15BTAR_E07 COMPUTATIONAL FLUID DYNAMICS 3 0 0 3 100

OBJECTIVES:

To study the flow of dynamic fluids by computational methods

UNIT - I FUNDAMENTAL CONCEPTS

Introduction - Basic Equations of Fluid Dynamics - Incompressible In viscid Flows: Source, vortex and doublet panel, methods - lifting flows over arbitrary bodies. Mathematical properties of Fluid Dynamics Equations -_ Elliptic, Parabolic and Hyperbolic equations - Well posed problems - discretization of partial Differential Equations -Transformations and grids - Explicit finite difference methods of subsonic, supersonic and viscous flows.

UNIT - II DISCRETIZATION

Boundary layer Equations and methods of solution -Implicit time dependent methods for inviscid and viscous compressible flows - Concept of numerical dissipation --Stability properties of explicit and implicit methods - Conservative upwind discretization for Hyperbolic systems - Further advantages of upwind differencing.

UNIT - III FINITE ELEMENT TECHNIQUES

Finite Element Techniques in Computational Fluid Dynamics; introduction - Strong and Weak Formulations of a Boundary Value Problem - Strong formulation - Weighted Residual Formulation - Galerkin Formulation - Weak Formulation - Variational Formulation - Piecewise defined shape functions - Implementation of the FEM - The Solution Procedure.

UNIT - IV FINITE VOLUME TECHNIQUES

Finite Volume Techniques - Cell Centered Formulation - Lax – Vendoroff Time Stepping - Runge - Kutta Time Stepping - Multi - stage Time Stepping - Accuracy -. Cell Vertex Formulation - Multistage Time Stepping - FDM -like Finite Volume Techniques - Central and Up-wind Type Discretizations - Treatment of Derivatives.

UNIT - V FLOW FIELD ANALYSIS

Finite volume methods -Representation of the pressure gradient term and continuity equation – Staggered grid – Momentum equations – Pressure and Velocity corrections – Pressure Correction equation, SIMPLE algorithm and its variants – PISO Algorithms.

TEXTBOOKS:

S.No.	Author	Title of the Book	Publisher	Year of Publication
1.	Fletcher C.A.J	Computational Techniques for Fluid Dynamics Vol. I and II	Springer - Verlag, Berlin.	2012
2.	Charles Hirsch	Numerical Computation of Internal and External Flows, Vols. I and II	John Wiley & Sons, New York.	2007

REFERENCES BOOKS:

S.No.	Author(s)	Title of the Book	Publisher	Year of Publication
1.	John F.Wendent (Editor)	Computational Fluid Dynamics –An Introduction	Springer -Verlag, Berlin.	2008
2.	Klaus A, Hoffman and Steve T.Chiang	Computational Fluid Dynamics for Engineers Vols.I and II	Engineering Education System, New York.	2000
3.	Jiri Blazek	Computational Fluid Dynamics: Principles and Applications	Elsevier Science Philadelphia, USA	2005

WEB REFERENCES:

- www.cfd-online.com
- audilab.bmed.mcgill.ca/AudiLab/teach/fem/fem.html
- www.cmi.univ-mrs.fr/~herbin/PUBLI/bookevol.pdf
- mathworld.wolfram.com
- www.particleincell.com/2011/finite-volume

Karpagam Academy of Higher Education

(Deemed to be University Established Under Section 3 of UGC Act 1956)
Pollachi Main Road, Eachanari Post,
Coimbatore, Tamil Nadu 641021

LESSON PLAN

Subject Name : Computational Fluid Dynamics
Subject Code : 15BTAR_E07 (Credits - 3)
Name of the Faculty : B.Akilan
Designation : Assistant Professor
Year/Semester/Section : III/VI
Branch : B.Tech Aerospace Engineering

Sl. No.	No. of Periods	Topics to be Covered	Support Materials
UNIT – I : FUNDAMENTAL CONCEPTS			
1.	1	Introduction to CFD	T1
2.	1	Basics of CFD	T1
3.	1	Basic Equations of Fluid Dynamics	T1
4.	1	Incompressible In viscid flows: Source, vortex and doublet panel	T1
5.	1	Methods - lifting flows over arbitrary bodies	T1
6.	1	Mathematical properties of Fluid Dynamics Equations	R1
7.	1	Elliptic, Parabolic and Hyperbolic equations	R1
8.	1	Discretization of partial Differential Equations	R1
9.	1	Transformations and grids	T1
10.	1	Explicit finite difference methods of subsonic. Supersonic & viscous flows	T1
11.	1	**Tutorial** (Governing Equations)	T1
Total No. of Hours Planned for Unit – I			**11**

Sl. No.	No. of Periods	Topics to be Covered	Support Materials
UNIT – II : DISCRETIZATION			
12.	1	Boundary layer Equations	T1
13.	1	Methods of solution	T1
14.	1	Implicit time dependent methods for inviscid	T1
15.	1	Implicit time dependent methods for viscous compressible flows	T1
16.	1	Concept of numerical dissipation	T1
17.	1	Stability properties of explicit methods, implicit methods	T1
18.	1	Consequence of Implicit time dependent method	T1
19.	1	Conservative upwind Discretization for Hyperbolic systems	T1
20.	1	Further advantages of upwind differencing	T1

21.	1	**Tutorial** (Problems on Implicit time dependent method)	T1
Total No. of Hours Planned for Unit – II			**10**

Sl. No.	No. of Periods	Topics to be Covered	Support Materials
UNIT – III : FINITE ELEMENT TECHNIQUES			
22.	1	Finite Element Techniques in Computational Fluid Dynamics	T2
23.	1	Strong formulation	T2
24.	1	Weighted Residual Formulation	T2
25.	1	Galerkin Formulation	T2
26.	1	Weak Formulation	T2
27.	1	Variational Formulation	T2
28.	1	Piecewise defined shape functions	T2
29.	1	Implementation of the FEM	T2
30.	1	The Solution Procedure	T2
31.	1	**Tutorial** (Problems on FEM approach)	T2
Total No. of Hours Planned for Unit – III			**10**

Sl. No.	No. of Periods	Topics to be Covered	Support Materials
UNIT – IV : FINITE VOLUME TECHNIQUES			
32.	1	Finite Volume Techniques	R3
33.	1	Cell Centered Formulation	R3
34.	1	Lax – Vendor off Time Stepping	R3
35.	1	Runge – Kutta Time Steppping	R3
36.	1	Multi - stage Time Stepping	R3
37.	1	Accuracy -. Cell Vertex Formulation	R3
38.	1	Effects on Multi stage	R3
39.	1	FDM -like Finite Volume Techniques	R3
40.	1	Central –and Up wind Type Discretizations	R3
41.	1	Treatment of Derivatives	R3
42.	1	**Tutorial** (Problems on formulation)	R3
Total No. of Hours Planned for Unit – IV			**11**

Sl. No.	No. of Periods	Topics to be Covered	Support Materials
UNIT – V : FLOW FIELD ANALYSIS			
43.	1	Finite volume methods	R1
44.	1	Representation of the pressure gradient term	R1
45.	1	Staggered grid	R1
46.	1	Momentum equations	R1
47.	1	Pressure and Velocity corrections	R1
48.	1	Pressure Correction equation	R1
49.	1	SIMPLE algorithm	R1
50.	1	PISO Algorithms	R1
51.	1	**Tutorial** (Online Questions Revision)	
52.	1	**Discussion on Competitive Examination related Questions / University previous year questions**	
Total No. of Hours Planned for Unit – V			**10**

TOTAL PERIODS : 52

TEXT BOOKS

T [1] – Fletcher C.A.J "Computational Techniques for Fluid Dynamics Vol. I and II", Springer -Verlag, Berlin, 2011

T [2] – Charles Hirsch "Numerical Computation of Internal and External Flows", Vols. I and II, John Wiley & Sons, New York.2012

REFERENCES

R [1] - Klaus A, Hoffman and Steve T.Chiang, "Computational Fluid Dynamics for Engineers Vols.I and II", Engineering Education System, New York, 2010

R [2] - Jiri Blazek "Computational Fluid Dynamics: Principles and Applications", Elsevier Science Philadelphia, USA 2013

R [3] - John F.Wendent,"Computational Fluid Dynamics –An Introduction", Springer-Verlag, Berlin. 2013

WEBSITES

W [1] - www.cfd-online.com

W [2] - mathworld.wolfram.com

W [3] - www.particleincell.com/2011/finite-volume

JOURNALS

J [1] - Journal of fluids engineering

J [2] – Computational fluid dynamics journal

J [3] – Numerical methods in fluid mechanics

UNIT	Total No. of Periods Planned	Lecture Periods	Tutorial Periods
I	11	10	01
II	10	09	01
III	10	09	01
IV	11	10	01
V	10	09	01
TOTAL	52	47	05

I. CONTINUOUS INTERNAL ASSESSMENT : 40 Marks

(Internal Assessment Tests: 30, Attendance: 5, Assignment/Seminar: 5)

II. END SEMESTER EXAMINATION : 60 Marks

TOTAL : 100 Marks

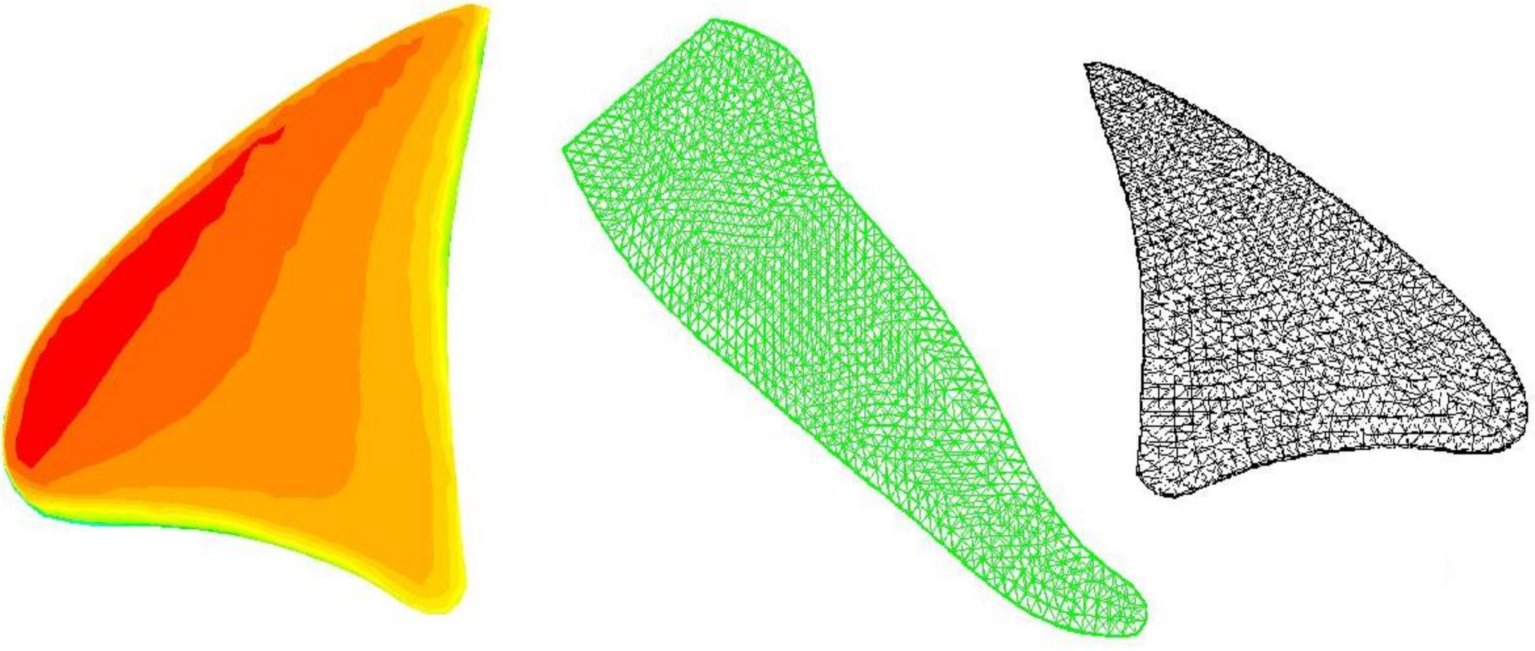
Enable | Enlighten | Enrich

KARPAGAM UNIVERSITY

(Under Section 3 of UGC Act 1956)

Computational Fluid Dynamics

Lecture Notes



Chapter Contents Page No.

UNIT I

FUNDAMENTAL CONCEPTS

- Introduction to Computational Fluid Dynamics
- Basic Equations of Fluid Dynamics
- Incompressible Inviscid Flows – Source, Vortex and Doublet
- Panel Methods – Lifting Flow over Arbitrary Bodies
- Mathematical Properties of Fluid Dynamic Equations – Elliptical, Parabolic and Hyperbolic Equations
- Well Posed Problems
- Discretization of Partial Differential Equations
- Explicit Finite Difference Method of Subsonic Flows
- Explicit Finite Difference Method of Supersonic Flows
- Explicit Finite Difference Method of Viscous Flows

1.1 Introduction to Computational Fluid Dynamics

Computational Fluid Dynamics (CFD) is the science of calculation of fluid flow and related variables using a computer. Usually the fluid body is divided into cells or elements forming a grid. Then the equations for unknown variables are solved for each grid. This requires good amount of computing resources.

"Computational Fluid Dynamics (CFD) is the technique of replacing the Partial Differential Equations (PDE'S) governing the fluid flow by a set of algebraic equations and solving them using the digital computer"

The transport equations that govern the fluid motion or flow are,

1. Continuity Equation
2. Momentum Equation
3. Energy Equation

Application of CFD

- Aerodynamics of aircraft and vehicles
- Hydrodynamics of ships
- Power plants
- Turbo machinery
- Electrical and electronics engineering
- Chemical process engineering
- External and internal environment of building
- Marine engineering
- Environmental engineering
- Hydrology and oceanography
- Meteorology
- Biomedical engineering

The availability of affordable high performance computing hardware and the introduction of user friendly interfaces have led to tremendous increase of CFD usage into the industrial community.

Approaches to study Fluid Mechanics

There are three approaches to study the fluids and their behavior they are,

1. Analytical Method

In this approach the transport equations that govern the fluid flow are solved mathematically for obtaining the specific solutions for various problems.

2. Experimental Method

This approach is based on utilizing the experimental setups to carry out the experimentation and to get the flow characteristics by flow visualization or by data acquisition to understand the phenomenon.

3. Computational Method

In this method the numerical approach is used to solve the governing equations of the fluid motion to obtain numerical solutions for the physical problems.

Major advantages of CFD over the Experimental Fluid Dynamics

1. Leas time in design and development is significantly reduced
2. CFD can simulate flow conditions which are not possible in experimental model tests
3. CFD can provide more detailed and comprehensive information
4. CFD is more cost effective that the experimental setup
5. CFD utilizes low energy

Components of a CFD code

CFD codes are the structured numerical algorithms that are used to solve the fluid flow problems. In order to provide ease access to their solving power all the commercial CFD codes include the three main elements

- **A Pre-processor**
- **A Solver**
- **A Post-processor**

A Pre-processor

The main activity in the pre-processor is to give input to the code in the form suitable for the solver by using the Graphical User Interface (GUI), the main activities in the pre-processor are,

- Definition of geometry and the region of interest
- Grid generation
- Definition of fluid properties
- Specification of boundary conditions
- Specification of solver criteria

A Solver

A solver takes the input from the preprocessor and form the governing equations in the form of PDE's which are again converted in to the algebraic form to solve the equations by the numerical approach. The iterations in the numerical approach stops when the solver criteria in the pre-processor matches.

A Post-processor

A post-processor takes the raw data from the solver and displays the results in the form the user desires. The features of the post processor includes

- Displaying geometry and grid
- Vector and Contour plots
- Particle tracking
- Pressure, Temperature and Density data

The exponential growth of the speed, memory and computing power of the digital computers in the modern era had led to the extensive development of many CFD codes which are generally used by the academia, industry and research fields. Some of the commercial CFD codes generally used are **CFX, FLUENT, COSMOFLOW, STAR-CD, FLOW-3D.**

Overview of Computational Fluid Dynamics

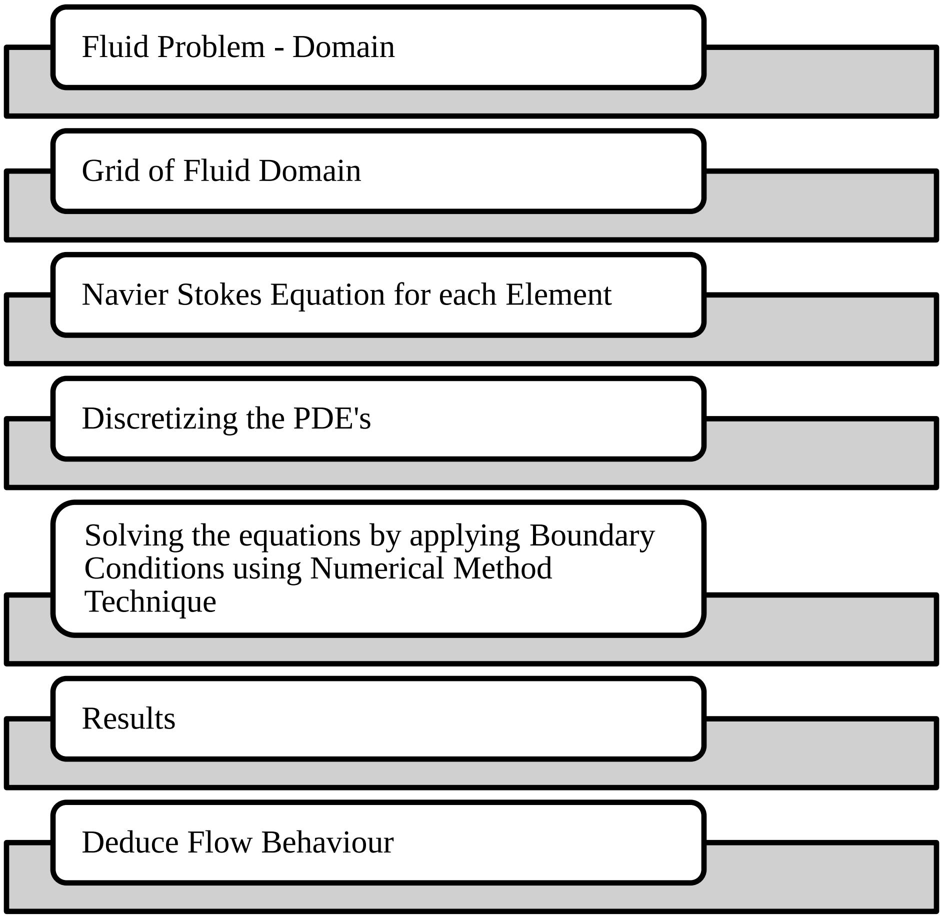

Fluid Problem - Domain
Grid of Fluid Domain
Navier Stokes Equation for each Element
Discretizing the PDE's
Solving the equations by applying Boundary Conditions using Numerical Method Technique
Results
Deduce Flow Behaviour

Figure 1.1 Overview of CFD

The complete steps followed in the CFD process are illustrated in the above figure 1.1. The steps are described below,

1. The fluid problem is taken and the physics of the problem is studied and based on it the fluid domain is created.

2. The fluid domain is divided in to finite grids

3. For each grid element the governing equations that govern the fluid flow are formed.

4. The PDE’s governing the fluid domain are discretized in to the Algebraic equations.

5. These algebraic equations are solved by applying the boundary conditions using the numerical method technique.

6. From the results of the algebraic equations the flow behavior is predicted.

1.2 Basic Equations of Fluid Dynamics

The fundamental equations of fluid motion are based on three conservation laws. For most of the engineering applications, the average measurable values of the flow properties are desired, the assumption of continuous distribution of matter is imposed. This assumption is known as **Continuum** and is valid as long as the characteristic length in a physical domain os much larger than the mean free path of molecules.

The basic equations of fluid motion are derived in either integral form or differential form are,

1. Conservation of mass – Continuity
2. Conservation of linear momentum – Newton’s second law
3. Conservation of energy – First law of thermodynamics

The conservation of linear momentum in differential form was derived by stokes and independently by Navier and therefore is known as the **Navier-Stokes** equation. It is common to refer to the entire system of equations in differential form composed of conservations of mass, momentum and energy as the **Navier-Stokes equations**. The system of equations may contain nine unknowns they are,

- ✓ Density – ρ
- ✓ Velocity Components u, v, w
- ✓ Total energy - e_t
- ✓ Pressure – p
- ✓ Temperature – T
- ✓ Dynamic viscosity - μ
- ✓ Thermal conductivity – k

There are nine unknowns ad five equations the analytical solution of such a system does not exist. Therefore, numerical techniques are employed to obtain the solution.

Integral Formulations

Integral forms of the equation are used if an average value of the fluid properties at a cross-section is desired. This approach does not provide a detailed analysis of the flow filed, however the application is simple and is used extensively. In general the integral form of the equation is derived for an extensive property and then the conservative laws are applied.

$$\frac{dN}{dt} = \frac{\partial}{\partial t}\int_{C.V.} \eta\rho \; d(\text{Vol}) + \int_{C.S.} \eta(\rho\vec{V}\cdot\vec{n})dS$$

……Equation 1.1

If N represents an extensive property, then a relation exists between the rate of change of extensive property for a system and the time rate change of the property within the control volume plus net efflux of the property across the control surfaces. Defining η as the extensive property per unit mass.

Where,

t represents time,

ρ is the density of the fluid

$\vec{V}$ is the velocity vector and

$\vec{n}$ is the unit vector

Conservation of Mass

This conservation law requires that mass is neither created nor destroyed; mathematically this is expressed as dM/dt = 0. In the above equation 1.1 the extensive property N = M and η = 1, the integral form of the conservation of mass is obtained as,

$$\frac{\partial}{\partial t}\int_{C.V.} \rho \; d(\text{Vol}) + \int_{C.S.} \rho\vec{V}\cdot\vec{n}dS = 0$$

……Equation 1.2

The physical interpretation of the above equation is that the sum of the rate of change of mass within the control volume and net efflux of mass across the control surface is zero.

Conservation of Linear Momentum

Newton’s second law applied to a non accelerating control volume which is referenced to a fixed coordinate system will result in the integral form of the momentum equation. In the case of linear momentum $\vec{G} = m\vec{V}$ is taken as extensive property and therefore $\eta = \vec{V}$. Newton’s second law in an inertial reference is expressed as $\sum\vec{F} = d\vec{G}/dt$. Thus the equation 1.1 can be represented as,

$$\Sigma\vec{F} = \frac{\partial}{\partial t}\int_{C.V.} \rho\vec{V} \; d(\text{Vol}) + \int_{C.S.} \vec{V}(\rho\vec{V}\cdot\vec{n})dS$$

……Equation 1.3

This equation states that the sum of the forces acting on a control volume is equal to the sum of the rate of change of linear momentum within the control volume and net efflux of the linear momentum across the control surfaces. The forces acting on a control volume usually represent a combination of the body forces and surface forces. For example x-component of is expressed as,

$$\Sigma F_x = \frac{\partial}{\partial t}\int_{C.V.} \rho u \; d(\text{Vol}) + \int_{C.S.} u(\rho\vec{V}\cdot\vec{n})dS$$

……Equation 1.4

Conservation of Energy

This conservation law is based on the first law of thermodynamics, which is expressed as,

$$\frac{d(\rho e_t)}{dt} = \frac{\partial Q}{\partial t} + \frac{\partial W}{\partial t}$$

……Equation 1.5

In this relation, e_t represents the total energy of the system per unit mass, while $\partial Q/\partial t$ and $\partial W/\partial t$ represent the rate of heat transfer to the system and the rate of work done on the system are defined positive. In the above equation 1.1 ρe_t is the extensive property N, and $\eta = e_t$ the total energy per unit mass. Hence,

$$\frac{\partial}{\partial t}\int_{C.V.} \rho e_t \; d(\text{Vol}) + \int_{C.S.} e_t(\rho\vec{V}\cdot\vec{n})dS = \dot{Q} + \dot{W}$$

……Equation 1.6

Differential Formulations

The differential forms of the equations of motion are utilized for situations where a detailed solution of the flow filed is required. These equations are obtained by the application of conservation laws to an infinitesimal fixed control volume. A typical differential control volume in a Cartesian coordinate system is shown in the below figure 1.2. The differential equations which are derived based on fixed coordinate system and control volume are known as Eulerian approach. If the coordinate system and control volume were allowed to move it is called Lagrangian approach.

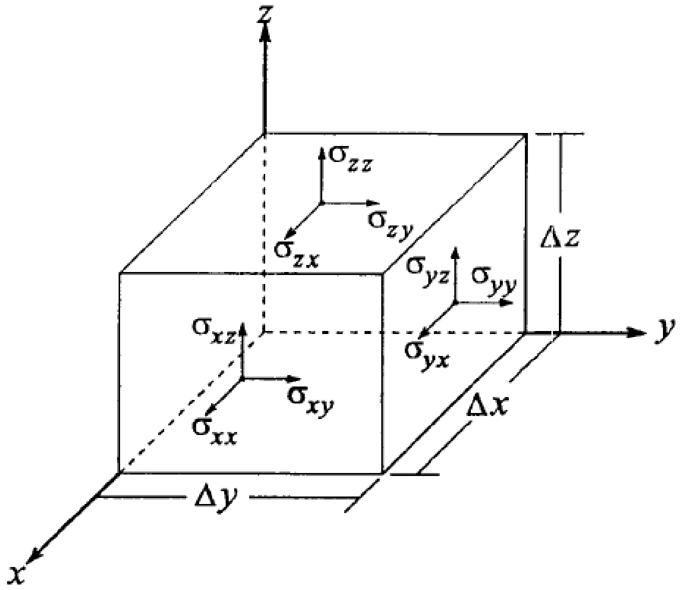


Figure 1.2 Differential element for Cartesian coordinate system

Conservation of Mass

The differential form of the conservation of the mass is known as the Continuity equation, it is written in the vector form as,

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \vec{V}) = 0$$

……Equation 1.7

This equation is written in terms of the total derivative as,

$$\frac{d\rho}{dt} + \rho \nabla \cdot \vec{V} = 0$$

……Equation 1.8

In the Cartesian coordinate system the equation can be written as,

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

……Equation 1.9

Conservation of Linear Momentum

The linear momentum equation is also known as the Navier-Stokes equation, i.e. obtained by the application of Newton’s second law to a differential element in an inertial coordinate system. If σ is used to represent stresses acting on the differential element, the components of the Navier-Stokes equations in a Cartesian coordinate system takes the following form;

$$\rho\frac{du}{dt} = \rho f_x + \frac{\partial}{\partial x}(\sigma_{xx}) + \frac{\partial}{\partial y}(\sigma_{yx}) + \frac{\partial}{\partial z}(\sigma_{zx})$$

……Equation 1.10

$$\rho\frac{dv}{dt} = \rho f_y + \frac{\partial}{\partial x}(\sigma_{xy}) + \frac{\partial}{\partial y}(\sigma_{yy}) + \frac{\partial}{\partial z}(\sigma_{zy})$$

……Equation 1.11

$$\rho\frac{dw}{dt} = \rho f_z + \frac{\partial}{\partial x}(\sigma_{xz}) + \frac{\partial}{\partial y}(\sigma_{yz}) + \frac{\partial}{\partial z}(\sigma_{zz})$$

……Equation 1.12

Shear stress σ is usually written in terms of pressure p and viscous stress τ. In tensor notation is expressed as,

$$\sigma_{ij} = -p\delta_{ij} + \tau_{ij}$$

……Equation 1.13

Where δ_{ij} is the Kronecker delta,

$$\delta_{ij} = \begin{cases} 1 & \text{for } i = j \\ 0 & \text{for } i \neq j \end{cases}$$

The components of the linear momentum equation can be written in terms of viscous stresses as,

$$\rho\frac{du}{dt} = \rho f_x - \frac{\partial p}{\partial x} + \frac{\partial \tau_{xx}}{\partial x} + \frac{\partial \tau_{yx}}{\partial y} + \frac{\partial \tau_{zx}}{\partial z}$$

……Equation 1.14

$$\rho\frac{dv}{dt} = \rho f_y - \frac{\partial p}{\partial y} + \frac{\partial \tau_{xy}}{\partial x} + \frac{\partial \tau_{yy}}{\partial y} + \frac{\partial \tau_{zy}}{\partial z}$$

……Equation 1.15

$$\rho\frac{dw}{dt} = \rho f_z - \frac{\partial p}{\partial z} + \frac{\partial \tau_{xz}}{\partial x} + \frac{\partial \tau_{yz}}{\partial y} + \frac{\partial \tau_{zz}}{\partial z}$$

……Equation 1.16

Viscous stresses are related to the rates of strain by a physical law. For most fluid this relation is linear and is known as Newtonian Fluid. For a Newtonian fluid, viscous stresses in a Cartesian coordinate system are written as,

$$\begin{aligned}
\tau_{xx} &= 2\mu\frac{\partial u}{\partial x} + \lambda\nabla\cdot\vec{V} \\
\tau_{yy} &= 2\mu\frac{\partial v}{\partial y} + \lambda\nabla\cdot\vec{V} \\
\tau_{zz} &= 2\mu\frac{\partial w}{\partial z} + \lambda\nabla\cdot\vec{V} \\
\tau_{xy} &= \tau_{yx} = \mu\left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right) \\
\tau_{xz} &= \tau_{zx} = \mu\left(\frac{\partial w}{\partial x} + \frac{\partial u}{\partial z}\right) \\
\tau_{yz} &= \tau_{zy} = \mu\left(\frac{\partial v}{\partial z} + \frac{\partial w}{\partial y}\right)
\end{aligned}$$

Where μ is known as the coefficient of viscosity or dynamic viscosity and λ is defined as the second coefficient of viscosity, the combination of μ and λ in the following form is known as the bulk viscosity k, i.e.

$$k = \lambda + \tfrac{2}{3}\mu$$

If bulk viscosity of a fluid is assumed negligible, then

$$\lambda = -\tfrac{2}{3}\mu$$

This is known as Stokes Hypothesis.

Finally the scalar components of the linear momentum equation in the conservation law form are written as,

X-component of Momentum Equation

$$\frac{\partial}{\partial t}(\rho u) + \frac{\partial}{\partial x}(\rho u^2 + p) + \frac{\partial}{\partial y}(\rho uv) + \frac{\partial}{\partial z}(\rho uw) = \frac{\partial}{\partial x}(\tau_{xx}) + \frac{\partial}{\partial y}(\tau_{xy}) + \frac{\partial}{\partial z}(\tau_{xz})$$

……Equation 1.17

Y-component of Momentum Equation

$$\frac{\partial}{\partial t}(\rho v) + \frac{\partial}{\partial x}(\rho uv) + \frac{\partial}{\partial y}(\rho v^2 + p) + \frac{\partial}{\partial z}(\rho vw) = \frac{\partial}{\partial x}(\tau_{xy}) + \frac{\partial}{\partial y}(\tau_{yy}) + \frac{\partial}{\partial z}(\tau_{yz})$$

……Equation 1.18

Y-component of Momentum Equation

$$\frac{\partial}{\partial t}(\rho w)+\frac{\partial}{\partial x}(\rho uw)+\frac{\partial}{\partial y}(\rho vw)+\frac{\partial}{\partial z}(\rho w^2+p)=\frac{\partial}{\partial x}(\tau_{xz})+\frac{\partial}{\partial y}(\tau_{yz})+\frac{\partial}{\partial z}(\tau_{zz})$$

……Equation 1.19

Energy Equation

The energy equation is derived from the first law of thermodynamics it is written as,

$$\frac{\partial}{\partial t}(\rho e_t)+\frac{\partial}{\partial x}(\rho ue_t+pu)+\frac{\partial}{\partial x}(\rho ve_t+pv)+\frac{\partial}{\partial x}(\rho we_t+pw)$$
$$=\frac{\partial}{\partial x}\left(u\tau_{xx}+v\tau_{xy}+w\tau_{xz}-q_x\right)+\frac{\partial}{\partial y}\left(u\tau_{yx}+v\tau_{yy}+w\tau_{yz}-q_y\right)$$
$$+\frac{\partial}{\partial z}\left(u\tau_{zx}+v\tau_{zy}+w\tau_{zz}-q_z\right)$$

……Equation 1.20

These equations can be written in a vector form as

$$\frac{\partial A}{\partial t}+\frac{\partial B}{\partial x}+\frac{\partial C}{\partial y}+\frac{\partial D}{\partial z}=\frac{\partial E}{\partial x}+\frac{\partial F}{\partial y}+\frac{\partial G}{\partial z}$$

……Equation 1.21

Where,

$$A=\left\{\begin{matrix}\rho\\ \rho u\\ \rho v\\ \rho w\\ \rho e_t\end{matrix}\right\}$$

$$B=\left\{\begin{matrix}\rho u\\ \rho u^2+p\\ \rho uv\\ \rho uw\\ \rho ue_t+pu\end{matrix}\right\}$$

$$C=\left\{\begin{matrix}\rho v\\ \rho vu\\ \rho v^2+p\\ \rho vw\\ \rho ve_t+pv\end{matrix}\right\}$$

$$D=\left\{\begin{matrix}\rho w\\ \rho wu\\ \rho wv\\ \rho w^2+p\\ \rho we_t+pw\end{matrix}\right\}$$

$$\boldsymbol{E} = \begin{Bmatrix} \mathbf{0} \\ \boldsymbol{\tau_{xx}} \\ \boldsymbol{\tau_{xy}} \\ \boldsymbol{\tau_{xz}} \\ \boldsymbol{u\tau_{xx} + v\tau_{xy} + w\tau_{xz} - q_x} \end{Bmatrix}$$

$$\boldsymbol{F} = \begin{Bmatrix} \mathbf{0} \\ \boldsymbol{\tau_{yx}} \\ \boldsymbol{\tau_{yy}} \\ \boldsymbol{\tau_{yz}} \\ \boldsymbol{u\tau_{yx} + v\tau_{yy} + w\tau_{yz} - q_y} \end{Bmatrix}$$

$$\boldsymbol{G} = \begin{Bmatrix} \mathbf{0} \\ \boldsymbol{\tau_{zx}} \\ \boldsymbol{\tau_{zy}} \\ \boldsymbol{\tau_{zz}} \\ \boldsymbol{u\tau_{zx} + v\tau_{zy} + w\tau_{zz} - q_z} \end{Bmatrix}$$

1.3 Incompressible Inviscid Flows – Source, Vortex and Doublet

This section discusses about the numerical analysis of incompressible Inviscid flows. The Incompressible and Inviscid flows are referred as the as the Ideal Flows, i.e. they have ρ = constant and μ = constant. They are flows where the density is constant and the viscosity effects are negligible.

The Uniform Flow

Consider a uniform flow with velocity *V*∞ moving in the *x*-direction, as sketched in Fig. 3.1. This flow is irrotational, and a solution of Laplace's equation for uniform flow yields:

$$\boxed{\phi = V_\infty x}$$

……Equation 1.22

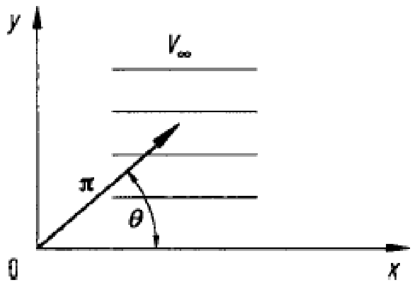


Figure 1.3 The uniform flow

In polar coordinates, (*r*, θ), the above equation can be expressed as

$$\boxed{\phi = V_\infty r \cos\theta}$$

……Equation 1.23

The Source Flow

Consider a flow with straight streamlines emanating from a point, where the velocity along each streamline varies inversely with distance from the point, as shown in Figure 1.4. Such flow is called *source flow*. This flow is also irrotational, and a solution of Laplace’s equation yields

$$\phi = \frac{\Lambda}{2\pi} \ln r$$

……Equation 1.24

where Λ is defined as the *source strength*; Λ is physically the rate of volume flow from the source, per unit depth perpendicular to the page in Figure 1.4. If Λ is negative, S*ink flow*, which is the opposite of source flow. In Figure 1.4. point 0 is the origin of the radial streamlines. We can visualize that point 0 is a *point source or sink* that *induces* the radial flow about it; in this interpretation, the point source or sink is a *singularity* in the flow field. We can also visualize that point 0 in Figure 1.4., is simply one point formed by the intersection of the plane of the paper and a *line* perpendicular to the paper. The line perpendicular to the paper is a *line source*, with strength Λ per unit length.

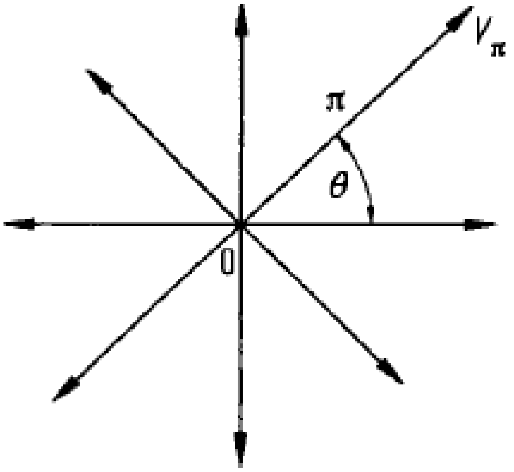


Figure 1.4 Source flow

The Vortex Flow

Consider a flow where all the streamlines are concentric circles about a given point, where the velocity along each streamline is inversely proportional to the distance from the centre, as shown in Figure 1.5. Such flow is called *vortex flow*. This flow is irrotational, and a solution of Laplace's equation yields

$$\phi = -\frac{\Gamma}{2\pi}\theta$$

……Equation 1.25

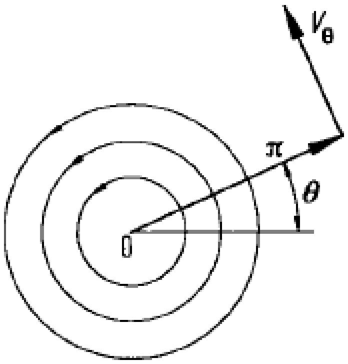


Figure 1.5 Vortex flow

Where Γ is the strength of the vortex. In Figure 1.5., point 0 can be visualized as a *point vortex* that *induces* the circular flow about it; in this interpretation, the point vortex is a *singularity* in the flow field. Visualize that point 0 in Figure 1.5. is simply one point formed by the intersection of the plane of the paper and a line perpendicular to the paper. This line is called a vortex *filament*, of strength Γ. The strength Γ is the *circulation* around the vortex filament, where circulation is defined as

$$\Gamma = -\oint \vec{V} \cdot \vec{\mathrm{d}}s$$

……Equation 1.26

In the above, the line integral of the velocity component tangent to a curve of elemental length d*s* is taken around a closed curve. This is the general definition of circulation. For a vortex filament, the above expression for Γ is defined as the vortex strength.

The Doublet

The source and sink pair leading to a singularity is called a Doublet Flow. The potential at some point P, caused by a doublet at Q, is given by

$$\phi(P) = \mu(Q)\frac{\partial}{\partial \mathbf{n}_Q}\left(\frac{1}{2\pi}\ln\left(r(P,Q)\right)\right)$$

……Equation 1.27

Here μ(Q) is the strength of the doublet and nQ is the direction of the doublet.

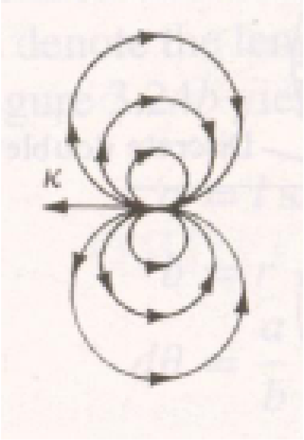


Figure 1.6 Doublet

Once more, we can put a lot of doublets in a row. We then get a doublet distribution. To find the velocity potential at P, we now have to use

$$\phi(P) = \int_S \mu(Q)\frac{\partial}{\partial \mathbf{n}_Q}\left(\frac{1}{2\pi}\ln\left(r(P,Q)\right)\right)ds.$$

……Equation 1.28

1.4 Panel Methods – Lifting Flow over Arbitrary Bodies

Panel method is a technique of approximating the flow by replacing the flow surface by a series of Line segments (2D) or Panels (3D) and placing the distribution of source or vortices or doublets on each panel. The advantages of this method include,

1. No need to define a throughout the flow field
2. Flexibility, i.e. capable of treating wide range of geometries
3. Economy – provides result with in a relative short time

Non-lifting Flows over Arbitrary Bodies – The Source Panel Method is used because source has zero circulation, therefore it is used only for non-lifting cases.

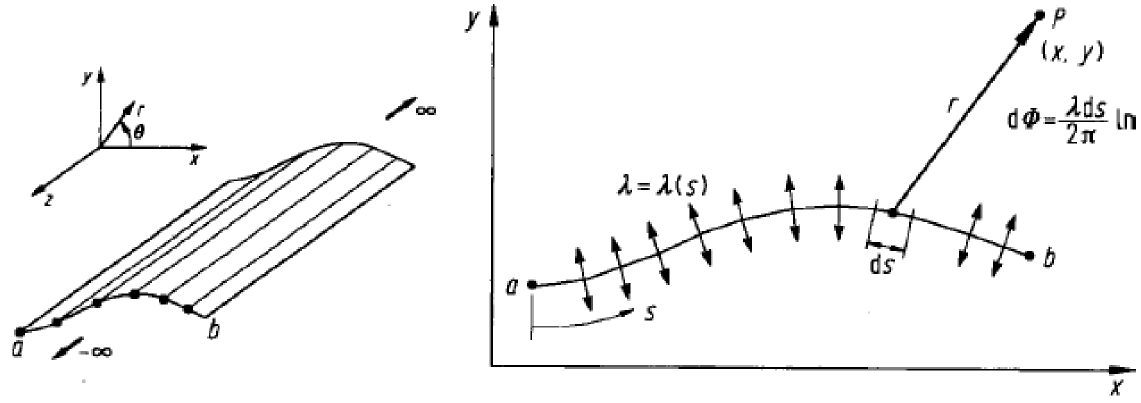


Figure 1.7 The Source Sheet

Lifting Flow over Arbitrary Bodies – The Vortex Panel Method is used because the vortices have circulation and they are used for lifting cases.

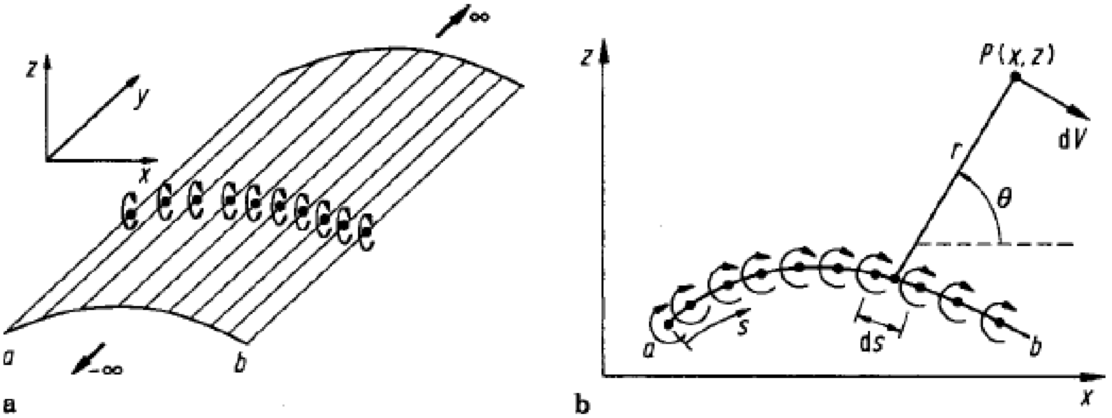


Figure 1.8 The Panel Sheet

In the present section, we introduce the analogous concept of a *vortex sheet*. Consider the straight vortex filament as shown in the above figure 1.8. Now imagine an infinite number of straight vortex filaments side by side, where the strength of each filament is infinitesimally small. These side-by-side vortex filaments form a *vortex sheet*, as shown in perspective in the figure 1.8. If we look along the series of vortex filaments the vortex sheet will appear as sketched at the lower right of Fig. 3.10. Here, we are looking at an edge view of the sheet; the vortex filaments are all perpendicular to the page. Let *s* be the distance measured along the vortex sheet in the edge view. Define $\gamma = \gamma(s)$ as the strength of the vortex sheet, per unit length along *s*. Thus, the strength of an infinitesimal portion d*s* of the sheet is γ d*s*. This small section of the vortex sheet can be treated as a distinct vortex of strength γ d*s*. Now consider point *P* in the flow, located a distance *r* from d*s*. The small section of the vortex sheet of strength γ d*s* induces a velocity potential at *P*, obtained from Equation 1.25 as

$$\mathrm{d}\Phi = -\frac{\gamma \,\mathrm{d}s}{2\pi}\theta$$

……Equation 1.29

The velocity potential at *P* due to the entire vortex sheet from *a* to *b* is

$$\Phi = -\frac{1}{2\pi}\int_a^b \theta\gamma \,\mathrm{d}s$$

……Equation 1.30

In addition, the circulation around the vortex sheet in Fig. 3.10 is the sum of the strengths of the elemental vortices, i.e.

$$\Gamma = \int_a^b \gamma \,\mathrm{d}s$$

……Equation 1.31

Another property of a vortex sheet is that the component of flow velocity tangential to the sheet experiences a discontinuous change across the sheet, given by

$$\gamma = u_1 - u_2$$

……Equation 1.32

where *u*1 and *u*2 are the tangential velocities just above and below the sheet respectively. Equation 1.32 is used to demonstrate that, for flow over an airfoil, the value of γ is zero at the trailing edge of the airfoil. This condition, namely

$$\gamma_{\mathrm{TE}} = 0$$

……Equation 1.33

is one form of the *Kutta condition* which fixes the precise value of the circulation around an airfoil with a sharp trailing edge. Finally we note that the circulation around the sheet is related to the lift force on the sheet through the Kutta–Joukowski theorem**:**

$$L = \rho_\infty V_\infty \Gamma$$

……Equation 1.34

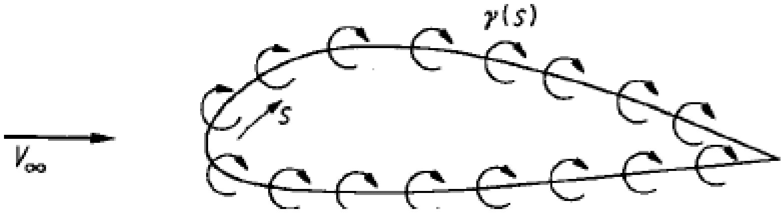


Figure 1.9 Simulation of an arbitrary airfoil by distributing a vortex sheet

Clearly, a finite value of circulation is required for the existence of lift. In the present section, we will see that the ultimate goal of the vortex panel method applied to a given body is to calculate the amount of circulation, and hence obtain the lift on the body from Equation 1.34. With the above in mind, consider an arbitrary two-dimensional body, shown in Figure 1.9. Let us wrap a vortex sheet over the complete surface of the body, as shown in Figure 1.9.We wish to find γ(*s*) such that the body surface becomes a streamline of the flow. This is the purpose of the vortex panel method.

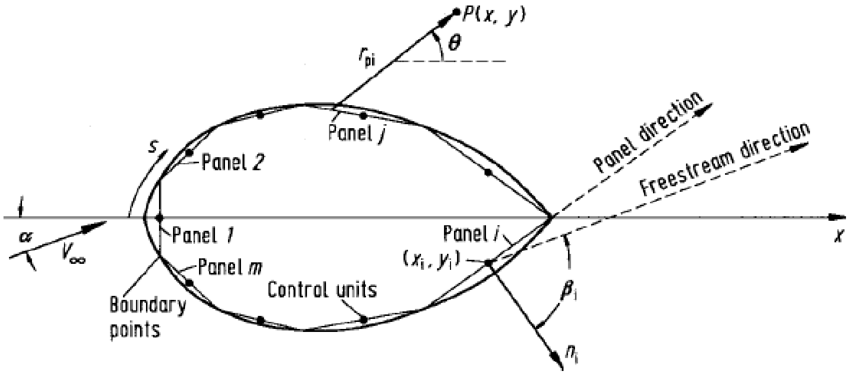


Figure 1.10 Source panel distribution over the surface of a body of arbitrary shape

Let us approximate the vortex sheet shown in Figure 1.9 by a series of straight panels. Let the vortex strength γ(*s*) per unit length be constant over a given panel, but allow it to vary from one panel to the next. That is, for the *n* panels shown in Figure 1.10, the vortex panel strengths per unit length are $\gamma_1, \gamma_2, \ldots, \gamma_j, \ldots, \gamma_n$. These panel strengths are unknowns; the main thrust of the panel technique is to solve for $\gamma_{j,\,j} = 1$ to *n*, such that the body surface becomes a streamline of the flow and such that the Kutta condition is satisfied.

Let *P* be a point located at (*x*, *y*) in the flow, and let r_{pj} be the distance from any point on the j^{th} panel to *P*, as shown in Figure 1.10. The radius r_{pj} makes the angle θ_{pj} with respect to the *x*-axis. The velocity potential induced at *P* due to the j^{th} panel, Δφj, is, from Equation 1.29,

$$\Delta\phi_j = -\frac{1}{2\pi}\int_j \theta_{pj}\gamma_j \, ds_j$$

……Equation 1.35

In Equation 1.35, γ_j is constant over the j^{th} panel, and the integral is taken over the j^{th} panel only. The angle θ_{pj} is given by

$$\theta_{pj} = \tan^{-1}\frac{y - y_j}{x - x_j}$$

……Equation 1.36

In turn, the potential at *P* due to *all* the panels is Equation 1.35 summed over all the panels:

$$\phi(P) = \sum_{j=1}^{n} \phi_j = -\sum_{j=1}^{n} \frac{\gamma_j}{2\pi}\int_j \theta_{pj} \, ds_j$$

……Equation 1.37

Since point *P* is just an arbitrary point in the flow, let us put *P* at the control point of the i^{th} panel shown in Figure 1.10. The coordinates of this control point are (x_i, y_i). Then Equation 1.36 and 1.37 become

$$\theta_{i,j} = \tan^{-1}\frac{y_i - y_j}{x_i - x_j}$$

$$\phi(x_i, y_i) = -\sum_{j=1}^{n} \frac{\gamma_j}{2\pi}\int_j \theta_{ij} \, ds_j$$

……Equation 1.38

Equation 1.38 is physically the contribution of *all* the panels to the potential at the control point of the i^{th} panel. At the control points, the normal component of the velocity is zero; this velocity is the superposition of the uniform flow velocity and the velocity induced by all the vortex panels. The component of V_∞ normal to the i^{th} panel is given

$$V_{\infty,n} = V_\infty \cos\beta_i$$

……Equation 1.39

The normal component of velocity induced at (xi, yi) by the vortex panels is

$$V_n = \frac{\partial}{\partial n_i}[\phi(x_i, y_i)]$$

……Equation 1.40

Combining Equation 1.38 and 1.40, we have

$$V_n = -\sum_{j=1}^{n} \frac{\gamma_j}{2\pi} \int_j \frac{\partial \theta_{ij}}{\partial n_i} \, \mathrm{d}s_j$$

……Equation 1.41

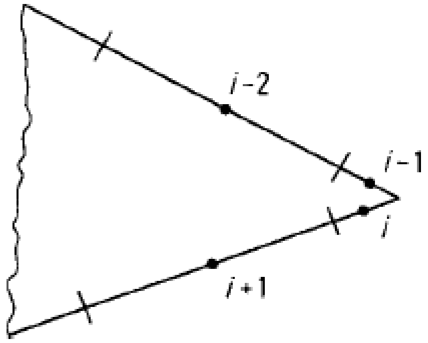


Figure 1.11 Vortex panel at the trailing edge

Equation 1.41 is a linear algebraic equation with *n* unknowns, γ_1, γ_2, . . . , γ_n. It represents the flow boundary condition evaluated at the control point of the i^{th} panel. If Equation 1.41 is applied to the control points of *all* the panels, we obtain a system of *n* linear equations with *n* unknowns. To this point, we have been deliberately paralleling the discussion of the source panel method however, the similarity stops here. For the source panel method, the *n* equations for the *n* unknown source strengths are routinely solved, giving the flow over a non-lifting body. In contrast, for the lifting case with vortex panels, in addition to the *n* equations given by Equation 1.41 applied at all the panels, we must also satisfy the Kutta condition, Equation 1.41. This can be done in several ways. For example, consider Figure 1.11, which illustrates a detail of the vortex panel distribution at the trailing edge. Note that the length of each panel can be different; their length and distribution over the body is up to your discretion. Let the two panels at the trailing edge (panels *i* and *i* − 1 in Figure 1.11) be very small. The Kutta condition is applied *precisely* at the trailing edge and is given by $\gamma_{(TE)} = 0$. To approximate this numerically, if points *i* and *i*−1 are close enough to the trailing edge, we can write

$$\gamma_i = -\gamma_{i=1}$$

……Equation 1.42

such that the strengths of the two vortex panels *i* and *i* − 1 exactly cancel at the point where they touch at the trailing edge. Thus, in order to impose the Kutta condition on the solution of the flow, Equation 1.42 must be included. Note that Equation 1.41 evaluated at all the panels and Equation 1.42 constitutes an *over-determined* system of *n* unknowns with *n* + 1 equations. Therefore, to obtain a determined system, Equation 1.41 is not evaluated at one of the control points on the body. That is, we choose to ignore one of the control points, and we evaluate Equation 1.41 at the other *n* − 1 control points. This, in combination with Equation 1.42, now gives a system of *n* linear algebraic equations with *n* unknowns, which can be solved by standard techniques.

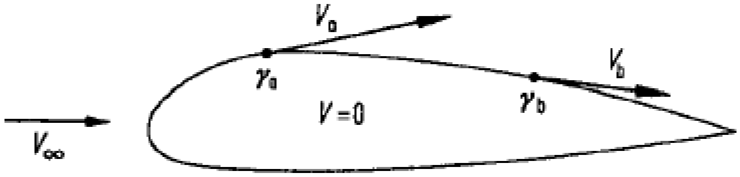


Figure 1.12 Airfoil as a solid body with zero velocity inside the profile

At this stage, we have conceptually obtained the values of $\gamma_1, \gamma_2, \ldots, \gamma_n$ which make the body surface a streamline of the flow and which also satisfy the Kutta condition. In turn, the flow velocity tangent to the surface can be obtained directly from γ. To see this more clearly, consider the airfoil shown in Figure 1.12. We are concerned only with the flow outside the airfoil and on its surface. Therefore, let the velocity be zero at every point *inside* the body, as shown in Figure 1.12. In particular, the velocity just inside the vortex sheet on the surface is zero.

$$\gamma = u_1 - u_2 = u_1 - 0 = u_1$$

……Equation 1.43

u denotes the velocity tangential to the vortex sheet. In terms of the picture shown in Figure 1.12, we obtain $V_a = \gamma_a$ at point *a*, $V_b = \gamma_b$ at point *b*, etc. Therefore, *the local velocities tangential to the airfoil surface are equal to the local values of* γ. In turn, the local pressure distribution can be obtained from Bernoulli's equation.

The total circulation and the resulting lift are obtained as follows. Let s_j be the length of the j^{th} panel. Then the circulation due to the j^{th} panel is $\gamma_j s_j$. In turn, the total circulation due to all the panels is

$$\Gamma = \sum_{i=1}^{n} \gamma_j s_j$$

……Equation 1.44

Hence, the lift per unit span is obtained from

$$L' = \rho_\infty V_\infty \sum_{n=1}^{n} \gamma_j s_j$$

……Equation 1.45

1.5 Mathematical Properties of Fluid Dynamic Equations – Elliptical, Parabolic and Hyperbolic Equations

The solution procedure of a partial differential equation (PDE) depends upon the type of equation, thus it is important to study the various classifications of PDE's. Imposition of the initial or boundary condition also depends upon the type of PDE.

Linear and Nonlinear PDE's

Linear PDE: In a Linear PDE the dependent variable and its derivative enter the equation linearly, i.e. there is no product of the dependent variable or its derivatives.

Example: One dimensional wave equation

$$\frac{\partial u}{\partial t} = -a\frac{\partial u}{\partial x}$$

……Equation 1.46

Where, a is the speed of sound which is assumed constant

Nonlinear PDE: A Nonlinear PDE contains product of the dependent variable and its derivative.

Example: Inviscid Burgers equation

$$\frac{\partial u}{\partial t} = -u\frac{\partial u}{\partial x}$$

……Equation 1.47

Second-Order PDE's

To classify the second-order PDE, Consider the following equation

$$A\frac{\partial^2\phi}{\partial x^2}+B\frac{\partial^2\phi}{\partial x\partial y}+C\frac{\partial^2\phi}{\partial y^2}+D\frac{\partial\phi}{\partial x}+E\frac{\partial\phi}{\partial y}+F\phi+G=0$$

……Equation 1.48

Where the coefficients A, B, C, D, E, F and G are functions of the independent variables x and y and of dependent variable Φ. By the definition we can express $d\Phi_x$ and $d\Phi_y$ as

$$d\phi_x=\frac{\partial\phi_x}{\partial x}dx+\frac{\partial\phi_x}{\partial y}dy=\frac{\partial^2\phi}{\partial x^2}dx+\frac{\partial^2\phi}{\partial x\partial y}dy$$

$$d\phi_y=\frac{\partial\phi_y}{\partial x}dx+\frac{\partial\phi_y}{\partial y}dy=\frac{\partial^2\phi}{\partial x\partial y}dx+\frac{\partial^2\phi}{\partial y^2}dy$$

……Equation 1.49

The equation 1.48 can be expressed as

$$A\frac{\partial^2\phi}{\partial x^2}+B\frac{\partial^2\phi}{\partial x\partial y}+C\frac{\partial^2\phi}{\partial y^2}=H$$

……Equation 1.50

Where,

$$H=-\left(D\frac{\partial\phi}{\partial x}+E\frac{\partial\phi}{\partial y}+F\phi+G\right)$$

Equations 1.49 and 1.50 are solved for Φ, using the cramers rule we get,

$$\frac{\partial^2\phi}{\partial x\partial y}=\frac{\begin{vmatrix} A & H & C \\ dx & d\phi_x & 0 \\ 0 & d\phi_y & dy \end{vmatrix}}{\begin{vmatrix} A & B & C \\ dx & dy & 0 \\ 0 & dx & dy \end{vmatrix}}$$

Since it is possible to have the discontinuous in the second order derivatives of the dependent variable across the characteristics, these derivatives are indeterminate. Thus setting the denominator equal to zero.

$$\begin{vmatrix} A & B & C \\ dx & dy & 0 \\ 0 & dx & dy \end{vmatrix}=0$$

Yields the equation

$$A\left(\frac{dy}{dx}\right)^2 - B\left(\frac{dy}{dx}\right) + C = 0$$

……Equation 1.51

Solving this quadratic equation yields the equation of the characteristic in physical space.

$$\left(\frac{dy}{dx}\right)_{\alpha,\beta} = \frac{B \pm \sqrt{B^2 - 4AC}}{2A}$$

……Equation 1.52

Depending on the value of B^2-4AC the characteristic curves are real or imaginary. They are classified as,

(a) elliptic if $B^2 - 4AC < 0$

(b) parabolic if $B^2 - 4AC = 0$

(c) hyperbolic if $B^2 - 4AC > 0$

Elliptical Equations

A partial differential equation is elliptical in a region if B^2-4AC is less than zero at all points in the region. An elliptic PDE has no real characteristic curves. A disturbance is propagated instantly in all directions within the region. The domain of solution of a elliptical equation is a closed region.

Example:

Laplace equation

$$\frac{\partial^2\phi}{\partial x^2} + \frac{\partial^2\phi}{\partial y^2} = 0$$

……Equation 1.53

Poisson's equation

$$\frac{\partial^2\phi}{\partial x^2} + \frac{\partial^2\phi}{\partial y^2} = f(x,y)$$

……Equation 1.54

Parabolic Equations

A partial differential equation is parabolic in a region if B^2-4AC is equal to zero at all points in the region. A parabolic PDE has solution domain as open region. An parabolic PDE has one real characteristic curve.

Example:

Unsteady heat conduction in one dimension

$$\frac{\partial T}{\partial t} = \alpha \frac{\partial^2 T}{\partial x^2}$$

……Equation 1.55

Diffusion of viscosity equation

$$\frac{\partial u}{\partial t} = \nu \frac{\partial^2 u}{\partial y^2}$$

……Equation 1.56

Hyperbolic Equations

A partial differential equation is called hyperbolic if B^2-4AC is greater than zero at all points in the region. A hyperbolic PDE has two real characteristic curves.

Example:

Second-order wave equation

$$\frac{\partial^2 \phi}{\partial t^2} = a^2 \frac{\partial^2 \phi}{\partial x^2}$$

……Equation 1.57

1.6 Well Posed Problems

The governing equation and auxiliary conditions (initial/boundary) are well posed mathematically if the following three conditions are met;

1. The solution exists
2. The solution is unique
3. The solution depends continuously on the auxiliary data

There are some flows for which multiple solutions may be expected on the physical grounds. These problems fail the above criteria of mathematical well posedness. This situation often arises for flows undergoing transition from laminar to turbulent motion.

1.7 Discretization of Partial Differential Equations

In order to solve the governing equations of the fluid motion, first their numerical analogue must be generated. This is done by a process referred to as discretization. In the discretization process, each term within the partial differential equation describing the flow is written in such a manner that the computer can be programmed to calculate. There are various techniques for numerical discretization. Here we will introduce three of the most commonly used techniques, namely:

(1) The Finite Difference Method,

(2) The Finite Element Method and

(3) The Finite Volume Method.

Spectral methods are also used in CFD, which will be briefly discussed.

The Finite Difference Method

Finite difference method utilizes the Taylor series expansion to write the derivatives of a variable as the differences between values of the variable at various points in space or time. Utilization of the Taylor series to discretize the derivative of dependent variable, e.g., velocity *u*, with respect to the independent variable, e.g., special coordinated *x*, is shown in Figure 1.13. Consider the curve in Figure 1.13 which represent the variation of *u* with *x*, *i.e., u(x)*. After discretization, the curve *u(x)* can be represented by a set of discrete points, u_i's. These discrete points can be related to each other using a Taylor series expansion. Consider two points, (*i+1*) and (*i-1*), a small distance Δx from the central point, (*i*). Thus velocity u_i can be expressed in terms of Taylor series expansion about point (*i*) as:

$$u_{i+1} = u_i + \left(\frac{\partial u}{\partial x}\right)\Delta x + \left(\frac{\partial^2 u}{\partial x^2}\right)\frac{(\Delta x)^2}{2} + \left(\frac{\partial^3 u}{\partial x^3}\right)_i \frac{(\Delta x^3)}{6} + \ldots$$

……Equation 1.58

$$u_{i-1} = u_i - \left(\frac{\partial u}{\partial x}\right)\Delta x + \left(\frac{\partial^2 u}{\partial x^2}\right)\frac{(\Delta x)^2}{2} - \left(\frac{\partial^3 u}{\partial x^3}\right)_i \frac{(\Delta x^3)}{6} + \ldots$$

……Equation 1.59

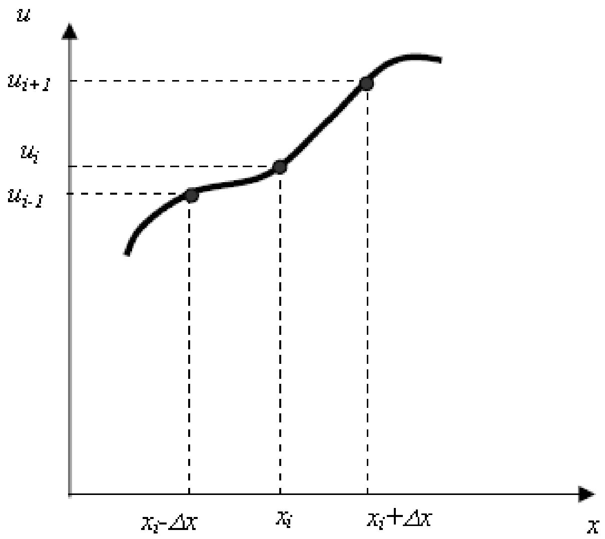


Figure 1.13 Location of points for Taylor series

These equations are mathematically exact if numbers of terms are infinite and Δ*x* is small .Note that ignoring these terms leads to a source of error in the numerical calculations as the equation for the derivatives is truncated. This error is referred to as the truncation error. For the second order accurate expression, the truncation error is:

$$\sum_{n=3}^{\infty}\left(\frac{\partial^n u}{\partial x^n}\right)_i \frac{(\Delta x)^{n-1}}{n!}$$

……Equation 1.60

By subtracting or adding these two equations, new equations can be found for the first and second derivatives at the central position *i*. These derivatives are

$$\left(\frac{\partial u}{\partial x}\right)_i = \frac{u_{i+1}-u_{i-1}}{2\Delta x} - \left(\frac{\partial^3 u}{\partial x^3}\right)_i \frac{(\Delta x)^2}{6}$$

……Equation 1.61

$$\left(\frac{\partial^2 u}{\partial x^2}\right)_i = \frac{u_{i-1}-2u_i+u_{i+1}}{(\Delta x)^2} + O(\Delta x)^2$$

……Equation 1.62

Equation 1.61 and 1.62 are referred as the Central Difference Equations for first and second order respectively. The first-order derivative can be formed as

$$\left(\frac{\partial u}{\partial x}\right)_i = \frac{u_{i+1} - u_i}{\Delta x} - \left(\frac{\partial^2 u}{\partial x^2}\right)_i \frac{(\Delta x)}{\mathbf{2}}$$

……Equation 1.63

This is referred to as the Forward difference. Similarly, another first-order derivative can be formed as

$$\left(\frac{\partial u}{\partial x}\right)_i = \frac{u_i - u_{i-1}}{\Delta x} - \left(\frac{\partial^2 u}{\partial x^2}\right)_i \frac{(\Delta x)}{\mathbf{2}}$$

……Equation 1.64

This is referred to as the Backward difference. As noted by the expressions, difference formulae are classified in two ways:

(1) By the geometrical relationship of the points, namely, central, forward, and backward differencing

(2) By the accuracy of the expressions, for instance, central difference is second-order accurate, whereas, both forward and backward differences are first-order accurate, as the higher order terms are neglected.

The Finite Element Method

In the finite element method, the fluid domain under consideration is divided into finite number of sub-domains, known as elements. A simple function is assumed for the variation of each variable inside each element. The summation of variation of the variable in each element is used to describe the whole flow field. Consider the two nodded element shown in Figure 1.14, in which variable *u* varies linearly inside the element. The end points of the element are called the nodes of the element. For a linear variation of *u*, the first derivative of *u* with respect to *x* is imply a constant. If *u* is assumed to vary linearly inside an element, we cannot define a second derivative for it. Since most fluid problems include second derivative, the following technique is designed to overcome this problem. First, the partial differential equation is multiplied by an unknown function, and then the whole equation can be integrated over the domain in which it applies. Finally the terms that need to have the order of their derivatives reduced are integrated by parts. This is known as producing a variational formulation.

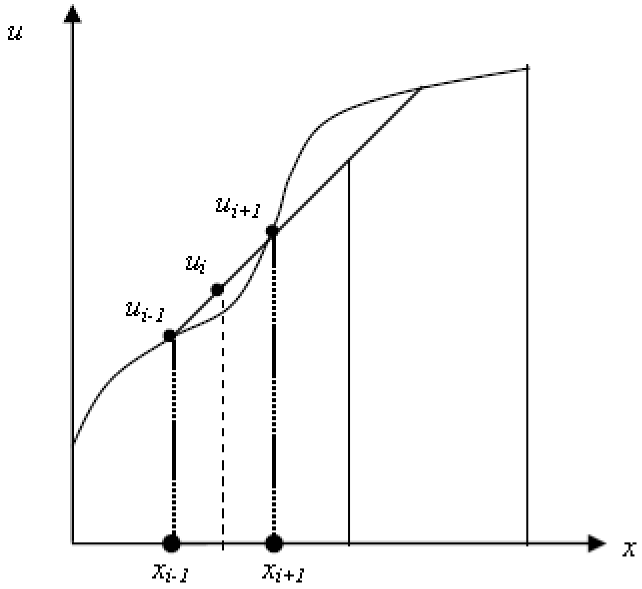


Figure 1.14 A two noded linear element

As an example, we will develop the finite element formulation of the Laplace's Equation in one dimensions:

$$\frac{d^2u}{dx^2} = 0$$

……Equation 1.65

where velocity *u* is a function of the spatial coordinates *x*. We multiply equation 1.65 by some function *W* and integrate it over the domain of interest denoted by Ω:

$$\int W\left[\frac{d^2u}{dx^2}\right]d\Omega = 0$$

……Equation 1.66

The above equation can be integrated by parts as,

$$\int\left[-\frac{dW}{dx}\frac{du}{dx}\right]d\Omega + \int\left[W\frac{du}{dx}n_x\right]d\Gamma = 0$$

……Equation 1.67

where Γ denotes the boundary of the domain Ω and *nx* is the unit outward normal vector to the boundary Γ. We will now divide the domain into several elements and assume a function for the variation of the variable *u* in each element. If a two-noded linear element is assumes, the variation of *u* in each element can be represented by

$$u_i = u_{i-1} + (u_{i+1} - u_{i-1})\left[\frac{x_i - x_{i-1}}{x_{i+1} - x_{i-1}}\right]$$

$$u_i = u_{i-1}\left[\frac{x_{i+1} - x_i}{x_{i+1} - x_{i-1}}\right] + u_{i+1}\left[\frac{x_i - x_{i-1}}{x_{i+1} - x_{i-1}}\right]$$

……Equation 1.68

The terms in the brackets are called the shape functions and are denoted as *Ni*'s. *ui-1* and *ui+1* are the nodal values of the variable *u* and are denoted as *u*i's. Therefore, the variable *u* can be written in the following form

$$u_i = N_{i-1}u_{i-1} + N_{i+1}u_{i+1}$$

……Equation 1.69

Thus, the shape functions corresponding to the two-nodal linear element, represented by

$$N_{i-1} = \frac{x_{i+1} - x_i}{x_{i+1} - x_{i-1}}$$

$$N_{i+1} = \frac{x_i - x_{i-1}}{x_{i+1} - x_{i-1}}$$

……Equation 1.70

We can now determine the derivatives of the variable *u*, using equation

$$\frac{du}{dx} = \sum_{i=1}^{m} \frac{dN_i}{dx} u_i$$

……Equation 1.71

Where *m* is the number of nodes on the element. Note that u_i's are nodal values of *u* and they are not variables, therefore, they are not differentiated. In order to solve equation we still need to describe the function *W*. There are several methods, which are used for the specification of the variable *W*. However, the most common method is the Galerkin method in which *W* is assumed to be the same as the shape function for each element.

The Finite Volume Method

The finite volume method is currently the most popular method in CFD. The main reason is that it can resolve some of the difficulties that the other two methods have. Generally, the finite volume method is a special case of finite element, when the function *W* is equal to 1 everywhere in the domain. A typical finite volume, or cell, is shown in Figure 1.15. In this figure the centroid of the volume, point P, is the reference point at which we want to discretize the partial differential equation.

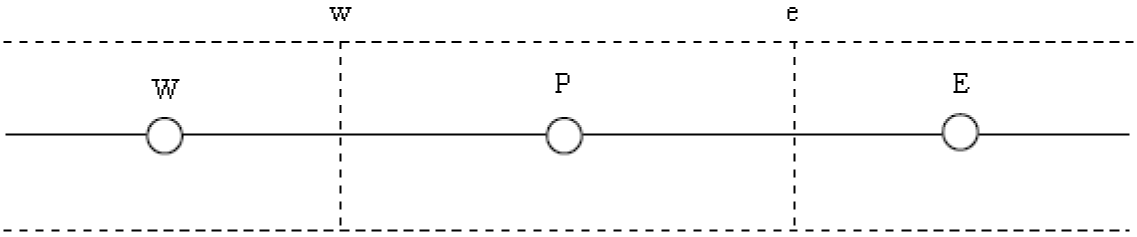


Figure 1.15 A finite volume in one dimension

The neighboring volumes are denoted as, W, volume to the west side, and E, the volume to the east side of the volume P. For the one-dimensional finite volume shown in Figure 1., the volume with centroid P, has two boundary faces at w and e. The second derivative of a variable at P can be written as the difference between the 1st derivatives of the variable evaluated at the volume faces:

$$\left[\frac{\partial^2 u}{\partial x^2}\right]_p = \frac{\left[\left(\frac{\partial u}{\partial x}\right)_e - \left(\frac{\partial u}{\partial x}\right)_w\right]}{x_e - x_w}$$

……Equation 1.72

The first derivatives at the volume faces can be written as to be the differences in the values of the variable at the neighboring volume centroids:

$$\left[\frac{\partial u}{\partial x}\right]_e = \frac{u_E - u_P}{x_E - x_P} \quad \left[\frac{\partial u}{\partial x}\right]_w = \frac{u_P - u_W}{x_P - x_W}$$

……Equation 1.73

We can apply this technique to equation 1.73 to obtain its finite volume formulation. The above method is also referred to as the Cell Centered (CC) Method, where the flow variables are allocated at the center of the computational cell. The CC variable arrangement is the most popular, since it leads to considerably simpler implementations than other arrangements. On the other hand, the CC arrangement is more susceptible to truncation errors, when the mesh departs from uniform rectangles. Traditionally the finite volume methods have used regular grids for the efficiency of the computations. However, recently, irregular grids have become more popular for simulating flows in complex geometries. Obviously, the computational effort is more when irregular grids are used, since the algorithm should use a table to lookup the geometrical relationships between the volumes or element faces. This involves finding data from a disk store of the computer, which increases the computational time.

Spectral Methods

Another method of generating a numerical analog of a differential equation is by using Fourier series or series of Chebyshev polynomials to approximate the unknown functions. Such methods are called the Spectral method. Fourier series or series of Chebyshev polynomials are valid throughout the entire computational domain. This is the main difference between the spectral method and the FDM and FEM, in which the approximations are local. Once the unknowns are replaced with the truncated series, certain constraints are used to generate algebraic equations for the coefficients of the Fourier or Chebyshev series. Either weighted residual technique or a technique based on forcing the approximate function to coincide with the exact solution at several grid points is used as the constraint.

Comparison between Discretization Methods

The main differences between the above three techniques include the followings. The finite difference method and the finite volume method both produce the numerical equations at a given point based on the values at neighboring points, whereas the finite element method produces equations for each element independently of all the other elements. It is only when the finite element equations are collected together and assembled into the global matrices that the interaction between elements is taken into account.

Both FDM and FVM can apply the fixed-value boundary conditions by inserting the values into the solution, but must modify the equations to take account of any derivative boundary conditions. However, the finite element method takes care of derivative boundary conditions when the element equations are formed and then the fixed values of variables must be applied to the global matrices. One advantage that the finite element method has is that the programs are written to create matrices for each element, which are then assembled to form the global equations before the whole problem is solved. Finite volume and finite difference programs, on the other hand, are written to combine the setting up of the equations and their solution. The decoupling of these two phases, in finite element programs, allows the programmer to keep the organization of the program very clear and the addition of new element types is not a major problem. Adding new cell types to a finite volume program can, however, be a major task involving a rewrite of the program and so some finite volume programs can exhibit problems if they have multiple cell types.

1.8 Explicit Finite Difference Method of Subsonic Flows – Elliptical Equations

The governing equation of subsonic fluid flows and heat transfer problems can be reduced to elliptic form for particular applications. Some of the examples are steady state heat conduction equation, velocity potential equation for incompressible, inviscid flow and stream function equation. Now consider Laplace equation,

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0$$

……Equation 1.74

The finite difference formulation of the above equation can be written by using five point formula as,

$$\frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{(\Delta x)^2} + \frac{u_{i,j+1} - 2u_{i,j} + u_{i,j-1}}{(\Delta y)^2} = 0$$

……Equation 1.75

The corresponding grid points are shown in the below figure 1.16.

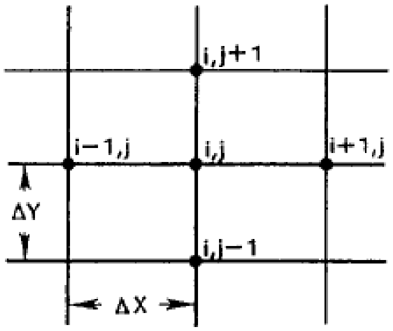


Figure 1.16 Grid for five point formula

The above equation 1.75 can be rewritten as,

$$u_{i+1,j} - 2u_{i,j} + u_{i-1,j} + \left(\frac{\Delta x}{\Delta y}\right)^2 (u_{i,j+1} - 2u_{i,j} + u_{i,j-1}) = 0$$

……Equation 1.76

Define the ratio of step sizes as β = Δx/Δy and by rearranging the above equation we get,

$$u_{i+1,j} + u_{i-1,j} + \beta^2 u_{i,j+1} + \beta^2 u_{i,j-1} - 2(1+\beta^2)u_{i,j} = 0$$

……Equation 1.77

In order to explore various solution procedures, first consider a square domain with Dirichlet boundary conditions. For example let us simple example of 6 x 6 grid system subjected to the following boundary conditions.

$$x = 0 \qquad u = u_2 \quad , \quad y = 0 \qquad u = u_1$$

$$x = L \qquad u = u_4 \quad , \quad y = H \qquad u = u_3$$

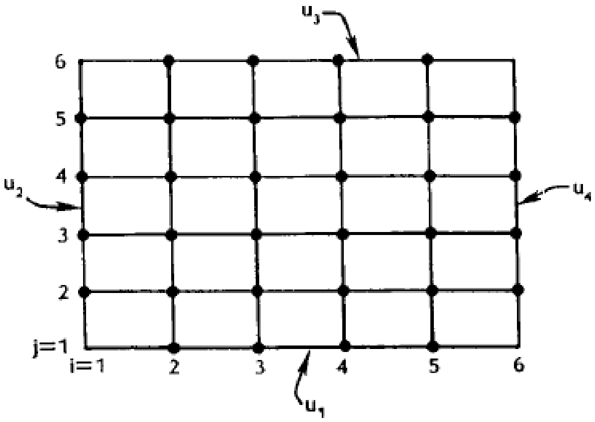


Figure 1.17 Grid system used for solution

By applying the above equation 1.77 to the interior grid points produce sixteen equations with sixteen unknowns.

$$u_{3,2} + u_{1,2} + \beta^2 u_{2,3} + \beta^2 u_{2,1} - 2(1+\beta^2)u_{2,2} = 0$$

$$u_{4,2} + u_{2,2} + \beta^2 u_{3,3} + \beta^2 u_{3,1} - 2(1+\beta^2)u_{3,2} = 0$$

$$u_{5,2} + u_{3,2} + \beta^2 u_{4,3} + \beta^2 u_{4,1} - 2(1+\beta^2)u_{4,2} = 0$$

$$u_{6,2} + u_{4,2} + \beta^2 u_{5,3} + \beta^2 u_{5,1} - 2(1+\beta^2)u_{5,2} = 0$$

$$u_{3,3} + u_{1,3} + \beta^2 u_{2,4} + \beta^2 u_{2,2} - 2(1+\beta^2)u_{2,3} = 0$$

$$u_{4,3} + u_{2,3} + \beta^2 u_{3,4} + \beta^2 u_{3,2} - 2(1+\beta^2)u_{3,3} = 0$$

$$u_{5,3} + u_{3,3} + \beta^2 u_{4,4} + \beta^2 u_{4,2} - 2(1+\beta^2)u_{4,3} = 0$$

$$u_{6,3} + u_{4,3} + \beta^2 u_{5,4} + \beta^2 u_{5,2} - 2(1+\beta^2)u_{5,3} = 0$$

$$u_{3,4} + u_{1,4} + \beta^2 u_{2,5} + \beta^2 u_{2,3} - 2(1+\beta^2)u_{2,4} = 0$$

$$u_{4,4} + u_{2,4} + \beta^2 u_{3,5} + \beta^2 u_{3,3} - 2(1+\beta^2)u_{3,4} = 0$$

$$u_{5,4} + u_{3,4} + \beta^2 u_{4,5} + \beta^2 u_{4,3} - 2(1+\beta^2)u_{4,4} = 0$$

$$u_{6,4} + u_{4,4} + \beta^2 u_{5,5} + \beta^2 u_{5,3} - 2(1+\beta^2)u_{5,4} = 0$$

$$u_{3,5} + u_{1,5} + \beta^2 u_{2,6} + \beta^2 u_{2,4} - 2(1+\beta^2)u_{2,5} = 0$$

$$u_{4,5} + u_{2,5} + \beta^2 u_{3,6} + \beta^2 u_{3,4} - 2(1+\beta^2)u_{3,5} = 0$$

$$u_{5,5} + u_{3,5} + \beta^2 u_{4,6} + \beta^2 u_{4,4} - 2(1+\beta^2)u_{4,5} = 0$$

$$u_{6,5} + u_{4,5} + \beta^2 u_{5,6} + \beta^2 u_{5,4} - 2(1+\beta^2)u_{5,5} = 0$$

These equations are expressed in the matrix form as,

$$\begin{bmatrix}
\alpha & 1 & 0 & 0 & \beta^2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & \alpha & 1 & 0 & 0 & \beta^2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & \alpha & 1 & 0 & 0 & \beta^2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & \alpha & 0 & 0 & 0 & \beta^2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\beta^2 & 0 & 0 & 0 & \alpha & 1 & 0 & 0 & \beta^2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & \beta^2 & 0 & 0 & 1 & \alpha & 1 & 0 & 0 & \beta^2 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & \beta^2 & 0 & 0 & 1 & \alpha & 1 & 0 & 0 & \beta^2 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & \beta^2 & 0 & 0 & 1 & \alpha & 0 & 0 & 0 & \beta^2 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & \beta^2 & 0 & 0 & 0 & \alpha & 1 & 0 & 0 & \beta^2 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & \beta^2 & 0 & 0 & 1 & \alpha & 1 & 0 & 0 & \beta^2 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & \beta^2 & 0 & 0 & 1 & \alpha & 1 & 0 & 0 & \beta^2 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & \beta^2 & 0 & 0 & 1 & \alpha & 0 & 0 & 0 & \beta^2 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \beta^2 & 0 & 0 & 0 & \alpha & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \beta^2 & 0 & 0 & 1 & \alpha & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \beta^2 & 0 & 0 & 1 & \alpha & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \beta^2 & 0 & 0 & 1 & \alpha
\end{bmatrix}
\begin{bmatrix}
u_{2,2} \\ u_{3,2} \\ u_{4,2} \\ u_{5,2} \\ u_{2,3} \\ u_{3,3} \\ u_{4,3} \\ u_{5,3} \\ u_{2,4} \\ u_{3,4} \\ u_{4,4} \\ u_{5,4} \\ u_{2,5} \\ u_{3,5} \\ u_{4,5} \\ u_{5,5}
\end{bmatrix}
=
\begin{bmatrix}
-u_{1,2} - \beta^2 u_{2,1} \\ -\beta^2 u_{3,1} \\ -\beta^2 u_{4,1} \\ -u_{6,2} - \beta^2 u_{5,1} \\ -u_{1,3} \\ 0 \\ 0 \\ -u_{6,3} \\ -u_{1,4} \\ 0 \\ 0 \\ -u_{6,4} \\ -u_{1,5} - \beta^2 u_{3,6} \\ -\beta^2 u_{3,6} \\ -\beta^2 u_{4,6} \\ -u_{6,5} - \beta^2 u_{5,6}
\end{bmatrix}$$

where $\alpha = -2(1+\beta^2)$.

Solution Algorithms

In general there are two methods of solution for the system of simultaneous linear algebraic equations they are Direct and Iterative Methods. Some of the familiar direct methods are, Cramer's Rule and Gaussian Elimination Method. The major disadvantage of this method is it has enormous amount of arithmetic operations to produce a solution. So in this chapter discusses only in the iterative method. Iterative procedures for solving a system of linear algebraic equations are simple and easy to program. The idea behind this method is to obtain the solution by iteration. The various formulations of the iterative method can be divided into two categories. If the formulations results only in one unknown this is called as Explicit/Point iterative method. If the formulation involves more than one unknown it is called as Implicit/Line iterative method.

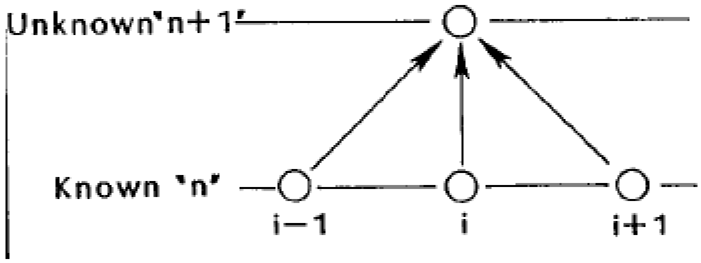


Figure 1.18 Explicit Formulation

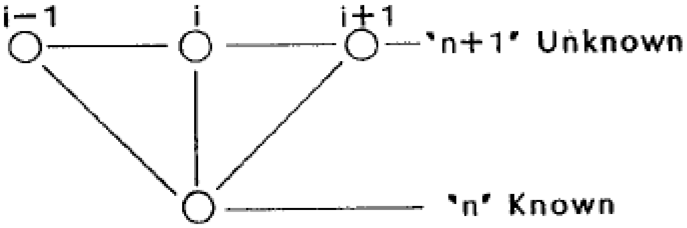


Figure 1.19 Implicit Formulation

The Jacobi Iteration Method

In this method the dependent variable at each grid point is solved using initial guessed values of the neighboring points or previously computed values. Therefore the equation is given by,

$$u_{i,j}^{k+1} = \frac{1}{2(1+\beta^2)}\left[u_{i+1,j}^{k} + u_{i-1,j}^{k} + \beta^2(u_{i,j+1}^{k} + u_{i,j-1}^{k})\right]$$

……Equation 1.78

Which is used to compute u_{ij} at the new iteration level of k+1 where k corresponds to the previously computed values. The computation is carried out until a specified convergence criteria is met. The results from the convergence can be called as ***Converged Solution*** if it has met the convergence criteria and as ***Steady-State Solution*** if the results does not vary with time.

The Point Gauss-Seidel Iteration Method

In this method the current values of the dependent variable is are used to compute the neighboring points as soon as they are available. This will certainly increase the convergence rate dramatically over the Jacobi method. The method is convergent if the largest elements are located in the main diagonal of the coefficient matrix.

The formal requirement for the convergence of the method is

$$|a_{ii}| \geq \sum_{\substack{j=1 \\ j \neq i}}^{n} |a_{ij}|$$

And at least for one row,

$$|a_{ii}| > \sum_{\substack{j=1 \\ j \neq i}}^{n} |a_{ij}|$$

……Equation 1.79

Since this is a sufficient condition, the method may converge even though the condition is met for all rows. The finite difference equation 1.79 can be written as

$$u_{i,j} = \frac{1}{2(1+\beta^2)}\left[u_{i+1,j} + u_{i-1,j} + \beta^2(u_{i,j+1} + u_{i,j-1})\right]$$

……Equation 1.80

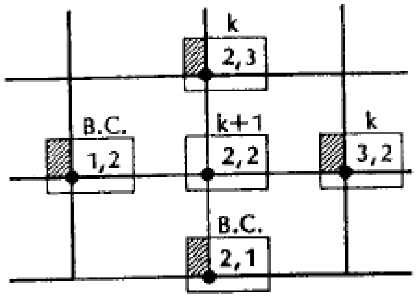


Figure 1.20 Grid Points for the equation 1.30

For the computation of the value at the point (2,2) the equation can be written as,

$$u_{2,2}^{k+1} = \frac{1}{2(1+\beta^2)}\left[u_{3,2} + \underline{u_{1,2}} + \beta^2(u_{2,3} + \underline{u_{2,1}})\right]$$

……Equation 1.81

In the above equation u_{12} and u_{21} are provided by the boundary conditions and values u_{23} and u_{32} are the values from the previous iteration. Thus in terms of the iteration level the equation can be written as,

$$u_{2,2}^{k+1} = \frac{1}{2(1+\beta^2)}\left[u_{3,2}^{k} + \underline{u_{1,2}} + \beta^2(u_{2,3}^{k} + \underline{u_{2,1}})\right]$$

……Equation 1.82

The general formulation is provided by the equation

$$u_{i,j}^{k+1} = \frac{1}{2(1+\beta^2)}\left[u_{i+1,j}^{k} + u_{i-1,j}^{k+1} + \beta^2(u_{i,j+1}^{k} + u_{i,j-1}^{k+1})\right]$$

……Equation 1.83

This is a point iteration method since only one unknown is sought. The grid points are shown in the below figure.

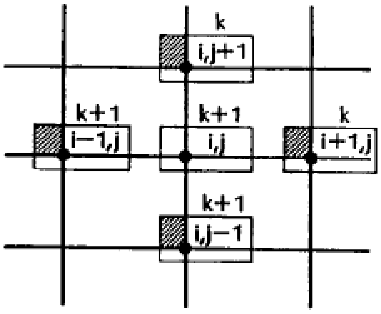


Figure 1.21 Grid points for the equation 1.83

Point Successive Over-Relaxation Method (PSOR)

In this solution process a trend in the computed values of the dependent variable is noticed, then the direction of change can be used to extrapolate for the next iteration and thereby accelerating the solution procedure. This procedure is known as successive over-relaxation(SOR).

Consider the point Gauss Seidel iteration method, which is given by

$$u_{i,j}^{k+1} = \frac{1}{2(1+\beta^2)}\left[u_{i+1,j}^{k} + u_{i-1,j}^{k+1} + \beta^2(u_{i,j+1}^{k} + u_{i,j-1}^{k+1})\right]$$

……Equation 1.84

Adding $u_{i,j}^{k} - u_{i,j}^{k}$ to the right hand side and collecting the terms we obtain

$$u_{i,j}^{k+1} = u_{i,j}^{k} + \frac{1}{2(1+\beta^2)}\left[u_{i+1,j}^{k} + u_{i-1,j}^{k+1} + \beta^2(u_{i,j+1}^{k} + u_{i,j-1}^{k+1}) - 2(1+\beta^2)u_{i,j}^{k}\right]$$

……Equation 1.85

As the solution proceeds u_{ij}^{k} must approach u_{ij}^{k+1} . To accelerate the solution the values in the bracket is multiplied by ω, the relaxation parameter.

So the equation becomes,

$$u_{i,j}^{k+1} = u_{i,j}^{k} + \frac{\omega}{2(1+\beta^2)}\left[u_{i+1,j}^{k} + u_{i-1,j}^{k+1} + \beta^2(u_{i,j+1}^{k} + u_{i,j-1}^{k+1}) - 2(1+\beta^2)u_{i,j}^{k}\right]$$

For the solution to converge it is necessary that 0 < ω < 2. If 0 < ω < 1 it is called under relaxation, the above equation is rearranged as,

$$u_{i,j}^{k+1} = (1-\omega)u_{i,j}^{k} + \frac{\omega}{2(1+\beta^2)}\left[u_{i+1,j}^{k} + u_{i-1,j}^{k+1} + \beta^2(u_{i,j+1}^{k} + u_{i,j-1}^{k+1})\right]$$

……Equation 1.86

1.9 Explicit Finite Difference Method of Supersonic Flows – Hyperbolic Equations

The model equation considered for studying the Explicit FDM methods for the Hyperbolic equations is First order wave equation,

$$\frac{\partial u}{\partial t} = -a\frac{\partial u}{\partial x} \qquad a > 0$$

……Equation 1.87

Which is linear equation for constant speed a.

Euler's FTFS method

In this explicit method, forward time and forward space approximations of the first-order are used, the resulting Finite Difference Equation (FDE) is

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} = -a\frac{u_{i+1}^n - u_i^n}{\Delta x}$$

……Equation 1.88

Euler's FTCS method

In this formulation central differencing of special derivative is used, the resulting FDE is

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} = -a\frac{u_{i+1}^n - u_{i-1}^n}{2\Delta x}$$

……Equation 1.89

The First Upwind Differencing Method

The backward differencing of the special derivative produces the FDE,

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} = -a\frac{u_i^n - u_{i-1}^n}{\Delta x}$$

……Equation 1.90

This method is stable when c is less than or equal to 1.

$$c \leq 1, \text{ where } c = a\Delta t/\Delta x$$

It is the Courant number. The FDE for the conditionally stable solution is

$$\frac{u_i^{n+1}-u_i^n}{\Delta t}=-a\frac{u_{i+1}^n-u_i^n}{\Delta x}$$

……Equation 1.91

The Lax method

If an average value of u_i^n in the Euler's FTCS method is used , we get a FDE of the form,

$$u_i^{n+1}=\frac{1}{2}(u_{i+1}^n+u_{i-1}^n)-\frac{a\Delta t}{2\Delta x}(u_{i+1}^n-u_{i-1}^n)$$

……Equation 1.92

This method is stable when, $c \leq 1$

Midpoint Leapfrog method

In this method, Central differencing of the second order is used of both the time and space derivative. This gives the FDE,

$$\frac{u_i^{n+1}-u_i^{n-1}}{2\Delta t}=-a\frac{u_{i+1}^n-u_{i-1}^n}{2\Delta x}$$

……Equation 1.93

This is of the order $[(\Delta t)^2,\ (\Delta x)^2]$. This method is stable when, $c \leq 1$

This requires the two sets of the initial values to start the solution. The Midpoint Leapfrog method has a higher order of accuracy.

The Lax-Wendroff method

This finite difference approximation of the PDE is derived from the Taylor series expansion of the dependent variable as follows.

$$u(x,\ t+\Delta t)=u(x,t)+\frac{\partial u}{\partial t}\Delta t+\frac{\partial^2 u}{\partial t^2}\frac{(\Delta t)^2}{2!}+O(\Delta t)^3$$

……Equation 1.94

In the terms of the indices

$$u_i^{n+1}=u_i^n+\frac{\partial u}{\partial t}\Delta t+\frac{\partial^2 u}{\partial t^2}\frac{(\Delta t)^2}{2!}+O(\Delta t)^3$$

……Equation 1.95

Now consider the model equation

$$\frac{\partial u}{\partial t} = -a\frac{\partial u}{\partial x}$$

……Equation 1.96

By taking the time derivative we get,

$$\frac{\partial^2 u}{\partial t^2} = -a\frac{\partial}{\partial t}\left(\frac{\partial u}{\partial x}\right) = -a\frac{\partial}{\partial x}\left(\frac{\partial u}{\partial t}\right) = a^2\frac{\partial^2 u}{\partial x^2}$$

……Equation 1.97

Substitute both the equations in the indices terms shown above we get,

$$u_i^{n+1} = u_i^n + \left(-a\frac{\partial u}{\partial x}\right)\Delta t + \frac{(\Delta t)^2}{2}\left(a^2\frac{\partial^2 u}{\partial x^2}\right)$$

……Equation 1.98

By applying the space derivative of the first and second order derivatives, we get

$$u_i^{n+1} = u_i^n - a\Delta t\left[\frac{u_{i+1}^n - u_{i-1}^n}{2\Delta x}\right] + \frac{1}{2}a^2(\Delta t)^2\left[\frac{u_{i+1}^n - 2u_i^n + u_{i-1}^n}{(\Delta x)^2}\right]$$

……Equation 1.99

This formulation is known as the Lax-Wendroff method, this method is stable for $c \leq 1$

1.10 Explicit Finite Difference Method of Viscous Flows – Parabolic Equations

The Forward Time/Central Space (FTCS) method

In this method forward difference approximation for the time derivative and central differencing for the space derivative which gives,

$$u_i^{n+1} = u_i^n + \frac{\alpha(\Delta t)}{(\Delta x)^2}\left(u_{i+1}^n - 2u_i^n + u_{i-1}^n\right)$$

……Equation 1.100

The above equation is stable for $\alpha\Delta t/(\Delta x)^2 \leq 1/2.$

The Richardson method

In this approximation method central differencing is used for both time and space derivatives, the resulting FDE is,

$$\frac{u_i^{n+1} - u_i^{n-1}}{2\Delta t} = \alpha\frac{u_{i+1}^n - 2u_i^n + u_{i-1}^n}{(\Delta x)^2}$$

……Equation 1.101

The above equation is unconditionally unstable and has no practical value.

The DuFort-Frankel method

In this formulation the time derivative is approximated by a central differencing and the second order space derivative is also approximated by the central differencing method. Due to stability constrains u_i^n in the right hand side is replaced by the average value. This is the modification of the Richardson method. The resulting FDE is

$$\frac{u_i^{n+1} - u_i^{n-1}}{2\Delta t} = \alpha \frac{u_{i+1}^n - 2\dfrac{u_i^{n+1} + u_i^{n-1}}{2} + u_{i-1}^n}{(\Delta x)^2}$$

……Equation 1.102

From which

$$u_i^{n+1} = u_i^{n-1} + \frac{2\alpha(\Delta t)}{(\Delta x)^2}\left[u_{i+1}^n - u_i^{n+1} - u_i^{n-1} + u_{i-1}^n\right]$$

……Equation 1.103

This can be rewritten as

$$\left[1 + \frac{2\alpha(\Delta t)}{(\Delta x)^2}\right] u_i^{n+1} = \left[1 - 2\frac{\alpha(\Delta t)}{(\Delta x)^2}\right] u_i^{n-1} + \frac{2\alpha(\Delta t)}{(\Delta x)^2}\left[u_{i+1}^n + u_{i-1}^n\right]$$

……Equation 1.104

This method is of the order of ,

$$[(\Delta t)^2,\ (\Delta x)^2,\ (\Delta t/\Delta x)^2].$$

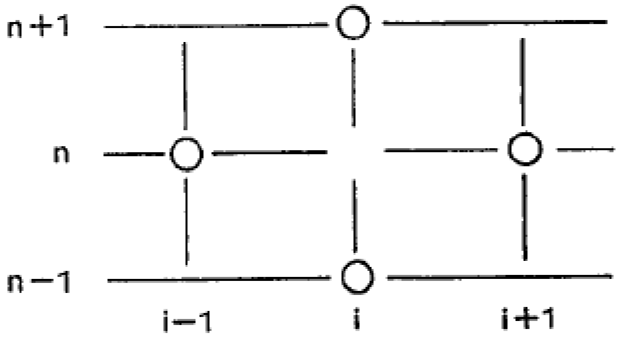


Figure 1.22 Grid points for the DuFort-Frankel method

UNIT II

GRID GENERATION

- Structured Grids
- Types and Transformations
- Generation of Structured Grids
- Unstructured Grids
- Delany Triangulation

2. GRID GENERATION

The governing partial differential equations (PDEs) of the fluid mechanics are solved numerically by converting the partial equations in to appropriate finite difference expressions which are used to rewrite PDEs as algebraic equations. These Finite Difference Equations (FDEs) are solved at the discrete points within the domain of interest; these discrete points are called Grids.

Structured Grid - The computational domain selected to be rectangular in shape where the interior grid points are distributed along grid lines. The grid points can be identified easily with reference to the appropriate grid lines. This type of grid is known as the ***structured grid.***

Unstructured Grid – The grid system that is constructed where the grid points cannot be associated with the orderly defined grid lines. This type grid system is known as the ***unstructured grid.***

Initial and Boundary Conditions

In order to obtain a unique solution of a PDE, a set of supplementary conditions must be provided to determine the arbitrary functions which result from the integration of the PDE, The supplementary conditions are classified as boundary and initial conditions.

An initial condition is a requirement for which the dependent variable is specified at some initial state.

A boundary condition is a requirement that the dependent variable or its derivative must satisfy on the boundary of the domain of the PDE.

Various types of boundary conditions which will be encountered are,

1. The Dirichlet boundary condition – If the dependent variable along with the boundary is prescribed, it is known as the Dirichlet type.

2. The Neumann boundary condition – If the normal gradient of the dependent variable along with the boundary is specified, it is called the Neumann type.

3. The Robin boundary condition – If the imposed boundary condition is a linear combination of the Dirichlet and Neumann types, it is known as the Robin type.

4. The Mixed boundary condition – The boundary condition along a certain portion of the boundary is the Dirichlet type and on another portion of the boundary, a Neumann type. This type is known as a mixed boundary condition.

2.1 Structured Grids

Structured Grid - The computational domain selected to be rectangular in shape where the interior grid points are distributed along grid lines. The grid points can be identified easily with reference to the appropriate grid lines. This type of grid is known as the ***structured grid.***

The generation of grid with uniform spacing is the simplest within a rectangular physical domain. Grid points may be specified as coincident with the boundaries of the physical domain, thus making specification of boundary conditions considerably less complex. Unfortunately the majority of the physical domain of interest are nonrectangular, Therefore imposing rectangular computational domain on such physical domain will require some sort of interpolation for the implementation of the boundary condition. Since the boundary condition have a dominant influence on the solution of the equation, such an interpolation causes inaccuracies at the place of greatest sensitivity.

2.2 Types and Transformations

A transformation from physical space to computational space is introduce to overcome some of the difficulties such as unequal grid spacing near the boundaries and inaccuracies at the place of sensitivity. This transformation is accomplished by specifying a generalized coordinate system which will map the nonrectangular grid system in the physical space to a rectangular uniform grid spacing in the computational space.

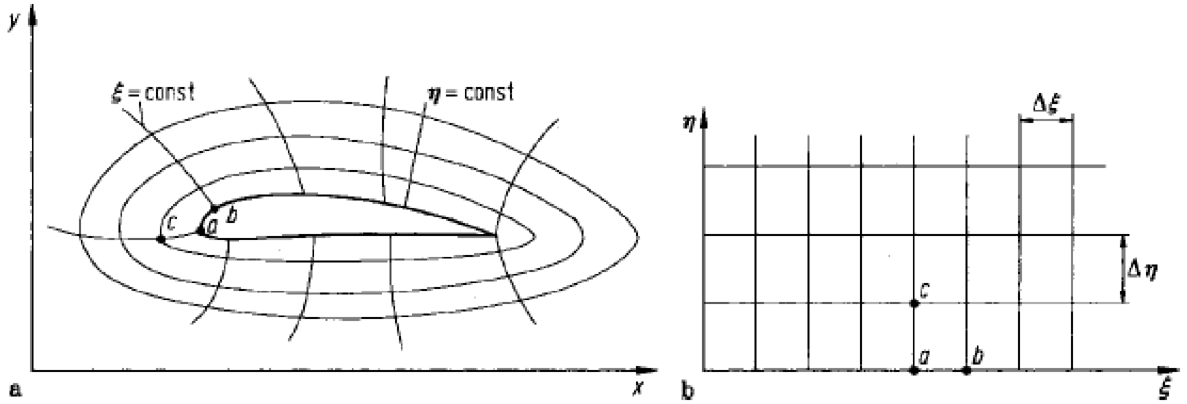


Figure 2.1 Transformation from physical space to computational space

In general grid generation techniques are classified based on complex variables as,

1. Algebraic methods

2. Partial differential methods

3. Conformal mapping

Conformal mapping is based on complex variables and are limited to 2-D problems so this will not be discussed.

Transformations of the Governing Partial Differential Equations

The relationship between the physical and computational space are given by,

$$\begin{aligned}\xi &= \xi(x,y)\\ \eta &= \eta(x,y)\end{aligned}$$

.........Equation 2.1

The chain rule for partial differentiation yields the following expression;

$$\frac{\partial}{\partial x} = \frac{\partial \xi}{\partial x}\frac{\partial}{\partial \xi} + \frac{\partial \eta}{\partial x}\frac{\partial}{\partial \eta}$$

.........Equation 2.2

The above equation can be written as,

$$\frac{\partial}{\partial x} = \xi_x \frac{\partial}{\partial \xi} + \eta_x \frac{\partial}{\partial \eta}$$

$$\frac{\partial}{\partial y} = \xi_y \frac{\partial}{\partial \xi} + \eta_y \frac{\partial}{\partial \eta}$$

.........Equation 2.3

Now consider a model PDE

$$\frac{\partial u}{\partial x} + a\frac{\partial u}{\partial y} = 0$$

The above equation can be transformed from the physical to the computational space by using the equations 2.3, we get

$$\xi_x \frac{\partial u}{\partial \xi} + \eta_x \frac{\partial u}{\partial \eta} + a\left(\xi_y \frac{\partial u}{\partial \xi} + \eta_y \frac{\partial u}{\partial \eta}\right) = 0$$

This can be rearranged as

$$(\xi_x + a\xi_y)\frac{\partial u}{\partial \xi} + (\eta_x + a\eta_y)\frac{\partial u}{\partial \eta} = 0$$

.........Equation 2.4

This equation is the one which will be solved in the computational domain. The transformation derivatives, ξ_x, ξ_y, η_x, and $\eta_y$are defined from the equation 2.1. Comparing the original equation and the transformed equation , the transformed equation is more complicated than the original equation. The advantage outweigh the complexity of the transformed PDE.

Metrics and the Jacobian of Transformation

Recall the equation 2.3. The terms such as ξ_x, ξ_y, η_x, and η_y appear.

$$\frac{\partial}{\partial x} = \xi_x \frac{\partial}{\partial \xi} + \eta_x \frac{\partial}{\partial \eta}$$

$$\frac{\partial}{\partial y} = \xi_y \frac{\partial}{\partial \xi} + \eta_y \frac{\partial}{\partial \eta}$$

These transformation derivatives are defined as the metrics of transformation or simply metrics. The interpolation of the metrics is obvious considering the following approximation;

$$\xi_x = \frac{\partial \xi}{\partial x} \cong \frac{\Delta \xi}{\Delta x}$$

……….Equation 2.5

This expression indicates that the metrics represent the ratio of the arc length in the computational space to that of the physical space.

From equation 2.1. the following differential expressions are obtained,

$$\begin{aligned} d\xi &= \xi_x dx + \xi_y dy \\ d\eta &= \eta_x dx + \eta_y dy \end{aligned}$$

……….Equation 2.6

This can be written in the compact form as,

$$\begin{bmatrix} d\xi \\ d\eta \end{bmatrix} = \begin{bmatrix} \xi_x & \xi_y \\ \eta_x & \eta_y \end{bmatrix} \begin{bmatrix} dx \\ dy \end{bmatrix}$$

……….Equation 2.7

Reversing the role of independent variables, we get,

$$\begin{aligned} x &= x(\xi, \eta) \\ y &= y(\xi, \eta) \end{aligned}$$

……….Equation 2.8

The above equation can be written as,

$$\begin{aligned} dx &= x_\xi d\xi + x_\eta d\eta \\ dy &= y_\xi d\xi + y_\eta d\eta \end{aligned}$$

……….Equation 2.9

In a compact form it may be written as,

$$\begin{bmatrix} dx \\ dy \end{bmatrix} = \begin{bmatrix} x_\xi & x_\eta \\ y_\xi & y_\eta \end{bmatrix} \begin{bmatrix} d\xi \\ d\eta \end{bmatrix}$$

……….Equation 2.10

Comparing equation 2.7 and 2.10 it can be concluded as,

$$\begin{bmatrix} \xi_x & \xi_y \\ \eta_x & \eta_y \end{bmatrix} = \begin{bmatrix} x_\xi & x_\eta \\ y_\xi & y_\eta \end{bmatrix}^{-1}$$

……….Equation 2.11

From which we can conclude that

$$\begin{aligned} \xi_x &= Jy_\eta \\ \xi_y &= -Jx_\eta \\ \eta_x &= -Jy_\xi \\ \eta_y &= Jx_\xi \end{aligned}$$

$$J = \frac{1}{x_\xi y_\eta - y_\xi x_\eta}$$

J is defined as the Jacobian of Transformation.

The Jacobian J is interpreted as the ration of the areas in 2D and ratio of volumes in 3D in the computational space to that of the physical space.

2.3 Generation of Structured Grids

Grid systems with the following features are desired;

1. A mapping which guarantees one-to-one correspondence ensuring grid lines of the same family do not cross each other;
2. Smoothness of the grid point distribution;
3. Orthogonality or near Orthogonality of the grid lines;
4. Options for grid point clustering;

Algebraic Grid Generation Techniques

The simplest grid generation is the algebraic method. The major advantage of this scheme is the speed with which a grid can be generated. An algebraic equation is used to relate the grid points in the computational domain to those of the physical domain. This objective is met by using an interpolation scheme between the specified boundary grid points to generate the interior grid points.

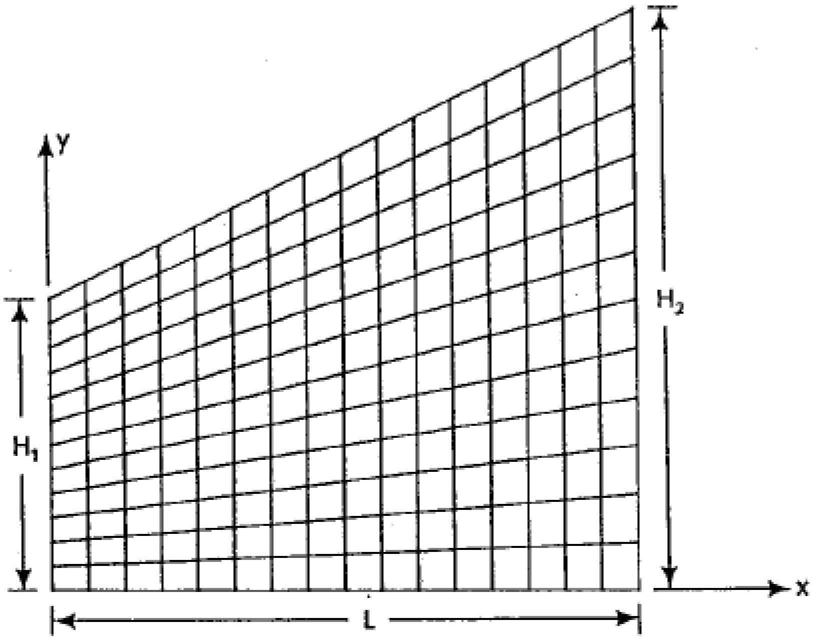


Figure 2.2 The physical space which must be transformed

Consider the simple physical domain depicted in the figure 2.2. Introducing the following algebraic relations will transform this non rectangular physical domain into a rectangular domain:

$$\xi = x$$
$$\eta = \frac{y}{y_t}$$

……….Equation 2.12

y_t represents the upper boundary which is expressed as

$$y_t = H_1 + \frac{H_2 - H_1}{L}x$$

Thus the equation 1.22 can be written as

$$\xi = x$$
$$\eta = \frac{y}{H_1 + \frac{H_2 - H_1}{L}x}$$

By rearranging the terms we get,

$$x = \xi$$

$$y = \left(H_1 + \frac{H_2 - H_1}{L}\xi\right)\eta$$

.........Equation 2.13

The grid system is generated as follows; the geometry in the physical space is defined. For this problem by specifying values of L, H_1 and H_2. Next the desired number of grid points defined by IM – the maximum number of grid points in ξ, and JM – the maximum number of grid points in η is specified. The equal grid spacing in the computational domain is produced as follows;

$$\Delta\xi = \frac{L}{IM - 1}$$

$$\Delta\eta = \frac{1.0}{JM - 1}$$

.........Equation 2.14

In the above equation η is normalized, its value varies from zero to one.

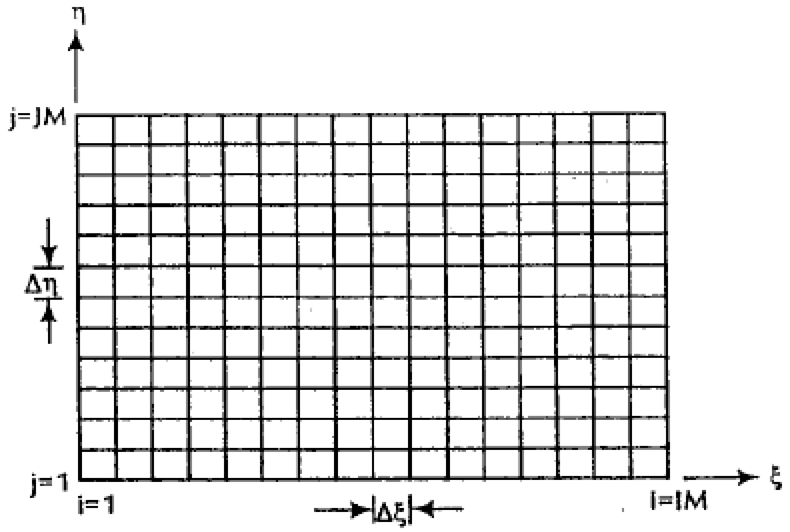


Figure 2.3 The rectangular computational domain with uniform grid spacing

The values of ξ and η are known at each grid point within the domain. The equation 2.13 can be used to identify the corresponding grid points in the physical space.

The metrics and the Jacobian of the transformation must be evaluated before any transformed PDEs can be solved. An algebraic model is used the metrics are calculated analytically. This aspect is an advantage of the algebraic methods since numerical computation of metrics will require additional computation time and may introduce some errors into the system of equations of motion that are to be solved.

Advantages:

1. Computationally they are very fast
2. Metrics may be evaluated analytically, thus avoiding numerical errors
3. The ability of cluster grid points in different regions can be easily implemented.

Disadvantages

1. Discontinuities at a boundary may propagate into the interior regions which could lead to errors due to sudden change in the metrics
2. Control of grid smoothness and skewness is a difficult task.

Partial Differential Equation Techniques

Elliptical Grid Generators

Simply-Connected Domain

Doubly-Connected Domain

Multiply-Connected Domain

Coordinate System Control

Grid Point Clustering

Orthogonality at the Surface

Hyperbolic Grid generation Techniques

Parabolic Grid Generators

UNIT III

DISCRETIZATION

Boundary Layer Equations and Methods of Solution-Implicit Time Dependent Methods for Inviscid and Viscous Compressible Flows-Concept of Numerical Dissipation-Stability Properties of Explicit and Implicit Methods-Conservative Upwind Discretization for Hyperbolic Systems-Further Advantages of Upwind Differencing

3.1 Boundary Layer Equations and Methods of Solution

Prandtl made an important contribution to the calculation of a specific type of flow for which the Reynolds number is very large. The Reynolds number has the form of a non-dimensional parameter

$$\mathrm{Re} = \frac{LV}{\nu} = \frac{\rho LV}{\mu} \qquad \text{------------------------- } 1$$

Where *L* is a characteristic length, usually the length of the considered body, *V* is the velocity of the flow where it is well-defined and undisturbed. The kinematic and dynamic viscosity is denoted by *v* and μ, respectively. The density of the fluid is ρ. The Reynolds number is the ratio of inertia to friction forces following the ‘principle of similarity

$$\mathrm{Re} = \frac{\rho u \partial u/\partial x}{\mu \partial^2 u/\partial x^2} \equiv \frac{\text{inertia force}}{\text{friction force}} \qquad \text{--------------- } 2$$

The velocity *u* at some point in the velocity field is proportional to the free stream velocity *V*. The velocity gradient $\partial u/\partial x$ is proportional to V/L and similarly $\partial^2 u/\partial x^2$ is proportional to V/L^2. Hence the ratio, Eq. (2) yields

$$\mathrm{Re} = \frac{\rho V^2/L}{\mu V/\mathrm{d}^2} = \frac{\rho LV}{\mu} \qquad \text{---------------- } 3$$

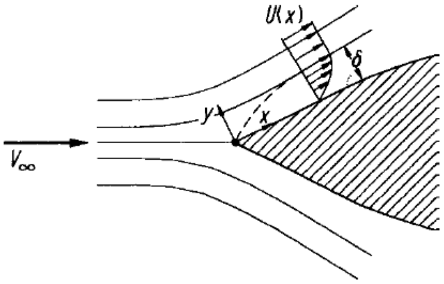


Figure 1:Boundary layer flow along a wall

Two flows are similar from the point of view of the relative importance of inertial and viscous effects if the Reynolds number is constant. Now the physical phenomenon of a flow with high Reynolds number is considered for the example of a cylindrical body shown in Fig. 1.

With the exception of the immediate neighborhood of the surface the flow velocity is comparable to the free stream velocity *V*. This flow region is nearly free of friction; it is a potential flow. Considering the region near the surface there is friction in the flow which means that the fluid is retarded until it adheres at the surface. The transition from zero velocity at the surface to the full magnitude at some distance from it takes place in a very thin layer, the so-called ‘boundary layer’. Its thickness is δ, which is a function of the downstream coordinate *x* and is assumed to be very small compared to the length of the body *L*. In the normal direction *y* inside the thin layer it is clear that the gradient $\partial u/\partial y$ is very large compared to gradients in the stream wise direction $\partial u/\partial x$. Although the viscosity was meant to be very small in this flow the shear stress $\tau = \mu(\partial u/\partial y)$ may assume large values. Outside the boundary layer the velocity gradients are negligibly small and the influence of the viscosity is unimportant. The flow is frictionless and potential.

The above assumptions are now used to simplify the Navier–Stokes equations for steady two-dimensional, laminar and incompressible flows, resulting from the non-conservation form by a formal procedure. Including the continuity equation they have the following dimensional form in Cartesian coordinates

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

---------------------------- 4

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

--------------------------- 5

$$\frac{\partial \bar{u}}{\partial \bar{x}} + \frac{\partial \bar{v}}{\partial \bar{y}} = 0$$

-------------------- 6

Here the velocity components ¯u and ¯v are directed towards the downstream ¯x and the normal ¯y-direction, respectively. The static pressure is denoted by ¯p, ¯ρ is the density and ¯μ is the dynamic viscosity of the fluid.

For convenience, this set of second order differential equations is non-dimensionalized which involves the Reynolds number necessary for the following reduction of the equations. The prescriptions for non dimensionalization are:

$$\begin{aligned} u &= \frac{\bar{u}}{V} = 0(1) \\ v &= \frac{\bar{v}}{V} = 0(\varepsilon) \\ p &= \frac{\bar{p}}{\bar{\rho}V^2} = 0(1) \qquad \mathrm{Re} = \frac{\bar{\rho}LV}{\bar{\mu}} = 0\left(\frac{1}{\varepsilon^2}\right) \\ x &= \frac{\bar{x}}{L} = 0(1) \\ y &= \frac{\bar{y}}{L} = 0(\varepsilon) \end{aligned}$$

----------------- 7

V is the dimensional free stream velocity and the pressure is non-dimensionalized by twice the dynamic pressure,

$$\bar{q} = \tfrac{1}{2}\bar{\rho}V^2$$

Using these definitions, Eqs. (4), (5) and (6) become:

$$u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} = -\frac{\partial p}{\partial x} + \frac{1}{\mathrm{Re}}\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right)$$

$$(1)\frac{(1)}{(1)}(\varepsilon)\frac{(1)}{(\varepsilon)} \qquad (1) \quad (\varepsilon^2)\left(\frac{(1)}{(1)} \quad \frac{(1)}{(\varepsilon^2)}\right)$$

--------------------- 8

$$u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y} = -\frac{\partial p}{\partial y} + \frac{1}{\mathrm{Re}}\left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2}\right)$$

$$(1)\frac{(\varepsilon)}{(1)} \quad (\varepsilon)\frac{(\varepsilon)}{(\varepsilon)} \qquad (\varepsilon^2)\left(\frac{(\varepsilon)}{(1)} \qquad \frac{(\varepsilon)}{(\varepsilon^2)}\right)$$

--------------- 9

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

$$\frac{(1)}{(1)} \quad \frac{(\varepsilon)}{(\varepsilon)}$$

----------------------- 10

Now the question is, what order of magnitude do the dimensionless substitutions Eqs. (8.7) have? As stated above, the boundary layer thickness δ is very small, so is the distance *y* compared to the length of the body *L*. Consequently *y* is of the order ε which describes a value much smaller than 1. The *u*-velocity component can reach the maximum value of *V*, therefore it is of the order 1. But the *v*-velocity component also has to be of the order ε as can be seen from the continuity equation, Eq. (8.10).If the derivative $\partial u/\partial x$ is of the order 1 because *x* becomes, at its maximum, the length *L*, then the second term in the continuity equation $\partial v/\partial y$ has also to be of the order 1. Consequently, *v* is not greater than ε. Now, with these assumptions the order of magnitude analysis can be done. It follows from the first equation of motion, Eq. (8.8), that the viscous forces in the boundary layer can become of the same order of magnitude as the inertia forces only if the Reynolds number is of the order of $1/\varepsilon^2$. The equation of continuity remains unaltered for very large Reynolds numbers.

The equation of continuity remains unaltered for very large Reynolds numbers. The downstream momentum equations can be reduced by the second derivative of the *u*-velocity component $\partial^2 u/\partial x^2$ and multiplied by 1/Re because it has the smallest order of magnitude in this equation. It only holds that the forcing function term ($-\mathrm{d}p/\mathrm{d}x$) will not exceed the order of 1 to be in balance with the other remaining terms.

All terms of the normal momentum equation, Eq. (8.9), are of a smaller magnitude than those of Eq. (8.8). This equation can only be in balance if the pressure term is of the same order of magnitude. Therefore, this equation delivers the information of negligible pressure gradient in the normal direction, i.e.

$$\frac{\partial p}{\partial y} = 0(\varepsilon)$$

The meaning of this result is that the pressure is practically constant; it is ‘impressed’ on the boundary layer by the outer flow. Therefore, the pressure *p* is only a function of x.

The derivation of Eq. (8.8) at the outer edge of the boundary layer gives, if the Inviscid velocity distribution $U(x) = {}^{-}u(x)/V$ is known:

$$U\frac{\mathrm{d}U}{\mathrm{d}x} = -\frac{1}{\rho}\frac{\mathrm{d}\rho}{\mathrm{d}x} \qquad \text{-----------------}$$

The other terms involving $\partial u/\partial y$ are zero since there remains no large velocity gradient. After integration of Eq. (8.12) the well-known Bernoulli equation is found:

$$p + \frac{1}{2}\rho U^2 = \text{const.}$$

Summing up , by the order of magnitude analysis the Navier–Stokes equations, Eqs. (8) and (9), and the continuity Eq. (10), have been simplified. They are known as ‘Prandtl’s boundary layer equations’:

$$u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} = -\frac{\partial p}{\partial x} + \frac{1}{\text{Re}}\frac{\partial^2 u}{\partial y^2} \quad \text{-----------11}$$

$$\frac{\partial p}{\partial y} = 0 \quad \text{------------------ 12}$$

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0 \quad \text{---------13}$$

The boundary conditions are:
On the surface:

$$y = 0 \qquad u = 0, \qquad v = 0 \quad \text{------14}$$

On the outer edge of the boundary layer:

$$y = \delta = \frac{\bar{\delta}}{L} \qquad u = U(x) \quad \text{-----------15}$$

This set of equations is reduced by the unknown pressure *p*, which is, because of Bernoulli’s equation, Eq. (8.13), a known value now, if only the inviscid velocity distribution at the surface *U*(*x*) is provided. It is still a coupled, non-linear, second order set of differential equations.

The order of magnitude analysis also described by Schlichting [6] is well suited to analyse the more complicated surface-oriented Navier–Stokes equations with additional surface curvature created Coriolis and centrifugal forces. At least the order of magnitude analysis gives an impression where the boundary layer equations and their more complicated extensions are situated in their level of approximation to the full Navier–Stokes equations. This overview will be given in the next section.

HIERARCHY OF THE BOUNDARY LAYER EQUATIONS

To develop a hierarchy of the fluid mechanical equations, the steady, compressible, laminar, two-dimensional Navier–Stokes equations should be written for the Euclidian space in a layer close to the surface. This will say that a coordinate system, which may be surface oriented for a better adaption to the flow problem considered, is related to the cartesian coordinate system. Both systems must be transferable from one to the other. The cartesian and the polar coordinate system, for example, are matched together following this demand of Euclidian space. In other words, the Jacobian matrix must exist.

If the Navier–Stokes equations can be formulated for such a surface-oriented coordinate system, they will contain many additional terms due to the surface curvature. These terms can be understood as Coriolis and centrifugal force terms caused by the change of the streamlines in downstream as well as in the cross flow direction depending on the curvature of the surface. Curvature-induced terms will have different orders of magnitude. Some are important and others can be neglected depending on the specific flow problems.

Now the question is to set the boundary layer equations including curvature terms in relation to Prandtl’s boundary layer equations developed in the foregoing chapter.

A simple two-dimensional surface-oriented coordinate system is fixed on an airfoil-like contour sketched in Fig. 1. The relations between the new coordinate system and the cartesian one are:

$$x = \int_0^s \cos\theta(x)\,\mathrm{d}s - n\sin\theta(x) \qquad \text{--- 1}$$

$$y = \int_0^s \sin\theta(x)\,\mathrm{d}s + n\cos\theta(x) \qquad \text{-- 2}$$

The resultant set of differential equations due to the coordinate transformation consists of two equations of motion in the downstream direction *s* and the perpendicular direction *n*, the energy and the continuity equations.

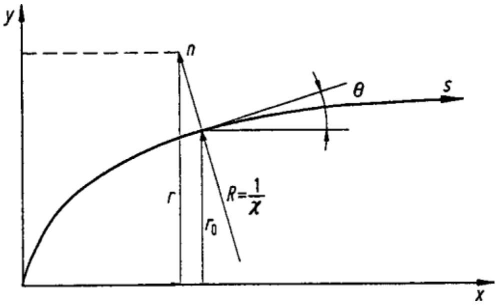


Fig. 1 Surface oriented coordinate system

Momentum equation in tangential direction:

$$\rho\left[u\frac{\partial u}{\partial s}+Hv\frac{\partial u}{\partial n}+\kappa uv\right]=-\frac{\partial p}{\partial s}+\frac{\partial}{\partial s}\left[\frac{4}{3}\frac{\mu}{H}\frac{\partial u}{\partial s}+\frac{4}{3}\frac{\kappa\mu v}{H}-\frac{2}{3}\mu\frac{\partial v}{\partial n}\right]$$
$$+H\frac{\partial}{\partial n}\left[\frac{\mu}{H}\frac{\partial v}{\partial s}+\mu\frac{\partial u}{\partial n}-\frac{\mu\kappa u}{H}\right] \qquad \text{----------------3}$$

Momentum equation in normal direction:

$$\rho\left[u\frac{\partial v}{\partial s}+Hv\frac{\partial v}{\partial n}-\kappa u^2\right]=-H\frac{\partial p}{\partial n}+H\frac{\partial}{\partial n}\left[\frac{4}{3}\mu\frac{\partial v}{\partial n}-\frac{2}{3}\frac{\mu}{H}\frac{\partial u}{\partial s}-\frac{2}{3}\frac{\mu\kappa v}{H}\right]$$
$$+\frac{\partial}{\partial s}\frac{\mu}{H}\frac{\partial v}{\partial s}+\mu\frac{\partial u}{\partial n}-\frac{\mu\kappa u}{H}$$
$$+2\kappa\left[\mu\frac{\partial v}{\partial n}-\frac{\mu}{H}\frac{\partial u}{\partial s}-\frac{\mu\kappa v}{H}\right] \qquad \text{---------- 4}$$

Energy equation:

$$c_p\rho\left[u\frac{\partial T}{\partial s}+Hv\frac{\partial T}{\partial n}\right]=u\frac{\partial p}{\partial s}+Hv\frac{\partial p}{\partial n}+\frac{\partial}{\partial s}\left[\lambda\frac{\partial T}{\partial s}\right]+H\frac{\partial}{\partial n}\left[\lambda\frac{\partial T}{\partial n}\right]$$
$$+H\frac{\mu}{2}\left\{\left[\frac{2}{H}\frac{\partial u}{\partial s}+2\frac{\kappa v}{H}\right]^2\right.$$
$$\left.+\left[2\frac{\partial v}{\partial n}\right]^2+2\left[\frac{1}{H}\frac{\partial v}{\partial s}+\frac{\partial u}{\partial n}-\frac{\kappa u}{H}\right]^2\right\}$$
$$-\frac{2}{3}H\mu\left[\frac{1}{H}\frac{\partial u}{\partial s}+\frac{\kappa v}{H}+\frac{\partial v}{\partial n}\right]^2 \qquad \text{---------------- 5}$$

Continuity equation:

$$\frac{\partial(\rho u)}{\partial s}+\frac{\partial(H\rho v)}{\partial n}=0 \qquad \text{------------- 6}$$

$$H=1+\kappa n=\frac{R+n}{R}$$

Here *u* and *v* are the velocity components in the tangential direction of the flow *s* and the normal direction *n*, respectively. The pressure is denoted by *p*, ρ is the density, μ and λ are the dynamic viscosity and the thermal heat conductivity, respectively. The curvature of the surface is involved in the geometrical coefficient *H*. This dimensional set of differential equations describes the laminar, compressible flow along arbitrary, two-dimensional curved surfaces.

Now these governing equations are analysed by predicting the order of magnitude of each term. As is usually done, the equations will be non-dimensionalized, the geometrical quantities by a characteristic length *L* and the flow properties by their free stream conditions denoted by subscript ∞. The order of magnitude of these quantities is defined as has been done in the case of a simple boundary layer without curvature in the preceding chapters.

$$\begin{array}{lll}
s = \dfrac{\bar{s}}{L} = 0(1), & n = \dfrac{n}{L} = 0(\varepsilon), & \kappa = \bar{\kappa}L = 0(1) \\
H = 1 + \bar{\kappa}\bar{\eta} = 0(1), & u = \dfrac{\bar{u}}{u_\infty} = 0(1), & v = \dfrac{\bar{v}}{u_\infty} = 0(\varepsilon) \\
T = \dfrac{\bar{T}}{T_\infty} = 0(1) & p = \dfrac{\bar{p}}{\rho_\infty u_\infty^2} = 0(1) & \rho = \dfrac{\bar{\rho}}{\rho_\infty} = 0(1) \\
\mu = \dfrac{\bar{\mu}}{\mu_\infty} = 0(1) & \lambda = \dfrac{\bar{\lambda}}{\lambda_\infty} = 0(1) & c_p = \dfrac{\bar{c}_p}{c_{p\infty}} = 0(1) \\
Re = \dfrac{\rho_\infty u_\infty L}{\mu_\infty} = 0\left(\dfrac{1}{\varepsilon^2}\right) & \text{Reynolds number} & \\
Pr = \dfrac{c_{p\infty}\mu_\infty}{\lambda_\infty} = 0(1) & \text{Prandtl number} & \\
Ec = \dfrac{u_\infty^2}{c_{p\infty}T_\infty} = 0(1) & \text{Eckert number} &
\end{array} \qquad \text{------------------------ } 7$$

It is to be mentioned that the radius of curvature *R* is not allowed to be much larger than the characteristic length *L*, otherwise κ would belong to another order of magnitude. The radius of curvature *R* is related to the curvature as follows

$$\kappa = \bar{\kappa}L = \frac{L}{R} \qquad \text{------------ } 8$$

When the radius *R* becomes very small compared to the length, *H* can exceed the order demanded above.

The combination of Eq. (7) with the governing equations, Eqs . (3), (4), (5) and (6), provides the order of magnitude of each term. A detailed development of the order of magnitude analysis applied to this set of equations seems not to be necessary here because in the preceding chapter an example was already presented. But in order to give an insight into the origin of the hierarchy of the boundary layer equations, the equations will be shown that retain terms only of the order 0(1) and 0(ε). The chosen equation is the tangential and normal momentum equation in dimensional unbarred form.

Order 0(1):

$$\rho\left(u\frac{\partial u}{\partial s}+Hv\frac{\partial u}{\partial n}\right)=-\frac{\partial p}{\partial s}+H\frac{\partial}{\partial n}\left(\mu\frac{\partial u}{\partial n}\right) \quad \text{----- 9}$$

$$\frac{\partial p}{\partial n}=0 \quad \text{------ 10}$$

These equations, including the continuity equation, are called the ‘first order boundary layer equations’. Curvature effects are included in the quantity *H* defined in Eq. (6). These equations become identical to Prandtl’s boundary layer equations when the curvature goes to zero. Hence, Prandtl’s equations are the lowest level of the hierarchy and therefore they should be called ‘Zeroth order boundary layer equations’.

Now terms of the order 0(1) and 0(ε) are retained.

Order of magnitude 0(ε):

$$\rho\left(u\frac{\partial u}{\partial s}+Hv\frac{\partial u}{\partial n}+\kappa uv\right)=-\frac{\partial p}{\partial s}+H\frac{\partial}{\partial n}\left(\mu\frac{\partial u}{\partial n}\right)-\kappa\frac{\partial u}{\partial n}u+\kappa\mu\frac{\partial u}{\partial n} \quad \text{------8.29}$$

$$\frac{\partial p}{\partial n}=\frac{\kappa\rho u^2}{H} \quad \text{----------------8.30}$$

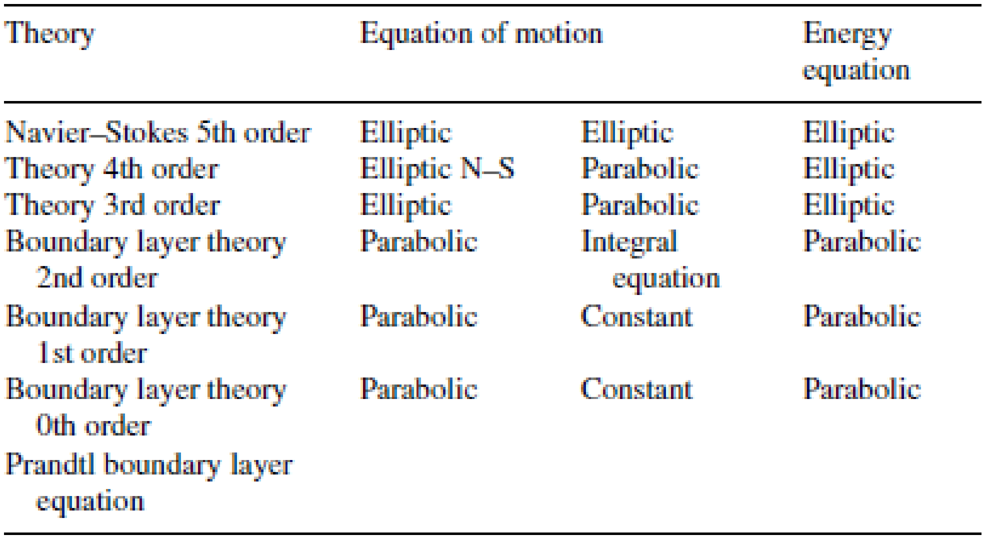
These equations show a significant extension of the foregoing ones. In Eq. (8.29) an additional centrifugal term *κuv* appears as well as dissipative terms due to curvature on the right-hand side; but the most important extension appears in the normal momentum equation, Eq. (8.30). The pressure gradient normal to the flow is no longer zero. Eq. (8.30) is an integral equation for the pressure which is no longer impressed on the boundary layer from the Inviscid flow. These equations are the so-called ‘second order boundary layer equations’ and take into account that, even in the outer Inviscid flow normal to the surface, there exist velocity gradients due to the streamline curvature. The outer edge of the boundary layer is matched to this gradient which is no longer equal to zero as the first order of boundary layer theory prescribes. Consequently terms of higher order than 0(ε) will be retained now. The result is summarized in Table 8.1.

A decisive development takes place proceeding from the second to the third-order set. The mathematical character of the equation changes from parabolic to elliptic. Elliptic differential equations are pure boundary value problems while parabolic equations are initial-boundary value problems. The latter can be solved by the so called ‘marching procedure’, but the former require the calculation of the entire flow field surrounded by the boundaries which implies a greater numerical effort.

The conclusion of this discussion is that a boundary layer theory of order higher than second order immediately leads to elliptic equations. This complicates the method of solution because the parabolic approach of the original idea of boundary layer theory no longer holds.

The subject of the following chapter will be to give an impression as to how transformations of the governing first-order boundary layer equations influence the solution techniques positively.

Table 8.1 Hierarchy of the boundary layer equations



Theory	Equation of motion		Energy equation
Navier–Stokes 5th order	Elliptic	Elliptic	Elliptic
Theory 4th order	Elliptic N–S	Parabolic	Elliptic
Theory 3rd order	Elliptic	Parabolic	Elliptic
Boundary layer theory 2nd order	Parabolic	Integral equation	Parabolic
Boundary layer theory 1st order	Parabolic	Constant	Parabolic
Boundary layer theory 0th order	Parabolic	Constant	Parabolic
Prandtl boundary layer equation			

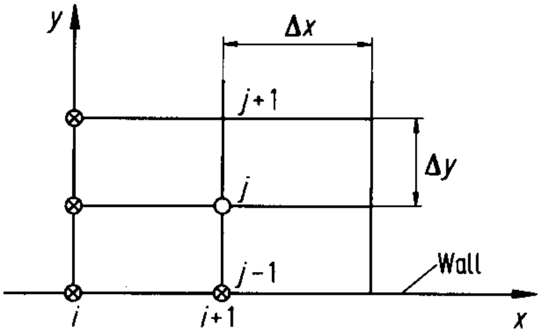
Numerical Solution Method:

Choice of Discretization Model

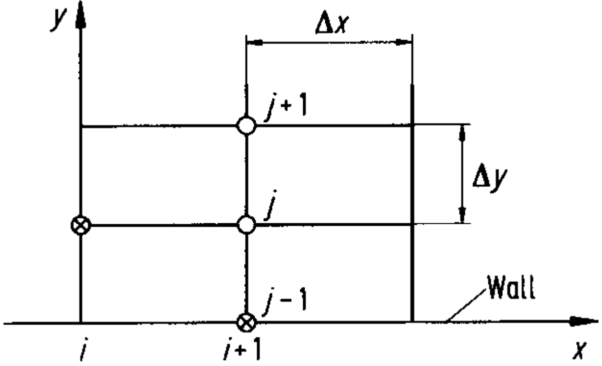
To come to a numerical solution of a set of partial differential equations it is usual to replace the differential quotients by finite difference quotients taking into account that a truncation error of a certain order of magnitude will now be induced to the set of equations. By rearranging the finite difference equations a system of algebraic equations is obtained which can be solved by means of the known methods. The techniques of the discretization are detailed in

Chap.5. It is stated there that the choice of the computational discretization grid is important as it affects the truncation error, the stability and the consistency. The form of these grids and the solution methods to which they lead will be summarized briefly.

Parabolic equations as observed here have a first order differential in the marching direction. As the flow is not allowed to reverse, the values of each quantity at the last upstream grid line normal to the surface are known. If we consider a grid as shown in Fig. 8.3, where Δx and Δy are the step sizes in the tangential and normal direction to the surface, the known points are on the left-hand side and the unknown on the right. Also the boundary conditions at the wall are given. Therefore, it is easy to calculate the flow quantities at the point with the open circle using discretization models as already given in Chap. 5. Because of the direct calculation of only one point on the grid line, this is called an 'explicit method'. The explicit method causes strong restrictions in the choice of the downstream step size as will briefly be repeated later, so the scheme is slow.



Grid for an explicit method

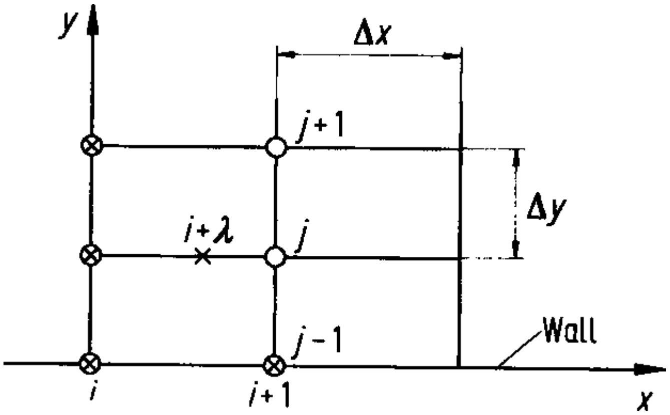


Grid for a fully implicit method

Figure 8.4 shows another extreme choice of a computational grid; the so-called 'fully implicit method'.

Only one known grid point from the preceding step is used, while on the actual one all points are unknown except the boundary values. That leads to an implicit form of the set of algebraic equations as will be shown later. This method is, concerning the choice of the step size, unconditionally stable but may lead to a poor accuracy. If there is no restriction on the step size in the downstream direction it becomes a fast calculation method which is desirable.

Now it is obviously possible to formulate something in between these extremes which will result in both a fast and accurate solution method. Figure 8.5 gives the computational mesh proposed by Crank–Nicholson [13] but in a more general form, so that the discretization methods described before are contained within it as special cases. Here, all points of the known and unknown grid lines are involved, but now the centre of discretization is located at the point *i* + λ. λ = 1/2 was originally proposed by Crank–Nicholson. Although the pure Crank–Nicholson scheme was described in detail in Part I, an example of a linear model equation is utilized to show its discrimination by the more generalized Crank–Nicholson scheme. In a following section the application to the two dimensional, rotational compressible boundary layer equations will be given.



Grid for a generalized implicit method

Generalized Crank–Nicholson Scheme

This section is taken directly from Arina & Benocci [5]. In order to analyse the stability and accuracy of a generalization of the Crank–Nicholson scheme, it isconvenient to utilize the linear model Eq. (8.40),

$$\frac{\partial \phi}{\partial x} = a\frac{\partial^2 \phi}{\partial y^2} \quad \text{------------8.40}$$

Equation (8.40) is discretized around the mesh point (*i* + λ, *j*), with λ ranging between 0 and 1. For λ = 0 an explicit scheme is recovered, while λ = 1 corresponds to the fully implicit case. If the grid is uniform, the *x*-derivative is approximated by the finite difference relation developed in Sect. 5.2.1.

$$\left(\frac{\partial \phi}{\partial x}\right)_{1+\lambda,\mathrm{j}} = \frac{\phi_{\mathrm{i+1,j}} - \phi_{\mathrm{i,j}}}{\Delta x} + \left(\lambda - \frac{1}{2}\right) 0(\Delta x) + 0(\Delta x^2)$$

and the *y*-derivative is replaced by the weighted mean

$$\left(\frac{\partial^2 \phi}{\partial \mathrm{y}^2}\right)_{\mathrm{i}+\lambda,\mathrm{j}} = \lambda \left(\frac{\partial^2 \phi}{\partial \mathrm{y}^2}\right)_{\mathrm{i+1,j}} + (1-\lambda) \left(\frac{\partial^2 \phi}{\partial \mathrm{y}^2}\right)_{\mathrm{i,j}}$$

Each second-order derivative is then replaced by the usual three-point centred finite difference relation:

$$\left(\frac{\partial^2 \phi}{\partial \mathrm{y}^2}\right)_{\mathrm{i,j}} = \frac{\phi_{\mathrm{i,j+1}} - 2\phi_{\mathrm{i,j}} + \phi_{\mathrm{i,j-1}}}{\Delta \mathrm{y}^2} + \left(\Delta \mathrm{y}^2\right)$$

Substituting Eqs. (8.41, 8.42, 8.43) into equation (8.40), a linear difference equation is obtained

$$\begin{aligned}\frac{\phi_{\mathrm{i+1,j}} - \phi_{\mathrm{i,j}}}{\Delta x} = \frac{a}{\Delta \mathrm{y}^2} \big[& \lambda(\phi_{\mathrm{i+1,j+1}} - 2\phi_{\mathrm{i+1,j}} + \phi_{\mathrm{i+1,j-1}}) \\ & + (1-\lambda)\left(\phi_{\mathrm{i,j+1}} - 2\phi_{\mathrm{i,j}} + \phi_{\mathrm{i,j-1}}\right)\big]\end{aligned}$$

which can be recast in the usual tridiagonal form

$$\left(-\lambda \frac{a\Delta x}{\Delta \mathrm{y}^2}\right)\phi_{\mathrm{i+1,j-1}} + \left(1 - 2\lambda \frac{a\Delta x}{\Delta \mathrm{y}^2}\right)\phi_{\mathrm{i+1,j}} + \left(\lambda \frac{a\Delta x}{\Delta \mathrm{y}^2}\right)\phi_{\mathrm{i+1,j+1}} = D_{\mathrm{j}}$$

with *D*j a function of φ computed at station *i*.

To perform the von Neumann stability analysis it is useful to express the numerical solution as a Fourier series, and then verify that none of the harmonics is amplified with respect to the evolution coordinate *x*. This stability analysis is described in detail in Sect. 4.4; Part I, and is repeated here as a reminder. Hence putting

$$\phi_{\mathrm{i,j}} = \rho^{\mathrm{i}} e^{\mathrm{I}\omega(\mathrm{j}\Delta y)}$$

where *I* in the exponent is the unit complex number, and ρi is the amplification factor at level *i*, and then substituting inside Eq. (8.45), actualizing the indices in Eq. (8.46), we have

$$G = \frac{\rho^{\mathrm{i+1}}}{\rho^{\mathrm{i}}} = \frac{1 + 2a(1-\lambda)\frac{\Delta x}{\Delta y^2}[\cos(\omega\Delta \mathrm{y}) - 1]}{1 + 2a\lambda\frac{\Delta x}{\Delta y^2}[\cos(\omega\Delta \mathrm{y}) - 1]}$$

To have stability, |*G*| ≤ 1 for all harmonics ωΔ*y*; this inequality together with Eq. (8.47), leads to the following stability condition for 0 ≤ λ < ½

$$C \le \frac{1}{2(1-2\lambda)}$$

where *C* = *a*Δ*x*/Δ*y*2. For 1/2 ≤ λ ≤ 1 no stability restriction is imposed on *C*. Hence the scheme presented is unconditionally stable for values of λ equal or larger than 1/2. In the case of the explicit scheme (λ = 0), there is a strong limitation to Δ*x* if Δ*y* is chosen rather small for accuracy requirements.

The consistency of the scheme can easily be verified expanding in Taylor series all other terms of Eq. (8.45) about the point (*i*+λ, *j*). The discretization error can be proved to be of 0(Δ*x*, Δ*y*2) if λ is not equal to 1 (Ref. [14]). The scheme is therefore second order accurate with respect to *y* and first-order with respect to *x*. To obtain second order accuracy with respect to *x*, λ should be taken equal to 1/2 (Crank–Nicholson scheme), or slightly different to 1/2 (e.g. = 1/2 + 0(Δ*x*)). However, for practical, non-linear problems it is often necessary to increase λ in order to avoid non-linear instabilities. For instance, the full implicit scheme is often very stable, but leads to a worse accuracy.

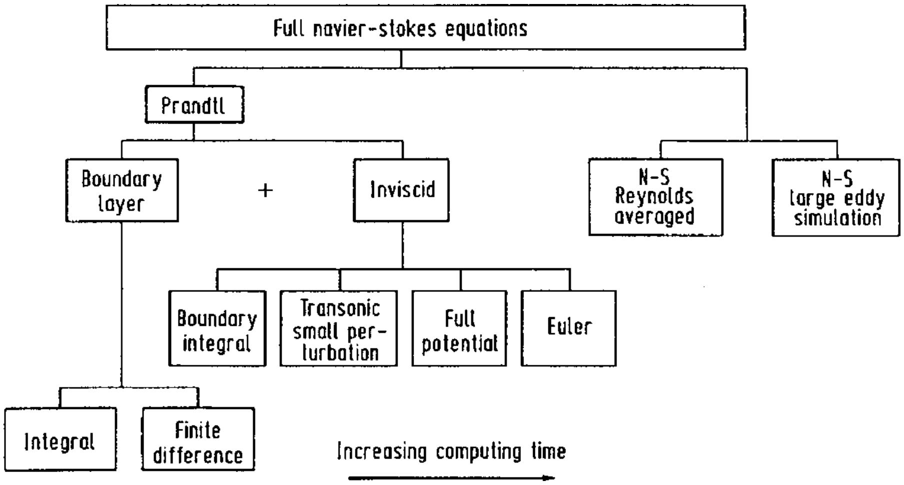
Equation (8.40) is a linear partial differential equation employed as a model to demonstrate the widely used generalized implicit Crank–Nicholson solution code. Now this will be applied to the boundary layer Eqs. (8.31), (8.32) and (8.33) of Sect. 8.4.

3.2 Implicit Time Dependent Methods for Inviscid and Viscous Compressible flows

The compressible Euler and Navier-Stokes equations represent the most sophisticated models of single-phase flows of single-component Newtonian fluids. As such, they allow the analysis of complex Inviscid and viscous flow phenomena including rotational flows caused by curved shock waves or viscous/Inviscid interactions leading to flow separation. As a counterpart, the numerical techniques required to solve these equations are also the most sophisticated and the numerical effort needed to obtain them is also the greatest. This is schematically represented in Fig. 9.1 taken from Green's [18] review of the state- of-the-art in numerical methods in aeronautical fluid dynamics.

The difficulties of solving complex steady compressible flows were already pointed out in the first part of this volume, in which the blunt body problem was taken as an illustrative example. It was shown that the crux of the difficulty lies in the mixed character of the flow, involving regions governed by "elliptic" equations and others governed by "hyperbolic" equations. Finally, the solution to the problem was found by solving the time dependent equations using a time marching method, taking advantage of the uniform nature of the unsteady equations with respect to time, independently of the subsonic or supersonic character of the flow.1 following that breakthrough, many methods were developed to integrate the unsteady Euler or Navier-Stokes equations. These methods can be classified in two main categories: explicit and implicit methods (Part I, Sect. 5.3).

Historically, explicit methods were developed earlier, because of their greater simplicity. Several examples were given in Part I, Chap. 7. The major limitation of these methods is their stability characteristics which impose an upper bound on the usable integration time step. In recent years, implicit methods have been developed to overcome this limitation and have proved more efficient than the former explicit method, which justifies their study.



Hierarchy of computational fluid dynamics after Discretization

In Sect. 9.2, we shall examine solution techniques for simpler flows and explain why these techniques fail for the solution of the steady compressible Euler/ Navier-Stokes equations. In Sect. 9.3, stability properties of numerical integration techniques will be studied in detail first for ordinary differential equations, then for

partial differential equations. In Sect. 9.4, it will be shown how an implicit method can be constructed to solve partial differential equations such as the Euler or Navier-Stokes equations. It will be seen that this can be subdivided into three steps, the choice of an explicit discretization scheme, the choice of an implicit operator and finally the choice of a solution strategy, which will be discussed in turn. For the first step, the issue of numerical dissipation will turn out to be crucial, and this concept will be discussed in detail. As in Part I, only the finite difference method is considered as the space discretization technique, but, as will be mentioned in the lecture,most of the concepts discussed and of the basic methods described apply equally to finite volume discretizations (especially on structured meshes) and some to finite element discretizations.

The content of these notes will remain rather basic except in a few instances, in accordance with the objectives of this book. In particular, no individual scheme will be examined in great detail. For additional information, the reader is referred to the very comprehensive survey of CFD methods by C. Hirsch [22, 23] and, finally, to the original literature.

3.3 Concept of Numerical Dissipation

Definition of Numerical Dissipation

As mentioned above, the question of numerical dissipation arises for advection dominated problems. Numerical dissipation is therefore defined in reference to the advection (wave) equation:

$$\frac{\partial u}{\partial t} + c\frac{\partial u}{\partial x} = 0$$

This equation describes the transport of the quantity *u* with speed *c*. Its general solution is *u* = *f* (*x*−*ct*). A particular solution is the periodical solution

$$u = e^{ik(x-ct)} = e^{ikx}e^{-i\omega t} \qquad \text{with } \omega = kc$$

which represents the unattenuated propagation of a wave of length 2π/*k* with speed c.

Let us compute the amplification ion factor *u*(*x*, *t*+Δ*t*)/*u*(*x*, *t*) for the exact solution. We find

$$\frac{u(x,t+\Delta t)}{u(x,t)} = e^{-i\omega\Delta t} = e^{-ikc\Delta t} = e^{-i\eta\nu}$$

with

$$\begin{aligned} \nu &= \frac{c\Delta t}{\Delta x} && \text{CFL number} \\ \eta &= k\Delta x && \text{dimensionless wave number} \end{aligned}$$

A numerical solution will yield

$$\frac{u_i^{n+1}}{u_i^n} = g(\eta,\nu)$$

When one wishes to accurately follow a true unsteady phenomenon, one obviously desires to have g(η,ν) as close as possible to $e^{-i\eta\nu}$. For stability, one must have $|g(\eta,\nu)| \leq 1$ for all η. The difference between $|g(\eta,\nu)|$ and 1 is called *dissipation* or else *dissipative error*, and the difference between arg(g(η,ν)) and −ην is called *dispersion* or *dispersive error*.

3.4 Stability Properties of Explicit and Implicit Methods

Since the outcome of the competition between explicit and implicit methods is governed by their respective stability properties, a closer look must be given to this issue. First, we observe that the space-discretization of a time-dependent partial differential equation produces a system of ordinary differential equations. Consider for example the diffusion equation

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} \qquad \text{-------------------- 1}$$

After discretization in space using central finite differences, we obtain the following system of ordinary differential equations

$$\frac{du_i}{dt} = \frac{1}{\Delta x^2}(u_{i-1} - 2u_i + u_{i+1}) \qquad \text{or} \qquad \frac{dU}{dt} = AU + F(t) \qquad \text{------------ 2}$$

With

$$U = \begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_{n-1} \\ u_n \end{pmatrix} \qquad A = \frac{1}{\Delta x^2}\begin{pmatrix} \text{B.C.} & & & & \\ 1 & -2 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & -2 & 1 \\ & & & & \text{B.C.} \end{pmatrix} \qquad \text{-------------3}$$

Therefore, the analysis of the stability of a time stepping scheme for solving the PDE reduces to the analysis of the stability of a time stepping scheme for solving a system of ODEs. Furthermore, when we consider a periodic solution in space,

$$\text{i.e. } u = u(t)e^{ik_m x},$$

the system of ODEs reduces to a single ODE. Indeed, inserting the periodic solution hypothesis in (2) and realizing that

$$u_{i+1} = u(t)e^{ik_m x_{i+1}} = u(t)e^{ik_m(x_i+\Delta x)} = u(t)e^{ik_m x_i}e^{ik_m\Delta x} = u_i e^{ik_m\Delta x}$$

and similarly

$$u_{i-1} = u_i e^{-ik_m\Delta x}$$

we obtain

$$\frac{du_i}{dt} = \frac{e^{ik_m\Delta x} - 2 + e^{-ik_m\Delta x}}{\Delta x^2}u_i = \underbrace{-\frac{4}{\Delta x^2}\sin^2\left(\frac{k_m\Delta x}{2}\right)}_{q}u_i$$

i.e. an ODE whose coefficient *q* depends on the reduced wavenumber *km*Δ*x*, the locus of *q* (in the complex plane) being called the Fourier footprint of the discretized equation. The stability analysis can then be reduced to the stability analysis for a single ordinary differential equations *du*/*dt* = *qu*, where *q* is a complex coefficient.

Definition:

Stability of the numerical integration of an ordinary differential equation is usually defined by the following statement. A method is said to be stable (weakly-stable) if the numerical solution remains bounded when the number of steps *n* goes to infinity and the time step size Δ*t* goes to zero with the product *n*Δ*t* remaining constant.

Stability Properties:

Weak Instability

The results of the calculation are displayed in Fig. 1. One notices that the perturbation on *u*1 gives rise to amplifying oscillations. In fact, as small as the initial perturbation may be - and there will always be one because of round off errors – it will eventually lead to an explosion of the numerical solution. This phenomenon is clearly inacceptable. It is named *weak instability*.

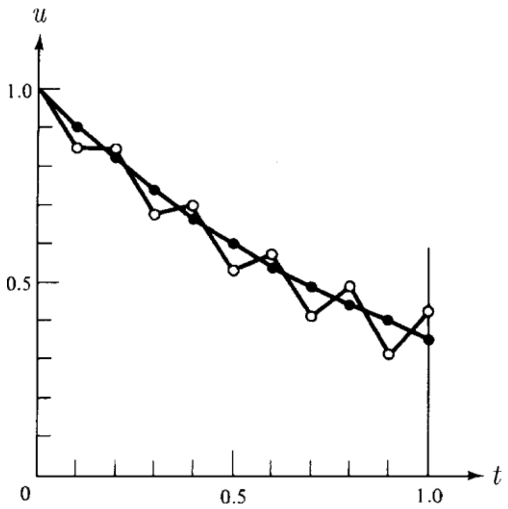


Fig. 1 Numerical solution of *du dt* = −*u*; *u*(*o*) = 1 with the 2-step explicit midpoint method

Region of (absolute) stability

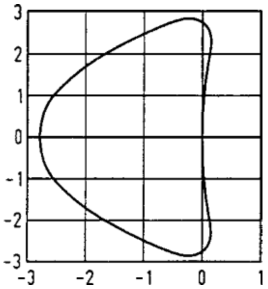
The concept of region of (absolute) stability was introduced by Dahlquist .The region of (absolute) stability of a numerical algorithm for integrating an O.D.E. is the set of values of *z* = *q*Δ*t* (*q* = complex parameter of the test equation *du/dt* = *qu*) such that the sequence *un* of numerical values remains bounded as *n*→∞. As the definition of stability required that the sequence *un* remain bounded for *n*→∞, Δ*t* → 0, this is equivalent to stating that the origin lies in the region of (absolute) stability [Δ*t*→0 implies *z* = *q*Δ*t*→0].

Stiff Problems

Problems where there is such a coexistence of phenomena with very disparate time scales are called stiff problems. Unfortunately they are not uncommon in many fields of engineering and in particular in fluid mechanics. For those problems, it would be desirable to have at our disposal schemes such that a physically stable problem would lead to a bounded solution irrespective of the value of the time step Δt. That property is called absolute stability or A – Stability.

Absolute Stability

Absolute stability was defined as a property by which the numerical solution of a physically stable problem would be bounded, irrespective of the time step. Let us translate this in mathematical terms. Test problems of the type *du/dt* = *qu* are stable if *Re*(*q*) ≤ 0. Therefore, the set of values of *q*Δ*t*, corresponding to stable problems is the left half plane. Absolute stability is thus equivalent to requiring that the region of stability include the left-half plane.



Region of stability of the Runge-Kutta method

Absolute stability ≡ the region of stability includes the left-half plane.

3.5 Conservative Upwind Discretization for Hyperbolic Systems

It will be shown how conservative upwind discretizations of hyperbolic systems such as the Euler equations can be constructed.

When systems of conservation laws like the Euler equations are considered, the extension of upwind schemes poses a problem, in the sense that wave speeds of both signs can be simultaneously present. Indeed, the characteristic speeds associated with the unsteady 1D Euler equations

$$\frac{\partial \mathbf{U}}{\partial t}+\frac{\partial \mathbf{F}}{\partial x}=\frac{\partial \mathbf{U}}{\partial t}+\mathbf{A}\frac{\partial \mathbf{U}}{\partial x}=0$$

are u, $u+a$ and $u-a$, so that speeds of both signs exist when the flow is subsonic. It is then impossible to use a biased discretization of the whole flux vector **F** since this would lead to a downwind discretization for one of the waves. If one considers the quasi-linear form of the equations, then one can decompose the original system in characteristic equations and upwind each equation according to the corresponding wave speed sign (Courant-Isaacson-Rees scheme [8]) but this approach does not satisfy the conservation property which is crucial for the correct treatment of discontinuities (Part I, Sect. 2.9). This is the main reason why schemes based on a central space discretization such as the Lax-Wendroff scheme and schemes involving artificial diffusion have been so popular in the sixties and seventies. Indeed, these schemes are indifferent to wave speed sign and therefore extend readily to systems:

$$\text{artificial diffusion} \qquad h_{i+1/2}=\frac{\mathbf{F}_i+\mathbf{F}_{i+1}}{2}-d(\mathbf{U}_{i+1}-\mathbf{U}_i)$$

Lax-Wendroff $$h_{i+1/2} = \frac{\mathbf{F}_i + \mathbf{F}_{i+1}}{2} - \frac{\mathbf{A}_{i+1/2}\Delta t}{2\Delta x}(\mathbf{F}_{i+1} - \mathbf{F}_i)$$

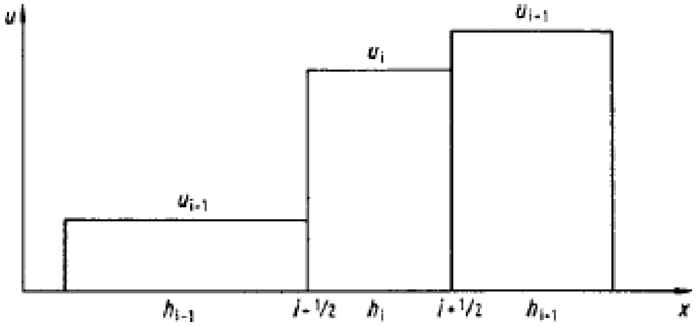
The early eighties have seen the development of conservative upwind schemes, which have since become extremely popular, because of their crisp resolution of discontinuities and their superior ability in following moving shock waves. The remainder of this section will therefore be devoted to a brief presentation of the two major families of conservative upwind schemes.

Flux Difference Splitting (FDS) Schemes — Approximate Riemann Solvers

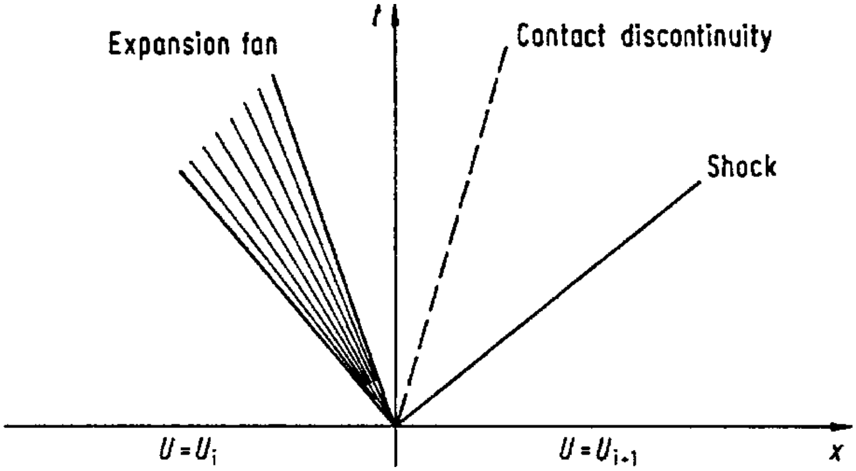
The starting point of Flux Difference Splitting scheme is the scheme developed in the late fifties by the Russian mathematician Godunov [17] for the unsteady 1D Euler equations. This scheme is based on the integral form of the equations.9 The integral form of the unsteady 1D Euler equations (9.36) is

$$\frac{d}{dt}\int_a^b \mathbf{U}dx + \mathbf{F}(\mathbf{U}_b) - \mathbf{F}(\mathbf{U}_a) = 0$$

For the numerical solution of the problem, the domain of interest is divided up into intervals (cells in the finite volume terminology) and the unknowns of the numerical solution **U***i* are the *average* flow quantities over the corresponding interval (see Fig. 9.13) rather than point values as in the finite difference method. The boundaries of interval *i* are noted *i*±1/2. As illustrated in the figure, the intervals need not be of constant length (*hi*−1 _ *hi* _ *hi*+1). The first step in Godunov's method consists in *reconstructing* a piecewise continuous distribution of the flow variables from the cell averages. The simplest choice is a piecewise constant reconstruction as illustrated in the figure.10 At the interval interfaces, the flow variable distributions are thus discontinuous. Now, there exists an *exact* solution of the 1D Euler equations for initial data consisting of two constant states separated by a discontinuity—this problem is known in the literature as the Riemann problem, and applies in particular to the flow in a shock tube. The solution consists of elementary waves (shock wave, contact discontinuity, expansion wave) originating from the interface, as illustrated in Fig. 9.14 for the shock tube problem. An interesting property of the solution is that flow variables are constant along straight lines in *x*−*t* space (which implies that the solution is self-similar). In particular, it is constant in time at the location of the interface. As long as the two solutions at each interface of an interval do not interact (which imposes an upper bound on the time step), it is thus possible to compute the *exact* solution at the new time level from the initial piecewise constant data. This constitutes the second step in Godunov's method, called the *evolution* step. From



Finite volume representation



Schematic representation of the solution of the Riemann problem

the exact solution at the new time level, it is then possible to compute the new cell averages in order to restart the process. This constitutes the third step of the method, called *projection* step.

Actually, it is possible to compute directly the cell averages at the new time level without computing the details of the solution. Indeed, integrating in time between *tn* and *tn*+1 = *tn*+Δ*t* the integral form of the equations applied to interval *i*, one finds

$$\int_{i-1/2}^{i+1/2} \mathbf{U}^{n+1} dx - \int_{i-1/2}^{i+1/2} \mathbf{U}^{n} dx + \int_{t^n}^{t^{n+1}} \mathbf{F}_{i+1/2} dt - \int_{t^n}^{t^{n+1}} \mathbf{F}_{i-1/2} dt = 0$$

This expression simplifies greatly since $\mathbf{U}^n = \mathbf{U}_i^n$ = const over the interval and $\mathbf{F}_{i\pm1/2}$ are constant over the time step. In addition, $\mathbf{U}_i^{n+1}$ being the average over the interval of the solution at the new time level, $\int_{i-1/2}^{i+1/2} \mathbf{U}^{n+1} dx = \mathbf{U}_i^{n+1} h_i$. The final result is thus

$$(\mathbf{U}_i^{n+1} - \mathbf{U}_i^n)h_i + (\mathbf{F}_{i+1/2} - \mathbf{F}_{i-1/2})\Delta t = 0$$

or dividing through by *hi*Δ*t*,

$$\frac{\mathbf{U}_i^{n+1} - \mathbf{U}_i^n}{\Delta t} + \frac{\mathbf{F}_{i+1/2} - \mathbf{F}_{i-1/2}}{h_i} = 0$$

from which we deduce that Godunov’s scheme is a conservative discretization of the 1D Euler equations with the numerical flux function

$$h_{i+1/2} = \mathbf{F}(\mathbf{U}_{\text{exact}}(x_{i+1/2}, t))$$

combined with forward Euler time stepping. That this is an upwind discretization clearly shows up by applying it to the linear advection equation. Since the exact solution of the linear advection equation is the initial solution moving with speed *c*, it results that (for *c* > 0)

$$h_{i+1/2} = c\,u_i \qquad \text{and} \qquad h_{i-1/2} = c\,u_{i-1}$$

and one recovers the first-order upwind discretization.

The essential drawback of Godunov’s scheme is that the computation of **U** exact (*xi*+1/2, *t*) requires the solution of a non-linear algebraic problem, i.e. it is computationally expensive. Now, as most of the information contained in the exact solution is lost by the averaging process, Roe [34] suggested to replace the exact Riemann problem by a linearized problem

$$\frac{\partial \mathbf{U}}{\partial t} + \tilde{\mathbf{A}}_{i+1/2}\frac{\partial \mathbf{U}}{\partial x} = 0$$

where $\tilde{\mathbf{A}}_{i+1/2}$ is a function of $\mathbf{U}_i$ and $\mathbf{U}_{i+1}$, chosen to satisfy certain properties:

1. $\tilde{\mathbf{A}}(\mathbf{U}, \mathbf{U}) = \mathbf{A}(\mathbf{U})$;
2. $\tilde{\mathbf{A}}_{i+1/2}$ has a complete set of real eigenvalues for any pair of $\mathbf{U}_i$, $\mathbf{U}_{i+1}$;
3. $\tilde{\mathbf{A}}_{i+1/2}(\mathbf{U}_{i+1} - \mathbf{U}_i) = \mathbf{F}_{i+1} - \mathbf{F}_i$.

The first condition is required for consistency, the second ensures that the linearized problem has a solution, and the third condition is a sufficient condition for the scheme to be conservative. It also has the nice additional property that the solution of the linearized problem is identical to the solution of the exact problem when a single wave is present.

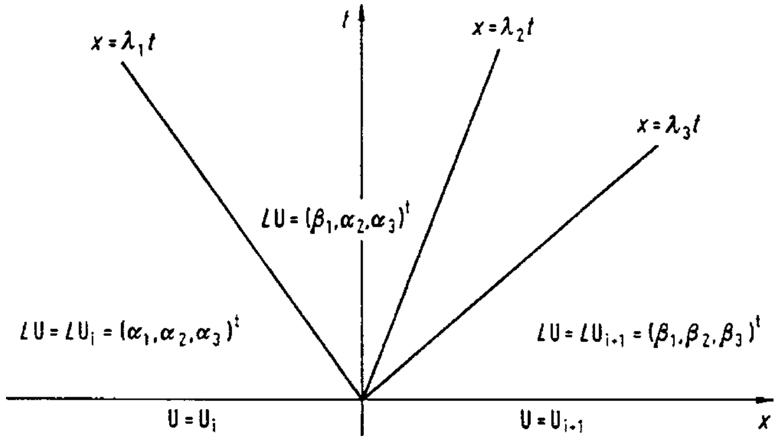
Now, the solution of the linearized problem is found quite easily by the theory of characteristics. Multiplying the linearized equation by the matrix *L* of left eigenvectors of ˜**A***i*+1/2, one obtains

$$L\frac{\partial \mathbf{U}}{\partial t} + L\tilde{\mathbf{A}}_{i+1/2}\frac{\partial \mathbf{U}}{\partial x} = L\frac{\partial \mathbf{U}}{\partial t} + \Lambda L\frac{\partial U}{\partial x} = 0$$

Where Λ is the (diagonal) matrix of eigenvalues of ˜**A***i*+1/2. These are decoupled linear advection equations for the characteristic variables, components of the vector *L***U**. For the 1D Euler equations, there are three components. Noting

$$L\mathbf{U}_i = (\alpha_1,\alpha_2,\alpha_3)^t; \qquad L\mathbf{U}_{i+1} = (\beta_1,\beta_2,\beta_3)^t$$

and arranging the eigenvalues in increasing order, the solution of the linear problem is schematically shown in Fig. 9.15 (in terms of characteristic variables) and for the case of the figure (λ1 < 0, λ2, λ3 > 0),


$x = \lambda_1 t$
t
$x = \lambda_2 t$
$x = \lambda_3 t$
$LU = (\beta_1,\alpha_2,\alpha_3)^t$
$LU = LU_i = (\alpha_1,\alpha_2,\alpha_3)^t$
$LU = LU_{i+1} = (\beta_1,\beta_2,\beta_3)^t$
$U = U_i$
$U = U_{i+1}$
x

Solution of the linearized Riemann problem

$$L\mathbf{U}_{i+1/2} = (\beta_1, \alpha_2, \alpha_3)^t \qquad \rightarrow \mathbf{U}_{i+1/2} = R(\beta_1, \alpha_2, \alpha_3)^t$$

where R is the matrix of right eigenvectors of $\tilde{\mathbf{A}}_{i+1/2}$ (inverse of L). The corresponding flux is thus (the flux being linear)

$$\begin{aligned}
\mathbf{F}_{i+1/2} &= \mathbf{F}_i + \tilde{\mathbf{A}}_{i+1/2}(\mathbf{U}_{i+1/2} - \mathbf{U}_i) = \mathbf{F}_i + \tilde{\mathbf{A}}_{i+1/2}R(\beta_1 - \alpha_1, 0, 0)^t \\
&= \mathbf{F}_i + R\Lambda(\beta_1 - \alpha_1, 0, 0)^t = \mathbf{F}_i + R\Lambda^- L(\mathbf{U}_{i+1} - \mathbf{U}_i) \\
&= \mathbf{F}_i + \tilde{\mathbf{A}}^-_{i+1/2}(\mathbf{U}_{i+1} - \mathbf{U}_i)
\end{aligned}$$

or

$$\begin{aligned}
\mathbf{F}_{i+1/2} &= \mathbf{F}_{i+1} - \tilde{\mathbf{A}}_{i+1/2}(\mathbf{U}_{i+1} - \mathbf{U}_{i+1/2}) = \mathbf{F}_{i+1} - \tilde{\mathbf{A}}_{i+1/2}R(0, \beta_2 - \alpha_2, \beta_3 - \alpha_3)^t \\
&= \mathbf{F}_{i+1} - R\Lambda(0, \beta_2 - \alpha_2, \beta_3 - \alpha_3)^t = \mathbf{F}_{i+1} - R\Lambda^+ L(\mathbf{U}_{i+1} - \mathbf{U}_i) \\
&= \mathbf{F}_{i+1} - \tilde{\mathbf{A}}^+_{i+1/2}(\mathbf{U}_{i+1} - \mathbf{U}_i)
\end{aligned}$$

relations from which it appears that the flux difference **F***i*+1 −**F***i* has been split into a positive and a negative part to calculate **F***i*+1/2, whence the name Flux Difference Splitting. By this splitting of the flux difference, the scheme automatically adapts the difference scheme to the local flow quantities. It is thus a solution-adaptive differencing scheme as alluded to in the introduction.

Averaging the two previous expressions, the following (third) form of Roe's scheme is obtained:

$$\mathbf{F}_{i+1/2} = \frac{\mathbf{F}_i + \mathbf{F}_{i+1}}{2} - \frac{1}{2}|\tilde{\mathbf{A}}_{i+1/2}|(\mathbf{U}_{i+1} - \mathbf{U}_i)$$

where $|\tilde{\mathbf{A}}_{i+1/2}| = \tilde{\mathbf{A}}^+_{i+1/2} - \tilde{\mathbf{A}}^-_{i+1/2}$.

Now, this has exactly the same form as the artificial diffusion flux formula (9.37) except that the diffusion coefficient is replaced here by a diffusion *matrix*.

The Flux Difference Splitting approach pioneered by Roe has met with a considerable success. Several schemes of this type, also called Approximate Riemann solvers, were developed since the beginning of the eighties [13, 15, 33, 38], among which the most popular is certainly Osher's scheme [33].

Flux Vector Splitting (FVS) Schemes

The idea of flux vector splitting was introduced in computational fluid dynamics by Steger and Warming [40]. The idea had been previously proposed in astrophysics by Sanders and Prendergast [36] but was rediscovered independently by Steger and Warming. The starting point of Steger and Warming's scheme is the observation that the compressible inviscid fluxes are homogeneous functions of degree 1 in the conservative variables. Consequently, by a theorem due to Euler,

$$\mathbf{F}(\mathbf{U}) = \frac{\partial \mathbf{F}(\mathbf{U})}{\partial \mathbf{U}}\mathbf{U} = \mathbf{A}(\mathbf{U})\mathbf{U}$$

Now, the flux Jacobian matrix A is fully diagonalizable and it is possible to split it between its positive and negative parts (see previous paragraph)

$$\mathbf{A} = R\Lambda L = R(\Lambda^{+} + \Lambda^{-})L = \underbrace{R\Lambda^{+}L}_{\mathbf{A}^{+}} + \underbrace{R\Lambda^{-}L}_{\mathbf{A}^{-}}$$

to which correspond the split fluxes

$$\mathbf{F}^{+} = \mathbf{A}^{+}\mathbf{U} \qquad\qquad \mathbf{F}^{-} = \mathbf{A}^{-}\mathbf{U}$$

Now, the split fluxes **F**± being associated with positive (respectively negative) eigenvalues only, it is possible to use upwind difference formulas to discretize the corresponding flux derivatives.

The Steger and Warming flux vector splitting suffers from a lack of continuity at those points where an eigenvalue of **A** vanishes (stagnation and sonic points).

To remedy this problem, van Leer developed an alternative, continuous, flux vector splitting [45], which is no longer based on the homogeneity property of the inviscid flux vectors. The basic requirements are

- ✓ the split fluxes sum up to the whole flux: **F**+ +**F**−= **F**;
- ✓ the split fluxes Jacobians have positive (respectively negative) eigenvalues only;
- ✓ **F**$^{-}$= 0 for supersonic flow (respectively **F**+ = 0 for supersonic flow with negative velocity).

Van Leer imposed a few additional requirements in particular to ensure the crisp resolution of discontinuities.

The flux vector splitting approach and van Leer's scheme in particular have become extremely popular in the CFD community [37, 43], but it was soon realized that flux vector splitting schemes are excessively dissipative at contact discontinuities (boundary and shear layers) [46]. To avoid this while keeping the robustness of flux vector splitting schemes, an improved flux vector splitting scheme was recently developed by Liou and Steffen [30](AUSM scheme). Jameson's CUSP scheme [25], although formulated in the artificial diffusion formalism, appears essentially equivalent to this latter scheme. Finally, Coquel and Liou [6] have proposed a procedure to construct hybrid flux vector/flux difference splitting schemes to combine the robustness of the flux vector splitting schemes with respect to strong shock and expansion waves and the accuracy of flux difference splitting schemes with respect to contact discontinuities. They examine in particular the van Leer/Osher hybrid, which provides results of comparable accuracy as Osher's scheme for viscous flow calculations at a cost only slighly superior to van Leer's FVS scheme.

Further Advantages of Upwind Differencing:

Let us illustrate the effects of numerical dissipation by considering a Couple of examples. Let us suppose we are looking for a numerical solution of the advection equation given below

$$\frac{\partial u}{\partial t} + c\frac{\partial u}{\partial x} = 0$$

Let us first select the space discretization scheme. We consider two possibilities: the central finite difference and the backward (upwind) finite difference formulas.

BACKWARD (UPWIND)

$$\frac{du_i}{dt} + c\frac{u_i - u_{i-1}}{\Delta x} = 0$$

CENTRAL

$$\frac{du_i}{dt} + c\frac{u_{i+1} - u_{i-1}}{2\Delta x} = 0$$

In order to select the time-integration scheme, let us first calculate the Fourier footprints of the discretized equations. Introducing the periodic solution hypothesis

$$(u = U(t)e^{Ikx} \to u^n_{i\pm1} = u^n_i e^{\pm I\eta}),$$

We get

$$\frac{du_i}{dt} = \underbrace{-c\frac{1-e^{-I\eta}}{\Delta x}}_{q} u_i$$

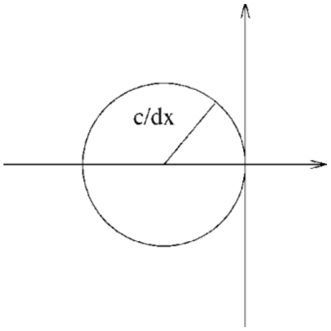
$$\begin{aligned}\frac{du_i}{dt} &= -c\frac{e^{I\eta}-e^{-I\eta}}{2\Delta x}u_i \\ &= \underbrace{-I\frac{c\sin\eta}{\Delta x}}_{q}u_i\end{aligned}$$

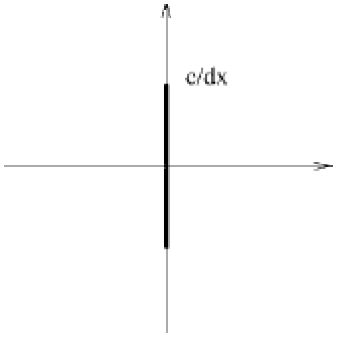
The discretized equation reduce to the model equation

$$\frac{du}{dt} = qu \qquad u(0) = 1$$

Where the q coefficient depends on the reduced wave number η .

The locus of q is the Fourier footprint of the discretized equation. They are shown below for the two discretization schemes





Now, the time-integration scheme should be selected so that $q\Delta t$ can lie within the region of stability. By comparing the respective loci of q with the region of stabilities of some of the schemes examined previously ,it appears quite clearly that the forward Euler scheme cannot be used together with the central space discretization, but the mid-point method, on the contrary, can, and the opposite conclusion applies to the backward finite difference discretization. Applying the mid-point method and the forward Euler method to the central and backward space discretization's respectively, the fully discrete schemes and their truncation errors are respectively

FIRST-ORDER UPWIND-FORWARD EULER

$$\frac{u_i^{n+1}-u_i^n}{\Delta t} + c\frac{u_i-u_{i-1}}{\Delta x} = 0$$

$$\text{TE} = O(\Delta x, \Delta t)$$

LEAPFROG (CENTRAL SPACE-MID POINT)

$$\frac{u_i^{n+1}-u_i^{n-1}}{2\Delta t} + c\frac{u_{i+1}-u_{i-1}}{2\Delta x} = 0$$

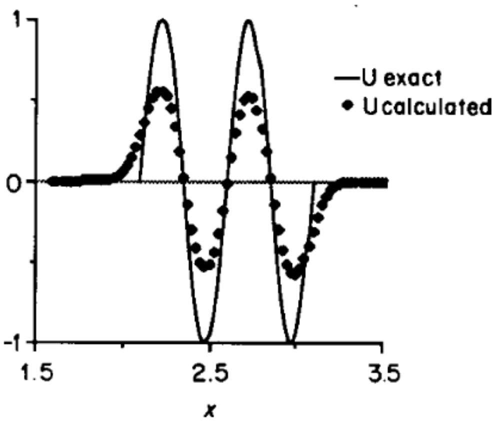
$$\text{TE} = O(\Delta x^2, \Delta t^2)$$

and, being central in both space and time, the leapfrog method is seen to be of superior accuracy. Let us now compute the amplification factors of both schemes.

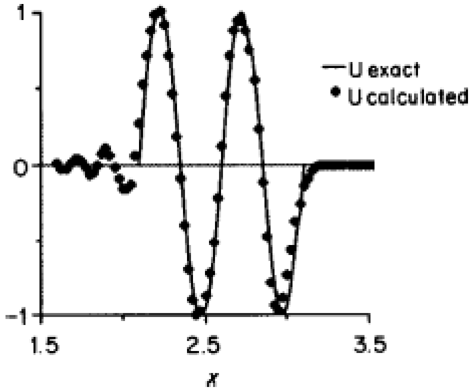
$$
\begin{aligned}
g &= 1 + q\Delta t \quad \rightarrow \\
g &= 1 - \frac{c\Delta t}{\Delta x}(1 - e^{-I\eta}) \\
&= 1 - \nu(1 - e^{-I\eta}) \\
&= 1 - \nu(1 - \cos\eta) - I\nu\sin\eta \\
|g|^2 &= 1 - 2\nu(1 - \cos\eta) + \\
&\qquad \nu^2[(1 - \cos\eta)^2 + \sin^2\eta] \\
&= 1 - 2\nu(1 - \cos\eta) + 2\nu^2(1 - \cos\eta) \\
&= 1 - 2\nu(1 - \nu)(1 - \cos\eta) < 1 \text{ for } 0 < \nu < 1
\end{aligned}
$$

$$
\begin{aligned}
g &= q\Delta t \pm \sqrt{1 + (q\Delta t)^2} \quad \rightarrow \\
g &= -I\frac{c\Delta t}{\Delta x}\sin\eta \pm \sqrt{1 - (\frac{c\Delta t}{\Delta x}\sin\eta)^2} \\
&= -I\nu\sin\eta \pm \sqrt{1 - \nu^2\sin^2\eta} \\
&= Ie^{\pm I(\frac{\pi}{2}+\alpha)} \text{ with } \sin\alpha = \nu\sin\eta \\
|g|^2 &= 1
\end{aligned}
$$

And we observe that the first order upwind-forward Euler method is dissipative whereas the leapfrog method is not. It thus seems that the leapfrog method is in all ways (truncation error, dissipative properties) superior to the first order upwind forward Euler method. Let us check this conclusion by looking at numerical examples. We first consider the advection of a wave packet of period 0.5 ($k = 4\pi$) on a mesh of size $\Delta x = 1/40$ (hence $\eta = k\Delta x = \pi/10$), using a time step such that the CFL number $\nu = 0.8$. The numerical results obtained by both methods shown below confirm the conclusions of the analysis: the dissipative properties of the first order upwind-forward Euler method result in a serious reduction of the wave amplitude, whereas in contrast, the leapfrog solution almost perfectly agrees with the exact solution. We can however observe some trailing oscillations in the leapfrog solution.



Forward time upwind scheme



Leapfrog scheme

UNIT IV

FINITE ELEMENT TECHNIQUES

Overview of Finite Element Techniques in Computational Fluid Dynamics- Strong and Weak Formulations of a Boundary Value Problem

4.1 Overview of Finite Element Techniques in Computational Fluid Dynamics:

The finite element method (FEM) is a numerical technique for solving partial differential equations (PDE's). Its first essential characteristic is that the continuum field, or domain, is subdivided into cells, called elements, which form a grid. The elements (in 2D) have a triangular or a quadrilateral form and can be rectilinear or curved. The grid itself need not be structured. With unstructured grids and curved cells, complex geometries can be handled with ease. This important advantage of the method is not shared by the finite difference method (FDM) which needs a structured grid, which, however, can be curved. The finite volume method (FVM), on the other hand, has the same geometric flexibility as the FEM.

The second essential characteristic of the FEM is that the solution of the discrete problem is assumed a priori to have a prescribed form. The solution has to belong to a function space, which is built by varying function values in a given way, for instance linearly or quadratically, between values in nodal points. The nodal points, or nodes, are typical points of the elements such as vertices, mid-side points, mid element points, etc. Due to this choice, the representation of the solution is strongly linked to the geometric representation of the domain. This link is, for instance, not as strong in the FVM.

The third essential characteristic is that a FEM does not look for the solution of the PDE itself, but looks for a solution of an integral form of the PDE. The most general integral form is obtained from a weighted residual formulation. By this formulation the method acquires the ability to naturally incorporate differential type boundary conditions and allows easily the construction of higher order accurate methods. The ease in obtaining higher order accuracy and the ease of implementation of boundary conditions form a second important advantage of the FEM. With respect to accuracy, the FEM is superior to the FVM, where higher order accurate formulations are quite complicated.

The combination of the representation of the solution in a given function space, with the integral formulation treating rigorously the boundary conditions, gives to the method an extremely strong and rigorous mathematical foundation.

A final essential characteristic of the FEM is the modular way in which the discretization is obtained. The discrete equations are constructed from contributions on the element level which afterwards are assembled.

Historically, the finite element method originates from the field of structural mechanics. This has some remnants in the terminology. In structural mechanics, the partial differential formulation of a problem can be replaced by an equivalent variational formulation, i.e. the minimization of an energy integral over the domain.

The variational formulation is a natural integral formulation for the FEM. In fluid dynamics, in general, a variational formulation is not possible. This makes it less obvious how to formulate a finite element method. The history of computational fluid dynamics (CFD) shows that every essential break-through has first been made in the context of the finite difference method or the finite volume method and that it always has taken considerable time, often much more than a decade, to incorporate the same idea into the finite element method. The history of CFD, on the other hand, also shows that, once a suitable FEM-formulation has been found, the FEM is almost exclusively used. This is due to the advantages with respect to the treatment of complex geometries and obtaining higher order accuracy.

The development of the finite element method in fluid dynamics is at present still far from ended. For the simplest problems such as potential flows, both compressible and incompressible, and incompressible Navier-Stokes flows at low Reynolds numbers, the finite element method is more or less full-grown, although new evolutions, certainly for Navier-Stokes problems, are still continuing. More complex problems like compressible flows governed by Euler- or Navier-Stokes equations or incompressible viscous flows at high Reynolds numbers still form an area of active research.

In this introductory text, the option is taken to explain the basic ingredients of the finite element method on a very simple, purely mathematical, problem and to give fluid dynamics illustrations in detail only for simple problems. For more complex problems, only a basic description is given with reference to further literature. Also in the explanation of the method, all mathematical aspects are systematically avoided. For the mathematical aspects, reference is made to further literature. This makes the text accessible for a reader with almost no knowledge of functional analysis and numerical analysis. For the fluid dynamics illustrations, the option has been taken to use only simple techniques, so that the detailed examples can be reproduced by the reader not really familiar with general computational fluid dynamics or even general fluid dynamics. This text therefore is to be seen as the absolute minimum introduction to the subject. The text is in no way complete and the author deliberately has taken the risk to be seen as naive by a more informed reader. A reference list is given for a deeper introduction. A reader beginning with computational fluid dynamics should be aware that a complete study of the finite element method may take considerable time and may necessitate, depending on background, a considerable effort. The method is much less intuitive than the finite difference method and the finite volume method and requires a more fundamental attitude for mathematical formulations. This introductory text therefore is also meant to create some enthusiasm for the method by showing its power with simple examples and to justify in this way the need for further study. It is the conviction of the author that a practitioner of CFD, even if it is not his or her intention to use the FEM, should have at least a basic knowledge of the method. This is in particular useful with respect to the treatment of boundary conditions. Also one should consider that the impact of the FEM in CFD is already extremely important and that it probably will grow in the future.

4.2 Strong and Weak Formulations of a Boundary Value Problem:

4.2.1 Strong Formulation:

Consider as an example, the following simple one-dimensional boundary value problem, consisting of the differential equation

$$\frac{d}{dx}\left(\lambda\frac{du}{dx}\right) = f \quad \text{on} \quad 0 \le x \le X \qquad (4.1)$$

and the boundary conditions

$$u(0) = u_0 \qquad (4.2)$$

and

$$\lambda\frac{du}{dx}(X) = q \qquad (4.3)$$

More generally, the differential equation is denoted by

$$a(u) = f \tag{4.4}$$

The domain to which it applies is denoted by Ω. The boundary condition of type (4.2) is called a Dirichlet boundary condition. More generally, it is denoted by

$$b_0(u) = g_0 \tag{4.5}$$

The boundary condition of type (4.3), which is formulated on the flux of the variable, is called a Neumann boundary condition. More generally, it is denoted by

$$b_1(u) = g_1 \tag{4.6}$$

The boundary of the domain Ω is denoted by Γ. The part to which the Dirichlet boundary condition applies is Γ0 and the part to which the Neumann boundary condition applies is Γ1.

The boundary value problem (4.1, 4.2 and 4.3) is said to be in its strong form, requiring the satisfaction of the differential equation (4.1) in all points of the domain Ω, the satisfaction of the Dirichlet boundary condition (4.2) in all points (here one) of the part of the boundary Γ0 and the satisfaction of the Neumann boundary condition (4.3) in all points (here one) of the part of the boundary Γ1.

One way of relaxing the requirements of the boundary value problem, notably the finite difference way, consists in requiring the approximate satisfaction of the differential equation and the boundary conditions in a finite number of points in the domain and at the boundary. These points usually are chosen to belong to a mesh with some form of regularity. For the one-dimensional domain, a typical mesh or grid is obtained by choosing equally spaced grid points, as shown on Fig. 4.1.

The grid spacing is denoted by Δx. Following standard finite difference methodology, du/dx is approximated in the mid-point of the interval (xi, xi+1) by

$$\left(\frac{du}{dx}\right)_{i+1/2} \approx \frac{u_{i+1} - u_i}{\Delta x} \tag{4.7}$$

Similarly, in the mid-point of the interval (xi−1, xi), the approximation is

$$\left(\frac{du}{dx}\right)_{i-1/2} \approx \frac{u_i - u_{i-1}}{\Delta x} \tag{4.8}$$

Using (4.7) and (4.8), (4.1) can be approximated by

$$\frac{\lambda_{i+1/2}(u_{i+1}-u_i)-\lambda_{i-1/2}(u_i-u_{i-1})}{\Delta x^2}=f_i \tag{4.9}$$

For constant λ, this simplifies to

$$\lambda\frac{u_{i+1}-2u_i+u_{i-1}}{\Delta x^2}=f_i \tag{4.10}$$

The Dirichlet boundary condition (4.2) is simply

$$u_0 = u_0 \tag{4.11}$$

The Neumann boundary condition can be introduced by the image point method. In this method, a point outside the domain (N+ 1) is defined which afterwards is eliminated. The discretization of the differential equation (1) in the end point of the domain is given by (4.9) for i = N.

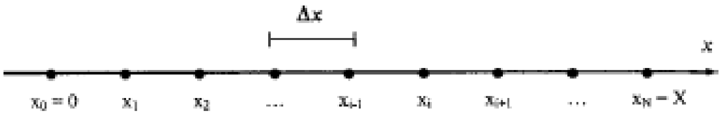


Fig. 4.1 Construction of a finite difference grid over the interval 0 ≤ × ≤ X

The discretization of the Neumann boundary condition (4.3) is

$$1/2\frac{\lambda_{N+1/2}(u_{N+1}-u_N)}{\Delta x}+1/2\frac{\lambda_{N-1/2}(u_N-u_{N-1})}{\Delta x}=q$$

Combination with the discretized differential equation gives

$$q-\lambda_{N-1/2}\frac{(u_N-u_{N-1})}{\Delta x}=1/2 f_N\Delta x \tag{4.12}$$

The resulting discretization is of second order. By taking the Taylor expansion of (4.10), this is obvious (for constant λ) for points inside the domain. At the Neumann boundary, the Taylor expansion up to second order (for constant λ) gives

$$u_{N-1}\approx u_N-\Delta x\left(\frac{du}{dx}\right)_N+1/2\Delta x^2\left(\frac{d^2u}{dx^2}\right)_N$$

Using the Neumann boundary condition

$$\lambda\left(\frac{du}{dx}\right)_N = q$$

and the differential equation in node N

$$\lambda\left(\frac{d^2u}{dx^2}\right)_N = f_N$$

this becomes

$$u_{N-1} \approx u_N - \frac{\Delta x}{\lambda}q + {}^1\!/_2\frac{\Delta x^2}{\lambda}f_N$$

For constant λ, this equation is identical to (4.12).

The originally continuous boundary value problem is now replaced by a discrete problem, consisting of the solution of the set of algebraic equations

$$K\,U = F \tag{4.13}$$

Where U is the vector consisting of the elements (u1, u2, . . . , uN), K is a matrix given by (in the case λ is a constant)

$$K = \begin{bmatrix} 2 & -1 & & & \\ -1 & 2 & -1 & & \\ & & \dots & & \\ & & -1 & 2 & -1 \\ & & & -1 & 1 \end{bmatrix}$$

and F is the right hand side, given by

$$F = \begin{bmatrix} u_0 - \dfrac{\Delta x^2}{\lambda}f_1 \\ -\dfrac{\Delta x^2}{\lambda}f_2 \\ \vdots \\ -\dfrac{\Delta x^2}{\lambda}f_{N-1} \\ \dfrac{\Delta x}{\lambda}q - \dfrac{\Delta x^2}{2\lambda}f_N \end{bmatrix}$$

The most typical feature of the finite difference method is that it only gives information about the function values at the grid points, but no information on the function values between these points.

4.2.2 Weighted Residual Formulation

The first basic ingredient of the finite element method is that an approximate solution is sought which belongs to some finite dimension function space. This function space is to be specified more in detail later. For the time being, we look for an approximate solution of the boundary value problem (4.1, 4.2 and 4.3) which has the form

$$\hat{u} = \psi + \sum_{k=1}^{N} \phi_k \, u_k \qquad (4.14)$$

where ψ is a function which satisfies the boundary conditions (4.2) and (4.3).For the given problem, the construction of ψ is obvious. The functions φk are called basis functions or shape functions. Since the dimension of the function space Φ = {φk; k = 1, 2, . . . , N} is finite, in general, an expression of type (4.14) cannot satisfy the differential equation (4.1) in each point of the domain. This means that the approximate solution ˆu cannot be identical with the exact solution u. Of course, the shape functions should be chosen so that by enriching the function space Φ, i.e. letting N grow, the approximation obtained by (10.14) becomes better. This means that the approximate solution converges to the exact solution. This is called the completeness requirement of the function space. Since a function ˆu given by (4.14) cannot satisfy the differential equation (4.1),upon substitution of (4.14) into (4.1), a residual is left:

$$r_\Omega = a(\hat{u}) - f \quad \text{in } \Omega \qquad (4.15)$$

An approximate solution of the boundary value problem now is obtained by finding a way to make this residual small in some sense. In the finite element method this is done by requiring that an appropriate number of weighted integrals of the residual over Ω be zero:

$$\int_\Omega w_i r_\Omega d\Omega = 0; \quad i = 1, 2, \ldots, N \qquad (4.16)$$

where W= {wi; i = 1, 2, . . . , N} is a set of weighting functions. Obviously, the convergence requirement now also implies a requirement of completeness of the space of weighting functions, i.e. (4.16) should imply rΩ →0 for N→∞.

Clearly, with satisfaction of the completeness, for N→∞, the weighted residual formulation (4.16) for a function of form (4.14) is completely equivalent to the strong formulation of the problem (4.1, 4.2 and 4.3). An approximate solution then is obtained for N being finite.

4.2.3 Galerkin Formulation

Among the possible choices for the set of weighting functions, the following ones are the most obvious.

The weighting functions can be chosen to be Dirac-delta functions in N points. This choice means making the residual equal to zero in a number of chosen points. The method is called the point collocation method. Obviously, it has much in common with the finite difference methodology.

A second possible choice of weighting functions is given by

$$\begin{aligned} w_i &= 1 \quad \text{for } x_i \leq x \leq x_{i+1} \\ &= 0 \quad \text{for } x < x_i \text{ or } x > x_{i+1} \end{aligned}$$

The weighted residual statements (4.16) now require the integral of the residual to be zero on N subdomains. This method is called the subdomain collocation method. The finite volume method, in which not the differential form of the equation but the integral form of the equation is discretized, is a special form of this method.

The most popular choice for the weighting functions in the finite element method is the shape functions themselves:

$$w_i = \phi_i$$

This method is called the Galerkin method. Its meaning is that the residual is made to be orthogonal to the space of the shape functions.

To illustrate the Galerkin method, consider the boundary value problem (4.1–4.3) with constant λ. Then:

$$\psi = u_0 + \frac{q}{\lambda} x$$

Consider further as an example of (4.14) a Fourier-sine expansion of u:

$$\hat{u} = \psi + \sum_{k=1}^{N} u_k \sin \frac{\pi k' x}{X}, \quad \text{with } k' = k - ½$$

Then:

$$r_\Omega = -\lambda \sum_{k=1}^{N} u_k \left(\frac{\pi k'}{X} \right)^2 \sin \frac{\pi k' x}{X} - f$$

The Galerkin method then gives

$$\lambda \sum_{k=1}^{N} u_k \left(\frac{\pi k'}{X} \right)^2 \int_0^X \sin \frac{\pi k' x}{X} \sin \frac{\pi i' x}{X} dx = -\int_0^X \sin \frac{\pi i' x}{X} f\, dx$$

Then noting that

$$\int_0^X \sin\frac{\pi k' x}{X}\sin\frac{\pi i' x}{X}dx = \frac{X}{2} \quad \text{for } k' = i'$$
$$= 0 \quad \text{for } k' \neq i'$$

we find

$$u_i = -\frac{2X}{\lambda\pi^2 i'^2}\int_0^X f\sin\frac{\pi i' x}{X}dx$$

The foregoing method used to determine an approximate solution of the boundary value problem (4.1, 4.2 and 4.3) is not a finite element method, but a spectral method. The finite element method however has the same starting point.

Before going on with the study of the building blocks of the finite element method, we should remark that a fourth weighted residual statement exists on which finite element methods can be based. The least squares formulation is based on the minimization of the integral

$$\int_\Omega r_\Omega^2 d\Omega$$

4.2.4 Weak Formulation:

In many problems, it is not practical to construct a function which satisfies the boundary conditions in order to arrive at an expression for the approximate solution, as is done in (14). More generally, an approximate solution can be expressed as

$$\hat{u} = \sum_{k=1}^{N}\phi_k u_k \tag{4.17}$$

Now the approximate solution not only has a residual with respect to the field equation (4.4), but also with respect to the boundary equations (4.5) and (4.6):

$$r_0 = b_0(\hat{u}) - g_0 \tag{4.18}$$

and

$$r_1 = b_1(\hat{u}) - g_1 \tag{4.19}$$

A weighted residual statement is now to be of the form

$$\int_{\Omega} w_i r d\Omega + \int_{\Gamma_0} w_i^0 r_0 d\Gamma + \int_{\Gamma_1} w_i^1 r_1 d\Gamma = 0$$

(4.20)

This complicates the formulation since now additional weighting functions on the boundaries are to be chosen. Since the number of degrees of freedom of the approximate solution (4.17) is N, an equal number of independent weighting functions wi can be chosen, while w0i and w1i are to depend on wi. There is however a natural way to choose the dependent weighting functions on the boundary.

For the problem (4.1, 4.2 and 4.3), (4.20) becomes

$$\int_0^X w_i \left[\frac{d}{dx}\left(\lambda \frac{d\hat{u}}{dx}\right) - f\right] dx + w_i^0[\hat{u}(0) - u_0] + w_i^1 \left[\lambda \frac{d\hat{u}}{dx}(X) - q\right] = 0$$

(4.21)

where the weighting functions at the boundary reduce to weighting factors w0i and w1i.

By one integration by parts on the first term, (4.21) becomes

$$w_i \lambda \frac{d\hat{u}}{dx}\bigg|_0^X - \int_0^X \lambda \frac{dw_i}{dx}\frac{d\hat{u}}{dx} dx - \int_0^X w_i f \, dx + w_i^0[\hat{u}(0) - u_0] + w_i^1 \left[\lambda \frac{d\hat{u}}{dx}(X) - q\right] = 0$$

This weighted residual statement is simplified by choosing the weighting factors on the Neumann boundary by

$$w_i^1 = -w_i(X)$$

The weighted residual statement then becomes

$$-\int_0^X \lambda \frac{dw_i}{dx}\frac{d\hat{u}}{dx} dx - w_i(0)\lambda \frac{d\hat{u}}{dx}(0) - \int_0^X w_i f \, dx + w_i^0[\hat{u}(0) - u_0] + w_i(X) q = 0$$

Furthermore, if the Dirichlet boundary condition can be imposed on the approximate solution, the weighting functions and the weighting factors can be chosen to be zero at the Dirichlet boundary, so that the weighted residual statement further simplifies to

$$-\int_0^X \lambda \frac{dw_i}{dx}\frac{d\hat{u}}{dx} dx - \int_0^X w_i f \, dx + w_i(X) q = 0$$

(4.22)

subject to the Dirichlet boundary conditions

$$\hat{u}(0) = u_0 \quad w_i(0) = 0$$

(4.23)

The weighted residual statement in form (4.22) is called the weak formulation.

The weak formulation (4.22 and 4.23) is not completely equivalent to the strong formulation (4.1, 4.2, 4.3), even not for N → ∞. By the construction of the weak formulation, any solution of the strong formulation satisfies the weak formulation. The reverse, however, is not true. The weak formulation allows solutions which have a lower degree of regularity than required for the strong solution. This is the origin of the term weak and strong. For instance for the problem (4.1, 4.2, 4.3), the solution must have continuous first derivatives. We express this by saying that the degree of regularity is to be C1. The corresponding weak formulation (4.22 and 4.23) only requires continuity of the function value itself. The necessary degree of regularity is here C0. This means that functions with discontinuous first derivatives are allowed by (4.22). We remark that this is precisely, certainly in fluid mechanics, what we want! Indeed, in fluid mechanics, the governing equations are obtained from integral statements, i.e. conservation laws, requiring a lower degree
of regularity than the partial differential equations which are obtained from these statements.

To conclude, we remark that the weak formulation (4.22), in case of sufficient regularity, through reverse integration by parts leads to a simplification of (4.21):

$$\int_0^X w_i \left[\frac{d}{dx}\left(\lambda \frac{d\hat{u}}{dx} \right) - f \right] dx - w_i(X) \left[\lambda \frac{d\hat{u}}{dx}(X) - q \right] = 0 \qquad (4.24)$$

For an infinite number of degrees of freedom (N→∞), this implies exact satisfaction of the differential equation and the Neumann boundary condition.

In the weak formulation (4.22 and 4.23), the Neumann boundary condition need not be imposed in an explicit way to the solution. Boundary conditions of this type enter through the integration by parts in a natural way into the formulation. Therefore these boundary conditions are called natural boundary conditions. The boundary conditions which have to be imposed explicitly in the weak formulation are called essential boundary conditions.

4.2.5 Variational formulation

For elliptic self-adjoint boundary value problems, the weak formulation is equivalent to the minimization of the functional associated to the boundary value problem. Historically, this minimization formulation, or variational formulation, has played a big role in the development of the finite element method. Variational methods still have an important role in, for instance, structural mechanics. Also the variational formulation plays an important role in the mathematical theory of finite element methods, for instance, with respect to questions on solvability and uniqueness.

UNIT V

FINITE VOLUME TECHNIQUES

Finite Volume Techniques - Cell Centered Formulation - Lax - Vendor off Time Stepping - Runge - Kutta Time Stepping - Multi - stage Time Stepping - Accuracy - Cell Vertex Formulation - Multistage Time Stepping - FDM -like Finite Volume Techniques - Central and Up-wind Type Discretizations - Treatment of Derivatives. Flux – splitting schemes. Pressure correction solvers – SIMPLE, PESO. Vorticity transport formulation. Implicit/semi-implicit schemes.

5.1 Finite Volume Techniques

The basic laws of fluid dynamics are conservation laws. They are statements that express the conservation of mass, momentum and energy in a volume closed by a surface. Only with the supplementary requirement of sufficient regularity of the solution can these laws be converted into partial differential equations. Sufficient regularity cannot always be guaranteed. Shocks form the most typical example of a discontinuous flow field. In case discontinuities occur, the solution of the partial differential equations is to be interpreted in a weak form, i.e. as a solution of the integral form of the equations. For example, the laws governing the flow through a shock, i.e. the Hugoniot-Rankine laws, are combinations of the conservation laws in integral form. For a correct representation of shocks, also in a numerical method, these laws have to be respected.

There are additional situations where an accurate representation of the conservation laws is important in a numerical method. A second example is the slip line which occurs behind an airfoil or a blade if the entropy production is different on streamlines on both sides of the profile. In this case, a tangential discontinuity occurs. Another example is incompressible flow where the imposition of incompressibility, as a conservation law for mass, determines the pressure field.

In the cases cited above, it is important that the conservation laws in their integral form are represented accurately. The most natural method to accomplish this is to *discretize the integral form of the equations* and not the differential form. This is the basis of a *finite volume* *method*. Further, in cases where strong conservation in integral form is not absolutely necessary, it is still physically appealing to use the basic laws in their most primitive form.

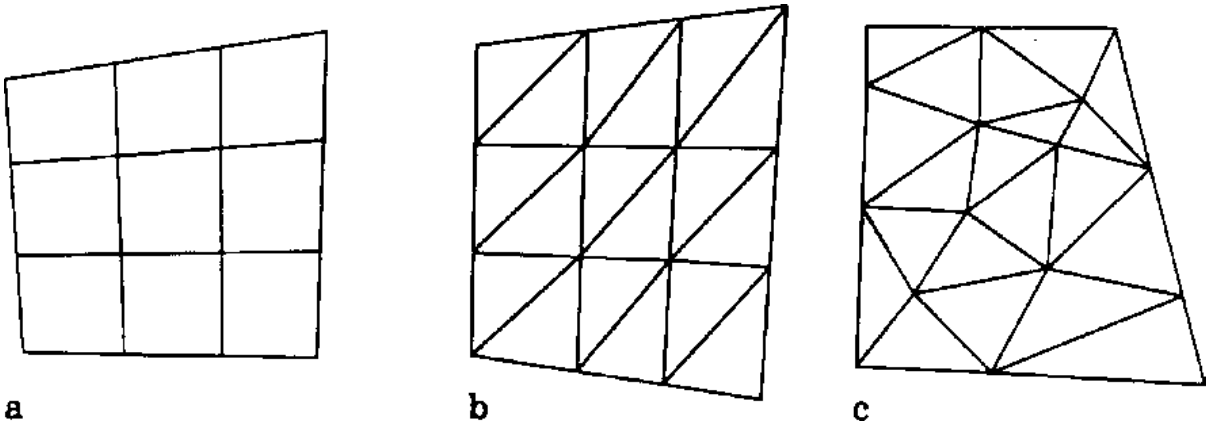


Fig. 5.1 *Typical choice of grids in the FVM; (**a**): structured quadrilateral grid; (**b**): structured triangular grid; (**c**): unstructured triangular grid*

The flow field or *domain* is subdivided, as in the finite element method, into a set of *non-overlapping cells* that *cover the whole domain*. In the finite volume method (FVM) the term *cell* is used instead of the term *element* used in the finite element method (FEM). The conservation laws are applied to determine the flow variables in some discrete points of the cells, called *nodes*. As in the FEM, these nodes are at typical locations of the cells, such as cell-centres, cell-vertices or mid sides. Obviously, there is considerable freedom in the choice of the cells and the nodes. Cells can be *triangular, quadrilateral*, etc. They can form a *structured grid* or an *unstructured grid*. The whole geometrical freedom of the FEM can be used in the FVM. Figure 5.1 shows some typical grids. The choice of the nodes can be governed by the wish to represent the solution by an interpolation structure, as in the FEM. A typical choice is then *cell-centres* for representation as piecewise constant functions or *cell-vertices* for representation as piecewise linear (or bilinear) functions. However, in the FVM, a function space for the solution need not be defined and nodes can be chosen in a way that does not imply an interpolation structure. Figure 5.2 shows some typical examples of choices of nodes with the associated definition of variables.

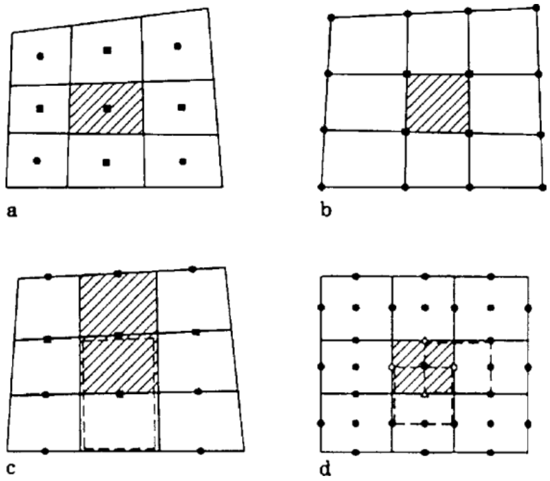


Fig. 5.2 Typical *choice of nodes in the FVM. The marked nodes are used in the flux balance of the control volume.(**a**): piecewise constant interpolation structure; (**b**): piecewise linear interpolation structure; (**c**): no interpolation structure with all variables defined in each node; (**d**): no interpolation structure with not all variables defined in each node; (Cartesian grid), o: ρ and p,: u, Δ: v*

The first two choices imply an interpolation structure, the last two do not. In the last example, function values are not defined in all nodes. The grid of nodes on which pressure and density are defined is different from the grid of nodes on which velocity-x components and velocity-y components are defined. This approach commonly is called the *staggered grid approach*. The third basic ingredient of the method is the choice of the volumes on which the conservation laws are applied. In Fig. 5.2 some possible choices of control volumes are shown (shaded). In the first two examples, control volumes coincide with cells. The third example in Fig. 5.2 shows that the *volumes* on which the conservation laws are applied *need not coincide with the cells* of the grid. Volumes even can be overlapping. Figure 5.3 shows some typical examples of volumes not coinciding with cells, for overlapping and non-overlapping cases. The term volume denotes the control volume to which the conservation laws are applied (i.e. connected to function value determination), while the term cell denotes a mesh of the grid (i.e. connected to geometry discretization). A consistency requirement for the cells is that they are non-overlapping and that they span the whole domain. The consistency requirement for the volumes is weaker. They can be overlapping so that families of volumes are formed. Each family should consist of non-overlapping volumes which span the whole domain. The consistency requirement is that a flux leaving a volume should enter another one.

Obviously, by the decoupling of volumes and cells, the freedom in the determination of the function representation of the flow field in the finite volume method becomes much larger than in both the finite element and finite difference method. It is in particular the combination of the formulation of a flow problem on control volumes which is the most *physical* way to obtain a discretization, with the *geometric* flexibility in the choice of the grid and the flexibility in defining the discrete *flow variables* which makes the finite volume method attractive for engineering applications.

The finite volume method (FVM) tries to combine the best from the *finite element method* (FEM), i.e. the *geometric flexibility*, with the best of the *finite difference method* (FDM), i.e. the flexibility in defining the *discrete flow field* (discrete values of dependent variables and their associated fluxes). Some formulations are near to finite element formulations and can be interpreted as subdomain collocation finite element methods (e.g. Fig. 5.2a). Other formulations are near to finite difference formulations and can be interpreted as conservative finite difference methods (e.g. Fig. 5.3a). Other formulations are in between these limits.

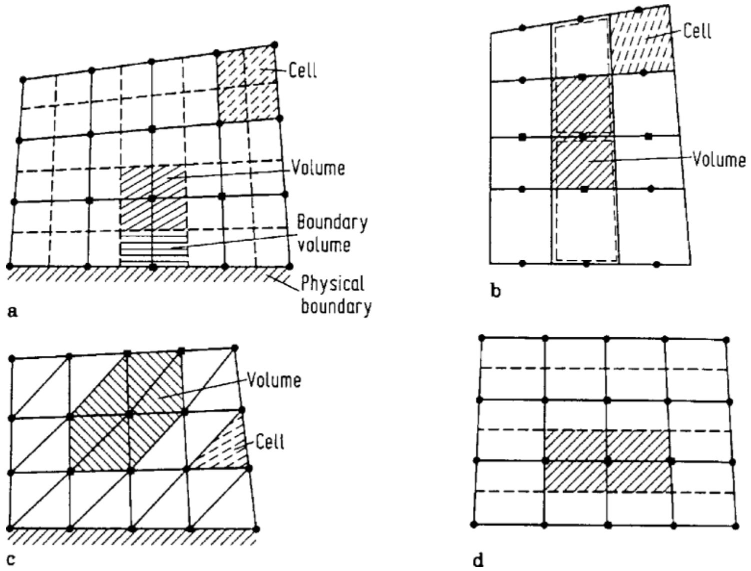


***Fig. 5.3** Choice of volumes not coinciding with cells, overlapping and non-overlapping cases. (**a**): volumes staggered with respect to cells, non-overlapping case; (**b**): volumes non-staggered with respect to cells, overlapping case; (**c**): volumes non-staggered with respect to cells, overlapping case; (**d**): volumes staggered with respect to cells, overlapping case*

The mixture of FEM-like and FDM-like approaches sometimes leads to confusion in terminology. Some authors with an FEM-background use the term *element* for *cell* and then often use the term (control) *cell* for (control) *volume*. Strictly speaking, the notion element is different from the notion cell. A grid is subdivided into meshes. A mesh has the significance of a cell if it only implies a subdivision of the geometry. If it also implies, in the FEM-sense, a definition of a function space, it is an element.

From the foregoing, it could be concluded that the FVM only has advantages over the FEM and the FDM and thus one could ask why all of computational fluid dynamics (CFD) is not based on the FVM. From the foregoing, it is already clear that the FVM has a difficulty in the accurate definition of derivatives. Since the computational grid is not necessarily orthogonal and equally spaced, as in the FDM, a definition of a derivative based on a Taylor-expansion is impossible. Also, there is no mechanism like a weak formulation, as in the FEM, to convert higher order derivatives into lower ones. Therefore, the FVM is best suited for flow problems in primitive variables, where the viscous terms are absent (Euler equations) or are not dominant (high Reynolds number Navier-Stokes equations). Further, a FVM has difficulties in obtaining higher order accuracy. Curved cell boundaries, as used in the FEM, or curved grid lines, as used in the FDM, are difficult to implement. In most methods, boundaries of cells are straight and grid lines are piecewise straight. Representation of function values or fluxes better than piecewise constant or piecewise linear is possible but rather complicated. Most FVM methods are only second order accurate. For many engineering applications, this accuracy is sufficient. The development of finite volume methods with better accuracy is nowadays an area of very active research and there is still no clear insight in how to reach higher accuracy in an efficient way.

Therefore, in the following, we focus on the Euler equations. So, for explanation of the basic algorithms, we avoid the discussion of the determination of derivatives. We treat methods for construction of derivatives at the end. Further, we only discuss classic algorithms with second-order spatial accuracy. For simplicity we do not discuss implicit time stepping schemes, since the choice between implicit schemes and explicit schemes is not linked to the choice of the space discretization. This introductory text also does not aim to give a complete overview of the FVM. It only aims to illustrate some of the basic properties on examples of methods that are widely used.

Fem-Like Finite Volume Methods

FEM-like finite volume methods use cells to which an interpolation structure is associated. So, the cells form elements in the FEM-sense. Two interpolation structures can be used: piecewise constant interpolation and piecewise linear (or bilinear) interpolation. Figure 5.4 shows some possibilities on (structured) quadrilateral and triangular grids. The piecewise constant interpolation is denoted by the *cell-centred* method, while the piecewise linear interpolation is denoted by the *cell vertex* method. In both methods, the cells and a group of cells around a node are used as volumes. In the first method, data are at cell centres. In the second method, data are al cell vertices.

We illustrate here some formulations for the Euler equations. The set of Euler equations can be written in two dimensions as

$$\frac{\partial U}{\partial t} + \frac{\partial f}{\partial x} + \frac{\partial g}{\partial y} = 0 \qquad \text{----------------------5.1}$$

with

$$U = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho E \end{bmatrix}, \qquad f = \begin{bmatrix} \rho u \\ \rho uu + p \\ \rho uv \\ \rho Hu \end{bmatrix}, \qquad g = \begin{bmatrix} \rho v \\ \rho uv \\ \rho vv + p \\ \rho Hv \end{bmatrix}$$

Where ρ is density, u and v are Cartesian components of velocity, p is pressure, E is total energy and H is total enthalpy (γ is the adiabatic constant).

$$E = \frac{1}{\gamma - 1}\frac{p}{\rho} + {}^{1}\!/_{2}u^2 + {}^{1}\!/_{2}v^2$$

$$H = E + \frac{p}{\rho}$$

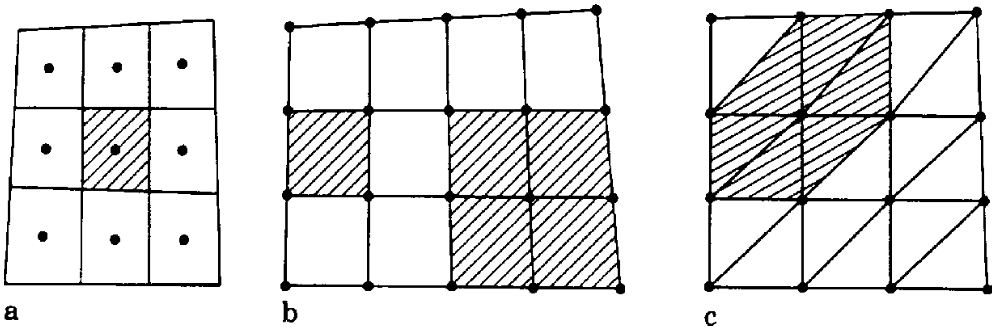


Fig. 5.4 *FEM-like finite volume methods. (**a**): cell-centred; (**b**): cell-vertex with non-overlapping*

*and overlapping volumes on quadrilateral cells; (**c**): cell-vertex on triangular cells*

5.2 Cell Centered Formulation

For a cell as shown in Fig. 5.5, the values of the dependent variables are stored in the centre of the cell. These values do not necessarily have to be seen as nodal values, but can also be seen as mean values over the cell. Therefore, in the cell-centred method, for visualization purposes, often, after completion of the calculations, values are attributed to the vertices of the grid by taking a weighted mean of the values in adjacent cells. Further, the interpretation as mean values allows higher order formulation, as we discuss in Sect. 5.6. First, we discuss the typical second-order accurate formulations.

Using the control volume of Fig. 5.5, a semi-discretization of (5.1) is obtained By

$$\Omega_{i,j}\frac{\partial U}{\partial t} + \int_{abcd} \vec{F}.\vec{n}\,dS = 0$$

where $\Omega_{i,j}$ denotes the volume (area) of the control volume. $\vec{F}$ is the flux vector: $\vec{F} = f\,\vec{1}_x + g\,\vec{1}_y$, dS is a surface element and $\vec{n}$ is the outward normal. By taking the positive sense as indicated in the figure, we have

$$\vec{n}\,dS = dy\,\vec{1}_x - dx\,\vec{1}_y$$

Inserting (5.3) into (5.2) gives

$$\Omega_{i,j}\frac{\partial U}{\partial t} + \int_{abcd} (f\,dy - g\,dx) = 0$$

Further, f and g have to be defined on the boundary of the volume. A mean value between adjacent nodes looks to be the simplest choice, for example:

$$f_{ab} = {}^1\!/\!{}_2(f_{i,j} + f_{i,j-1}), \qquad g_{ab} = {}^1\!/\!{}_2(g_{i,j} + g_{i,j-1})$$

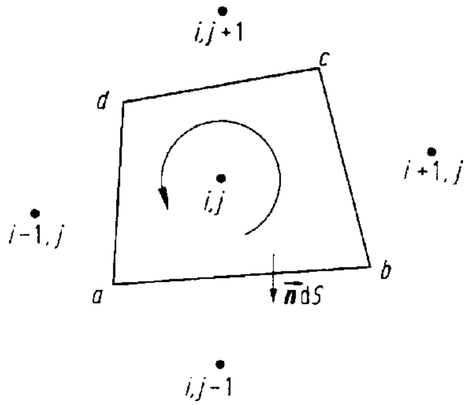


Fig. 5.5 *Cell-centred formulation*

Since the flux functions are non-linear functions of the dependent variables, an alternative for (5.5) is

$$f_{ab} = f[{}^1\!/\!_2(U_{i,j} + U_{i,j-1})], \qquad g_{ab} = g[{}^1\!/\!_2(U_{i,j} + U_{i,j-1})]$$

With (5.6) is meant that the dependent variables are first averaged and that afterwards flux vectors are calculated. This is not a popular choice, since it implies about twice as many flux evaluations as (5.5). Indeed, when in a structured quadrilateral grid, there are nx subdivisions in longitudinal direction and ny subdivisions in transversal direction, then there are nxny cells, but nx(ny + 1) + ny(nx + 1) cell faces. This does not imply that the work involved in (5.6) is twice as much as the work involved in (5.5). A lot of computational effort can be gained by remarking that a momentum flux is a mass flux multiplied by an average velocity, etc. Nevertheless, the definition (5.5) is the cheapest. Therefore, (5.5) is the only central flux definition used in the following (one-sided flux definitions are also possible, as discussed later). With the definition of the discrete fluxes f and g, the semi-discretization (5.4) is completed. It is now to be integrated in time.

5.3 Lax Wendroff Time Stepping

Since Lax-Wendroff time-stepping is a very classic explicit time integration method in the finite difference method, explained in previous chapters, we begin by discussing how this time-stepping can be applied to a finite volume formulation. We first recall the principles of a Lax-Wendroff method with the use of the one dimensional scalar model equation

$$\frac{\partial u}{\partial t} + \frac{\partial f(u)}{\partial x} = 0$$

A Taylor series expansion to second order gives

$$u^{n+1} \approx u^n + \Delta t\left(\frac{\partial u}{\partial t}\right)^n + \frac{\Delta t^2}{2}\left(\frac{\partial^2 u}{\partial t^2}\right)^n$$

and

$$\frac{\partial^2 u}{\partial t^2} = \frac{\partial}{\partial t}\left(\frac{\partial u}{\partial t}\right) = -\frac{\partial}{\partial t}\left(\frac{\partial f}{\partial x}\right) = -\frac{\partial}{\partial x}\left(\frac{\partial f}{\partial t}\right)$$

or

$$\frac{\partial^2 u}{\partial t^2} = -\frac{\partial}{\partial x}\left(\frac{\partial f}{\partial u}\frac{\partial u}{\partial t}\right) = \frac{\partial}{\partial x}\left(a\frac{\partial f}{\partial x}\right)$$

with

$$a = \frac{\partial f}{\partial u}$$

Combination of (5.8) and (5.9) gives

$$u^{n+1} \approx u^n + \Delta t\left(-\frac{\partial f^n}{\partial x}\right) + \frac{\Delta t^2}{2}\frac{\partial}{\partial x}\left(a\frac{\partial f^n}{\partial x}\right)$$

The two-dimensional analogue of (5.10) on the Euler equations (5.1) is

$$U^{n+1} \approx U^n + \Delta t\left(-\frac{\partial f^n}{\partial x} - \frac{\partial g^n}{\partial y}\right) + \frac{\Delta t^2}{2}\left\{\frac{\partial}{\partial x}\left[A^n\left(\frac{\partial f^n}{\partial x} + \frac{\partial g^n}{\partial y}\right)\right] + \frac{\partial}{\partial y}\left[B^n\left(\frac{\partial f^n}{\partial x} + \frac{\partial g^n}{\partial y}\right)\right]\right\}$$

Where A and B are the Jacobian matrices of the flux vectors:

$$A = \frac{\partial f}{\partial U} \quad , \qquad B = \frac{\partial g}{\partial U}$$

In the finite-difference method, a discretization of (5.10) or (5.5) is called a one-step Lax-Wendroff method. As explained in previous chapters, a possible procedure is to expand the second-order derivatives in space in (5.10) or (5.5) and to replace these derivatives by central difference approximations. In principle, a finite volume formulation on (5.10) or (5.5) is possible since these equations take the form of a flux-balance. The fluxes contain however derivatives. Since the definition

of derivatives is not simple in the finite volume method, one-step methods are never used. The most popular two-step formulations, such as the Richtmyer variant and the MacCormack variant, can however be used without problems in the FVM.

Further, in the one-step method the primitive flux balances are lost while these are visible in the two-step formulations. Since the MacCormack variant was explained in previous chapters, we illustrate here how this variant can be formulated in finite volume form.

In the MacCormack variant of the Lax-Wendroff method, (5.8) is written as

$$u^{n+1} = {}^1\!/\!_2 u^n + {}^1\!/\!_2 \Delta t \left(\frac{\partial u}{\partial t}\right)^n + {}^1\!/\!_2 u^n + {}^1\!/\!_2 \Delta t \left[\frac{\partial}{\partial t}\left(u + \Delta t \frac{\partial u}{\partial t}\right)\right]^n$$

With (predictor)

$$u^{\overline{n+1}} = u^n + \Delta t \left(\frac{\partial u}{\partial t}\right)^n$$

(5.12) can be written as (corrector)

$$u^{n+1} = {}^1\!/\!_2\left[u^n + u^{\overline{n+1}} + \Delta t \frac{\partial}{\partial t} u^{\overline{n+1}}\right]$$

The discretization by MacCormack of (5.13) and (5.14) is

$$u_i^{\overline{n+1}} = u_i^{\,n} - \Delta t \left(\frac{f_{i+1}^n - f_i^n}{\Delta x}\right)$$

$$u_i^{\,n+1} = {}^1\!/\!_2\left[u_i^{\,n} + u_i^{\overline{n+1}} - \Delta t \left(\frac{f_i^{\overline{n+1}} - f_{i-1}^{\overline{n+1}}}{\Delta x}\right)\right]$$

Equations (5.15) and (5.16) form the forward-backward variant. Obviously the forward and backward discretizations can be interchanged. In the terminology of ordinary differential equations, the MacCormack method is a predictor-corrector method.

The implementation of the MacCormack variant of the Lax-Wendroff method is rather straightforward. In the forward-backward formulation, in the predictor step on Fig. 5.5, the fluxes at the sides ab, bc, cd and da are evaluated with function values in the nodes (i,j), (i+1, j), (i, j+1) and (i,j), respectively. In the corrector step this is (i, j−1), (i,j), (i,j) and (i−1, j).

At inflow and outflow boundaries, the FVM can be used as the FDM. This means that, in general, extrapolation formulas are used to define values in nodes outside the domain. For instance, for a subsonic inflow, it is common practice to extrapolate the Mach number from the flow field and to impose stagnation properties and flow direction. At a subsonic outflow, the reverse can be done, i.e. extrapolation of stagnation properties and flow direction and fixing of a Mach number. Very often,
pressure is imposed at outflow.

At solid boundaries, the convective flux can be set to zero. This means that in the flux through a cell surface on a solid boundary, only the pressure comes in:

$$f\,dy - g\,dx = p\begin{bmatrix} 0 \\ dy \\ -dx \\ 0 \end{bmatrix}$$

The pressure at the boundary can be taken to be the pressure in the cell. Sometimes, as in the FDM, an extrapolation of pressure is used. It is however not always easy to define extrapolation formulas on distorted or unstructured grids.

Obviously, four geometrical variants in the choice of the biasing of the fluxes are possible. Figure 5.6 shows schematically the possibilities for the predictor step. In

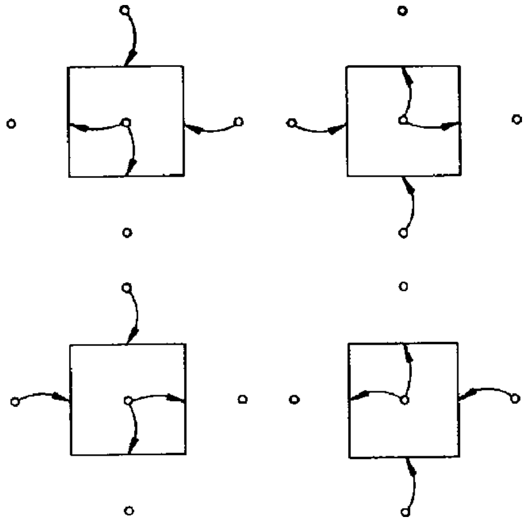


Fig. 5.6 Possible variants of the biasing for flux functions in the predictor step of a MacCormack method

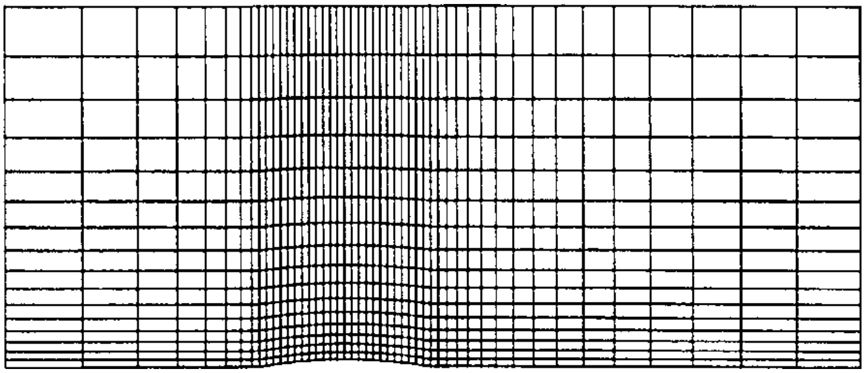


Fig. 5.7 GAMM-channeltest problem

the corrector step, the biasing is inverted. In practice, the four possibilities are used alternatively.

We illustrate now the cell-centred MacCormack scheme on the well-known GAMM-channel test problem for transonic flows [1]. This problem is shown in Fig. 5.7, discretized with a 49×17 grid. The result shown in Fig. 5.8 is however obtained on a once refined grid, i.e. a 97×33 grid. The channel of Fig. 5.7 is almost straight except for a small circular perturbation on the lower boundary with height 4.2% of the chord. The result of Fig. 5.8 is obtained with the MacCormack method described above. Pressure is imposed at the outlet, corresponding to an isentropic Mach number of 0.85. As in the finite-difference method, to obtain this result, some artificial viscosity is needed to stabilize the solution in the shock region (see discussion in previous chapters). This is done here in a rather primitive way by adding to each step a smoothing of form

$$\mu\left[U^n_{i+1,j}+U^n_{i-1,j}+U^n_{i,j+1}+U^n_{i,j-1}-4U^n_{i,j}\right],$$

where μ is a very small coefficient. For the result in Fig. 5.8: $\mu = 0.001$. This is enough to stabilize the shock. Of course, by increasing μ, the observed wiggles can be eliminated completely, but this increases the smearing of the shock. Therefore it is preferred to keep some of the wiggles in the solution.

The CFL-restriction for the time step in the MacCormack scheme is given by (with c the velocity of sound):

$$\Delta t \le \frac{1}{\frac{|u|}{\Delta x}+\frac{|v|}{\Delta y}+c\sqrt{\frac{1}{(\Delta x)^2}+\frac{1}{(\Delta y)^2}}}$$

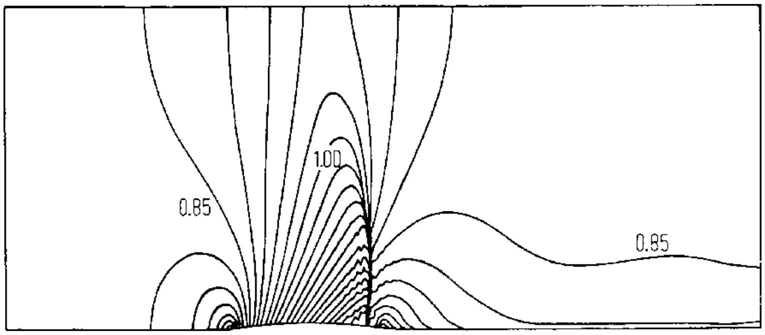


Fig. 5.8 IsoMachlines obtained by cell-centred MacCormack scheme

where

$$\Delta x = \frac{x_{i+1,j} - x_{i-1,j}}{2} \quad , \qquad \Delta y = \frac{y_{i,j+1} - y_{i,j-1}}{2}$$

5.4 Runge Kutta Time Stepping -Multi Stage Time Stepping

Runge-Kutta time stepping schemes for ordinary differential equations are unstable when applied to the semi-discretization (5.4) with the central flux (5.5):

$$\begin{aligned}\Omega_{i,j}\frac{\partial U}{\partial t} &+ {}^1\!/_2(\Delta y_{ab}\, f_{i,j-1} - \Delta x_{ab}\, g_{i,j-1}) \\ &+ {}^1\!/_2(\Delta y_{bc}\, f_{i+1,j} - \Delta x_{bc}\, g_{i+1,j}) \\ &+ {}^1\!/_2(\Delta y_{cd}\, f_{i,j+1} - \Delta x_{cd}\, g_{i,j+1}) \\ &+ {}^1\!/_2(\Delta y_{da}\, f_{i-1,j} - \Delta x_{da}\, g_{i-1,j}) = 0\end{aligned}$$

There is no contribution of the central node in the flux balance in (5.17), since the flux balance for a constant flux on a closed surface is zero. As a consequence, (5.17) is an exact analogue of a central type finite difference discretization.

The instability of Runge-Kutta time stepping can be seen by considering a Fourier analysis on a central space discretization of the model equation (5.7) for the case of constant a = ∂f/∂u:

$$\frac{\partial u_i}{\partial t} = -a\frac{u_{i+1} - u_{i-1}}{2\Delta x}$$

Inserting

$$u = Z\, e^{j\omega x}$$

where ω is the wave-number and j now stands for $\sqrt{-1}$, gives

$$Z' = -Z\, a\frac{e^{j\theta} - e^{-j\theta}}{2\Delta x} = -Z\, ja\frac{\sin\theta}{\Delta x}$$

where $\theta = \omega\Delta x$.

Equation (5.19) has the form

$$Z' = \lambda Z$$

With

$$\lambda = -j\, a\frac{\sin\theta}{\Delta x}$$

Figure 5.9 shows the stability domain for λΔt for the Runge-Kutta second, third and fourth-order, time-integration methods, according to [2].

Since λ according to (5.20) is on the imaginary axis, the second-order Runge- Kutta method is unstable. Higher order Runge-Kutta methods are marginally stable.

Higher order Runge-Kutta methods can be stabilized by introducing a small amount of artificial viscosity. For example, equation (5.18) can be modified to

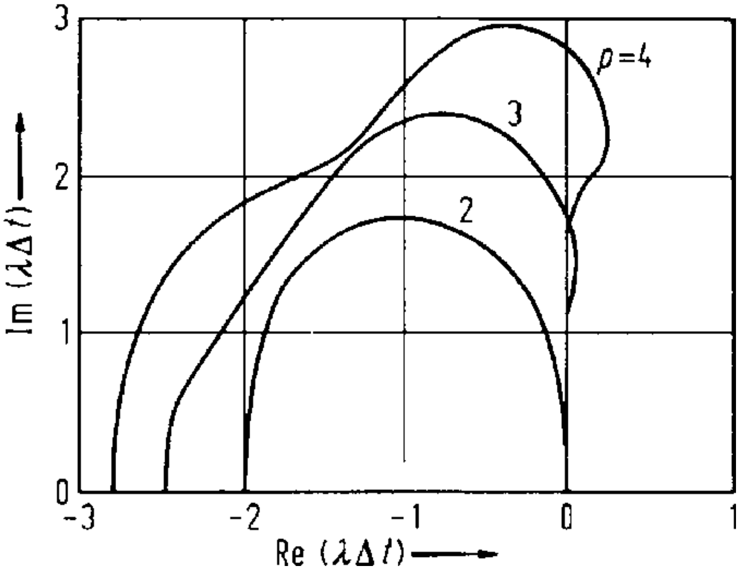


Fig. 5.9 Stability regions in the complex plane for classic explicit Runge-Kutta methods

$$\frac{\partial u}{\partial t} = -a\frac{u_{i+1} - u_{i-1}}{2\Delta x} + \varepsilon\frac{u_{i+1} - 2u_i + u_{i-1}}{\Delta x^2}$$

The value of λ according to the previous analysis now becomes

$$\lambda = -j\, a\frac{\sin\theta}{\Delta x} - \frac{2\varepsilon}{\Delta x^2}(1 - \cos\theta)$$

Since there is now a small negative real part in λ, higher order Runge-Kutta time stepping now becomes stable, according to Fig. 5.9, when subject to a CFLcondition which restricts the time step. Note that a modification of equation (5.18) by adding a fourth-order derivative term instead of a second-order derivative term leads to a similar stabilization effect.

Runge-Kutta time stepping was introduced in the finite volume method by Jameson et al. in 1981 [3] and is nowadays a very popular method.

The fourth-order method, with simplifications, is mostly used since it gives the best ratio of allowable time step to computational work per time step. A simplified fourth-order scheme can be written as

$$U_{i,j}^{0} = U_{i,j}^{n}$$

$$U_{i,j}^{1} = U_{i,j}^{0} - \alpha_1 \frac{\Delta t}{\Omega_{i,j}} R^0$$

$$U_{i,j}^{2} = U_{i,j}^{0} - \alpha_2 \frac{\Delta t}{\Omega_{i,j}} R^1$$

$$U_{i,j}^{3} = U_{i,j}^{0} - \alpha_3 \frac{\Delta t}{\Omega_{i,j}} R^2$$

$$U_{i,j}^{4} = U_{i,j}^{0} - \alpha_4 \frac{\Delta t}{\Omega_{i,j}} R^3$$

$$U_{i,j}^{n+1} = U_{i,j}^{4}$$

with $\alpha_1 = {}^1/_4$, $\alpha_2 = {}^1/_3$, $\alpha_3 = {}^1/_2$, $\alpha_4 = 1$
and where the residual R is given by

$$R = \int (f\,dy - g\,dx)$$

and where the superscript denotes the (intermediate) time level.

Obviously (5.21) is not a classic fourth-order Runge-Kutta scheme. In a Runge- Kutta scheme, the fourth step is

$$U_{i,j}^{4} = U_{i,j}^{0} - \alpha_4 \frac{\Delta t}{\Omega_{i,j}} \left(\frac{R^0 + 2R^1 + 2R^2 + R^3}{6} \right)$$

with the choice of coefficients

$$\alpha_1 = {}^1/_2, \quad \alpha_2 = {}^1/_2, \quad \alpha_3 = 1, \quad \alpha_a = 1$$

The accuracy of the fourth-order Runge-Kutta scheme is fourth order in time. This is unnecessarily high since the space accuracy of the discretization is only second order. The simplification (5.21) has second-order accuracy in time for a non-linear equation, which is sufficient. The simplified multi-stage time-stepping

(5.21) requires less storage than a classic Runge-Kutta time-stepping. Originally, Jameson used the classic Runge-Kutta method. The low storage modification, later introduced by Jameson, is nowadays universally used. For a discussion of it the reader is referred to [4].

The scheme (5.21) can be constructed by considering a Taylor expansion up to fourth order

$$U^{n+1} \approx U^n + \Delta t \frac{\partial U}{\partial t} + {}^1\!/\!_2 \Delta t^2 \frac{\partial^2 U}{\partial t^2} + {}^1\!/\!_6 \Delta t^2 \frac{\partial^3 U}{\partial t^3} + {}^1\!/\!_{24} \Delta t^4 \frac{\partial^4 U}{\partial t^4}$$

The following grouping defines (5.21):

$$U^{n+1} \approx U^n + \Delta t \frac{\partial}{\partial t}\left[U^n + {}^1\!/\!_2 \Delta t \frac{\partial}{\partial t}\left\{U^n + {}^1\!/\!_3 \Delta t \frac{\partial}{\partial t}\left(U^n + {}^1\!/\!_4 \Delta t \frac{\partial U}{\partial t}\right)\right\}\right]$$

The stability domain of the multi-stage time stepping is the same as that of the fourth-order Runge-Kutta scheme shown in Fig. 5.9.

The artificial viscosity introduced by Jameson is a blend of a second-order and a fourth-order term. It is used in all steps of (5.21).

In order to keep the calculation conservative, the added dissipative term is, for a structured quadrilateral grid:

$$d_{i+{}^1\!/\!_2,j} - d_{i-{}^1\!/\!_2,j} + d_{i,j+{}^1\!/\!_2} - d_{i,j-{}^1\!/\!_2}$$

where

$$d_{i+{}^1\!/\!_2,j} = \varepsilon^{(2)}_{i+{}^1\!/\!_2,j}(U_{i+1,j} - U_{i,j}) - \varepsilon^{(4)}_{i+{}^1\!/\!_2,j}(U_{i+2,j} - 3U_{i+1,j} + 3U_{i,j} - U_{i-1,j})$$

with similar definitions of the other terms in (5.22).

The coefficients of the second-order term ε(2) and the fourth-order term ε(4) are chosen in a self-adaptive way.

As a detector of the smoothness of the flow field, for the definition of the coefficients in (5.23), Jameson uses

$$\nu^i_{i,j} = \frac{|p_{i+1,j} - 2p_{i,j} + p_{i-1,j}|}{p_{i+1,j} + 2p_{i,j} + p_{i-1,j}}$$

and then defines

$$\varepsilon^{(2)}_{i+{}^1\!/\!_2,j} = \kappa^{(2)} \max(\nu^{i}_{i+1,j}, \nu^{i}_{i,j})$$

$$\varepsilon^{(4)}_{i+{}^1\!/\!_2,j} = \max(0, \kappa^{(4)} - \varepsilon^{(2)}_{i+{}^1\!/\!_2,j})$$

with $\kappa^{(2)} = {}^1/_4,\ \kappa^{(4)} = {}^1/_{256}$.

By this definition, the second-order term is only significant in shock regions. In smooth regions of the flow, the second-order term has a very small coefficient and the fourth-order term dominates. The fourth-order term constitutes the so-called background dissipation. For equal stabilization effect, it diffuses the solution less than a second-order term. Therefore it is used in smooth regions of the flow. In shock regions, the fourth-order dissipation has to be switched off since it causes wiggles and the second-order dissipation is to be used to eliminate wiggles. Therefore the second-order dissipation is called the shock dissipation.

At solid boundaries, the dissipative terms in (5.22) in the direction normal to the boundary are to be set equal to zero. In the foregoing definition of the dissipative terms (5.22, 5.23) the so-called second-order and fourth-order terms only correspond to second-order derivatives and fourth-order derivatives on a smooth grid.

However, the expressions (5.22, 5.23) do not have to be changed on an irregular grid. First, they are not meant to simulate a physical viscosity. Second, they are also meant to eliminate spurious modes, i.e. the non-physical solutions of the discretization. Figure 5.10 shows the perturbation patterns in fluxes, and as a consequence also in dependent variables, not detected by the central type flux balance for quadrilateral and triangular grids.

Authors using Jameson’s Runge-Kutta scheme often have their own variant of the dissipative term. Also very often, the dissipative correction in the second to fourth step is taken to be the same as in the first step.

A formulation of the artificial viscosity applicable to unstructured grids, which is a slight extension of the formulation given by Jameson and Mavriplis [5], is given hereafter.

The time-step limit is calculated from (for CFL = 1)

$$\Delta t = \frac{\Omega_i}{\sum_e (|V_n| + c)\Delta s}$$

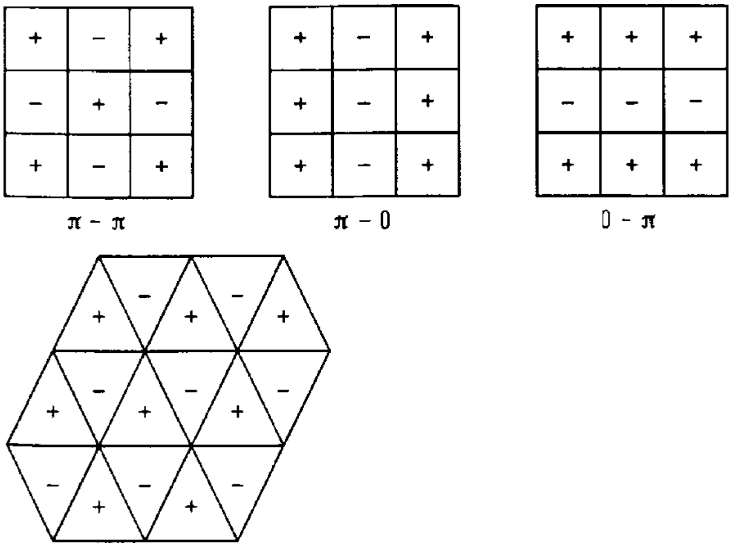


Fig. 5.10 Spurious modes for cell-centred central discretization

where the subscript i denotes the node, Vn is the normal velocity on an edge, obtained by averaging, c is the velocity of sound obtained in a similar way, and Δs is the length of the edge. Ωi is the volume and the summation is taken over all edges.

The second-order smoothing operator is then, similar to (5.23), obtained by a sum of terms:

$$\varepsilon^{(2)}_{i,j}\ \sigma_{i,j}(U_j - U_i)$$

where the subscript j denotes the surrounding nodes. The weight function εi,j is obtained from

$$\varepsilon^{(2)}_{i,j} = \kappa^{(2)} \max(\nu_i, \nu_j)$$

where νi and νj are pressure switches. The pressure switch νi is defined by

$$\nu_i = \frac{\mathrm{abs}\{\sum_j (p_j - p_i)\}}{\sum_j (p_j + p_i)}$$

$\sigma_{i,j}$ is a scaling factor given by

$$\sigma_{i,j} = \max\left(\frac{\Omega_i}{\Delta t}, \frac{\Omega_j}{\Delta t}\right)$$

with Δt the time step obtained from (5.24) for CFL = 1.

To define the fourth-order smoothing, first un-weighted pseudo-Laplacians are constructed by

$$\Delta U_i = \sum_j (U_j - U_i)$$

The fourth-order term is then given by a sum of terms:

$$\varepsilon_{i,j}^{(4)} \sigma_{i,j} (\Delta U_j - \Delta U_i)$$

Where

$$\varepsilon_{i,j}^{(4)} = \max(0, \kappa^{(4)} - \varepsilon_{i,j}^{(2)})$$

The scaling factors $\sigma_{i,j}$ allow the writing of the effective flux through a cell-face as

$$F_{i,j} - D_{i,j}$$

where Fi,j is the physical flux and Di,j is the dissipation term, given by

$$D_{i,j} = \sigma_{i,j}[\varepsilon_{i,j}^{(2)}(U_j - U_i) - \varepsilon_{i,j}^{(4)}(\Delta U_j - \Delta U_i)]$$

The resulting flux (5.26) usually is called a numerical flux.

5.6 Accuracy

The stencils obtained by the finite volume cell-centred formulation are very similar to the stencils obtained by the analogous finite difference methods. This means that if the grid is sufficiently smooth, such that the cell-centres are themselves on a grid which is sufficiently smooth, i.e. a grid which can be obtained by a continuous mapping from a square grid, the methods discussed in the previous sections are second-order accurate in space in a finite difference sense. This can easily be seen by comparison of the result in Fig. 5.8 with the result obtained by second-order finite difference methods [1]. Since, however, the representation of the solution is done in a piecewise constant way, on an irregular grid the accuracy is formally of first order. In practice, the order is between one and two.

5.7 Cell Vertex Formulation

In the cell-vertex formulation, the variables are stored at the vertices of the grid. The control volumes either coincide with cells (non-overlapping case) or consist of a group of cells around a node (overlapping case). Figure 5.5 shows some of the possibilities. In all cases, a linear interpolation of the fluxes is now possible. Therefore, cell-vertex formulations have the possibility to be second-order accurate in space, irrespective of the irregularity of the grid.

Multi-Stage Time Stepping – Overlapping Control Volumes

For the overlapping cases, the methods discussed in the previous sections can be adapted directly. Very popular nowadays is the formulation of the multi-stage time stepping scheme. For the overlapping control volumes of Fig. 5.5, the semi discretization is very similar to (5.17), now involving, however, six or eight surrounding nodes. At solid boundaries, half volumes are formed. The impermeability

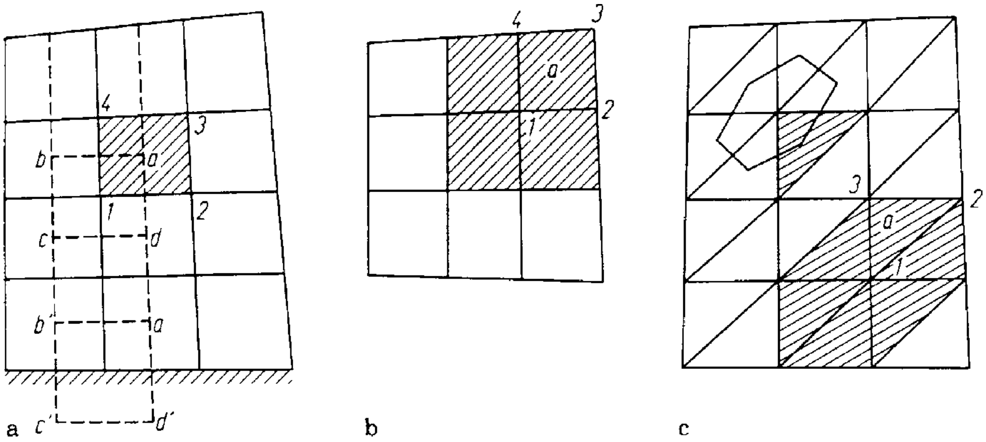


Fig. 5.5 Cell-vertex formulation.

(a): quadrilateral cells, non-overlapping volumes (with interweaving grid); (b): quadrilateral cells, overlapping volumes; (c): triangular cells, overlapping and non-overlapping volumes

can be expressed by setting the convective fluxes to zero. Another approach is to treat the control volume as permeable and to impose tangency. This means that, between steps, the normal component of velocity is set equal to zero.

Again, in order to stabilize the scheme, some form of artificial viscosity is necessary. The artificial viscosity is also necessary to eliminate the spurious modes in the solution. Figure 5.12 shows the spurious modes that are possible for the quadrilateral and triangular grids. As in the basic method of Jameson, a blend of a second-order smoothing and a fourth-order smoothing can be used. Often, the dissipative operator of the cellcentred method is used. This operator is then a sum of terms of form (5.23) for a quadrilateral grid. The method loses then its pure cell-vertex character. The resulting flux balance of inviscid and dissipative terms is then a balance over a control volume centred around a vertex as shown on Fig. 5.13. Such a control volume is called a dual control volume

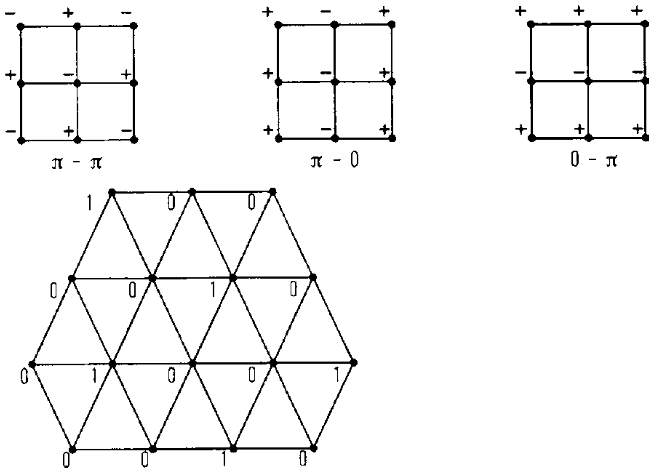


Fig. 5.12 Spurious modes for cell-vertex central discretization

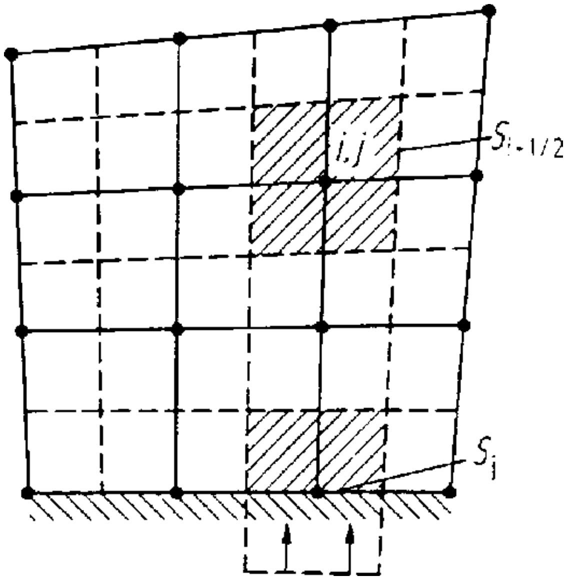


Fig. 5.13 Vertex-based

FVM

The inviscid flux balance over the dual control volume can be defined as one fourth of the flux balance over the volume formed by the four surrounding cells. Strictly, the method then becomes a vertex-centred or vertexbased method according to the terminology introduced in Sect. 5.3.

A pure cell-vertex method can be obtained by changing the construction of the dissipator. The same methodology as for the cell-centred method is used, but summations now run over cells surrounding a node rather than over surrounding nodes.

This means that differences of values used in the expression (5.27) have to be modified.

For instance $U_j - U_i$ is to be replaced by

$$\tfrac{1}{2}(U_{j1} + U_{j2}) - U_i$$

or

$$\tfrac{1}{3}(U_{j1} + U_{j2} + U_{j3}) - U_i \quad \text{or} \quad \tfrac{1}{2}(U_{j1} + U_{j3}) - U_i$$

for triangular and quadrilateral cells respectively, where j1, j2 and j3 denote the nodes not coinciding with node i of the surrounding cells. Also the scaling factors

$\sigma_{i,j}$ and the weight factors ε(2)

$_{i,j}$, ε(4)

$_{i,j}$ now involve maxima over all nodes of a cell.

The foregoing smoothing procedure is conservative in the sense that the content of a cell is not changed by the dissipator. The formula for the update of a node is the sum of contributions of the surrounding cells. The update coming from the inviscid flux balance over a cell is modified by the dissipator. The modification is such that the flux balance over a cell can be seen as distributed to its vertices in an unequal way but with a sum of distribution factors equal to one. So, the dissipator acts as a redistributor of the flux balances of the cells.

The pure cell-vertex method is not very often used. Most researchers employ the first described vertex-based like approach, but call it a cell-vertex method. The pure cell-vertex method has an obvious difficulty on a triangular grid. Since there are about twice as many cells than nodes, it is not possible to satisfy the flux balances of individual cells and reach steady state. Even on a structured grid, it is rather delicate to satisfy flux balances over individual cells. For a discussion on this topic the reader is referred to [6].

Lax-Wendroff Time-Stepping Non-Overlapping Control Volumes

For the non-overlapping case, a Lax-Wendroff variant exists due to Ni, developed in 1981 [7]. It requires the use of a second set of control volumes centred around the nodes, obtained in the way as shown in Fig. 5.5a. Ni's variant starts from the Lax-Wendroff formulation (5.8), (5.9). Without loss of accuracy in (5.9), ∂f/∂t can be replaced by a first-order accurate difference Δf/Δt. The result is

$$u^{n+1} \approx u^n - \Delta t \frac{\partial f^n}{\partial x} - \frac{\Delta t}{2} \frac{\partial}{\partial x}(\Delta f)$$

In two dimensions, on the Euler equations, this is

$$\Omega_{i,j}(U^{n+1} - U^n) = -\Delta t \left[\int (f^n\, dy - g^n\, dx) + {}^1\!/\!_2 \int (\Delta f\, dy - \Delta g\, dx) \right]$$

On the quadrilateral grid of Fig. 5a, the method is then as follows. Based on the cell 1-2-3-4, using an Euler step, i.e. a step forward in time, a first order approximation of the increment of the flux vectors is obtained from

$$\Omega_a \Delta U_a = -\Delta t \int_{1234} (f\, dy - g\, dx)^n$$

And

$$\Delta f_a = A\, \Delta U_a, \qquad \Delta g_a = B\, \Delta U_a$$

where A and B are the Jacobians of the flux vectors f and g with respect to U. A

and B are taken to be the mean values of the Jacobians evaluated at the nodes 1, 2, 3 and 4.
The area-weighted mean value of the first-order increments given by (5.29) over the four cells surrounding the node 1, gives a first-order increment for the dependent variables:

$$\Delta U_1^1$$

The discretization of (5.28) on the cell abcd is then:

$$\Omega_1(U_1^{n+1} - U_1^n) = \Omega_1\ \Delta U_1^1 - {}^1\!/\!_2\ \Delta t \int\limits_{abcd} (\Delta f\ dy - \Delta g\ dx)$$

The spatial integration is again taken to be piecewise linear.
The CFL-restriction for the time step, given by Ni is

$$\Delta t \le \min\left(\frac{\Delta x}{|u| + c}, \quad \frac{\Delta y}{|u| + c}\right)$$

with

$$\Delta x = \frac{x_{i+1,j} - x_{i-1,j}}{2} \quad , \qquad \Delta y = \frac{y_{i,j+1} - y_{i,j-1}}{2}$$

The boundary conditions at solid boundaries for the first step (5.29) can be implemented by setting convective fluxes equal to zero, as in the previous methods. In the second step (5.30), a half-volume is needed around a boundary node. This half volume can be seen to be half the complete volume shown in Fig. 5.5a. Step (5.30) can be done by setting the first-order changes in the fictitious cells c’and d’equal to zero. So the boundary node only receives both first-order and second-order contributions from the inward cells a’and b’As a consequence, for a boundary node, there is no implicit imposition of impermeability in step (5.30). Tangency is then imposed afterwards by setting the normal component of the velocity equal to zero.
It is to be remarked that, although an intermediate grid is used, the Ni-method is a true cell-vertex method. Indeed, if the flux balance of a cell is satisfied, there is no contribution to both first- and second-order terms and flow parameters are not changed. Therefore step (5.30) often is called the distribution step since its function can be seen to be the distribution of changes in the control volumes to the nodes.
As already mentioned, in a triangular grid, there are about twice as many cells as nodes. This means that in a cell-vertex formulation, flux-balances cannot be satisfied for all cells. The steady

state result of a cell-vertex time stepping scheme then corresponds to some combinations of flux balances being zero. In a quadrilateral grid, all flux-balances can be satisfied at steady state. We also note that the distribution of the changes in the control volumes for triangular cells can be done with upwind methods. For a discussion on these much more complex methods we refer to [8].

5.8 Finite Difference Method

FDM-Like Finite Volume Methods

In the finite difference method, the nodes are at the vertices of the grid. This is particularly attractive with respect to data on boundaries. For instance, pressure extrapolation at solid boundaries is then not necessary. A cell-centred FVM is therefore less attractive. A cell-vertex FVM does not have this drawback, but on the other hand the flux through a volume surface is continuous. This does not allow an upwind definition of a flux.

More freedom in the definition of a flux, combined with nodes at the vertices of the grid, can be obtained by using an interweaving grid, as shown in Fig. 5.13. The interweaving grid can be constructed by connecting the cell-centres. The cells of this interweaving grid can now be considered as control volumes for the nodes inside them. Fluxes at volume faces can, for instance, be defined as averages of fluxes calculated with function values in adjacent nodes. The semi-discretization is then very close to a finite difference semi-discretization and can be called a conservative finite difference method. We prefer here to call a finite volume method of this type a vertex-based FVM or a vertex-centred FVM. The method has gained much popularity in recent years. The central type discretization obtained with it is the same as the discretization by a Galerkin-FEM. So it is very easy to bring concepts from FEM into this type of FVM. Moreover, it is very easy to use upwinding in this type of FVM.

5.9 Central and Upwind Type Discretizations

Central Type Discretizations

The adaptation of a Lax-Wendroff time-stepping or a multi-stage time-stepping, as discussed for the cell-centred FVM, to the vertex-based FVM is straightforward. The formulations obtained with both methods are very similar, except at solid boundaries.

Upwind Type Discretizations

As an example of an upwind discretization we treat here the flux-difference splitting technique introduced by Roe [9].

The flux through a surface (i + 1/2) of the control volume on Fig. 5.13 can be written as

$$F_{i+1/2} = \Delta y_{i+1/2} f_{i+1/2} - \Delta x_{i+1/2} g_{i+1/2}$$

where $f_{i+1/2}$ and $g_{i+1/2}$ have to be defined using the values of the flux vectors in the nodes (i,j) and (i+1, j).We switch here to the classic finite difference notation using halves in the subscripts to denote intermediate points. Also, non-varying subscripts are not written. We denote by Fi the value of $F_{i+1/2}$ using the function values in (i,j) and by F_{i+1}, the value using the function values in (i+1, j). The flux (5.31) can be written as

$$F_{i+1/2} = \Delta s_{i+1/2}(n_x f_{i+1/2} + n_y g_{i+1/2})$$

with

$$n_x = \Delta y_{i+1/2}/\Delta s_{i+1/2}, \qquad n_y = -\Delta x_{i+1/2}/\Delta s_{i+1/2}, \qquad \Delta s^2_{i+1/2} = \Delta x^2_{i+1/2} + \Delta y^2_{i+1/2}$$

In order to define an upwind flux, we consider the flux-difference

$$\Delta F_{i,i+1} = \Delta s_{i+1/2}(n_x \Delta f_{i,i+1} + n_y \Delta g_{i,i+1})$$

where

$$\Delta f_{i,i+1} = f_{i+1,j} - f_{i,j}, \qquad \Delta g_{i,i+1} = g_{i+1,j} - g_{i,j}$$

For construction of the flux, it is essential that the linear combination of Δf and Δg in (5.33) can be written as

$$\Delta\phi = n_x \Delta f + n_y \Delta g = A \Delta U$$

where A is a discrete Jacobian matrix with similar properties as the analytical Jacobians of the flux vectors. This means that the eigenvalues of A are real and that the matrix has a complete set of eigenvectors. Of course, for consistency, the eigenvalues and eigenvectors should be approximations of the eigenvalues and eigenvectors of the linear combination of the analytical Jacobians. The construction of the discrete Jacobian is not unique and many formulations have been proposed after the first formulation by Roe [9]. For the numerical illustration later in this section, we use the formulation by the author [10]. The algebraic manipulations in the construction of the discrete Jacobian are not relevant for a principal discussion of the methodology and we do not describe these here.

The matrix A can be split into positive and negative parts by

$$A^+ = R\Lambda^+ L, \qquad A^- = R\Lambda^- L$$

where R and L denote the right and left eigenvector matrices in orthonormal form and where

$$\Lambda^+ = \mathrm{diag}(\lambda_1^+, \lambda_2^+, \lambda_3^+, \lambda_4^+), \qquad \Lambda^- = \mathrm{diag}(\lambda_1^-, \lambda_2^-, \lambda_3^-, \lambda_4^-)$$

$$\text{with } \lambda_i^+ = \max(\lambda_i, 0),\ \lambda_i^- = \min(\lambda_i, 0).$$

Positive and negative matrices denote matrices with, respectively, non-negative and non-positive eigenvalues.

This allows a splitting of the flux-difference (5.34) by

$$\Delta\phi = A^+\Delta U + A^-\Delta U$$

As a consequence (5.33) can be written as

$$\Delta F_{i,i+1} = F_{i+1} - F_i = \Delta s_{i+1/2} A_{i,i+1} \Delta U_{i,i+1}$$

where the matrix $A_{i,i+1}$ can be split into positive and negative parts. The absolute value of the flux-difference is defined by

$$\left|\Delta F_{i,i+1}\right| = \Delta s_{i+1/2}(A_{i,i+1}^+ - A_{i,i+1}^-)\Delta U_{i,i+1}$$

Based on (5.36) an upwind definition of the flux is

$$F_{i+1/2} = 1/2\left[F_i + F_{i+1} - \left|\Delta F_{i,i+1}\right|\right]$$

That this represents an upwind flux can be verified by writing (5.37) in either of the two following ways, which are completely equivalent:

$$F_{i+1/2} = F_i + 1/2\Delta F_{i,i+1} - 1/2\left|\Delta F_{i,i+1}\right| = F_i + \Delta s_{i+1/2} A_{i,i+1}^- \Delta U_{i,i+1}$$

$$F_{i+1/2} = F_{i+1} - 1/2\Delta F_{i,i+1} - 1/2\left|\Delta F_{i,i+1}\right| = F_{i+1} - \Delta s_{i+1/2}A^{+}_{i,i+1}\Delta U_{i,i+1}$$

Indeed, when $A_{i,i+1}$ has only positive eigenvalues, the flux $F_{i+1/2}$ is taken to be Fi and when $A_{i,i+1}$ has only negative eigenvalues, the flux $F_{i+1/2}$ is taken to be F_{i+1}.

The fluxes on the other surfaces of the control volume $S_{i-1/2}$,$S_{j+1/2}$,$S_{j-1/2}$ can be treated in a similar way as the flux on the surface S $_{i+1/2}$. With (5.38) and (5.39), the flux balance on the control volume of Fig. 5.13 can be brought into the form

$$\begin{aligned}\Delta s_{i+1/2}A^{-}_{i,i+1}[U_{i+1} - U_{i}] + \Delta s_{i-1/2}A^{+}_{i,i-1}[U_{i} - U_{i-1}]\\ + \Delta s_{j+1/2}A^{-}_{j,j+1}[U_{j+1} - U_{j}] + \Delta s_{j-1/2}A^{+}_{j,j-1}[U_{j} - U_{j-1}] = 0\end{aligned}$$

or

$$\begin{aligned}CU_{i,j} = \Delta s_{i-1/2}A^{+}_{i,i-1}U_{i-1,j} + \Delta s_{i+1/2}(-A^{-}_{i,i+1})U_{i+1,j}\\ + \Delta s_{j-1/2}A^{+}_{j,j-1}U_{i,j-1} + \Delta s_{j+1/2}(-A^{-}_{j,j+1})U_{i,j+1}\end{aligned}$$

where C is the sum of the matrix-coefficients on the right-hand side. The matrix coefficients in (5.41) have non-negative eigenvalues. The positivity of the coefficients on the right hand side of (5.41) and the (weak) dominance of the central coefficient guarantee that the solution can be obtained by a collective variant of any scalar relaxation method. By a collective variant is meant that in each node all components of the vector of dependent variables U are relaxed simultaneously.

In order to illustrate the boundary treatment, we consider now the half-volume on a solid boundary as shown in Fig. 5.13. This half-volume can be seen as the limit of a complete volume in which one of the sides tends to the boundary.

The flux on the side Sj of the control volume at the solid boundary can be expressed By

$$F_{j} - \Delta s_{j}A^{+}_{i,j}(U_{j} - U_{j-1})$$

where the matrix Ai,j is calculated with the function values in the node (i,j). With the definition (5.42), the flux balance on the control volume takes the form (5.40) in which a node outside the domain comes in. This node, however, can be eliminated.

It is easily seen that on a solid boundary, three combinations of (5.42) exist, eliminating the outside node [10]. The combinations are the left eigenvectors corresponding to the zero eigenvalues in A+i,j. These equations are to be supplemented by the boundary condition of tangency.

As an illustration, Fig. 5.14 shows the solution obtained by the previous method for the test-case of Fig. 5.7 under the same conditions as for Fig. 5.8. Comparison of the upwind result with the central result shows the superiority of the upwind calculation with respect to sharpness of the shock.

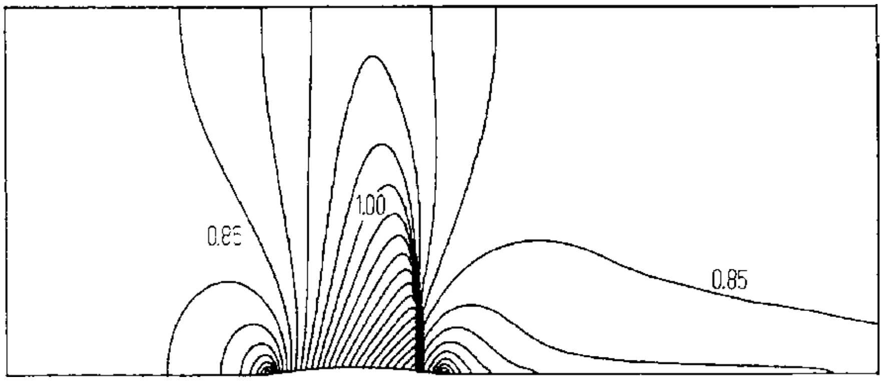


Fig. 5.14 IsoMachlines obtained by a vertex-based upwind FVM

In the above, the upwind discretization is used in first-order form. For more complex flows, of course, at least second-order accuracy is needed. In this introductory text we prefer not to enter the discussion of higher order upwinding. For second order formulations on unstructured grids, the reader is referred to [5].

Examples of vertex-centred methods for Euler and Navier-Stokes equations can be found in [12]. Flux-difference splitting is used to define inviscid fluxes. The paper is in particular interesting for its discussion on treatment of viscous fluxes. An example of a vertex-centred method with central discretization of the inviscid fluxes and stabilization by artificial viscosity can be found in [13]. In this paper, viscous fluxes are treated by FEM. This becomes nowadays a widely accepted procedure and can be recommended. The vertex-centred FVM can be combined easily with a Galerkin-type FEM. References [5] and [13] use multigrid methods in order to obtain a steady solution in a fast way. The multigrid method is nowadays a standard method to accelerate the convergence to steady state. For a general discussion on the choice between central and upwind finite volume methods, the reader is referred to [14]. In [15] a general discussion on the choice between cell-centred and vertex-centred methods and the choice between central and upwind methods is given. An interesting example of a cell-centred method using upwinding is given in [16]. Reference [17] discusses different time stepping algorithms for upwind methods both for vertex-centred and cell-centred formulations.

Finally, the reader is referred to [18] for an overview of current finite volume methods. This reference dates from more than a decade ago, but there have not been major developments on basic algorithms in recent times.

5.10 Treatment of Derivatives

When derivatives are needed for the definition of viscous terms, these commonly are calculated by the use of Gauss' theorem. For instance for the cell-centred formulation shown in Fig. 5.15, in order to define a derivative in the vertex a, an integration over the shaded volume gives

$$\left(\frac{\partial\phi}{\partial x}\right)_a \approx \frac{1}{\Omega_a}\int_{\Omega_a}\frac{\partial\phi}{\partial x}dxdy = \frac{1}{\Omega_a}\int_{S_a}\phi\ dy$$

Thus

$$\left(\frac{\partial\phi}{\partial x}\right)_a \approx \frac{1}{\Omega_a}\Bigg[\phi_{i+1,j+1}\frac{y_{i,j+1}-y_{i+1,j}}{2}+\phi_{i,j+1}\frac{y_{i,j}-y_{i+1,j+1}}{2}$$
$$+\phi_{i,j}\frac{y_{i+1,j}-y_{i,j+1}}{2}+\phi_{i+1,j}\frac{y_{i+1,j+1}-y_{i,j}}{2}\Bigg]$$

with

$$\Omega_a \approx \frac{y_{i+1,j+1}-y_{i,j}}{2}(x_{i+1,j}-x_{i,j+1})+\frac{y_{i,j+1}-y_{i+1,j}}{2}(x_{i+1,j+1}-x_{i,j})$$

A similar procedure can be used for the other vertices of the cell abcd. This allows a definition of the viscous terms on the boundary of the cell.

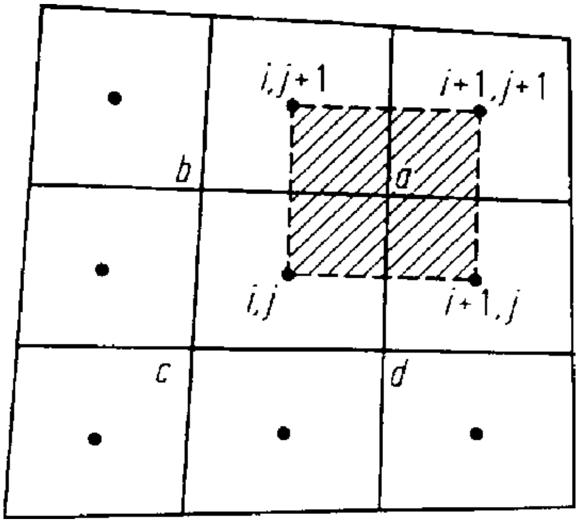


Fig. 5.15 Definition of a derivative

17.2.3 SIMPLE Formulations

This family of algorithms is based on a finite volume (Sect. 5.2) discretisation on a staggered grid (Sect. 17.1.1) of the governing equations. The method was introduced by Patankar and Spalding (1972) and is described in detail by Patankar (1980). The acronym, SIMPLE, stands for Semi-Implicit Method for Pressure-Linked Equations and describes the iterative procedure by which the solution to the discretised equations is obtained. The iterative procedure will be interpreted as a pseudotransient treatment of the unsteady governing equations (17.1–3) in discrete form to obtain the steady-state solution. An important link with the auxiliary potential function method (Sect. 17.2.2) will be indicated.

On a staggered grid different control volumes are used, Fig. 17.9, to discretise different equations. In addition the grid notation associated with particular dependent variables is staggered, Fig. 17.9. Thus the physical locations of $p_{j+1/2,k}$ and $u_{j,k}$

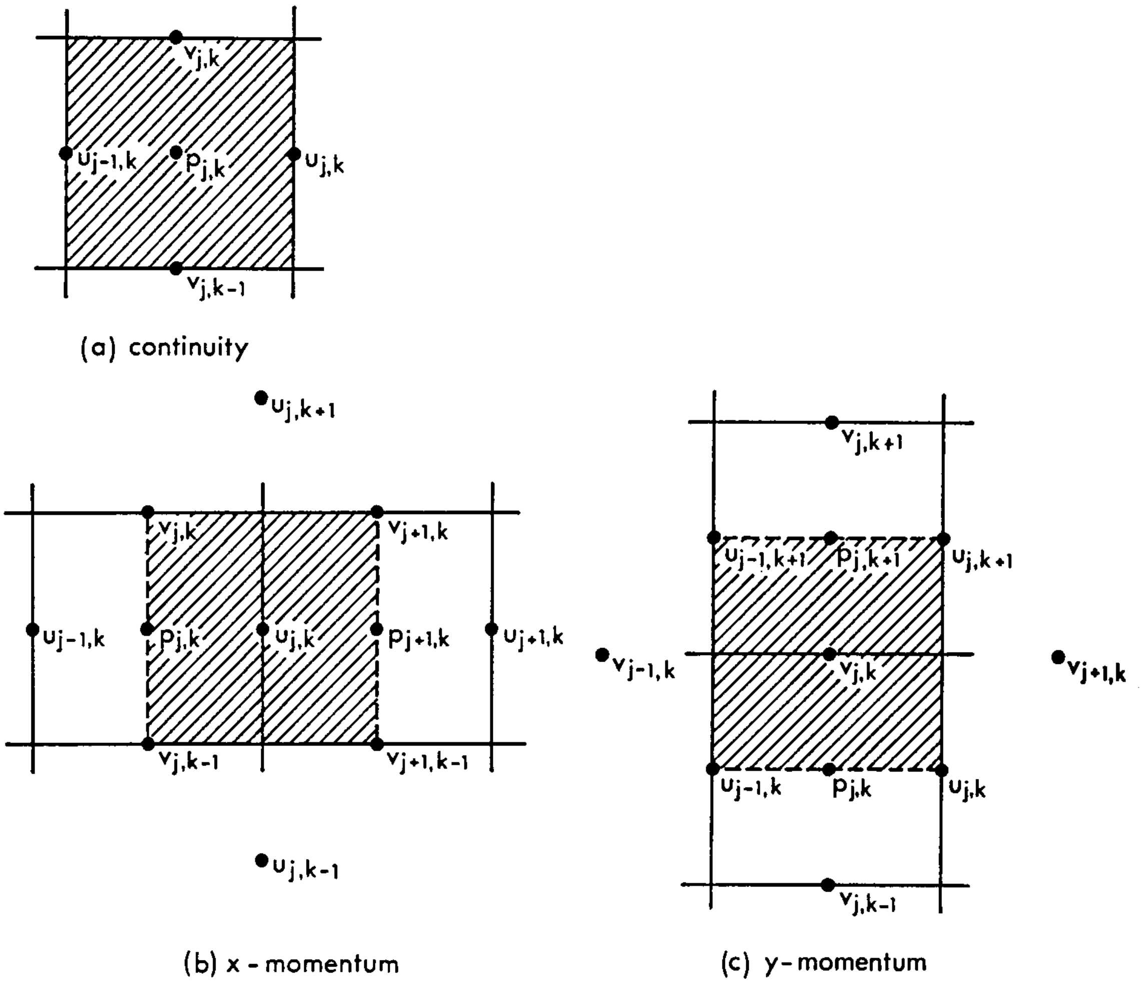


Fig. 17.9. Control volumes used in SIMPLE formulation

are the same, as are the physical locations of $p_{j,k+1/2}$ and $v_{j,k}$. The discretisation indicated below corresponds to a uniform grid. The more general case of a nonuniform grid can be obtained from Sect. 5.2 or Patankar (1980).

For the control volume shown in Fig. 17.9a the application of the finite volume method (Sect. 5.2) to the continuity equation (17.1) produces the discrete equation

$$(u^{n+1}_{j,k} - u^{n+1}_{j-1,k})\Delta y + (v^{n+1}_{j,k} - v^{n+1}_{j,k-1})\Delta x = 0 \ . \tag{17.67}$$

Application of the finite volume method to the x-momentum equation (17.2) using the control volume shown in Fig. 17.9b leads to the discrete equation

$$\left(\frac{\Delta x \Delta y}{\Delta t}\right)(u^{n+1}_{j,k} - u^{n}_{j,k}) + (F^{(1)}_{j+1/2,k} - F^{(1)}_{j-1/2,k})\Delta y + (G^{(1)}_{j,k+1/2} - G^{(1)}_{j,k-1/2})\Delta x$$
$$+ (p^{n+1}_{j+1,k} - p^{n+1}_{j,k})\Delta y = 0 \ , \tag{17.68}$$

where

$$F^{(1)}=u^2-\frac{1}{\text{Re}}\frac{\partial u}{\partial x}\quad\text{and}\quad G^{(1)}=uv-\frac{1}{\text{Re}}\frac{\partial u}{\partial y}\ .$$

Thus

$$F^{(1)}_{j+1/2,k}=0.25(u_{j,k}+u_{j+1,k})^2-\frac{1}{\text{Re}}\frac{u_{j+1,k}-u_{j,k}}{\Delta x}\quad\text{and}$$

$$G^{(1)}_{j,k+1/2}=0.25(v_{j,k}+v_{j+1,k})(u_{j,k}+u_{j,k+1})-\frac{1}{\text{Re}}\frac{u_{j,k+1}-u_{j,k}}{\Delta y}\ .$$

Consequently (17.68) can be written as

$$\left(\frac{\Delta x\Delta y}{\Delta t}+a^u_{j,k}\right)u^{n+1}_{j,k}+\sum a^u_{nb}u^{n+1}_{nb}+b^u+\Delta y(p^{n+1}_{j+1,k}-p^{n+1}_{j,k})=0\ , \tag{17.69}$$

where $\sum a^u_{nb}u^{n+1}_{nb}$ denotes all the convection and diffusion contributions from neighbouring nodes. The coefficients $a^u_{j,k}$ and a^u_{nb} depend on the grid sizes and the solution u, v at the nth time level. The term $b^u=-\Delta x\Delta y\ u^n_{j,k}/\Delta t$. It may be noted that some terms in $F^{(1)}$ and $G^{(1)}$ have been evaluated at the nth time-level to ensure that (17.69) is linear in u^{n+1}.

Using the control volume shown in Fig. 17.9c the discretised form of the y-momentum equation (17.3) can be written

$$\left(\frac{\Delta x\Delta y}{\Delta t}\right)(v^{n+1}_{j,k}-v^n_{j,k})+(F^{(2)}_{j+1/2,k}-F^{(2)}_{j-1/2,k})\Delta y+(G^{(2)}_{j,k+1/2}-G^{(2)}_{j,k-1/2})\Delta x$$
$$+(p^{n+1}_{j,k+1}-p^{n+1}_{j,k})\Delta x=0\ , \tag{17.70}$$

where

$$F^{(2)}=uv-\frac{1}{\text{Re}}\frac{\partial v}{\partial x}\quad\text{and}\quad G^{(2)}=v^2-\frac{1}{\text{Re}}\frac{\partial v}{\partial y}\ .$$

Substituting for $F^{(2)}$ and $G^{(2)}$ allows (17.70) to be written

$$\left(\frac{\Delta x\Delta y}{\Delta t}+a^v_{j,k}\right)v^{n+1}_{j,k}+\sum a^v_{nb}v^{n+1}_{nb}+b^v+\Delta x(p^{n+1}_{j,k+1}-p^{n+1}_{j,k})=0\ , \tag{17.71}$$

where the various coefficients have a similar interpretation to that indicated for (17.69).

At any intermediate stage of the SIMPLE iterative procedure the solution is to be advanced from the nth time level to the $(n+1)$-th time level. The velocity solution is advanced in two stages. First the momentum equations (17.69 and 71) are solved to obtain an approximation, $\mathbf{u}^*$, of $\mathbf{u}^{n+1}$ that does not satisfy continuity.

where

$$F^{(1)}=u^2-\frac{1}{\mathrm{Re}}\frac{\partial u}{\partial x}\quad\text{and}\quad G^{(1)}=uv-\frac{1}{\mathrm{Re}}\frac{\partial u}{\partial y}\ .$$

Thus

$$F^{(1)}_{j+1/2,k}=0.25(u_{j,k}+u_{j+1,k})^2-\frac{1}{\mathrm{Re}}\frac{u_{j+1,k}-u_{j,k}}{\Delta x}\quad\text{and}$$

$$G^{(1)}_{j,k+1/2}=0.25(v_{j,k}+v_{j+1,k})(u_{j,k}+u_{j,k+1})-\frac{1}{\mathrm{Re}}\frac{u_{j,k+1}-u_{j,k}}{\Delta y}\ .$$

Consequently (17.68) can be written as

$$\left(\frac{\Delta x\Delta y}{\Delta t}+a^u_{j,k}\right)u^{n+1}_{j,k}+\sum a^u_{nb}u^{n+1}_{nb}+b^u+\Delta y(p^{n+1}_{j+1,k}-p^{n+1}_{j,k})=0\ ,\tag{17.69}$$

where $\sum a^u_{nb}u^{n+1}_{nb}$ denotes all the convection and diffusion contributions from neighbouring nodes. The coefficients $a^u_{j,k}$ and a^u_{nb} depend on the grid sizes and the solution u, v at the nth time level. The term $b^u=-\Delta x\Delta y\ u^n_{j,k}/\Delta t$. It may be noted that some terms in $F^{(1)}$ and $G^{(1)}$ have been evaluated at the nth time-level to ensure that (17.69) is linear in u^{n+1}.

Using the control volume shown in Fig. 17.9c the discretised form of the y-momentum equation (17.3) can be written

$$\left(\frac{\Delta x\Delta y}{\Delta t}\right)(v^{n+1}_{j,k}-v^n_{j,k})+(F^{(2)}_{j+1/2,k}-F^{(2)}_{j-1/2,k})\Delta y+(G^{(2)}_{j,k+1/2}-G^{(2)}_{j,k-1/2})\Delta x$$
$$+(p^{n+1}_{j,k+1}-p^{n+1}_{j,k})\Delta x=0\ ,\tag{17.70}$$

where

$$F^{(2)}=uv-\frac{1}{\mathrm{Re}}\frac{\partial v}{\partial x}\quad\text{and}\quad G^{(2)}=v^2-\frac{1}{\mathrm{Re}}\frac{\partial v}{\partial y}\ .$$

Substituting for $F^{(2)}$ and $G^{(2)}$ allows (17.70) to be written

$$\left(\frac{\Delta x\Delta y}{\Delta t}+a^v_{j,k}\right)v^{n+1}_{j,k}+\sum a^v_{nb}v^{n+1}_{nb}+b^v+\Delta x(p^{n+1}_{j,k+1}-p^{n+1}_{j,k})=0\ ,\tag{17.71}$$

where the various coefficients have a similar interpretation to that indicated for (17.69).

At any intermediate stage of the SIMPLE iterative procedure the solution is to be advanced from the nth time level to the $(n+1)$-th time level. The velocity solution is advanced in two stages. First the momentum equations (17.69 and 71) are solved to obtain an approximation, $\mathbf{u}^*$, of $\mathbf{u}^{n+1}$ that does not satisfy continuity.

Comparing (17.79) and (17.62) indicates that δp is an effective velocity potential and the velocity correction, $\mathbf{u}^c$, is irrotational. The complete SIMPLE algorithm can be summarised as follows:

1) $\mathbf{u}^*$ is obtained from (17.72 and 73),
2) δp is obtained from (17.77),
3) $\mathbf{u}^c$ is obtained from (17.75) and equivalent form for v^c,
4) p^{n+1} is obtained from $p^{n+1}=p^n+\alpha_p\delta p$, where α_p is a relaxation parameter.

The SIMPLE algorithm contains two relaxation parameters α_p and $E(\equiv\Delta t)$. Solving the steady momentum equation is equivalent to setting $E=\infty$. In this case it is recommended that $\alpha_p=0.075$ to achieve a stable convergence. A more rapid convergence is found, empirically, if $E=1$ and $\alpha_p=0.8$ (Patankar 1980).

Raithby and Schneider (1979) have made a systematic study of SIMPLE-type algorithms and conclude that a more efficient algorithm is obtained if $E\approx 4$ and

$$\alpha_p=\frac{1}{1+E} \quad . \tag{17.80}$$

Van Doormaal and Raithby (1984) call (17.80) a consistent SIMPLE algorithm, or SIMPLEC as an acronym. However, Van Doormaal and Raithby give an alternative interpretation of SIMPLEC. It is argued that the approximation inherent in passing from (17.74) to (17.75) causes an increase in the number of iterations to convergence, although it does improve the economy of each iteration.

A closer approximation to (17.74) is obtained by subtracting $\Sigma\, a^u_{nb}u^c_{j,k}$ from both sides and dropping $\Sigma\, a^u_{nb}(u^c_{nb}-u^c_{j,k})$ from the right-hand side to give

$$u^c_{j,k}=d'_{j,k}(\delta p_{j,k}-\delta p_{j+1,k}) \quad , \tag{17.81}$$

in place of (17.75), where

$$d'_{j,k}=E\Delta y/[(1+E)a^u_{j,k}-E\,\Sigma\, a^u_{nb}] \quad .$$

If the correction u^c is slowly varying in space only a small error is introduced in dropping $\Sigma\, a^u_{nb}(u^c_{nb}-u^c_{j,k})$, but (17.81) retains the significant economy of being explicit. In forming the Poisson equation for δp, (17.81) is used instead of (17.75) and similarly when computing $u^c_{j,k}$. However, when obtaining p^{n+1}, as in step 4) of the SIMPLE algorithm, no underrelaxation is required, i.e. $\alpha_p=1$, if the SIMPLEC option (17.81) is introduced. A conceptually similar modification to SIMPLE is discussed by Connell and Stow (1986).

Application of the original SIMPLE algorithm to a range of problems suggests that δp is an effective mechanism for adjusting the velocity field but is often less effective in rapidly converging the pressure field. Patankar (1980) introduced a revised algorithm, SIMPLER, to improve the situation. The SIMPLER algorithm consists of the following steps:

1) A velocity field $\hat{\mathbf{u}}$ is computed from (17.72 and 73) with the pressure terms deleted from the right-hand sides.

2) Equation (17.77) then becomes a Poisson equation for p^{n+1} rather than δp with $\hat{\mathbf{u}}$ replacing the $\mathbf{u}^*$ terms in b.
3) The p^{n+1} (solution from step 2) replaces p^n in (17.72 and 73), which are solved as in SIMPLE to give $\mathbf{u}^*$.
4) Equation (17.77) is solved for δp and it is used to provide $\mathbf{u}^{n+1} = \mathbf{u}^* + \mathbf{u}^c$ but *no* further adjustment is made to p^{n+1} from step 2.

Clearly SIMPLER involves solving two Poisson steps and two momentum steps per iteration cycle. Although more expensive per iteration than SIMPLE, convergence is reached in sufficiently few iterations that SIMPLER is typically 50% more efficient. It may be noted that steps 1) and 2) of SIMPLER correspond to the projection method (17.22 and 24).

Van Doormaal and Raithby (1984) have compared SIMPLE, SIMPLEC and SIMPLER for a recirculating flow and flow over a backward-facing step. In obtaining solutions, (17.77) is repeated v times at each iteration until $\|r_p\|^v \leqq \gamma_p \|r_p\|^0$, where $\|r_p\|$ is the rms residual of (17.77), i.e.

$$r_p = \sum a_{nb}^p \, \delta p_{nb} + b^p - a_{j,k}^p \delta p_{j,k} \quad \text{and} \quad \|r_p\| = \left[\sum_j \sum_k r_p^2 \right]^{1/2} .$$

Optimal values of γ_p range from 0.05 to 0.25. A comparison of the computational effort (CPU-secs) to reach convergence is shown in Fig. 17.10. Clearly both SIMPLEC and SIMPLER are more efficient than SIMPLE, with SIMPLEC to be preferred. However, the optimal choice of E, and to a lesser extent γ_p, is problem dependent.

SIMPLE-type algorithms on staggered grids have also been used with generalised (body-fitted) coordinates (Chap. 12). Raithby et al. (1986) have used the

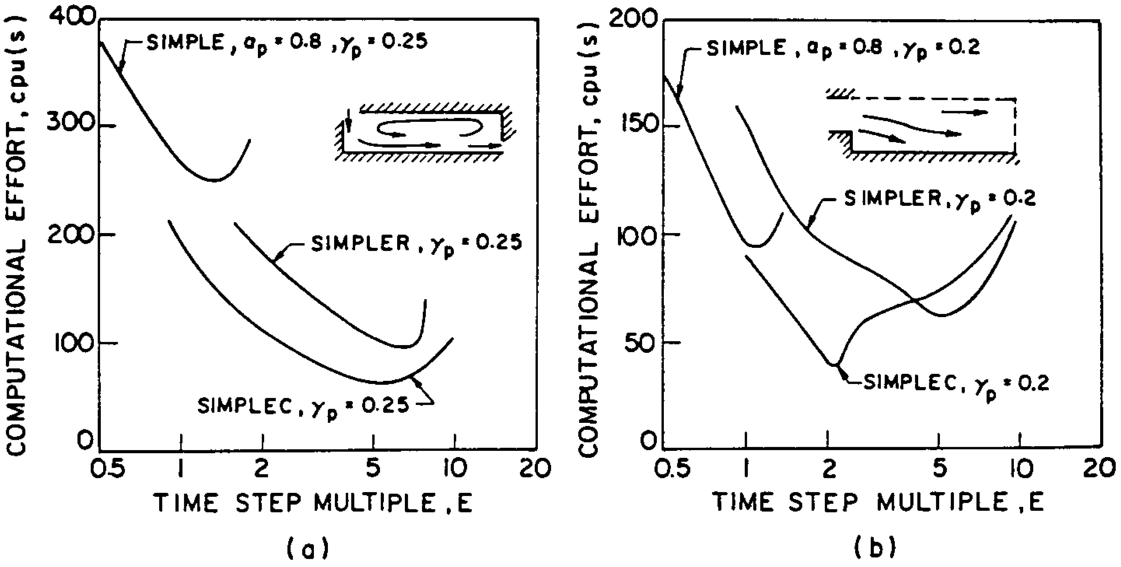


Fig. 17.10a, b. Comparison of SIMPLE, SIMPLEC and SIMPLER (after van Doormaal and Raithby, 1984; reprinted with permission of Hemisphere Publishing Co.)

SIMPLEC algorithm with orthogonal generalised coordinates. It is found that formulating and discretising the problem at the stress level, as in (11.26) produces a more efficient code. Appropriate laminar or turbulent stress strain relations are introduced subsequently in an appropriate discrete form. However, if first-order upwind discretisations are introduced at this stage, the overall solution accuracy is often reduced. If higher-order discretisations are used more grid points are coupled and the solution algorithm is less economical.

Shyy et al. (1985) have applied the SIMPLE algorithm on a staggered grid with non-orthogonal generalised coordinates. The QUICK differencing scheme (Sect. 17.1.5) and a three-point second-order upwind scheme [$q=1.5$ in (9.53)] for the convective terms are compared. Two-dimensional turbulent flow in a kidney-shaped channel provides the test problem and 31 × 26 and 56 × 36 grids are used. Although this problem has no exact solution the second-order upwind scheme is generally preferred as it is more robust and more efficient without producing obviously less accurate solutions. The QUICK scheme [$q=0.375$ in (9.53)] is divergent on a very distorted mesh and generally requires more iterations when it is convergent.

The lack of robustness of the QUICK scheme has also been reported by Pollard and Siu (1982) and Patel and Markatos (1986) in applying the SIMPLE algorithm on a Cartesian grid. In relation to (9.53) it can be seen that reducing q produces a scheme closer to the three-point centred difference formula ($q=0$). Thus the lack of robustness with the QUICK scheme ($q=0.375$) compared with the second-order upwind scheme ($q=1.5$) is not unexpected.

Phillips and Schmidt (1985) have combined a SIMPLE algorithm on a staggered grid with QUICK differencing of the convective terms. A multigrid procedure (Sect. 6.3.5) is employed to accelerate the convergence to the steady state. Phillips and Schmidt consider the driven cavity problem at $\mathrm{Re}=400$ and natural convection in a vertical cavity (de Vahl Davis and Jones 1983) at $\mathrm{Ra}=10^6$. The multigrid procedure is used with different grid refinements in different parts of the domain. Typically the finest grids ($h=1/32$) are introduced adjacent to the walls but a less fine grid ($h=1/16$) is employed in the interior. The coarsest grid ($h=1/4$) in the multigrid procedure is used throughout the domain.

17.3.4 Pressure Solution

In the stream function vorticity formulation the pressure does not appear explicitly. However, once the velocity solution is available the pressure solution can be obtained without difficulty. Techniques will be discussed here for steady flow; the extension for unsteady flow is straightforward.

The most direct means of computing the pressure is to treat the momentum equations (17.2 and 3) as ordinary differential equations in $\dot{p}$. This technique is reasonably effective close to regions of known pressure, e.g. the freestream, and if the spatial pressure gradients are not large. However, the errors in the velocity field accumulate so that a long integration may imply a significant error. In addition if the pressure at a particular point is obtained by integrating along different paths some means of averaging or smoothing will need to be introduced to avoid a multivalued pressure solution. Such a technique is described by Raithby and Schneider (1979) as a modification to the SIMPLE algorithm (Sect. 17.2.3). For the flow over a backward-facing step Fletcher and Srinivas (1983) have used parallel integration of the momentum equations and normal extrapolation to obtain the pressure at the surface.

To obtain the pressure in the interior it is preferable to construct a Poisson equation from the momentum equations. In two dimensions, this can be written

$$\frac{\partial^2 p}{\partial x^2}+\frac{\partial^2 p}{\partial y^2}=2\left(\frac{\partial u}{\partial x}\frac{\partial v}{\partial y}-\frac{\partial v}{\partial x}\frac{\partial u}{\partial y}\right) , \tag{17.135}$$

where the right-hand side of (17.135) is known from the stream function, vorticity solution. Equation (17.135) is applicable to both steady and unsteady flow.

Boundary conditions to suit (17.135) are usually Dirichlet boundary conditions in the freestream and Neumann boundary conditions at a solid surface. The Neumann boundary conditions are obtained from the normal momentum equation, which reduces to the following nondimensional form:

$$\frac{\partial p}{\partial n}=\frac{1}{\mathrm{Re}}\frac{\partial \zeta}{\partial s} , \tag{17.136}$$

where s is measured along the boundary. For high Reynolds number flow parallel to a flat surface, (17.136) reduces to the boundary layer assumption $\partial p/\partial n=0$. The solution of (17.135) must also satisfy the global integral constraint (17.16). This implies

$$\iint\left(\frac{\partial^2 p}{\partial x^2}+\frac{\partial^2 p}{\partial y^2}\right)dx\,dy=0=\int_{c}\frac{\partial p}{\partial n}ds . \tag{17.137}$$

For internal flow problems where a Neumann boundary condition is specified on all boundaries it is important to ensure that (17.137) is satisfied.

Since (17.135) is a Poisson equation any of the techniques suitable for linear strongly elliptic problems are available to solve the discrete form of (17.135). If the discretisation is undertaken on a uniform grid direct Poisson solvers (Sect. 6.2.6) are suitable. For both uniform and nonuniform grids the iterative techniques described in Sect. 6.3 are appropriate.

For external flows, like the flow over a backward-facing step, there is an advantage of working with the Bernoulli variable, H, instead of the pressure (11.49). In nondimensional form

$$H=c_{\mathrm{p}}+u^2+v^2 , \tag{17.138}$$

where the pressure coefficient $c_p = (P - P_\infty)/0.5\varrho U_\infty^2$. From the momentum equations, a Poisson equation for H replaces (17.135):

$$\frac{\partial^2 H}{\partial x^2} + \frac{\partial^2 H}{\partial y^2} = 2\left(\frac{\partial(u\zeta)}{\partial y} - \frac{\partial(v\zeta)}{\partial x}\right) . \tag{17.139}$$

Equation (17.139) is applicable to both steady and unsteady flow. Neumann and Dirichlet boundary conditions for H are obtained from the momentum equations. Where the flow is locally inviscid H is a constant. Consequently for flow about an isolated body it is possible to solve the discrete form of (17.139) with the farfield boundary much closer to the body than would be the case when solving (17.135). Equation (17.139) is solved to obtain the global pressure distribution for the flow past rearward-facing cavities (Fletcher and Barbuto 1986a, b).

For steady two-dimensional flow (17.135) or (17.139) need only be solved once after the velocity solution has been obtained. If the pressure is required for an unsteady flow it is necessary to solve (17.135) or (17.139) at every time step. In this case a primitive variable approach is often preferred to a stream function vorticity formulation.

17.4 Vorticity Formulations for Three-Dimensional Flows

In two dimensions the vorticity stream function formulation is often more efficient than a primitive variable formulation, primarily because the use of the stream function avoids explicit solution of the continuity equation (17.1). In three dimensions vorticity-related formulations lead to more dependent variables, typically six, than is the case for primitive variables, typically four. As a result three-dimensional vorticity-related formulations have not been used very often.

In this section two alternative formulations are examined. Both use the three-component vorticity transport equations and avoid the explicit appearance of the pressure. They differ in the choice of additional equations to obtain the velocity field.

17.4.1 Vorticity, Vector Potential Formulation

The extension of the vorticity stream function formulation (Sect. 17.3) to three-dimensional flow requires replacement of the stream function by a three-component vector potential and requires consideration of all three vorticity components.

The three-component vorticity transport equation, replacing (17.90), is

$$\frac{\partial \zeta}{\partial t} + \nabla \cdot (\mathbf{u}\,\zeta) - (\zeta \cdot \nabla)\mathbf{u} - \frac{1}{\mathrm{Re}}\nabla^2 \zeta = 0 . \tag{17.140}$$

The structure of (17.140) is similar to that of (17.90) except that a new term $(\zeta \cdot \nabla)\mathbf{u}$ appears, which can be thought of as a vortex stretching term. In Cartesian coordinates the x-component of (17.140) is

$$\frac{\partial \zeta_x}{\partial t}+\frac{\partial}{\partial x}(u\zeta_x)+\frac{\partial}{\partial y}(v\zeta_x)+\frac{\partial}{\partial z}(w\zeta_x)-\zeta_x\frac{\partial u}{\partial x}-\zeta_y\frac{\partial u}{\partial y}-\zeta_z\frac{\partial u}{\partial z}$$

$$-\frac{1}{\mathrm{Re}}\left(\frac{\partial^2\zeta_x}{\partial x^2}+\frac{\partial^2\zeta_x}{\partial y^2}+\frac{\partial^2\zeta_x}{\partial z^2}\right)=0 \ . \qquad (17.141)$$

The three vorticity components are related to the velocity components by $\boldsymbol{\zeta}=\operatorname{curl}\mathbf{u}$. However, to obtain the velocity field from the vorticity field it is necessary to introduce a vector potential $\boldsymbol{\psi}$, such that

$$\mathbf{u}=\operatorname{curl}\boldsymbol{\psi} \ , \quad \text{i.e.} \qquad (17.142)$$

$$u=\frac{\partial\psi_z}{\partial y}-\frac{\partial\psi_y}{\partial z} \ , \qquad v=\frac{\partial\psi_x}{\partial z}-\frac{\partial\psi_z}{\partial x} \ , \qquad w=\frac{\partial\psi_y}{\partial x}-\frac{\partial\psi_x}{\partial y} \ .$$

Clearly the vector potential $\boldsymbol{\psi}$ is the three-dimensional extension of the scalar stream function in two dimensions ($\psi=\psi_z$, $\psi_x=\psi_y=0$).

The three-dimensional equivalent of (17.92) is

$$\nabla\boldsymbol{\psi}=-\boldsymbol{\zeta} \ . \qquad (17.143)$$

Thus three-dimensional viscous incompressible flow is governed by (17.140, 142 and 143). Since each equation has three components the solution of three-dimensional flow is less economical using the vorticity, vector potential formulation than using primitive variables (Sects. 17.1 and 17.2). However, since (17.140) are transport equations and (17.143) are Poisson equations the same computational techniques are appropriate as in two dimensions.

For confined flows, such as the driven cavity problem, boundary conditions for the vector potential are given by Aziz and Hellums (1967) as

$$\text{i) Surface } x=\text{const:}\ \frac{\partial\psi_x}{\partial x}=\psi_y=\psi_z=0 \ ,$$

$$\text{ii) Surface } y=\text{const:}\ \frac{\partial\psi_y}{\partial y}=\psi_x=\psi_z=0 \ , \qquad (17.144)$$

$$\text{iii) Surface } z=\text{const:}\ \frac{\partial\psi_z}{\partial y}=\psi_x=\psi_y=0 \ ,$$

and for the vorticity:

$$\text{i) Surface } x=\text{const:}\ \zeta_x=0 \ , \qquad \zeta_y=-\frac{\partial w}{\partial x} \ , \qquad \zeta_z=\frac{\partial v}{\partial x} \ ,$$

$$\text{ii) Surface } y=\text{const:}\ \zeta_x=\frac{\partial w}{\partial y} \ , \qquad \zeta_y=0 \ , \qquad \zeta_z=-\frac{\partial u}{\partial y} \ ,$$

$$\text{iii) Surface } z=\text{const:}\ \zeta_x=-\frac{\partial v}{\partial z} \ , \ \zeta_y=\frac{\partial u}{\partial z} \ , \zeta_z=0 \ . \qquad (17.145)$$

The vorticity, vector potential formulation has been used by Aziz and Hellums (1967) and by Mallinson and de Vahl Davis (1977) to study three-dimensional natural convection in a box.

For problems with inflow and outflow the boundary conditions given by (17.144, 145) must be generalised. Although this is possible (Hirasaki and Hellums 1968) the result is cumbersome. A preferred procedure (Hirasaki and Hellums 1970) is to replace (17.142) with

$$\mathbf{u}=\operatorname{curl}\boldsymbol{\psi}+\nabla\phi\ , \tag{17.146}$$

where ϕ is an auxiliary potential (compare Sect. 17.2.2) introduced to provide a simpler prescription of the inflow, outflow boundary conditions. The satisfaction of continuity implies that

$$\nabla^2\phi=0\ . \tag{17.147}$$

The other governing equations remain as before. The boundary conditions for (17.147) are of Neumann type,

$$\frac{\partial\phi}{\partial n}=-n\cdot\mathbf{u}\ . \tag{17.148}$$

Thus a prescribed inflow/outflow velocity distribution enters through (17.148). At a solid surface (17.148) reduces to $\partial\phi/\partial n=0$. In addition, boundary conditions (17.144 and 145) are applicable without further modification. Aregbesola and Burley (1977) have used the vorticity, vector potential, auxiliary potential formulation to study three-dimensional duct flows.

Wong and Reizes (1984) demonstrate that the introduction of the auxiliary potential no longer automatically satisfies continuity when the discrete form of the equations are considered. Consequently they prefer to replace (17.146) with

$$\mathbf{v}=\operatorname{curl}\boldsymbol{\psi}+w_0\ , \tag{17.149}$$

where $w_0(x, y)$ is the specified inlet velocity distribution for a straight duct aligned parallel to the z axis. In this formulation (17.144) are applicable at solid surfaces and at an inflow boundary, $z=\text{const}$. At an outflow boundary, $z=\text{const}$. Wong and Reizes (1984) recommend the following boundary condition in place of (17.144):

$$\frac{\partial\psi_x}{\partial z}=\frac{\partial\psi_y}{\partial z}=0\ ,\quad \frac{\partial\psi_z}{\partial z}=-\left(\frac{\partial\psi_x}{\partial x}+\frac{\partial\psi_y}{\partial y}\right)\ . \tag{17.150}$$

The boundary conditions on the vorticity are as indicated in (17.145). The numerical implementation of these boundary conditions is the same as in two dimensions (Sect. 17.3.2).

17.4.2 Vorticity, Velocity Formulation

In this formulation (Fasel 1978; Dennis et al. 1979) the vorticity transport equations (17.140) are retained. However, from the definition of the vorticity,

$\zeta = \text{curl}\,\mathbf{u}$, and the continuity equation it is possible to derive the following Poisson equations for the velocity components:

$$\nabla^2 u = \frac{\partial \zeta_y}{\partial z} - \frac{\partial \zeta_z}{\partial y} \ , \qquad \nabla^2 v = \frac{\partial \zeta_z}{\partial x} - \frac{\partial \zeta_x}{\partial z} \ ,$$
$$\nabla^2 w = \frac{\partial \zeta_x}{\partial y} - \frac{\partial \zeta_y}{\partial x} \ . \tag{17.151}$$

In the present formulation (17.140 and 151) provide the governing equations. At solid surfaces boundary conditions are given by no slip, $u = v = w = 0$, and by (17.145) for the vorticity. For inflow boundaries it is appropriate to specify the velocity field; at outflow Neumann boundary conditions for the velocity components are specified (17.17). In addition simplifications to the vorticity transport equations (as in Chap. 16) may avoid the need to prescribe downstream boundary conditions.

Dennis et al. (1979) use modified exponential differencing (Dennis 1985) for the convective terms in the vorticity transport equations. Conventional three-point differencing is used for the second derivative terms in (17.140) and all the terms in (17.151). The discretised steady form of (17.140) and the discretised form of (17.151) form a global diagonally dominant system of equations. Dennis et al. solve these using successive over-relaxation. They compute the flow in a three-dimensional driven cavity for Reynolds numbers up to $\text{Re} = 400$ on a 25 × 25 × 25 grid.

It is possible to consider a two-dimensional version of the vorticity, velocity formulation, namely the vorticity transport equation (17.90), the vorticity definition equation (17.89) and the continuity equation (17.1). Gatski et al. (1982) have used such a formulation to examine the driven cavity problem and more complex unsteady viscous flows (Gatski and Grosch 1985). Gatski et al. combine the discrete form of (17.1 and 89) into a global block matrix equation. This equation is not diagonally dominant and it is reported (Gatski et al. 1982) that the iterative algorithm to solve the block matrix equation is not very efficient.

To solve the discrete form of (17.90), auxiliary variables are introduced for $\partial\zeta/\partial x$ and $\partial\zeta/\partial y$. This allows a form of differencing similar to the Keller box scheme (Sect. 15.1.3). The discrete vorticity equation is solved using a modified ADI procedure. The overall algorithm is sequential rather than coupled, i.e. the vorticity transport equation is solved separately from the vorticity definition, continuity equation combination.

Fasel (1976) has solved (17.90) in conjunction with (17.151) with $\zeta_x = \zeta_y = 0$ to examine transition phenomena in two-dimensional boundary layer flows. Orlandi (1987) uses a related scheme but includes the differentiated form of the continuity equation as well. Orlandi constructs a block ADI-like scheme in which all equations are coupled at each half time-step. For the first half time-step the coupled equations are (17.90, 151b) and

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 v}{\partial x \partial y} = 0 \ . \tag{17.152}$$

For the second half time-step the coupled equations are (17.90, 151a) and

$$\frac{\partial^2 u}{\partial x \partial y} + \frac{\partial^2 v}{\partial y^2} = 0 \ . \tag{17.153}$$

Equations (17.152 and 153) are constructed by differentiating the continuity equation (17.1). It may also be noted that the differentiated form of the continuity equation is used in constructing (17.151).

It would appear that using a differentiated form of the incompressible continuity equation, $\partial D/\partial x = \partial D/\partial y = 0$, where D is the dilatation (11.13), does not guarantee that continuity is satisfied unless D is set to zero at least at one point. Orlandi does this by explicitly imposing (17.1) on the boundary. Orlandi notes that this guarantees exact mass conservation for the discretised equations.

Orlandi indicates that in practice this leads to a more efficient scheme since fewer iterations are required to produce a velocity field that satisfies continuity. Orlandi demonstrates the scheme for the driven cavity problem and the flow over a backward-facing step.

It may be concluded that vorticity-based formulations are not so efficient as primitive variables in three dimensions unless the vortex motion, in particular unsteady, is of special interest. In addition it may be noted that almost all practical turbulence modelling has been undertaken in terms of the primitive variables.

The SIMPLE algorithm

The acronym SIMPLE stands for Semi-Implicit Method for Pressure-Linked Equations. The algorithm was originally put forward by Patankar and Spalding (1972) and is essentially a guess-and-correct procedure for the calculation of pressure on the staggered grid arrangement introduced above. The method is illustrated by considering the two-dimensional laminar steady flow equations in Cartesian co-ordinates.

To initiate the SIMPLE calculation process a pressure field p^* is guessed. **Discretised momentum equations** (6.8) and (6.10) **are solved** using the guessed pressure field to yield velocity components u^* and v* as follows:

$$\boxed{a_{i,J}u^*_{i,J} = \sum a_{nb}u^*_{nb} + (p^*_{I-1,J} - p^*_{I,J})A_{i,J} + b_{i,J}} \tag{6.12}$$

$$\boxed{a_{I,j}v^*_{I,j} = \sum a_{nb}v^*_{nb} + (p^*_{I,J-1} - p^*_{I,J})A_{I,j} + b_{I,j}} \tag{6.13}$$

Now we define the correction p' as the difference between the correct pressure field

p and the guessed pressure field p^*, so that

$$\boxed{p = p^* + p'} \tag{6.14}$$

Similarly we define velocity corrections u' and v' to relate the correct velocities u and v to the guessed velocities u^* and v^*

$$\boxed{u = u^* + u'} \tag{6.15}$$

$$\boxed{v = v^* + v'} \tag{6.16}$$

Substitution of the correct pressure field p into the momentum equations yields the correct velocity field (u, v). Discretised equations (6.8) and (6.10) link the correct velocity fields with the correct pressure field.

Subtraction of equations (6.12) and (6.13) from (6.8) and (6.10), respectively, gives

$$\begin{aligned} a_{i,J}(u_{i,J} - u^*_{i,J}) = {} & \sum a_{nb}(u_{nb} - u^*_{nb}) \\ & + \left[(p_{I-1,J} - p^*_{I-1,J}) - \left(p_{I,J} - p^*_{I,J}\right)\right]A_{i,J} \end{aligned} \tag{6.17}$$

$$\begin{aligned} a_{I,j}(v_{I,j} - v^*_{I,j}) = {} & \sum a_{nb}(v_{nb} - v^*_{nb}) \\ & + [(p_{I,J-1} - p^*_{I,J-1}) - (p_{I,J} - p^*_{I,J})]A_{I,j} \end{aligned} \tag{6.18}$$

Using correction formulae (6.14–6.16) the equations (6.17–6.18) may be rewritten as follows:

$$a_{i,J}u'_{i,J} = \sum a_{nb}u'_{nb} + \left(p'_{I-1,J} - p'_{I,J}\right)A_{i,J} \tag{6.19}$$

$$a_{I,j}v'_{I,j} = \sum a_{nb}v'_{nb} + \left(p'_{I,J-1} - p'_{I,J}\right)A_{I,j} \tag{6.20}$$

At this point an approximation is introduced: $\sum a_{nb}u'_{nb}$ and $\sum a_{nb}v'_{nb}$ are dropped to simplify equations (6.19) and (6.20) for the velocity corrections. **Omission of these terms is the main approximation** of the SIMPLE algorithm. We obtain

$$u'_{i,J} = d_{i,J}\left(p'_{I-1,J} - p'_{I,J}\right) \tag{6.21}$$

$$v'_{I,j} = d_{I,j}\left(p'_{I,J-1} - p'_{I,J}\right) \tag{6.22}$$

$$\text{where} \quad d_{i,J} = \frac{A_{i,J}}{a_{i,J}} \text{ and } d_{I,j} = \frac{A_{I,j}}{a_{I,j}} \tag{6.23}$$

Equations (6.21) and (6.22) describe the corrections to be applied to velocities through formulae (6.15) and (6.16), which gives

$$\boxed{u_{i,J} = u^*_{i,J} + d_{i,J}\left(p'_{I-1,J} - p'_{I,J}\right)} \tag{6.24}$$

$$\boxed{v_{I,j} = v^*_{I,j} + d_{I,j}\left(p'_{I,J-1} - p'_{I,J}\right)} \tag{6.25}$$

Similar expressions exist for $u_{i+1,J}$ and $v_{I,j+1}$:

$$u_{i+1,J} = u^*_{i+1,J} + d_{i+1,J}\left(p'_{I,J} - p'_{I+1,J}\right) \tag{6.26}$$

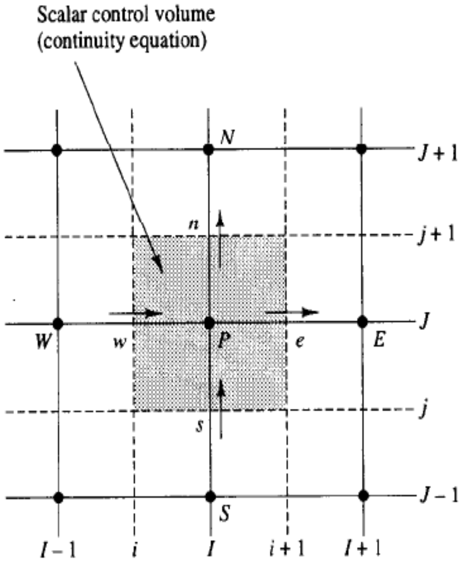
$$v_{I,j+1} = v^*_{I,j+1} + d_{I,j+1}\left(p'_{I,J} - p'_{I,J+1}\right) \tag{6.27}$$

$$\text{where } d_{i+1,J} = \frac{A_{i+1,J}}{a_{i+1,J}} \text{ and } d_{I,j+1} = \frac{A_{I,j+1}}{a_{I,j+1}} \tag{6.28}$$

Thus far we have only considered the momentum equations but, as mentioned earlier, the velocity field is also subject to the constraint that it should satisfy continuity equation (6.3). Continuity is satisfied in discretised form for the scalar control volume shown in Figure 6.5:

$$\left[(\rho uA)_{i+1,J} - (\rho uA)_{i,J}\right] + \left[(\rho vA)_{I,j+1} - (\rho vA)_{I,j}\right] = 0 \tag{6.29}$$

Fig. 6.5 The scalar control volume used for the discretisation of the continuity equation



Substitution of the corrected velocities of equations (6.24–6.27) into discretised continuity equation (6.29) gives

$$\begin{aligned}&\Big[\rho_{i+1,J}A_{i+1,J}\Big(u^*_{i+1,J} + d_{i+1,J}\left(p'_{I,J} - p'_{I+1,J}\right)\Big)\\ &\quad -\rho_{i,J}A_{i,J}\Big(u^*_{i,J} + d_{i,J}\left(p'_{I-1,J} - p'_{I,J}\right)\Big)\Big]\\ &\quad \mid \Big[\rho_{I,j+1}A_{I,j+1}\Big(v^*_{I,j+1} \mid d_{I,j+1}\left(p'_{I,J} - p'_{I,J+1}\right)\Big)\\ &\quad -\rho_{I,j}A^*_{I,j}\Big(v^*_{I,j} + d_{I,j}\left(p'_{I,J-1} - p'_{I,J}\right)\Big)\Big] = 0\end{aligned} \tag{6.30}$$

This may be re-arranged to give

$$\begin{aligned}&\left[(\rho dA)_{i+1,J} + (\rho dA)_{i,J} + (\rho dA)_{I,j+1} + (\rho dA)_{I,j}\right]p'_{I,J}\\ &\quad - (\rho dA)_{i+1,J}p'_{I+1,J} + (\rho dA)_{i,J}p'_{I-1,J} + (\rho dA)_{I,j+1}p'_{I,J+1}\\ &\quad + (\rho dA)_{I,j}p'_{I,J-1}\\ &\quad + \left[(\rho u^*A)_{i,J} - (\rho u^*A)_{i+1,J} + (\rho v^*A)_{I,j} - (\rho v^*A)_{I,j+1}\right]\end{aligned} \tag{6.31}$$

Identifying the coefficients of p' this may be written as

$$\boxed{\begin{aligned} a_{I,J}p'_{I,J} = {} & a_{I+1,J}p'_{I+1,J} + a_{I-1,J}p'_{I-1,J} + a_{I,J+1}p'_{I,J+1} \\ & + a_{I,J-1}p'_{I,J-1} + b'_{I,J} \end{aligned}} \tag{6.32}$$

where $a_{I,J} = a_{I+1,J} + a_{I-1,J} + a_{I,J+1} + a_{I,J-1}$ and the coefficients are given below:

$a_{I\,\,1,J}$	$a_{I-1,J}$	$a_{I,J+1}$	$a_{i,J-1}$	$b'_{I,J}$
$(\rho dA)_{i+1,J}$	$(\rho dA)_{i,J}$	$(\rho dA)_{I,j+1}$	$(\rho dA)_{I,j}$	$\left(\rho u^* A\right)_{i,J} - \left(\rho u^* A\right)_{i+1,J} + \left(\rho v^* A\right)_{I,j} - \left(\rho v^* A\right)_{I,j+1}$

Equation (6.32) represents the discretised continuity equation as an **equation for pressure correction p'**. The source term b' in the equation is the continuity imbalance arising from the incorrect velocity field u^*, v^*. By solving equation (6.32), the pressure correction field p' can be obtained at all points. Once the pressure correction field is known, the correct pressure field may be obtained using formula (6.14) and velocity components through correction formulae (6.24–6.27). The omission of terms such as $\sum a_{nb}u'_{nb}$ in the derivation does not affect the final solution because the pressure correction and velocity corrections will all be zero in a converged solution giving $p^* = p, u^* = u$ and $v^* = v$.

The pressure correction equation is susceptible to divergence unless some **under-relaxation** is used during the iterative process and new, improved, pressures p^{new} are obtained with

$$p^{new} = p^* \; + \; \alpha_p p' \tag{6.33}$$

where α_p is the pressure under-relaxation factor. If we select α_p equal to 1 the guessed pressure field p^* is corrected by p'. However, the corrections p', in particular when the guessed field p^* is far away from the final solution, is often too large for stable computations. A value of α_p equal to zero would apply no correction at all, which is also undesirable. Taking α_p between 0 and 1 allows us to add to guessed field p^* a fraction of the correction field p' that is large enough to move the iterative improvement process forward, but small enough to ensure stable computation.

The velocities are also under-relaxed. The iteratively improved velocity components u^{new} and v^{new} are obtained from

$$u^{new} = \alpha_u u + (1 - \alpha_u)u^{(n-1)} \tag{6.34}$$

$$v^{new} = \alpha_v v + (1 - \alpha_v)v^{(n-1)} \tag{6.35}$$

where α_u and α_v are the u- and v-velocity under-relaxation factors with values between 0 and 1, u and v are the corrected velocity components without relaxation and $u^{(n-1)}$ and $v^{(n-1)}$ represent their values obtained in the previous iteration. After some algebra it can be shown that with under-relaxation the discretised u-momentum equation takes the form

$$\frac{a_{i,J}}{\alpha_u}u_{i,J} = \sum a_{nb}u_{nb} + (p_{I-1,J} - p_{I,J})A_{i,J} + b_{i,J} + \left[(1 - \alpha_u)\frac{a_{i,J}}{\alpha_u}\right]u_{i,J}^{(n-1)} \tag{6.36}$$

and the discretised *v*-momentum equation

$$\frac{a_{I,j}}{\alpha_v} v_{I,j} = \sum a_{nb} v_{nb} + (p_{I,J-1} - p_{I,J}) A_{I,j} + b_{I,j} + \left[(1 - \alpha_v) \frac{a_{I,j}}{\alpha_v} \right] v_{I,j}^{(n-1)} \tag{6.37}$$

The pressure correction equation is also affected by velocity under-relaxation and it can be shown that *d*-terms of the pressure correction equation become

$$d_{i,J} = \frac{A_{i,J}\alpha_u}{a_{i,J}}, \quad d_{i+1,J} = \frac{A_{i+1,J}\alpha_u}{a_{i+1,J}}, \quad d_{I,j} = \frac{A_{I,j}\alpha_v}{a_{I,j}}$$

and

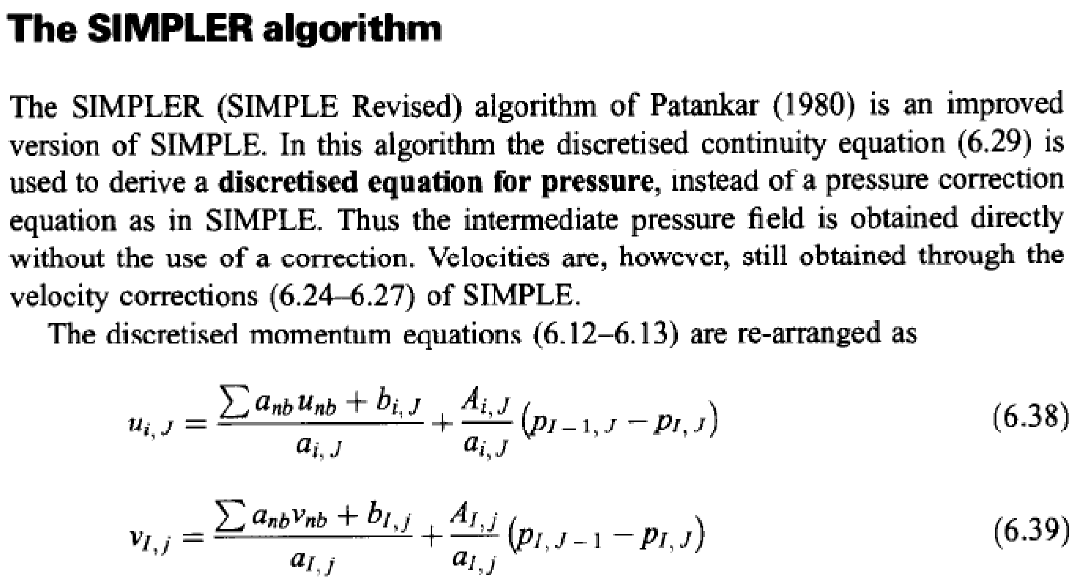
$$d_{I,j+1} = \frac{A_{I,j+1}\alpha_v}{a_{I,j+1}}$$

Note that in these formulae $a_{i,J}, a_{i+1,J}, a_{I,j}$ and $a_{I,j+1}$ are the central coefficients of discretised velocity equations at positions (i,J), $(i+1,J)$, (I,j) and $(I,j+1)$ of a scalar cell centred around *P*.

A correct choice of under-relaxation factors α is essential for cost-effective simulations. Too large a value of α may lead to oscillatory or even divergent iterative solutions and a value which is too small will cause extremely slow convergence. Unfortunately, the optimum values of under-relaxation factors are flow dependent and must be sought on a case-by-case basis. The use of under-relaxation will be discussed further in Chapter 8.

Assembly of a complete method

The SIMPLE algorithm gives a method of calculating pressure and velocities. The method is iterative and when other scalars are coupled to the momentum equations, the calculation needs to be done sequentially. The sequence of operations in a CFD procedure which employs the SIMPLE algorithm is given in Figure 6.6.



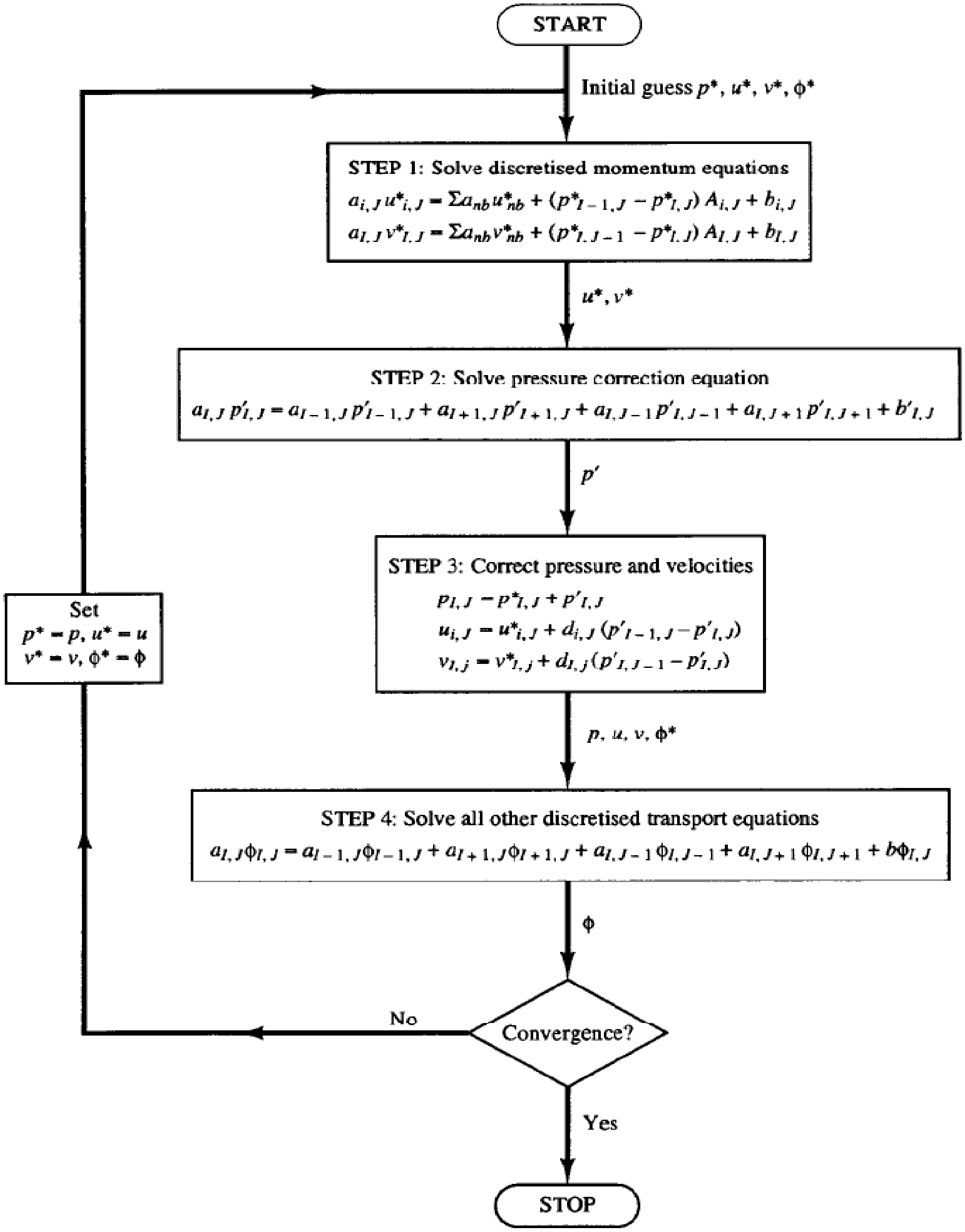
The SIMPLER algorithm

The SIMPLER (SIMPLE Revised) algorithm of Patankar (1980) is an improved version of SIMPLE. In this algorithm the discretised continuity equation (6.29) is used to derive a **discretised equation for pressure**, instead of a pressure correction equation as in SIMPLE. Thus the intermediate pressure field is obtained directly without the use of a correction. Velocities are, however, still obtained through the velocity corrections (6.24–6.27) of SIMPLE.

The discretised momentum equations (6.12–6.13) are re-arranged as

$$u_{i,J} = \frac{\sum a_{nb}u_{nb} + b_{i,J}}{a_{i,J}} + \frac{A_{i,J}}{a_{i,J}}\left(p_{I-1,J} - p_{I,J}\right) \tag{6.38}$$

$$v_{I,j} = \frac{\sum a_{nb}v_{nb} + b_{I,j}}{a_{I,j}} + \frac{A_{I,j}}{a_{I,j}}\left(p_{I,J-1} - p_{I,J}\right) \tag{6.39}$$


START
Initial guess p^*, u^*, v^*, ϕ^*
STEP 1: Solve discretised momentum equations
$a_{i,J}u^*_{i,J} = \Sigma a_{nb}u^*_{nb} + (p^*_{I-1,J} - p^*_{I,J})A_{i,J} + b_{i,J}$
$a_{I,J}v^*_{I,J} = \Sigma a_{nb}v^*_{nb} + (p^*_{I,J-1} - p^*_{I,J})A_{I,J} + b_{I,J}$
u^*, v^*
STEP 2: Solve pressure correction equation
$a_{I,J}p'_{I,J} = a_{I-1,J}p'_{I-1,J} + a_{I+1,J}p'_{I+1,J} + a_{I,J-1}p'_{I,J-1} + a_{I,J+1}p'_{I,J+1} + b'_{I,J}$
p'
STEP 3: Correct pressure and velocities
$p_{I,J} = p^*_{I,J} + p'_{I,J}$
$u_{i,J} = u^*_{i,J} + d_{i,J}(p'_{I-1,J} - p'_{I,J})$
$v_{I,j} = v^*_{I,j} + d_{I,j}(p'_{I,J-1} - p'_{I,J})$
p, u, v, ϕ^*
STEP 4: Solve all other discretised transport equations
$a_{I,J}\phi_{I,J} = a_{I-1,J}\phi_{I-1,J} + a_{I+1,J}\phi_{I+1,J} + a_{I,J-1}\phi_{I,J-1} + a_{I,J+1}\phi_{I,J+1} + b_{\phi_{I,J}}$
ϕ
Convergence?
No
Yes
STOP
Set
$p^* = p, u^* = u$
$v^* = v, \phi^* = \phi$

In the SIMPLER algorithm pseudo-velocities $\hat{u}$ and $\hat{v}$ are now defined as follows:

$$\hat{u}_{i,J} = \frac{\sum a_{nb}u_{nb} + b_{i,J}}{a_{i,J}} \tag{6.40}$$

$$\hat{v}_{I,j} = \frac{\sum a_{nb}v_{nb} + b_{I,j}}{a_{I,j}} \tag{6.41}$$

Equations (6.38) and (6.39) can now be written as

$$u_{i,J} = \hat{u}_{i,J} + d_{i,J}\left(p_{I-1,J} - p_{I,J}\right) \tag{6.42}$$

$$v_{I,j} = \ddot{v}_{I,j} + d_{I,j}\left(p_{I,J-1} - p_{I,J}\right) \tag{6.43}$$

The definition for d, introduced in the developments of section 6.4, is applied in (6.42–6.43). Substituting for $u_{i,J}$ and $v_{I,j}$ from these equations into the discretised continuity equation (6.29), using similar forms for $u_{i+1,J}$ and $v_{I,j+1}$, results in

$$\begin{aligned}&\Big[\rho_{i+1,J}A_{i+1,J}\left(\hat{u}_{i+1,J} + d_{i+1,J}\left(p_{I,J} - p_{I+1,J}\right)\right)\\ &\quad -\rho_{i,J}A_{i,J}\left(\hat{u}_{i,J} + d_{i,J}\left(p_{I-1,J} - p_{I,J}\right)\right)\Big]\\ &\quad +\Big[\rho_{I,j+1}A_{I,j+1}\left(\hat{v}_{I,j+1} + d_{I,j+1}\left(p_{I,J} - p_{I,J+1}\right)\right)\\ &\quad -\rho_{I,j}A_{I,j}\left(\hat{v}_{I,j} + d_{I,j}\left(p_{I,J-1} - p_{I,J}\right)\right)\Big] = 0\end{aligned} \tag{6.44}$$

Equation (6.44) may be re-arranged to give a discretised pressure equation

$$\boxed{\begin{aligned}a_{I,J}p_{I,J} = a_{I+1,J}p_{I+1,J} + a_{I-1,J}p_{I-1,J} + a_{I,J+1}p_{I,J+1}\\ + a_{I,J-1}p_{I,J-1} + b_{I,J}\end{aligned}} \tag{6.45}$$

where $a_{I,J} = a_{I+1,J} + a_{I-1,J} + a_{I,J+1} + a_{I,J-1}$ and the coefficients are given below:

$a_{I+1,J}$	$a_{I-1,J}$	$a_{I,J+1}$	$a_{I,J-1}$	$b_{I,J}$
$(\rho dA)_{i+1,J}$	$(\rho dA)_{i,J}$	$(\rho dA)_{I,j+1}$	$(\rho dA)_{I,j}$	$(\rho\hat{u}A)_{i,J} - (\rho\hat{u}A)_{i+1,J} + (\rho\hat{v}A)_{I,j} - (\rho\hat{v}A)_{I,j+1}$

Note that the coefficients of equation (6.45) are the same as those in the discretised pressure correction equation (6.32), with the difference that the source term b is evaluated using the pseudo-velocities. Subsequently, the discretised momentum equations (6.12–6.13) are solved using the pressure field obtained above. This yields the velocity components u^* and v^*. The velocity correction equations (6.24–6.27) are used in the SIMPLER algorithm to obtain corrected velocities. Therefore, the p'-equation (6.32) must also be solved to obtain the pressure corrections needed for the velocity corrections. The full sequence of operations is described in Figure 6.7.

6.7 The SIMPLEC algorithm

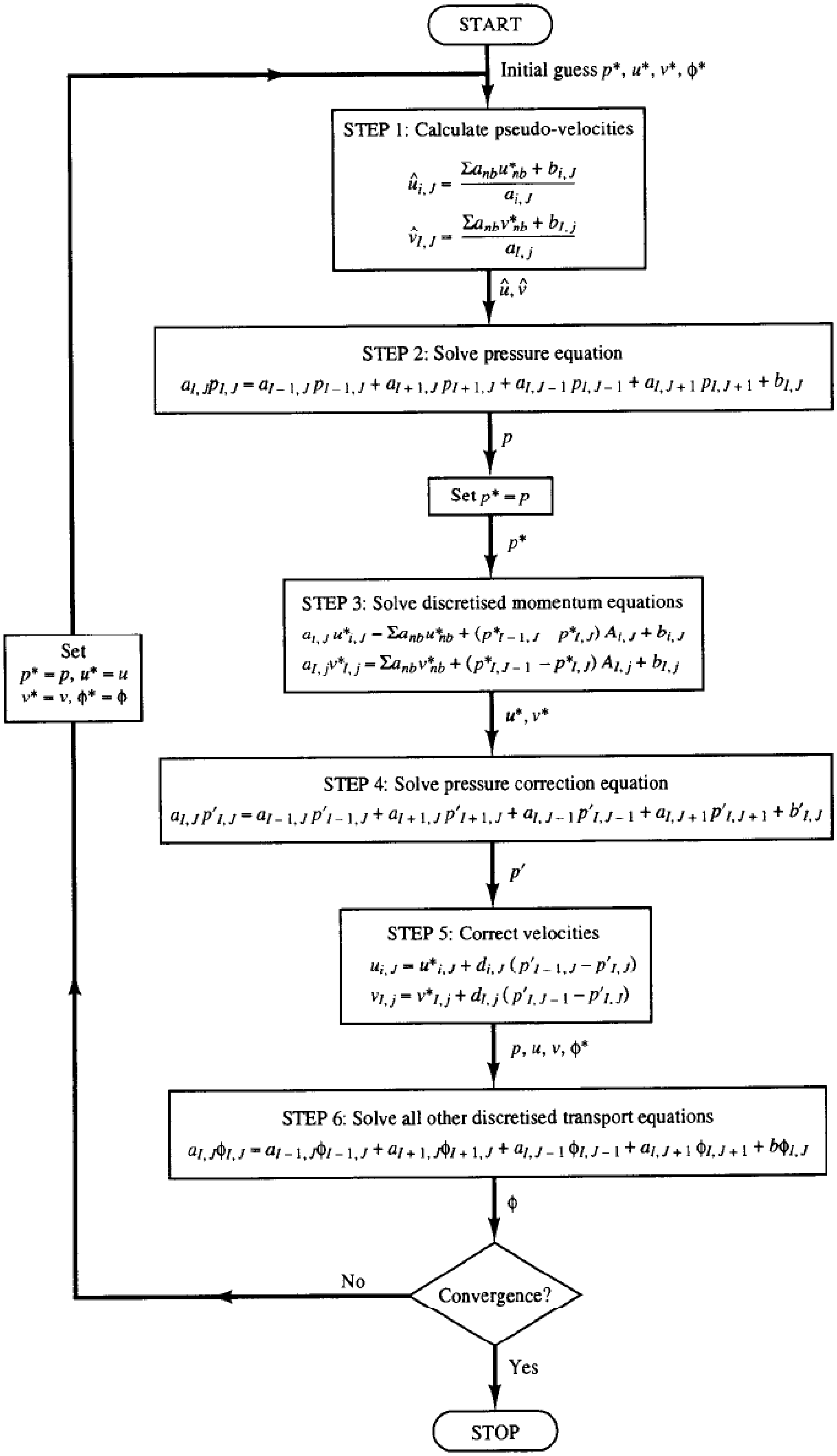
The SIMPLEC (SIMPLE-Consistent) algorithm of Van Doormal and Raithby (1984) follows the same steps as the SIMPLE algorithm, with the difference that the momentum equations are manipulated so that the SIMPLEC velocity correction equations omit terms that are less significant than those omitted in SIMPLE.

The u-velocity correction equation of SIMPLEC is given by

$$u'_{i,J} - d_{i,J}\left(p'_{I-1,J} - p'_{I,J}\right) \tag{6.46}$$

where

$$d_{i,J} = \frac{A_{i,J}}{a_{i,J} - \sum a_{nb}} \tag{6.47}$$


START
Initial guess p^*, u^*, v^*, ϕ^*
STEP 1: Calculate pseudo-velocities
$\hat{u}_{i,J} = \frac{\Sigma a_{nb} u^*_{nb} + b_{i,J}}{a_{i,J}}$
$\hat{v}_{I,J} = \frac{\Sigma a_{nb} v^*_{nb} + b_{I,j}}{a_{I,j}}$
$\hat{u}, \hat{v}$
STEP 2: Solve pressure equation
$a_{I,J} p_{I,J} = a_{I-1,J} p_{I-1,J} + a_{I+1,J} p_{I+1,J} + a_{I,J-1} p_{I,J-1} + a_{I,J+1} p_{I,J+1} + b_{I,J}$
p
Set $p^* = p$
p^*
STEP 3: Solve discretised momentum equations
$a_{i,J} u^*_{i,J} - \Sigma a_{nb} u^*_{nb} + (p^*_{I-1,J} \quad p^*_{I,J}) A_{i,J} + b_{i,J}$
$a_{I,j} v^*_{I,j} = \Sigma a_{nb} v^*_{nb} + (p^*_{I,J-1} - p^*_{I,J}) A_{I,j} + b_{I,j}$
u^*, v^*
STEP 4: Solve pressure correction equation
$a_{I,J} p'_{I,J} = a_{I-1,J} p'_{I-1,J} + a_{I+1,J} p'_{I+1,J} + a_{I,J-1} p'_{I,J-1} + a_{I,J+1} p'_{I,J+1} + b'_{I,J}$
p'
STEP 5: Correct velocities
$u_{i,J} = u^*_{i,J} + d_{i,J} (p'_{I-1,J} - p'_{I,J})$
$v_{I,j} = v^*_{I,j} + d_{I,j} (p'_{I,J-1} - p'_{I,J})$
p, u, v, ϕ^*
STEP 6: Solve all other discretised transport equations
$a_{I,J} \phi_{I,J} = a_{I-1,J} \phi_{I-1,J} + a_{I+1,J} \phi_{I+1,J} + a_{I,J-1} \phi_{I,J-1} + a_{I,J+1} \phi_{I,J+1} + b\phi_{I,J}$
ϕ
Convergence?
No
Yes
STOP
Set
$p^* = p, u^* = u$
$v^* = v, \phi^* = \phi$

Similarly the modified *v*-velocity correction equation is

$$v'_{I,j} = d_{I,j}\left(p'_{I,J-1} - p'_{I,J}\right) \tag{6.48}$$

$$\text{where} \quad d_{I,j} = \frac{A_{I,j}}{a_{I,j} - \sum a_{nb}} \tag{6.49}$$

The discretised pressure correction equation is the same as in SIMPLE, except that the *d*-terms are calculated from equations (6.47) and (6.49). The sequence of operations of the SIMPLEC algorithm is identical to that of SIMPLE (see section 6.5).

The PISO algorithm

The PISO algorithm, which stands for Pressure Implicit with Splitting of Operators, of Issa (1986) is a pressure–velocity calculation procedure developed originally for the non-iterative computation of unsteady compressible flows. It has been adapted successfully for the iterative solution of steady state problems. PISO involves one predictor step and two corrector steps and may be seen as an extension of SIMPLE, with a **further corrector step** to enhance it.

Predictor step

Discretised momentum equations (6.12–6.13) are solved with a guessed or intermediate pressure field p^* to give velocity components u^* and v^* using the same method as the SIMPLE algorithm.

Corrector step 1

The fields u^* and v^* will not satisfy continuity unless the pressure field p^* is correct. The first corrector step of SIMPLE is introduced to give a velocity field (u^{**}, v^{**}) which satisfies the discretised continuity equation. The resulting equations are the same as the velocity correction equations (6.21–6.22) of SIMPLE but, since there is a further correction step in the PISO algorithm, we use a slightly different notation:

$$p^{**} = p^* + p'$$
$$u^{**} = u^* + u'$$
$$v^{**} = v^* + v'$$

These formulae are used to define corrected velocities u^{**} and v^{**}:

$$u^{**}_{i,J} = u^*_{i,J} + d_{i,J}\left(p'_{I-1,J} - p'_{I,J}\right) \tag{6.50}$$

$$v^{**}_{I,j} = v^*_{I,j} + d_{I,j}\left(p'_{I,J-1} - p'_{I,J}\right) \tag{6.51}$$

As in the SIMPLE algorithm equations (6.50–6.51) are substituted into the discretised continuity equation (6.29) to yield the pressure correction equation (6.32) with its coefficients and source term. In the context of the PISO method equation (6.32) is called the first pressure correction equation. It is solved to yield the first pressure correction field p'. Once the pressure corrections are known, the velocity components u^{**} and v^{**} can be obtained through equations (6.50–6.51).

Corrector step 2

To enhance the SIMPLE procedure PISO performs a second corrector step. The discretised momentum equations for u^{**} and v^{**} are

$$a_{i,J} u^{**}_{i,J} = \sum a_{nb} u^{*}_{nb} + (p^{**}_{I-1,J} - p^{**}_{I,J}) A_{i,J} + b_{i,J} \tag{6.12}$$

$$a_{I,j} v^{**}_{I,j} = \sum a_{nb} v^{*}_{nb} + (p^{**}_{I,J-1} - p^{**}_{I,J}) A_{I,j} + b_{I,j} \tag{6.13}$$

A twice-corrected velocity field (u^{***}, v^{***}) may be obtained by solving the momentum equations once more:

$$a_{i,J} u^{***}_{i,J} = \sum a_{nb} u^{**}_{nb} + (p^{***}_{I-1,J} - p^{***}_{I,J}) A_{i,J} + b_{i,J} \tag{6.52}$$

$$a_{I,j} v^{***}_{I,j} = \sum a_{nb} v^{**}_{nb} + (p^{***}_{I,J-1} - p^{***}_{I,J}) A_{I,j} + b_{I,j} \tag{6.53}$$

Note that the summation terms are evaluated using the velocities u^{**} and v^{**} calculated in the previous corrector step.

Subtraction of equation (6.12) from (6.52) and (6.13) from (6.53) gives

$$\boxed{u^{**}_{i,J} = u^{**}_{i,J} + \frac{\sum a_{nb}(u^{**}_{nb} - u^{*}_{nb})}{a_{i,J}} + d_{i,J}\left(p''_{I-1,J} - p''_{I,J}\right)} \tag{6.54}$$

$$\boxed{v^{***}_{I,j} = v^{**}_{I,j} + \frac{\sum a_{nb}(v^{**}_{nb} - v^{*}_{nb})}{a_{I,j}} + d_{I,j}\left(p''_{I,J-1} - p''_{I,J}\right)} \tag{6.55}$$

where p'' is the second pressure correction so that p^{***} may be obtained by

$$p^{***} = p^{**} + p'' \tag{6.56}$$

Substitution of u^{***} and v^{***} in the discretised continuity equation (6.29) yields a second pressure correction equation

$$\boxed{\begin{aligned} a_{I,J} p''_{I,J} = {} & a_{I+1,J} p''_{I+1,J} + a_{I-1,J} p''_{I-1,J} + a_{I,J+1} p''_{I,J+1} \\ & + a_{I,J-1} p''_{I,J-1} + b''_{I,J} \end{aligned}} \tag{6.57}$$

with $a_{I,J} = a_{I+1,J} + a_{I-1,J} + a_I,\ J + 1 + a_{I,J-1}$ and the coefficients are as follows:

$a_{I+1,J}$	$a_{I-1,J}$	$a_{I,J+1}$	$a_{I,J-1}$	$b''_{I,J}$
$(\rho dA)_{i+1,J}$	$(\rho dA)_{i,J}$	$(\rho dA)_{I,j+1}$	$(\rho dA)_{I,j}$	$\left[\left(\frac{\rho A}{a}\right)_{i,J}\sum a_{nb}(u_{nb}^{**}-u_{nb}^{*})\right.$ $-\left(\frac{\rho A}{a}\right)_{i+1,J}\sum a_{nb}(u_{nb}^{**}-u_{nb}^{*})$ $+\left(\frac{\rho A}{a}\right)_{I,j}\sum a_{nb}(v_{nb}^{**}-v_{nb}^{*})$ $-\left(\frac{\rho A}{a}\right)_{I,j+1}\sum a_{nb}(v_{nb}^{**}-v_{nb}^{*})$

In the derivation of (6.57) the source term

$$\left[(\rho Au^{**})_{i,J}-(\rho Au^{**})_{i+1,J}+(\rho Av^{**})_{I,j}-(\rho Av^{**})_{I,j+1}\right]$$

is zero since the velocity components u^{**} and v^{**} satisfy continuity.

Equation (6.57) is solved to obtain the second pressure correction field p'' and a twice-corrected pressure field is obtained from

$$\boxed{p^{***}=p^{**}+p''=p^{*}+p'+p''} \tag{6.58}$$

Finally the twice-corrected velocity field is obtained from equations (6.54–6.55).

In the non-iterative calculation of unsteady flows the pressure field p^{***} and the velocity fields u^{***} and v^{***} are considered to be the correct u, v and p. The sequence of operations for an iterative steady state PISO calculation is given in Figure 6.8.

The PISO algorithm solves the pressure correction equation twice so the method requires additional storage for calculating the source term of the second pressure correction equation. As before under-relaxation is required with the above procedure to stabilise the calculation process. Although this method implies a considerable increase in computational effort it has been found to be efficient and fast. For example, for a benchmark, laminar, backward-facing step problem Issa *et al* (1986) report a reduction of CPU time by a factor of 2 compared to standard SIMPLE.

The PISO algorithm presented above is the adapted, steady state version of an algorithm that was originally developed for non-iterative time-dependent calculations. The transient algorithm can also be applied to steady state calculations by starting with guessed initial conditions and solving as a transient problem for a long period of time until the steady state is achieved. This will be discussed in Chapter 8.

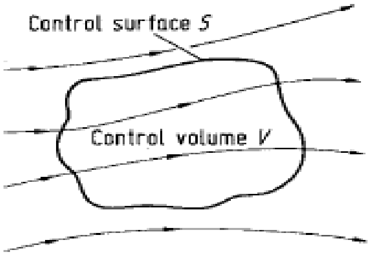
Two mark Question bank

1. What are the fundamental governing equations of fluid dynamics?
   a. continuity,
   b. momentum
   c. Energy equations.

2. What are the mathematical statements of three fundamental physical principles of fluid dynamics equations?
   (1) Mass is conserved;
   (2) F = ma (Newton's second law);
   (3) Energy is conserved.

3. What is the philosophy followed by the basic equations of fluid motion?
   (1) Choose the appropriate fundamental physical principles from the laws of physics, such as
      (a) Mass is conserved.
      (b) F = ma (Newton's 2nd Law).
      (c) Energy is conserved.
   (2) Apply these physical principles to a suitable model of the flow.
   (3) From this application, extract the mathematical equations which embody such physical principles.
4. Define control volume?
   A closed volume drawn within a finite region of the flow. This volume is defines as a control volume, V.

5. Define control surface?
   It is the closed surface which bounds the volume. The control volume may be fixed in space with the fluid moving through it; it is define as a control surface, S.



6. Define substantial derivation.

$$\frac{D}{Dt} \equiv \frac{\partial}{\partial t} + \left(\vec{V} \cdot \nabla\right)$$

D/Dt is the substantial derivative, which is physically the time rate of change following a moving fluid element.

7. Define local derivative

   ∂/∂t is called the local derivative, which is physically the time rate of change at a fixed point.

8. Define convective derivative

   V・ Δ is called the convective derivative, which is physically the time rate of change due to the movement of the fluid element from one location to another in the flow field where the flow properties are spatially different.

9. Write the complete Navier–Stokes equations in conservation form.

$$\begin{aligned}
&\frac{\partial(\rho u)}{\partial t}+\frac{\partial(\rho u^2)}{\partial x}+\frac{\partial(\rho uv)}{\partial y}+\frac{\partial(\rho uw)}{\partial z}\\
&\quad=-\frac{\partial p}{\partial x}+\frac{\partial}{\partial x}\left(\lambda\nabla\cdot\vec{V}+2\mu\frac{\partial u}{\partial x}\right)+\frac{\partial}{\partial y}\left[\mu\left(\frac{\partial v}{\partial x}+\frac{\partial u}{\partial y}\right)\right]\\
&\qquad+\frac{\partial}{\partial z}\left[\mu\left(\frac{\partial u}{\partial z}+\frac{\partial w}{\partial x}\right)\right]+\rho f_x\\
&\frac{\partial(\rho v)}{\partial t}+\frac{\partial(\rho uv)}{\partial x}+\frac{\partial(\rho v^2)}{\partial y}+\frac{\partial(\rho vw)}{\partial z}\\
&\quad=-\frac{\partial p}{\partial y}+\frac{\partial}{\partial x}\left[\mu\left(\frac{\partial v}{\partial x}+\frac{\partial u}{\partial y}\right)\right]+\frac{\partial}{\partial y}\left(\lambda\nabla\cdot\vec{V}+2\mu\frac{\partial v}{\partial y}\right)\\
&\qquad+\frac{\partial}{\partial z}\left[\mu\left(\frac{\partial w}{\partial y}+\frac{\partial v}{\partial z}\right)\right]+\rho f_y\\
&\frac{\partial(\rho w)}{\partial t}+\frac{\partial(\rho uw)}{\partial x}+\frac{\partial(\rho vw)}{\partial y}+\frac{\partial(\rho w^2)}{\partial z}\\
&\quad=-\frac{\partial p}{\partial z}+\frac{\partial}{\partial x}\left[\mu\left(\frac{\partial u}{\partial z}+\frac{\partial w}{\partial x}\right)\right]+\frac{\partial}{\partial y}\left[\mu\left(\frac{\partial w}{\partial y}+\frac{\partial v}{\partial z}\right)\right]\\
&\qquad+\frac{\partial}{\partial z}\left(\lambda\nabla\cdot\vec{V}+2\mu\frac{\partial w}{\partial z}\right)+\rho f_z
\end{aligned}$$

10. Write the conservation form of the energy equation, written in terms of the internal energy.

$$\begin{aligned}
\frac{\partial(\rho e)}{\partial t}+\nabla\cdot(\rho e\vec{V})=\rho\dot{q}&+\frac{\partial}{\partial x}\left(k\frac{\partial T}{\partial x}\right)+\frac{\partial}{\partial y}\left(k\frac{\partial T}{\partial y}\right)\\
&+\frac{\partial}{\partial z}\left(k\frac{\partial T}{\partial z}\right)-p\left(\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}+\frac{\partial w}{\partial z}\right)\\
&+\lambda\left(\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}+\frac{\partial w}{\partial z}\right)^2+\mu\left[2\left(\frac{\partial u}{\partial x}\right)^2\right.\\
&+2\left(\frac{\partial v}{\partial y}\right)^2+2\left(\frac{\partial w}{\partial z}\right)^2+\left(\frac{\partial u}{\partial y}+\frac{\partial v}{\partial x}\right)^2\\
&\left.+\left(\frac{\partial u}{\partial z}+\frac{\partial w}{\partial x}\right)^2+\left(\frac{\partial v}{\partial z}+\frac{\partial w}{\partial y}\right)^2\right]
\end{aligned}$$

11. Write the conservation form of the energy equation, written in terms of the total energy, (e+V^2/2).

$$\begin{aligned}&\frac{\partial}{\partial t}\left[\rho\left(e+\frac{V^2}{2}\right)\right]+\nabla\cdot\left[\rho\left(e+\frac{V^2}{2}\vec{V}\right)\right]\\&\quad=\rho\dot{q}+\frac{\partial}{\partial x}\left(k\frac{\partial T}{\partial x}\right)+\frac{\partial}{\partial y}\left(k\frac{\partial T}{\partial y}\right)\\&\qquad+\frac{\partial}{\partial z}\left(k\frac{\partial T}{\partial z}\right)-\frac{\partial(up)}{\partial x}-\frac{\partial(vp)}{\partial y}-\frac{\partial(wp)}{\partial z}+\frac{\partial(u\tau_{xx})}{\partial x}\\&\qquad+\frac{\partial(u\tau_{yx})}{\partial y}+\frac{\partial(u\tau_{zx})}{\partial z}+\frac{\partial(v\tau_{xy})}{\partial x}+\frac{\partial(v\tau_{yy})}{\partial y}+\frac{\partial(v\tau_{zy})}{\partial z}\\&\qquad+\frac{\partial(w\tau_{xz})}{\partial x}+\frac{\partial(w\tau_{yz})}{\partial y}+\frac{\partial(w\tau_{zz})}{\partial z}+\rho\vec{f}\cdot\vec{V}\end{aligned}$$

.

12. Write the Continuity equations for viscous flow.

Non-conservation form

$$\frac{D\rho}{Dt}+\rho\nabla\cdot\vec{V}=0$$

Conservation form

$$\frac{\partial\rho}{\partial t}+\nabla\cdot(\rho\vec{V})=0$$

13. Write the Momentum equations for viscous flow.

Non-conservation form

$$\begin{aligned}&x\text{-component :}\quad \rho\frac{Du}{Dt}=-\frac{\partial p}{\partial x}+\frac{\partial\tau_{xx}}{\partial x}+\frac{\partial\tau_{yx}}{\partial y}+\frac{\partial\tau_{zx}}{\partial z}+\rho f_x\\&y\text{-component :}\quad \rho\frac{Dv}{Dt}=-\frac{\partial p}{\partial y}+\frac{\partial\tau_{xy}}{\partial x}+\frac{\partial\tau_{yy}}{\partial y}+\frac{\partial\tau_{zy}}{\partial z}+\rho f_y\\&z\text{-component :}\quad \rho\frac{Dw}{Dt}=-\frac{\partial p}{\partial z}+\frac{\partial\tau_{xz}}{\partial x}+\frac{\partial\tau_{yz}}{\partial y}+\frac{\partial\tau_{zz}}{\partial z}+\rho f_z\end{aligned}$$

Conservation form

$$\begin{aligned}&x\text{-component :}\quad \frac{\partial(\rho u)}{\partial t}+\nabla\cdot(\rho u\vec{V})=-\frac{\partial p}{\partial x}+\frac{\partial\tau_{xx}}{\partial x}+\frac{\partial\tau_{yx}}{\partial y}+\frac{\partial\tau_{zx}}{\partial z}+\rho f_x\\&y\text{-component :}\quad \frac{\partial(\rho v)}{\partial t}+\nabla\cdot(\rho v\vec{V})=-\frac{\partial p}{\partial y}+\frac{\partial\tau_{xy}}{\partial x}+\frac{\partial\tau_{yy}}{\partial y}+\frac{\partial\tau_{zy}}{\partial z}+\rho f_y\\&z\text{-component :}\quad \frac{\partial(\rho w)}{\partial t}+\nabla\cdot(\rho w\vec{V})=-\frac{\partial p}{\partial z}+\frac{\partial\tau_{xz}}{\partial x}+\frac{\partial\tau_{yz}}{\partial y}+\frac{\partial\tau_{zz}}{\partial z}+\rho f_z\end{aligned}$$

14. Write the Energy equations for viscous flow.

Non-conservation form

$$\begin{aligned}\rho\frac{D}{Dt}\left(e+\frac{V^2}{2}\right)&=\rho\dot{q}+\frac{\partial}{\partial x}\left(k\frac{\partial T}{\partial x}\right)+\frac{\partial}{\partial y}\left(k\frac{\partial T}{\partial y}\right)+\frac{\partial}{\partial z}\left(k\frac{\partial T}{\partial z}\right)\\&-\frac{\partial(up)}{\partial x}-\frac{\partial(vp)}{\partial y}-\frac{\partial(wp)}{\partial z}+\frac{\partial(u\tau_{\mathrm{xx}})}{\partial x}\\&+\frac{\partial(u\tau_{\mathrm{yx}})}{\partial y}+\frac{\partial(u\tau_{\mathrm{zx}})}{\partial z}+\frac{\partial(v\tau_{\mathrm{xy}})}{\partial x}+\frac{\partial(v\tau_{\mathrm{yy}})}{\partial y}\\&+\frac{\partial(v\tau_{\mathrm{zy}})}{\partial z}+\frac{\partial(w\tau_{\mathrm{xz}})}{\partial x}+\frac{\partial(w\tau_{\mathrm{yz}})}{\partial y}+\frac{\partial(w\tau_{\mathrm{zz}})}{\partial z}+\rho\vec{f}\cdot\vec{V}\end{aligned}$$

Conservation form

$$\begin{aligned}&\frac{\partial}{\partial t}\left[\rho\left(e+\frac{V^2}{2}\right)\right]+\nabla\cdot\left[\rho\left(e+\frac{V^2}{2}\vec{V}\right)\right]\\&=\rho\dot{q}+\frac{\partial}{\partial x}\left(k\frac{\partial T}{\partial x}\right)+\frac{\partial}{\partial y}\left(k\frac{\partial T}{\partial y}\right)\\&+\frac{\partial}{\partial z}\left(k\frac{\partial T}{\partial z}\right)-\frac{\partial(up)}{\partial x}-\frac{\partial(vp)}{\partial y}-\frac{\partial(wp)}{\partial z}+\frac{\partial(u\tau_{\mathrm{xx}})}{\partial x}\\&+\frac{\partial(u\tau_{\mathrm{yx}})}{\partial y}+\frac{\partial(u\tau_{\mathrm{zx}})}{\partial z}+\frac{\partial(v\tau_{\mathrm{xy}})}{\partial x}+\frac{\partial(v\tau_{\mathrm{yy}})}{\partial y}\\&+\frac{\partial(v\tau_{\mathrm{zy}})}{\partial z}+\frac{\partial(w\tau_{\mathrm{xz}})}{\partial x}+\frac{\partial(w\tau_{\mathrm{yz}})}{\partial y}+\frac{\partial(w\tau_{\mathrm{zz}})}{\partial z}+\rho\vec{f}\cdot\vec{V}\end{aligned}$$

15. Write the Continuity equations for inviscous flow.

Non-conservation form

$$\frac{D\rho}{Dt}+\rho\nabla\cdot\vec{V}=0$$

Conservation form

$$\frac{\partial\rho}{\partial t}+\nabla\cdot(\rho\vec{V})=0$$

16. Write the Momentum equations for viscous flow.

Non-conservation form

$$\begin{aligned}&x\text{-component}: &\rho\frac{Du}{Dt}&=-\frac{\partial p}{\partial x}+\rho f_{\mathrm{x}}\\&y\text{-component}: &\rho\frac{Dv}{Dt}&=-\frac{\partial p}{\partial y}+\rho f_{\mathrm{y}}\\&z\text{-component}: &\rho\frac{Dw}{Dt}&=-\frac{\partial p}{\partial z}+\rho f_{\mathrm{z}}\end{aligned}$$

Conservation form

$$x\text{-component}: \quad \frac{\partial(\rho u)}{\partial t} + \nabla \cdot (\rho u \vec{V}) = -\frac{\partial p}{\partial x} + \rho f_x$$

$$y\text{-component}: \quad \frac{\partial(\rho v)}{\partial t} + \nabla \cdot (\rho v \vec{V}) = -\frac{\partial p}{\partial y} + \rho f_y$$

$$z\text{-component}: \quad \frac{\partial(\rho w)}{\partial t} + \nabla \cdot (\rho w \vec{V}) = -\frac{\partial p}{\partial z} + \rho f_z$$

17. Write the Energy equations for viscous flow.

Non-conservation form

$$\rho \frac{D}{Dt}\left(e + \frac{V^2}{2}\right) = p\dot{q} - \frac{\partial(up)}{\partial x} - \frac{\partial(vp)}{\partial y} - \frac{\partial(wp)}{\partial z} + \rho \vec{f} \cdot \vec{V}$$

Conservation form

$$\frac{\partial}{\partial t}\left[\rho\left(e + \frac{V^2}{2}\right)\right] + \nabla \cdot \left[\rho\left(e + \frac{V^2}{2}\right)\vec{V}\right] = \rho\dot{q} - \frac{\partial(up)}{\partial x} - \frac{\partial(vp)}{\partial y} - \frac{\partial(wp)}{\partial z} + \rho \vec{f} \cdot \vec{V}$$

18. Write the Comments on the Governing Equations of CFD.

(1) They are a coupled system of non-linear partial differential equations, and hence are very difficult to solve analytically. To date, there is no general closed-form solution to these equations.

(2) For the momentum and energy equations, the difference between the non-conservation and conservation forms of the equations is just the left-hand side.

The right-hand side of the equations in the two different forms is the same.

(3) Note that the conservation form of the equations contain terms on the left-hand side which include the divergence of some quantity, such as Δ・(ρ_V), Δ・(ρu_V), etc. For this reason, the conservation form of the governing equations is sometimes called the divergence form.

(4) The normal and shear stress terms in these equations are functions of the velocity gradients.

(5) The system contains five equations in terms of six unknown flow-field variables, ρ, p, u, v, w, e. In aerodynamics, it is generally reasonable to assume the gas is a perfect gas (which assumes that intermolecular forces are negligible. For a perfect gas, the equation of state is

$$p = \rho RT$$

where R is the specific gas constant. This provides a sixth equation, but it also introduces a seventh unknown, namely temperature, T. A seventh equation to close the entire system must be a thermodynamic relation between state variables.

For example,

$$e = e(T, p)$$

For a calorically perfect gas (constant specific heats), this relation would be

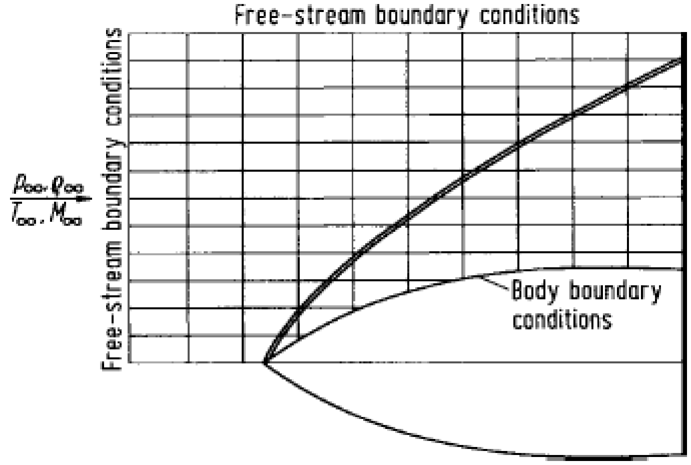
$e = C_vT$

where C_v is the specific heat at constant volume.

(6) The momentum equations for a viscous flow were identified as the Navier–Stokes equations, which is historically accurate. However, in the modern CFD literature, this terminology has been expanded to include the entire system of flow equations for the solution of a viscous flow continuity and energy as well as momentum. Therefore, when the computational fluid dynamic literature discusses a numerical solution to the 'complete Navier–Stokes equations', it is usually referring to a numerical solution of the complete system of equations. In this sense, in the CFD literature, a 'Navier–Stokes solution' simply means a solution of a viscous flow problem using the full governing equations.

19. Define shock-capturing approach (or) Example for implicitly method.

Many computations of flows with shocks are designed to have the shock waves appear naturally within the computational space as a direct result of the overall flow field solution, i.e. as a direct result of the general algorithm, without any special treatment to take care of the shocks themselves. Such approaches are called shock capturing methods.

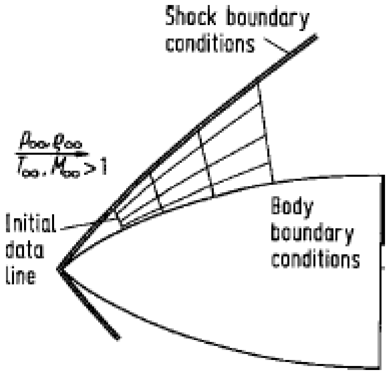


Mesh for the shock-capturing approach

20. Define shock-fitting method? (or) Example of explicitly method?

shock waves are explicitly introduced into the flow-field solution, the exact Rankine–Hugoniot relations for changes across a shock are used to relate the flow immediately ahead of and behind the shock, and the governing flow equations are used to calculate the remainder of the flow field. This approach is called the shock-fitting method.

Mesh for the shock-fitting approach



21. What are the advantages of shock –capturing method?

The shock-capturing method is ideal for complex flow problems involving shock waves for which we do not know either the location or number of shocks. Here, the shocks simply form within the computational domain as nature would have it. Moreover, this takes place without requiring any special treatment of the shock within the algorithm, and hence simplifies the computer programming.

22. What are the disadvantages of shock –capturing method?

A disadvantage of this approach is that the shocks are generally smeared over a number of grid points in the computational mesh, and hence the numerically obtained shock thickness bears no relation what-so-ever to the actual physical shock thickness, and the precise location of the shock discontinuity is uncertain within a few mesh sizes.

23. What are the advantages of shock –fitting method?

The advantage of the shock-fitting method is that the shock is always treated as a discontinuity, and its location is well-defined numerically. However, for a given problem you have to know in advance approximately where to put the shock waves, and how many there are.

24. What are the disadvantages of shock –fitting method?

For complex flows, shock –fitting method can be a distinct disadvantage

25. What is panel method?

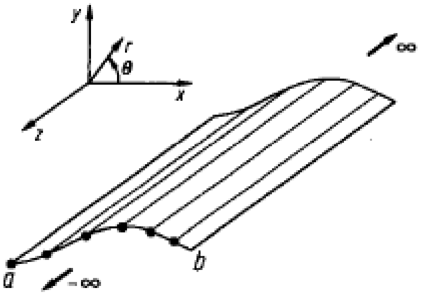
Panel methods are numerical methods which require a high-speed digital computer for their implementation; therefore we include panel methods as part of the overall structure of computational fluid dynamics.

26. What are the types of panel method?

a. Source panel method
b. Vortex panel method
c. Doublet panel method

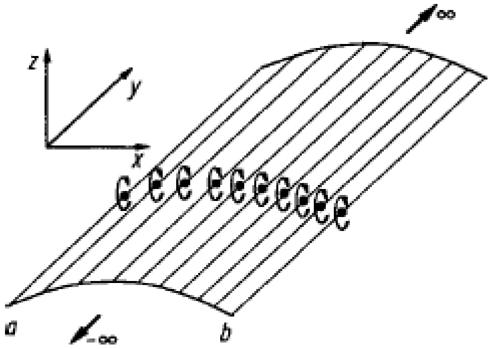
27. Define source sheet.

Imagine that we have an infinite number of such line sources side-by-sides, where the strength of each line source is infinitesimally small. These side-by-side line sources form a source sheet, as shown in perspective in the Fig.



28. Define vortex sheet.

Imagine an infinite number of straight vortex filaments side by side, where the strength of each filament is infinitesimally small. These side-by-side vortex filaments form a vortex sheet, as shown in perspective in the Fig.



29. Define Over-determined system.

The system having “n” equation and “n+1” unknown’s means that system is called over determined system.

30. Write the crux of the source panel method?

$$\frac{\lambda_i}{2} + \sum_{\substack{j=1 \\ (j\neq i)}}^{n} \frac{\lambda_j}{2\pi} \int_j \frac{\partial}{\partial n_i}(\ln\ r_{ij})\, ds_j + V_\infty \cos\beta_i = 0$$

Above equation is the crux of the source panel method.

30. Write the crux of the vortex panel method?

$$V_\infty \cos\beta_\mathrm{i} - \sum_{j=1}^{n} \frac{\gamma_\mathrm{j}}{2\pi} \int_j \frac{\partial \theta_\mathrm{ij}}{\partial n_\mathrm{i}} \, \mathrm{d}s_\mathrm{j} = 0$$

Above equation is the crux of the vortex panel method.

31. Write the Classification of Partial Differential Equations.

    (1) Hyperbolic Partial Differential Equations

    (2) Parabolic Partial Differential Equations

    (3) Elliptic Partial Differential Equations

32. Explain the Classification of Partial Differential Equations

Consider the system of quasilinear equations given below.

$$a_1 \frac{\partial u}{\partial x} + b_1 \frac{\partial u}{\partial y} + c_1 \frac{\partial v}{\partial x} + d_1 \frac{\partial v}{\partial y} = f_1$$

$$a_2 \frac{\partial u}{\partial x} + b_2 \frac{\partial u}{\partial y} + c_2 \frac{\partial v}{\partial x} + d_2 \frac{\partial v}{\partial y} = f_2$$

………………..(1)

$$\text{since } u = u(x,\ y): \ \mathrm{d}u = \frac{\partial u}{\partial x}\mathrm{d}x + \frac{\partial u}{\partial y}\mathrm{d}y$$

$$\text{since } v = v(x,\ y): \ \mathrm{d}v = \frac{\partial v}{\partial x}\mathrm{d}x + \frac{\partial v}{\partial y}\mathrm{d}y$$

…………..(2)

In matrix form of above equation

$$\begin{bmatrix} a_1 & b_1 & c_1 & d_1 \\ a_2 & b_2 & c_2 & d_2 \\ \mathrm{d}x & \mathrm{d}y & 0 & 0 \\ 0 & 0 & \mathrm{d}x & \mathrm{d}y \end{bmatrix} \begin{bmatrix} \partial u/\partial x \\ \partial u/\partial y \\ \partial v/\partial x \\ \partial v/\partial y \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \\ \mathrm{d}u \\ \mathrm{d}v \end{bmatrix}$$

……………...(3)

$$\begin{vmatrix} a_1 & b_1 & c_1 & d_1 \\ a_2 & b_2 & c_2 & d_2 \\ \mathrm{d}x & \mathrm{d}y & 0 & 0 \\ 0 & 0 & \mathrm{d}x & \mathrm{d}y \end{vmatrix} = 0$$

……………..(4)

$$(a_1c_2 - a_2c_1)(\mathrm{d}y)^2 - (a_1d_2 - a_2d_1 + b_1c_2 - b_2c_1)(\mathrm{d}x)(\mathrm{d}y) + (b_1d_2 - b_2d_1)(\mathrm{d}x)^2 = 0$$

$$(a_1c_2 - a_2c_1)\left(\frac{\mathrm{d}y}{\mathrm{d}x}\right)^2 - (a_1d_2 - a_2d_1 + b_1c_2 - b_2c_1)\frac{\mathrm{d}y}{\mathrm{d}x} + (b_1d_2 - b_2d_1) = 0$$

…(5)

$$a = (a_1c_2 - a_2c_1)$$
$$b = -(a_1d_2 - a_2d_1 + b_1c_2 - b_2c_1)$$
$$c = (b_1d_2 - b_2d_1)$$

$$a\left(\frac{\mathrm{d}y}{\mathrm{d}x}\right)^2 + b\left(\frac{\mathrm{d}y}{\mathrm{d}x}\right) + c = 0$$

………………..(6)

$$\frac{\mathrm{d}y}{\mathrm{d}x} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}, \qquad D = b^2 - 4ac$$

The characteristic lines may be real and distinct, real and equal, or imaginary, depending on the value of D. Specifically:

If D > 0: Two real and distinct characteristics exist through each point in the xy- plane. When this is the case, the system of equations given by
Eqs. (1) is called hyperbolic.
If D = 0: One real characteristic exists. Here, the system of Eqs. (1) is called parabolic.
If D < 0: The characteristic lines are imaginary. Here, the system of Eqs. (1) is called elliptic.

33. Define Well-Posed Problems.

In the solution of partial differential equations it is sometimes easy to attempt a solution using incorrect or insufficient boundary and initial conditions. Whether the solution is being attempted analytically or numerically, such an ‘ill-posed’ problem will usually lead to spurious results.

Therefore, we define a well-posed problem as follows: If the solution to a partial differential equation exists and is unique, and if the solution depends continuously upon the initial and boundary conditions, then the problem is well-posed. In CFD, it is important that you establish that your problem is well-posed before you attempt to carry out a numerical solution.

34. Define grid point.

Analytical solutions of partial differential equations involve closed-form expressions which give the variation of the dependent variables continuously throughout the domain. In contrast, numerical solutions can give answers at only discrete points in the domain, called grid points.

35. What are error influence numerical solutions the PDE?

a. Discretization error
b. Round-off error

36. Define Discretization error.

The difference between the exact analytical solution of the partial differential equation and the exact solution of the corresponding difference equation. Discretization error for the difference is simply the truncation error for the difference equation plus any errors introduced by the numerical trement of the boundary condition.

37. Define Round-off error.

The numerical error introduced after a repetitive number of calculation in which the computer is constantly rounding the number to some significant figure.

38. Classify the type of grid generation.

a. structured,
b. unstructured

39. How to reduce the truncation error?

The truncation error can be reduced by:

(a) Carrying more terms in the Taylor's series, This leads to higher-order accuracy in the representation of ui+1,j.
(b) Reducing the magnitude of Δx

40. Write the advantage of explicit approach.

Relatively simple to set up and program.

41. Write the Disadvantage of the explicit approach.

Given Δx, Δt must be less than some limit imposed by stability constraints. In many cases, Δt must be very small to maintain stability; this can result in long computer running times to make calculations over a given interval of t.

42. Write the advantage of implicit approach.

Stability can be maintained over much larger values of Δt, hence using considerably fewer time steps to make calculations over a given interval of t. This results in less computer time.

42. Write the advantage of implicit approach.

a. More complicated to set up and program.
b. Since massive matrix manipulations are usually required at each time step, the computer time per time step is much larger than in the explicit approach.
c. Since large Δt can be taken, the truncation error is larger, and the use of implicit methods to follow the exact transients (time variations of the independent variable) may not be as

accurate as an explicit approach. However, for a time-dependent solution in which the steady state is the desired result, this relative time-wise inaccuracy is not important.

43. What is Lax method?

$$\frac{\partial u}{\partial t} = \frac{u_i^{n+1} - \frac{1}{2}(u_{i+1}^n + u_{i+1}^n)}{\Delta t} \qquad \text{……….(a)}$$

The differencing used in the above equation, where Eq. (a) is used to represent the time derivative, is called the Lax method.

44. Define Courant number. (or) What is the important stability criterion for hyperbolic equation?

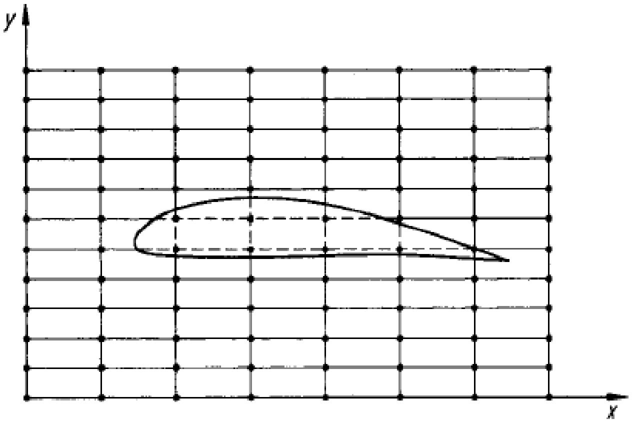
$$C = c\frac{\Delta t}{\Delta x} \leq 1$$

C is called the Courant number. This equation says that $\Delta t \leq \Delta x/c$ for the numerical solution of Eq. (5.45) to be stable. Moreover, Eq. (5.47) is called the Courant–Friedrichs–Lewy condition, generally written as the CFL condition. It is an important stability criterion for hyperbolic equations.

45. Explain the problems with rectangular grid.

(1) Some grid points fall inside the airfoil, where they are completely out of the flow. What values of the flow properties do we ascribe to these points?

(2) There are few, if any, grid points that fall on the surface of the airfoil. This is not good, because the airfoil surface is a vital boundary condition for the determination of the flow, and hence the airfoil surface must be clearly and strongly seen by the numerical solution.



46. Define physical plane and computational plane (or) What is one-to-one correspondence?

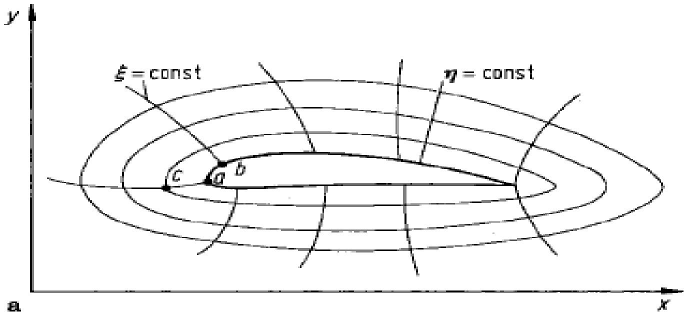


Fig. a. Physical plane

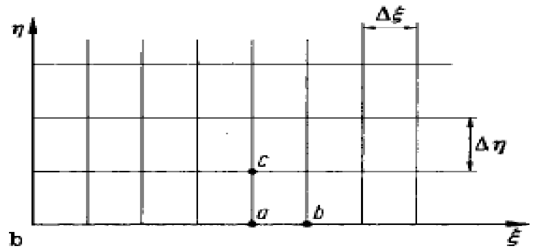


Fig. b. computational plane

This is shown in Fig. b, which illustrates a rectangular grid in terms of ξ and η. The rectangular mesh shown in Fig. b is called the computational plane.

There is a one-to-one correspondence between this mesh, and the curvilinear mesh in Fig. a, called the physical plane.

For example, points a, b and c in the physical plane (Fig.a) correspond to points a, b and c in the computational plane, which involves uniform Δξ and uniform Δη. The computed information is then transferred back to the physical plane. Moreover, when the governing equations are solved in the computational space, they must be expressed in terms of the variables ξ and η rather than x and y; i.e., the governing equations must be transformed from (x, y) to (ξ, η) as the new independent variables.

47. Define metrics

The terms involving the geometry of the grids, such as $\partial\xi/\partial x$, $\partial\xi/\partial y$, $\partial\eta/\partial x$, $\partial\eta/\partial y$, etc., are called metrics.

48. Write the Jacobian determinant

Jacobian determinant denoted by J

$$J \equiv \frac{\partial(x,y)}{\partial(\xi,n)} \equiv \begin{vmatrix} \dfrac{\partial x}{\partial \xi} & \dfrac{\partial y}{\partial \xi} \\ \dfrac{\partial x}{\partial \eta} & \dfrac{\partial y}{\partial \eta} \end{vmatrix}$$

49. Define Adaptive Grids

An adaptive grid is a grid network that automatically clusters grid points in regions of high flow field gradients; it uses the solution of the flow field properties to locate the grid points in the physical plane.

The adaptive grid evolves in steps of time in conjunction with a time-dependent solution of the governing flow field equations, which computes the flow field variables in steps of time. During the course of the solution, the grid points in the physical plane move in such a fashion to 'adapt' to regions of large flow field radients. Hence, the actual grid points in the physical plane are constantly in motion during the solution of the flow field, and become stationary only when the flow solution approaches a steady state. Where the generation of the grid is completely separate from the flow field solution, an adaptive grid is intimately linked to the flow field solution, and changes as the flow field changes.

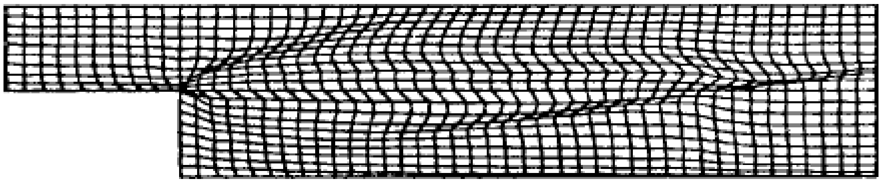


Fig. Adapted grid for the rearward-facing step problem

50. Write the advantages of adaptive grid

An adaptive grid is expected because the grid points are clustered in regions where the 'action' is occurring.

These advantages are:

(1) Increased accuracy for a fixed number of grid points, or
(2) For a given accuracy, fewer grid points are needed.

51. What are the topologies used in grid generation?

C-Grid Topology
O-Grid Topology
H-Grid Topology

51. Define Unstructured Grids

Unstructured grids are typically composed of triangles in 2D and of tetrahedral in **3D.**

52. Define mixed element grids

Nowadays it becomes increasingly popular to build unstructured grids from various element types. For example, hexahedra or prisms are employed to discretise boundary layers. The rest of the flow domain is filled with tetrahedra. Pyramids are used as transitional elements between the hexahedra or the prisms and the tetrahedral. Hence the name mixed element grids.

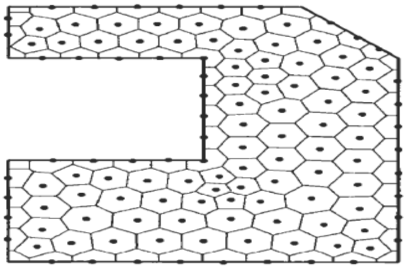
53. Write the unstructured grid generation methodology.

Unstructured grid generation methodologies for CFD applications are mostly based on either an

(1) Delaunay, method. or
(2) Advancing-front method.

54. Define Dirichlet tessellatipn or Voronoj diagram.

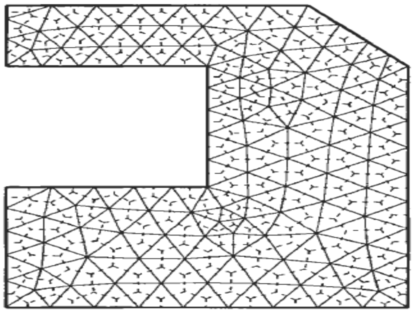
The Delaunay triangulation is based on a methodology proposed by Dirichlet in 1850 for the unique subdivision of space into a set of packed convex regions. Given a set of points, each region represents the space around the particular point, which is closer to that point than to any other. The regions form polygons (polyhedra in 3D) which are known **as** the **Dirichlet tessellation** or the **Voronoj diagram .**



the Voronoj diagram

55. Define Delaunay triangulation.

If we connect point pairs which share some segment(face) of the Voronoj diagram by straight lines, we obtain the Delaunay triangulation. The triangulation defines a set of triangles (tetrahedra in 3D),which cover the convex hull of the points. This is displayed in Fig. a. The DeIaunay triangulation is the dual of the Voronoj diagram. The nodes of the Voronoj polygons are in 2D the centres of circumcircles of the triangles. In 3D, the nodes represent the centres of circumspheres of the tetrahedra. This implies that the circumcircle of every triangle (circumsphere of every tetrahedron) contains no point from the set in its interior.



The Delaunay triangulation

56. Write the 'Prandtl's boundary layer equations'.

$$u\frac{\partial u}{\partial x}+v\frac{\partial u}{\partial y}=-\frac{\partial p}{\partial x}+\frac{1}{\mathrm{Re}}\frac{\partial^2 u}{\partial y^2}$$

$$\frac{\partial p}{\partial y}=0$$

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

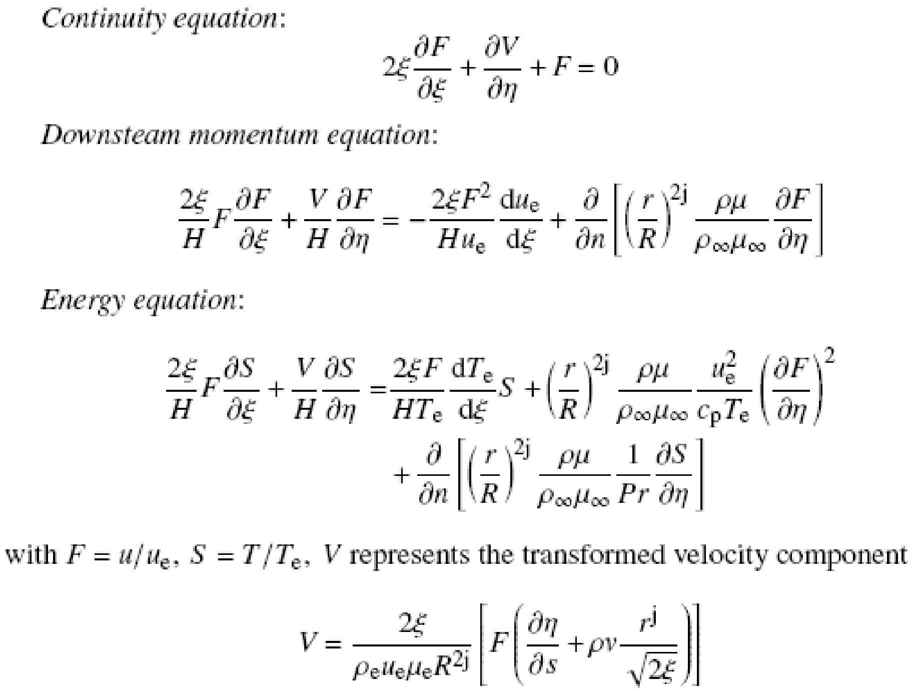
The boundary conditions are:
On the surface:

$$y=0 \qquad u=0, \qquad v=0$$

On the outer edge of the boundary layer:

$$y=\delta=\frac{\bar{\delta}}{L} \qquad u=U(x)$$

57. Write Transformation of the Boundary Layer Equations.



Continuity equation:

$$2\xi\frac{\partial F}{\partial\xi}+\frac{\partial V}{\partial\eta}+F=0$$

Downsteam momentum equation:

$$\frac{2\xi}{H}F\frac{\partial F}{\partial\xi}+\frac{V}{H}\frac{\partial F}{\partial\eta}=-\frac{2\xi F^2}{Hu_\mathrm{e}}\frac{\mathrm{d}u_\mathrm{e}}{\mathrm{d}\xi}+\frac{\partial}{\partial n}\left[\left(\frac{r}{R}\right)^{2\mathrm{j}}\frac{\rho\mu}{\rho_\infty\mu_\infty}\frac{\partial F}{\partial\eta}\right]$$

Energy equation:

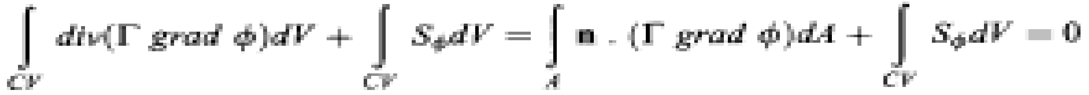
$$\frac{2\xi}{H}F\frac{\partial S}{\partial\xi}+\frac{V}{H}\frac{\partial S}{\partial\eta}=\frac{2\xi F}{HT_\mathrm{e}}\frac{\mathrm{d}T_\mathrm{e}}{\mathrm{d}\xi}S+\left(\frac{r}{R}\right)^{2\mathrm{j}}\frac{\rho\mu}{\rho_\infty\mu_\infty}\frac{u_\mathrm{e}^2}{c_\mathrm{p}T_\mathrm{e}}\left(\frac{\partial F}{\partial\eta}\right)^2$$
$$+\frac{\partial}{\partial n}\left[\left(\frac{r}{R}\right)^{2\mathrm{j}}\frac{\rho\mu}{\rho_\infty\mu_\infty}\frac{1}{Pr}\frac{\partial S}{\partial\eta}\right]$$

with $F=u/u_\mathrm{e}$, $S=T/T_\mathrm{e}$, V represents the transformed velocity component

$$V=\frac{2\xi}{\rho_\mathrm{e}u_\mathrm{e}\mu_\mathrm{e}R^{2\mathrm{j}}}\left[F\left(\frac{\partial\eta}{\partial s}+\rho v\frac{r^\mathrm{j}}{\sqrt{2\xi}}\right)\right]$$

e denotes the values at the outer edge of the boundary layer flow and R denotes the local radius of a body of revolution.

58. Write the governing equation of steady diffusion.



$$\int_{CV}div(\Gamma\ grad\ \phi)dV+\int_{CV}S_\phi dV=\int_{A}\mathbf{n}\cdot(\Gamma\ grad\ \phi)dA+\int_{CV}S_\phi dV=0$$

59. What is weak instability?

The results of the calculation are displayed in Fig.a. One notices that the perturbation on u1 gives rise to amplifying oscillations. In fact, as small as the initial perturbation may be - and there will always be one because of round off errors – it will eventually lead to an explosion of the numerical solution. This phenomenon is clearly inacceptable. It is named weak instability.

60. Define finite element method.

The finite element method (FEM) is a numerical technique for solving partial differential equations (PDE’s).

61. Write the essential characteristic of FEM in CFD.

Its first essential characteristic is that the continuum field, or domain, is subdivided into cells, called elements, which form a grid. The elements (in 2D) have a triangular or a quadrilateral form and can be rectilinear or curved. The grid itself need not be structured. With unstructured grids and curved cells, complex geometries can be handled with ease.

The second essential characteristic of the FEM is that the solution of the discrete problem is assumed a priori to have a prescribed form. The solution has to belong to a function space, which is built by varying function values in a given way, for instance linearly or quadratically, between values in nodal points. The nodal points, or nodes, are typical points of the elements such as vertices, mid-side points, mid element points, etc.

The third essential characteristic is that a FEM does not look for the solution of the PDE itself, but looks for a solution of an integral form of the PDE. The most general integral form is obtained from a weighted residual formulation. By this formulation the method acquires the ability to naturally incorporate differential type boundary conditions and allows easily the construction of higher order accurate methods. The ease in obtaining higher order accuracy and the ease of implementation of boundary conditions form a second important advantage of the FEM. With respect to accuracy. The FEM is superior to the FVM, where higher order accurate formulations are quite complicated.

A final essential characteristic of the FEM is the modular way in which the discretization is obtained. The discrete equations are constructed from contributions on the element level which afterwards are assembled.

62. What is the basis of FVM?

It is important that the conservation laws in their integral form are represented accurately. The most natural method to accomplish this is to discretize the integral form of the equations and not the differential form. This is the basis of a finite volume method.

63. Define cell.

The flow field or domain is subdivided, as in the finite element method, into a set of non-overlapping cells that cover the whole domain. In the finite volume Method (FVM) the term cell is used instead of the term element used in the finite Element method (FEM).

64. Define node.

The conservation laws are applied to determine the flow variables in some discrete points of the cells, called nodes. As in the FEM, these nodes are at typical locations of the cells, such as cell-centres, cell-vertices or mid sides

65. Define cell-centres,cell-vertices.

The choice of the nodes can be governed by the wish to represent the solution by an interpolation structure, as in the FEM. A typical choice is then cell-centres for representation as piecewise constant functions or cell-vertices for representation as piecewise linear (or bilinear) functions. However, in the FVM, a function space for the solution need not be defined and nodes can be chosen in a way that does not imply an interpolation structure. Figure shows some typical examples of choices of nodes with the associated definition of variables.

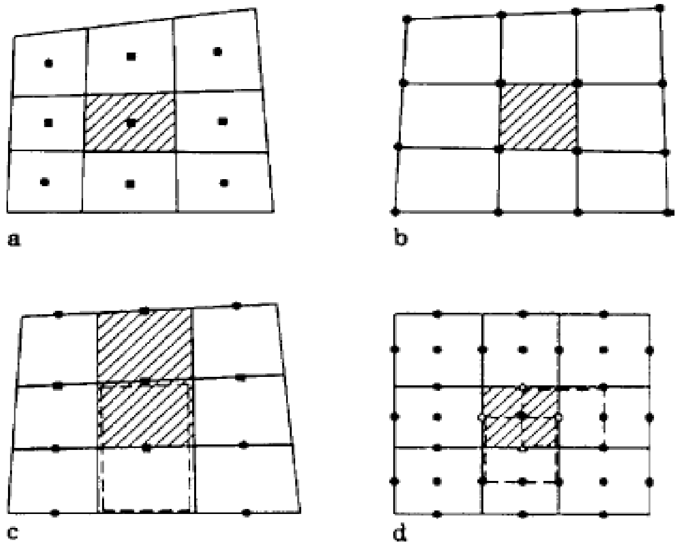


Fig. . Typical choice of nodes in the FVM. The marked nodes are used in the flux balance of the control volume. (**a**): piecewise constant interpolation structure; (**b**): piecewise linear interpolation structure; (**c**): no interpolation structure with all variables defined in each node; (**d**): no interpolation structure with not all variables defined in each node; (Cartesian grid), o: ρ and p,: u, Δ: v

66. Define staggered grid approach.

A typical choice is then cell-centres for representation as piecewise constant functions or cell-vertices for representation as piecewise linear (or bilinear) functions. choices imply an interpolation structure, the last two do not. In the last example, function values are not defined in all nodes. The grid of nodes on which pressure and density are defined is different from the grid of nodes on which velocity-x components and velocity-y components are defined. This approach commonly is called the staggered grid approach.

67. What is the formulation the FVM?

The finite volume method (FVM) tries to combine the best from the finite element method (FEM), i.e. the geometric flexibility, with the best of the finite difference method (FDM), i.e. the flexibility in defining the discrete flow field (discrete values of dependent variables and their associated fluxes).

68. What is Lax-Wendroff time-stepping?

Lax-Wendroff time-stepping is a very classic explicit time integration method in the finite difference method the principles of a Lax-Wendroff method with the use of the one dimensional scalar model equation

$$\frac{\partial u}{\partial t} + \frac{\partial f(u)}{\partial x} = 0$$

69. What is CFD?

CFD is the simulation of fluids engineering systems using modelling (mathematical physical problem formulation) and numerical methods (discretization methods, solvers, numerical parameters, and grid generations, etc.)

Historically only Analytical Fluid Dynamics (AFD) and Experimental Fluid Dynamics (EFD).CFD made possible by the advent of digital computer and advancing with improvements of computer resources (500 flops, 1947□20 teraflops, 2003)

70. Why use CFD?

Simulation-based design instead of "build & test" More cost effective and more rapid than EFD CFD provides high-fidelity database for diagnosing flow field

a. Simulations of physical fluid phenomena are difficult for experiments like Full scale simulations (e.g., ships and airplanes) Environmental effects (wind, weather, etc.) Hazards (e.g., explosions, radiation, pollution)Physics (e.g., planetary boundary layer, stellar evolution)
b. Knowledge and exploration of flow physics and Analysis and Design

71. List out the applications of CFD in industry.

CFD is used in Aerospace, Automotive, Biomedical, Chemical, Processing, HVAC, Hydraulics, Marine, Oil & Gas, Power Generation, Sports.

72. What are the Commercial Software used for CFD analysis?

The market is currently dominated by four codes:

1) PHOENICS
2) FLUENT
3) FLOW3D
4) STAR-CD

73. What are the Advantages of CFD over EFD?

- Substantial reduction of lead times and costs of new designs.
- Ability to study systems where controlled experiments are difficult or impossible to perform (e.g. very large systems).
- Ability to study systems under hazardous conditions at and beyond their normal performance limits (e.g. safety studies and accident scenarios).
- Practically unlimited level of detail of results.

Reg. No. : □□□□□□□□□□□

Question Paper Code :

B.E./B.Tech. DEGREE EXAMINATION, NOVEMBER/DECEMBER 2011.

Seventh Semester

Aeronautical Engineering

AE 2402 — COMPUTATIONAL FLUID DYNAMICS

(Regulation 2008)

Time : Three hours Maximum : 100 marks

Answer ALL questions.

PART A — (10 × 2 = 20 marks)

1. What are the physical significance/meaning of the various terms in conservative form of momentum equation?

2. What are limitations of the panel methods?

3. Define

   (a) Convergence, and

   (b) Lax equivalence theorem.

4. Name the important errors that commonly occur in numerical solution.

5. Transform the steady, incompressible continuity equity from the x, y physical plane to the ξ , η computational plane.

6. What is the importance of CFL condition?

7. Compare implicit and explicit methods.

8. What are the different categories of boundary conditions give example for each category?

9. What is the necessity of staggered grid in control volume method?

10. Define Peclet number and state its importance.

PART B — (5 × 16 = 80 marks)

11. (a) (i) What is the need for classification of PDE and how do you classify second order PDE? (8)

(ii) What are the discretization techniques and how do you discretize the equations for subsonic and supersonic flows? (8)

Or

(b) Write down the procedure for the calculation of pressure coefficient distribution around a circular cylinder using the source panel technique.

12. (a) (i) How is conformal mapping of a polygon carried out by Schwarz-Christoffel transformation? (8)

(ii) Illustrate the basic ideas in algebraic transformations of two-dimensional, steady, boundary layer flow over flat plate with suitable transformation relations. (8)

Or

(b) (i) What is the need for grid generation? Mention the different grid generation technique and list down their relative merits and demerits. (6)

(ii) Explain how grid generation is achieved by the numerical solution of elliptic Poison's equations. (10)

13. (a) (i) What is meant by "Wiggles" in the numerical solution? Describe with an example. (6)

(ii) Consider steady 1-D convection-diffusion equation of a property φ

$$\frac{d}{dx}\left(pu\,\phi\right)=\frac{d}{dx}\left(\Gamma\frac{d\phi}{dx}\right)$$

Using control volume approach discritise the above equation and obtain the neighboring coefficients by

(1) Central difference scheme

(2) Upwind differencing scheme. (10)

Or

(b) What is meant by hierarchy of boundary layer equations? Derive Zeroth, first and second order boundary layer equations.

14. (a) Write short notes on the following :

(i) Strong formulation

(ii) Weighted Residual formulation

(iii) Galerkin Formulation and

(iv) Weak formulation (4 × 4 = 16)

Or

(b) Consider a cylindrical fin with uniform cross - sectional area A. The base is at a temperature of 100° C (T_B) and the end is insulated. The fin is exposed to an ambient temperature of 20°C. One-dimensional heat transfer in this situation is governed by

$$\frac{d}{dx}\left(kA\,\frac{dT}{dx}\right)-hP\,(T-T_\infty)=0$$

where h is the convective heat transfer coefficient, P the perimeter, k the thermal conductivity of the material and T_∞ the ambient temperature. Calculate the temperature distribution along the fin using five equally placed control volumes. Take $hP/(kA)=25\,m^2$ (note kA is constant)

15. (a) (i) Explain explicit Lax-Wendroff scheme of time dependent methods. (8)

(ii) Discuss Cell entered formulation in Finite Volume Techniques.(8)

Or

(b) Draw a flow chart and describe SIMPLE algorithm for two-dimensional laminar stead flow equations in Cartesian co-ordinates.

Question Paper Code : D 2531

B.E./B.Tech. DEGREE EXAMINATION, APRIL/MAY 2010.

Eighth Semester

Aeronautical Engineering

ME 1011 — COMPUTATIONAL FLUID DYNAMICS

(Regulation 2004)

Time : Three hours

Maximum : 100 marks

Answer ALL questions.

PART A — (10 × 2 = 20 marks)

1. What is the physical significance of Euler's equations and write the equation in its vector form?

2. What is no-slip condition in incompressible flows?

3. How is Panel method connected with boundary element method?

4. State Lax's equivalence theorem.

5. What is grid independent solution?

6. Why is computer simulation necessary in fluid flow and heat transfer?

7. Explain briefly the disadvantages of using upwind scheme for convective terms.

8. What is Oseen approximation in simplifying Navier-Stokes equations?

9. Discuss the errors in finite element discretization.

10. What are the advantages of finite volume method over finite difference method?

PART B — (5 × 16 = 80 marks)

11. (a) (i) Discuss the mathematical properties of the elliptic, parabolic and hyperbolic differential equations.

(ii) Transform Laplace's equation

$$\frac{\partial^2 \phi}{\partial x^2} + \frac{\partial^2 \phi}{\partial y^2} = 0$$

into generalized coordinates $\xi = \xi(x, y), \eta = \eta(x, y)$ and show that the resulting equation is elliptic.

Or

(b) (i) In each of the following partial differential equations write down its order and linearity.

$$\frac{\partial u}{\partial x} - c\frac{\partial u}{\partial y} = 0 \qquad (1)$$

$$\frac{\partial^2 u}{\partial x^2} - u\frac{\partial u}{\partial y} = 0 \qquad (2)$$

$$\left(\frac{\partial^3 u}{\partial x^3}\right)^2 + \frac{\partial^2 u}{\partial x \partial y} + \frac{\partial u}{\partial y} = 0 \qquad (3)$$

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = v\frac{\partial^2 u}{\partial x^2} \qquad (4)$$

where u is dependent variable and c and v are constants. Explain the difference in linearity in equations (2), (3) and (4) above.

(ii) Classify the following system of partial differential equations :

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

$$u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} + \frac{\partial p}{\partial x} = 0$$

$$u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y} + \frac{\partial p}{\partial y} = 0$$

12. (a) Describe the panel method for solving inviscid incompressible flows.

Or

(b) Simulate the flow around an inclined aerofoil using panel method.

13. (a) (i) Determine the coefficients a, b, c, d in the formula given below :

$$\frac{d^2T}{dx^2} = aT_j + bT_{j+1} + cT_{j+2} + dT_{j+3}.$$

What is the truncation error of this formula?

(ii) How do you determine the accuracy of the discretization process? What are the uses and difficulties of approximating the derivatives with higher order finite difference schemes? How do you overcome these difficulties?

Or

(b) Determine the stability requirement of FTCS explicit method for the following equations using Von Neumann stability analysis.

$$\frac{\partial u}{\partial t} = \alpha \frac{\partial^2 u}{\partial x^2}$$

$$\frac{\partial u}{\partial t} = -\alpha \frac{\partial u}{\partial x} + \alpha \frac{\partial^2 u}{\partial x^2}$$

14. (a) Explain the linear and quadratic interpolation approximating methods in finite element analysis.

Or

(b) Solve the simplified Sturm-Lioville equation :

$$\frac{d^2y}{dx^2} + y = F$$

With boundary conditions $y(0) = 0$ and

$$\frac{dy}{dx}(1) = 0$$

Using Galerkin finite element method.

Reg.No: □□□□□□□□

B.E (FULL TIME) DEGREE EXAMINATION, NOV / DEC 2007.

AERONAUTICAL ENGINEERING

AE471 COMPUTATIONAL FLUID DYNAMICS

VII Semester

(R-2004)

Time: Three hours Maximum : 100 marks

Answer ALL the questions.

PART A — (10 x 2 = 20 marks)

1. What are adaptive grid techniques?

2. Distinguish between structured and unstructured grids.

3. Bring out any two differences between zeroth order and first order boundary layer equations.

4. Explain why transformation of boundary layer equations is needed for many applications of engineering.

5. Give any two important applications of time dependent methods.

6. What are stiff problems in CFD?

7. Mention the salient features of approximate factorization schemes.

8. What is the need for staggered grid approach in finite volume methods?

9. Mention any two types of flux evaluation schemes in finite volume methods.

10. What is the role of shape function in finite element methods?

PART B (5 X 16 = 80 Marks)

11.i. Explain how orthogonality of grid lines can be implemented at the desired boundary segment in numerical grid generation. (6)

ii. Briefly describe how a numerical grid generation can be carried out using a set of Poission's equations, with neat sketches. Give two examples. (10)

PART B — (5 × 16 = 80 marks)

11. (a) Explain mathematical properties of Elliptic, Parabolic and Hyperbolic equations with respect to computational fluid flow.

Or

(b) Using Taylor series expansion, derive first order forward difference for $\partial^2 f / \partial x^2$.

12. (a) Explain with suitable example how a source panel method is used in non lifting flows over arbitrary two dimensional bodies.

Or

(b) Mention the procedure used in vortex panel method for lifting flows over arbitrary two dimensional bodies.

13. (a) Explain the description of Prandtl boundary layer equations.

Or

(b) State and explain the stability properties of explicit and implicit methods.

14. (a) What are Quadrilateral Lagrange Elements and Isoparametric elements in FEM?

Or

(b) What is strong formulation? Explain with the help of one dimensional boundary value problem.

15. (a) What is cell centered formulation? Explain, with the help of using control volume, semi discretization equation $\Omega_{i,j} \, \partial U / \partial t + \int F \cdot nds = 0$.

Or

(b) State and explain the Spurious modes for Runge—Kutta cell vertex formulation in FVM.