

A Computational Fluid Dynamics

A.1 Introduction

Numerical methods using digital computers are, of course, commonly utilized to solve a wide variety of flow problems. As discussed in Chapter 6, although the differential equations that govern the flow of Newtonian fluids [the Navier–Stokes equations (Eq. 6.127)] were derived many years ago, there are few known analytical solutions to them. However, with the advent of high-speed digital computers it has become possible to obtain approximate numerical solutions to these (and other fluid mechanics) equations for a wide variety of circumstances.

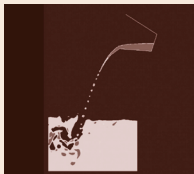
Computational fluid dynamics (CFD) involves replacing the partial differential equations with discretized algebraic equations that approximate the partial differential equations. These equations are then numerically solved to obtain flow field values at the discrete points in space and/or time. Since the Navier–Stokes equations are valid everywhere in the flow field of the fluid continuum, an analytical solution to these equations provides the solution for an infinite number of points in the flow. However, analytical solutions are available for only a limited number of simplified flow geometries. To overcome this limitation, the governing equations can be discretized and put in algebraic form for the computer to solve. The CFD simulation solves for the relevant flow variables only at the discrete points, which make up the grid or mesh of the solution (discussed in more detail below). Interpolation schemes are used to obtain values at non-grid point locations.

CFD can be thought of as a numerical experiment. In a typical fluids experiment, an experimental model is built, measurements of the flow interacting with that model are taken, and the results are analyzed. In CFD, the building of the model is replaced with the formulation of the governing equations and the development of the numerical algorithm. The process of obtaining measurements is replaced with running an algorithm on the computer to simulate the flow interaction. Of course, the analysis of results is common ground to both techniques.

CFD can be classified as a subdiscipline to the study of fluid dynamics. However, it should be pointed out that a thorough coverage of CFD topics is well beyond the scope of this textbook. This appendix highlights some of the more important topics in CFD, but is only intended as a brief introduction. The topics include discretization of the governing equations, grid generation, boundary conditions, application of CFD, and some representative examples.



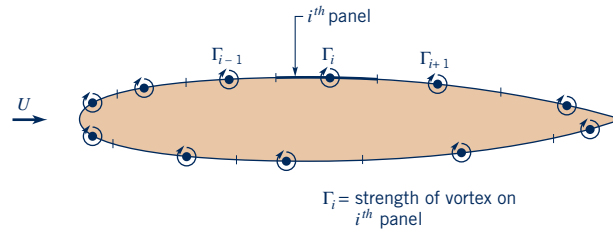
VA.1 Pouring
a liquid



A.2 Discretization

The process of *discretization* involves developing a set of algebraic equations (based on discrete points in the flow domain) to be used in place of the partial differential equations. Of the various discretization techniques available for the numerical solution of the governing differential equations, the following three types are most common: (1) the finite difference method, (2) the finite element (or finite volume) method, and (3) the boundary element method. In each of these methods, the continuous flow field (i.e., velocity or pressure as a function of space and time) is described in terms of discrete (rather than continuous) values at prescribed locations. Through this technique the differential equations are replaced by a set of algebraic equations that can be solved on the computer.

For the *finite element* (or *finite volume*) method, the flow field is broken into a set of small fluid elements (usually triangular areas if the flow is two-dimensional, or small volume elements if the flow is three-dimensional). The conservation equations (i.e., conservation of mass, momentum, and energy) are written in an appropriate form for each element, and the set of resulting



■ **Figure A.1** Panel method for flow past an airfoil.

algebraic equations for the flow field is solved numerically. The number, size, and shape of elements are dictated in part by the particular flow geometry and flow conditions for the problem at hand. As the number of elements increases (as is necessary for flows with complex boundaries), the number of simultaneous algebraic equations that must be solved increases rapidly. Problems involving one million to ten million (or more) grid cells are not uncommon in today's CFD community, particularly for complex three-dimensional geometries. Further information about this method can be found in Refs. 1 and 2.

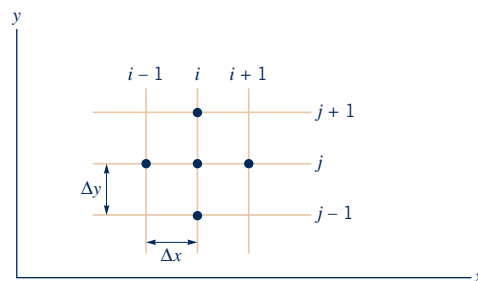
For the *boundary element method*, the boundary of the flow field (not the entire flow field as in the finite element method) is broken into discrete segments (Ref. 3) and appropriate singularities such as sources, sinks, doublets, and vortices are distributed on these boundary elements. The strengths and type of the singularities are chosen so that the appropriate boundary conditions of the flow are obtained on the boundary elements. For points in the flow field not on the boundary, the flow is calculated by adding the contributions from the various singularities on the boundary. Although the details of this method are rather mathematically sophisticated, it may (depending on the particular problem) require less computational time and space than the finite element method. Typical boundary elements and their associated singularities (vortices) for two-dimensional flow past an airfoil are shown in Fig. A.1. Such use of the boundary element method in aerodynamics is often termed the *panel method* in recognition of the fact that each element plays the role of a panel on the airfoil surface (Ref. 4).

The *finite difference method* for computational fluid dynamics is perhaps the most easily understood of the three methods listed above. For this method the flow field is dissected into a set of grid points and the continuous functions (velocity, pressure, etc.) are approximated by discrete values of these functions calculated at the grid points. Derivatives of the functions are approximated by using the differences between the function values at local grid points divided by the grid spacing. The standard method for converting the partial differential equations to algebraic equations is through the use of Taylor series expansions. (See Ref. 5.) For example, assume a standard rectangular grid is applied to a flow domain as shown in Fig. A.2.

This grid stencil shows five grid points in x - y space with the center point being labeled as i, j . This index notation is used as subscripts on variables to signify location. For example, $u_{i+1,j}$ is the u component of velocity at the first point to the right of the center point i, j . The grid spacing in the i and j directions is given as Δx and Δy , respectively.

To find an algebraic approximation to a first derivative term such as $\partial u / \partial x$ at the i, j grid point, consider a Taylor series expansion written for u at $i + 1$ as

$$u_{i+1,j} = u_{i,j} + \left(\frac{\partial u}{\partial x} \right)_{i,j} \frac{\Delta x}{1!} + \left(\frac{\partial^2 u}{\partial x^2} \right)_{i,j} \frac{(\Delta x)^2}{2!} + \left(\frac{\partial^3 u}{\partial x^3} \right)_{i,j} \frac{(\Delta x)^3}{3!} + \dots \quad (\text{A.1})$$



■ **Figure A.2** Standard rectangular grid.

Solving for the underlined term in the above equation results in the following:

$$\left(\frac{\partial u}{\partial x}\right)_{i,j} = \frac{u_{i+1,j} - u_{i,j}}{\Delta x} + O(\Delta x) \quad (\text{A.2})$$

where $O(\Delta x)$ contains higher order terms proportional to Δx , $(\Delta x)^2$, and so forth. Equation A.2 represents a forward difference equation to approximate the first derivative using values at $i + 1, j$ and i, j along with the grid spacing in the x direction. Obviously in solving for the $\partial u/\partial x$ term we have ignored higher order terms such as the second and third derivatives present in Eq. A.1. This process is termed *truncation* of the Taylor series expansion. The lowest order term that was truncated included $(\Delta x)^2$. Notice that the first derivative term contains Δx . When solving for the first derivative, all terms on the right-hand side were divided by Δx . Therefore, the term $O(\Delta x)$ signifies that this equation has error of “order (Δx) ,” which is due to the neglected terms in the Taylor series and is called truncation error. Hence, the forward difference is termed first-order accurate.

Thus, we can transform a partial derivative into an algebraic expression involving values of the variable at neighboring grid points. This method of using the Taylor series expansions to obtain discrete algebraic equations is called the finite difference method. Similar procedures can be used to develop approximations termed backward difference and central difference representations of the first derivative. The central difference makes use of both the left and right points (i.e., $i - 1, j$ and $i + 1, j$) and is second-order accurate. In addition, finite difference equations can be developed for the other spatial directions (i.e., $\partial u/\partial y$) as well as for second derivatives ($\partial^2 u/\partial x^2$), which are also contained in the Navier–Stokes equations (see Ref. 5 for details).

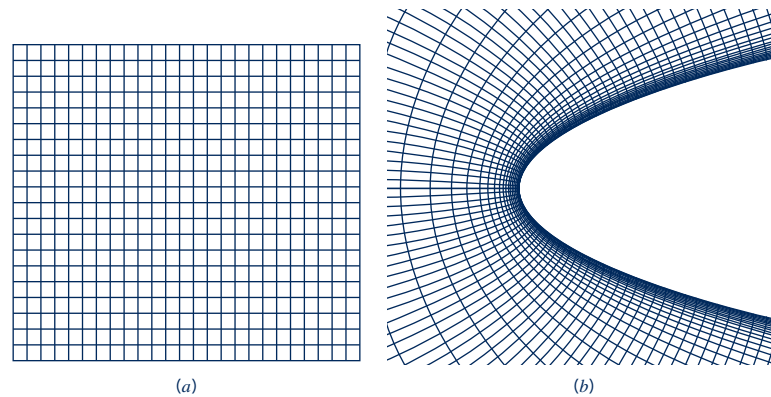
Applying this method to all terms in the governing equations transfers the differential equations into a set of algebraic equations involving the physical variables at the grid points (i.e., $u_{i,j}, p_{i,j}$ for $i = 1, 2, 3, \dots$ and $j = 1, 2, 3, \dots$, etc.). This set of equations is then solved by appropriate numerical techniques. The larger the number of grid points used, the larger the number of equations that must be solved.

A student of CFD should realize that the discretization of the continuum governing equations involves the use of algebraic equations that are an approximation to the original partial differential equation. Along with this approximation comes some amount of error. This type of error is termed truncation error because the Taylor series expansion used to represent a derivative is “truncated” at some reasonable point and the higher order terms are ignored. The truncation errors tend to zero as the grid is refined by making Δx and Δy smaller, so grid refinement is one method of reducing this type of error. Another type of unavoidable numerical error is the so-called round-off error. This type of error is due to the limit of the computer on the number of digits it can retain in memory. Engineering students can run into round-off errors from their calculators if they plug values into the equations at an early stage of the solution process. Fortunately, for most CFD cases, if the algorithm is set up properly, round-off errors are usually negligible.

A.3 Grids

CFD computations using the finite difference method provide the flow field at discrete points in the flow domain. The arrangement of these discrete points is termed the *grid* or the *mesh*. The type of grid developed for a given problem can have a significant impact on the numerical simulation, including the accuracy of the solution. The grid must represent the geometry correctly and accurately, since an error in this representation can have a significant effect on the solution.

The grid must also have sufficient grid resolution to capture the relevant flow physics, otherwise they will be lost. This particular requirement is problem dependent. For example, if a flow field has small-scale structures, the grid resolution must be sufficient to capture these structures. It is usually necessary to increase the number of grid points (i.e., use a finer mesh) where large gradients are to be expected, such as in the boundary layer near a solid surface. The same can also be



■ **Figure A.3** Structured grids. (a) Rectangular grid. (b) Grid around a parabolic surface.

said for the temporal resolution. The time step, Δt , used for unsteady flows must be smaller than the smallest time scale of the flow features being investigated.

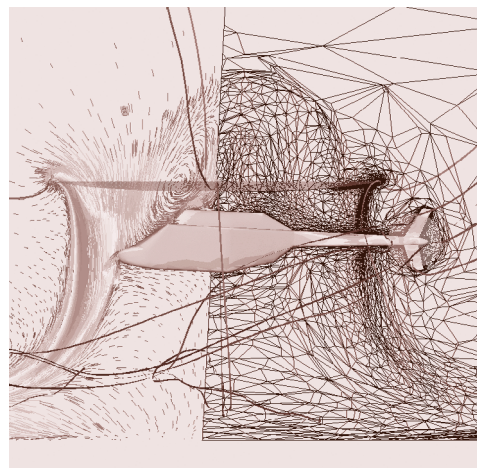
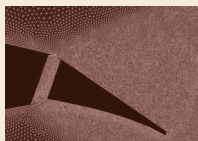
Generally, the types of grids fall into two categories: structured and unstructured, depending on whether or not there exists a systematic pattern of connectivity of the grid points with their neighbors. As the name implies, a *structured grid* has some type of regular, coherent structure to the mesh layout that can be defined mathematically. The simplest structured grid is a uniform rectangular grid, as shown in Fig. A.3a. However, structured grids are not restricted to rectangular geometries. Figure A.3b shows a structured grid wrapped around a parabolic surface. Notice that grid points are clustered near the surface (i.e., grid spacing in normal direction increases as one moves away from the surface) to help capture the steep flow gradients found in the boundary layer region. This type of variable grid spacing is used wherever there is a need to increase grid resolution and is termed *grid stretching*.

For the *unstructured grid*, the grid cell arrangement is irregular and has no systematic pattern. The grid cell geometry usually consists of various-sized triangles for two-dimensional problems and tetrahedrals for three-dimensional grids. An example of an unstructured grid is shown in Fig. A.4. Unlike structured grids, for an unstructured grid each grid cell and the connection information to neighboring cells is defined separately. This produces an increase in the computer code complexity as well as a significant computer storage requirement. The advantage to an unstructured grid is that it can be applied to complex geometries, where structured grids would have severe difficulty. The finite difference method is usually restricted to structured grids whereas the finite volume (or finite element) method can use either structured or unstructured grids.

Other grids include hybrid, moving, and adaptive grids. A grid that uses a combination of grid elements (rectangles, triangles, etc.) is termed a *hybrid grid*. As the name implies, the *moving grid*



VA.2 Dynamic grid



■ **Figure A.4** Anisotropic adaptive mesh for flow induced by the rotor of a helicopter in hover, with ground effect. Left: flow; Right: grid. (From Newmerical Technologies International, Montreal, Canada. Used by permission.)

is helpful for flows involving a time-dependent geometry. If, for example, the problem involves simulating the flow within a pumping heart or the flow around a flapping wing, a mesh that moves with the geometry is desired. The nature of the adaptive grid lies in its ability to literally adapt itself during the simulation. For this type of grid, while the CFD code is trying to reach a converged solution, the grid will adapt itself to place additional grid resources in regions of high-flow gradients. Such a grid is particularly useful when a new problem arises and the user is not quite sure where to refine the grid due to high-flow gradients.

A.4 Boundary Conditions

The same governing equations, the Navier–Stokes equations (Eq. 6.127), are valid for all incompressible Newtonian fluid flow problems. Thus, if the same equations are solved for all types of problems, how is it possible to achieve different solutions for different types of flows involving different flow geometries? The answer lies in the *boundary conditions* of the problem. The boundary conditions are what allow the governing equations to differentiate between different flow fields (for example, flow past an automobile and flow past a person running) and produce a solution unique to the given flow geometry.

It is critical to specify the correct boundary conditions so that the CFD simulation is a well-posed problem and is an accurate representation of the physical problem. Poorly defined boundary conditions can ultimately affect the accuracy of the solution. One of the most common boundary conditions used for simulation of viscous flow is the no-slip condition, as discussed in Section 1.6. Thus, for example, for two-dimensional external or internal flows, the x and y components of velocity (u and v) are set to zero at the stationary wall to satisfy the no-slip condition. Other boundary conditions that must be appropriately specified involve inlets, outlets, far-field, wall gradients, etc. It is important to not only select the correct physical boundary condition for the problem, but also to correctly implement this boundary condition into the numerical simulation.

A.5 Basic Representative Examples

A very simple one-dimensional example of the finite difference technique is presented in the following example.

EXAMPLE A.1 Flow from a Tank

A viscous oil flows from a large, open tank and through a long, small-diameter pipe as shown in Fig. EA.1a. At time $t = 0$ the fluid depth is H . Use a finite difference technique to determine the

liquid depth as a function of time, $h = h(t)$. Compare this result with the exact solution of the governing equation.

SOLUTION

Although this is an unsteady flow (i.e., the deeper the oil, the faster it flows from the tank) we assume that the flow is “quasisteady” and apply steady flow equations as follows.

As shown by Eq. 6.152, the mean velocity, V , for steady laminar flow in a round pipe of diameter D is given by

$$V = \frac{D^2 \Delta p}{32 \mu \ell} \quad (1)$$

where Δp is the pressure drop over the length ℓ . For this problem the pressure at the bottom of the tank (the inlet of the pipe) is γh and that at the pipe exit is zero. Hence, $\Delta p = \gamma h$ and Eq. 1 becomes

$$V = \frac{D^2 \gamma h}{32 \mu \ell} \quad (2)$$

Conservation of mass requires that the flowrate from the tank, $Q = \pi D^2 V / 4$, is related to the rate of change of depth of oil in the tank, dh/dt , by

$$Q = -\frac{\pi}{4} D_T^2 \frac{dh}{dt}$$

where D_T is the tank diameter. Thus,

$$\frac{\pi}{4} D^2 V = -\frac{\pi}{4} D_T^2 \frac{dh}{dt}$$

or

$$V = -\left(\frac{D_T}{D}\right)^2 \frac{dh}{dt} \quad (3)$$

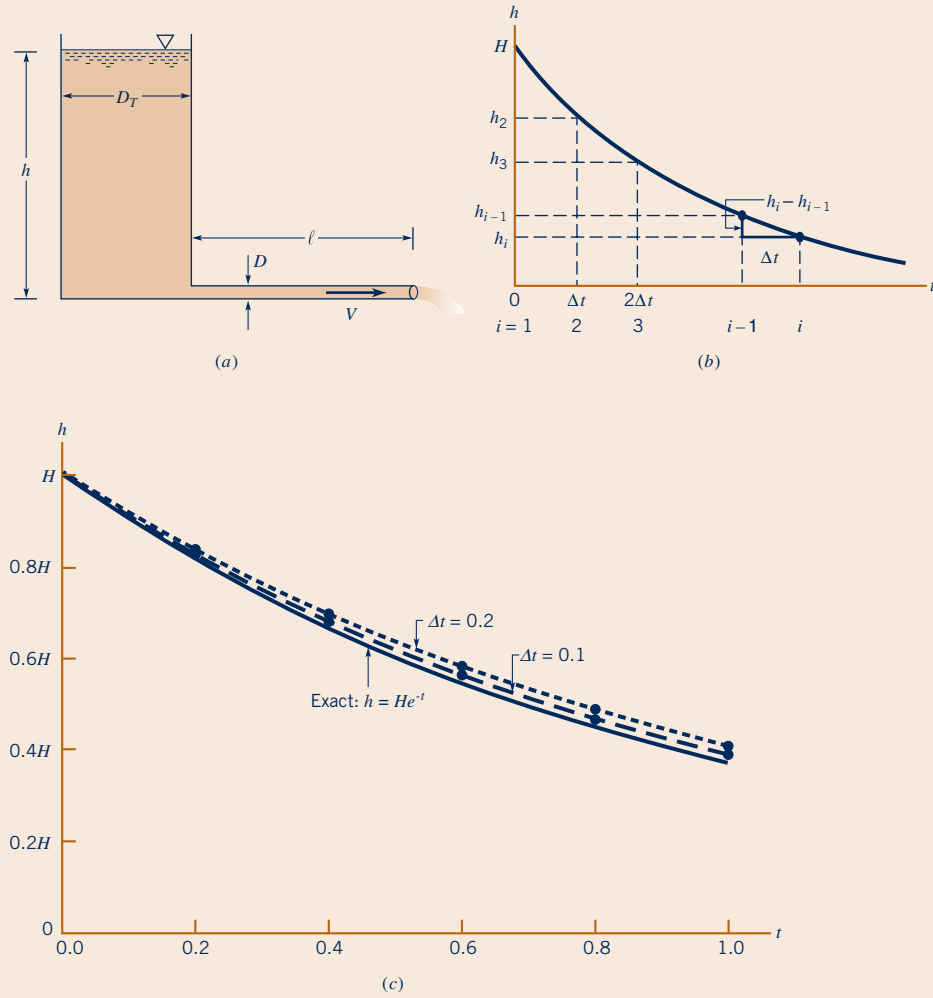


Figure EA.1

By combining Eqs. 2 and 3 we obtain

$$\frac{D^2 \gamma h}{32 \mu \ell} = - \left(\frac{D_T}{D} \right)^2 \frac{dh}{dt}$$

or

$$\frac{dh}{dt} = -Ch$$

where $C = \gamma D^4 / 32 \mu \ell D_T^2$ is a constant. For simplicity we assume the conditions are such that $C = 1$. Thus, we must solve

$$\frac{dh}{dt} = -h \quad \text{with} \quad h = H \text{ at } t = 0 \quad (4)$$

The exact solution to Eq. 4 is obtained by separating the variables and integrating to obtain

$$h = He^{-t} \quad (5)$$

However, assume this solution was not known. The following finite difference technique can be used to obtain an approximate solution.

As shown in Fig. EA.1b, we select discrete points (nodes or grid points) in time and approximate the time derivative of h by the expression

$$\left. \frac{dh}{dt} \right|_{t=t_i} \approx \frac{h_i - h_{i-1}}{\Delta t} \quad (6)$$

where Δt is the time step between the different node points on the time axis and h_i and h_{i-1} are the approximate values of h at nodes i and $i - 1$. Equation 6 is called the backward-difference approximation to dh/dt . We are free to select whatever value of Δt that we wish. (Although we do not need to space the nodes at equal distances, it is often convenient to do so.) Since the governing equation (Eq. 4) is an ordinary differential equation, the “grid” for the finite difference method is a one-dimensional grid as shown in Fig. EA.1b rather than a two-dimensional grid (which occurs for partial differential equations) as shown in Fig. EA.2b, or a three-dimensional grid.

Thus, for each value of $i = 2, 3, 4, \dots$ we can approximate the governing equation, Eq. 4, as

$$\frac{h_i - h_{i-1}}{\Delta t} = -h_i$$

or

$$h_i = \frac{h_{i-1}}{(1 + \Delta t)} \quad (7)$$

We cannot use Eq. 7 for $i = 1$ since it would involve the non-existing h_0 . Rather we use the initial condition (Eq. 4), which gives

$$h_1 = H$$

The result is the following set of N algebraic equations for the N approximate values of h at times $t_1 = 0, t_2 = \Delta t, \dots, t_N = (N - 1)\Delta t$.

$$h_1 = H$$

$$h_2 = h_1/(1 + \Delta t)$$

$$h_3 = h_2/(1 + \Delta t)$$

$$\vdots$$

$$h_N = h_{N-1}/(1 + \Delta t)$$

For most problems the corresponding equations would be more complicated than those just given, and a computer would be used to solve for the h_i . For this problem the solution is simply

$$h_2 = H/(1 + \Delta t)$$

$$h_3 = H/(1 + \Delta t)^2$$

$$\vdots$$

or in general

$$h_i = H/(1 + \Delta t)^{i-1}$$

The results for $0 < t < 1$ are shown in Fig. EA.1c. Tabulated values of the depth for $t = 1$ are listed in the table below.

Δt	i for $t = 1$	h_i for $t = 1$
0.2	6	0.4019H
0.1	11	0.3855H
0.01	101	0.3697H
0.001	1001	0.3681H
Exact (Eq. 5)	—	0.3678H

It is seen that the approximate results compare quite favorably with the exact solution given by Eq. 5. It is expected that the finite difference results would more closely approximate the exact results as Δt is decreased since in the limit of $\Delta t \rightarrow 0$ the finite difference approximation for the derivatives (Eq. 6) approaches the actual definition of the derivative.

For most CFD problems the governing equations to be solved are partial differential equations [rather than an ordinary differential equation as in the above example (Eq. A.1)] and the finite difference method becomes considerably more involved. The following example illustrates some of the concepts involved.

EXAMPLE A.2 Flow Past a Cylinder

Consider steady, incompressible flow of an inviscid fluid past a circular cylinder as shown in Fig. EA.2a. The stream function, ψ , for this flow is governed by the Laplace equation (see Section 6.5)

$$\frac{\partial^2 \psi}{\partial x^2} + \frac{\partial^2 \psi}{\partial y^2} = 0 \quad (1)$$

The exact analytical solution is given in Section 6.6.3.

Describe a simple finite difference technique that can be used to solve this problem.

SOLUTION

The first step is to define a flow domain and set up an appropriate grid for the finite difference scheme. Since we expect the flow field to be symmetrical both above and below and in front of and behind the cylinder, we consider only one-quarter of the entire flow domain as indicated in Fig. EA.2b. We locate the upper boundary and right-hand boundary far enough from the cylinder so that we expect the flow to be essentially uniform at these locations. It is not always clear how far from the object these boundaries must be located. If they are not far enough, the solution obtained will be incorrect because we have imposed

artificial, uniform flow conditions at a location where the actual flow is not uniform. If these boundaries are farther than necessary from the object, the flow domain will be larger than necessary and excessive computer time and storage will be required. Experience in solving such problems is invaluable!

Once the flow domain has been selected, an appropriate grid is imposed on this domain (see Fig. EA.2b). Various grid structures can be used. If the grid is too coarse, the numerical solution may not be capable of capturing the fine scale structure of the actual flow field. If the grid is too fine, excessive computer time and

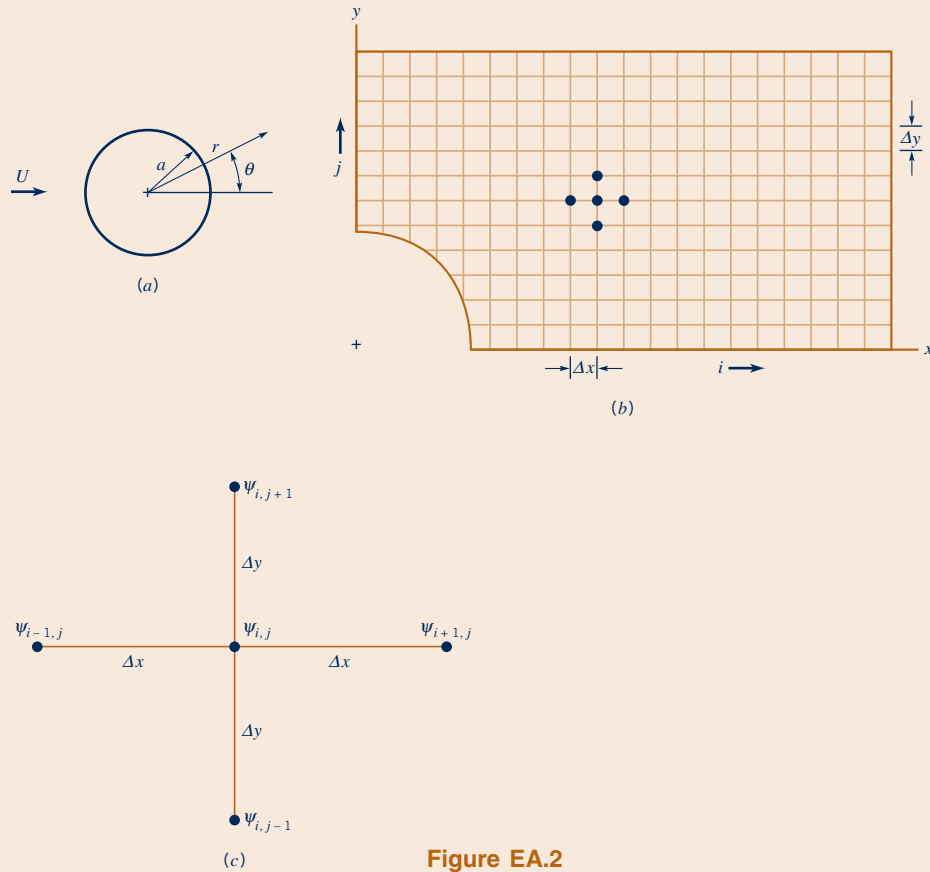


Figure EA.2

storage may be required. Considerable work has gone into forming appropriate grids (Ref. 6). We consider a grid that is uniformly spaced in the x and y directions, as shown in Fig. EA.2b.

As shown in Eq. 6.112, the exact solution to Eq. 1 (in terms of polar coordinates r, θ rather than Cartesian coordinates x, y) is $\psi = Ur(1 - a^2/r^2) \sin \theta$. The finite difference solution approximates these stream function values at a discrete (finite) number of locations (the grid points) as $\psi_{i,j}$, where the i and j indices refer to the corresponding x_i and y_j locations.

The derivatives of ψ can be approximated as follows:

$$\frac{\partial \psi}{\partial x} \approx \frac{1}{\Delta x} (\psi_{i+1,j} - \psi_{i,j})$$

and

$$\frac{\partial \psi}{\partial y} \approx \frac{1}{\Delta y} (\psi_{i,j+1} - \psi_{i,j})$$

This particular approximation is called a forward-difference approximation. Other approximations are possible. By similar reasoning, it is possible to show that the second derivatives of ψ can be written as follows:

$$\frac{\partial^2 \psi}{\partial x^2} \approx \frac{1}{(\Delta x)^2} (\psi_{i+1,j} - 2\psi_{i,j} + \psi_{i-1,j}) \quad (2)$$

and

$$\frac{\partial^2 \psi}{\partial y^2} \approx \frac{1}{(\Delta y)^2} (\psi_{i,j+1} - 2\psi_{i,j} + \psi_{i,j-1}) \quad (3)$$

Thus, by combining Eqs. 1, 2, and 3 we obtain

$$\begin{aligned} \frac{\partial^2 \psi}{\partial x^2} + \frac{\partial^2 \psi}{\partial y^2} &\approx \frac{1}{(\Delta x)^2} (\psi_{i+1,j} + \psi_{i-1,j}) + \frac{1}{(\Delta y)^2} (\psi_{i,j+1} \\ &+ \psi_{i,j-1}) - 2 \left(\frac{1}{(\Delta x)^2} + \frac{1}{(\Delta y)^2} \right) \psi_{i,j} = 0 \end{aligned} \quad (4)$$

Equation 4 can be solved for the stream function at x_i and y_j to give

$$\begin{aligned} \psi_{i,j} &= \frac{1}{2[(\Delta x)^2 + (\Delta y)^2]} [(\Delta y)^2 (\psi_{i+1,j} + \psi_{i-1,j}) \\ &+ (\Delta x)^2 (\psi_{i,j+1} + \psi_{i,j-1})] \end{aligned} \quad (5)$$

Note that the value of $\psi_{i,j}$ depends on the values of the stream function at neighboring grid points on either side and above and below the point of interest (see Eq. 5 and Fig. EA. 2c).

To solve the problem (either exactly or by the finite difference technique), it is necessary to specify boundary conditions for points located on the boundary of the flow domain (see Section 6.6.3). For example, we may specify that $\psi = 0$ on the lower boundary of the domain (see Fig. EA.2b) and $\psi = C$, a constant, on the upper boundary of the domain. Appropriate boundary conditions on the two vertical ends of the flow domain can also be specified. Thus, for points interior to the boundary Eq. 5 is valid; similar equations or specified values of $\psi_{i,j}$ are valid for boundary points. The result is an equal number of equations and unknowns, $\psi_{i,j}$, one for every grid point. For this problem, these equations represent a set of linear algebraic equations for $\psi_{i,j}$, the solution

of which provides the finite difference approximation for the stream function at discrete grid points in the flow field. Streamlines (lines of constant ψ) can be obtained by interpolating values of $\psi_{i,j}$ between the grid points and “connecting the dots” of $\psi = \text{constant}$. The velocity field can be obtained from the derivatives of the stream function according to Eq. 6.74. That is,

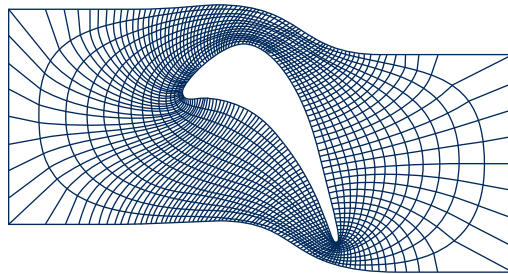
$$u = \frac{\partial \psi}{\partial y} \approx \frac{1}{\Delta y} (\psi_{i,j+1} - \psi_{i,j})$$

and

$$v = -\frac{\partial \psi}{\partial x} \approx -\frac{1}{\Delta x} (\psi_{i+1,j} - \psi_{i,j})$$

Further details of the finite difference technique can be found in standard references on the topic (Refs. 5, 7, 8). Also, see the completely solved viscous flow CFD problem in Appendix I.

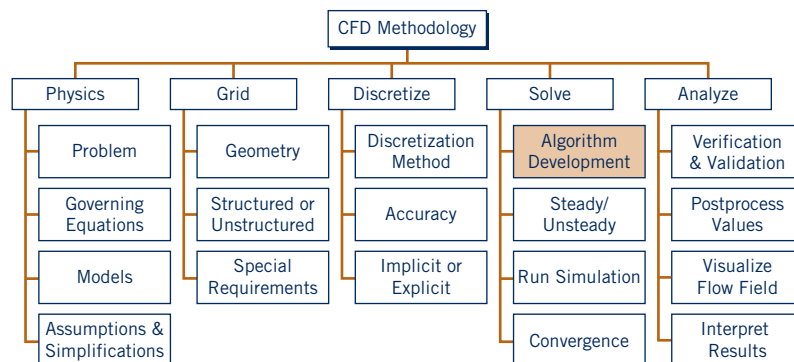
The preceding two examples are rather simple because the governing equations are not too complex. A finite difference solution of the more complicated, nonlinear Navier–Stokes equation (Eq. 6.127) requires considerably more effort and insight and larger and faster computers. A typical finite difference grid for a more complex flow, the flow past a turbine blade, is shown in Fig. A.5. Note that the mesh is much finer in regions where large gradients are to be expected (i.e., near the leading and trailing edges of the blade) and coarser away from the blade.



■ **Figure A.5** Finite difference grid for flow past a turbine blade. (From Ref. 9, used by permission.)

A.6 Methodology

In general, most applications of CFD take the same basic approach. Some of the differences include problem complexity, available computer resources, available expertise in CFD, and whether a commercially available CFD package is used, or a problem-specific CFD algorithm is developed. In today's market, there are many commercial CFD codes available to solve a wide variety of problems. However, if the intent is to conduct a thorough investigation of a specific fluid flow problem such as in a research environment, it is possible that taking the time to develop a problem-specific algorithm may be most efficient in the long run. The features common to most CFD applications can be summarized in the flow chart shown in Fig. A.6. A complete, detailed CFD solution for a viscous flow obtained by using the steps summarized in the flow chart can be accessed from the book's web site at www.wiley.com/college/munson.



■ **Figure A.6** Flow chart of general CFD methodology.

The Algorithm Development box is grayed because this step is required only when developing your own CFD code. When using a commercial CFD code, this step is not necessary. This chart represents a generalized methodology to CFD. There are other more complex components that are hidden in the above steps, which are beyond the scope of a brief introduction to CFD.

A.7 Application of CFD

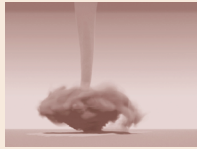
In the early stages of CFD, research and development was primarily driven by the aerospace industry. Today, CFD is still used as a research tool, but it also has found a place in industry as a design tool. There is now a wide variety of industries that make at least some use of CFD, including automotive, industrial, HVAC, naval, civil, chemical, biological, and others. Industries are using CFD as an added engineering tool that complements the experimental and theoretical work in fluid dynamics.

A.7.1 Advantages of CFD

There are many advantages to using CFD for simulation of fluid flow. One of the most important advantages is the realizable savings in time and cost for engineering design. In the past, coming up with a new engineering design meant somewhat of a trial-and-error method of building and testing multiple prototypes prior to finalizing the design. With CFD, many of the issues dealing with fluid flow can be flushed out prior to building the actual prototype. This translates to a significant savings in time and cost. It should be noted that CFD is not meant to replace experimental testing, but rather to work in conjunction with it. Experimental testing will always be a necessary component of engineering design. Other advantages include the ability of CFD to: (1) obtain flow information in regions that would be difficult to test experimentally, (2) simulate real flow conditions, (3) conduct large parametric tests on new designs in a shorter time, and (4) enhance visualization of complex flow phenomena.



VA.3 Tornado simulation



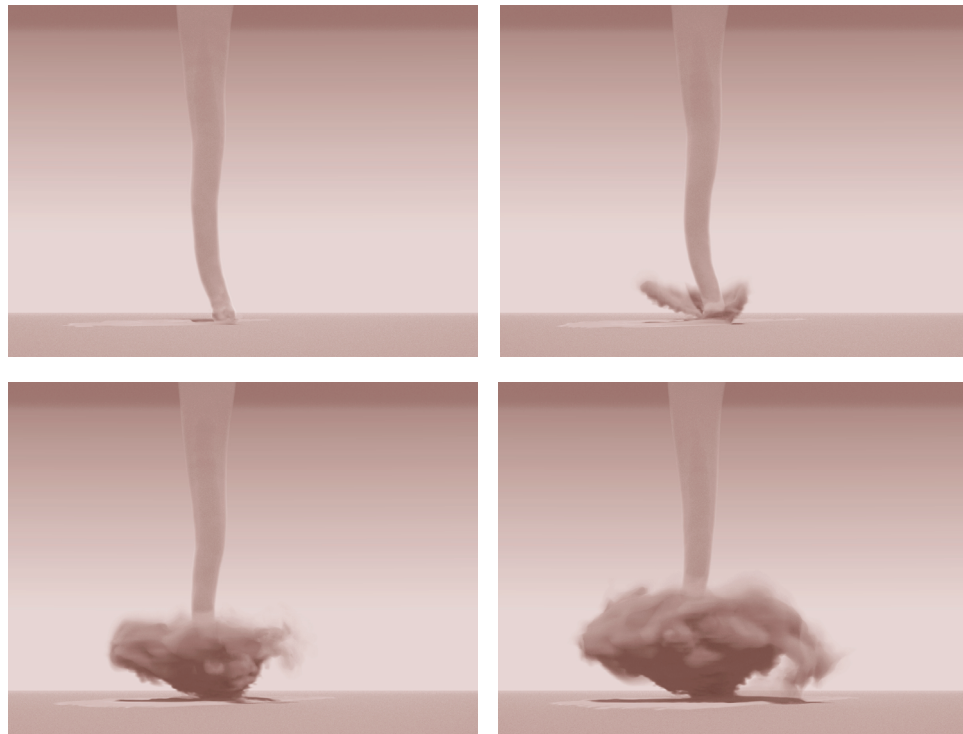
A good example of the advantages of CFD is shown in Figure A.7. Researchers use a type of CFD approach called “large-eddy simulation” or LES to simulate the fluid dynamics of a tornado as it encounters a debris field and begins to pick up sand-sized particles. A full animation of this tornado simulation can be accessed by visiting the book web site. The motivation for this work is to investigate whether there are significant differences in the fluid mechanics when debris particles are present. Historically it has been difficult to get comprehensive experimental data throughout a tornado, so CFD is helping to shine some light on the complex fluid dynamics involved in such a flow.

A.7.2 Difficulties in CFD

One of the key points that a beginning CFD student should understand is that one cannot treat the computer as a “magic black box” when performing flow simulations. It is quite possible to obtain a fully converged solution for the CFD simulation, but this is no guarantee that the results are physically correct. This is why it is important to have a good understanding of the flow physics and how they are modeled. Any numerical technique (including those discussed above), no matter how simple in concept, contains many hidden subtleties and potential problems. For example, it may seem reasonable that a finer grid would ensure a more accurate numerical solution. While this may be true (as Example A.1), it is not always so straightforward; a variety of stability or convergence problems may occur. In such cases the numerical “solution” obtained may exhibit unreasonable oscillations or the numerical result may “diverge” to an unreasonable (and incorrect) result. Other problems that may arise include (but are not limited to): (1) difficulties in dealing with the nonlinear terms of the Navier–Stokes equations, (2) difficulties in modeling or capturing turbulent flows, (3) convergence issues, (4) difficulties in obtaining a quality grid for complex geometries, and (5) managing resources, both time and computational, for complex problems such as unsteady three-dimensional flows.

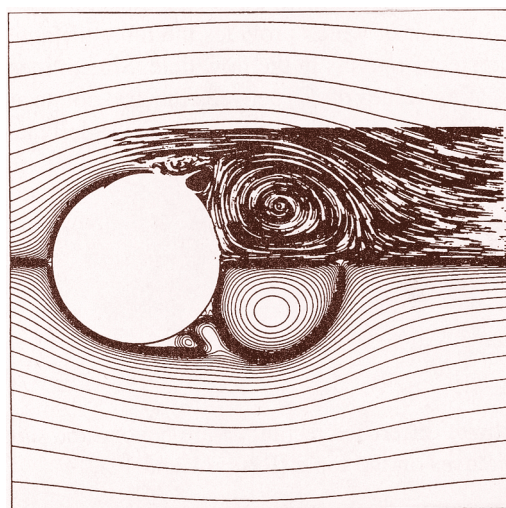
A.7.3 Verification and Validation

Verification and validation of the simulation are critical steps in the CFD process. This is a necessary requirement for CFD, particularly since it is possible to have a converged solution that is non-physical. Figure A.8 shows the streamlines for viscous flow past a circular cylinder at a given instant



■ **Figure A.7** Results from a large-eddy simulation showing the visual appearance of the debris and funnel cloud from a simulated medium swirl F3-F4 tornado. The funnel cloud is translating at 15 m/s and is ingesting 1-mm-diameter “sand” from the surface as it encounters a debris field. Please visit the book web site to access a full animation of this tornado simulation. (Photographs and animation courtesy of Dr. David Lewellen (Ref. 10) and Paul Lewellen, West Virginia University.)

after it was impulsively started from rest. The lower half of the figure represents the results of a finite difference calculation; the upper half of the figure represents the photograph from an experiment of the same flow situation. It is clear that the numerical and experimental results agree quite well. For any CFD simulation, several levels of testing need to be accomplished before one can have confidence in the solution. The most important verification to be performed is grid convergence testing. In its simplest form, it consists of proving that further refinement of the grid (i.e., increasing the number of grid points) does not alter the final solution. When this has been achieved, you have a grid-independent solution. Other verification factors that need to be investigated include



■ **Figure A.8** Streamlines for flow past a circular cylinder at a short time after the flow was impulsively started. The upper half is a photograph from a flow visualization experiment. The lower half is from a finite difference calculation. (See the photograph at the beginning of Chapter 9.) (From Ref. 9, used by permission.)

the suitability of the convergence criterion, whether the time step is adequate for the time scale of the problem, and comparison of CFD solutions to existing data, at least for baseline cases. Even when using a commercial CFD code that has been validated on many problems in the past, the CFD practitioner still needs to verify the results through such measures as grid-dependence testing.

A.7.4 Summary

In CFD, there are many different numerical schemes, grid techniques, etc. They all have their advantages and disadvantages. A great deal of care must be used in obtaining approximate numerical solutions to the governing equations of fluid motion. The process is not as simple as the often-heard “just let the computer do it.” Remember that CFD is a tool and as such needs to be used appropriately to produce meaningful results. The general field of computational fluid dynamics, in which computers and numerical analysis are combined to solve fluid flow problems, represents an extremely important subject area in advanced fluid mechanics. Considerable progress has been made in the past relatively few years, but much remains to be done. The reader is encouraged to consult some of the available literature.

References

1. Baker, A. J., *Finite Element Computational Fluid Mechanics*, McGraw-Hill, New York, 1983.
2. Carey, G. F., and Oden, J. T., *Finite Elements: Fluid Mechanics*, Prentice-Hall, Englewood Cliffs, N.J., 1986.
3. Brebbia, C. A., and Dominguez, J., *Boundary Elements: An Introductory Course*, McGraw-Hill, New York, 1989.
4. Moran, J., *An Introduction to Theoretical and Computational Aerodynamics*, Wiley, New York, 1984.
5. Anderson, J. D., *Computational Fluid Dynamics: The Basics with Applications*, McGraw-Hill, New York, 1995.
6. Thompson, J. F., Warsi, Z. U. A., and Mastin, C. W., *Numerical Grid Generation: Foundations and Applications*, North-Holland, New York, 1985.
7. Peyret, R., and Taylor, T. D., *Computational Methods for Fluid Flow*, Springer-Verlag, New York, 1983.
8. Tannehill, J. C., Anderson, D. A., and Pletcher, R. H., *Computational Fluid Mechanics and Heat Transfer*, 2nd Ed., Taylor and Francis, Washington, D.C., 1997.
9. Hall, E. J., and Pletcher, R. H., *Simulation of Time Dependent, Compressible Viscous Flow Using Central and Upwind-Biased Finite-Difference Techniques*, Technical Report HTL-52, CFD-22, College of Engineering, Iowa State University, 1990.
10. Lewellen, D. C., Gong, B., and Lewellen, W. S., *Effects of Debris on Near-Surface Tornado Dynamics*, 22nd Conference on Severe Local Storms, Paper 15.5, American Meteorological Society, 2004.