
Chapter 11

Numerical Simulation of Inviscid Flows

OBJECTIVES AND GUIDELINES

In this first chapter dedicated to the practical implementation of the simplest of inviscid flows, we will guide you through the various steps required to write a CFD code.

Inviscid flows are modeled in general by the system of Euler equations. Considering flow conditions with uniform inflow, that is an irrotational flow far upstream, we know from inviscid flow theory and Helmholtz theorem, that the flow will remain irrotational everywhere. In other words, it is equivalent to a potential flow. The latter is the highest level of simplification of a flow description, as all the flow variables can be obtained from the single scalar potential function.

The main difference between the potential and the time-dependent Euler equation models, as identified in Chapter 3, lies in their mathematical properties. We have seen, and we refer you to Chapter 3, Sections 3.2.2 and 3.4.1, that the steady potential equation is elliptic at subsonic speeds and hyperbolic in the supersonic range; while the unsteady Euler equations are always hyperbolic in space and time, independently of the flow regime. This has major consequences on the discretization approach, as the schemes for Laplace-like equations of subsonic potential flows, fail when applied to supersonic conditions; while this will not be the case with the Euler equations. Hence, we will focus on time-dependent discretization methods for the Euler equations, looking for the numerical steady solution, as advocated in the previous chapters.

The other important difference is connected to the possible generation of numerical entropy with the Euler equation model, as a consequence of the numerical dissipation of the selected scheme. ***This will allow you to identify one of the most important properties of numerical CFD solutions, namely the ‘visible’ effects of the numerical dissipation.*** We will show that numerical dissipation generates vorticity and entropy and as a consequence regions of entropy increase provide a picture of the influence of numerical dissipation. Since potential flows are by definition irrotational, this *marker* of numerical dissipation is not present with the potential model.

Another important property of flows around solid bodies is the generation of lift and drag, resulting from the balance of pressure and shear stresses on the surface. With inviscid flows, there is no viscous drag and the numerical ‘production’ of drag is also a global marker of the influence of numerical viscosity.

As the Euler equations form the basis for the full Navier–Stokes solutions, particularly at high Reynolds numbers where the flow properties are dominated by convection, the observed properties will be of critical importance when the same discretization of the convective terms is applied to viscous flows. Indeed, when the numerical viscosity is too high, for instance when applying first order schemes on coarse grids, or when the grid resolution or quality is not sufficient with second order

schemes, the effects of this numerical dissipation on the flow behavior can overshadow the effects of the molecular or turbulent viscosity. Therefore, an accurate identification of the level and of the flow areas where this is likely to happen is essential in order to establish the reliability of your numerical results.

It is therefore advocated, when you run a Navier–Stokes simulation, to first perform an Euler simulation of the same test case and on the same grid, looking for the regions influenced by numerical dissipation, by tracking the growth of entropy.

Before starting, we also wish to provide you with some general guidelines on how to structure and organize your program. The key issue is to be able to verify, at each step, the correctness of groups of instructions, as well as your input and output sections. This requires a systematic and modular construction of your program, based on ‘building blocks’ formed by separate subroutines or modules, each one of them being verified separately.

We recommend you to proceed as follows:

- Start with a main program whose function is exclusively to control all the steps of your algorithm, as well as the input and output modules. Do not introduce algorithmic elements in the main program, concentrating them in separate subroutines.
- Define subroutines with a single objective, i.e. avoid different functionalities in the same subroutine. For instance, if you need to solve algebraic systems, consider a routine that fulfills this objective without adding other functionalities, such as preparing output plots for instance.
- Verify each of these subroutines as an isolated subprogram. For the example mentioned, create algebraic systems with known exact solutions and apply your subprogram to verify that it operates correctly. Once this is achieved, and if errors appear in your program, you will be assured that it does not come from the verified subroutine, but most probably from incorrect or invalid input data.
- During the debugging phase, introduce print instructions at all steps of your program, before and after the instructions, which you will remove when all the bugs are fixed. For instance, when you read in geometrical or flow data from a file, we recommend printing immediately these data, to make sure that they are correctly read, in the proper format and in the expected units. Similarly, when the main program calls a subroutine, put print instructions before and after the call, to verify that the input data as well as the output data are as expected.

The potential and Euler models and some of their essential properties are presented in Sections 11.1 and 11.2. The steady potential flow solutions are treated in Section 11.3. The flow case suggested for the development of your first CFD program is the potential flow around a cylinder, for which we know an exact solution in the limit of incompressible flows. This will allow you to verify the accuracy of your numerical solution, in function of discretization options and grid density. The extension to compressible potential flows will be included, and when keeping the Mach number low enough, you will also be able to verify your numerical solution by comparison with the exact incompressible solution.

Section 11.4 focuses on the application of the Finite Volume Method (FVM) to the system of Euler equations, on a cell-centered grid. We will select a central space

discretization, requiring the introduction of *artificial dissipation*, which we will couple to a Runge–Kutta time integration method. A most critical issue with the Euler equations is the definition and the number of physical boundary conditions we are allowed to impose. This will be related to the hyperbolic properties and the associated characteristic speeds of propagation.

Section 11.5 will guide you to the applications of your FVM code to three test cases. The first one is the flow around the cylinder, as a first verification case. The second case is the compressible internal flow between two solid walls, with a circular bump on the lower wall. Although this flow has no known exact solution, a numerical solution obtained on a fine grid will serve as a reference solution for verification. A third case, with an exact solution, is the supersonic flow over a wedge, generating an oblique shock and will put you in a first contact with some important issues related to the presence of shock discontinuities.

Figure 11.0.1 summarizes the guide through this chapter.

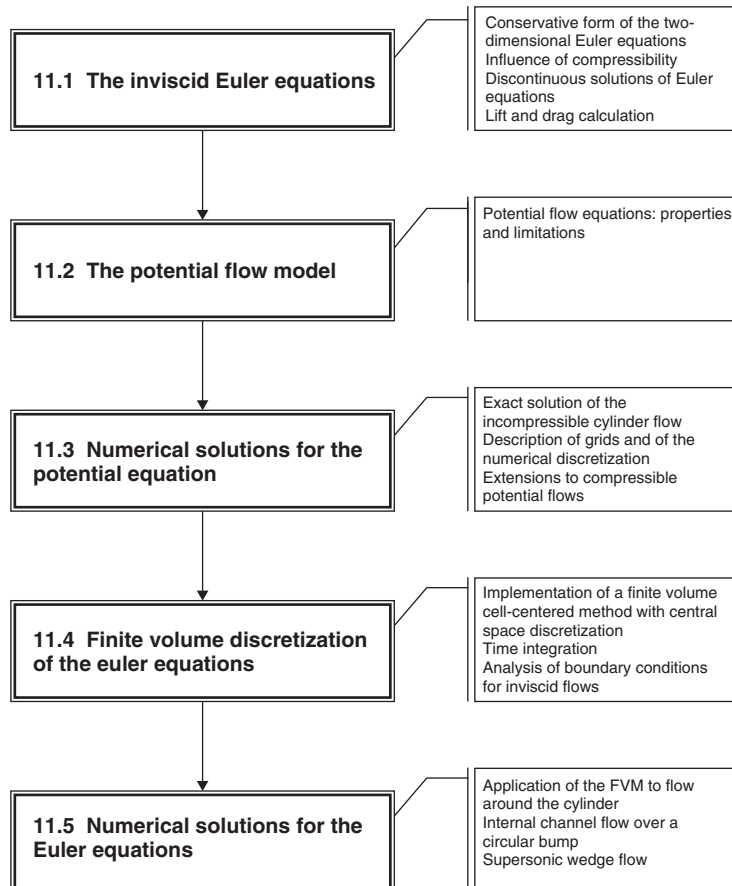


Figure 11.0.1 Content and guide to this chapter.

11.1 THE INVISCID EULER EQUATIONS

The system of Euler equations describes flows where the influence of viscous shear stresses and heat conduction effects can be neglected.

Referring to Section 2.7, the system of Euler equations, is written in a compact, conservative form as

$$\frac{\partial U}{\partial t} + \vec{\nabla} \cdot \vec{F} = 0 \quad (11.1.1)$$

This system of first order partial differential equations is hyperbolic in time and space, and in two dimensions the flux vector \vec{F} has the Cartesian components (f, g) given by

$$f = \begin{vmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ \rho uH \end{vmatrix} \quad g = \begin{vmatrix} \rho v \\ \rho vu \\ \rho v^2 + p \\ \rho vH \end{vmatrix} \quad U = \begin{vmatrix} \rho \\ \rho u \\ \rho v \\ \rho E \end{vmatrix} \quad (11.1.2)$$

Note that when combining the continuity equation (first equation) with the energy conservation equation (fourth equation), we obtain

$$\frac{\partial H}{\partial t} + u \frac{\partial H}{\partial x} + v \frac{\partial H}{\partial y} = \frac{1}{\rho} \frac{\partial p}{\partial t} \quad (11.1.3)$$

which reduces, for steady flows, to the constancy of total enthalpy H :

$$H = H_{\text{inlet}} \quad \text{along each streamline} \quad (11.1.4)$$

It is important to notice the properties of the entropy variations in an inviscid flow. From equation (2.7.3) and in absence of heat sources, the entropy equation for continuous flow variations reduces to

$$T \left(\frac{\partial s}{\partial t} + \vec{v} \cdot \vec{\nabla} s \right) = 0 \quad (11.1.5)$$

expressing that entropy is constant along a flow path. For steady flows, we have

$$s = s_{\text{inlet}} \quad \text{along each streamline} \quad (11.1.6)$$

The value of the entropy can however vary from one flow path to another. This is best seen from Crocco's form of the momentum equation (1.5.13), which reduces, for a stationary inviscid flow in absence of external forces, to

$$-(\vec{v} \times \vec{\zeta}) = T \vec{\nabla} s - \vec{\nabla} H \quad (11.1.7)$$

In an intrinsic coordinate system with unit vectors $(\vec{e}_l, \vec{e}_n, \vec{e}_b)$, where l is directed along the velocity and b is the binormal direction, this equation becomes, when projected in the normal direction n , for a uniform total enthalpy,

$$|\vec{v}| \zeta_b = T \frac{\partial s}{\partial n} \quad (11.1.8)$$

This relation shows that entropy variations in the direction normal to the local velocity direction are connected to vorticity. **Hence entropy variations will generate vorticity and inversely, vorticity will create entropy gradients.**

11.1.1 Steady Compressible Flows

The constancy of the total enthalpy, identical to the constancy of the total temperature T_0 , takes the following form, assuming perfect gas conditions, where A is a point on the inlet surface:

$$H = c_p T_0 = H_A = c_p T_{0A}$$

$$T_0 = T + \frac{\vec{v}^2}{2c_p} = T \left(1 + \frac{\gamma - 1}{2} M^2 \right) \quad (11.1.9)$$

The isentropic relations between pressure, temperature and density, are

$$\frac{\rho}{\rho_0} = \left[\frac{T}{T_0} \right]^{1/(\gamma-1)} = \left[\frac{p}{p_0} \right]^{1/\gamma} \quad (11.1.10)$$

Important property

Combining the isentropic condition $s = \text{const.}$, with equation (11.1.9), expressing the constancy of total temperature or total enthalpy, and taking into account equation (1.4.36), repeated here for convenience

$$s - s_A = -r \ln \frac{p_0/p_{0A}}{(H/H_A)^{\gamma/(\gamma-1)}} \quad (11.1.11)$$

we see immediately that the **total pressure** has to be constant

$$p_0 = p \left[1 + \frac{\gamma - 1}{2} M^2 \right]^{\gamma/(\gamma-1)} = p_{0A} \quad (11.1.12)$$

Hence, all stagnation conditions are constant along streamlines. Note that these constant values can change from one streamline to the other. However, if the incoming flow is uniform, leading to a potential flow, then there is only one constant value over the whole flow field.

11.1.2 The Influence of Compressibility

An important question in practical applications is related to the choice between a compressible flow model and a purely incompressible model, for low velocity flows of gases, such as air.

In other words, below which velocity, or Mach number levels, can we consider a flow of air as incompressible?

The answer to this question can be obtained from the above relations, for instance by evaluating the numerical influence of Mach number on total pressure, equation (11.1.12).

Table 11.1.1 *Mach number influence on compressibility effects.*

M	$\gamma M^4/8$	$M^2/4$
0.01	1.75×10^{-9}	0.000025
0.05	1.09375×10^{-6}	0.000625
0.1	1.75×10^{-5}	0.0025
0.2	0.000280	0.01
0.3	0.00141750	0.0225

It is known from basic fluid mechanics and Bernoulli equation in particular, that in an incompressible flow with constant density, the stagnation pressure is defined by

$$p_0 = p + \rho \frac{\vec{v}^2}{2} \quad (11.1.13)$$

Referring to the general definition (11.1.12), we can expand this equation in power series of Mach number M , in order to compare with equation (11.1.13).

We obtain

$$\frac{p_0}{p} = \left[1 + \frac{\gamma-1}{2} M^2 \right]^{\gamma/(\gamma-1)} \cong 1 + \frac{\gamma}{2} M^2 + \frac{\gamma}{8} M^4 + \frac{2-\gamma}{48} \gamma M^6 + O(M^8) \quad (11.1.14)$$

With the definition of Mach number as

$$M^2 = \frac{\vec{v}^2}{\gamma r T} \quad (11.1.15)$$

where the speed of sound is defined as $c = \sqrt{\gamma r T}$, and with the perfect gas law $p = \rho r T$, we obtain

$$p_0 \cong p + \rho \frac{\vec{v}^2}{2} + \frac{\gamma}{8} p M^4 + p \frac{2-\gamma}{48} \gamma M^6 + O(M^8) \quad (11.1.16)$$

The dominating correction factor for compressibility influence is the third term of this equation $\gamma p M^4/8$, but we prefer to evaluate the ratio of this correction to the second term, ratio equal to $M^2/4$, as seen from equation (11.1.14). For the standard value of $\gamma = 1.4$, valid in particular for air, these factors take the following values for various Mach numbers (Table 11.1.1).

The second column is the ratio between the third term and the static pressure p (first term), while the third column is the ratio between the third term and the **dynamic pressure** (second term). At $M = 0.1$, the compressibility effect represents a correction to the dynamic pressure of 0.25% and reaches 1% at $M = 0.2$.

For practical reasons we may consider a 1% error as acceptable and conclude that for $M \leq 0.2$, the compressibility effects can be neglected and the gas flow can be considered as incompressible. For a flow of air at atmospheric conditions ($T = 288$ K), the speed of sound is $c \approx 340$ m/s and the limit $M = 0.2$ corresponds to a velocity of 68 m/s or 250 km/h. This can be considered as a serious storm if we think about it as an atmospheric wind speed.

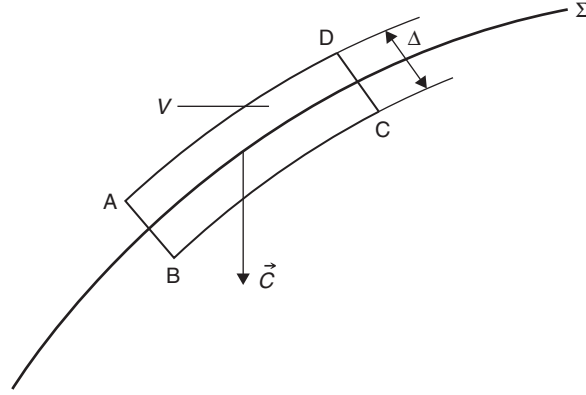


Figure 11.1.1 Control volume around a moving discontinuity surface Σ (Δ is an infinitesimal distance normal to Σ).

11.1.3 The Properties of Discontinuous Solutions

As is known, the set of Euler equations allows also discontinuous solutions in certain cases, namely **vortex sheets**, **contact discontinuities** or **shock waves** occurring in supersonic flows. The properties of these discontinuous solutions can only be obtained from the integral form of the conservation equations, since the gradients of the fluxes are not defined at discontinuity surfaces.

For a discontinuity surface Σ , moving with velocity \vec{C} , the integral conservation laws are applied to the infinitesimal volume V of Figure 11.1.1.

Referring to equation (1.1.2) in absence of source terms, the integral form of the Euler equations, takes the following form:

$$\frac{\partial}{\partial t} \int_V U \, d\Omega + \oint_S \vec{F} \cdot d\vec{S} = 0 \quad (11.1.17)$$

The time derivative of the volume integral has to take into account the motion of the surface Σ and hence of the control volume V , through

$$\begin{aligned} \frac{\partial}{\partial t} \int_V U \, d\Omega &= \int_V \frac{\partial U}{\partial t} d\Omega + \int_V U \frac{\partial}{\partial t} (d\Omega) \\ &= \int_V \frac{\partial U}{\partial t} d\Omega - \oint_S U \vec{C} \cdot d\vec{S} \end{aligned} \quad (11.1.18)$$

expressing the conservation of the volume V in the translation with velocity \vec{C} .

The flux term in equation (11.1.17) can be rewritten for vanishing volumes $V(\Delta \rightarrow 0)$ as

$$\oint_S \vec{F} \cdot d\vec{S} = \int_{\Sigma} (\vec{F}_2 - \vec{F}_1) \cdot d\vec{\Sigma} \equiv \int_{\Sigma} [\vec{F} \cdot \vec{e}_n] d\Sigma \quad (11.1.19)$$

where $d\vec{\Sigma}$ is normal to the discontinuity surface Σ and where the notation

$$[A] \equiv A_2 - A_1 \quad (11.1.20)$$

denotes the jump in the variable A when crossing the discontinuity.

Combining (11.1.18) and (11.1.19) we obtain, for vanishing volumes V ,

$$\int_{\Sigma} ([\vec{F}] - \vec{C}[U]) \cdot d\vec{\Sigma} = 0 \quad (11.1.21)$$

leading to the local form of the conservation laws over a discontinuity, called the **Rankine–Hugoniot relations**

$$[\vec{F}] \cdot \vec{e}_n - \vec{C}[U] \cdot \vec{e}_n = 0 \quad (11.1.22)$$

If $\Sigma(\vec{x}, t) = 0$ is the discontinuity surface, then we have

$$\frac{d\Sigma}{dt} \equiv \frac{\partial \Sigma}{\partial t} + (\vec{C} \cdot \vec{\nabla})\Sigma = 0 \quad (11.1.23)$$

With the unit vector along the normal \vec{e}_n defined by

$$\vec{e}_n = \frac{\vec{\nabla}\Sigma}{|\vec{\nabla}\Sigma|} \quad (11.1.24)$$

equation (11.1.22) takes the form

$$[\vec{F}] \cdot \vec{\nabla}\Sigma + [U] \frac{\partial \Sigma}{\partial t} = 0 \quad (11.1.25)$$

Various forms of discontinuities are physically possible:

- **Shocks** where all flow variables undergo a discontinuous variation.
- **Contact discontinuities and vortex sheets**, also called **slip lines**, across which no mass transfer takes place but where density, as well as the tangential velocity, maybe discontinuous, although pressure and normal velocity remain continuous.

The properties of these discontinuous solutions can best be seen from a reference system moving with the discontinuity. In this system the discontinuity surface is stationary, $C = 0$, and the **Rankine–Hugoniot relations** for the Euler equations become

$$[\rho v_n] = 0 \quad (11.1.26a)$$

$$[\rho v_n \vec{v}] + [p] \vec{e}_n = 0 \quad (11.1.26b)$$

$$\rho v_n [H] = 0 \quad (11.1.26c)$$

where v_n is the normal component of the velocity vector $v_n = \vec{v} \cdot \vec{e}_n$.

The third equation shows that total enthalpy always remains constant through the discontinuity.

This system admits solutions with the following properties.

11.1.3.1 **Contact discontinuities**

They are defined by the condition of no mass flow through the discontinuity:

$$v_{n1} = v_{n2} = 0 \quad (11.1.27)$$

and, following equation (11.1.26), by continuity of pressure

$$[p] = 0 \quad (11.1.28)$$

allowing non-zero values for the jump in specific mass, as seen from equation (11.1.26a)

$$[\rho] \neq 0 \quad (11.1.29)$$

The tangential velocity variation over the discontinuity could be continuous or not, as seen from the tangential projection of equation (11.1.26b).

When the tangential velocity is continuous, we have a **contact discontinuity**

$$[v_t] = 0 \quad (11.1.30)$$

11.1.3.2 **Vortex sheets or slip lines**

They are also defined by the conditions of no mass flow through the discontinuity, continuous pressure and discontinuous density, as for the contact discontinuity, but with a jump in tangential velocity:

$$\begin{aligned} v_{n1} &= v_{n2} = 0 \\ [p] &= 0 \\ [\rho] &\neq 0 \\ [v_t] &\neq 0 \end{aligned} \quad (11.1.31)$$

11.1.3.3 **Shock surfaces**

Shocks are solutions of the Rankine–Hugoniot relations with *non-zero mass flow* through the discontinuity, which appear with **supersonic flows**. Consequently, pressure and normal velocity undergo discontinuous variations, while the tangential velocity remains continuous. Hence, shocks satisfy the following properties:

$$\begin{aligned} [v_n] &\neq 0 \\ [p] &\neq 0 \\ [\rho] &\neq 0 \\ [v_t] &= 0 \end{aligned} \quad (11.1.32)$$

Note that since the stagnation pressure p_0 is *not constant* across the shock, the inviscid shock relations imply a **discontinuous entropy variation** through the shock. This variation has to be positive, corresponding to compression shocks and excluding hereby expansion shocks, for physical reasons connected to the second principle of thermodynamics, (Shapiro, (1953); Zucrow and Hoffmann, (1976)).

It has to be added that expansion shocks, whereby the entropy jump is negative, are also valid solutions of the inviscid equations. Hence, there is no mechanism allowing to distinguish between discontinuities with entropy increase (positive entropy jump) or entropy decrease (negative entropy variation). An additional condition, called the **entropy condition** has therefore to be added to the inviscid equation in order to exclude these non-physical solutions, (Lax, 1973). This is necessary for all inviscid flow models and a more detailed discussion of the entropy condition is presented in Volume II.

The mathematical formulation of the second principle of thermodynamics can be expressed, for an adiabatic flow without heat conduction nor heat sources $q_H = 0$, following equation (1.4.18)

$$\rho T \left(\frac{\partial s}{\partial t} + \vec{v} \cdot \vec{\nabla} s \right) = \varepsilon_v \quad (11.1.33)$$

Since ε_v is the viscous dissipation and always positive, this equation states that any solution of the Euler equations which has a physical sense as a limit, for vanishing viscosity, of real fluid flow phenomena, has to satisfy the following entropy inequality:

$$\left(\frac{\partial s}{\partial t} + \vec{v} \cdot \vec{\nabla} s \right) \geq 0 \quad (11.1.34)$$

In addition, a non-uniform discontinuity such as a shock with varying intensity will generate a non-uniform entropy field in the direction normal to the velocity. Equation (11.1.8) then shows that as a consequence, vorticity will be generated downstream of the shock. **Hence, even for irrotational flow conditions upstream of the shock, a rotational flow will be created by a non-uniform shock intensity.**

11.1.4 Lift and Drag on Solid Bodies

The lift and drag resulting forces exerted by the flow on a solid body can be obtained by an extension of the momentum conservation law in integral form, equation (1.3.9). See also the ‘Advanced’ section A1.6.1 in Chapter 1.

If the control volume Ω contains a solid body, then an additional force ($-\vec{R}$) has to be added to the right-hand side of equation (1.3.9), where \vec{R} is the total force exerted **by** the fluid on the body. The total force \vec{R} is composed of the **lift force** \vec{L} , defined as the component normal to the incoming velocity, and the **drag force** \vec{D} defined as the component parallel and opposed to the incoming velocity direction.

We obtain, in absence of external forces, for stationary flows:

$$\oint_S \rho \vec{v} (\vec{v} \cdot d\vec{S}) = - \oint_S p d\vec{S} + \oint_S \vec{\tau} \cdot d\vec{S} - \vec{R} \quad (11.1.35)$$

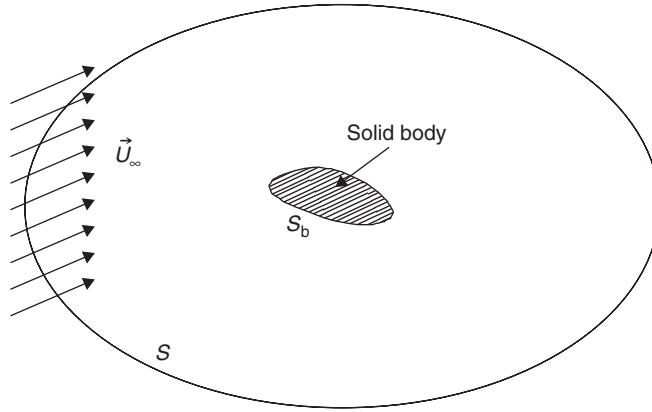


Figure 11.1.2 Far-field control surface for lift and drag determination on enclosed solid body.

Based on this equation, two methods can be applied to calculate lift and drag forces:

- (i) For a surface S located in the far field (Figure 11.1.2), where the viscous shear stresses can be considered as negligible, the sum of the stationary lift and drag forces are given by the following relation, for stationary flows:

$$-\vec{R} = -(\vec{L} + \vec{D}) = \oint_S \rho \vec{v} (\vec{v} \cdot d\vec{S}) + \oint_S p d\vec{S} \quad (11.1.36)$$

- (ii) On the other hand, if the control surface S is taken **on the solid body surface** S_b , where the velocity field is either zero due to the non-slip condition of viscous flows or having zero normal velocity with inviscid flows, then the lift and drag forces are also defined by the following relation, since the left-hand side term of equation (11.1.35) is zero

$$\vec{R} = \vec{L} + \vec{D} = - \oint_{S_b} p d\vec{S} + \oint_{S_b} \vec{\tau} \cdot d\vec{S} \quad (11.1.37)$$

This is an important relation, which is currently applied to determine lift and drag forces from computed flow fields.

For inviscid flows, there are no shear stresses and the lift force over the body is the resultant of the pressure forces:

$$\vec{L} = - \oint_{S_b} p d\vec{S} \quad (11.1.38)$$

Note that, for aeronautical applications, the lift force is often considered as defined by the **vertical component of the pressure forces** and its horizontal component is called the **pressure drag**, as it will act alongside the drag force, although it results from the inviscid pressure forces.

11.2 THE POTENTIAL FLOW MODEL

The most impressive simplification of the mathematical description of a flow system is obtained with the approximation of a non-viscous, **irrotational** flow.

The condition of zero vorticity

$$\vec{\zeta} = \vec{\nabla} \times \vec{v} = 0 \quad (11.2.1)$$

will be automatically satisfied if the three-dimensional velocity field can be described by a single scalar potential function ϕ , defined by

$$\vec{v} = \vec{\nabla}\phi \quad (11.2.2)$$

since the rotation of a gradient is identical zero, for any value of the function ϕ .

This reduces the knowledge of the three velocity components to the determination of a single scalar function ϕ .

It is known from the theory of inviscid flows, that the vorticity remains constant in any streamtube. Hence, in a flow where the rotation free condition (11.2.1) is satisfied at the inlet boundary, it will remain so everywhere in the flow domain. In particular, since a uniform flow is rotation free, any subsonic inviscid flow with uniform inlet conditions will remain rotation free everywhere.

This has far-reaching consequences, namely that all the flow variables of a 3D inviscid potential flow are completely defined by the single potential function. Hence we are left with one unknown instead of five, which represents a considerable simplification.

The applications handled in this chapter will be restricted to steady potential flows, which offers an additional simplification.

It is seen from equation (11.1.8) that a potential flow is always isentropic, and with the isentropic relation (11.1.10), the density itself is completely defined by the potential function, as shown by equation (2.8.6), assuming perfect gas conditions:

$$\frac{\rho}{\rho_0} = \left[1 - \frac{\vec{v}^2}{2H} \right]^{1/(\gamma-1)} = \left[1 - \frac{(\vec{\nabla}\phi)^2}{2H} \right]^{1/(\gamma-1)} \quad (11.2.3)$$

where ρ_0 is the stagnation density, constant throughout the whole flow field and γ is the specific heat ratio, equal to $\gamma = 1.4$ for air.

Since all stagnation conditions are constant throughout the whole flow field all thermodynamic properties are known as soon as we know the potential function. This demonstrates the simplification introduced by potential flows, where the knowledge of the single scalar potential function determines all the five flow variables in 3D.

From equation (2.8.5) we obtain the steady potential equation:

$$\vec{\nabla} \cdot (\rho \vec{\nabla}\phi) = 0 \quad (11.2.4)$$

Both for steady and unsteady flows, the inviscid boundary condition along a solid boundary is zero normal relative velocity between flow and solid boundary

$$v_n = \frac{\partial \phi}{\partial n} = \vec{u}_w \cdot \vec{e}_n = 0 \quad (11.2.5)$$

where \vec{u}_w is the velocity of the solid boundary with respect to the considered system of reference and n is the direction normal to the solid wall.

11.2.1 The Limitations of the Potential Flow Model for Transonic Flows

If we consider the steady state potential model for continuous flows, the constancy of entropy and total enthalpy, coupled to irrotationality, form a set of conditions fully consistent with the system of Euler equations. Hence, the model defined by

$$s = s_0 = \text{const.}$$

$$H = H_0 = \text{const.} \quad (11.2.6)$$

and $\vec{v} = \vec{\nabla} \phi$ or $\vec{\nabla} \times \vec{v} = 0$, where ϕ is solution of the mass conservation equation, ensures that the momentum and energy conservation laws are also satisfied. **Therefore, it can be considered that an inviscid continuous flow, with initial conditions satisfying the condition (11.2.1), will be exactly described by the potential flow model.**

However, in presence of discontinuities such as shock waves, this will not be the case anymore since the Rankine–Hugoniot relations lead to an entropy increase through a shock. If the shock intensity is uniform, then the entropy will remain uniform downstream of the shock, but at another value than the initial constant value. In this case, according to equation (11.1.8), the flow remains irrotational. However, if the shock intensity is not constant, which is most likely to occur in practice, for instance for curved shocks, then equation (11.1.8) shows that the flow is not irrotational anymore and hence the mere existence of a potential downstream of the discontinuity cannot be justified rigorously. Therefore, the *potential flow model in presence of shock discontinuities cannot be made fully compatible with the system of Euler equations*, since the potential model implies constant entropy and has therefore no mechanisms to generate entropy variations over discontinuities.

11.2.2 Incompressible Potential Flows

When the density is constant the potential equation (11.2.4) reduces to the simplest Laplace equation:

$$\Delta \phi = 0 \quad (11.2.7)$$

with the boundary condition of zero normal velocity for a fixed cylinder, expressed by the Neumann boundary condition:

$$\frac{\partial \phi}{\partial n} = 0 \quad \text{on the solid walls} \quad (11.2.8)$$

In the far field, where the inflow velocity \vec{U} is constant, we have from the definition of the potential function:

$$\phi - \phi_0 = \vec{U} \cdot \vec{x} \quad (11.2.9)$$

11.3 NUMERICAL SOLUTIONS FOR THE POTENTIAL EQUATION

We will deal here with a single case, defined by the flow around a cylinder, for which an exact solution is available in the incompressible limit. It will offer a first opportunity to illustrate several aspects of a CFD code related to grid properties and to the accuracy dependence with grid density.

Although applied to the simplest of the models, the conclusions will nevertheless be of more general validity.

11.3.1 Incompressible Flow Around a Circular Cylinder

We start with one of the simplest potential flows, the 2D *incompressible* flow around a circular cylinder of radius a , with uniform inlet velocity U , as illustrated in Figure 11.3.1.

An exact solution is known from classic fluid dynamics, defined by the complex potential function $\zeta = \phi + I\psi$ ($I = \sqrt{-1}$), where ψ is the streamfunction

$$\zeta(z) = \phi(x, y) + I\psi(x, y) \quad z = x + Iy \quad (11.3.1)$$

The complex velocity is defined by

$$u - Iv = \frac{d\zeta}{dz} \quad (11.3.2)$$

leading to the relations

$$\begin{aligned} u &= \frac{\partial \phi}{\partial x} = \frac{\partial \psi}{\partial y} \\ v &= \frac{\partial \phi}{\partial y} = -\frac{\partial \psi}{\partial x} \end{aligned} \quad (11.3.3)$$

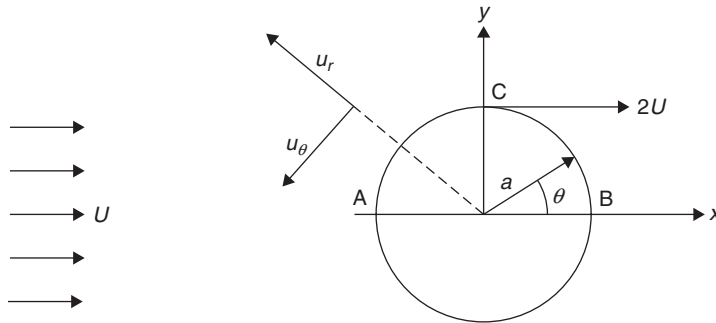


Figure 11.3.1 Two-dimensional, incompressible potential flow around a circular cylinder for a uniform incident velocity field.

The exact incompressible potential solution for a cylinder of radius a , is defined by

$$\zeta(z) = \phi(x, y) + I\psi(x, y) = U \left(z + \frac{a^2}{z} \right) \quad (11.3.4)$$

leading to

$$\phi(x, y) = Ux \frac{x^2 + y^2 + a^2}{x^2 + y^2} \quad \psi(x, y) = Uy \frac{x^2 + y^2 - a^2}{x^2 + y^2} \quad (11.3.5)$$

and

$$u - Iv = U \left(1 - \frac{a^2}{z^2} \right) \quad (11.3.6)$$

or

$$u = U \left[1 - \frac{a^2(x^2 - y^2)}{(x^2 + y^2)^2} \right] \quad v = -2Ua^2 \frac{xy}{(x^2 + y^2)^2} \quad (11.3.7)$$

On the cylinder surface, that is for $x^2 + y^2 = a^2$, we have

$$\begin{aligned} \phi &= 2Ux & \psi &= 0 \\ u &= 2U \left(1 - \frac{x^2}{a^2} \right) & v &= -2U \frac{xy}{a^2} \end{aligned} \quad (11.3.8)$$

Another representation is obtained when replacing the complex position variable z by its polar, instead of its Cartesian, form

$$z = x + Iy = re^{I\theta} = r(\cos \theta + I \sin \theta) \quad (11.3.9)$$

The exact solution to this potential flow becomes, in cylindrical coordinates

$$\begin{aligned} \zeta(z) &= \phi(r, \theta) + I\psi(r, \theta) = U \left(re^{I\theta} + \frac{a^2}{r} e^{-I\theta} \right) \\ \phi(r, \theta) &= U \left(r + \frac{a^2}{r} \right) \cos \theta & \psi(r, \theta) &= U \left(r - \frac{a^2}{r} \right) \sin \theta \end{aligned} \quad (11.3.10)$$

The polar velocity components are defined by

$$\begin{aligned} u_r &= \frac{\partial \phi}{\partial r} = \frac{1}{r} \frac{\partial \psi}{\partial \theta} = U \left(1 - \frac{a^2}{r^2} \right) \cos \theta \\ u_\theta &= \frac{1}{r} \frac{\partial \phi}{\partial \theta} = -\frac{\partial \psi}{\partial r} = -U \left(1 + \frac{a^2}{r^2} \right) \sin \theta \end{aligned} \quad (11.3.11)$$

In particular on the cylinder surface, defined by

$$z = ae^{I\theta} = a(\cos \theta + I \sin \theta) \quad (11.3.12)$$

the complex potential function on the surface reduces to

$$\phi(r, \theta) = 2Ua \cos \theta \quad \psi(r, \theta) = 0 \quad (11.3.13)$$

and the velocity components take the following values:

$$u_r = 0 \quad u_\theta = -2U \sin \theta \quad (11.3.14)$$

This confirms that the cylinder surface is a streamline where the velocity is along the tangential direction. The negative sign results from the definition of the tangential velocity component as being positive in the anti-clockwise direction, while the incoming velocity is in the positive x -direction, as shown on Figure 11.3.1. Note that the velocity is zero at the stagnation points A, B, but reaches a value equal to $2U$, that is twice the incoming velocity at the top of the cylinder in point C, at $\theta = 90^\circ$.

The pressure field is obtained from the constancy of the stagnation pressure (11.1.13), written here for incompressible flows

$$p_0 = p + \rho \frac{\vec{v}^2}{2} = p_\infty + \rho \frac{U^2}{2} \quad (11.3.15)$$

and is best expressed by a non-dimensional pressure coefficient C_p , which is *independent of the inlet velocity*:

$$C_p = \frac{p - p_\infty}{\rho \frac{U^2}{2}} = 1 - \frac{\vec{v}^2}{U^2} = 1 - \frac{|\vec{\nabla}\phi|^2}{U^2} \quad (11.3.16)$$

The pressure coefficient on the surface (indicated by a subscript S) is generally plotted in function of the solid wall arc length, and becomes here, with (11.3.14)

$$C_p|_S = \frac{p_S - p_\infty}{\rho \frac{U^2}{2}} = 1 - \frac{\vec{v}_S^2}{U^2} = 1 - 4 \sin^2 \theta \quad (11.3.17)$$

Figure 11.3.2 displays the streamlines and potential lines, as well as the surface pressure coefficient.

We now proceed by following the steps of the previous chapters, as developed in Parts I–IV:

- (a) Select the mathematical model.
- (b) Define the grid.
- (c) Define the numerical scheme.
- (d) Establish the stability and accuracy properties of the scheme, based on the material of the previous chapters. If necessary, perform a new analysis.
- (e) Solve the algebraic system.
- (f) Analyze the results and evaluate the grid dependence and overall accuracy.

These steps should be translated into a flowchart, establishing the structure of your main program, as seen on Figure 11.3.3. Each box should represent a call to one or

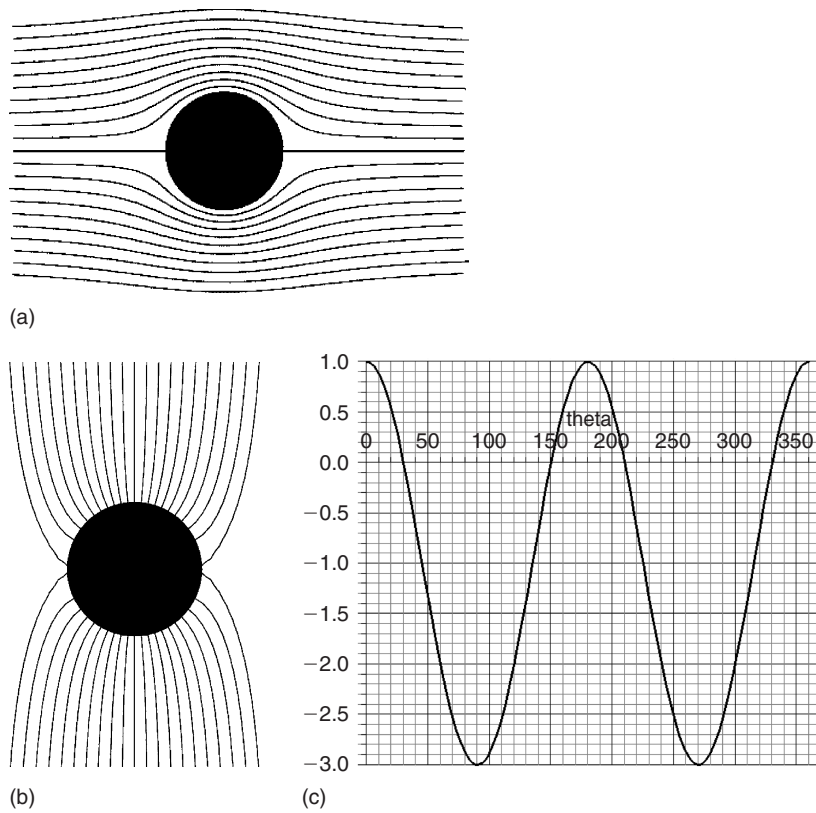


Figure 11.3.2 Potential flow on a circular cylinder: (a) streamlines, (b) potential lines and (c) surface pressure coefficient C_{ps} in function of angular position θ .

more separate subroutines, each one having a single objective. It is recommended during the debugging phase, to add print instructions before and after each call to a subroutine, to verify that the intended operations are correctly executed.

Let us now apply this to the mathematical model of the incompressible potential flow equation (11.2.17) with the boundary conditions (11.2.18) and (11.2.9).

11.3.1.1 Define the grid

We will select a straightforward analytical grid in polar coordinates, formed by circles and radial lines, allowing you to have full control of mesh density and mesh spacing.

You are now faced with your first decision, namely you have to fix the outer boundary of your computational domain for this external flow problem. This is an important decision, since we apply free undisturbed flow conditions on this boundary and therefore it should be located far enough from the solid body in order to ensure that its influence is negligible. Keep in mind that in subsonic flows, all the points in the flow domain influence each other, as is typical for elliptic equations. Hence, in theory

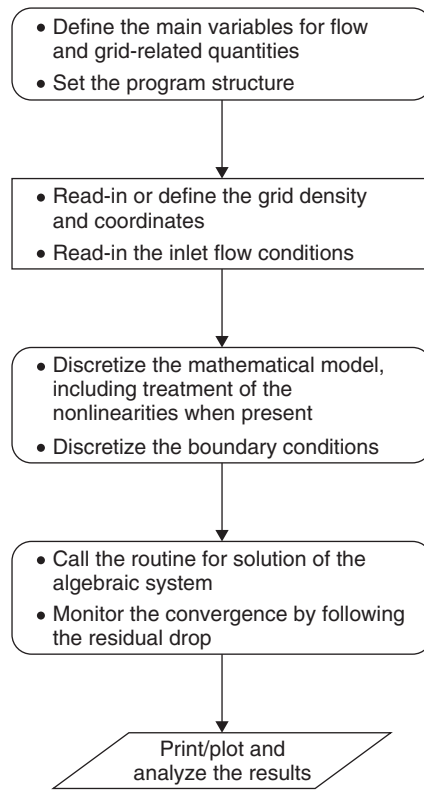


Figure 11.3.3 Structure of main program.

we will always have some disturbance from uniform flow conditions on the outer boundary, but if far enough it could be neglected.

In practical terms, a distance of the order of 40–50 times the radius should be recommended for the outer boundary, at least for non-lifting bodies.

For lift generating airfoils, the far field is influenced by a free vortex singularity defined by the circulation around the airfoil, which tends to zero like the inverse of the distance. Either this correction is introduced in the far field, or we have to increase the distance of the outer boundary to values closer to 100 chords.

This initial decision can be a first source of errors, and in case of doubt, you should apply your code with several positions of the downstream boundary, and verify its influence on the solution.

How to choose the grid spacing?

Option 1: Select circles equally spaced in radius value (index i) and radial lines equally spaced in angular position (index j), as shown on Figure 11.3.4. If the outer boundary is fixed at 40 times the cylinder radius a , then if we consider a mesh with N_i points in the radial direction, the radial spacing between the circles should be equal to $40a/N_i$.

Option 2: The first option might seem straightforward, but at second thought it is not such a good idea. Indeed, an important guideline to a good grid is to concentrate more

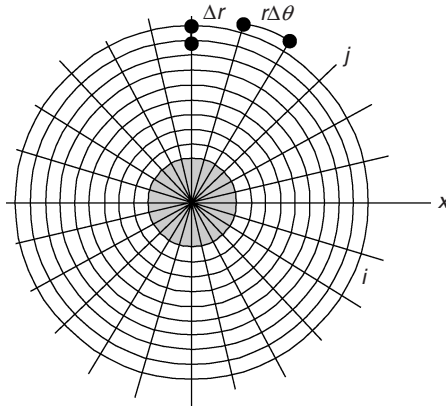


Figure 11.3.4 Equidistant mesh for circular cylinder.

grid points in regions of stronger flow variations that is in regions of strong gradients of the flow variables. If we denote du/dn a gradient of a variable u in the n -direction, a good criterion is to aim at a grid where the numerical variation $\Delta u \simeq (\partial u / \partial n) \Delta n$ over the cells remains of the same order over the flow domain. This implies that the grid spacing Δn in the n -direction should be inversely proportional to the local gradient intensity. ***This guideline leads to smaller grid spacing near the solid boundaries, where the flow gradients will always be significant and larger grid spacing when approaching the far field, where the flow approaches uniform conditions.*** Hence, we should select a grid where the radial spacing increases from the solid body surface to the far field with a clustering factor R , as defined for instance by equation (4.3.1), written here as

$$r_{i+1} = aR^i \quad \text{or} \quad \Delta r_i \triangleq r_{i+1} - r_i = r_i(R - 1) \quad (11.3.18)$$

The factor R is defined by the position of the outer boundary and the number of mesh points N_i in the radial direction:

$$r_{N_i} = aR^{N_i-1} \quad (11.3.19)$$

For instance, selecting $N_i = 33$ points in the radial direction and an outer boundary at $40a$, we obtain

$$40a = aR^{32} \quad \text{or} \quad R = 40^{1/32} = 1.12218478 \quad (11.3.20)$$

The resulting radial coordinates are listed in Table 11.3.1, as the second column, applying equation (11.3.18), for $a = 1$.

Option 3: This grid can still be improved in terms of quality, if we look at the curvilinear polar grid (r, θ) as a transformation from the grid in the physical space. For the considered polar grid, we have

$$x = r \cos \theta \quad y = r \sin \theta \quad (11.3.21)$$

The arc length in the radial direction is Δr and the arc length of a cell side along the circular mesh lines is $r\Delta\theta$. Since the Cartesian grid is considered as the ideal grid, having equal spacing in both directions, we can improve the generated grid by requiring that the polar grid also satisfies the condition of equal arc lengths in the radial and circular directions: $\Delta r_i = r_i \Delta\theta$.

This condition can be satisfied on a grid defined by equation (11.3.18), leading to $(R - 1) = \Delta\theta$, and on a mesh with 128 points in the circular direction, we would have $R = 1 + 2\pi/128 = 1.04908738522$. This value does not allow controlling the

Table 11.3.1 Radial coordinates for grid options around circular cylinder of radius 1, based on equations (11.3.18) and (11.3.24).

Radial index i	Radius number $i + 1 = a \cdot R^i$ equation (11.3.18)	Radius number $i + 1$ equation (11.3.24)
1	1	1
2	1.122185	0.3454
3	1.259299	0.402619
4	1.413166	0.469318
5	1.585833	0.547065
6	1.779598	0.637692
7	1.997038	0.743333
8	2.241045	0.866474
9	2.514867	1.010015
10	2.822145	1.177334
11	3.166968	1.372373
12	3.553924	1.599721
13	3.988159	1.864732
14	4.475452	2.173645
15	5.022284	2.533732
16	5.63593	2.953472
17	6.324555	3.442747
18	7.09732	4.013075
19	7.964504	4.677884
20	8.937645	5.452826
21	10.02969	6.356146
22	11.25516	7.409109
23	12.63037	8.636508
24	14.17361	10.06724
25	15.90541	11.73499
26	17.84881	13.67901
27	20.02967	15.94509
28	22.47699	18.58656
29	25.22333	21.66563
30	28.30524	25.25477
31	31.76371	29.4385
32	35.64475	34.3153
33	40	40

outer boundary position with a user defined number of mesh points in the radial direction. To reach an outer boundary of $40a$, this value of R would require 77 radial mesh points. Inversely, if we keep the value of R defined by condition (11.3.20), we would be constrained in the number of points N_j in the circular direction by $(R - 1) = \Delta\theta = 2\pi/N_j$, leading to $N_j \sim 51$.

To keep full control of the number of mesh points in both directions, we should introduce an additional mesh scaling parameter. The straightforward attempt to rescale the clustered grid defined by equation (11.3.18), with a factor k .

$$r_{i+1} = kaR^i \quad \text{or} \quad \Delta r_i \triangleq r_{i+1} - r_i = kr_i(R - 1) \quad (11.3.22)$$

leads to the values $R = 1.165660769$ and $k = 0.296312673$. Because of the low value of k , the first few circular mesh lines are at a radius below the cylinder radius. Hence, this option is not acceptable.

Instead, we could require a mesh clustering, defined by

$$r_{i+1} = r_i + kaR^i \quad (11.3.23)$$

leading to

$$r_{i+1} = a \left(1 + k \sum_{m=0}^i R^m \right) \quad (11.3.24)$$

The condition of equal arc lengths of the curvilinear cells, $\Delta r_i = r_i \Delta\theta$ implies

$$\Delta r_i = kaR^i = r_i \Delta\theta = r_i \frac{2\pi}{N_j} \quad (11.3.25)$$

which clearly cannot be satisfied for all cells, as it would require a relation such as (11.3.22), which is different from the choice (11.3.24). However, you could satisfy this condition in an approximate way, by applying it at one point, for instance at the center of the computational domain, for a certain value of the mesh index i . Another simple option is to approach this condition at the level of the cylinder, for $i = 1$, with the choice $k = 2\pi/N_j$, neglecting the effect of the factor R . For $N_j = 128$ with $k = 2\pi/128$, applying (11.3.24) at a distance of $40a$ with 33 mesh points in the radial direction, using symbolic algebra software tools, such as MAPLE or Mathematica, you can obtain the value of the clustering factor R , as $R = 1.1580372$. The resulting radial coordinates are given in the third column of Table 11.3.1 for $a = 1$.

The main message at this initial stage of the mesh generation of your first CFD code is that each step requires sound judgment and a readiness to make choices and approximations.

It is up to you to choose, either to satisfy the equality of arc lengths everywhere, giving away the full control of the grid density and number of mesh points, or to fully control the number of mesh points in both directions, giving away the arc length uniformity.

And of course many other options for generation of grids around the circular cylinder are possible, as presented in Chapter 6.

We consider in the following table that you have selected one of the meshes just described, summarized in Table 11.3.1.

11.3.1.2 Define the numerical scheme

Here again, you are faced with many choices for the discretization of the Laplace equation (11.2.7) and its associated boundary conditions. We can write the potential equation in Cartesian coordinates or in cylindrical coordinates and apply a finite difference method (FDM). Alternatively, we can write the equation in integral form and apply a finite volume method (FVM), after having made a selection between the large numbers of possible choices for the control volumes. Finally, you could also apply a finite element method.

We will focus here on the FDM, while the FVM will be applied for the Euler equations.

Finite difference method in cylindrical coordinates

The Laplace equation for the potential function (11.2.7) in cylindrical coordinates is written as

$$\frac{\partial}{\partial r} \left(r \frac{\partial \phi}{\partial r} \right) + \frac{\partial}{\partial \theta} \left(\frac{1}{r} \frac{\partial \phi}{\partial \theta} \right) = 0 \quad (11.3.26)$$

and we wish to discretize this equation directly in the (r, θ) space, based on the mesh formed by circles and radial lines.

We refer you to Chapter 4 and equation (4.2.17) for a second order central discretization of this equation. Applied to the configuration of Figure 11.3.5, we obtain the scheme

$$\begin{aligned} & \frac{1}{(r_{i+1/2,j} - r_{i-1/2,j})} \left(r_{i+1/2,j} \frac{\phi_{i+1,j} - \phi_{i,j}}{\Delta r_i} - r_{i-1/2,j} \frac{\phi_{i,j} - \phi_{i-1,j}}{\Delta r_{i-1}} \right) \\ & + \frac{1}{\Delta \theta} \left(\frac{1}{r_{i,j+1/2}} \frac{\phi_{i,j+1} - \phi_{i,j}}{\Delta \theta} - \frac{1}{r_{i,j-1/2}} \frac{\phi_{i,j} - \phi_{i,j-1}}{\Delta \theta} \right) = 0 \end{aligned} \quad (11.3.27)$$

The mid-point radii are defined as follows:

$$r_{i\pm 1/2,j} = \frac{1}{2}(r_{i\pm 1,j} + r_{i,j}) \quad r_{i,j\pm 1/2} = r_{i,j} \quad (11.3.28)$$

where $r_{i,j}$ is independent of j , as the mesh line i is of constant radius and the sum

$$r_{i+1/2,j} - r_{i-1/2,j} = \frac{1}{2}(r_{i+1,j} - r_{i-1,j}) \quad (11.3.29)$$

Also

$$\Delta r_i = r_{i+1} - r_i \quad \Delta r_{i-1} = r_i - r_{i-1} \quad (11.3.30)$$

All these quantities are defined by the selected grid point distribution.

Refer here to Section 4.3.1.1 in Chapter 4 for an evaluation of the truncation errors associated to these conservative finite difference formulas, in particular

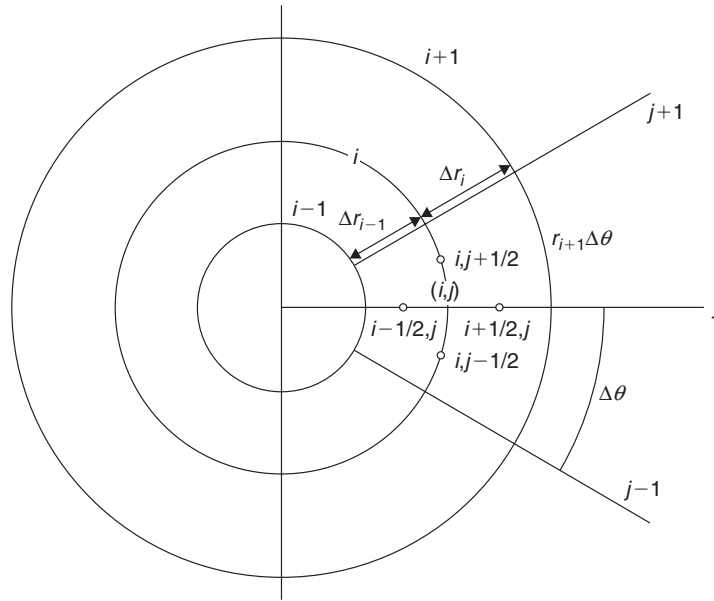


Figure 11.3.5 Flow around a cylinder: mesh formed by circular and radial lines. The i -index refers to constant radii mesh lines and the j -index to constant angular positions.

formula (4.3.9). As point (i, j) is not at the center between the points $(i - 1/2, j)$ and $(i + 1/2, j)$, the formulas are first order accurate in general, but in the present case, with the smooth grid variations defined by (11.3.18) or (11.3.24) the factor $(\Delta r_{i+1} - \Delta r_i) \sim O(\Delta r^2)$ and the scheme remains of second order accuracy.

Boundary conditions

Equation (11.3.27) can be applied from the value $i = 2$ on, while for $i = 1$, which is the surface of the cylinder, we have to apply the Neumann boundary condition (11.2.8). This condition can be discretized by a forward difference along the normal direction to connect the points at $i = 2$ to the points at $i = 1$. With the selected mesh this is straightforward since the radial mesh lines are the normals to the cylinder surface.

This is a clear example of the importance of a grid selection adapted to the geometry, since in more general cases the FDM discretization of the normal derivative is slightly more complicated. We will come back to this important issue when applying the finite volume method.

Along a constant radial j -line, the Neumann boundary condition can be approximated as

$$\phi_{1,j} = \phi_{2,j} \quad (11.3.31)$$

When introduced in equation (11.3.20), written for $i = 2$, we obtain

$$\begin{aligned} & \frac{2}{(r_{3,j} - r_{1,j})} \left(r_{5/2,j} \frac{\phi_{3,j} - \phi_{2,j}}{r_{3,j} - r_{2,j}} \right) \\ & + \frac{1}{\Delta\theta} \left(\frac{1}{r_{2,j+1/2}} \frac{\phi_{2,j+1} - \phi_{2,j}}{\Delta\theta} - \frac{1}{r_{2,j-1/2}} \frac{\phi_{2,j} - \phi_{2,j-1}}{\Delta\theta} \right) = 0 \end{aligned} \quad (11.3.32)$$

once $\phi_{2,j}$ is obtained from the solution of the algebraic system, the wall value is known from equation (11.3.31).

You will notice that the accuracy of the forward difference (11.3.31) will be enhanced if the distance between the two first radii at $i = 2$ and $i = 1$ is small. This adds to the requirements for a higher grid density close to the solid boundaries.

On the far-field side, we have to apply the Dirichlet condition (11.2.9), where we put $\phi_0 = 0$, leading to, if N_i is the index of the outer boundary circle

$$\phi_{N_i,j} = \vec{U} \cdot \vec{x}_{N_i,j} = U r_{N_i,j} \cos \theta_{N_i,j} \quad (11.3.33)$$

This value will be inserted in the system equation (11.3.27), written for $i = N_i - 1$.

We have hereby completed the algebraic system of unknowns for the potential function by combining equations (11.3.32) for $i = 2$, (11.3.27) from $i = 3$ to $i = N_i - 1$, while equation (11.3.33) fixes the values at the outer boundary $i = N_i$.

This system can be written in condensed form, defining the coefficients $a(i, j)$, $b(i, j)$, $c(i, j)$, $d(i, j)$, $e(i, j)$ and the right-hand side $f(i, j)$, by

$$a(i, j)\phi_{i+1,j} + b(i, j)\phi_{i-1,j} + c(i, j)\phi_{i,j+1} + d(i, j)\phi_{i,j-1} - e(i, j)\phi_{i,j} = f(i, j) \quad (11.3.34)$$

with the consistency condition

$$e(i, j) = a(i, j) + b(i, j) + c(i, j) + d(i, j) - f(i, j) \quad (11.3.35)$$

For $i = 2$, we have

$$\begin{aligned} & \frac{1}{(r_{3,j} - r_{1,j})} \left((r_{3,j} + r_{2,j}) \frac{\phi_{3,j} - \phi_{2,j}}{r_{3,j} - r_{2,j}} \right) \\ & + \frac{1}{\Delta\theta} \left(\frac{1}{r_{2,j}} \frac{\phi_{2,j+1} - \phi_{2,j}}{\Delta\theta} - \frac{1}{r_{2,j}} \frac{\phi_{2,j} - \phi_{2,j-1}}{\Delta\theta} \right) = 0 \\ & a(2, j) = \frac{1}{(r_{3,j} - r_{1,j})} \frac{r_{3,j} + r_{2,j}}{r_{3,j} - r_{2,j}} \quad b(2, j) = 0 \\ & c(2, j) = \frac{1}{r_{2,j}(\Delta\theta)^2} \quad d(2, j) = \frac{1}{r_{2,j}(\Delta\theta)^2} \quad f(2, j) = 0 \\ & e(2, j) = a(2, j) + c(2, j) + d(2, j) \end{aligned} \quad (11.3.36)$$

For $i = 3$ to $Ni - 2$

$$\begin{aligned}
 & \frac{1}{(r_{i+1,j} - r_{i-1,j})} \left((r_{i+1,j} + r_{i,j}) \frac{\phi_{i+1,j} - \phi_{i,j}}{r_{i+1,j} - r_{i,j}} - (r_{i-1,j} + r_{i,j}) \frac{\phi_{i,j} - \phi_{i-1,j}}{r_{i,j} - r_{i-1,j}} \right) \\
 & + \frac{1}{\Delta\theta} \left(\frac{1}{r_{i,j}} \frac{\phi_{i,j+1} - \phi_{i,j}}{\Delta\theta} - \frac{1}{r_{i,j}} \frac{\phi_{i,j} - \phi_{i,j-1}}{\Delta\theta} \right) = 0 \\
 a(i, j) &= \frac{1}{(r_{i+1,j} - r_{i-1,j})} \frac{r_{i+1,j} + r_{i,j}}{r_{i+1,j} - r_{i,j}} \quad b(i, j) = \frac{1}{(r_{i+1,j} - r_{i-1,j})} \frac{r_{i,j} + r_{i-1,j}}{r_{i,j} - r_{i-1,j}} \\
 c(i, j) &= \frac{1}{r_{i,j}(\Delta\theta)^2} \quad d(i, j) = \frac{1}{r_{i,j}(\Delta\theta)^2} \quad f(i, j) = 0 \quad (11.3.37)
 \end{aligned}$$

For $i = Ni - 1$, the system generates a right-hand side as a consequence of the boundary condition (11.3.33), leading to

$$\begin{aligned}
 & \frac{1}{(r_{Ni,j} - r_{Ni-2,j})} \left((r_{Ni,j} + r_{Ni-1,j}) \frac{\phi_{Ni,j} - \phi_{Ni-1,j}}{r_{Ni,j} - r_{Ni-1,j}} \right. \\
 & \quad \left. - (r_{Ni-2,j} + r_{Ni-1,j}) \frac{\phi_{Ni-1,j} - \phi_{Ni-2,j}}{r_{Ni-1,j} - r_{Ni-2,j}} \right) \\
 & + \frac{1}{\Delta\theta} \left(\frac{1}{r_{Ni-1,j}} \frac{\phi_{Ni-1,j+1} - \phi_{Ni-1,j}}{\Delta\theta} - \frac{1}{r_{Ni-1,j}} \frac{\phi_{Ni-1,j} - \phi_{Ni-1,j-1}}{\Delta\theta} \right) = 0 \\
 a(Ni - 1, j) &= 0 \quad b(Ni - 1, j) = \frac{1}{(r_{Ni,j} - r_{Ni-2,j})} \frac{r_{Ni-1,j} + r_{Ni-2,j}}{r_{Ni-1,j} - r_{Ni-2,j}} \\
 c(Ni - 1, j) &= \frac{1}{r_{Ni-1,j}(\Delta\theta)^2} \quad d(Ni - 1, j) = \frac{1}{r_{Ni-1,j}(\Delta\theta)^2} \\
 f(Ni - 1, j) &= -\frac{1}{(r_{Ni,j} - r_{Ni-2,j})} \frac{r_{Ni,j} + r_{Ni-1,j}}{r_{Ni,j} - r_{Ni-1,j}} U r_{Ni,j} \cos \theta_{Ni,j} \\
 e(i, j) &= \frac{1}{(r_{Ni,j} - r_{Ni-2,j})} \frac{r_{Ni,j} + r_{Ni-1,j}}{r_{Ni,j} - r_{Ni-1,j}} + b(Ni - 1, j) \\
 & + c(Ni - 1, j) + d(Ni - 1, j) \quad (11.3.38)
 \end{aligned}$$

We are now ready to move to the next step, the resolution of the algebraic system.

11.3.1.3 Solve the algebraic system

To solve the system (11.3.34), we refer you to Chapter 10 and you can apply any of the presented methods.

We suggest that you start with the Jacobi method, which can be applied in a straightforward way.

Next you can program the Gauss–Seidel method, taking into account that the coefficients $b(i, j)$ and $d(i, j)$ coefficients form the lower diagonal matrix E of equation (10.1.13).

11.3.1.4 Analyze the results and evaluate the accuracy

Select a grid among the three options described above, selecting a distance of 40 times the radius $a = 1$, for the outer boundary of the computational domain. Define a series of grids $N_i * N_j$ with $33 * 128$, $17 * 64$, $9 * 32$, $5 * 16$ mesh points. The first number N_i refers to the circular mesh lines and the second number N_j to the radial lines.

Perform the following tests:

- Monitor the convergence rate by plotting the residual in function of the iteration number.
- Compare the convergence rates of Jacobi and Gauss–Seidel methods.
- Plot the wall pressure coefficients C_{pS} and compare with the exact solution for different grid densities. Compare in particular the uniform grid option of Figure 11.3.4 with the other two options, where the grid density in clustered near the solid surface.
- Compare the velocity components with the exact values by applying appropriate finite difference formulas to the potential mesh point values, based on the definitions (11.3.11), since the mesh lines follow the cylindrical coordinates.

Here again you are faced with various options, as you can choose to derive the velocity components in the mesh points (i, j) , or at the mid-face values $(i \pm 1/2, j)$ and $(i, j \pm 1/2)$. Referring to Figure 11.3.5, you can apply the following formulas of nominally second order accuracy.

At mid-points:

$$\begin{aligned} (u_r)_{i \pm 1/2, j} &= \frac{\phi_{i \pm 1, j} - \phi_{i, j}}{r_{i \pm 1, j} - r_{i, j}} \\ (u_\theta)_{i, j \pm 1/2} &= \pm \frac{\phi_{i, j \pm 1} - \phi_{i, j}}{r_{i, j} \Delta \theta} \end{aligned} \quad (11.3.39)$$

At the mesh points, a good approximation is provided by

$$\begin{aligned} (u_r)_{i, j} &= \frac{1}{2} [(u_r)_{i+1/2, j} + (u_r)_{i-1/2, j}] \\ (u_\theta)_{i, j} &= \frac{1}{2} [(u_\theta)_{i, j+1/2} + (u_\theta)_{i, j-1/2}] \end{aligned} \quad (11.3.40)$$

You can also apply other formulas based on Chapter 4, Section 4.3.

Compare the numerical values of the velocity components with their exact values.

- Calculate lift and drag by applying equation (11.1.38). In the present case, this can be calculated as follows:

$$\begin{aligned} \vec{L} &= - \oint_{S_b} p \, d\vec{S} = - \oint_{S_b} p \, dx \, \vec{e}_y + \oint_{S_b} p \, dy \, \vec{e}_x \\ &= \oint_{S_b} p r \, d\theta \, \vec{e}_r \\ &= \oint_{S_b} p r \cos \theta \, d\theta \, \vec{e}_x + \oint_{S_b} p r \sin \theta \, d\theta \, \vec{e}_y \end{aligned} \quad (11.3.41)$$

based on the relations between the cylindrical and Cartesian coordinates, on the cylinder surface of radius $r = a$

$$\begin{aligned} d\vec{S} &= -dy \vec{e}_x + dx \vec{e}_y = -r d\theta \vec{e}_r = -r d\theta (\cos \theta \vec{e}_x + \sin \theta \vec{e}_y) \\ &= -r \cos \theta d\theta \vec{e}_x - r \sin \theta d\theta \vec{e}_y \end{aligned} \quad (11.3.42)$$

The integrals in the last line of equation (11.3.41) are easily evaluated numerically by applying a trapezium formula, over the N_j points in the circular direction.

For the circular cylinder, apply

$$\begin{aligned} L_y &= a \sum_{j=1}^{N_j-1} \frac{1}{2} [p_{1,j} \sin \theta_{1,j} + p_{1,j+1} \sin \theta_{1,j+1}] \Delta \theta \\ L_x &= a \sum_{j=1}^{N_j-1} \frac{1}{2} [p_{1,j} \cos \theta_{1,j} + p_{1,j+1} \cos \theta_{1,j+1}] \Delta \theta \end{aligned} \quad (11.3.43)$$

- Check the influence of the far-field boundary position by comparing the wall pressure for ratios of distance to radius, ranging from 20 to 80.
- Apply a series of grids: $33 * 128$, $17 * 64$, $9 * 32$, $5 * 16$ and plot the error (for instance the L_2 -norm of the surface pressure) in function of number of mesh points in log scale. The slope should be close to 2 for second order accuracy.

11.3.2 Compressible Potential Flow Around the Circular Cylinder

Extend your program to handle compressibility by introducing the density based on equation (11.2.4).

The modification to the numerical scheme (11.3.27) is straightforward and leads to the scheme:

$$\begin{aligned} &\frac{1}{(r_{i+1/2,j} - r_{i-1/2,j})} \left(\rho_{i+1/2,j} r_{i+1/2,j} \frac{\phi_{i+1,j} - \phi_{i,j}}{\Delta r_i} \right. \\ &\quad \left. - \rho_{i-1/2,j} r_{i-1/2,j} \frac{\phi_{i,j} - \phi_{i-1,j}}{\Delta r_{i-1}} \right) \\ &+ \frac{1}{\Delta \theta} \left(\frac{\rho_{i,j+1/2} \phi_{i,j+1} - \phi_{i,j}}{r_{i,j+1/2} \Delta \theta} - \frac{\rho_{i,j-1/2} \phi_{i,j} - \phi_{i,j-1}}{r_{i,j-1/2} \Delta \theta} \right) = 0 \end{aligned} \quad (11.3.44)$$

However, two additional complications arise now, since the density is not constant and secondly its dependence on the potential function is not linear. Hence, you have to make choices on these two issues.

11.3.2.1 Numerical estimation of the density and its nonlinearity

The density is function of the velocity, following equation (11.2.3) and you can apply directly formulas (11.3.39) to obtain the densities at these mid-point values.

However, you need the two velocity components in the same mesh points:

$$\begin{aligned}\rho_{i\pm 1/2,j} &= \rho_0 \left[1 - \frac{(\vec{v}^2)_{i\pm 1/2,j}}{2H} \right]^{1/(\gamma-1)} \\ (\vec{v}^2)_{i\pm 1/2,j} &= (u_r^2)_{i\pm 1/2,j} + (u_\theta^2)_{i\pm 1/2,j}\end{aligned}\quad (11.3.45)$$

We leave it to you now to apply various formulas for the u_θ components at the points $(i \pm 1/2, j)$, and similarly for the u_r components at the points $(i, j \pm 1/2)$.

In equation (11.3.44), the density depends on the potential values and you have to linearize the system by either an explicit or an implicit method. The simplest method is obviously the explicit option, whereby you estimate the density based on the velocities of the previous iteration.

You can also consider improving this approximation by performing a few local iterations on this nonlinear treatment, for each of the relaxation iterations.

Solve the compressible potential flow for a low Mach number value, for instance, $M = 0.05$, and compare with the incompressible exact solution. As seen in Table 11.1.1, at this low value of Mach number the compressibility effect is of the order of 0.0625% of the dynamic pressure, which should correspond to a deviation on the velocity of the order of 0.03125%. Even at $M = 0.1$, the compressibility effect on the dynamic pressure is of 0.25% or 0.125% on the velocity.

11.3.2.2 Transonic potential flow

If you push your curiosity to increase the incident Mach number for the compressible version of your program, say at $M = 0.6$ or higher, a shock will appear on the upper surface of the cylinder in the region around $\theta = 90^\circ$ and your program will ‘blow up’ and no convergent solution will be possible.

This is due to the fact that the potential equation becomes hyperbolic in the variables (x, y) in the supersonic regions, as seen in Chapter 3. The method to cure this problem is to take into account the physical properties of hyperbolic supersonic flows at the level of the discretization.

The first successful computation of a steady transonic potential flow was obtained by Murman and Cole (1971) for the small disturbance equation in two dimensions. This basic work marked a breakthrough that initiated considerable activity in this field, giving rise to an extremely rapid development which led, in about 10 years time, to the situation where the computation of transonic potential flows could be considered as a practically solved problem. A large number of operational codes exist by now, which compute three-dimensional transonic potential flows in a few seconds of CPU time on the most advanced computers, (Holst and Thomas, 1983).

The original idea of Murman and Cole consisted of using different finite difference formulas in the supersonic and subsonic regions. In the subsonic, elliptic, region a central difference is adequate and compatible with the physics of diffusion; while in the supersonic, hyperbolic, region the direction of propagation has to be respected, which leads to the choice of an upwind difference. As with many ideas which appear simple afterwards, the original development required deep understanding of the underlying problems both numerical and physical.

This work was a major landmark in the history of CFD. It is a beautiful example of how new discoveries and progress are made in science in general, and CFD in particular. It is fascinating and instructive to read the historical account of the genesis of these ideas, as reported by Hall (1981) and we hope that you will share our pleasure in quoting this account:

Earll Murman had been working for a year or so at Boeing on finite difference methods for integrating the compressible Navier–Stokes equations when, in 1968, Julian Cole arrived on a 1-year visit.

Cole writes: *‘It was Goldberg who suggested that transonic flow was a timely subject. I decided on a joint analytical and numerical approach and he said that Earll and I could work together (since my programming was feeble)’. Our approach was founded on several bits of previous experience.*

- (iii) *The fact that (the) Lax–Wendroff (scheme) could give the correct shock jumps (had) made a deep impression and I (had) learned about artificial viscosity, diffusion and dispersion of difference schemes. Yosh (Yoshihara) was convinced that steady flows could not be calculated directly but I decided while at Boeing to try using a conservative scheme (a la Lax) in order to catch shocks.*
- (iv) *I was aware of Howard Emmons very early ‘successful’ relaxation calculations of mixed flows in nozzles and decided to try a relaxation method.*
- (v) *I had studied the fundamentals of small disturbance theory ... rather extensively earlier. I knew it had all the essential difficulties and could even be a good approximation. It was clear that it would make the numerical work easier.*

Murman writes that Cole ‘spent several months systematically deriving a (transonic) small disturbance (TSP) theory from the complete Euler equations. It laid the theoretical groundwork for our later developments. In January 1969 we started some computations solving Laplace equations and then the TSP using centered finite differences. By April we found that we could not get the calculations to converge for supercritical flow. It was in the following several months that we hit upon the idea of switching and type dependent schemes. I believe that the idea grew out of an afternoon brainstorming session when we were discussing finite difference methods for elliptic and hyperbolic problems and how the two were basically different. Julian, I believe, threw out a comment that maybe we could combine them somehow.

I have often reflected back on that event to realize how important it is in research to be open-minded, imaginative, and receptive to unconventional suggestions’.

Cole adds, *‘I knew enough numerical analysis to know that hyperbolic schemes were unstable if the domain of dependence was incorrect. Even though the time-like direction was unclear I thought that perhaps we should have only downstream influence. So we decided to switch schemes: explicit hyperbolic was ruled out by the CFL condition near the sonic line’.*

Murman continues *‘My experience the previous year on the Navier–Stokes computations allowed us to make rapid progress. It was clear that we should*

maintain conservation form to calculate shock waves. Unfortunately we missed the essential point of the shock point operator. For stability reasons, the hyperbolic operator had to be implicit. This naturally led to a line relaxation algorithm so that the method would work in the limits of both purely supersonic and purely subsonic flow. In July we programmed up the first code and it worked almost immediately'.

After this initial work, Murman and Cole's procedure was extended to three dimensions by Ballhaus and Bailey (1972); to the non-conservative full potential equation for two dimensions by Steger and Lomax (1972), Garabedian and Korn (1972), and three dimensions by Jameson (1974). The conservative full potential equation was solved initially by Jameson (1975) for two-dimensional flows and extended to three-dimensional configurations by Jameson and Caughey (1977).

These developments will be dealt with in Volume II.

11.3.3 Additional Optional Tasks

If you have successfully achieved the tasks suggested in the previous section, you have now a good basis to go a step further and extend your experience to other options. We list here a few of them for your consideration, and of course you can think of many others, based on your personal interest and curiosity:

- Apply other iterative methods, such as point and line relaxation methods, as described in Chapter 10.
- The streamfunction ψ is also a solution of the Laplace equation:

$$\Delta\psi = 0 \quad (11.3.46)$$

with the boundary conditions

$$\begin{aligned} \psi &= 0 && \text{on the solid surface} \\ \psi &= U_y && \text{in the far field} \end{aligned} \quad (11.3.47)$$

You can apply the same discretization as defined by equation (11.3.27), the only change will occur for the values $i = 2$ and $i = Ni - 1$, due to the change in boundary condition.

11.4 FINITE VOLUME DISCRETIZATION OF THE EULER EQUATIONS

This section introduces you to the numerical solutions for the system of Euler equations in two dimensions, and we will guide you through the steps required to solve the same inviscid flow around the cylinder as treated in the previous section, as well as two new test cases. One is typical of internal flows and is represented by the flow on a bump placed on a flat plate, while the second case is a supersonic flow over a wedge with an oblique shock, introducing you to some of the issues related to the numerical simulation of shocks.

In the subsonic range, the solutions of the Euler equations for uniform inflow conditions should be identical to the potential flow solutions, although we have to

solve four differential equations, instead of a single one for the potential equation. But the main difference, as stated in the introduction to this chapter, is that Euler equations do not guarantee that the calculated flow remains irrotational or isentropic. Although equation (11.1.5) indicates that in an inviscid subsonic flow with uniform inlet conditions, entropy remains constant and uniform in the whole flow domain, you should be aware that the numerical dissipation generated by the numerical scheme will ‘mimic’ in some way the physical dissipation of viscous flows. Seen from the point of view of the discretized system, we have to realize indeed that the computer cannot distinguish physical dissipation from numerical dissipation. *Therefore, the numerical system will obey equation (11.1.33) instead, where the right-hand side represents the global dissipation of the numerical model. As a consequence, entropy will not remain constant and we can consider it as a unique indicator of the presence of numerical dissipation in the scheme. In regions where the calculated entropy increases, we can be assured that these regions are influenced by numerical dissipation.* This is a very important property, which provides a direct measure of the quality of the numerical scheme on the selected grid, by monitoring and post-processing the evolution of entropy. Since the numerical dissipation is proportional to a power of the mesh size, the only way to reduce the dissipation of the selected scheme is to refine the grid in the regions where an excessive entropy generation would occur. This will be the case in regions with high-velocity gradients, such as leading edges, trailing edges of airfoils, or regions with abrupt geometry changes, such as sharp corners, and these regions will require higher mesh densities to keep the numerical dissipation influence below an acceptable level.

The system of time-dependent Euler equations is hyperbolic in space and time and we will select the time-dependent numerical formulation, as recommended in the previous chapters, to find the steady state solution.

There is however a problem with incompressible flows, since taking constant density removes the time-dependent term in the continuity equation and this requires special methods to overcome this absence, namely methods of *artificial compressibility* (Chorin, 1967) also called *preconditioning methods* (Merkle and Choi, 1985). These methods extend the time-dependent approach to cover all speed regimes, including the incompressible limit, but their presentation is outside the scope of this introductory text. It will be treated in more details in Volume II.

Hence, we will handle the cylinder case in the domain of small Mach numbers, for instance at values below $M = 0.1$, where compressibility is present but remains small, so that a comparison with the incompressible solution remains meaningful.

We will select a most representative scheme of the family of separate space and time discretizations, combining a centered space discretization with an explicit Runge–Kutta method, as described in Sections 9.3.4 and 9.3.6.

Also, we will apply here a *finite volume method*, in order to generate a program that you can apply to any grid configuration.

11.4.1 Finite Volume Method for Euler Equations

We refer you to Chapter 5 for the general formulation of finite volume methods (FVM) and their practical discretization. If needed, we suggest you study again this chapter to refresh your memory with the main properties of finite volume methods. In particular,

the most general formulation is provided by equation (5.2.7), repeated here, for the case of the homogenous Euler equations without source terms, as

$$\frac{d}{dt} [\bar{U}_{i,j} \Omega_{i,j}] = - \sum_{\text{faces}} \vec{F}^* \cdot \Delta \vec{S} \equiv -R_{i,j} \quad (11.4.1)$$

The right-hand side defines the **residual** $R_{i,j}$ as the balance of fluxes over all the faces forming the cell (i,j) .

Some of the essential properties of the FVM are:

- The solution $\bar{U}_{i,j}$ of the system (11.4.1) is the **cell-averaged value** of the conservative variable U over the cell (i,j) .
- When the results of the simulation are post-processed, we will assign the cell-averaged values to the center of the cell. **This introduces an error, generally of second order**, which is part of the discretization error.
- The numerical flux \vec{F}^* represents the discretization of the physical fluxes, as defined by the selected numerical scheme.

11.4.1.1 Space discretization

For two-dimensional structured grids, we can select either a cell-centered or a cell-vertex approach. We will select here the cell-centered method, which is also a most current approach as illustrated in Figure 11.4.1.

In the cell-centered approach the grid coordinates, as read-in from your mesh input file, are the points such as A, B, C, D, while we attribute the numerical coordinates (i,j) to a cell-center point such as $P(i,j)$. Its coordinates are obtained by the arithmetic average of the four corner cells A, B, C, D. For instance, point $P(i,j)$ is defined by its coordinates

$$\vec{x}_{i,j} = \frac{1}{4}(\vec{x}_A + \vec{x}_B + \vec{x}_C + \vec{x}_D) \quad (11.4.2)$$

and similarly for the other centers of the control volumes.

Figure 11.4.1 shows the relation between a curvilinear grid, formed for instance by the circular arcs of the cylinder mesh applied for the potential solution, and the associated quadrilateral grid that is seen by the program, when each cell side is considered as formed by a straight line. This is typical for second order approximations and is compatible with second order numerical schemes. If higher order schemes would be considered, then a more accurate representation of the cell sides, as curved boundaries, would have to be considered.

To construct your main program, proceed as follows with each step forming the subject of a separate subroutine:

- Read the grid coordinates.
- Define the cell-center coordinates, the cell areas and the face normals, based on the formulas defined in equations (5.3.2):

$$\Delta \vec{S}_{i+1/2,j} = \Delta \vec{S}_{AB} = \Delta y_{AB} \vec{I}_x - \Delta x_{AB} \vec{I}_y = (y_B - y_A) \vec{I}_x - (x_B - x_A) \vec{I}_y \quad (11.4.3)$$

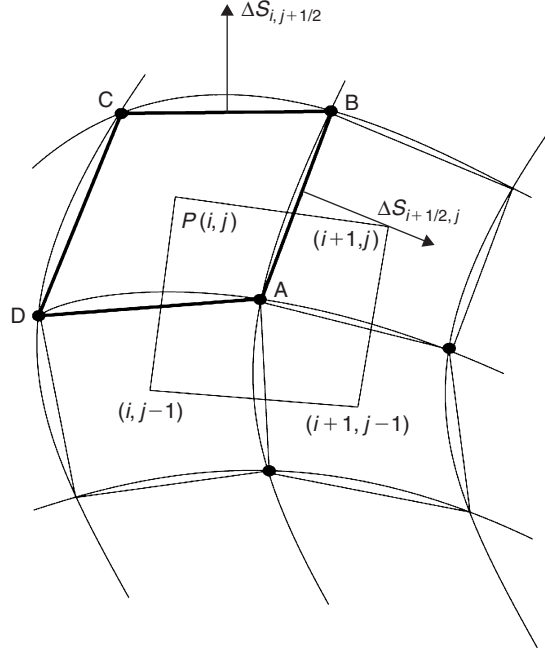


Figure 11.4.1 Cell-centered finite volume space discretization on a structured grid.

and the area, following equation (5.3.4)

$$\begin{aligned}
 \Omega_{i,j} &= \Omega_{ABCD} = \frac{1}{2} |\vec{x}_{AC} \times \vec{x}_{BD}| \\
 &= \frac{1}{2} [(x_C - x_A)(y_D - y_B) - (x_D - x_B)(y_C - y_A)] \\
 &= \frac{1}{2} (\Delta x_{AC} \Delta y_{BD} - \Delta x_{BD} \Delta y_{AC})
 \end{aligned} \tag{11.4.4}$$

- The numerical flux \vec{F}^* is written as the addition of a central flux, representing the central scheme, plus a dissipation term, under the following form:

$$(\vec{F}^* \cdot \Delta \vec{S})_{i+1/2,j} \equiv \left[\frac{1}{2} (\vec{F}_{i,j} + \vec{F}_{i+1,j}) - D_{i+1/2,j} \right] \cdot \Delta \vec{S}_{i+1/2,j} \tag{11.4.5}$$

where the dissipation term $d_{i+1/2,j}$ is either part of the numerical dissipation or/and contains artificial dissipation terms.

- For the central scheme, an artificial viscosity term is added, following the formulation of Section 9.3.6. It is written here as a difference of third order derivatives, with

$$D_{i+1/2,j} = \gamma_{i+1/2,j} (U_{i+2,j} - 3U_{i+1,j} + 3U_{i,j} - U_{i-1,j}) \tag{11.4.6}$$

- Organize the subroutine for the calculation of this artificial dissipation term, as it requires mesh point values from two additional rows of cells. The value of the γ -coefficient is based on the formulation of Jameson et al. (1981), with the spectral radius of the Jacobian as coefficient

$$\gamma_{i+1/2,j} = \frac{1}{2} \kappa^{(4)} [\vec{v} \cdot \Delta \vec{S} + c |\Delta \vec{S}|]_{i+1/2,j} \quad (11.4.7)$$

where c is the speed of sound and $\kappa^{(4)}$ is a non-dimensional coefficient of dissipation, with the minimum value of $\kappa^{(4)} = 1/256$.

- Calculate the cell residuals by *programming a loop over all the cell faces*. For each face, calculate the associated flux, following formula:

$$(\vec{F}^* \cdot \Delta \vec{S})_{i+1/2,j} = (\vec{F}^* \cdot \Delta \vec{S})_{AB} = f_{AB}^* (y_B - y_A) - g_{AB}^* (x_B - x_A) \quad (11.4.8)$$

- Send the contribution to the right cell and its negative value to the left cell, based on an anti-clockwise positive orientation.
- For the faces situated on the boundaries, take into account the particular boundary conditions, as defined hereafter. In particular on a solid boundary face, all the convective fluxes are zero, since the normal velocity v_n has to be zero and the dissipation term is set to zero.

11.4.1.2 Time integration

Apply the fourth stage Runge–Kutta time integration method, following Section 9.3.4, equations (9.3.47–9.3.50), with the following coefficient:

$$\begin{aligned} U_{i,j}^{(1)} &= U_{i,j}^n - \frac{\Delta t}{\Omega_{i,j}} \alpha_1 R_{i,j}^n \\ U_{i,j}^{(2)} &= U_{i,j}^n - \frac{\Delta t}{\Omega_{i,j}} \alpha_2 R_{i,j}^{(1)} \\ U_{i,j}^{(3)} &= U_{i,j}^n - \frac{\Delta t}{\Omega_{i,j}} \alpha_3 R_{i,j}^{(2)} \\ U_{i,j}^{n+1} &= U_{i,j}^n - \frac{\Delta t}{\Omega_{i,j}} \alpha_4 R_{i,j}^{(3)} \end{aligned} \quad (11.4.9)$$

with

$$\alpha_1 = \frac{1}{4} \quad \alpha_2 = \frac{1}{3} \quad \alpha_3 = \frac{1}{2} \quad \alpha_4 = 1 \quad (11.4.10)$$

Another option, which optimizes the dissipation of the scheme, is given by

$$\alpha_1 = \frac{1}{8} \quad \alpha_2 = 0.306 \quad \alpha_3 = 0.587 \quad \alpha_4 = 1 \quad (11.4.11)$$

- Choose the CFL number, under the stability condition $CFL < 2.8$.
- The time step has to be evaluated based on the sufficient condition expressing that the physical domain of dependence should be completely contained in the

numerical domain of dependence. However, as we are not interested in the transient behavior of the solution, we can choose a **local time step**, whereby each cell progresses at its maximum possible time step $\Delta t_{i,j}$. This looses the time consistency of the transient since each cell has its own time step, but provides significant convergence acceleration. Hence, we make $\Delta t_{i,j}$ proportional to the local cell size $\Omega_{i,j}$ and is calculated as follows:

$$\Delta t_{i,j} \leq \text{CFL} \frac{\Omega_{i,j}}{|\vec{v} + c)_{i,j} \cdot \Delta \vec{S}_i| + |(\vec{v} + c)_{i,j} \cdot \Delta \vec{S}_{j+1/2}|}$$

$$\Delta \vec{S}_i = \frac{1}{2}(\Delta \vec{S}_{i+1/2,j} + \Delta \vec{S}_{i-1/2,j}) \quad \Delta \vec{S}_j = \frac{1}{2}(\Delta \vec{S}_{i,j+1/2} + \Delta \vec{S}_{i,j-1/2})$$
(11.4.12)

- To start the calculation you need to define an initial solution. By lack of knowledge it is customary to take a uniform initial solution corresponding to the inlet condition, distributed uniformly over the mesh. Note that the convergence behavior can be very sensitive to the initial solution: the closer this initial guess is to the final solution, the faster the convergence.

Observe here that once again many ‘local’ decisions have to be taken as to where to evaluate the velocities and speed of sound. We have in these simulations, where all quantities vary from point to point, to continuously decide if we evaluate the values at the mesh points or at the cell faces, and how we connect one to the other.

11.4.1.3 Boundary conditions for the Euler equations

The last item of importance in the generation of your program for the Euler equations is the definition of the boundary conditions. This is a most critical component of any CFD code, and has to be compatible with both physical and numerical properties of the problem to be solved.

The time-dependent hyperbolic system of Euler equations contains four unknown dependent variables, and we have to determine how to handle these variables at the boundaries of the computational domain. These boundaries are of three types: solid walls, inlet and outlet boundaries, and each one of them will require a dedicated treatment.

We know from Chapter 3 that the system of time-dependent Euler equations is propagation dominated and we are faced therefore with the following questions:

- (i) How many conditions of physical origin are to be imposed at a given boundary?
- (ii) What physical quantities are to be imposed at a boundary?
- (iii) How are the remaining variables to be defined at the boundaries?

Go back to Chapter 3, Section 3.4.1 and to the Example E.3.4.3 dealing with the isentropic form of the Euler equation, where the energy equation has been removed. The three eigenvalues of the system, which correspond to the speed of propagation

of three characteristic quantities, are given by

$$\begin{aligned}\lambda_1 &= \vec{v} \cdot \vec{k} / \kappa \\ \lambda_2 &= \vec{v} \cdot \vec{k} / \kappa + c \\ \lambda_3 &= \vec{v} \cdot \vec{k} / \kappa - c\end{aligned}\tag{11.4.13}$$

where \vec{k} is the wave number vector. The last two correspond to the speeds of the acoustic waves.

It can be shown, that by adding the energy equation a fourth eigenvalue appears equal to the first one, which becomes double valued.

Since, the transport properties at a surface are determined by the normal components of the fluxes, the number and type of conditions at a boundary of a multi-dimensional domain will be defined by the eigenvalue spectrum of the Jacobian matrices associated to the normal to the boundary. Hence, at a boundary surface, the behavior of the Euler system will be determined by the propagation of waves with the following speeds:

$$\begin{aligned}\lambda_1 &= \vec{v} \cdot \vec{e}_n = v_n \\ \lambda_2 &= \vec{v} \cdot \vec{e}_n = v_n \\ \lambda_3 &= \vec{v} \cdot \vec{e}_n + c = v_n + c \\ \lambda_4 &= \vec{v} \cdot \vec{e}_n - c = v_n - c\end{aligned}\tag{11.4.14}$$

where v_n is the normal velocity component at the considered surface.

This defines locally **quasi-one-dimensional propagation properties** and we can therefore look at how the propagation of information behaves at a boundary, in function of the sign of these quantities. The first two eigenvalues are equal to the normal component of the velocity and correspond to the entropy and vorticity waves, while the two remaining eigenvalues, are associated to the acoustic waves. Hence, the sign of these eigenvalues will be determined by the velocity components normal to the boundary surfaces.

The key to the understanding of the issue of the number of boundary conditions is indeed the awareness that the characteristics convey information in the (n, t) space formed by the local normal direction and time. If λ represents the propagation speed, the trajectory of the corresponding information path is given by $dn/dt = \lambda$. When information is propagated from outside toward the inside of the computational domain, it means that this information has to be defined from outside and represents a **physical boundary condition**. When λ is positive, the information carried by the associated characteristics, propagates from the boundary toward the interior of the flow domain and a **physical boundary condition** has to be imposed. On the other hand, when the eigenvalue λ is negative and the propagation occurs from the interior of the domain toward the boundary, it means that the related information is determined at the boundary by the interior flow and *cannot* be imposed from the outside. It will have to be expressed numerically, through **numerical boundary conditions**.

In summary, the number of physical conditions to be imposed at a boundary with normal vector \vec{n} , pointing toward the flow domain, is defined by the number of characteristics entering the domain.

Referring to Figure 11.4.2 for an inlet boundary, if the inlet flow is subsonic in the direction normal to the inlet surface, three eigenvalues are positive and one, $\lambda_4 = v_n - c$ is negative. **Therefore, three quantities will have to be fixed by the physical flow conditions at the inlet of the flow domain, while the remaining one will be determined by the interior conditions, through a numerical boundary condition.**

At an outlet boundary (Figure 11.4.3) with subsonic normal velocity, three eigenvalues are negative, since the normals are defined as pointing toward the interior flow domain. Three numerical boundary conditions have therefore to be set, while the fourth condition, associated to the positive eigenvalue $(-|v_n| + c)$, propagates

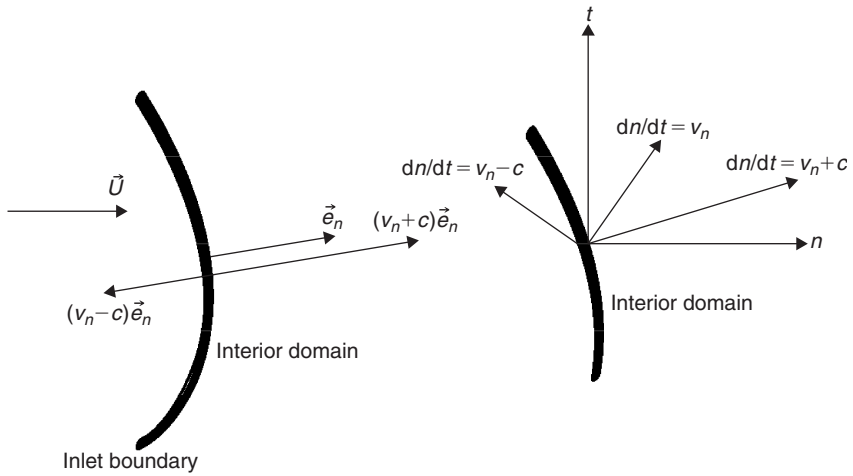


Figure 11.4.2 Characteristic propagation properties at an inlet boundary with subsonic conditions.

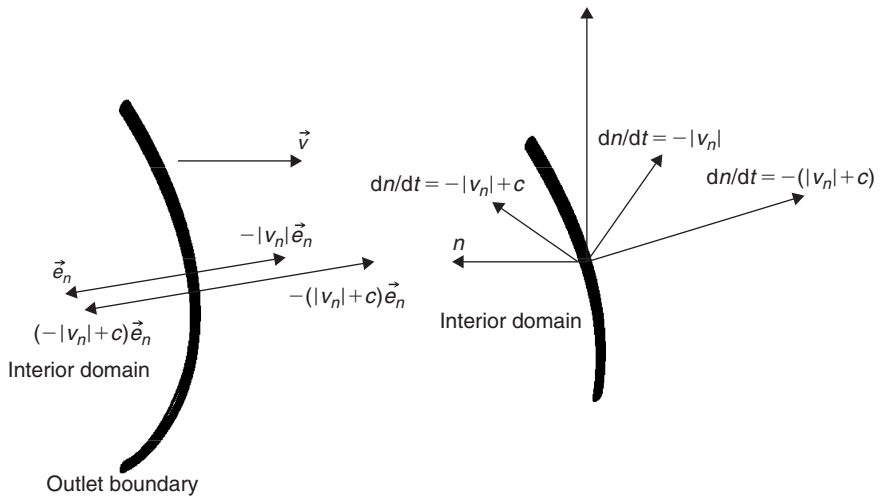


Figure 11.4.3 Characteristic propagation properties at an outlet boundary with subsonic conditions.

information from the boundary toward the flow region. It is consequently associated to a physical boundary condition.

This is a very important result, as it indicates that we cannot fix all four quantities at a subsonic inlet, but only three of them, neither can we know all quantities at a subsonic outlet, where we have to impose one variable, the other three being defined from the inlet flow properties.

If the flow is supersonic normal to the inlet surface, all boundary conditions are physical. With the same circumstances at outlet, all eigenvalues are of negative sign and no physical conditions have to be given. All the boundary variables are defined by the interior flow, for instance via extrapolation formulas.

The next question to answer is: ***What quantities should be fixed as physical boundary conditions?***

This question has no unique answer and forms a vast complex subject, which is outside the scope of this introductory text. It is treated more in details in Volume II. For practical applications, you can proceed as follows:

Inlet boundary: You can select to fix the inlet velocity and inlet temperature for an external flow problem.

Another option, often applied to internal flows such as channels or cascade computations, is to specify two thermodynamic variables such as the upstream stagnation pressure and temperature, and an inlet Mach number or velocity magnitude, and have the inlet flow angle defined by the computed flow, or inversely, fix the incident flow angle, determining inlet Mach number from the computed flow. This has as consequence that the mass flow is not defined, but is a result from the computation.

Outlet boundary: The most appropriate physical condition, particularly for internal flows and corresponding to most experimental situations, consists in fixing the downstream static pressure. This can also be applied for external flow problems. However in this latter case, free stream velocity could also be imposed.

At a solid wall boundary: The normal velocity is zero, since no mass or other convective flux can penetrate the solid body. Hence, only one eigenvalue is positive and only one physical condition can be imposed, namely $v_n = 0$. The other variables at the wall, in particular tangential velocity components and pressure have to be determined by extrapolation from the interior to the boundary (Figure 11.4.4).

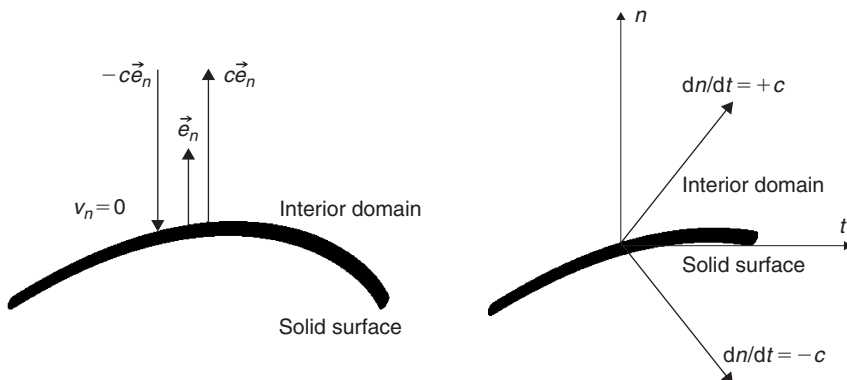


Figure 11.4.4 *Characteristic propagation properties at a solid surface boundary.*

Numerical boundary conditions: The other issue to be fixed in your program is to implement the numerical boundary conditions, in order to obtain the numerical values of the remaining variables at the boundaries. This is particularly important for solid walls, where we want to determine the pressure variations.

We suggest you apply a simple extrapolation from the inside point to the next surface point along the mesh lines. This is strictly valid if they are orthogonal to the surface, but should be acceptable for regular grids of the type suggested here. An even simpler way is to take the value at the boundary equal to the value at the cell center of the associated cell. This assumes the quantity to be piecewise constant in the considered cell and is a form of zero order extrapolation.

A more accurate way, but less robust, is to perform a linear extrapolation from two cells in the normal direction.

11.5 NUMERICAL SOLUTIONS FOR THE EULER EQUATIONS¹

We will treat here in detail several test cases, in order to apply the numerical methods to a variety of flow conditions and identify the associated numerical effects. We will handle first the cylinder flow, in order to compare with the potential flow model and illustrate in particular the effects of numerical dissipation through the entropy field.

The next case will be typical of internal channel flows for several Mach number flow conditions; while the last case will be representative of supersonic flows with an oblique shock. All these cases will be treated assuming *perfect gas* relations for the considered fluid.

11.5.1 Application to the Flow Around a Cylinder

Proceed as follows:

- Consider the grids applied in the previous section, following equation (11.3.18) with 33 points in the radial direction and 128 points in the circumferential direction (see Figure 11.5.1). By removing every second point in both directions, new grids with a reduced density can be constructed, defining hereby a fine mesh (33×128), an intermediate mesh (17×64) and a coarse mesh (9×32).
- Assume atmospheric conditions, for pressure and temperature with $p_a = 101300$ Pa, $T_a = 288$ K, and take $M = 0.1$ as incoming flow conditions.
- In order to match these inlet conditions, the total pressure and total temperature are fixed at the inlet to 102010 Pa and 288.6 K, respectively, by following equations (11.1.9) and (11.1.12). The inlet velocity direction is imposed along the x -direction. The atmospheric pressure is also imposed at the outlet. Since the computational domain is circular, we will consider that the left half circle of the outer boundary forms the inlet section, while the right half circle will form the outlet section of the computational domain.
- Apply the Runge–Kutta method (11.4.9) with the coefficients (11.4.11) and take $CFL = 2$. You can choose here between a local time step based on equations

¹ This section has been written with the active participation of Dr. Benoit Tartinvill, who produced also the results of the three test cases.

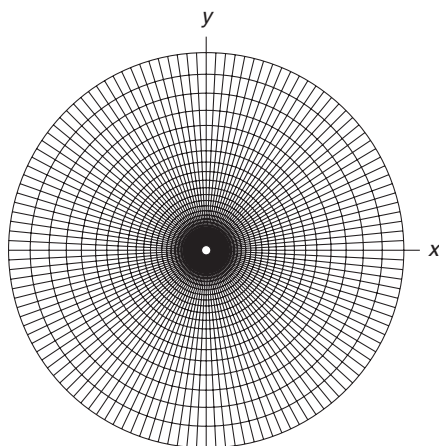


Figure 11.5.1 *Fine mesh used for the circular cylinder flow.*

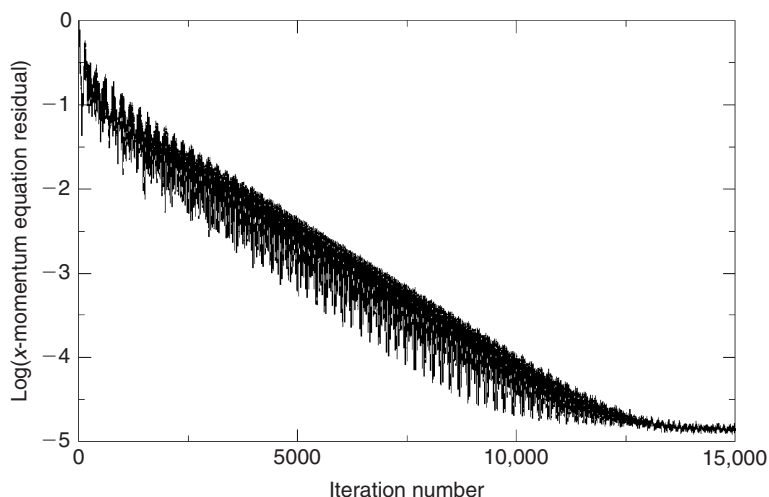


Figure 11.5.2 *Convergence history of the normalized L_2 -norm of the momentum residual for the calculation on the fine mesh.*

(11.4.12), or a global time step by taking the minimum value of all the local time steps. We choose here the first alternative, but you can experience with both options.

- To start the calculation take a uniform initial solution equal to the values defined by the boundary conditions.
- Solve the Euler equations for the different grid densities considered, until a steady state is reached, by selecting a value of the artificial dissipation coefficient $\kappa^{(4)} = 1/50$, in equation (11.4.7).
- Monitor and plot the L_2 -norm of the residuals in function of number of time steps. Figure 11.5.2 shows the evolution of the axial velocity residual, normalized by

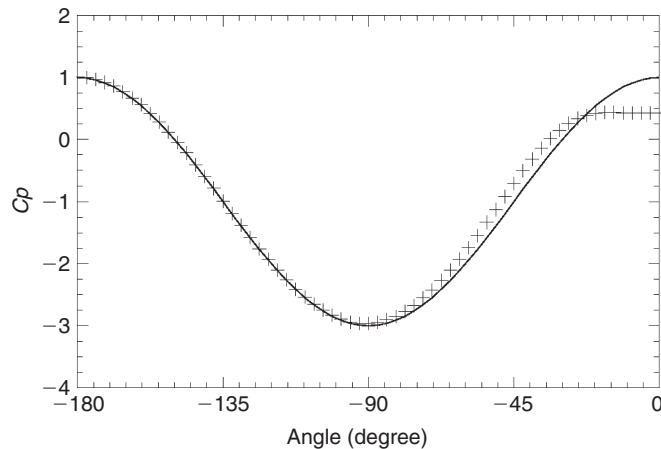


Figure 11.5.3 *Cumferential distribution of pressure coefficient from the numerical result (plus signs), compared to the exact solution (continuous line).*

the residual of the initial solution for the finest mesh, with the ordinate plotted on a logarithmic scale. The residual has dropped by nearly 5 orders of magnitude, which is close to single precision machine accuracy.

- If you cannot reach machine accuracy, if the residual drops for instance by 3 orders of magnitude only, it will not necessarily mean that your solution is not valid, but it indicates that somewhere in the flow domain, often around the boundaries, there is a local error source. This error could result from an inaccuracy in the implementation or from some small local oscillation between values at neighboring points. This is called a '*limit cycle*' in the convergence behavior. You can sometimes detect it by looking at the flow regions with the highest values of the residual.
- Observe that the convergence requires many time steps before reaching machine accuracy. To accelerate the convergence, you would need to apply the multigrid method, which would reduce the number of iterations easily by a factor of the order of 10 or more. Alternatively you could apply an implicit method, which is however also more complicated to implement.
- Note also the oscillatory behavior of the residual. This is generally an indication that the acoustic waves, which transport part of the initial errors, are reflected at the boundaries prior to their damping by the numerical scheme. This occurs when the boundary condition implementation does not allow these error waves to leave the computational domain without being reflection. The imposition of the outlet pressure is typical of what is called a **reflecting boundary condition**. To avoid these effects, we have to impose **non-reflecting boundary conditions**, based on the characteristic properties associated to the reflected waves. This is outside the scope of this introductory text and will be dealt with in more details in Volume II.
- Plot the wall surface pressure coefficient and compare with the exact incompressible solution, equation (11.3.17), as shown on Figure 11.5.3.

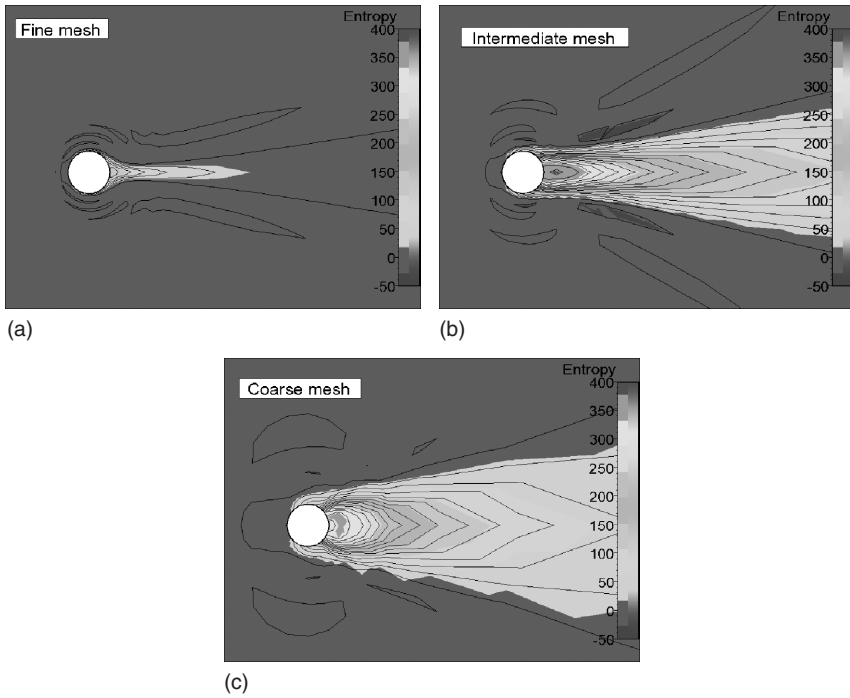


Figure 11.5.4 Distribution of entropy as computed on three different meshes: (a) fine mesh, (b) intermediate mesh and (c) coarse mesh (for color image refer Plate 11.5.4).

- In the front part of the cylinder, between the points A and C of Figure 11.3.1, the solution is nearly perfect. However in the back part, between C and B a growing discrepancy can be noticed, with a significant deviation in the wake region around point B. You can observe that the left–right symmetry of the exact solution is lost. This is a major property of Euler calculations and is due, as explained above, to the numerical dissipation and the related numerical generation of entropy, as seen from Figures 11.5.4.
- Plot the entropy isolines and color maps, by monitoring the quantity:

$$s - s_{\infty} = \frac{p}{\rho^{\gamma}} - \left(\frac{p}{\rho^{\gamma}} \right)_{\infty} \quad (11.5.1)$$

where the subscript ∞ refers to the far field upstream conditions.

- For an inviscid flow, the entropy has to remain constant and equal to its upstream value, which is clearly not the case for the results of the Euler calculations on the three grids, as you see from Figure 11.5.4. Looking at the entropy evolution along the axis of symmetry and the cylinder surface, you observe that the entropy increases already from the upstream stagnation point (point A of Figure 11.3.1) along the solid surface, which appears as a source of entropy, like in a viscous boundary layer and keeps increasing further downstream, forming a wake.

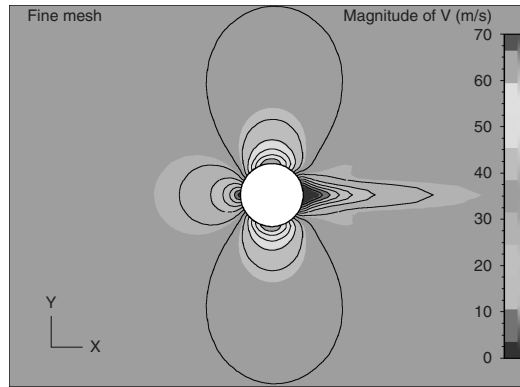


Figure 11.5.5 Distribution of velocity magnitude and iso-velocity lines, as computed on the finest mesh (for color image refer Plate 11.5.8).

- Observe also that the level of entropy decreases when the mesh is refined. It would require finer grids to reduce this effect to an acceptable low level and we recommend that you further refine the grid by dividing the cell sizes by two in each direction, until you reduce the entropy to a sufficiently low level, reached when you recover a nearly symmetrical solution.
- This effect can be seen also on the iso-velocity plot of Figure 11.5.5, where the ‘numerical’ wake is clearly visible. The behavior of the downstream region mimics viscous flows, and is the direct consequence of the presence of the numerical dissipation.

This is a major requirement on accuracy. You should always attempt to refine the grids up to a level where the numerical dissipation is sufficiently low as to minimize the numerical entropy generation.

- Calculate the lift and drag components following equation (11.3.43). Both quantities should be zero and their non-zero values are a direct measure of the numerical errors of the simulation. Since the mesh is symmetrical in the present test case, the calculated lift should reflect the machine accuracy and only the drag coefficient is a measure of the numerical dissipation. The drag coefficient is defined by

$$C_D = \frac{L_x}{\frac{1}{2}\rho U^2 S} \quad \text{with } S = 2\pi a \quad (11.5.2)$$

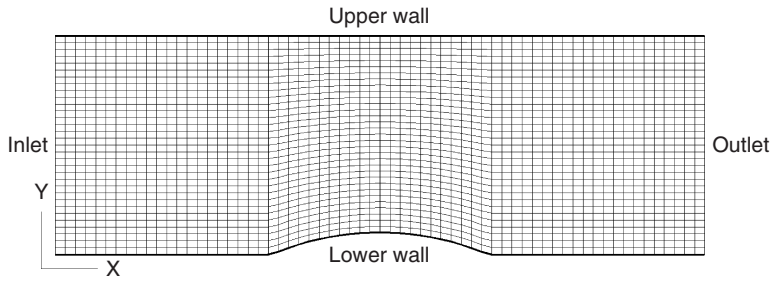
It should tend to zero, as the mesh is refined as seen from Table 11.5.1.

11.5.2 Application to the Internal Flow in a Channel with a Circular Bump

This is a representative example of an internal flow configuration. It consists of a channel of height L and length $3L$ with, along the bottom wall, a circular arc of length L and thickness equal to $0.1L$. We have no analytical exact solution for this case, but a reference solution can be obtained from a fine mesh simulation.

Table 11.5.1 Drag coefficient on the cylinder, computed on three different meshes.

	Drag coefficient
Exact	0
128×32	0.0094
65×17	0.0784
33×9	0.1867

**Figure 11.5.6** Mesh used for the channel flow test case.

Proceed now as follows:

- An H-grid can be constructed using 65 points in the axial direction with equal spacing and 33 points in-between walls. A quasi-uniform grid distribution is selected by taking $\Delta x = 3L/64$ and dividing each vertical mesh line in 32 uniform steps (see Figure 11.5.6).
- Assume atmospheric conditions, for pressure and temperature with $p_a = 101300$ Pa and $T_a = 288$ K, and $M = 0.1$ as inlet flow condition.
- In order to match these flow conditions, the total pressure and temperature are fixed at the inlet to 102010 Pa and 288.6 K, respectively, by following equations (11.1.9) and (11.1.12). The inlet velocity vector is imposed in the axial direction. The atmospheric pressure is also imposed at the outlet.
- At the solid walls, obtain the pressure and tangential velocity by taking them equal to the values of the corresponding cell centers.
- Apply the Runge–Kutta method (11.4.9) with the coefficients (11.4.11) and take $CFL = 2$. Select a local time step, based on equations (11.4.12).
- To start the calculation take a uniform initial solution equal to the values defined by the boundary conditions.
- Solve the Euler equations until a steady state is reached, with $CFL = 2$ and $\kappa^{(4)} = 1/256$.
- For internal flows, an important consistency check is the satisfaction of mass flow conservation, that is the mass flow rate through any section should be equal, up to an acceptably level of accuracy, to the inlet mass flow. The mass flow rate through a vertical mesh line i is equal to

$$\dot{m} = \int_{\text{section}} \rho \vec{v} \cdot d\vec{S} = \sum_j \left[\frac{(\rho u)_{i,j} + (\rho u)_{i,j+1}}{2} (y_{i,j+1} - y_{i,j}) \right] \quad (11.5.3)$$

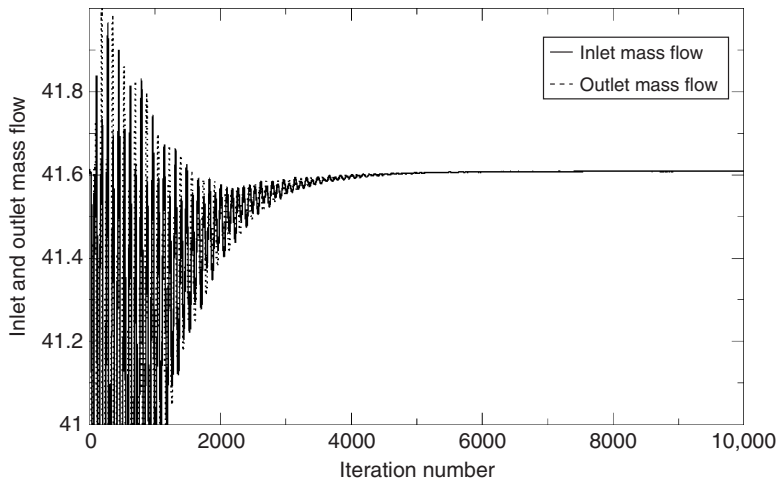


Figure 11.5.7 Convergence history of the inlet and outlet mass flow (kg/s/m).

referring to Figure 11.4.1, and applying a second order trapezium formula for the integration,

- Typically you should aim at mass flow errors between inlet and exit, below 0.1%. Note that this can only be achieved when the steady state is reached, since during the compressible transient the time derivative of density is not zero and hence the mass flow rate, as defined by equation (11.5.3), will not be constant.
- This is clearly seen on Figure 11.5.7, which shows the evolution of the inlet and exit mass flows (in kg/s/m) during the iterative process. You observe that after 5000 time steps, we have reached a satisfactory level of mass conservation, with an error below 0.01%.
- The Mach number isolines and color maps are shown on Figure 11.5.8 for the (65×33) mesh superimposed on the color map of a reference solution on a (225×113) mesh. One isoline has been drawn on the color map every 0.001 ranging from 0.07 to 0.13. The bottom figure compares the Mach number isolines of the two solutions, where the darkest line is the reference solution. You can observe here again the loss of symmetry between the upstream and downstream parts of the circular bump. The reference solution, obtained on a grid of 225×113 mesh points, has still a small deviation from symmetry, while the numerical solution on the selected grid shows a larger loss of symmetry in the wake region.
- If we compare the distribution of the wall surface pressure coefficient with the reference solution, shown on Figure 11.5.9, you observe that we cannot distinguish them. ***This indicates that the pressure is less sensitive to the numerical dissipation than entropy or drag.***
- Compare the solutions obtained at lower and higher Mach numbers (for instance 0.01, 0.05, 0.1 and 0.7) to investigate the effects of compressibility. This can be seen on Figure 11.5.10. You can observe that within the low Mach number range, the solution does not greatly depend on Mach number and is practically identical

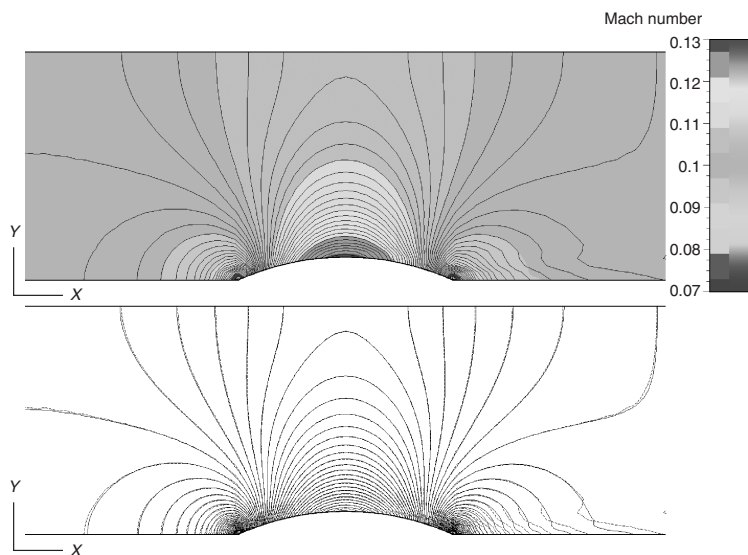


Figure 11.5.8 Isolines of Mach number as computed using the (65×33) mesh superimposed on the color map of a reference solution on a (225×113) mesh. One isoline has been drawn every 0.001 ranging from 0.07 to 0.13. The bottom figure compares the Mach number isolines of the two solutions, where the darkest line is the reference solution (for color image refer Plate 11.5.8).

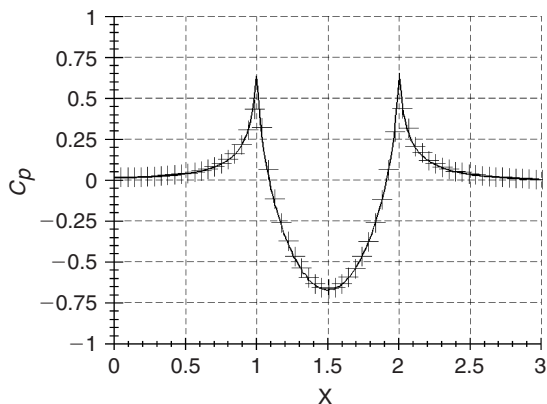


Figure 11.5.9 Axial distribution of pressure coefficient as computed using the (65×33) mesh (plus signs) and compared to the (225×113) mesh (continuous line).

to the incompressible flow. When increasing the Mach number, compressibility effects become more important and for an inlet Mach number slightly above 0.6, the sonic speed of $M = 1$ is reached on the bump surface, followed by a supersonic region terminated by a shock. This can be seen for the case of an incident velocity of $M = 0.7$.

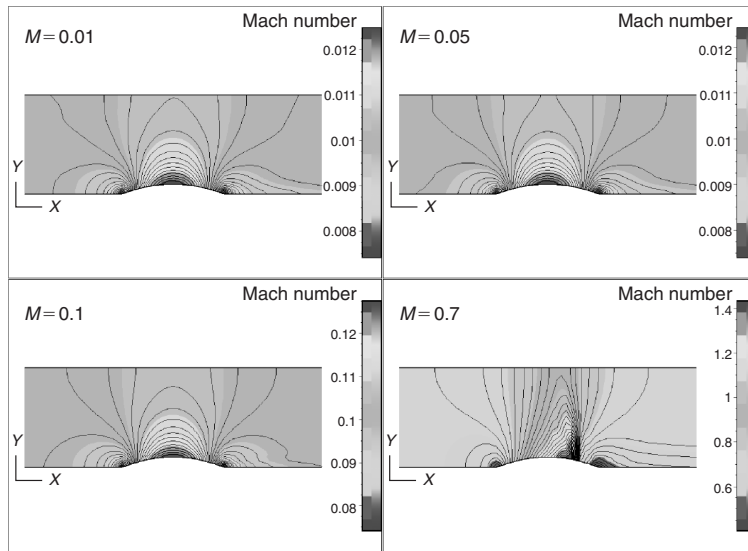


Figure 11.5.10 Distribution of velocity as computed on the (65×33) mesh, for different values of the incident Mach number, from 0.01 to 0.7. Observe the shock appearing at $M = 0.7$ (for color image refer Plate 11.5.10).

11.5.3 Application to the Supersonic Flow on a Wedge at $M = 2.5$

This example of a supersonic flow over a wedge of angle 15° generates an oblique shock. The Mach number upstream of the shock is fixed to 2.5. This test case has an exact analytical solution, satisfying the Rankine–Hugoniot relations (11.1.26); see for instance Anderson (1982), formed by two regions of constant states, separated by the oblique shock.

The downstream flow conditions are listed in Table 11.5.2.

Table 11.5.2 Analytical solution for the supersonic flow on a 15° wedge at $M = 2.5$.

Downstream Mach number (M_2)	1.87
Pressure ratio (p_2/p_1)	2.47
Entropy ratio (s_2/s_1)	1.03

This problem cannot be solved with the implementation of the Euler solver as applied for the two previous test cases. Indeed, in presence of shocks, we have to add an additional dissipation term to control the numerical oscillations appearing in presence of discontinuities, as shown for linear convection in Chapters 7 and 8. To this end, a second order artificial dissipation term is added to capture numerically strong gradients such as shock waves. Therefore, equation (11.4.6) is reformulated by adding a nonlinear dissipation term of second order, that reduces locally to first order at shock positions, in order to achieve a nearly monotone behavior, following

(Jameson et al. 1981):

$$D_{i+1/2,j} = \eta_{i+1/2,j}(U_{i+1,j} - U_{i,j}) + \gamma_{i+1/2,j}(U_{i+2,j} - 3U_{i+1,j} + 3U_{i,j} - U_{i-1,j}) \quad (11.5.4)$$

The coefficient η and γ are based on the following relations:

$$\begin{aligned} \eta_{i+1/2,j} &= \frac{1}{2}\kappa^{(2)}[\vec{v} \cdot \Delta \vec{S} + c|\Delta \vec{S}|]_{i+1/2,j} \max(v_{i-1}, v_i, v_{i+1}, v_{i+2}) \\ \gamma_{i+1/2,j} &= \max\left(0, \frac{1}{2}\kappa^{(4)}[\vec{v} \cdot \Delta \vec{S} + c|\Delta \vec{S}|]_{i+1/2,j} - \eta_{i+1/2,j}\right) \end{aligned} \quad (11.5.5)$$

where the variable v_i are sensors that activate the second order dissipation in regions of strong gradients. They are based on the pressure variations and are defined as

$$v_i = \left| \frac{p_{i+1,j} - 2p_{i,j} + p_{i-1,j}}{p_{i+1,j} + 2p_{i,j} + p_{i-1,j}} \right| \quad (11.5.6)$$

Observe that the numerator is proportional to the second derivative of the pressure and is a second order discretization of $\Delta x^2(\partial^2 p / \partial x^2)$ when the pressure variations are smooth. In the shock region, however, this term becomes close to 1 and the first term of equation (11.5.4) reduces to first order.

$\kappa^{(2)}$ is a non-dimensional coefficient of the order of unity.

You can notice that if the pressure variations are linear, the sensor v_i vanishes, the second order dissipation is not activated and the dissipation term is identical to the expression (11.4.6) defined for the subsonic applications.

With the addition of the second order dissipation to the inviscid numerical fluxes, we can proceed as follows:

- A uniform H-grid can be constructed using 97 points in the axial direction and 65 points in-between walls. The distribution of grid points is set to uniform along the x-direction and in-between walls (see Figure 11.5.11).
- In order to match the flow conditions, the static pressure and temperature are fixed at the inlet to 101353 Pa and 288.9 K, respectively. The inlet velocity vector is fixed to axial with a magnitude of 852.4 m/s.
- Since the outlet is supersonic, none of the characteristics enter the domain through this boundary. Therefore, all the variables are extrapolated at this boundary.
- Apply the Runge–Kutta method (11.4.9) with the coefficients (11.4.11) and take $CFL = 2$. Select here a local time step, based on equations (11.4.12).
- To start the calculation take a uniform initial solution equal to the values defined by the inlet boundary conditions.
- Solve the Euler equations until a steady state is reached, with $CFL = 2$, $\kappa^{(2)} = 1$ and $\kappa^{(4)} = 1/10$.
- Monitor the L_2 -norm of the axial momentum residual in function of number of time steps (see Figure 11.5.12). The residual has decreased by more than five orders of magnitude, which is in accordance with single precision machine

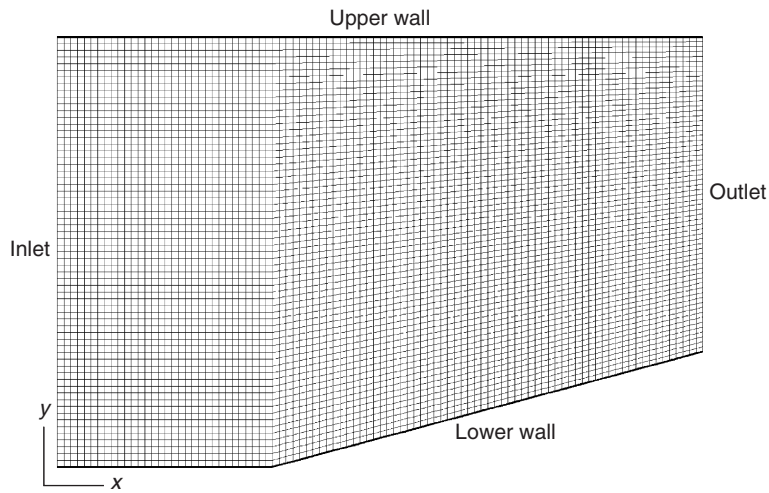


Figure 11.5.11 Mesh used for the wedge test case.

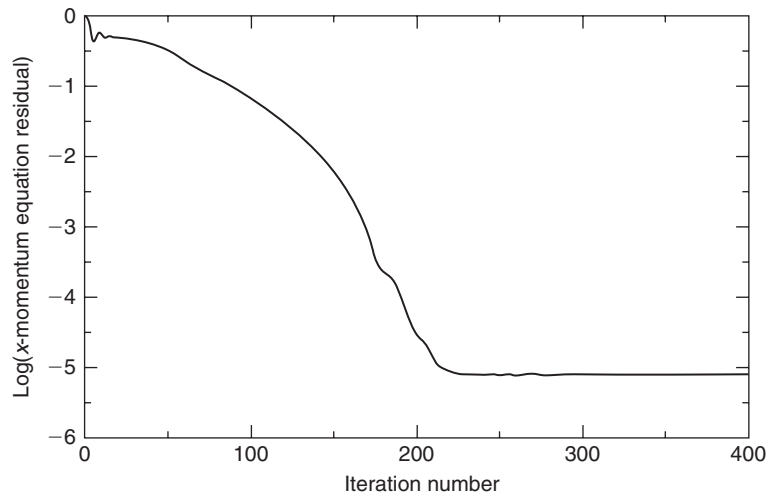


Figure 11.5.12 Convergence history of the normalized L_2 -norm of the x -momentum residual.

accuracy. You can reproduce the same calculation by using double precision machine accuracy and observe the drop in residual to much lower levels.

- If you compare this convergence behavior with the two previous test cases, you can notice how much faster the supersonic case converges, requiring only 150 iterations, compared to thousands for the subsonic cases. This is very typical and is largely connected to the boundary conditions and to the intrinsic properties of supersonic flows, which do not allow for upstream wave propagation of any quantity, including numerical errors.

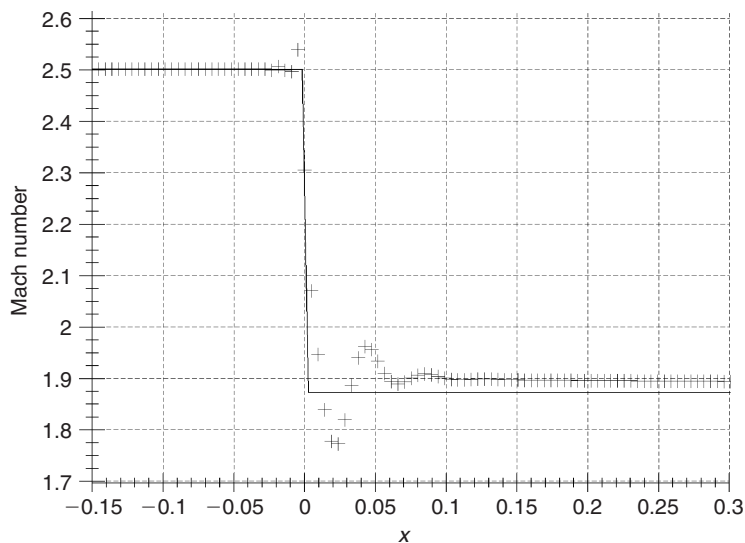


Figure 11.5.13 Axial distribution of Mach number along the bottom wall for the exact solution (continuous line) and from the numerical result (plus signs).

- Plot the Mach number variations along the bottom wall, and compare with the exact solution (see Figure 11.5.13). Upstream of the wall corner the solution is nearly perfect. Downstream of the shock, along the inclined wall, the numerical solution shows an oscillatory behavior of the Mach number. This illustrates the non-monotonic behavior of the second order central scheme with the dissipation term (11.5.4), which does not take into account the monotonicity requirements, discussed in Chapter 8.
- In addition, the first undershoot of the Mach number after the shock is due to the zero order interpolation of the pressure from the cell center to the wall, which is incorrect when the cell center is located upstream of the shock in the cells near the wedge corner. This illustrates the difficulties in imposing numerical boundary conditions, in presence of sharp local changes in geometry and flow conditions.
- Plot the iso-Mach numbers and iso-entropy lines, and observe the non-uniform behavior upstream and downstream of the shock wave, resulting from the non-monotone behavior of the scheme (see Figures 11.5.14 and 11.5.15). An important numerical error source is generated at the wedge corner, as a result of the local treatment of the wall boundary condition.
- Increase the mesh size in both directions and observe the progressive reduction of the errors.

This example clearly shows that, although a representative solution can be obtained for this simple supersonic flow, additional work has still to be done to eliminate the numerical oscillations, by imposing more severe monotone schemes with appropriate limiters, as discussed in Chapter 8.

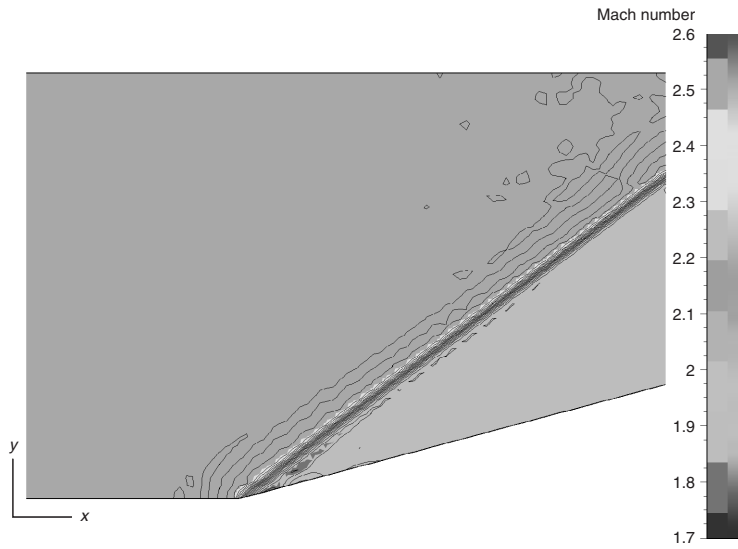


Figure 11.5.14 *Distribution of computed Mach number (for color image refer Plate 11.5.14).*

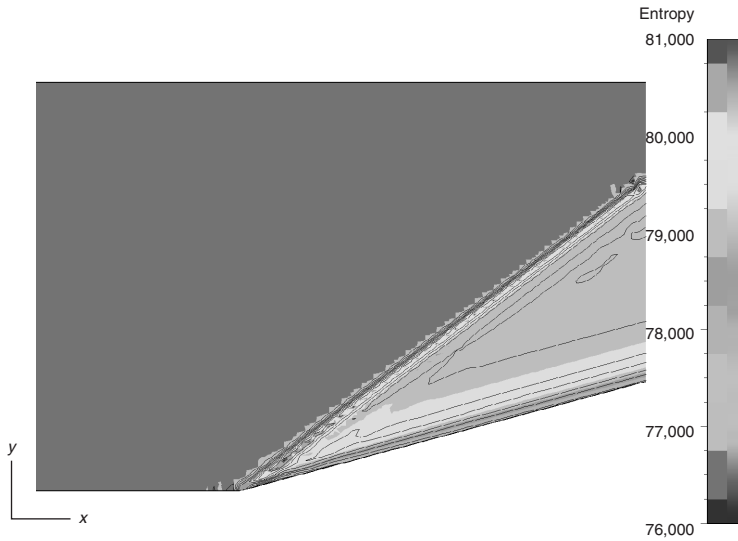


Figure 11.5.15 *Distribution of computed entropy (for color image refer Plate 11.5.15).*

11.5.4 Additional Hands-On Suggestions

With the developed programs, you can now extend the range of applications and experience with many other set-ups. In particular, you can investigate the effects of

many parameters, such as inflow conditions; grid densities and qualities, numerical algorithms. Some suggested actions are:

- Exercise with different grid types and grid densities to develop a ‘feeling’ and experience with ill-conditioned grids. In particular, introduce grid distortions and grid discontinuities close to the solid walls to observe their effects on the quality of the numerical solution.
- Look at the influence of the artificial dissipation coefficients $\kappa^{(4)}$ and $\kappa^{(2)}$ on convergence rate and calculated solution accuracy.
- Investigate the effects of compressibility by applying your potential or Euler code to the considered flow cases, with increasing incident Mach number. Observe what is happening to your code when shocks appear and in particular the non-monotone behavior over the shock with the Euler equations.
- Observe the sensitivity of the Mach number distribution downstream of the shock wave to the second and fourth order dissipation coefficients $\kappa^{(2)}$ and $\kappa^{(4)}$, for the wedge flow test case.
- Try other time integration methods, such as the Beam and Warming schemes introduced in Chapter 9.

Other options are left to your imagination and creativity.

CONCLUSIONS AND MAIN TOPICS TO REMEMBER

The main objective of this chapter is to guide you in writing your first CFD codes for inviscid potential and Euler flows on simple configurations, for which we have either an analytic solution or a reference numerical solution obtained on fine grids.

In running these test cases, we have insistently drawn your attention to the many error sources associated to the discretization of the inviscid Euler equations, at the level of implementation issues, boundary conditions as well as grid sensitivity.

On all the three test cases, you will have observed that the obtained results are not perfect and the errors shown are highly representative of ‘real life’ CFD simulations of industrial relevant systems. When you do not have a reference solution to compare with, the internal consistency checks, such as mass flow conservation, and the monitoring of entropy will provide you with a clear picture of the error levels of your simulation.

Main topics to remember are:

- Great care has to be exercised at all levels of the detailed implementation and programming choices.
- Give great attention to the implementation of boundary conditions.
- Perform as many ‘consistency’ checks as possible, by monitoring global conserved quantities, such as mass flow with internal flows, or total enthalpy. The variations of these quantities are indicators of local error sources.
- The issue of the entropy generated by the numerical dissipation is a most critical issue in CFD. Applying the Euler equation model is a unique method to identify the regions where this dissipation has a significant influence on the numerical solution. This should be your guide toward the required mesh refinement, in

order to bring this effect below an acceptable level. The same is true for the numerical calculated drag forces, which should be zero for inviscid flows.

- Grid quality and grid density have a major influence on the accuracy of the numerical simulation.

REFERENCES

- AIAA (1998). *Guide for the Verification and Validation of Computational Fluid Dynamics Simulations*. American Institute of Aeronautics and Astronautics, AIAA-G-077-1998, Reston, VA.
- Anderson, J.D. (1982). *Modern Compressible Flow*. McGraw Hill Inc., New York.
- ASME (2006). *V&V 10-2006 Guide for Verification and Validation in Computational Solid Mechanics*. ASME Publications; <http://catalog.asme.org/Codes/PrintBook>.
- Ballhaus, W.F. and Bailey, F.R. (1972). *Numerical Calculation of Transonic Flow about Swept Wings*. AIAA Paper 72-677.
- Chorin, A.J. (1967). A numerical method for solving incompressible viscous flow problems. *J. Computational Physics* 2, 12.
- Garabedian, P.R. and Korn, D. (1972). Analysis of transonic airfoils. *Commun. Pure Appl. Math* 24, 841–851.
- Hall, M.G. (1981). Computational Fluid Dynamics – A Revolutionary Force in Aerodynamics. AIAA Paper 81-1014 – Proc. Fifth AIAA Computational Fluid Dynamics Conference, pp.176–188.
- Holst, T. and Thomas, S. (1983). Numerical solution of transonic wing flow fields. *AIAA Journal* 21, 863–870.
- Jameson, A. (1974). Iterative solutions of transonic flows over airfoils and wings, including flows at Mach 1. *Commun. Pure Appl. Math*, 27, 283–309.
- Jameson, A. (1975). Transonic potential flow calculations using conservative form. *Proceedings of the AIAA Second Computational Fluid Dynamics Conference*, Hartford, pp.148–161.
- Jameson, A. and Caughey, D.E. (1977). A Finite Volume Method for Transonic Potential Flow Calculations. Proc. AIAA 3rd Computational Fluid Dynamics Conference, Albuquerque, pp. 35–54.
- Jameson, A., Schmidt, W. and Turkel, E. (1981). Numerical Simulation of the Euler Equations by Finite Volume Methods using Runge–Kutta Time Stepping Schemes. AIAA Paper 81-1259, AIAA 5th Computational Fluid Dynamics Conference.
- Lax, P.D. (1973). *Hyperbolic Systems of Conservation Laws and the Mathematical Theory of Shock Waves*, SIAM Publications, Philadelphia, USA.
- Merkle, C.L. and Choi, Y.-H. (1985). Computation of low speed compressible flows with time-marching methods. *Int. J. Numer. Method. Eng.*, 25, 293–311.
- Murman, E.M. and Cole, J.D. (1971). Calculation of plane steady transonic flows. *AIAA J.* 9, 114–121.
- Oberkampf, W.L., Trucano, T.G. and Hirsch, Ch. (2004). Verification, validation and predictive capability in computational engineering and physics. *Appl. Mech. Rev.*, 57(5), pp. 345–384.
- Shapiro, A.H. (1953). *The Dynamics and Thermodynamics of Compressible Fluid Flow*. Ronald Press, New York.
- Steger, J.L. and Lomax, H. (1972). Transonic flow about two-dimensional airfoils by relaxation procedures. *AIAA J.* 10, 49–54.
- Zucrow, M.J., Hoffman, J.D. (1976). *Gas Dynamics Vol. I & II*. J. Wiley & Sons, New York.