence do not exist for vortex wakes). Fortunately, the dynamics of vortex wakes is mostly inviscid. However, in the case of a secondary flow calculation, while the combined Eulerian/Lagrangian solver gives realistic results near the entry of the bend (or bends of small turnings), the modeling of the migration of the viscous effects towards the pipe center would bring the computed results closer to experiments for high degrees of turning. In this case, the Lagrangian technique would not be used near the pipe walls since the resolution provided by a Navier-Stokes grid ensures accurate solution of the viscous stresses near the walls. Still, the Lagrangian technique would be useful near the pipe center where the grid resolution is comparatively coarse for the accurate solution of the secondary vortex.

The Lagrangian equations defined in Chapter 4 are not valid through a shock. The implementation of the combined Eulerian/Lagrangian scheme for shock flows is feasible if the markers do not cross the shock region. Therefore, the Lagrangian state vector has to be reinitialized (using the Eulerian solution) from one side of the shock to the other.

Since the correct jump conditions across the shock are ensured for any conservative Eulerian scheme, this procedure should provide the markers with correct values of entropy and vorticity. Also, as mentioned in [20], the exact location where the markers have to be stopped and reinitialized should not be critical since the usual grid refinement used in Eulerian solutions near the shock region will ensure a locally accurate solution.

Bibliography

[1] S. Abdallah and A. Hamed. Inviscid Solution for the Secondary Flow in Curved Ducts. *AIAA Journal*, 19(8):993–999, August 1981.

[2] J. Ackeret. Aspects of Internal Flow. In Servan, editor, *Fluid Mechanics of Internal Flow*, pages 1–26, 1965.

[3] John D. Jr. Anderson. *Modern Compressible Flow: With Historical Perspective.* McGraw-Hill, 1982.

[4] H. Ashley and M. Landahl. *Aerodynamics of Wings and Bodies.* Addison-Wesley, Reading, Mass., 1965.

[5] G. R. Baker. The Cloud-in-Cell Technique Applied to the Roll Up of Vortex Sheets. *Journal of Computational Physics*, 31:76–95, 1979.

[6] S. A. Barton. An Experimental Investigation of the Influence of Inlet Distortion on the Fluid Borne Noise of a Centrifugal Pump. Master's thesis, MIT, September 1991.

[7] J. Basuki and J. M. R. Graham. Discrete Vortex Computation of Separated Airfoil Flow. *AIAA Journal*, 25(11):1409–1410, November 1987.

[8] G. K. Batchelor. *An Introduction to Fluid Dynamics.* Cambridge Univ. Press, 1967.

[9] T. M. Becker. Hybrid 3-D Euler/Navier-Stokes Calculations of Transonic Vortex Flows over the NTF Delta Wing. Master's thesis, MIT, August 1989.

[10] S. A. Berger, L. Talbot, and L.-S. Yao. Flow in Curved Pipes. *Annual Review of Fluid Mechanics*, 15:461–512, 1983.

[11] P. Bradshaw. Effect of streamline curvature on turbulent flow. AGARDograph 169, 1973.

[12] W. R. Briley and H. McDonald. Three-dimensional viscous flows with large secondary velocity. *Journal of Fluid Mechanics*, 144:47–77, 1984.

[13] J. L. C. Chang and D. Kwak. On the Method of Pseudo-Compressibility for Numerically Solving Incompressible Flows. AIAA Paper 84-0252, 1984.

[14] A. J. Chorin. A Numerical Method for Solving Incompressible Viscous Flow Problems. *Journal of Computational Physics*, 2:12–26, 1967.

[15] L. J. Chow, T. H. Pulliam, and J. L. Steger. A General Perturbation Approach for the Equations of Fluid Dynamics. AIAA Paper 83-1903CP, 1983.

[16] J. P. Christiansen. Numerical Simulation of Hydrodynamics by the Method of Point Vortices. *Journal of Computational Physics*, 13:363, 1973.

[17] S. C. R. Dennis, D. B. Ingham, and R. N. Cook. Finite-Difference Methods for Calculating Steady Incompressible Flows in Three Dimensions. *Journal of Computational Physics*, 33:325–339, 1979.

[18] J. D. Denton. The Use of a Distributed Body Force to Simulate Viscous Effects in 3-D Flow Calculations. ASME Paper 86-GT-144, 1986.

[19] R. W. Detra. The Secondary Flow in Curved Pipes. Mitteilung aus dem Institut fuer Aerodynamik 20, Eidgenoessische Technische Hochshule, Zuerich, 1953.

[20] M. Drela and E. M. Murman. Prospects for Eulerian CFD Analysis of Helicopter Vortex Flows. American Helicopter Society Specialist Meeting, Arlington Texas, February 1987 also MIT CFDL Technical Report TR-87-3, 1987.

[21] M. M. Enayet, M. M. Gibson, A. M. K. P. Taylor, and M. Yianneskis. Laser Doppler Measurements of Laminar and Turbulent Flow in a Pipe Bend. NASA CR-3551, May 1982.

[22] M. B. Giles. Non-Reflecting Boundary Conditions for the Euler Equations. Technical Report TR-88-1, MIT Computational Fluid Dynamics Laboratory, 1988.

[23] M. B. Giles. UNSFLO: A Numerical Method for Unsteady Inviscid Flow in Turbomachinery. Technical Report 195, MIT Gas Turbine Laboratory, 1988.

[24] F. H. Harlow and J. E. Welch. Numerical Calculation of Time-Dependent Viscous Incompressible Flow with Free Surface. *Physics of Fluids*, 8:2182–2189, December 1965.

[25] A. A. Hassan, C. Tung, and L. N. Sankar. Euler Solutions for Self-Generated Rotor Blade-Vortex Interactions. AIAA Paper 90-1588, 1990.

[26] W. R. Hawthorne. Secondary Circulation in Fluid Flow. *Proc. Roy. Soc.*, 206:374, 1951.

[27] W. R. Hawthorne. *Research Frontiers in Fluid Dynamics*, volume 15 of *Interscience*, chapter 1, pages 1–29. Seeger, R. J. and Temple, G., 1965.

[28] W. R. Hawthorne. The Applicability of Secondary Flow Analyses to the Solution of Internal Flow Problems. In Servan, editor, *Fluid Mechanics of Internal Flow*, pages 238–269, 1965.

[29] W. R. Hawthorne. Secondary Vorticity in Stratified Compressible Fluids in Rotating Systems. CUED/A-Turbo/TR 63, University of Cambridge, Department of Engineering, 1974.

[30] E. H. Hirschel and A. Rizzi. The Mechanism of Vorticity Creation in Euler Solutions for Lifting Wings. Proceedings of Intl. Vortex Flow Symposium, FFA Technical Report Stockholm, 1987.

[31] D. G. Holmes and S. D. Connell. Solution of the 2-D Navier-Stokes Equations on Unstructured Adaptive Grids. AIAA Paper 89-1932-CP, 1989.

[32] D. Howard. Study of Incompressible Flow with 3-D Finite Elements. In *Applications of Supercomputers in Engineering*, pages 277–292, 1991.

[33] J. A. C. Humphrey, H. Iacovides, and B. E. Launder. Some numerical experiments on developing laminar flow in circular-sectioned bends. *Journal of Fluid Mechanics*, 154:357–375, 1985.

[34] J. A. C. Humphrey, A. M. K. Taylor, and J. H. Whitelaw. Laminar flow in a square duct of strong curvature. *Journal of Fluid Mechanics*, 83:508–527, 1977.

[35] J. A. C. Humphrey, J. H. Whitelaw, and G. Yee. Turbulent flow in a square duct with strong curvature. *Journal of Fluid Mechanics*, 103:443–463, 1981.

[36] A. Jameson. Time Dependent Calculations Using Multigrid, with Applications to Unsteady Flows Past Airfoils and Wings. AIAA Paper 91-1596, 1991.

[37] A. Jameson and S. Yoon. Multigrid Solution of the Euler Equations Using Implicit Schemes. *AIAA Journal*, 24(11):1737–1743, November 1986.

[38] O. M. Knio and A. F. Ghoniem. Three Dimensional Vortex Simulation of Entrainement Augmentation Due to Streamwise Structures. AIAA Paper 89-0574, 1989.

[39] B. Lakshminarayana and J. H. Horlock. Generalized expressions for secondary vorticity using intrinsic co-ordinates. *Journal of Fluid Mechanics*, 59:97–115, 1973.

[40] A. Landsberg. Adaptation for Vortex Flows using a 3-D Finite Element Solver. Master's thesis, MIT, November 1990.

[41] A. Leonard. Computing Three-Dimensional Incompressible Flows With Vortex Elements. *Annual Review of Fluid Mechanics*, 17:523–559, 1985.

[42] R. Löhner. An Adaptative Finite Element Scheme for Transient Problems in CFD. *Computer Methods in Applied Mechanics and Engineering*, 61:323–338, 1987.

[43] C. L. Merkle and M. Athavale. Time-Accurate Unsteady Incompressible Flow Algorithms Based on Artificial Compressibility. AIAA Paper 87-1137, 1987.

[44] R. H. Miller, T. W. Roberts, and E. M. Murman. Computational Methods for Wake Modeling and Blade Airload Determination in Hover and Forward Flight. *Comp. & Maths. with Applications*, 12A(1):29–52, 1986.

[45] E. M. Murman and A. Rizzi. Applications of Euler Equations to Sharp Edge Delta Wings With Leading Edge Vortices. Presented at the AGARD Symposium

on Application of Computational Fluid Dynamics in Aeronautics, AGARD PEP, 1986.

[46] K. Nakahashi and G. S. Deiwert. Self-Adaptive-Grid Method with Application to Airfoil Flow. *AIAA Journal*, 25(4):513–520, April 1987.

[47] R.-H. Ni. A Multiple Grid Scheme for Solving the Euler Equations. *AIAA Journal*, 20:1565–1571, Nov 1981.

[48] R.-H. Ni and J. C. Bogoian. Prediction of 3-D Multi-Stage Turbine Flow Field Using a Multiple-Grid Euler Solver. AIAA Paper 89-0203, 1989.

[49] K. G. Powell, M. A. Beer, and G. W. Law. An Adaptive Embedded Mesh Procedure for Leading-Edge Vortex Flows. AIAA Paper 89-0080, 1989.

[50] K. G. Powell and E. M. Murman. Total Pressure Losses in Vortical Solutions of the Conical Euler Equations. *AIAA Journal*, 25:360–368, 1986.

[51] K. G. Powell, E. M. Murman, R. M. Wood, and D. S. Miller. A Comparison of Experimental and Numerical Results for Delta Wings with Vortex Flaps. AIAA Paper 86-1840-CP, 1986.

[52] V. S. Pratap and D. B. Spalding. Numerical Computations of the Flow in Curved Ducts. *The Aeronautical Quarterly*, pages 219–228, August 1975.

[53] M. M. Rai. Navier-Stokes Simulations of Blade-Vortex Interaction Using High-Order Accurate Upwind Schemes. AIAA Paper 87-0543, 1987.

[54] K. Ramachandran, F. X. Caradonna, and J. Steinhoff. The Free-Wake Computation of Rotor-Body Flows. AIAA Paper 90-1540, 1990.

[55] A. Rizzi and L.-E. Eriksson. Computations of flow around wings based on the Euler Equations. *Journal of Fluid Mechanics*, 148:45–71, November 1984.

[56] T. W. Roberts. *Euler Equation Computations for the Flow over a Hovering Helicopter Rotor*. PhD thesis, MIT, November 1986.

[57] T. W. Roberts and E. M. Murman. Euler Solutions for the Flow Around a Hovering Helicopter Rotor. AIAA Paper 86-1784, 1986.

[58] S. E. Rogers, J. L. C. Chang, and D. Kwak. A Diagonal Algorithm for the Method of Pseudo-Compressibility. *Journal of Computational Physics*, 73:364–379, 1987.

[59] M. Rowe. Measurements and computations of flow in pipe bends. *Journal of Fluid Mechanics*, 43:771–783, 1970.

[60] I. L. Ryhming. *Dynamique des Fluides.* Presses Polytechniques Romandes, Lausanne, Switzerland, 1985.

[61] A. Saxer, H. Felici, C. Neury, and I. L. Ryhming. Euler Flows in Hydraulic Turbines and Ducts Related to Boundary Conditions Formulation. In M. Deville, editor, *Notes on Numerical Fluid Mechanics*, pages 343–354. GAMM, 1988. Proceedings of the Seventh GAMM Conference on Numerical Methods in Fluid Mechanics, Volume 20.

[62] H. Schlichting. *Boundary-Layer Theory.* McGraw-Hill, 1979.

[63] M. A. Schmatz. Calculation of Strong Viscous/Inviscid Interactions on Airfoils by Zonal Solutions of the Navier-Stokes Equations. Proceedings of the Sixth GAMM-Conference on Numerical Methods in Fluid Dynamics, 1986.

[64] M. Scully. Computations of Helicopter Rotor Wake Geometry and its Influence on Rotor Harmonic Airload. MIT ASRL Technical Report TR-178-1, Massachusetts Institute of Technology, 1975.

[65] C. H. Shieh. Three-Dimensional Grid Generation Using Poisson Equations. In Joe F. Thompson, editor, *Numerical Grid Generation*, pages 687–694. Elsevier Science Publishing Company, 1982.

[66] R. L. Sorenson and J. L. Steger. Grid Generation in Three Dimensions by Poisson Equations with Control of Cell Size and Skewness at Boundary Surfaces. In *Advances in Grid Generation*, Houston, Texas, 1983. Applied Mechanics, Bioengineering, and Fluids Engineering Conference.

[67] H. B. Squire and K. G. Winter. The Secondary Flow in a Cascade of Airfoils in a Nonuniform Stream. *Journal of the Aeronautical Science*, 18:271–277, 1951.

[68] G. R. Srinivasan, W. J. Mccroskey, and J. D. Baeder. Aerodynamics of Two-Dimensional Blade-Vortex Interaction. *AIAA Journal*, 24(10):1569–1576, 1986.

[69] G. R. Srinivasan, W. J. McCroskey, and P. Kutler. Numerical Simulation of the Interaction of a Vortex with Stationary Airfoil in Transonic Flow. AIAA Paper 84-0254, 1984.

[70] J. L. Steger and P. Kutler. Implicit Finite-Difference Procedures for the Computation of Vortex Wakes. *AIAA Journal*, 15(4):581–590, 1977.

[71] J. Steinhoff and K. Ramachandran. Free Wake Analysis of Compressible Rotor Flows. AIAA Paper 87-0542, 1987.

[72] J. Steinhoff, H. Senge, and F. Decker. Application of Vortex Embedding to Aircraft Flows. AIAA Paper 90-1626, 1990.

[73] J. C. Strikwerda. *Finite Differences Schemes and Partial Differential Equations.* Wadsworth & Brooks/Cole, Belmont, CA, 1989.

[74] A. M. K. P. Taylor, J. H. Whitelaw, and M. Yianneskis. Developing Flow in S-Shaped Ducts. NASA CR-3759, February 1984.

[75] J. F. Thompson, F. C. Thames, and C. W. Mastin. Automatic Numerical Generation of Body-Fitted Curvilinear Coordinate System for Field Containing Any Number of Arbitrary Two-Dimensional Bodies. *Journal of Computational Physics*, 15:299–319, 1974.

[76] L. N. Trefethen and L. Halpern. Well-Posedness of One-Way Wave Equations and Absorbing Boundary Conditions. *Mathematics of Computation*, 47:421–435, 1986.

[77] K.M. Wanie, Hirschel E.H., and Schmatz M.A. *Analysis of Numerical Solutions for Three-Dimensional Lifting Wing Flows.* Z. Flugwiss. Weltraumforsch. 15 (107-118). Springer-Verlag, 1991.

[78] R. P. Weston. *Refinement of a Method for Determining the Induced and Profile Drag of a Finite Wing from Detailed Wake Measurements.* PhD thesis, University of Florida, 1981.

[79] R. W. Williams. Computational Study of Duct and Pipe Flow Using the Method of Pseudo-Compressibility. AIAA Paper 91-1656-CP, 1991.

Appendix A

Mesh Generation

The meshes used in this work are generated by solving a set of partial differential equations (PDE) of Poisson type using an iterative procedure, see Reference [75] for instance. A boundary-conforming procedure using a structured mesh is employed which consists of mapping the three-dimensional physical domain of Cartesian coordinates (x, y, z) onto a cubic computational domain of coordinates (ξ,η,ζ). The first step in defining the transformation is to specify Dirichlet and/or Neumann conditions on the limiting surfaces of the physical region; those boundaries being represented by a constant ξ or η or ζ in the computational domain. In the elliptic partial differential method, the distribution of the interior grid points is then governed by the following Poisson system

$$\begin{aligned} \xi_{xx} + \xi_{yy} + \xi_{zz} &= P(\xi, \eta, \zeta) \\ \eta_{xx} + \eta_{yy} + \eta_{zz} &= Q(\xi, \eta, \zeta) \\ \zeta_{xx} + \zeta_{yy} + \zeta_{zz} &= R(\xi, \eta, \zeta), \end{aligned} \tag{A.1}$$

where P, Q and R are source terms that can be selected to control the mesh points distribution. Since it is much easier to solve a system of PDE on the uniformly spaced grid of the computational domain, it is useful to transform system (A.1) onto the computational space. This is achieved by interchanging the roles of the dependent (ξ,η,ζ) and independent (x, y, z) variables in Eqs. (A.1). This yields an elliptic system of quasi-linear equations that can be written in the vector form [65]

$$a_{11}(\vec{r}_{\xi\xi}+\phi\vec{r}_{\xi}) + a_{22}(\vec{r}_{\eta\eta}+\psi\vec{r}_{\eta}) + a_{33}(\vec{r}_{\zeta\zeta}+\lambda\vec{r}_{\zeta}) + 2(a_{12}\vec{r}_{\xi\eta}+a_{13}\vec{r}_{\xi\zeta}+a_{23}\vec{r}_{\eta\zeta}) = 0, \tag{A.2}$$

where $\vec{r} = (x, y, z)$ is the position vector,

$$a_{ij} = \sum_{m=1}^{3} A_{mi} A_{mj}$$

and A_{mi} is the cofactor of the (m, i) element in the following matrix

$$M = \begin{bmatrix} x_\xi & x_\eta & x_\zeta \\ y_\xi & y_\eta & y_\zeta \\ z_\xi & z_\eta & z_\zeta \end{bmatrix}.$$

The forcing functions ϕ, ψ and λ serve to control the interior mesh points distribution,

$$\phi = \frac{J^2 P}{a_{11}}, \quad \psi = \frac{J^2 Q}{a_{22}}, \quad \lambda = \frac{J^2 R}{a_{33}}$$

where

$$J = \frac{\partial(x, y, z)}{\partial(\xi, \eta, \zeta)} = det(\vec{r}_\xi, \vec{r}_\eta, \vec{r}_\zeta).$$

Given a proper choice of the source terms P, Q and R, this transformation defines a one-to-one correspondence between the two spaces. The source terms are automatically evaluated in order to provide a control of the cell size and the skewness at the chosen domain boundaries according to the general procedure described by Sorenson [66].

Depending on the flow cases, two types of mesh generation strategies are used in this work. For internal, pipe flow computations, a 2-D version of the elliptic PDE solver is used to generate the cross flow grid planes, i.e. surfaces approximately normal to the main flow direction. These mesh planes are then arranged in the streamwise direction along the bend, see Figure A.1. Notice that due to the symmetry of the problem, only half of the pipe geometry is actually needed. The 2-D mesh plane is defined by a mixed O-H topology, see Figure A.1, with the O mesh located close to the pipe wall, allowing for a good control of the spacing and orthogonality at this boundary. The mesh singularity at the center of the pipe is removed by filling this part with an H-type grid. The connection between the two types of grid generates two field nodes that are surrounded by only three cells instead of four. As discussed in Section 3.3 this leads to a minor change in the numerical algorithm. However, this mixed grid

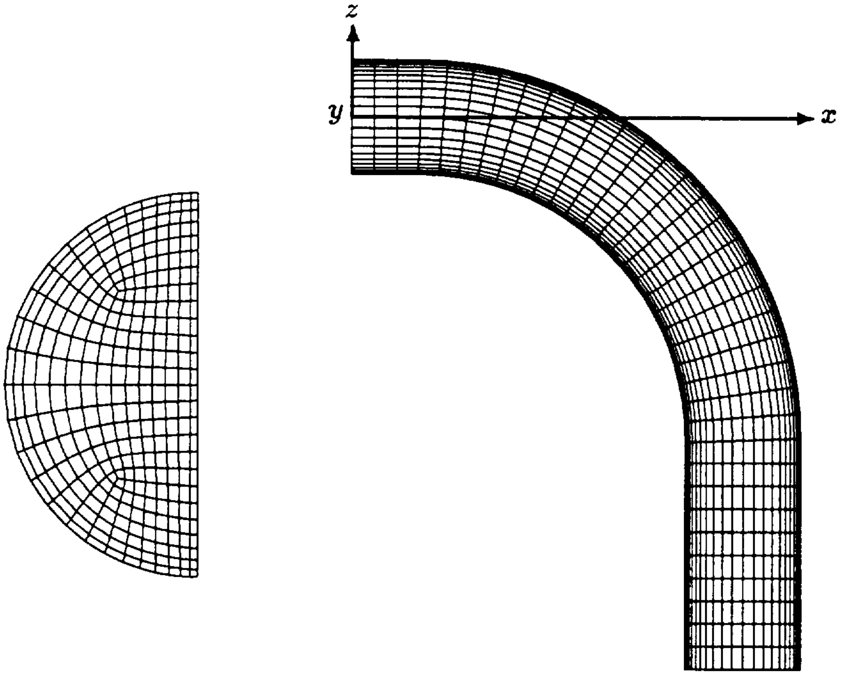


Figure A.1: Cross-sectional view of grid and distribution of cross-sections along the pipe.

topology provides much more flexibility in terms of nodes distribution than a single O- or H-type. Also this removes any mesh singularity from the pipe boundary and the symmetry plane. Notice that the two singular field nodes are allowed to move during the iterative solution procedure according to the number of nodes allocated for the H and O parts, and the grid control parameters set at the wall and the symmetry surface. In fact the position of these singular nodes is computed as an average of the position of the three surrounding nodes, thus enforcing identical cell volumes in this area.

For these pipe geometries, grid control is applied on the solid wall as well as at the symmetry surface. This provides a good control of the discrete representation of the boundary-layer velocity profile near the wall. Grid control at the symmetry surface is required in order to get a good representation of the secondary vortex generated close to the inside wall. This vortex is then moving along the symmetry plane due to the presence of its image and the wall.

For the wing flow computation of Chapter 12, the full set of 3-D elliptic equations are solved to define the interior domain nodes distribution. The initial distribution of nodes is obtained by setting up the wing cross-sectional planes perpendicularly aroung the airfoil. As an option, the elliptic procedure can also be performed in two dimensions for each of the cross-sections, to define a smoother initial condition for the three-dimensional elliptic grid solution. For this case a C-H type of mesh is used. The C part is required for a good resolution of the strong gradients at the leading edge and the wake flow. The distribution of the nodes on the wing surface is established first using multiple cubic spline functions along the span and the chord. This allows to cluster the nodes near the leading and the trailing edge as well as close to the wing tip, see Figures A.3 and 12.2. A set of mixed Dirichlet/von Neumann boundary conditions are used to define the symmetry plane and the outer surface. Control of the spacing and the orthogonality is applied on the wing surface, in the wake portion (i.e. from the trailing edge to the exit) and also on the outer surface. The use of a symmetric profile allows for the generation of only one half (upper or lower) of the mesh, which is then duplicated and finally tilted upwards from the trailing edge to the exit to follow approximately the upward movement of the wake and trailing vortex, see Figure 12.3.

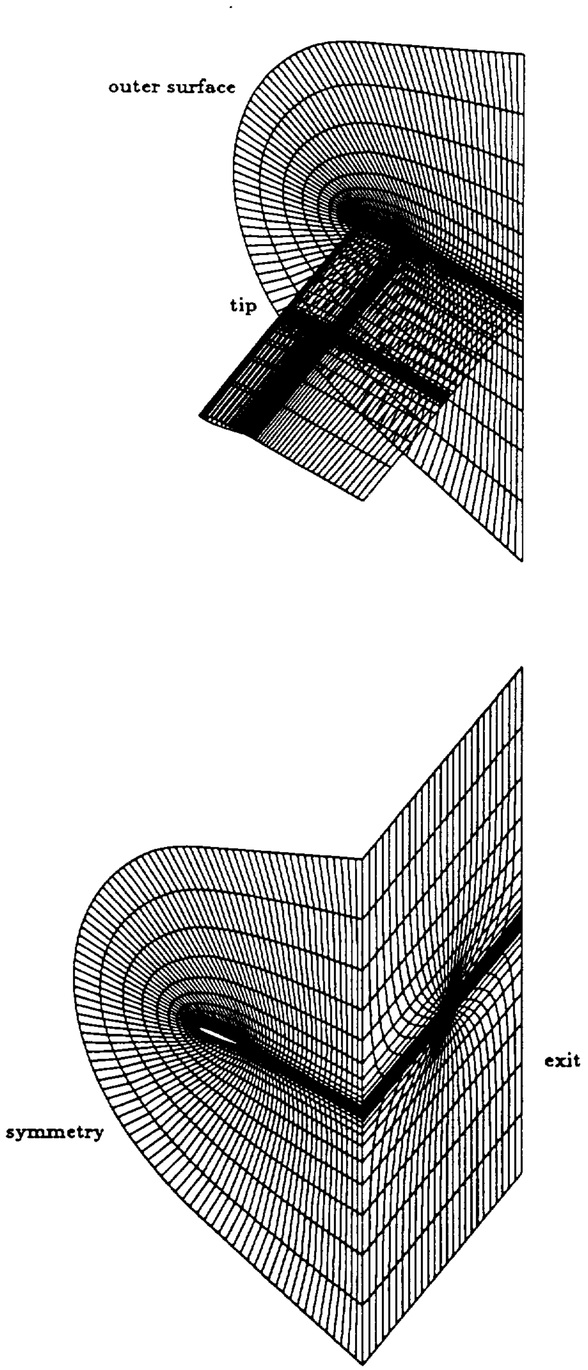


Figure A.2: C-H mesh structure shown by different mesh surfaces.

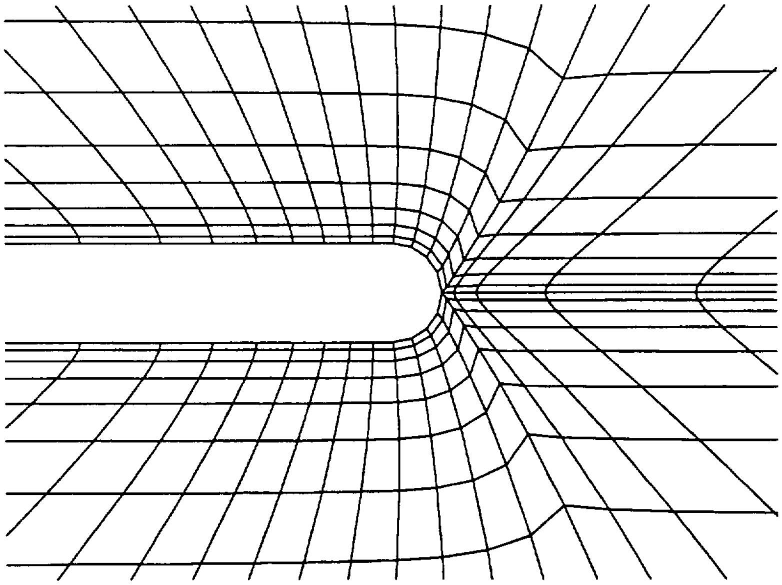


Figure A.3: Detail of wing-tip region for a streamwise cross-section located at $\sim 50\%$ chord with clustering in the direction perpendicular to the wall and near the tip (the derivatives are also prescribed using the source-terms).

Appendix B

Volume and area calculations

This appendix presents the computation of the cell volumes, the faces area and the volumes associated with the nodes which are required by the Lax-Wendroff algorithm.

The volume V_c of a cell is computed as the sum of the volumes of the five constituent tetrahedra shown on Figure B.1.

$$V_c = V_{5627} + V_{5214} + V_{5874} + V_{2734} + V_{5274}, \tag{B.1}$$

where the indices refer to the nodes constituting each tetrahedron. The volume of a tetrahedron is found through a determinant constructed by the tetrahedron's vertices. For example, the volume of tetrahedron 5627 is found as

$$V_{5627} = - \begin{vmatrix} 1 & x5 & y5 & z5 \\ 1 & x6 & y6 & z6 \\ 1 & x2 & y2 & z2 \\ 1 & x7 & y7 & z7 \end{vmatrix} .$$

The area of a cell face is given by half the cross product of its diagonals. The surface vectors are chosen to point outwards and are written as

$$\begin{aligned} \vec{S}_1 &= \frac{1}{2}(\vec{r}_3 - \vec{r}_6) \times (\vec{r}_7 - \vec{r}_2) \\ \vec{S}_2 &= \frac{1}{2}(\vec{r}_8 - \vec{r}_1) \times (\vec{r}_4 - \vec{r}_5) \\ \vec{S}_3 &= \frac{1}{2}(\vec{r}_7 - \vec{r}_4) \times (\vec{r}_3 - \vec{r}_8) \end{aligned}$$

$$\begin{aligned} \vec{S}_4 &= \frac{1}{2}(\vec{r}_6 - \vec{r}_1) \times (\vec{r}_5 - \vec{r}_2) \\ \vec{S}_5 &= \frac{1}{2}(\vec{r}_7 - \vec{r}_5) \times (\vec{r}_8 - \vec{r}_6) \\ \vec{S}_6 &= \frac{1}{2}(\vec{r}_3 - \vec{r}_1) \times (\vec{r}_2 - \vec{r}_4) \end{aligned} \tag{B.2}$$

Figure B.1 shows the vector surface numbering $\vec{S}_1$ to $\vec{S}_6$.

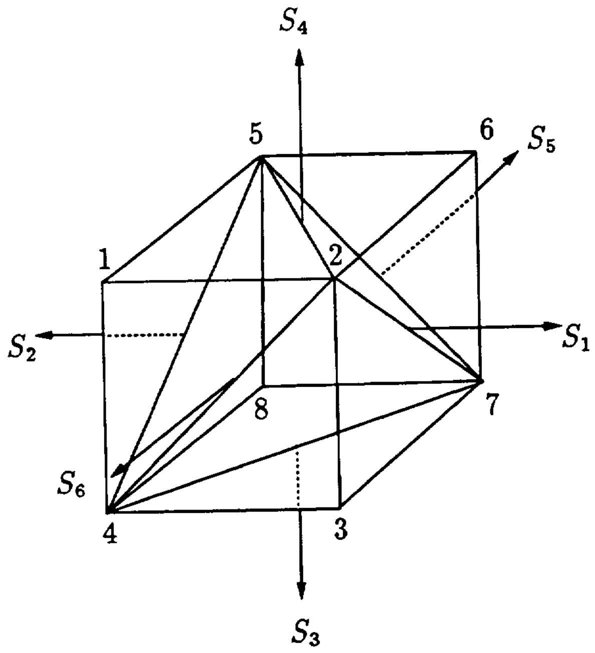


Figure B.1: Dividing of cell into five tetrahedra and surface vectors numbering definition.

The Lax-Wendroff algorithm also uses volumes associated with nodes. A 'node volume' is defined at node n as the average of the volumes of the eight surrounding cells.

$$V_n = \frac{1}{8} \sum_{8\text{cells}} V_{\text{cell}}. \tag{B.3}$$

In the case of nodes lying on boundary surfaces like a wall surface or an inlet/exit surface, only four cells or less (corner nodes) contribute to the volume associated with the node. For a wall surface case, this is taken into account when applying the wall boundary condition as mentioned in Section 3.5.

Because of the symmetry condition, the volume associated with a node lying on a symmetry surface is found by doubling the node volume contribution from the four existing boundary cells.

Appendix C

Stability analysis

In order to perform a stability analysis on the Euler equations, the system is first recast in primitive form and computational coordinates.

C.1 Primitive form of Euler equations in computational coordinates

Starting from the Euler equations in conservative form and Cartesian coordinates

$$\frac{\partial U}{\partial t}+\frac{\partial F}{\partial x}+\frac{\partial G}{\partial y}+\frac{\partial H}{\partial z}=0, \tag{C.1}$$

and applying the chain rule derivation to the above equations gives

$$\frac{\partial U}{\partial t}+\frac{\partial F}{\partial \xi}\xi_x+\frac{\partial F}{\partial \eta}\eta_x+\frac{\partial F}{\partial \zeta}\zeta_x+\frac{\partial G}{\partial \xi}\xi_y+\frac{\partial G}{\partial \eta}\eta_y+\frac{\partial G}{\partial \zeta}\zeta_y+\frac{\partial H}{\partial \xi}\xi_z+\frac{\partial H}{\partial \eta}\eta_z+\frac{\partial H}{\partial \zeta}\zeta_z=0, \tag{C.2}$$

where index $_x$ is short for derivative with respect to x, and so on for y and z. J is the Jacobian of the transformation from the (x,y,z) physical space to the (ξ,η,ζ) computational space and is defined as

$$J=x_\xi(y_\eta z_\zeta-y_\zeta z_\eta)-x_\eta(y_\xi z_\zeta-y_\zeta z_\xi)+x_\zeta(y_\xi z_\eta-y_\eta z_\xi). \tag{C.3}$$

The metrics of the transformation are

$$\xi_x \;=\; \frac{y_\eta z_\zeta-y_\zeta z_\eta}{J},\quad \xi_y=-\frac{x_\eta z_\zeta-x_\zeta z_\eta}{J},\quad \xi_z=\frac{x_\eta y_\zeta-x_\zeta y_\eta}{J}, \tag{C.4}$$

$$\eta_x = \frac{y_\xi z_\zeta - y_\zeta z_\xi}{J}, \quad \eta_y = -\frac{x_\xi z_\zeta - x_\zeta z_\xi}{J}, \quad \eta_z = \frac{x_\xi y_\zeta - x_\zeta y_\xi}{J}, \tag{C.5}$$

$$\zeta_x = \frac{y_\xi z_\eta - y_\eta z_\xi}{J}, \quad \zeta_y = -\frac{x_\xi z_\eta - x_\eta z_\xi}{J}, \quad \zeta_z = \frac{x_\xi y_\eta - x_\eta y_\xi}{J}. \tag{C.6}$$

Noticing that

$$\begin{aligned}
\frac{\partial}{\partial\xi}(y_\eta z_\zeta - y_\zeta z_\eta) &- \frac{\partial}{\partial\eta}(y_\xi z_\zeta - y_\zeta z_\xi) - \frac{\partial}{\partial\zeta}(y_\xi z_\eta - y_\eta z_\xi) = 0,\\
\frac{\partial}{\partial\xi}(x_\eta z_\zeta - x_\zeta z_\eta) &- \frac{\partial}{\partial\eta}(x_\xi z_\zeta - x_\zeta z_\xi) - \frac{\partial}{\partial\zeta}(x_\xi z_\eta - x_\eta z_\xi) = 0,\\
\frac{\partial}{\partial\xi}(x_\eta y_\zeta - x_\zeta y_\eta) &- \frac{\partial}{\partial\eta}(x_\xi y_\zeta - x_\zeta y_\xi) - \frac{\partial}{\partial\zeta}(x_\xi y_\eta - x_\eta y_\xi) = 0,
\end{aligned}$$

the Euler equations recasted into computational coordinates are

$$\begin{aligned}
& J\frac{\partial U}{\partial t} + \frac{\partial F'}{\partial\xi} + \frac{\partial G'}{\partial\eta} + \frac{\partial H'}{\partial\zeta}\\
- & F\underbrace{\left[\frac{\partial}{\partial\xi}(y_\eta z_\zeta - y_\zeta z_\eta) - \frac{\partial}{\partial\eta}(y_\xi z_\zeta - y_\zeta z_\xi) - \frac{\partial}{\partial\zeta}(y_\xi z_\eta - y_\eta z_\xi)\right]}_{=0}\\
- & G\underbrace{\left[\frac{\partial}{\partial\xi}(x_\eta z_\zeta - x_\zeta z_\eta) - \frac{\partial}{\partial\eta}(x_\xi z_\zeta - x_\zeta z_\xi) - \frac{\partial}{\partial\zeta}(x_\xi z_\eta - x_\eta z_\xi)\right]}_{=0}\\
- & H\underbrace{\left[\frac{\partial}{\partial\xi}(x_\eta y_\zeta - x_\zeta y_\eta) - \frac{\partial}{\partial\eta}(x_\xi y_\zeta - x_\zeta y_\xi) - \frac{\partial}{\partial\zeta}(x_\xi y_\eta - x_\eta y_\xi)\right]}_{=0} = 0,
\end{aligned} \tag{C.7}$$

where F', G', H' are the contravariant fluxes defined as

$$\begin{aligned}
F' &= (y_\eta z_\zeta - y_\zeta z_\eta)F - (x_\eta z_\zeta - x_\zeta z_\eta)G + (x_\eta y_\zeta - x_\zeta y_\eta)H,\\
G' &= -(y_\xi z_\zeta - y_\zeta z_\xi)F + (x_\xi z_\zeta - x_\zeta z_\xi)G - (x_\xi y_\zeta - x_\zeta y_\xi)H,\\
H' &= (y_\xi z_\eta - y_\eta z_\xi)F - (x_\xi z_\eta - x_\eta z_\xi)G + (x_\xi y_\eta - x_\eta y_\xi)H.
\end{aligned}$$

The following relations will be used in the next section.

$$\begin{aligned}
r_1 &= u(y_\eta z_\zeta - y_\zeta z_\eta) - v(x_\eta z_\zeta - x_\zeta z_\eta) + w(x_\eta y_\zeta - x_\zeta y_\eta),\\
r_2 &= -u(y_\xi z_\zeta - y_\zeta z_\xi) + v(x_\xi z_\zeta - x_\zeta z_\xi) - w(x_\xi y_\zeta - x_\zeta y_\xi),\\
r_3 &= u(y_\xi z_\eta - y_\eta z_\xi) - v(x_\xi z_\eta - x_\eta z_\xi) + w(x_\xi y_\eta - x_\eta y_\xi),
\end{aligned}$$

whose derivatives are

$$\frac{\partial r_1}{\partial\xi} = \frac{\partial u}{\partial\xi}(y_\eta z_\zeta - y_\zeta z_\eta) - \frac{\partial v}{\partial\xi}(x_\eta z_\zeta - x_\zeta z_\eta) + \frac{\partial w}{\partial\xi}(x_\eta y_\zeta - x_\zeta y_\eta), \tag{C.8}$$

$$\begin{aligned}
\frac{\partial r_2}{\partial \eta} &= -\frac{\partial u}{\partial \eta}(y_\xi z_\zeta - y_\zeta z_\xi) + \frac{\partial v}{\partial \eta}(x_\xi z_\zeta - x_\zeta z_\xi) - \frac{\partial w}{\partial \eta}(x_\xi y_\zeta - x_\zeta y_\xi), \qquad \text{(C.9)} \\
\frac{\partial r_3}{\partial \zeta} &= \frac{\partial u}{\partial \zeta}(y_\xi z_\eta - y_\eta z_\xi) - \frac{\partial v}{\partial \zeta}(x_\xi z_\eta - x_\eta z_\xi) + \frac{\partial w}{\partial \zeta}(x_\xi y_\eta - x_\eta y_\xi). \qquad \text{(C.10)}
\end{aligned}$$

C.1.1 Primitive form: incompressible flow

In the case of an incompressible flow, the state vector U and the contravariant fluxes F', G', H' are

$$U = \begin{pmatrix} p^* \\ u \\ v \\ w \end{pmatrix}, \quad F' = \begin{pmatrix} c_a^2 r_1 \\ r_1 u + p^*(y_\eta z_\zeta - y_\zeta z_\eta) \\ r_1 v - p^*(x_\eta z_\zeta - x_\zeta z_\eta) \\ r_1 w - p^*(x_\eta y_\zeta - x_\zeta y_\eta) \end{pmatrix},$$

$$G' = \begin{pmatrix} c_a^2 r_2 \\ r_2 u - p^*(y_\xi z_\zeta - y_\zeta z_\xi) \\ r_2 v + p^*(x_\xi z_\zeta - x_\zeta z_\xi) \\ r_2 w - p^*(x_\xi y_\zeta - x_\zeta y_\xi) \end{pmatrix}, \quad H' = \begin{pmatrix} c_a^2 r_3 \\ r_3 u + p^*(y_\xi z_\eta - y_\eta z_\xi) \\ r_3 v - p^*(x_\xi z_\eta - x_\eta z_\xi) \\ r_3 w - p^*(x_\xi y_\eta - x_\eta y_\xi) \end{pmatrix}.$$

The continuity equation is

$$J\frac{\partial p^*}{\partial t} = -c_a^2\left(\frac{\partial r_1}{\partial \xi} + \frac{\partial r_2}{\partial \eta} + \frac{\partial r_3}{\partial \zeta}\right), \qquad \text{(C.11)}$$

where the RHS is found from Equations (C.8) to (C.10). The first component of the momentum gives

$$\begin{aligned}
J\frac{\partial u}{\partial t} = &- r_1\frac{\partial u}{\partial \xi} - r_2\frac{\partial u}{\partial \eta} - r_3\frac{\partial u}{\partial \zeta} - u\left(\frac{\partial r_1}{\partial \xi} + \frac{\partial r_2}{\partial \eta} + \frac{\partial r_3}{\partial \zeta}\right) \\
&- (y_\eta z_\zeta - y_\zeta z_\eta)\frac{\partial p^*}{\partial \xi} + (y_\xi z_\zeta - y_\zeta z_\xi)\frac{\partial p^*}{\partial \eta} - (y_\xi z_\eta - y_\eta z_\xi)\frac{\partial p^*}{\partial \zeta} \\
&- p^*\underbrace{\left(\frac{\partial}{\partial \xi}(y_\eta z_\zeta - y_\zeta z_\eta) - \frac{\partial}{\partial \eta}(y_\xi z_\zeta - y_\zeta z_\xi) + \frac{\partial}{\partial \zeta}(y_\xi z_\eta - y_\eta z_\xi)\right)}_{=0}. \qquad \text{(C.12)}
\end{aligned}$$

Similarly, one finds for the two other components of the momentum equation

$$
\begin{aligned}
J\rho\frac{\partial v}{\partial t} = & - r_1\frac{\partial v}{\partial \xi} - r_2\frac{\partial v}{\partial \eta} - r_3\frac{\partial v}{\partial \zeta} - v\left(\frac{\partial r_1}{\partial \xi} + \frac{\partial r_2}{\partial \eta} + \frac{\partial r_3}{\partial \zeta}\right) \\
& + (x_\eta z_\zeta - x_\zeta z_\eta)\frac{\partial p^*}{\partial \xi} - (x_\xi z_\zeta - x_\zeta z_\xi)\frac{\partial p^*}{\partial \eta} + (x_\xi z_\eta - x_\eta z_\xi)\frac{\partial p^*}{\partial \zeta}, \quad \text{(C.13)}
\end{aligned}
$$

$$
\begin{aligned}
J\rho\frac{\partial w}{\partial t} = & - r_1\frac{\partial w}{\partial \xi} - r_2\frac{\partial w}{\partial \eta} - r_3\frac{\partial w}{\partial \zeta} - w\left(\frac{\partial r_1}{\partial \xi} + \frac{\partial r_2}{\partial \eta} + \frac{\partial r_3}{\partial \zeta}\right) \\
& - (x_\eta y_\zeta - x_\zeta y_\eta)\frac{\partial p^*}{\partial \xi} + (x_\xi y_\zeta - x_\zeta y_\xi)\frac{\partial p^*}{\partial \eta} - (x_\xi y_\eta - x_\eta y_\xi)\frac{\partial p^*}{\partial \zeta}. \quad \text{(C.14)}
\end{aligned}
$$

Collecting Equations (C.11) to (C.14) gives the Euler equations modified by the artificial compressiblity concept and written in primitive form and computational coordinates

$$
\frac{\partial U}{\partial t} + J^{-1}\left(A_1\frac{\partial U}{\partial \xi} + A_2\frac{\partial U}{\partial \eta} + A_3\frac{\partial U}{\partial \zeta}\right) = 0, \qquad \text{where} \qquad U = \begin{pmatrix} p^* \\ u \\ v \\ w \end{pmatrix},
$$

$$
A_1 = \begin{pmatrix}
0 & c_a^2(y_\eta z_\zeta - y_\zeta z_\eta) & -c_a^2(x_\eta z_\zeta - x_\zeta z_\eta) & c_a^2(x_\eta y_\zeta - x_\zeta y_\eta) \\
(y_\eta z_\zeta - y_\zeta z_\eta) & u(y_\eta z_\zeta - y_\zeta z_\eta) + r_1 & -u(x_\eta z_\zeta - x_\zeta z_\eta) & u(x_\eta y_\zeta - x_\zeta y_\eta) \\
-(x_\eta z_\zeta - x_\zeta z_\eta) & v(y_\eta z_\zeta - y_\zeta z_\eta) & -v(x_\eta z_\zeta - x_\zeta z_\eta) + r_1 & v(x_\eta y_\zeta - x_\zeta y_\eta) \\
(x_\eta y_\zeta - x_\zeta y_\eta) & w(y_\eta z_\zeta - y_\zeta z_\eta) & -w(x_\eta z_\zeta - x_\zeta z_\eta) & w(x_\eta y_\zeta - x_\zeta y_\eta) + r_1
\end{pmatrix},
$$

$$
A_2 = \begin{pmatrix}
0 & -c_a^2(y_\xi z_\zeta - y_\zeta z_\xi) & c_a^2(x_\xi z_\zeta - x_\zeta z_\xi) & -c_a^2(x_\xi y_\zeta - x_\zeta y_\xi) \\
-(y_\xi z_\zeta - y_\zeta z_\xi) & -u(y_\xi z_\zeta - y_\zeta z_\xi) + r_2 & u(x_\xi z_\zeta - x_\zeta z_\xi) & -u(x_\xi y_\zeta - x_\zeta y_\xi) \\
(x_\xi z_\zeta - x_\zeta z_\xi) & -v(y_\xi z_\zeta - y_\zeta z_\xi) & v(x_\xi z_\zeta - x_\zeta z_\xi) + r_2 & -v(x_\xi y_\zeta - x_\zeta y_\xi) \\
-(x_\xi y_\zeta - x_\zeta y_\xi) & -w(y_\xi z_\zeta - y_\zeta z_\xi) & w(x_\xi z_\zeta - x_\zeta z_\xi) & -w(x_\xi y_\zeta - x_\zeta y_\xi) + r_2
\end{pmatrix},
$$

$$
A_3 = \begin{pmatrix}
0 & c_a^2(y_\xi z_\eta - y_\eta z_\xi) & -c_a^2(x_\xi z_\eta - x_\eta z_\xi) & c_a^2(x_\xi y_\eta - x_\eta y_\xi) \\
(y_\xi z_\eta - y_\eta z_\xi) & u(y_\xi z_\eta - y_\eta z_\xi) + r_3 & -u(x_\xi z_\eta - x_\eta z_\xi) & u(x_\xi y_\eta - x_\eta y_\xi) \\
-(x_\xi z_\eta - x_\eta z_\xi) & v(y_\xi z_\eta - y_\eta z_\xi) & -v(x_\xi z_\eta - x_\eta z_\xi) + r_3 & v(x_\xi y_\eta - x_\eta y_\xi) \\
(x_\xi y_\eta - x_\eta y_\xi) & w(y_\xi z_\eta - y_\eta z_\xi) & -w(x_\xi z_\eta - x_\eta z_\xi) & w(x_\xi y_\eta - x_\eta y_\xi) + r_3
\end{pmatrix}.
$$

C.1.2 Primitive form: compressible flow

In the case of a compressible flow the state vector U and the contravariant fluxes F', G', H' are

$$U = \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ \rho e \end{pmatrix}, \quad F' = \begin{pmatrix} \rho r_1 \\ \rho r_1 u + p(y_\eta z_\zeta - y_\zeta z_\eta) \\ \rho r_1 v - p(x_\eta z_\zeta - x_\zeta z_\eta) \\ \rho r_1 w + p(x_\eta y_\zeta - x_\zeta y_\eta) \\ r_1(\rho e_0 + p) \end{pmatrix}, \quad G' = \begin{pmatrix} \rho r_2 \\ \rho r_2 u - p(y_\xi z_\zeta - y_\zeta z_\xi) \\ \rho r_2 v + p(x_\xi z_\zeta - x_\zeta z_\xi) \\ \rho r_2 w - p(x_\xi y_\zeta - x_\zeta y_\xi) \\ r_2(\rho e_0 + p) \end{pmatrix},$$

$$H' = \begin{pmatrix} \rho r_3 \\ \rho r_3 u + p(y_\xi z_\eta - y_\eta z_\xi) \\ \rho r_3 v - p(x_\xi z_\eta - x_\eta z_\xi) \\ \rho r_3 w + p(x_\xi y_\eta - x_\eta y_\xi) \\ r_3(\rho e_0 + p) \end{pmatrix}.$$

The continuity equation is

$$\begin{aligned} J\frac{\partial \rho}{\partial t} &= -\frac{\partial}{\partial \xi}(\rho r_1) - \frac{\partial}{\partial \eta}(\rho r_2) - \frac{\partial}{\partial \zeta}(\rho r_3) \\ &= -r_1\frac{\partial \rho}{\partial \xi} - r_2\frac{\partial \rho}{\partial \eta} - r_3\frac{\partial \rho}{\partial \zeta} - \rho\left(\frac{\partial r_1}{\partial \xi} + \frac{\partial r_2}{\partial \eta} + \frac{\partial r_3}{\partial \zeta}\right). \end{aligned} \qquad (C.15)$$

The first component of momentum is

$$J\rho\frac{\partial u}{\partial t} = J\left(\frac{\partial}{\partial t}(\rho u) - u\frac{\partial \rho}{\partial t}\right). \qquad (C.16)$$

Replacing the expression for $\partial(\rho u)/\partial t$ and using Equation (C.15) for $\partial \rho/\partial t$, the first component of momentum becomes

$$J\rho\frac{\partial u}{\partial t} = -\rho r_1\frac{\partial u}{\partial \xi} - \rho r_2\frac{\partial u}{\partial \eta} - \rho r_3\frac{\partial u}{\partial \zeta} - \frac{\partial p}{\partial \xi}(y_\eta z_\zeta - y_\zeta z_\eta) + \frac{\partial p}{\partial \eta}(y_\xi z_\zeta - y_\zeta z_\xi) - \frac{\partial p}{\partial \zeta}(y_\xi z_\eta - y_\eta z_\xi). \qquad (C.17)$$

Similarly, one finds for the two other components of the momentum equation

$$J\rho\frac{\partial v}{\partial t} = -\rho r_1\frac{\partial v}{\partial \xi}-\rho r_2\frac{\partial v}{\partial \eta}-\rho r_3\frac{\partial v}{\partial \zeta}+\frac{\partial p}{\partial \xi}(x_\eta z_\zeta-x_\zeta z_\eta)-\frac{\partial p}{\partial \eta}(x_\xi z_\zeta-x_\zeta z_\xi)+\frac{\partial p}{\partial \zeta}(x_\xi z_\eta-x_\eta z_\xi), \tag{C.18}$$

$$J\rho\frac{\partial w}{\partial t} = -\rho r_1\frac{\partial w}{\partial \xi}-\rho r_2\frac{\partial w}{\partial \eta}-\rho r_3\frac{\partial w}{\partial \zeta}-\frac{\partial p}{\partial \xi}(x_\eta y_\zeta-x_\zeta y_\eta)+\frac{\partial p}{\partial \eta}(x_\xi y_\zeta-x_\zeta y_\xi)-\frac{\partial p}{\partial \zeta}(x_\xi y_\eta-x_\eta y_\xi). \tag{C.19}$$

By taking the time-derivative of the equation of state we get

$$J\frac{1}{\gamma-1}\frac{\partial p}{\partial t} = J\left(\frac{\partial}{\partial t}(\rho E)-\frac{1}{2}(u^2+v^2+w^2)\frac{\partial \rho}{\partial t}-\rho u\frac{\partial u}{\partial t}-\rho v\frac{\partial v}{\partial t}-\rho w\frac{\partial w}{\partial t}\right). \tag{C.20}$$

Replacing the expressions for $\partial(\rho E)/\partial t$ and using Equations (C.15) and (C.17), (C.18) and (C.19) we get

$$J\frac{\partial p}{\partial t} = -r_1\frac{\partial p}{\partial \xi}-r_2\frac{\partial p}{\partial \eta}-r_3\frac{\partial p}{\partial \zeta}-\gamma p\left(\frac{\partial r_1}{\partial \xi}+\frac{\partial r_2}{\partial \eta}+\frac{\partial r_3}{\partial \zeta}\right). \tag{C.21}$$

Thus, the Euler equations for a compressible flow written in primitive form and in computational coordinates are

$$\frac{\partial U_p}{\partial t}+J^{-1}\left(A_1\frac{\partial U_p}{\partial \xi}+A_2\frac{\partial U_p}{\partial \eta}+A_3\frac{\partial U_p}{\partial \zeta}\right)=0, \qquad \text{where} \qquad U_p=\begin{pmatrix}\rho\\ u\\ v\\ w\\ p\end{pmatrix},$$

$$A_1=\begin{pmatrix} r_1 & \rho(y_\eta z_\zeta-y_\zeta z_\eta) & -\rho(x_\eta z_\zeta-x_\zeta z_\eta) & \rho(x_\eta y_\zeta-x_\zeta y_\eta) & 0\\ 0 & r_1 & 0 & 0 & \frac{1}{\rho}(y_\eta z_\zeta-y_\zeta z_\eta)\\ 0 & 0 & r_1 & 0 & -\frac{1}{\rho}(x_\eta z_\zeta-x_\zeta z_\eta)\\ 0 & 0 & 0 & r_1 & \frac{1}{\rho}(x_\eta y_\zeta-x_\zeta y_\eta)\\ 0 & \gamma p(y_\eta z_\zeta-y_\zeta z_\eta) & -\gamma p(x_\eta z_\zeta-x_\zeta z_\eta) & \gamma p(x_\eta y_\zeta-x_\zeta y_\eta) & r_1 \end{pmatrix},$$

$$A_2 = \begin{pmatrix} r_2 & -\rho(y_\xi z_\zeta - y_\zeta z_\xi) & \rho(x_\xi z_\zeta - x_\zeta z_\xi) & -\rho(x_\xi y_\zeta - x_\zeta y_\xi) & 0 \\ 0 & r_2 & 0 & 0 & -\frac{1}{\rho}(y_\xi z_\zeta - y_\zeta z_\xi) \\ 0 & 0 & r_2 & 0 & \frac{1}{\rho}(x_\xi z_\zeta - x_\zeta z_\xi) \\ 0 & 0 & 0 & r_2 & -\frac{1}{\rho}(x_\xi y_\zeta - x_\zeta y_\xi) \\ 0 & -\gamma p(y_\xi z_\zeta - y_\zeta z_\xi) & \gamma p(x_\xi z_\zeta - x_\zeta z_\xi) & -\gamma p(x_\xi y_\zeta - x_\zeta y_\xi) & r_2 \end{pmatrix},$$

$$A_3 = \begin{pmatrix} r_3 & \rho(y_\xi z_\eta - y_\eta z_\xi) & -\rho(x_\xi z_\eta - x_\eta z_\xi) & \rho(x_\xi y_\eta - x_\eta y_\xi) & 0 \\ 0 & r_3 & 0 & 0 & \frac{1}{\rho}(y_\xi z_\eta - y_\eta z_\xi) \\ 0 & 0 & r_3 & 0 & -\frac{1}{\rho}(x_\xi z_\eta - x_\eta z_\xi) \\ 0 & 0 & 0 & r_3 & \frac{1}{\rho}(x_\xi y_\eta - x_\eta y_\xi) \\ 0 & \gamma p(y_\xi z_\eta - y_\eta z_\xi) & -\gamma p(x_\xi z_\eta - x_\eta z_\xi) & \gamma p(x_\xi y_\eta - x_\eta y_\xi) & r_3 \end{pmatrix}.$$

C.2 Stability

The discretization following Ni's method gives

$$\begin{aligned} U_p^{n+1} = U_p^n + \Delta t \mu_\zeta \mu_\eta \mu_\xi \; (A_1 \mu_\zeta \mu_\eta \delta_\xi + A_2 \mu_\zeta \mu_\xi \delta_\eta + A_3 \mu_\eta \mu_\xi \delta_\zeta) \; U_p^n + \\ \frac{1}{2} \Delta t^2 \; (A_1 \mu_\zeta \mu_\eta \delta_\xi + A_2 \mu_\zeta \mu_\xi \delta_\eta + A_3 \mu_\eta \mu_\xi \delta_\zeta)^2 \; U_p^n, \end{aligned} \quad \text{(C.22)}$$

where the operators δ and μ are defined by

$$\delta_\xi U_{ijk} = U_{i+\frac{1}{2}jk} - U_{i-\frac{1}{2}jk}, \quad \delta_\eta U_{ijk} = U_{ij+\frac{1}{2}k} - U_{ij-\frac{1}{2}k}, \quad \text{(C.23)}$$

$$\delta_\zeta U_{ijk} = U_{ijk+\frac{1}{2}} - U_{ijk-\frac{1}{2}}, \quad \text{(C.24)}$$

$$\mu_\xi U_{ijk} = \frac{1}{2}(U_{i+\frac{1}{2}jk} + U_{i-\frac{1}{2}jk}), \quad \mu_\eta U_{ijk} = \frac{1}{2}(U_{ij+\frac{1}{2}k} + U_{ij-\frac{1}{2}k}), \quad \text{(C.25)}$$

$$\mu_\zeta U_{ijk} = \frac{1}{2}(U_{ijk+\frac{1}{2}} + U_{ijk-\frac{1}{2}}). \quad \text{(C.26)}$$

Considering a wave-like solution $U_p^n = \overline{U_p^n} e^{i(j\theta_1 + k\theta_2 + m\theta_3)}$, the amplification matrix G is

$$G(\theta_1, \theta_2, \theta_3) = I - 2i\ \cos\frac{\theta_1}{2}\cos\frac{\theta_2}{2}\cos\frac{\theta_3}{2}A_{123} + 2\Delta t^2 A_{123}^2, \qquad \text{(C.27)}$$

where

$$A_{123} = A_1\ \sin\frac{\theta_1}{2}\cos\frac{\theta_2}{2}\cos\frac{\theta_3}{2} + A_2\ \sin\frac{\theta_2}{2}\cos\frac{\theta_1}{2}\cos\frac{\theta_3}{2} + A_3\ \sin\frac{\theta_3}{2}\cos\frac{\theta_1}{2}\cos\frac{\theta_2}{2}. \qquad \text{(C.28)}$$

Defining

$$D = \Delta t\ (s_1 A_1 + s_2 A_2 + s_3 A_3), \qquad \text{(C.29)}$$

$$s = \sqrt{(\sin\frac{\theta_1}{2}\cos\frac{\theta_2}{2}\cos\frac{\theta_3}{2})^2 + (\cos\frac{\theta_1}{2}\sin\frac{\theta_2}{2}\cos\frac{\theta_3}{2})^2 + (\cos\frac{\theta_1}{2}\cos\frac{\theta_2}{2}\sin\frac{\theta_3}{2})^2}, \qquad \text{(C.30)}$$

$$s_1 = \frac{\sin\frac{\theta_1}{2}\cos\frac{\theta_2}{2}\cos\frac{\theta_3}{2}}{s}, \quad s_2 = \frac{\cos\frac{\theta_1}{2}\sin\frac{\theta_2}{2}\cos\frac{\theta_3}{2}}{s}, \quad s_3 = \frac{\cos\frac{\theta_1}{2}\cos\frac{\theta_2}{2}\sin\frac{\theta_3}{2}}{s}, \qquad \text{(C.31)}$$

with the property $s_1^2 + s_2^2 + s_3^2 = 1$, the amplification matrix G can be written as

$$G(\theta_1, \theta_2, \theta_3) = I - 2i\ s\ \cos\frac{\theta_1}{2}\cos\frac{\theta_2}{2}\cos\frac{\theta_3}{2}\ D - 2\ s^2\ D^2. \qquad \text{(C.32)}$$

If λ_D is an eigenvalue of D, the corresponding eigenvector is also an eigenvector of G with an associated eigenvalue

$$\lambda_G = 1 - 2i\ s\ \cos\frac{\theta_1}{2}\cos\frac{\theta_2}{2}\cos\frac{\theta_3}{2}\lambda_D - 2s^2{\lambda_D}^2. \qquad \text{(C.33)}$$

If λ_D is real and $|\lambda_D| \leq 1$, then

$$\begin{aligned}
|\lambda_G|^2 &= 1 - 4s^2\lambda_D^2\left(1 - s^2\lambda_D^2 - {\cos\frac{\theta_1}{2}}^2{\cos\frac{\theta_2}{2}}^2{\cos\frac{\theta_3}{2}}^2\right) \\
&\leq 1 - 4s^2\lambda_D^2\left(1 - s^2 - {\cos\frac{\theta_1}{2}}^2{\cos\frac{\theta_2}{2}}^2{\cos\frac{\theta_3}{2}}^2\right) \\
&= 1 - 4s^2\lambda_D^2\left({\sin\frac{\theta_1}{2}}^2{\sin\frac{\theta_2}{2}}^2{\cos\frac{\theta_3}{2}}^2 + {\cos\frac{\theta_1}{2}}^2{\sin\frac{\theta_2}{2}}^2{\sin\frac{\theta_3}{2}}^2 + {\sin\frac{\theta_1}{2}}^2{\cos\frac{\theta_2}{2}}^2{\sin\frac{\theta_3}{2}}^2\right) \\
&\leq 1.
\end{aligned}$$

Thus the condition $\lambda_D \leq 1$ is sufficient (but may be not necessary) for stability.

C.2.1 Stability: incompressible flow

Let

$$\begin{aligned} r &= r_1 s_1 + r_2 s_2 + r_3 s_3, && \text{(C.34)} \\ a &= (y_\eta z_\zeta - y_\zeta z_\eta)s_1 - (y_\xi z_\zeta - y_\zeta z_\xi)s_2 + (y_\xi z_\eta - y_\eta z_\xi)s_3, && \text{(C.35)} \\ b &= -(x_\eta z_\zeta - x_\zeta z_\eta)s_1 + (x_\xi z_\zeta - x_\zeta z_\xi)s_2 - (x_\xi z_\eta - x_\eta z_\xi)s_3, && \text{(C.36)} \\ d &= (x_\eta y_\zeta - x_\zeta y_\xi)s_1 - (x_\xi y_\zeta - x_\zeta y_\xi)s_2 + (x_\xi y_\eta - x_\eta y_\xi)s_3. && \text{(C.37)} \end{aligned}$$

Using Equation (C.29) and the previous definitions for r, a, b and d, the matrix D is written as

$$D = \begin{pmatrix} 0 & c_a^2 a & c_a^2 b & c_a^2 d \\ a & ua + r & ub & ud \\ b & va & vb + r & vd \\ d & wa & wb & wd + r \end{pmatrix}.$$

The eigenvalues of D are

$$\frac{\Delta t}{J}\left(r,\ r,\ r + \sqrt{r^2 + c_a^2(a^2 + b^2 + d^2)},\ r - \sqrt{r^2 + c_a^2(a^2 + b^2 + d^2)}\right). \qquad \text{(C.38)}$$

A conservative estimate for the maximum eigenvalue is

$$\lambda_{D_{max}} \leq \frac{\Delta t}{J}\left(\sqrt{r_1^2 + r_2^2 + r_3^2} + \sqrt{r_1^2 + r_2^2 + r_3^2 + c_a^2(\overline{a}^2 + \overline{b}^2 + \overline{d}^2)}\right), \qquad \text{(C.39)}$$

where the value of r has been maximized by $\sqrt{r_1^2 + r_2^2 + r_3^2}$ since $s_1^2 + s_2^2 + s_3^2 = 1$, and the values of a^2, b^2, d^2 have been maximized by $\overline{a}^2, \overline{b}^2, \overline{d}^2$ with

$$\begin{aligned} \overline{a}^2 &= (y_\eta z_\zeta - y_\zeta z_\eta)^2 + (y_\xi z_\zeta - y_\zeta z_\xi)^2 + (y_\xi z_\eta - y_\eta z_\xi)^2, && \text{(C.40)} \\ \overline{b}^2 &= (x_\eta z_\zeta - x_\zeta z_\eta)^2 + (x_\xi z_\zeta - x_\zeta z_\xi)^2 + (x_\xi z_\eta - x_\eta z_\xi)^2, && \text{(C.41)} \end{aligned}$$

$$\overline{d}^2 = (x_\eta y_\zeta - x_\zeta y_\xi)^2 + (x_\xi y_\zeta - x_\zeta y_\xi)^2 + (x_\xi y_\eta - x_\eta y_\xi)^2. \tag{C.42}$$

Thus a sufficient criterion for stability is

$$\Delta t \leq \frac{J}{\sqrt{r_1^2 + r_2^2 + r_3^2} + \sqrt{r_1^2 + r_2^2 + r_3^2 + c_a^2(\overline{a}^2 + \overline{b}^2 + \overline{d}^2)}}. \tag{C.43}$$

C.2.2 Stability: compressible flow

From Equation (C.29) and using the definitions for r, a, b and d, the matrix D is

$$D = \begin{pmatrix} r & \rho a & \rho b & \rho d & 0 \\ 0 & r & 0 & 0 & a/\rho \\ 0 & 0 & r & 0 & b/\rho \\ 0 & 0 & 0 & r & d/\rho \\ 0 & \gamma pa & \gamma pb & \gamma pd & r \end{pmatrix}.$$

The eigenvalues of D are

$$\frac{\Delta t}{J}\left(r,\ r,\ r,\ r + c\sqrt{a^2 + b^2 + d^2},\ r - c\sqrt{a^2 + b^2 + d^2}\right), \tag{C.44}$$

where c is the speed of sound $c = \sqrt{\gamma p/\rho}$. A conservative estimate for the maximum eigenvalue is

$$\lambda_{D_{max}} \leq \frac{\Delta t}{J}\left(\sqrt{r_1^2 + r_2^2 + r_3^2} + c\sqrt{a^2 + b^2 + d^2}\right), \tag{C.45}$$

where the value of r has been maximized by $\sqrt{r_1^2 + r_2^2 + r_3^2}$ since $s_1^2 + s_2^2 + s_3^2 = 1$. Thus a sufficient criterion for stability is

$$\Delta t \leq \frac{J}{\sqrt{r_1^2 + r_2^2 + r_3^2} + c\sqrt{a^2 + b^2 + d^2}}. \tag{C.46}$$

Appendix D

Brute force location of markers

This type of marker location in the whole domain is used for the initial location of the markers or as a last resort search when the marker has moved over a distance larger than one cell from its previous position and the usual search in the neighbouring cells has failed. First, nearly all cells are disqualified by eliminating the cells whose ranges of nodes coordinates do not encompass the marker coordinates. Each remaining cell is then divided into 6 tetrahedra $T_{1245}, T_{4568}, T_{2456}, T_{2346}, T_{3468}, T_{3678}$ (the indices refer to the node numbering defined in Appendix B). A search is performed in each tetrahedron by expressing the marker position vector $\vec{\rho}$ defined on Figure D.1 in the local coordinate system $(\vec{e}_1, \vec{e}_2, \vec{e}_3)$ as $\vec{\rho} = \alpha\vec{e}_1 + \beta\vec{e}_2 + \gamma\vec{e}_3$ where α, β, γ are found by the relations

$$\alpha = \frac{(\vec{e}_2 \times \vec{e}_3) \cdot \vec{\rho}}{(\vec{e}_1 \times \vec{e}_2) \cdot \vec{e}_3}, \quad \beta = -\frac{(\vec{e}_1 \times \vec{e}_3) \cdot \vec{\rho}}{(\vec{e}_1 \times \vec{e}_2) \cdot \vec{e}_3}, \quad \gamma = \frac{(\vec{e}_1 \times \vec{e}_2) \cdot \vec{\rho}}{(\vec{e}_1 \times \vec{e}_2) \cdot \vec{e}_3}. \tag{D.1}$$

The necessary conditions for the marker to be located in the tetrahedron are

$$\alpha \geq 0, \quad \beta \geq 0, \quad \gamma \geq 0, \quad \alpha + \beta + \gamma \leq 1. \tag{D.2}$$

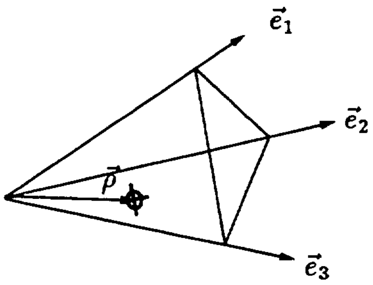


Figure D.1: Local coordinates in tetrahedron.

Appendix E

Newton-Raphson procedure

E.1 Marker location in cell

The (ξ, η, ζ) coordinates of a marker located at (x, y, z) in a given cell are found by solving the implicit system

$$\vec{r} = \sum_{k=1}^{8} N_k(\xi, \eta, \zeta)\, \vec{r}_k \tag{E.1}$$

by a few Newton iterations. Or, in shorthand notation,

$$\begin{bmatrix} g1(\xi, \eta, \zeta) \\ g2(\xi, \eta, \zeta) \\ g3(\xi, \eta, \zeta) \end{bmatrix} = \vec{g}(\vec{\alpha}) = \vec{r} - \sum_{k=1}^{8} N_k(\vec{\alpha})\, \vec{r}_k = \vec{0}. \tag{E.2}$$

Starting from an initial guess $\vec{\alpha}_0$ and using a Newton-Raphson iteration, the location of the marker in (ξ, η, ζ) is found as

$$\vec{g}(\vec{\alpha}) = \vec{g}(\vec{\alpha}_0) + \left[\frac{\partial \vec{g}}{\partial \vec{\alpha}}\right]_{\vec{\alpha}_0} \Delta\vec{\alpha}. \tag{E.3}$$

The Newton-Raphson procedure consists then in solving the following equations iteratively until convergence is obtained, that is when $\Delta\vec{\alpha} \rightarrow \vec{0}$.

$$\Delta\vec{\alpha} = -\left[\frac{\partial \vec{g}}{\partial \vec{\alpha}}\right]^{-1}_{\vec{\alpha}_0} \vec{g}(\alpha_0), \qquad \vec{\alpha}_0 = \vec{\alpha}_0 + \Delta\vec{\alpha} \tag{E.4}$$

The first guess for $\vec{\alpha}_0$ is set to $\vec{0}$ so that the search begins from the cell center. In practice only three iterations suffice to determine the location of the marker with acceptable accuracy. The Jacobian matrix is obtained from Equation (E.2) as

$$\frac{\partial \vec{g}}{\partial \vec{\alpha}} = -\sum_{k=1}^{8} \frac{\partial N_k}{\partial \vec{\alpha}} \vec{r}_k. \tag{E.5}$$

Using Equations (5.1) for the definition of the tri-linear functions N_1 to N_8 gives

$$\begin{aligned}
\frac{\partial \vec{g}}{\partial \xi} = \frac{1}{8} \; & [(1-\eta)((1-\zeta)(\vec{r}_1 - \vec{r}_2) + (1+\zeta)(\vec{r}_5 - \vec{r}_6)) \\
& +(1+\eta)((1-\zeta)(\vec{r}_4 - \vec{r}_3) + (1+\zeta)(\vec{r}_8 - \vec{r}_7))], \\
\frac{\partial \vec{g}}{\partial \eta} = \frac{1}{8} \; & [(1-\xi)((1-\zeta)(\vec{r}_1 - \vec{r}_4) + (1+\zeta)(\vec{r}_5 - \vec{r}_8)) \\
& +(1+\xi)((1-\zeta)(\vec{r}_2 - \vec{r}_3) + (1+\zeta)(\vec{r}_6 - \vec{r}_7))], \\
\frac{\partial \vec{g}}{\partial \zeta} = \frac{1}{8} \; & [(1-\xi)((1-\eta)(\vec{r}_1 - \vec{r}_5) + (1+\eta)(\vec{r}_4 - \vec{r}_8)) \\
& +(1+\xi)((1-\eta)(\vec{r}_2 - \vec{r}_6) + (1+\eta)(\vec{r}_3 - \vec{r}_7))].
\end{aligned}$$

E.2 Metrics derivatives at marker location

The terms $(\partial\vec{\alpha}/\partial\vec{r})$ required for the evaluation of the derivative of the function f at the marker location in Equation (5.8) are obtained by deriving Equation (E.2) as

$$\frac{\partial \vec{g}}{\partial \vec{\alpha}} = -\frac{\partial \vec{r}}{\partial \vec{\alpha}}. \tag{E.6}$$

Hence

$$\frac{\partial \vec{\alpha}}{\partial \vec{r}} = \begin{bmatrix} \xi_x & \xi_y & \xi_z \\ \eta_x & \eta_y & \eta_z \\ \zeta_x & \zeta_y & \zeta_z \end{bmatrix} = \left[\frac{\partial \vec{r}}{\partial \vec{\alpha}}\right]^{-1} = -\left[\frac{\partial \vec{g}}{\partial \vec{\alpha}}\right]^{-1}, \tag{E.7}$$

where the indices ${}_{\xi,\eta,\zeta}$ are shorthand for $\partial/\partial\xi, \partial/\partial\eta, \partial/\partial\zeta$. Or explicitly

$$\begin{aligned}
\xi_x &= \frac{g2_\eta g3_\zeta - g2_\zeta g3_\eta}{J}, & \xi_y &= -\frac{g1_\eta g3_\zeta - g1_\zeta g3_\eta}{J}, & \xi_z &= \frac{g1_\eta g2_\zeta - g1_\zeta g2_\eta}{J}, \\
\eta_x &= -\frac{g2_\xi g3_\zeta - g2_\zeta g3_\xi}{J}, & \eta_y &= \frac{g1_\xi g3_\zeta - g1_\zeta g3_\xi}{J}, & \eta_z &= -\frac{g1_\xi g2_\zeta - g1_\zeta g2_\xi}{J}, \\
\zeta_x &= \frac{g2_\xi g3_\eta - g2_\eta g3_\xi}{J}, & \zeta_y &= -\frac{g1_\xi g3_\eta - g1_\eta g3_\xi}{J}, & \zeta_z &= \frac{g1_\xi g2_\eta - g1_\eta g2_\xi}{J},
\end{aligned}$$

where

$$J = -g1_\xi(g2_\eta g3_\zeta - g2_\zeta g3_\eta) + g1_\eta(g2_\xi g3_\zeta - g2_\zeta g3_\xi) - g1_\zeta(g2_\xi g3_\eta - g2_\eta g3_\xi). \quad \text{(E.8)}$$

Appendix F

Change in Circulation Due to Diffusion

The loss in circulation due to diffusion can be expressed by writing the convective change in circulation around a closed curve as

$$\frac{D\Gamma}{Dt} = \frac{D}{Dt}(\oint \vec{v} \cdot \vec{dr}) = \oint \frac{D}{Dt}(\vec{v} \cdot \vec{dr}) = \underbrace{\oint \vec{v} \cdot \vec{dv}}_{\frac{1}{2}\oint \vec{v}\cdot\vec{v}=0} + \oint \frac{D\vec{v}}{dt} \cdot \vec{dr} = \oint \frac{D\vec{v}}{dt} \cdot \vec{dr}. \qquad \text{(F.1)}$$

Using the momentum equation to substitute for $\frac{D\vec{v}}{Dt}$, the convective change in circulation is

$$\frac{D\Gamma}{Dt} = \oint (-\frac{\nabla p}{\rho}) \cdot \vec{dr} + \oint (\nu \nabla^2 \vec{v}) \cdot \vec{dr}. \qquad \text{(F.2)}$$

By using Stokes theorem, the first integral is expressed as

$$\oint (-\frac{\nabla p}{\rho}) \cdot \vec{dr} = \oint (-\nabla(\frac{1}{\rho}) \times \nabla p) \cdot \vec{dS}, \qquad \text{(F.3)}$$

which gives a zero contribution for incompressible flows. Transforming the second integral of Equation (F.2) with

$$\nabla^2 \vec{v} = -\nabla \times (\nabla \times \vec{v}) + \nabla(\underbrace{\nabla \cdot \vec{v}}_{=0}) = -\nabla \times \vec{\omega}, \qquad \text{(F.4)}$$

the convective change in circulation due to viscous effects in an incompressible flow is written as

$$\frac{D\Gamma}{Dt} = -\nu \oint (\nabla \times \vec{\omega}) \cdot \vec{dr}. \qquad \text{(F.5)}$$