JOURNAL OF COMPUTATIONAL PHYSICS 69, 1–48 (1987)

Review

Simulation of Incompressible Turbulent Flows

JOEL H. FERZIGER

Department of Mechanical Engineering,
Stanford University, Stanford, California 94305

Received October 16, 1984; revised October 16, 1985

I. AN OVERVIEW OF TURBULENT FLOW SIMULATION

A. *Introduction*

Most flows of technological interest are turbulent, at least in some regions. For many of these flows, relatively simple prediction methods suffice to produce results of engineering accuracy. For others, mainly in high technology applications, more complex prediction methods are required. Computers now routinely run methods whose use would not have been considered even ten years ago. As the state of the art of computers has advanced, so has the range and size of the tasks demanded of them, a trend that will continue for the foreseeable future.

Early applications of computers in turbulent fluid mechanics were essentially automated versions of existing methods; this type of code is still in common use. As larger, faster, machines came along, more complex flow models were applied. Two areas of fluid dynamics that were in their infancy twenty years ago—computational fluid dynamics and turbulence modeling—have become heavily populated.

As users become ever more sophisticated, there will be a constant demand for improvement. Bigger, faster, and less expensive computers will always be important but, if maximum benefit is to be obtained from them and smaller existing machines,

0021-9991/87 $3.00

three other areas need attention. These are: computational algorithms (which can be subdivided into discretization methods and solution techniques), methods of dealing with complex geometries, and turbulence models. These areas are discussed, albeit briefly, in this paper.

The introduction of supercomputers has made possible entirely new approaches to fluid dynamics. One can now simulate simple turbulent flows accurately and use the results as a complement and/or partial substitute for laboratory data. The computer thus becomes a tool for studying the physics and modeling of turbulence. The methods used are called large eddy and full simulation.

B. *Levels of Turbulent Flow Simulation*

Kline, Lilley, and Cantwell [35] suggested a classification scheme for methods of turbulent flow simulation; this scheme was used with slight modification at the 1981 Conference on Complex Turbulent Flows (Ferziger, Bardina, and Allen [23]. A short summary of this scheme will be given here; readers are referred to the original papers for an elaboration of it. The scheme contains six levels:

(i) *Correlations.* For flows in standard geometries for which sufficient data are available, well-established engineering correlations do an adequate job.

(ii) *Integral Methods.* Prior to the introduction of computers, methods in which the time averaged Navier–Stokes equations are reduced to ordinary differential equations by integration over one or more coordinates were the most complicated ones in common use. Nonlinearities in the original equations prevent closure of the resulting equations so that approximations and/or correlations are required in these integral methods.

(iii) *One-Point-Average Methods.* Methods based on one-point (time or ensemble) averages of the Navier–Stokes equations have become very popular in recent years. A number of models of this type have been developed. These include one- and two-equation and Reynolds stress models; they will be reviewed below.

(iv) *Two-Point-Average Methods.* To permit explicit information on length scales of the turbulence to enter the model, methods based on two-point averages of the Navier-Stokes equations have been introduced. It is not yet clear whether these methods will prove practical for flows in complex geometries but they are of value in studying turbulence models.

(v) *Large Eddy Simulation* (LES). This method calculates the large-scale structures of the turbulence explicitly, while modeling the small ones. It is three dimensional and time dependent, and provides considerable detail about a flow. This makes it a valuable tool for investigating both the physics of turbulence and the models used to represent it. Its application is currently limited to simple flows and relatively low Reynolds numbers. It is too expensive for direct engineering use at the present time, but may see limited use in engineering design as the cost of supercomputers comes down and their numbers increase.

(vi) *Full-Turbulence Simulation* (FTS). In this approach, one solves the unaveraged time dependent Navier–Stokes equations for a turbulent flow numerically. The set of flows which can be treated in this way is small, but its accuracy makes FTS a complement to laboratory experiments for investigation of both physics and turbulence models. It is also ideal for investigating phenomena associated with the small scales of turbulence, especially subgrid scale models for large-eddy simulation.

With respect to this scheme, we should note:

(a) Methods at each level compute more details of a flow than those at lower levels, albeit at increased cost.

(b) Higher level model computations are generally less sensitive to the quality of the turbulence model than lower level ones. However, a lower-level model may, in a particular case, be more accurate than a higher-level one, due to better tuning or calibration or greater availability of the necessary data.

(c) The range of flows that may be simulated with a given model becomes broader as the level increases.

(d) The numerical methods used at each level differ.

(e) The amount of detail in the laboratory data required to provide initial and/or boundary conditions increases significantly with increasing level.

The working engineer will normally use methods at the lowest level capable of producing the desired accuracy.

In this paper, we shall concentrate on methods at levels (iii), (v), and (vi). Although we shall try to unify them, separate treatment is needed in some areas.

A cautionary note: A number of authors refer to their programs as "Navier–Stokes solvers." Nearly all of these programs actually solve an averaged form of the Navier–Stokes equations rather than the exact form, i.e., they are at level (iii) of the above scheme rather than level (vi). They would better be called Reynolds averaged Navier–Stokes (RANS) solvers.

C. *Full Field versus Zonal Treatment*

Use of the same set of equations everywhere in a flow, a single grid system over the entire domain, the same numerical approximation everywhere, and solution of the complete system of equations as a unit is very appealing. Unfortunately, for complex flows, this approach may be cumbersome and require too many grid points and too much computer time. Practical considerations dictate that different methods be used in different parts of the flow; differences in any or all of the component areas may be needed.

First, consider turbulence modeling. There is enormous appeal as well as considerable simplicity to using a single universal model. Unfortunately, turbulence physics differs considerably from zone to zone in most flows. Modeling any one of these zones is a difficult task; fitting all with a single model is much more difficult

and may not be possible at all; at the least, a complicated model would probably be required. Consequently, the author and some of his colleagues are not of the opinion that models need to be matched to the nature of each zone within a flow; the zonal models then need to be patched together. This approach may also better illuminate the areas in which new data are needed.

Second, consider geometry. Technological considerations dictate geometry; shapes may be quite complicated. While it is possible to map any region into a rectangle, finding the mapping for complex shapes can be very difficult; indeed, generation of the mapping can become a significant part of the total cost of a flow simulation. Furthermore, it is difficult to arrange the grid points to provide resolution where it is most needed; often the locations where resolution is needed most are not known in advance. Again, it is the opinion of the author and some of his colleagues that it is best not to use a single mapping for the entire flow. Patching together geometrically simple regions may be a better approach.

Third, consider numerical approximation. All three types of partial differential equations occur in fluid mechanics: hyperbolic, parabolic, and elliptic; often various zones of a single flow are of different types. Each type generally requires a different numerical treatment. The choice of numerical approximation may also depend on cell Reynolds number, flow direction, or other parameters. Consequently, it is not unusual to use different finite difference methods in different parts of the flow.

Finally, solution techniques which are efficient for one type of flow or one region of a flow perform poorly in other flows or regions. To circumvent this, one can use different solution techniques in different parts of the flow.

Thus, in every aspect of the numerical computation of turbulent flows, the trend seems to be toward methods which depend on the local properties of the flow. We call such methods zonal methods. This is at present a catchall phrase; the nomenclature will need refinement as the field develops. It is certain that fluid dynamics codes of the future will be more complicated than those of the past.

D. *Error Assessment*

A nagging difficulty in turbulent flow computation and, indeed, in many related areas is that of assessing and controlling errors. Compounding the problem is the fact that errors arise from several distinct sources. First, turbulence models may not accurately represent the physics of turbulence. Second, errors arise from numerical approximations; these may be due to the mapping, the finite difference approximation, and/or the lack of complete convergence of the solution technique. It is a rare case in which the error in a numerical solution is accurately evaluated and still more rare to find attribution of the sources of the error.

If the accuracy of turbulence models is to be assessed, numerical methods capable of solving a given set of equations with assured accuracy are needed. Grid refinement is the traditional way of assessing numerical error; while it is valid if the errors are small enough, it is always very costly. New methods of assessing and con-

trolling errors are badly needed. The most promising avenue of attack on this problem at the present time is the use of solution adaptive grids.

E. *Outline of This Paper*

We have laid out a number of issues that need to be addressed. Each could occupy a review paper of its own; most have been the subjects of such papers. Rather than attempting an exhaustive review of each issue, we shall attempt to integrate them into a whole. In so doing, we shall provide only enough detail of each area to show how the parts fit together and to point out where the major remaining difficulties lie.

These issues will occupy the remainder of this paper. In Section II, turbulence models will be discussed. Section III addresses the treatment of geometric complexity, Section IV will discuss the need for conservation of important variables. Discretization is taken up in Section V and solution methods in Section VI. Some special methods are described in Section VII, and Section VIII summarizes the present state of the art. Completeness is neither claimed not attempted in any area.

II. Reynolds Average Turbulence Models

A. *The Reynolds Stresses*

This section will concentrate on models used in conjunction with the Reynolds-averaged Navier–Stokes equations. This review is brief; readers desiring a more extensive survey are referred to the monograph by Rodi [54] or the earlier review by Reynolds [52]. This section is also limited in scope; compressibility, heat transfer, combustion, and other issues are not dealt with here.

To set the stage, let us briefly examine why turbulence models are needed. The time-dependent, incompressible Navier–Stokes equations are, in tensor notation,

$$\rho \frac{\partial u_i}{\partial t} + \rho \frac{\partial}{\partial x_j} u_i u_j = -\frac{\partial p}{\partial x_i} + \mu \frac{\partial^2 u_i}{\partial x_j \partial x_j}, \tag{2.1}$$

where $u_i(\mathbf{r}, t)$ is the local fluid velocity, ρ the density, p the pressure, and μ the viscosity. These equations must be solved together with the continuity equation, which requires the velocity field to be divergence-free:

$$\frac{\partial u_i}{\partial x_i} = 0. \tag{2.2}$$

This equation is linear and causes no problems so far as averaging is concerned.

The mean velocity field may be defined by any of a number of averaging operations; ensemble averaging, time averaging, and averaging over spatial coordinates in which the mean flow is homogeneous have been used. Any of these

averages, together with the notion that the velocity can be decomposed into a mean and turbulent fluctuations, i.e., $u_i = U_i + u_i'$, where U_i and u_i' represent the mean and fluctuating velocities, respectively, leads to the Reynolds-averaged Navier–Stokes equations

$$\rho \frac{\partial U_i}{\partial t} + \rho \frac{\partial}{\partial x_j} U_i U_j = -\frac{\partial \bar{p}}{\partial x_i} + \mu \frac{\partial^2 U_i}{\partial x_j \partial x_j} - \frac{\partial}{\partial x_j} \rho \overline{u_i' u_j'}. \tag{2.3}$$

It is important to observe that the mean velocity is a defined quantity and may not capture the physics of a turbulent flow. Indeed, the observation that long times are required to achieve stable averages in the laboratory should make one cautious about accepting the mean-turbulent decomposition as anything more than a convenience.

The terms $\rho \overline{u_i' u_j'}$ in Eq. (2.3) are called the Reynolds stresses. They result from the nonlinearity of Eq. (2.1) and represent the effect of the turbulence on the mean field. Their presence results in the unknowns outnumbering the equations and the system not being closed. Closure is a central problem in simulating turbulent flows.

B. *The Eddy Viscosity Concept*

Two of the most important effects of turbulence on the mean flow are: (i) it absorbs kinetic energy from the mean flow and converts it to thermal energy (dissipation), and (ii) it increases the rate of transport of mass, momentum, and energy normal to the streamlines of the flow (diffusion). As both of these effects are mediated by the viscosity in laminar flows, it is natural to assume that the effect of turbulence on the mean flow can be represented as an increased, or eddy, viscosity. This leads to

$$\overline{u_i' u_j'} = -\nu_T \left(\frac{\partial U_i}{\partial x_j} + \frac{\partial U_j}{\partial x_i} \right) + \frac{2}{3} k \delta_{ij} \tag{2.4}$$

as a model of the Reynolds stresses in turbulent flows; here $k = \overline{u_i u_i}/2$ and ν_T is the eddy viscosity.

To close Eq. (2.3) with the model (2.4), one must either express the eddy viscosity in terms of the mean velocity or provide equations which govern its evolution. Both approaches are used. For shear flows, the most important class of flows in practice, the most significant components of the Reynolds stress tensor are its off-diagonal elements. In two-dimensional flows, the only off-diagonal element is $\overline{u'v'}$, and the eddy viscosity becomes the coefficient of proportionality between this Reynolds shear stress and rate of hear, dU/dy. Good correlations for the eddy viscosity in two-dimensional flows are available (see below). However, the proportionality assumed in Eq. (2.4) is not correct in general and should be used with caution in more complex flows. In three dimensions, the situation is much more difficult, as the eddy viscosity should be regarded as a fourth-rank tensor.

The simplest descriptions of turbulence are those which characterize it by its kinetic energy, $\rho q^2/2$, and an average length scale, L. In terms of these, dimensional analysis shows that the eddy viscosity, which carries dimensions L^2/T, must be written

$$\nu_T = CqL, \tag{2.5}$$

where q and L are velocity and length scales of the turbulence, respectively, and C is an empirical constant or a function of dimensionless parameters. This is the basis of a number of models; some are described in the following two sections.

C. *Mixing Length Models*

To use Eq. (2.5), we need expressions for q and L. In mixing length models, q and L are written as algebraic expressions involving the mean velocity field.

For shear flows, experiments show that, far from solid walls, the largest turbulence length scales are some fraction of the width of the shear layer; the latter is then the natural length scale. Near a solid boundary, the distance to the surface bounds the size of the eddies and becomes the natural length scale. In the region closest to the wall, a further correction for viscous effects is required. Thus, for wall-bounded flows, the length scale typically has the distribution depicted in Fig. 1. For free shear flows, a constant length scale usually suffices.

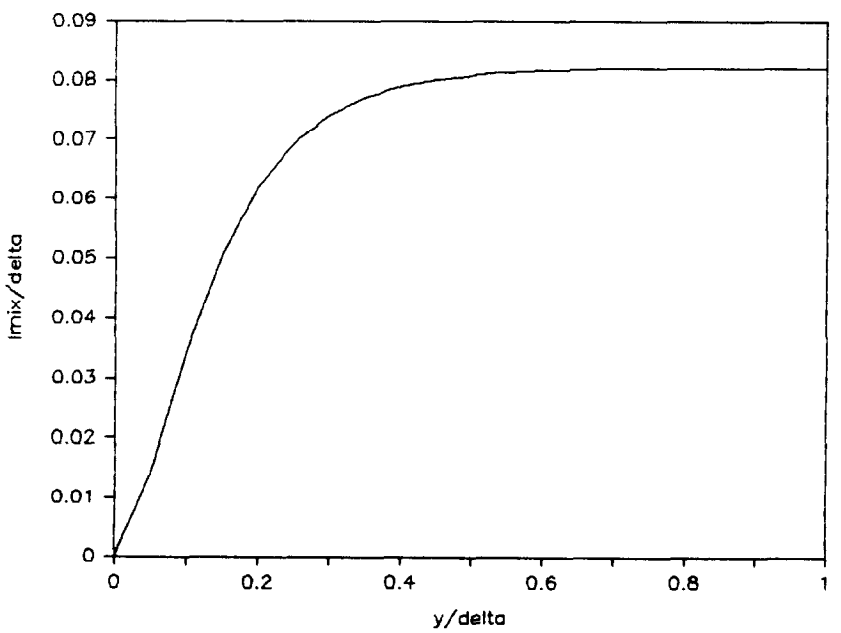

FIG. 1. The mixing length distribution in a turbulent boundary layer.

To complete the model, an expression for the turbulent kinetic energy or, equivalently, the turbulent velocity scale, q, is required. The production of turbulent kinetic energy is proportional to the product of the rate of shear and the turbulent kinetic energy itself (see below). Using this and an assumption that the production and dissipation of turbulence energy are in local balance, one can show that a good approximation to the local turbulent velocity scale is

$$q = CL\left|\frac{\partial U}{\partial y}\right| \tag{2.6}$$

(cf. Tennekes and Lumley [59]). For three-dimensional flows, one replaces $\partial U/\partial y$ by the r.m.s. strain rate of the mean field. This model was first derived by Prandtl using different arguments.

Although some of the arguments used to derive them are open to question, mixing-length models have proven of great value in a variety of simple flows. Their simplicity makes them easy to modify to account for effects such as pressure gradients, curvature, and transpiration through a wall. Kays, Moffat, and their students (cf. Kays and Crawford [31]) have made a long series of experiments to investigate these effects and have built a model which does an excellent job for a wie range of boundary layers. Cebeci and Smith (cf. Cebeci and Bradshaw [13]) have an equally good alternative model. An example of what this model can do is shown in Fig. 2; equally good results can be obtained for free shear flows.

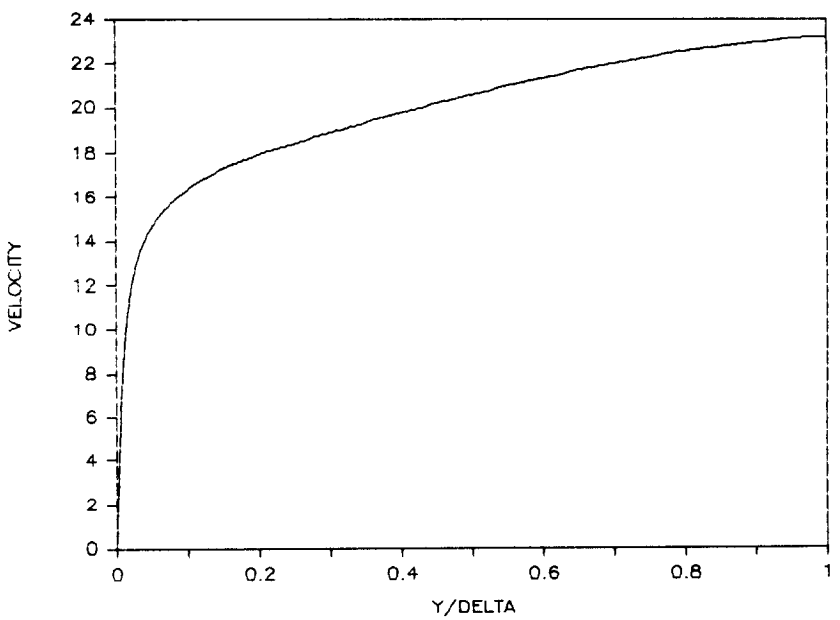

FIG. 2. The velocity profile in a turbulent boundary layer as calculated by the mixing length model.

Mixing-length models are excellent for simple, i.e., single-zone, flows. Their chief deficiency is in switching from one type of region to another, e.g., from a boundary layer to a free shear layer, within a single flow, as is required in simulations of separated flows. They do not include history effects on the turbulence. Indeed, good models for the mixing length and the turbulent velocity scale for flows containing separation and reattachment do not exist, although the Baldwin–Lomax model (Baldwin and Lomax [5]) has been used in this application. This deficiency of mixing-length models is one of the principal reasons for the development and popularity of two-equation models.

D. *Two-Equation* $(k-\varepsilon)$ *Models*

The first step beyond the mixing length is a model in which the length scale is prescribed, but the evolution of the turbulence kinetic energy is determined by a partial differential equation; these are called one-equation models. As the weakness of the mixing-length model is due as much to the inadequacy of the length scale as to the velocity scale assumptions, one-equation models are not much of an improvement on mixing-length models. We shall therefore bypass them and proceed to the two-equation models, which are finding a great deal of application at present.

Two-equation models were first developed by Harlow and Hirt [73] and later developed by Jones and Launder [30] and have since been used in many applications. These models employ the Boussinesq eddy-viscosity concept, Eq. (2.4), but the evolution of both the velocity and length scales is obtained by solving partial differential equations. An equation for the turbulent kinetic energy determines the velocity scale. This equation can be derived from the Navier–Stokes equations by subtracting the mean equation from the unaveraged equation to obtain an equation for the fluctuating velocity. Taking the scalar product of this equation with the fluctuating velocity and averaging according to any of the methods mentioned above yields the equation for the turbulent kinetic energy:

$$\frac{\partial k}{\partial t}+\frac{\partial}{\partial x_j}U_j k=\underbrace{-\overline{u_i'u_j'}\left(\frac{\partial U_i}{\partial x_j}+\frac{\partial U_j}{\partial x_i}\right)}_{\text{Production}}\underbrace{-\nu\overline{\frac{\partial u_i'}{\partial x_j}\frac{\partial u_i'}{\partial x_j}}}_{\text{Dissipation}}+\underbrace{\text{Diffusion}}_{\text{Convection}}, \tag{2.7}$$

where $k=q^2/2$.

The significance of the convective term is clear; the production term accounts for creation of new turbulence energy from the mean field; the dissipation term represents the conversion of turbulence energy to thermal energy by the action of viscosity; the diffusion term, which we have not written explicitly, contains third-order correlations of the fluctuating velocity and pressure-velocity correlations. The convection term is exact, the production term can be computed from the model itself, the method of obtaining the dissipation is explained below, and the diffusion

term must be modeled. We shall not pursue this further here; the interested reader is referred to the references cited above for details.

This leaves the question of how to obtain the length scale and the dissipation term in Eq. (2.7). One of the best accepted arguments in turbulence theory is that the dissipation, ε, and the integral length scale, L, are related by

$$\varepsilon = Cq^3/L. \tag{2.8}$$

Since the dissipation occurs in the energy equation and it can be related to the length scale by Eq. (2.8), one is led to using an equation for the dissipation as a means of obtaining the length scale. Other possibilities have been tried, but none has had as much success as this one.

An equation for the dissipation can be derived from the Navier–Stokes equation by an extension of the method used to derive the energy equation and has a similar form

$$\underset{\text{Convection}}{\frac{\partial \varepsilon}{\partial t} + \frac{\partial U_j \varepsilon}{\partial x_j}} = \underset{\text{Dissipation}}{\text{Production of}} - \underset{\text{Dissipation}}{\text{Destruction of}} + \text{Diffusion} \tag{2.9}$$

Here the convection term causes no problem, but all the others require modeling. The modeling used in one popular version of the $k-\varepsilon$ model is given in Table I and is taken from Rodi [54].

TABLE I

Rodi's $k-\varepsilon$ Model

$$\frac{\partial k}{\partial t} + U_j \frac{\partial k}{\partial x_j} = -\overline{u_i' u_j'} \left(\frac{\partial U_i}{\partial x_j} + \frac{\partial U_j}{\partial x_i} \right) - \varepsilon + \frac{\partial}{\partial x_j} \frac{\nu_T}{\sigma_k} \frac{\partial k}{\partial x_j}$$

$$\frac{\partial \varepsilon}{\partial t} + U_j \frac{\partial \varepsilon}{\partial x_j} = C_{1\varepsilon} \frac{P\varepsilon}{k} - C_{2\varepsilon} \frac{\varepsilon^2}{k} + \frac{\partial}{\partial x_j} \frac{\nu_T}{\sigma_\varepsilon} \frac{\partial \varepsilon}{\partial x_j},$$

where

$$P = -\overline{u_i' u_j'} \left(\frac{\partial U_i}{\partial x_j} + \frac{\partial U}{\partial x_i} \right)$$

is the production

$$\overline{u_i' u_j'} = \nu_T \left(\frac{\partial U_i}{\partial x_j} + \frac{\partial U_j}{\partial x_i} \right),$$

$$\nu_T = C_\mu k^2/\varepsilon,$$

and the constants are

$$C_{1\varepsilon} = 1.44, \quad C_{2\varepsilon} = 1.92, \quad \sigma_k = 1.0, \quad \sigma_\varepsilon = 1.3, \quad C_\mu = 0.09.$$

As stated earlier, two-equation models (and the $k-\varepsilon$ model in particular) are quite successful and have become very popular. With only little modification, they are able to simulate a large variety of flows with reasonable accuracy, including flows that are difficult to treat with mixing-length models.

Despite this success, caution is need. Although it is rare for the standard $k-\varepsilon$ model to be badly in error, it is often not accurate enough for engineering purposes, particularly in applications demanding high accuracy. In the latter, well-tuned mixing-length models may sometimes be more accurate. One needs to be especially careful about accepting results for flows which are very different from those for which the model was calibrated.

Some specific problem areas are the following:

(i) In flows which are three-dimensional in the mean, experience with these models is limited. What evidence is available indicates that the models have difficulty in simulating the effect on turbulence of the changes in mean-flow direction across the shear layers in these flows. The Boussinesq stress-strain relationship (2.4) is not valid under these conditions, forcing one to use a more complex model which takes the history of the Reynolds stress into account.

(ii) In highly strained flows, the model badly overpredicts the production of turbulence (Lee and Reynolds [36]).

(iii) In natural convection flows, two-equation models are incapable of predicting the production of vorticity by the action of buoyancy, and more complex models are required (Naot, Shavit, and Wolfshtein [47]).

(iv) In compressed flows, the dissipation cannot act to determine both the destruction of turbulence energy and the length scale, and a third equation must be added to the model (Wu, Ferziger, and Chapman [66]).

There are undoubtedly other flows which demonstrate weaknesses of the two-equation model, but these examples argue for the caution recommended above.

E. *Reynolds Stress Models*

Although they have had many successes, two-equation models have a number of deficiencies, some of which are consequences of the eddy-viscosity relationship for the Reynolds stresses. In addition to the items mentioned above, they cannot predict counter-gradient fluxes, i.e., fluxes which tend to increase rather than decrease the gradient of a quantity; there are important flows in which such fluxes exist. Perhaps more significantly, models based on the eddy-viscosity concept produce abrupt and sometimes large changes in the Reynolds stresses when a new strain is applied to the mean flow; this is at odds with observation.

Reynolds stress models were developed to eliminate these problems. They are based on evolutionary equations for the Reynolds stresses which can be derived in

much the same way as the turbulent kinetic-energy equation. There results a set of equations that can be written:

$$\frac{\partial R_{ij}}{\partial t}+\frac{\partial}{\partial x_k}u_k R_{ij}=-R_{ik}\frac{\partial U_j}{\partial x_k}-R_{jk}\frac{\partial U_i}{\partial x_k}-\nu\overline{\frac{\partial u_i'}{\partial x_k}\frac{\partial u_j'}{\partial x_k}}-\overline{p'\left(\frac{\partial u_i'}{\partial x_j}+\frac{\partial u_j'}{\partial x_i}\right)} \quad (2.10)$$

Convection | Production + Diffusion | Dissipation | Redistribution

Most of the terms in these equations are analogs of terms in the turbulent kinetic-energy equation (which is the trace of these equations). The new element is the redistribution term which produces transfer among the components of the Reynolds stress tensor. The new terms, plus the fact that all terms are now tensors, make modeling considerably more difficult. These models contain more constants than the models described above. This provides the hope of greater universality but also increases the difficulty of finding the relationships and constants needed. The problem is made more difficult by the lack of methods of measuring fluctuating pressure in turbulent flow. This prevents accumulation of the data needed.

Reynolds stress models have been applied to relatively few engineering flows to date but have found some application in meteorology. Their record is mixed. The model rectifies some of the shortcomings of the simpler models, but in the 1980–1981 Stanford meeting (Kline *et al.* [35]) it was found to be no better on average than the models described above. More recently, Lee and Reynolds [36] have found that a complicated model is needed even to deal with homogeneous turbulence. Tzuoo *et al.* [63] found that, even for homogeneous flows, Reynolds stress models need to be tailored to the type of strain imposed (see Sect. G). As this model is also more expensive to apply, its future is not necessarily assured.

As mentioned above, Reynolds stress models were developed to overcome deficiencies in mixing length and two-equation models. However, they are not without problems of their own; the question is whether the benefits outweigh the costs. On the deficit side of the ledger, Reynolds stress models nearly double the cost of computing a given flow and contain terms which are very difficult to measure in the laboratory. Their development has been relatively slow; the major difficulty seems to be the lack of accurate data on which to base them.

F. *Algebraic Stress Models*

An approach to modeling which attempts to obtain the advantages of both two-equation and Reynolds stress models (with as few of the disadvantages as possible) was proposed by Rodi [53]; this is the algebraic approach to turbulence modeling. This approach begins with the Reynolds stress model equations, but, for steady flows, reduces the number of differential equations by making approximations to the terms which contain spatial derivatives. In particular, the convection and diffusion terms are approximated in terms of local variables and derivatives of mean quantities. The result is that the partial differential equations for the Reynolds

stresses are reduced to a system of algebraic equations at each point in the flow. Equations for the turbulence kinetic energy and dissipation are retained, and the resulting model is only slightly more expensive in application than the two-equation model.

This model has considerable appeal, as it offers the advantages of both of the preceding approaches with few of the disadvantages. In practice, it seems to perform as well as two-equation models but no better. However, as there is less experience with algebraic methods than with two-equation models, it may be that algebraic models will improve as they mature.

G. *Zonal Models*

At the present state of the art in turbulence modeling, one can use models with confidence only to simulate flows which are not too different from those used to calibrate the models; caution is required when new flows are computed. To use Bradshaw's term, the models are *post*dictive rather than *pre*dictive in nearly all instances.

All of the turbulence models presented above are based on the notion that high-order moments of the fluctuating velocity can be represented as unique functionals of the low-order moments. This idea was borrowed from the kinetic theory of gases, where it can be justified rigorously and works well. Unfortunately, the experimental evidence is that, in turbulent flows, this assumption is not correct. In all flows for which there are data, the low-order moments reach their final values before the high-order ones do. Thus a fundamental premise of much of turbulence modeling is apparently incorrrect.

What is one to do? Turbulence modeling is difficult, and good new ideas are rare. A lot has been accomplished with the existing models, and one ought to take advantage of the accumulated knowledge. A change in point of view offers a possibility. Rather than regarding them as scientific laws, one can think of turbulence models as sophisticated engineering correlations. Compared to the correlations at level (i) that were described in Section I.-B, they are at a much more fundamental level. A single model allows the prediction of many more flows in a greater range of geometries than do the simple correlations. If we then admit that a universal model may not be attainable and adopt a more pragmatic approach, we are led to the idea of zonal modeling (Kline [34]).

Over the past decade, experimenters have shown that there are coherent structures which are responsible for much of the important physics of turbulent flows. The structures differ from flow to flow in simple flows, so no single model should be expected to represent the physics of all of them. However, it may be possible to find a separate model for each type of flow; in fact, it should be much easier to do so, provided the list of flows for which modeling is needed is not too large; this appears to be the case.

In complex flows, including nearly all flows of interest in technology, there are a

number of regions or zones with differing physics. Separated flows, for example, have both free shear layers and boundary layers. In each zone, one can use a model appropriate to its physics. In the transitions from one zone to another, some kind of blending model is required. Since models for many of the zones already exist, the development of the blending models is the most difficult part of the task. This idea has been tested in a simple way by one of the author's students for homogeneous flows and found to work very well (Tzuoo *et al.* [63]). It shares a philosophy with the well-known mathematical method of patching together asymptotically correct solutions. Testing on more complex flows is in progress.

III. Treatment of Geometrically Complex Flows

A. *Overview*

We noted earlier that many technologically important flows involve geometries that do not fit easily into standard coordinate systems. The simulation of flows in arbitrarily shaped regions has received much more attention for compressible flow, but much of what has been done can be applied to incompressible flow as well. The area is far too broad to be covered in detail here; for further information, the interested reader is referred to the references cited below.

The choices of methods of dealing with geometric complexity and discretizing the equations are closely connected. A large majority of the computations of incompressible flows have used staggered grids, which require rectangular or other simple coordinate systems. Consequently, they have mainly dealt with simple geometries.

The oldest methods for dealing with irregular shapes employ rectangular grids. Intersections of the grid lines with the domain boundaries are taken as nodes. This causes some boundary grids to be much smaller than the interior grids. Difference formulas on these grids produce large and irregular truncation errors near the boundary, resulting in unsatisfactory solutions. A related method which uses a "staircase" fit to the boundary also yields poor results. These methods may make a comeback when embedding or adaptive methods are developed.

Finite element methods were developed specifically to deal with arbitrary geometries; they do this very well. Taken together with their variational base, this has gained them dominance in solid mechanics. Their principal disadvantage is that the cost of solving the discretized equations is larger than for finite-difference methods; work on reducing the cost is in progress. In fluid mechanics, some of the advantages of finite elements vanish, and it is not clear whether they are the method of choice. Furthermore, nearly any significant development in finite difference methods is quickly adapted to finite elements and vice versa, so it is sufficient to discuss one or the other. For a more complete discussion of finite element methods for fluid mechanics, see Baker [4].

At present, the most popular method of dealing with complex geometries is to

map the flow domain onto a simpler region, such as a rectangle. For two-dimensional problems, this has proved successful, particularly for compressible flows. Although this approach becomes very complicated for extremely complex geometries (e.g., multiply-connected regions) or three dimensions, mapping methods will be useful as components of more advanced methods. We shall give a short overview of this subject in this section and end with a review of some recent developments in adaptive and patched grids.

B. *Mapping Methods*

1. *Strategy*

As noted above, the poor fit of rectangular grids to the boundaries of irregular domains accurately can lead to large errors. A coordinate system fitted to the domain boundary can eliminate this problem. Such coordinate systems must be curvilinear but need not be orthogonal. Of course, the coordinate lines must not be too nearly parallel; a rule of thumb requires the grid lines to cross at no less than 45°.

A mapping appropriate to a two-diminsional wing is illustrated in Fig. 3. Due to the shape of the coodinate surfaces, this is called an *O*-mapping; *C*- and *H*-mappings are also frequently employed. In the case illustrated, the body is mapped into the lower boundary of the computational coordinate system. The outer boundary of the physical coordinate system, which is the computational approximation to the far-field boundary conditions, is mapped into the upper boundary of the transformed domain, while the two parts of the cut required by the multivalued nature of the coordinate system become the lateral boundaries of the transformed system.

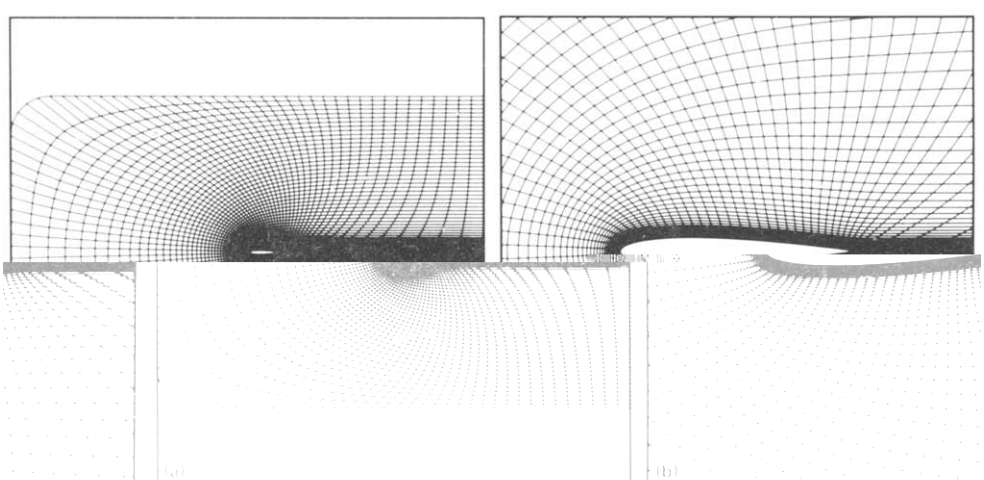

FIG. 3. Grid system for the computation of flow about a two-dimensional wing generated by the elliptic mapping method. Figure provided by C. Hessenius.

Several methods of generating body-fitted coordinate systems have been proposed. All have advantages, but none is clearly superior to all others. Consequently, a number of methods are in current use. In this section, we briefly describe three popular approaches. The descriptions are brief; for details, the reader is referred to the cited papers.

The methods described in this section involve making a transformation of the independent variables from the physical coordinates (x, y) to computational coordinates (ξ, η) such that the considered region becomes a rectangle in computational space. The first two methods we shall consider generate the transformation via the solution of partial differential equations. However, computing $\xi(x, y)$, $\eta(x, y)$ requires generating this solution in the original geometry, a problem we are trying to avoid. Consequently, the standard practice is to compute the inverse transformation, $x(\xi, \eta)$, $y(\xi, \eta)$, which can be done in the transformed rectangular domain. A typical derivative appearing in the Navier–Stokes equations is, in two dimensions,

$$\phi_x = J^{-1}(\phi_\xi y_\eta - \phi_\eta y_\xi), \tag{3.1}$$

where $J = x_\xi y_\eta - x_\eta y_\xi$ is the Jacobian of the transformation.

In the remainder of this section, we shall look at some particular coordinate transformations.

2. *Elliptic Mapping Methods*

One method of generating mappings uses elliptic partial-differential equations. An obvious choice uses the potential and streamlines of the potential flow in the given geometry as computational coordinates. However, that approach does not give enough control over the spacing of the grid points. To permit greater control of the grid spacing, Thompson, Thames, and Mastin [61] suggested using a Poisson equation:

$$\nabla^2 \xi = G_1(x, y) \tag{3.2}$$

as a grid generator; the source term allows control of the grid density in almost any way the user likes. As mentioned above, one actually solves for the inverse transformation; this requires solution of a modified Poisson equation in the transformed coordinate system.

3. *Hyperbolic Mapping Method*

The elliptic mapping scheme just described has a number of advantages, including flexibility. Its principal disadvantage is that elliptic equations require iterative solution. As a result, the cost of generating the coordinate transformation may become a significant fraction of the total cost of solving the problem. This led Sorenson and Steger [56] to introduce hyperbolic methods for grid generation.

Hyperbolic equations offer the advantage that they can be solved by noniterative marching methods. For the wing-shaped body of Fig. 3, one assigns the body a

fixed value of one coordinate, ξ, say, and assigns values of η around the body. The directions of the lines $\eta = \text{constant}$ are fixed by requiring them be orthogonal to the body, and the next line of constant ξ is fixed by requiring all of the grid volumes to be equal (other choices are possible). This leads the following equations for the coordinates.

$$x_\xi x_\eta + y_\xi y_\eta = 0 \qquad \text{(orthogonality)}, \tag{3.3a}$$

$$x_\xi y_\eta - y_\xi x_\eta = 1 \qquad \text{(volume)}. \tag{3.3b}$$

One then solves these equations by a marching procedure until the coordinates have been extended sufficiently far from the body.

The principal difficulty with this approach is control. If the body is peculiarly shaped, the coordinate lines may cross and extra logic is needed. The method is also difficult to apply to multiply connected regions.

4. *Algebraic Mapping Methods*

A number of other methods of generating grids have been suggested. At present, the two presented above are among the most popular, but other methods are still frequently used.

A simple approach is to draw a grid system on the physical region in the manner of a draftsman. One then supplies the coordinates of the grid points to the computer, and all of the derivatives required are computed numerically. This has the advantage relative to the method described above that the computor can use his/her experience to place the grid points where they are most needed. However, the disadvantages of requiring a great deal of personal attention and the difficulty of applying it in three-dimensional grids have reduced the popularity of this approach. If artificial intelligence methods such as expert systems can be used to automate this method, it may prove valuable in the future.

Another alternative has been advocated by Eiseman and colleagues (Eiseman and Smith [21]). In this method, the considered region is broken into a number of smaller, geometrically simple regions, each of which can be fit with a simple grid. The grids are made to match on their boundaries.

C. *Patched Grids*

For geometries that are very complicated, especially multiply-connected and three-dimensional domains, the grid-generation methods described above do not work well. There seems to be general agreement that a good strategy is to divide the domain on which the computation is to be done into geometrically simpler subdomains, each of which can be gridded by the techniques described above; this is sometimes called a zonal method. There is not yet agreement, however, on whether the grids should overlap or not. The argument in favor of the non-overlapping approach that it wastes no grid points. The advocates of overlapping grids claim faster convergence and greater flexibility as their chief advantages.

Rai [50] has given a method using non-overlapping patched grids. Figure 4 shows a grid of this type used by Hessenius and Rai [28]. These authors have pointed out the need for careful attention to conservation at the subdomain boundaries. A similar method has been given by Mastin and McConnaughey [45].

Overlapping patched grids have been used by Glowinski *et al.* [72]. A major effort centers on the development of fast solution strategies for this type of grid.

D. *Adaptive Grids*

Fluid flows contain phenomena on a wide range of length scales; thin layers occur in many kinds of flows. Well-known examples are boundary layers on solid surfaces, shock waves, and hydraulic jumps. Resolution of these thin layers is often essential to the achievement of an accurate overall solution. Resolution requires that these thin layers be covered with a substantial number of grid points. Unfortunately, in most gridding procedures, including those described in Section B, putting enough points where they are needed also places an unnecessarily large number of points in regions where they are not needed.

The problem is made still more difficult when the location of the thin regions is

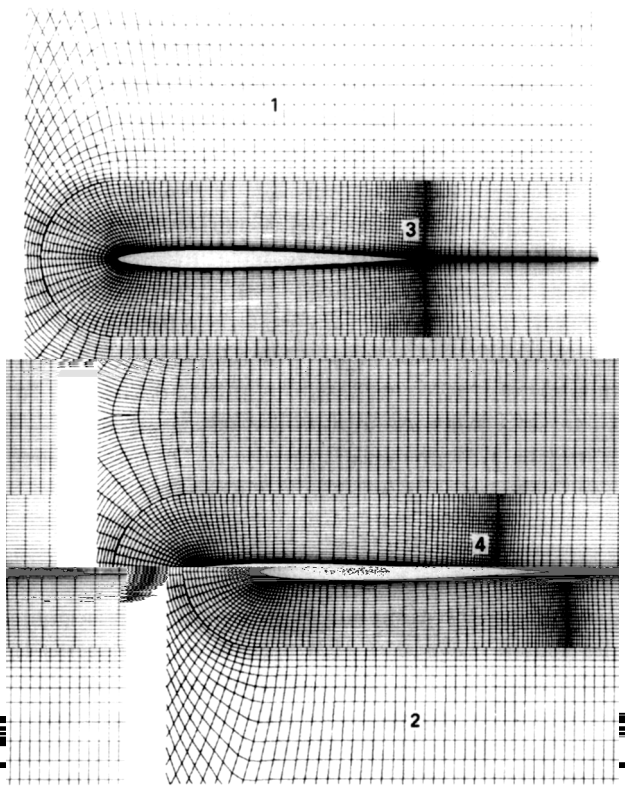

FIG. 4. A composite or zonal grid for a two-airfoil configuration. Adapted from Hessenius and Rai [28].

not known a priori. It is then impossible to generate a satisfactory grid in advance of computing the flow. Instead, the grid must be adjusted as the solution proceeds; techniques of this kind are called solution adaptive gridding methods. Such methods have been developed only recently; we shall discuss some of the more promising methods; there is no agreement as yet on which methods are best.

Adaptive procedures are made up of several parts. First, there must be a means of detecting where additional grid points are needed. Then a procedure for inserting or moving the grid points is required. Finally, depending on the strategy adopted, either a means of initializing the data on the new grid or a method of passing information between the grids is required; the choices may depend on the nature of the problem.

Since thin layers contain large derivatives of at least some flow variables, one [redacted]

Anderson and his colleagues (cf. Anderson [1]). These methods work quite well for some problems but have difficulty in others.

An alternative strategy was proposed by Berger [9]. In this approach, the error is detected by computing the solution on two grids of different refinement; these are normally the grid on which one actually wishes to do the calculation and one twice as coarse. The error is obtained from a Richardson-type estimator of the kind used in ordinary differential equations; for a first-order method, one would use

$$\varepsilon_h \simeq \phi_h - \phi_{2h}, \tag{3.4}$$

where ϕ_h is the solution obtained with a grid of size h. This requires that the error be expressible as a series in h, which is usually the case. The points with large errors are grouped into clusters which are fit with rectangles; for efficient coverage, the rectangles are allowed to have arbitrary rotation and to overlap. An example of the kind of grid produced by this method is shown in Fig. 5.

For some types of problems, including many hyperbolic and parabolic equations, the coarse grid solution can be accepted as accurate outside the refinement region

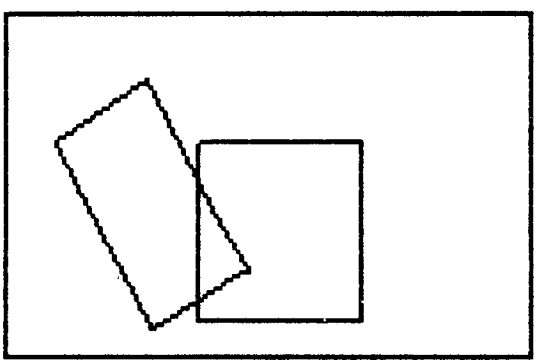

FIG. 5. An illustration of the adaptive grid procedure of Berger [9].

and used to provide boundary conditions to the finer grids. In other problems, particularly elliptic ones, it is necessary to iterate between the coarse and fine grids. In either case, the method can be used recursively: grids are nested until the desired accuracy is achieved; Berger reported cases with as many as six levels of grids. This method has the advantage that the grids tend to align with the flow, thus minimizing numerical diffusion (see Sect. V).

IV. Conservation Laws and Their Interaction with Numerical Methods

A. *Conservation Properties*

Incompressible flow computation offers mathematical difficulties absent in compressible flow; these affect the choice of numerical method in a strong way. In incompressible flows, the momentum equation is responsible for both momentum and kinetic energy conservation; compressible flows have a separate energy conservation equation. The absence of a time derivative in the continuity equation is another manifestation of this problem. Methods of solving the incompressible equations that conserve both momentum and energy are not easily constructed.

The primary issue considered in this section is global conservation of mass, momentum, and kinetic energy; these are important properties of the Navier–Stokes equations whose preservation in the discretized equations is of considerable importance. There may be further properties whose global conservation is of significance, depending of flow type.

The global conservation laws may be derived by integration of the differential conservation equations over a finite volume. The resulting equations state that the total amount of the conserved quantity in the macroscopic volume is changed only by flows through the surface of that volume; in the case of kinetic energy, there is a volumetric viscous dissipation term.

For the incompesssible Navier–Stokes equations (2.1), momentum conservation is generally easy to preserve in the discretization process; it follows easily if each term of the continuum equation is written as a divergence, i.e., in so-called conservation form. Kinetic energy conservation, on the other hand, is a bit more difficult. One takes the scalar product of Eq. (2.1) with the velocity and integrates over the larger control volume. For the convective and pressure gradient terms, one must use the chain rule for differentiation (or its equivalent, partial integration) and the continuity equation (2.2). The resulting macroscopic form of the energy conservation equation is

$$\frac{d}{dt}\int_V \frac{\rho u_i u_i}{2}\,dV = \underbrace{-\int_S \left(\frac{\rho u_i u_i}{2} + p\right) u_i\,dS_j}_{\text{Flow work}} + \underbrace{\mu \int_S \frac{\partial u_i}{\partial x_j}\,u_i\,dS_j}_{\substack{\text{Surface Viscous} \\ \text{Stresses}}} \underbrace{-\mu \int_V \frac{\partial u_i}{\partial x_j}\frac{\partial u_i}{\partial x_j}\,dV}_{\text{Dissipation}} \tag{4.1}$$

which states that the rate of change of kinetic energy in the control volume is due entirely to surface terms and viscous dissipation.

A discretization scheme is said to be conservative if it satisfies a numerical form of Eq. (4.1). Numerical global conservation equations are derived by the discrete equivalent of the procedures followed for the continuum equations. The problem is that the discretized operators may not possess all of the necessary properties. In particular, many finite difference approximations do not admit the chain rule. Finding a conservative scheme then becomes a hit-or-miss affair. An excellent example of a scheme that has many desirable conservation properties is that of Arakawa [2].

There has been considerable controversy over whether a method needs to be fully conservative or not. There is evidence on both sides of this issue. It appears that fully conservative methods are less prone to instability, particularly when iterative solution methods are used. However, accurate solutions have been obtained with nonconservative methods.

We offer a speculation about the necessity of conservation in discretization schemes. Nonconservative methods almost certainly contain artificial sources of properties that are not explicitly conserved. If the artificially produced property can leave the computational region through advection or diffusion, it need not build up and instability may be avoided. Destruction of the artificially introduced property by artifical sinks is possible, but relying on this mechanism for stabilization is potentially dangerous. Thus, it appears that stability may depend on the flow being simulated and the method used to solve the discretized equations, as well as the discretization method. If this speculation is correct, flows with closed streamlines are more likely to yield numerical instability than flows without recirculation. Similarly, time-like methods (see Sect. VI for a definition) are more likely to be unstable than direct methods.

B. *A Conservative Form of the Navier–Stokes Equations*

Although the equations of fluid dynamics are conservation equations, it is common to talk about conservative and nonconservative forms of these equations. In the former, each term of the equation is the divergence of a vector. The incompressible Navier–Stokes equations, can easily be written in a momentum-conservative form. Unfortunately, as we have seen, demonstration of energy conservation requires something more. This, and the lack of a chain rule for many discrete operators, makes the construction of methods conservative of both momentum and kinetic energy a difficult task.

This difficulty can be circumvented by writing the Navier–Stokes equations in a form that does not require the chain rule for the demonstration of energy conservation. Such a form was provided by Mansour *et al.* [43],

$$\frac{\partial u_i}{\partial t} - \rho u_j \left(\frac{\partial u_i}{\partial x_j} - \frac{\partial u_j}{\partial x_i} \right) = -\frac{\partial}{\partial x_i} \left(p + \frac{1}{2} \rho u_j u_j \right) + \mu \frac{\partial^2 u_i}{\partial x_i \, \partial x_i}. \tag{4.2}$$

For these equations, energy conservation follows from symmetry, and almost any discretization scheme yields both momentum and energy conservation. This form of the equations has been used by the author and his colleagues; the price paid is a modest increase in the cost of a computation.

C. *Staggered versus Standard Grids*

In compressible flow calculations all variables are normally located at the same grid points. This type of grid is also used for incompressible flow simulations but the staggered grid arrangement is more common in this application.

Introduction of the staggered grid is usually credited to Harlow and Welsh [26], but it may have been used earlier. As shown in Fig. 6 for the two-dimensional case, the staggered grid locates the various components of the velocity and the pressure at different grid points. The control volumes used for each variable are also different, as illustrated in Fig. 6. The difference equations for this grid will be given in Section V. For now, we point out that the staggered grid is more accurate than the regular grid and difference schemes that conserve mass (continuity), momentum, and energy are easily constructed for it. Treatment of the boundary conditions is also easier when this coordinate system is used, especially for the pressure.

V. Discretization Methods

A. *Model Equations*

We next consider the discretization of the partial differential equations of fluid mechanics. Finite element methods, although very important, are not covered here. As we saw in Chapter IV, conservation laws need to be taken into account in the

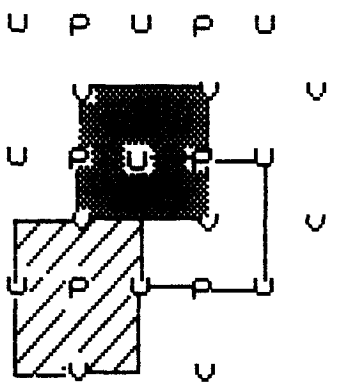

FIG. 6. The location of the velocity components and the pressure in the two-dimensional staggered grid. Also shown are the control volumes for the x momentum ■, y momentum ▨, and continuity □ equations.

construction of discretization schemes. Furthermore, there are important interactions between discretization procedures and the solution techniques which will be covered in Chapter VI.

The partial differential equations of fluid mechanics are nonlinear and too complex to allow straightforward analysis of discretization methods; thus there are essentially no analyses of numerical methods for the full equations. Consequently, we must use model equations as stand-ins for the actual equations. Selection of what is essential in the Navier–Stokes equations is critical; various choices lead to different and, sometimes, contradictory conclusions. In this section, we shall look at some common choices and the analyses based on them.

1. *Unsteady Convection Equations*

Interest centers on high Reynolds-number flows, for which the viscous terms are of little importance. In many flows there are large regions in which the pressure gradient plays no significant role; in these regions, this term can be neglected. The time derivative term is retained in order to study time-like methods (see Sect. VI). The convective terms are nonlinear so, to render them amenable to analysis, we linearize them to obtain

$$\frac{\partial u}{\partial t} + c_j \frac{\partial u}{\partial x_j} = 0, \tag{5.1}$$

i.e., the unsteady convection equation for a passive scalar. This equation is frequently used as a model for study of numerical methods for the Navier–Stokes equations.

The analysis of methods for Eq. (5.1) usually begins by discretizing the convection term and analyzing the stability of the resulting semidiscretized equation by the von Neumann method (Isaacson and Keller [75]). The problem then reduces to

(a) Any method which approximates the derivative of e^{ikx} by a purely imaginary multiple of e^{ikx}, when used in conjunction with Euler's method for time advancement, is unconditionally unstable. This is equivalent to the well-known instability of the Euler method on the imaginary axis. Discretization methods for the convective term which have this property include the well-known central difference approximation:

$$\left.\frac{\partial u}{\partial x}\right|_{x_k} = \frac{u_{k+1} - u_{k-1}}{2\Delta x} \tag{5.2}$$

and spectral methods.

(b) Explicit time-advancement methods which are stable on some part of the imaginary axis for ordinary differential equations yield conditionally stable methods

when applied to the convection equation. For any such method, the stability criterion takes the form:

$$\mathrm{CFL} = \frac{c\,\Delta t}{\Delta x} \leqslant \text{const.} \tag{5.3}$$

This is the Courant–Friedrichs–Lewy condition; CFL is the Courant or CFL number.

(c) An implicit method based on the trapezoid rule (Crank–Nicolson) is neutrally stable when applied to this equation. Neutral stability is a desirable property, and the Crank–Nicolson method is valuable whenever implicit methods can be used without severe cost penalties. However, when applied to nonlinear equations, the method may be unstable or dissipative.

(d) In order to avoid the instability associated with combination of central differencing and the Euler time-advance method (point (a) above), upwind differencing is often used, i.e., instead of Eq. (5.2) we have:

$$\left.\frac{\partial u}{\partial x}\right|_{x_k} = \frac{u_k - u_{k-1}}{\Delta x} \tag{5.4}$$

for $c > 0$. For $c < 0$, $(u_{k+1} - u_k)/\Delta x$ is used to estimate the spatial derivative. The use of the upstream velocity in the difference approximation gives the method its name. The Euler method applied to this equation is conditionally stable, obeying Eq. (5.3) with $\mathrm{CFL} = 1$. The computational molecule for this method also mimics the characterics of Eq. (5.1) better than the molecule for Eq. (5.2).

The stability produced by upwind differencing has made it popular. However, this method is only first-order accurate; in fact, to first order in Δx, it is equivalent to replacing the one-dimensional form of Eq. (5.1) by the modified equation

$$\frac{\partial u}{\partial t} + c\frac{\partial u}{\partial x} = \frac{c\,\Delta x}{2}\frac{\partial^2 u}{\partial x^2}. \tag{5.5}$$

The error has the form of a viscous term with artificial viscosity $\nu_a = c\,\Delta x/2$. In high Reynolds-number flows, the artificial viscosity dominates the true viscosity and the solution produced is much smoother than the correct one.

The results for the unsteady convection equation in two or three dimensions are nearly the same as those for one dimension. Euler's method in conjunction with central differencing is unstable, and upwind differencing is often used to stabilize the method. Each spatial derivative is estimated by a one-sided difference whose direction depends on the sign of the velocity component in that direction. The CFL condition (5.3) becomes

$$\mathrm{CFL} = \left(\frac{|c_1|}{\Delta x_1} + \frac{|c_2|}{\Delta x_2}\right)\Delta t < \text{const.} \tag{5.6}$$

and the extension to three dimensions is fairly obvious.

In two- or three-dimensional problems, the artificial viscosity is anisotropic and depends on the angle between the streamlines and the grid; an expression for it was given by de Vahl Davis and Mallinson (1971):

$$\nu_a = \frac{|c|\,\Delta x_1\,\Delta x_2 \sin 2\theta}{4(\Delta x_1 \cos^3\theta + \Delta x_2 \sin^3\theta)}. \tag{5.7}$$

When the flow is at an angle to the grid, the artificial viscosity tends to reduce velocity differences normal to the streamlines. Equation (5.7) shows that the problem is most severe when the angle is 45°. Alignment of the grid with the flow thus aids the achievement of quality simulations, especially when discretizations with diffusive error are employed. For further details on this point, see Trefethen [62].

2. *Diffusion Equation*

Many flows of engineering interest contain thin shear layers: regions where there are strong gradients of the velodity in the direction normal to the streamlines. In such layers, the dominant mechanisms are the transport of momentum parallel to the streamlines by convection and normal to the streamline by viscosity and/or turbulence. We saw in Section II that the effect of turbulence is represented, in simple models, by an increased effective viscosity. Under the further approximation that the pressure is constant in these layers, the equations become parabolic. An appropriate model equation is then the diffusion equation:

$$c\frac{\partial u}{\partial x_1} = \nu\frac{\partial^2 u}{\partial x_2^2}, \tag{5.8}$$

where, to facilitate analysis, we have linearized the equation by setting the convection velocity to a constant. The time-like variable is x_1/c.

For the diffusion equation, there is no reason to use anything other than a central difference approximation for the diffusion term; if one discretizes only the diffusion term, the result is a stiff set of ordinary differential equations. Numerical solution is best effected with an implicit method; nearly all calculations of thin shear layers in recent years have used some variation of the Crank–Nicolson method. The Keller box method (Cebeci and Keller [70]) can be considered as a variant of Crank–Nicolson. Thus we have

$$\frac{c(u_{i+1,j} - u_{ij})}{\Delta x_1} = \frac{\nu}{2\,\Delta x_2^2}\left[u_{i+1,j+1} - 2u_{i+1,j} + u_{i+1,j-1} + u_{i,j+1} - 2u_{ij} + u_{i,j-1}\right]. \tag{5.9}$$

Explicit methods are unstable unless used with very small step sizes in the time-like direction.

3. *Laplace's Equation*

Another characteristic of incompressible flows is the elliptic nature of the pressure. This is best seen by taking the divergence of the Navier–Stokes equations

(2.1) and using the continuity equation (2.2). One obtains the Poisson equation for the pressure

$$\frac{\partial^2 p}{\partial x_j \partial x_j} = -\rho \frac{\partial u_i}{\partial x_j} \frac{\partial u_j}{\partial x_i} \tag{5.10}$$

which is elliptic. Flows which are dominated by inertial effects, i.e., a balance between the kinetic energy and pressure, have this character. Indeed, incompressible and steady, subsonic, compressible flows are often called elliptic flows. From the numerical point of view, the distinguishing feature of these flows is the existence of upstream influence, i.e., effects that propagate from downstream via the pressure field. These effects dictate that iterative methods of solution be used.

The simplest elliptic equation is Laplace's equation, the homogeneous version of Eq. (5.10). It is worth noting that under restriction to inviscid flow and irrotationality, the velocity becomes the gradient of a potential which satisfies Laplace's equation.

Except in rectangular geometry, Laplace's equation is almost always solved by iterative methods. Good all-purpose iterative methods are SOR and ADI. Some of the solution methods to be described in Section VI are based on these methods, and we shall put off descriptions until then.

4. *Steady Convection–Diffusion Equation*

Another frequently used model is the steady convection–diffusion equation

$$c_j \frac{\partial u}{\partial x_j} = \nu \frac{\partial^2 u}{\partial x_j \partial x_j} \tag{5.11}$$

in which convection is balanced by diffusion in both the streamwise and the normal directions. If one ignores cross-stream diffusion, a one-dimensional convection–diffusion equation, similar to the unsteady convection equation considered in Section 1, is obtained

$$c \frac{\partial u}{\partial x} = \nu \frac{\partial^2 u}{\partial x^2}. \tag{5.12}$$

On the other hand, if streamwise diffusion is ignored, as is reasonable for high Reynolds-number flows, one has the diffusion equation considered in Section 2. Many authors have used Eqs. (5.11) or (5.12) as a model; the results obtained are what one would expect on the basis of the results of the preceding sections.

We shall present those results for the one-dimensional convection–diffusion equation which might not be anticipated from earlier results. When the coefficients are constant, Eq. (5.12) has an exact solution. If this solution is compared to numerical approximations for a single grid cell, one comes to the conclusion that central differencing is preferred for the convection term only when the absolute value of the cell Reynolds number, $\mathrm{Re}_{\mathrm{cell}} = u\,\Delta x/\nu$, is less than two. For larger

values of Re_{cell}, upwind differencing is more faithful to the exact result. This has led to the widespread use of a hybrid differencing sheme based on cell Reynolds number, i.e., central differencing is used if $|Re_{cell}| < 2$ and upwind differencing when $|Re_{cell}| > 2$. Still better matches to the exact single-cell solution may be obtained by using polynomial or exponential fits (Patankar [48]).

Upwind differencing also provides more stability than central difference approximations when certain time-like iterative solution methods are used. Stability has been advanced as an important advantage of upwind differences, but it is important to point out that stability is dependent on the solution method. We shall give arguments below which call the importance of using upwind approximations into question.

Since the upwind-difference approximation is only first-order accurate, a number of schemes which retain the stability of upwind differencing while providing higher-order accuracy have been proposed. Leonard [39] proposed a scheme which uses central differences, but the velocities at the cell boundaries are obtained by third-order asymmetric interpolation using two upstream points and one downstream point. The result is known as the QUICK scheme. For an equally spaced grid in one dimension, it estimates the first derivative by

$$\left.\frac{\partial \phi}{\partial x}\right|_{x_j} \simeq (8\Delta x)^{-1}[3\phi_{j+1} + 3\phi_j - 7\phi_{j-1} + \phi_{j-2}]. \tag{5.13}$$

When applied to the Navier–Stokes equations, this method is second-order accurate, provides smooth solutions in most cases, and has good stability properties of the upwind method. It has been used by a number of authors with good results.

Methods used for the two-dimensional convection–diffusion equation are, in the main, extensions of the one-dimensional methods. As for the unsteady convection equation, the upwind differencing concept is readily extended to two dimensions; the same is true for the QUICK scheme (5.13). An alternative improvement on upwind differencing which has no one-dimensional analog is the skewed upwind difference method suggested by Raithby [51]. In this method, the convective derivative is evaluated by first determining the intersections of the velocity vector through the center of the control volume with the boundaries of the control volume. One estimates the velocity components at these intersections by interpolating from the neighbor grid points and constructs a finite difference formula based on them. Using the notation illustrated in Fig. 7, we find the difference formula

$$\frac{\partial \phi}{\partial x} \simeq (\Delta x)^{-1}[\phi_{ij} - (1-\alpha)\phi_{i-1j} - \alpha\phi_{i-1j-1}]. \tag{5.14}$$

Results obtained with both the QUICK and skewed upwind methods are con-

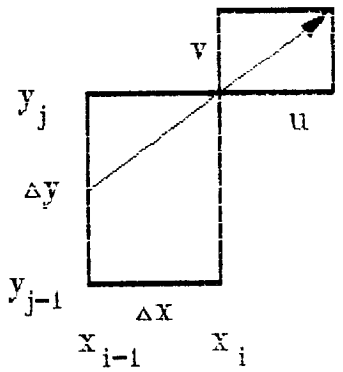

FIG. 7. An illustration of the skewed upwind differencing method of Raithby [51].

Analogous methods have also been developed in conjunction with finite element methods. The concepts employed are quite similar, but adjustments must be made to account for the differences in the construction of the methods. Such methods have been developed by Hemker [27], Hughes, Liu, and Brooks [29], and Chin and Krasny [71].

Methods which introduce a large artificial viscosity in the streamwise direction and a much smaller artificial viscosity in the direction normal to the flow have been introduced in both the finite difference and finite element contexts. The objective is to obtain the stability that artificial viscosity provides without creating unphysical growth of shear layers. However, it is difficult to control the artificial viscosity in each direction completely, so the results obtained using these methods need to be carefully tested for accuracy before they are accepted.

5. *Burgers' Equation*

Burgers' equation

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = \nu\frac{\partial^2 u}{\partial x^2} \tag{5.15}$$

is the closest one-dimensional analog to the Navier–Stokes equations. Consequently, it has been the basis of a number of studies, both analytical and numerical, of methods solving the Navier–Stokes equations. For this application, its principal advantage over the model equations presented above is its nonlinearity; there are also a number of exact solutions of Burgers' equation to which numerical solutions can be compared.

At high Reynolds number, solutions of Bergers' equation develop sharp fronts resembling shock waves. However, as such fronts do not occur in incompressible flows, the study of this equation has had relatively little impact on methods used for incompressible flows.

6. *Boundary and Initial Conditions*

Boundary conditions have not been discussed above; they often receive little attention but, in fact, the quality of a simulation often depends on proper treatment of the boundary conditions. To begin with, it is important to note that the proper boundary conditions to be applied to the Navier–Stokes equations are still, in part, an open question, largely because the proper conditions for the pressure are not known. For the model equations, boundary conditions are well known; however, they do depend on the type of the model equation. All models require that the velocity be provided at any inflow boundary. At a solid surface, for any model which does not include diffusion (viscous effects), only the velocity normal to the boundary must be set to zero; when viscous effects are included, all components of the velocity must be set to zero. At outflow boundaries, no condition is needed if the model equations are hyperbolic or parabolic but some type of artificial condition is needed if the equations are elliptic. Care is needed in extending the results to the Navier–Stokes equations. We shall return to this issue in Section B.

For diffusive equations, the boundary conditions lead to the generation of two kinds of boundary layers. At lateral boundaries, one has the standard hydrodynamic boundary layer whose thickness is proportional to the inverse square root of the Reynolds number; in these layers, the velocity has algebraic behavior. This was pointed out by Levinson [41] and further developed by Chin *et al.* [15].

The boundary conditions at the outflow, on the other hand, may give rise to a thin region with exponential variation of the velocity. Improper treatment of this boundary can cause pollution of computed elliptic flow fields. The problem can be avoided by a combination of a careful choice of the outlet boundary condition and difference approximation (cf. Hedstrom and Osterheld [74]. In particular, a combination of zero gradient (or, what is often the same thing, zero flux) boundary conditions and upwind differencing usually reduces the seriousness of the problem.

B. *Applications to the Navier–Stokes Equations*

1. *Difference Equations*

The preceding section shows the richness of possible behavior in incompressible flows; when additional effects such as compressibility, density differences, or multiphase mixtures are included, the menu becomes even more varied. A single flow may exhibit several of the kinds of behavior described. This makes the choice of a differencing method difficult, and it is hardly surprising that a variety of methods have been proposed and that there are contradictory remarks on the subject in the literature.

There is no unique best method at present. Authors, naturally, make claims for their methods and, in some cases, arguments against competing methods. There are probably a number of methods capable of producing the solution to any given problem. Conversely, there are problems that can make almost any method look

bad. Thus, we cannot recommend a single all-purpose method. Rather, we shall look at some of the most promising candidates.

The most popular method of solving incompressible turbulent flow problems at present is the TEACH method developed by Spalding and colleagues (cf. Caretto *et al.* [12]) and extensions of this method. We shall present a short overview of the basis for this method.

Most methods for incompressible flow, including TEACH, use staggered grids. As noted in Section IV, the staggered grid has a number of advantages. In particular, by locating the variables at the centers of the control volumes, the accuracy of the difference formula is increased. Also, the staggered-grid difference scheme conserves mass, momentum, and kinetic energy in a very natural way. Offsetting these advantages are the difficulties that staggered grids have in using variable-grid spacing and boundary-fitted grids.

We shall give the differenced form of the Navier–Stokes equations for the staggered grid shown in Fig. 6. For this purpose, it is sufficient to consider the equations without a turbulence model; issues that arise when a turbulence model is included will be discussed later. When central differences are used, the difference form of the x-momentum equation is

$$\begin{aligned}
&\frac{\partial u_i}{\partial t} + [(u_i + u_{i+1})^2 - (u_i + u_{i-1})^2]/4\Delta x_1 \\
&\quad + [(u_{ij} + u_{ij+1})(v_{ij} + v_{i-1j}) - (u_{ij} + u_{ij-1})(v_{ij-1} + v_{i-1j-1})]/4\Delta x_2 \\
&= -\rho^{-1}(p_{ij} - p_{i-1j})/\Delta x_1 \\
&\quad + \nu\left[\frac{(u_{i+1j} - 2u_{ij} + u_{i-1})}{\Delta x_1^2} + \frac{(u_{ij} - 2u_{ij} + u_{ij-1})}{\Delta x_2^2}\right].
\end{aligned} \tag{5.16}$$

The difference form of the y-momentum equation is essentially just a rotated form of this equation and will not be given here. The difference form of the continuity equation is

$$\frac{u_{i+1j} - u_{ij}}{\Delta x_1} + \frac{v_{ij} - v_{ij-1}}{\Delta x_2} = 0. \tag{5.17}$$

Any of the difference schemes described in the preceding section can be applied in place of central differencing for the convective terms. The hybrid approach in which the choice between central and upwind differences is governed by the cell Reynolds number was commonly used until recently; however, it is only first-order accurate and has a diffusive error. In the past few years, the deficiencies of that approach (principally the first-order accuracy) have come to be recognized, and a more accurate hybrid approach which uses central differences when the cell Reynolds number is small and either the QUICK or the skewed upwind method for high cell Reynolds number has become more popular. Although individual authors prefer one or the other of the upwind methods, the evidence does not clearly favor either

method. Leschziner [40] showed this and also showed the superiority of these methods over the first-order hybrid method.

The argument for hybrid methods and the upwind difference methods on which they are based rests on two points. The first is the behavior of these methods when applied to the convective-diffusion model equation and the second, experience with a particular solution technique, described in the next section, for which upwind differencing is stable and central differencing is not. However, the hybrid methods may not be appropriate in flows that are dominated by either free shear layer behavior or by pressure forces. For both of these cases, central differencing is more faithful to the physics of the problem at all cell Reynolds numbers than are the upwind schemes. Indeed, good results have been obtained at high Reynolds numbers using central differencing (Kim and Moin [33]); this suggests that the upwind methods may be required more for stability of the solution method than for accuracy. The lesson to be learned here is that differencing schemes and solution methods interact in ways which are not well understood and caution is therefore required in extending results obtained for model problems to the Navier–Stokes equations.

In the absence of a clear best method for all purposes, it is best to adopt a scheme that has been successfully applied to flows similar to the one to be simulated. It is likely that, in the future, methods which use difference schemes adapted to particular regions of a flow will be developed; such schemes are already in use in compressible flows where the distinction depends on the equation type changing from elliptic to hyperbolic as the Mach number crosses unity.

The arguments relating to conservation are similar. A number of authors have found methods which are unstable when conservation is lacking and have generalized their findings by stating that conservation is always required. However, calculations have been made with nonconservative methods that appear to be both stable and accurate.

The difference approximations applied to the partial differential equations representing the turbulence model are essentially the same as those applied to the continuity and momentum equations. In the staggered grid scheme, the turbulent kinetic energy and dissipation are located at the pressure nodes. Derivation of the difference form of the equations follows the lines used for the other equations. The major change introduced by turbulence models is that the additional equations introduced by the models are generally stiff; therefore, more care is needed in solving them.

2. *Boundary Conditions*

The issue of boundary conditions nearly always receives less attention than the differencing of the equations themselves. However, the quality of a simulation may depend very strongly on boundary conditions used. We shall consider only the case of elliptic flows.

Initial conditions, i.e., conditions at the inflow boundary, are part of the problem statement; one would expect them to be easy to specify. However, all

variables—velocity, pressure, and any variables required by the turbulence model—must be specified at the inflow. Many flows are very sensitive to small changes in the initial conditions. Usually this is a consequence of placement of the upstream boundary near a corner. The state of a flow at such a location is difficult to determine, and improper specification of it may lead to large errors in the solution. The best cure for this problem is to place the boundary at a location farther upstream, where the state of the flow is more accurately known. Turbulence-model quantities are difficult to specify accurately, because measurements are often not available for them; and some quantities (such as the dissipation used in two-equation models) are not measurable at all. Incorrect specification of these quantities may result in inaccurate solutions some distance downstream of the inflow boundary.

At outflow boundaries, the situation is not clear; fortunately, errors made at the outflow boundary normally propagate only a few grid points upstream into the flow and are therefore of less consequence than errors made almost anywhere else. However, the placement of the outflow boundary may have an important influence on the results. Most authors prescribe the gradients of all velocity components, the pressure, and turbulence quantities to be zero at outflow boundaries. For time-dependent problems, some authors prefer to set the convective or substantial derivatives to zero at outflow boundaries.

At solid walls, two types of boundary conditions are used. The first is similar to the conditions used for laminar flow, i.e., all velocity components are set to zero at the wall. For the staggered grid, the boundary coincides with the nodes used for the normal velocity, as shown in Fig. 8, and the normal velocity is set to zero at these nodes. For the tangential components of the velocity, the boundary condition must be enforced by extrapolation from the two mesh points closest to the wall. The difference formula for the viscous term is then very inaccurate at the first grid point; the approximation normally used is not even first-order accurate.

The second type of boundary condition is not actually applied at the wall. In turbulent flows, the steepest velocity gradients occur in the region closest to the wall. Very fine nonuniform grids are required for accurate resolution of this region. Typically, the innermost five percent of the flow absorbs half of the total grid points. Furthermore, some turbulence models are not accurate in the vicinity of solid boundaries. For these reasons, it is common practice to omit the innermost region of the flow from the simulation and to apply an artificial boundary condition

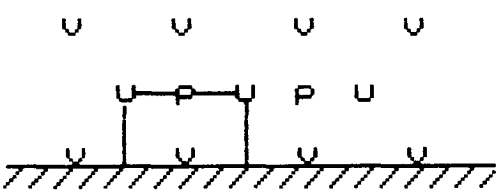

FIG. 8. The treatment of the solid boundaries using the staggered grid.

at a point some distance from the boundary. The artificial boundary condition is generated by solving an approximation to the momentum equation for the region between the actual wall and the first mesh point. This approximate solution provides a relation between the tangential velocity at the first mesh point and its normal derivative.

The boundary conditions to be applied to the pressure at a solid boundary are the subject of some controversy. The issue arises mainly when the pressure is determined from the Poisson equation (5.10); this is usually the case for time-like methods of solutions (see Sect. VI). With staggered grids, it is possible to avoid specifying any boundary conditions for the pressure at the surface. Although acceptable solutions are obtained, this violates a fundamental property of elliptic equations. Since it is the differentiation of the fundamental equations which produces the Poisson equation and thereby causes the need for the boundary condition, Strikwerda [58] suggests avoiding use of this method.

Boundary conditions for the pressure can be derived from either of the momentum equations. However, these lead to different boundary conditions and it is not clear that both sets can be satisfied simultaneously. Indeed, Moser and Moin [76] suggested that the pressure may not be analytic near a solid surface.

Turbulence models are not as well developed for the vicinity of the wall as they are for the freestream: this is a major reason for using the artificial boundary conditions described above. Boundary conditions on turbulence-model quantities are difficult to fix. The turbulent kinetic energy goes to zero at the wall, but, when artificial boundaries are used, the boundary is in a region in which zero-gradient boundary conditions may be more appropriate. Zero-gradient boundary conditions are also applied to the turbulence dissipation.

C. *Spectral Methods*

Spectral methods, although they are quite specialized, are seeing increased application. They are based on Fourier series and its generalizations.

Continuous Fourier series are familiar to nearly everyone. Not quite as well known are the discrete Fourier series and its generalizations; the reader unfamiliar with them is referred to the books by Bracewell [10] and Brigham [11]. In brief, in a discrete Fourier series, a function defined at an evenly spaced set of grid points is represented as a finite series of Fourier (e^{ikx}) functions. The coefficients of the series can themselves be expressed as finite Fourier series with the values of the function at the grid points acting as the coefficients.

Thus, given the values of a function at the mesh points, it is possible to compute its discrete Fourier coefficients. If these are substituted into the discrete Fourier series, the function is reconstructed at the mesh points. More significantly, a discrete Fourier series can be extended to a continuous function by replacing the discrete variable by a continuous one, i.e., discrete Fourier series can be used as an interpolation scheme. The resultant interpolant can then be differentiated, yielding accurate approximations to the derivatives of the function. Indeed, for smooth

functions, the result is more accurate than any finite-difference formula. This makes spectral methods very attractive as differentiators and, therefore, as tools for the solution of differential equations.

The discovery of a fast algorithm for the computation of discrete Fourier transforms (Cooley and Tukey [18]) has reduced the cost of spectral methods and thereby made them practical.

Standard Fourier series are most valuable when periodic boundary conditions can be applied. When nonperiodic boundary conditions are required, accuracy is lost. To achieve the potential of the spectral approach in these cases, one must use other functions. For domains with solid boundaries, Chebychev polynomials have been used by a number of authors, as have other sets of orthogonal polynomials. Many of these function sets do not permit use of the fast-Fourier-transform algorithm, thus increasing the cost. One can usually use orthogonal functions in one coordinate direction with only a modest increase in the overall cost of a calculation. For a review of spectral methods, see Gottlieb and Orszag [25].

VI. Solution Techniques

A. *Classification of Solution Methods*

After a turbulence model has been put in place and a finite difference approximation chosen, one needs to select a method of solving the resulting equations. In part, the choice depends on the nature of the problem. For unsteady problems, one clearly needs a time-accurate method. For stationary problems, on the other hand, a number of strategies are available, including relaxation procedures based on unsteady methods. In this section, we shall be primarily concerned with methods for the stationary case; methods for unsteady problems will be considered primarily for their application to the steady case.

The system of algebraic equations resulting from finite differencing the steady Navier–Stokes equations is large, nonlinear, and sparse. For such systems, iterative techniques are the only recourse. For nearly all techniques, the number of iterations required to obtain a converged result is only mildly sensitive to the quality of the initial guess, so there is no point in putting effort into the construction of an accurate initial field. Usually, some simple initial condition, such as a uniform flow, is assumed.

Our interest centers on solution techniques for steady elliptic flows. In such flows, the velocity in the principal flow direction may change sign somewhere in the flow; such flows are also referred to as recirculating or separated flows. For flows in which velocity reversal does not occur, e.g., attached boundary layers and free shear layers, the equations can often be approximated by parabolic ones, and marching methods of solution are then available and preferable; we shall not consider methods for these problems here; see Patankar [48] or Cebeci and Bradshaw [13] for these.

For elliptic problems, three major solution strategies are employed. These are:

(a) One can consider the steady solution as the infinite time limit of an unsteady problem. As the time history is then merely a route to be steady state, time accuracy is irrelevant and one should use the largest possible time step. We call these time-like methods.

(b) Excellent methods have been developed for solving the compressible Navier–Stokes equations. To apply these to the incompressible equations, one can modify the latter to make them resemble the compressible equations. The terms added to the equations must, of course, disappear in the steady limit. These are called artificial compressibility methods.

(c) One can regard the steady-flow, finite-difference equations as a large set of nonlinear algebraic equations and solve them by some standard technique for such systems. We call these fully coupled methods. They, too, are iterative.

These categories of methods will be reviewed in the remainder of this chapter.

B. *Time-like Methods*

1. *Basic Strategy*

A familiar strategy for solving stationary problems involving partial differential equations is to run a method appropriate to unsteady problems until a steady solution is found. The primary difference between a truly unsready problem and a relaxation method for a steady problem is that in the latter case only the long-time solution matters. As the time history need not be accurate, one may use a much larger time step than in the unsteady problem.

The principal obstacle to solution of the unsteady, incompressible, Navier–Stokes equations is the lack of a time derivarive in the continuity equation; the lack of an explicit equation governing the pressure is another manifestation of the problem. To remedy this situation, it is common to use the Poisson equation (5.10) for the pressure in place of the continuity equation.

For time-accurate solutions, accuracy demands small time steps, so there is little reason to use implicit methods; explicit methods are preferred. In relaxation methods for steady problems, the larger time step permitted by implicit methods becomes important. However, fully implicit methods for nonlinear equations are costly, so, in practice, linearized implicit methods are employed.

The basic method of time accurately solving the equations was given by Chorin [16] and is applicable to either explicit or implicit methods. For simplicity, we shall demonstrate the method with the explicit Euler method, but it is easily adapted to a variety of other time-advance methods. It updates the velocity via the momentum equations

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} = \frac{\delta}{\delta x_j} u_i^n u_j^n - \frac{1}{\rho} \frac{\delta p^n}{\delta x_i} + \nu \frac{\delta}{\delta x_j} \frac{\delta}{\delta x_j} u_i^n, \tag{6.1}$$

where $\delta/\delta x$ represents a finite difference operator. The resulting velocity field u_i^{n+1} is not divergence-free. However, taking the divergence of Eq. (6.1) (the numerical divergence operator must be the one used in differencing the continuity equation), assuming that the velocity field at t_n is divergence-free, and requiring that the divergence of the velocity be zero at t_{n+1}, we arrive at a numerical form of the Poisson equation which we write as

$$\frac{\delta}{\delta x_i}\frac{\delta}{\delta x_i}p^n = -\rho\frac{\delta}{\delta x_i}\frac{\delta}{\delta x_j}u_i^n u_j^n \tag{6.2}$$

so that, if the pressure p^n satisfies Eq. (6.2), the velocity field satisfies continuity at t_{n+1}. It is important to note that the difference form of the Poisson equation cannot be chosen independently of the numerical approximations made in the momentum and continuity equations; for some difference schemes, this requires application of an inaccurate difference scheme to the Poisson equation.

This method inherits the difficulties associated with the pressure boundary conditions noted in Section V.B.2. Also, except in simple geometries for which fast, direct techniques are available, the Poisson equation must be solved by an iterative method; this may make the method as a whole slower. The method described in the following section overcomes some of these difficulties.

2. *The* SIMPLE *Method*

The arguments made in the preceding section suggest solving steady-state problems by a combination of linearized implicit time advancement of the velocity and solution of the Poisson equation for the pressure. The most expensive part of such a calculation would be iterative calculation of the pressure. However, there is no need to compute the pressure accurately until the steady state is reached, so a better strategy is to iterate the Poisson equation no more than a few times or, perhaps, only once, at each step.

Some of the best methods of this type are the SIMPLE algorithm developed by Caretto *et al.* [12] and extensions of it. Although the algorithm is designed for the steady-state equations, it is a kind of relaxation algorithm based on a fractional step time-like method. We shall illustrate the method with the central differences for the spatial derivatives, but any of the other approximations described earlier could be used.

The method begins with guesses at the pressure and velocity fields, p^n and u_i^n. A new velocity field is computed using a linearized implicit approximation to the staggered grid finite difference momentum equations. At mesh point j, k for the x-momentum equation, one has:

$$u_{j,k}^* - u_{j,k}^n = \Delta t\left[-\frac{\delta}{\delta x}2u^*u^n - \frac{\delta}{\delta y}u^*v^n - \frac{\delta}{\delta y}u^n v^* - \frac{1}{\rho}\frac{\delta}{\delta x}p^n + \nu\left(\frac{\delta^2 u^*}{\delta x^2} + \frac{\delta^2 u^n}{\delta y^2}\right)\right]. \tag{6.3}$$

These equations are solved by a block or line relaxation (line-by-line) method. On each line in one direction, one solves these equations using a block tridiagonal algorithm. For efficiency reasons, only one iteration (one sweep) is carried out. The result is a first approximation to the velocity at the new time step, u^*.

To update the pressure, the momentum equations are first linearized about p^n and u^*. These equations are further simplified by eliminating all of the contributions from the velocities at the neighboring points to give an equation of the form

$$\Delta t(\ )u'_{j,k} = \frac{\delta p'}{\delta x}. \tag{6.4}$$

The empty brackets indicate that the contents depend on the difference scheme used at the particular point. When the divergence of these equations is taken, the result is an approximate Poisson equation for a pressure correction, which is also given a single line iteration. The resulting pressure correction is then used to update the pressure, and the velocity is updated using Eq. (6.4). At this point, an iteration is complete and we are ready to begin another one.

Some features of this method reflect the empiricism employed in its construction. It is difficult to apply the methods of numerical analysis to study the behavior of this method (or most other methods for these equations). Consequently, knowledge of the behavior of this method is based on experience in computing flows with it. The following behavior of this method has been observed:

(1) It requires an upwind difference approximation to converge at high Reynolds numbers. For discussion of this point, see Section V.A.

(2) It is capable of handling a wide range of flows without difficulty.

(3) Its rate of convergence is nearly independent of Reynolds number. Typically, 200–400 iterations are required.

The last result may be a consequence of the use of upwind differencing. As shown in Section V, upwind differencing introduces artificial viscosity into a simulation, thus causing the effective Reynolds number to increase much more slowly than the actual Reynolds number. Some of the author's students (Avva *et al.* [67]) have recently shown that, when central differences are used, the number of iterations required for convergence increases more rapidly with Reynolds number than for upwind differences.

The number of iterations required for convergence of these methods is greater than the number of iterations required by similar methods applied to simple, elliptic, partial differential equations such as Laplace's equation. A number of reasons for this have been given; these include the asymmetry of the matrices, the lack of diagonal dominance in the matrices, nonlinearity, and the mixed nature of the equations, i.e., they are not entirely elliptic. While all of these are certainly contributors to the problem, none is a completely satisfactory explanation. Our ability to analyze methods of solving the equations of fluid mechanics is very limited at the

present time. As a result, most of what is known about these methods is based on extensions of and analogies with results for model equations. As we have seen in Section V, this kind of reasoning is quite risky.

A number of modifications/extensions of the SIMPLE method have been made. Some, such as the SIMPLER method of Patankar [48] offer faster convergence at the expense of higher computation cost per iteration. The overall savings are not large.

3. *Fractional Step Methods*

In fractional step methods, a differential equation is treated as a composite of simpler equations. These methods were originally developed in the Soviet Union (cf. Marchuk [44]). Their chief advantage is the great flexibility they offer in the selection of algorithms; they bear a strong resemblance to splitting (ADI-like) methods.

To apply the fractional step method to the incompressible Navier–Stokes equations, one must regard the latter as a composite of convective, pressure gradient, and viscous equations. Each of these may be advanced in time independently or in combination with the others. Consequently, a great variety of methods is possible. As an example, we shall describe the method of Kim and Moin [33]; as it is primarily designed and used as a time-accurate method, we shall be brief.

The method was developed because, for wall-bounded turbulent flows in which the boundary conditions are applied at the actual surface, accuracy demands small grids in the normal direction near the wall. Some of the viscous terms must then be treated implicitly in order to avoid severe restrictions on the time step. Efficiency requires use of a splitting method. Kim and Moin's method consists of three fractional steps. In the first step, provisional velocities are computed by treating the convection terms explicitly and viscous diffusion in the x_1 direction implicitly. The second step treats the x_2 direction diffusion implicitly and the convection terms explicitly. The velocity so obtained does not satisfy continuity, so the pressure is constructed in a way that removes the divergence of the velocity; this requires a Poisson equation and is similar to the method of Section A.

We end this section with a few remarks specific to computations containing turbulence models. Introduction of a turbulence model increases the number of equations that need to be solved and, therefore, the cost and storage requirements. Moreover the equations for the $k-\varepsilon$ model are stiff; as a result, calculations containing turbulence models often require smaller underrelaxation factors for convergence, and the number of iterations required increases.

C. *Artificial Compressibility Methods*

Ability to predict transonic flows is essential to the design of aircraft and engines. [illegible] the development of methods for computing compressible flows. Consequently, the state of the art is more advanced for computation of compressible flows than for incompressible flows. Some particularly important advances have been the develop-

ment of a fast, explicit algorithm by MacCormack [42] and the development of fast, split, implicit (ADI-like) methods by Beam and Warming [8] and Briley and McDonald [68]. Also, as noted earlier, most of the methods for handling complex geometry were first developed for compressible flows.

It is natural to attempt to apply methods developed for compressible flow to the incompressible case. The first such method was presented by Chorin [16]. Improvements, mainly in the direction of using newer compressible flow methods, have been made by Steger and Kutler [57] and by Chang and Kwak [14].

The basis of these methods is the introduction of an unphysical time-derivative term into the continuity equation. If time accuracy is to be maintained, the added term must have a small coefficient. On the other hand, if the method is used only to find steady-state solutions, the added term has no effect on the converged result and its coefficient is of little import.

As the density is constant in incompressible flows and the pressure is the variable for which there is no explicit evolution equation, it is natural to add to the continuity equation a term proportional to the time derivative of the pressure. Thus, the authors referenced above use

$$\frac{1}{\beta}\frac{\partial p}{\partial t}+\frac{\partial}{\partial x_j}u_j=0, \tag{6.5}$$

where β^{-1} is the artificial compressibility and must be large if Eq. (6.5) is to approximate the incompressible equation. This equation is not be solved together with the momentum equations (2.1); they form a set which is hyperbolic in character and therefore solvable with methods used for the compressible equations. Chang and Kwak [14] showed that, for the one-dimensional form of these equations, the effective sound speed is

$$c=(u^2+\beta)^{1/2}. \tag{6.6}$$

The effective Mach number thus is always less than unity. Waves travel at speeds of $u\pm c$. As the two wave speeds differ considerably in magnitude, especially if $\beta\gg u^2$, the problem is stiff and implicit methods of solution are required.

Chang and Kwak used the Beam–Warming [8] method for solving these equations. In this method, the equations are time-differenced using the trapezoid rule (Crank–Nicolson), linearized, and approximately factored in the ADI manner. One then has to solve a block-tridiagonal system on each line of the grid. These authors offered the following criterion for choosing the compressibility parameter:

$$\beta\gg\left(1+\frac{4}{\mathrm{Re}_L}\right)^2-1 \tag{6.7}$$

where L is the length of the computational region. They found that with parameters satisfying this criterion, the method converged in approximately 200 iterations; the number of iterations is not very sensitive to the choice of the compressibility

parameter. This is comparable to the number of iterations required by the time-like methods of the preceding section.

D. *Fully Coupled Methods*

As noted above, iterative methods which work well for simple types of partial differential equations work less well for solving the equations of fluid dynamics, both compressible and incompressible. The convergence of iterative methods is slow, and analysis of them is difficult. Reasons were given earlier; the major problem seems to be that the matrices do not allow the well-established iterative methods for simpler equations to function at their best.

An alternative to these methods is to regard the finite-differenced incompressible Navier–Stokes and continuity equations as a large set of nonlinear equations to be solved as such. One can apply iterative methods such as Newton–Raphson to this system. Since these methods typically converge in a small number of iterations, this approach is very attractive.

The basis of the method is well known. The system of finite-difference equations can be written formally as:

$$f_j(\mathbf{u}) = 0, \qquad j = 1, 2, \ldots, N, \tag{6.8}$$

where the vector $\mathbf{u}$ represents the variables at the mesh points. In the arrangement usually preferred, $\mathbf{u} = (u_{1,1}, v_{1,1}, p_{1,1}, u_{1,2}, \ldots)$. These equations are linearized about some initial guess or previous iterate of the solution

$$f_j \simeq f_j(\mathbf{u}^0) + (u_k - u_k^0) \frac{\partial f_j}{\partial u_k}(\mathbf{u}^0). \tag{6.9}$$

The result is a very large but sparse system of linear algebraic equations; in two dimensions, the size of the matrix is $3N \times 3N$, where N is the number of mesh points used.

A number of points should be noted. First, the method is similar to the time-like methods of the preceding sections. Second, the matrix is so large it will not fit in the central memory of most computers, even for moderate-sized problems in two dimensions, unless some special storage scheme that takes advantage of its sparseness is used. Third, it is possible to solve the linearized problem with iterative methods; indeed, for large matrices with banded structure, such as the one in Eq. (6.9), iterative methods have been the methods of choice.

However, Vanka and Leaf [64] poiinted out that one of the major difficulties with methods described in the preceding two sections is the poor behavior of iterative methods for the matrices that arise in fluid-dynamical applications. Although they did not try iterative methods, they suggest that direct solvers are a better choice. To avoid dealing with a large matrix, a sparse matrix solver was chosen. An alternative, suggested by Vanka [65] is to use a decomposition method.

Experience with these methods is limited to just a few cases at the present time. However, for those cases, Vanka reports that the method converges in the small number of iterations expected of the Newton–Raphson method. The use of direct solution methods makes the cost of an iteration much larger than in time-like methods, but overall computation times of one fifth of those required by SIMPLE have been reported. Thus, the method seems promising; further experience with it is required before it can be promoted as the preferred method.

E. *Special Tools*

Before closing this chapter, we should make note of methods that promise to become very important in the future. These include multigrid methods, domain-decomposition methods, and algorithms for parallel computers. Although our discussion of each will be brief (mainly, because these methods are under intensive development at present), the potential of these methods should not be underestimated.

1. *Multigrid Methods*

One reason why many iterative methods converge slowly is that information propagates slowly; in explicit methods, signals may travel only one mesh point per iteration. As a result, most iterative methods tend to damp out the rapidly oscillating (high wavenumber) components of the error relatively quickly; reducing the slowly varying part of the error requires many iterations.

To improve the rate of convergence, the multigrid method uses grids of various coarseness. First, a few iterations are performed on the finest grid to remove most of the high wavenumber error. The solution is then restricted to a coarser grid (usually the grid size is doubled) on which a few iterations are performed to remove low-wavenumber error components. The solution is then interpolated back onto the fine grid, completing one cycle; the entire procedure is repeated until convergence is achieved. Naturally, more than two grid levels can be employed.

Usually only a few cycles are required to achieve convergence. Together with the low cost of iterations on the coarser grids (in two dimensions, each iteration at a given level costs $\frac{1}{4}$ of the cost on the next finer level), this results in a considerable reduction of the total cost.

2. *Domain Decomposition*

One means of overcoming the difficulty with complex geometries, noted in Section 3, is domain decomposition. In this method, the solution domain is divided into geometrically simple subregions which may or may not overlap. Each subregion is easily mapped into a rectangle, and the problem is thereby reduced to one on a set of rectangles.

One of the standard solution methods described above is applied to each subregion successively. Where the subregions touch or overlap, boundary conditions are taken from the latest solution on the neighboring regions.

This method is a promising approach to dealing with problems in complex geometries and is well matched to application on parallel computers (see below).

3. *Parallel Computers*

In the past, computer speeds were increased by the application of newer technologies and by decreasing the size of the circuits. Although there is much still to be gained by improvements of these types, some fundamental limits are beginning to be reached. Also, the low price of microprocessor chips makes the possibility of using multiple processors attractive; numbers of processors in the thousands have been considered.

The difficulty with parallel computers is that many algorithms in common use (e.g., tridiagonal matrix solvers and fast Fourier transforms) are inherently serial and cannot make the most effective use of parallel machines. Thus there is a need for the development of new algorithms for parallel machines.

This chapter has presented three types of methods for solving the discretized, incompressible Navier–Stokes equations and some approaches that promise to provide significant imporvements in computation time in the near future. Of the methods presented, the time-like methods, especially those based on alternating velocity and pressure updates, are the most advanced and most commonly used; they are reliable and robust, but most of them require upwind differencing to converge and are relatively slow. Artificial compressibility methods are next in both state of development and popularity; they allow access to the large body of work for compressible flows but, at the present time, seem to be no faster than methods of the first category. Fully coupled methods offer the possibility of faster computation times, but experience with them is limited and considerable work will need to be done before they see wide use.

VII. Other Approaches to Turbulent Flow Computation

The methods described above may be called full-field methods; they compute the velocity everywhere in the flow by a single method. These methods have a number of desirable features, the most important of which is the ability to simulate a variety

When the velocity field is largely determined by small but intense concentrations of vorticity, modeling the flow by following the vorticity field rather than the velocity field is a sound approach. This makes it possible to concentrate the computational resources in a limited spatial domain, thereby achieving a much more accurate calculation for a given cost. A number of methods based on following the vorticity field have been developed. These include field methods which solve the partial differential equations for the vorticity (vorticity field methods), others which compute interactions between vortex elements directly (vortex tracking methods), and hybrids between the two (vortex-in-cell methods); within each category there are a number of variations.

Vortex-tracking methods are completely Lagrangian and grid-independent. The coordinates of a vortex may be completely arbitrary, contributing to the accuracy of the method. Interactions between the vortices are computed using the Biot–Savart law. The major source of inaccuracy in these methods arises from the representation of the vorticity by discrete elements rather than continuous distributions. The greatest disadvantage of vortex-tracking methods is that the cost scales with the square of the number of vortex elements and grows much more rapidly than for other methods. Also, for three-dimensional flows, the geometry of the elements can become complex, increasing both the difficulty of programming and the cost.

Vortex methods are ideally suited to vortex flows and compete very well with other methods in turbulent flows in which the turbulence is confined to thin regions. Free shear flows and some boundary layers are examples of this kind of flow. They have also been used, with more limited success, in flows with widely distributed turbulence.

As with other subjects touched upon in this paper, a thorough discussion of vortex methods would require a separate paper. The reader interested in details is referred to the review papers of Leonard [37, 38].

C. *Full and Large Eddy Simulations*

Supercomputers have made it possible to solve the Navier–Stokes equations for the three-dimensional, time-dependent motions that are the essence of a turbulent flow. Two different approaches have been employed. Full simulations attempt to resolve all of the scales of the motion and apply only to simple low Reynolds number flows. Large eddy simulation models the small-scale motions (in much the same way as the models described in Section II) while computing the large scales explicitly and can be applied to a broader range of flows and higher Reynolds numbers.

The cost of these simuations is high; a single simulation may require anywhere from a few minutes to many hours on the largest computers. Consequently, the community of users of these methods is small, but is growing rapidly. Interest in these methods derives from their ability to produce information about turbulent flows that is complementary to experimental data and the ability to control con-

ditions in a way not available to most experimentalists. A typical simulation yields all three velocity components and the pressure on a mesh of $64 \times 64 \times 64$ points for several hundred to a few thousand time steps, representing several seconds of physical time. The time span is much shorter than most laboratory records, but the wealth of spatial resolution is invaluable.

The nature of these methods dictates that they be used primarily to study the physical mechanisms of turbulence and to develop correlations and models of the kind described in Section II. They are proving very valuable in this connection. Again, space does not permit covering this area in detail here; for recent reviews, see Ferziger [24] or Rogallo and Moin [55].

VIII. State of the Art

All of the areas covered in this paper are the subjects of intensive research. A paper on this subject written a few years from now will probably be quite different. In this section, we shall try to assess what can be done today and what developments might be expected in the future. Some of the opinions expressed are controversial and are those of the author rather than a consensus of a large number of people; the latter probably does not exist today.

First, the importance of the advances in hardware in the coming years cannot be overemphasized. VLSI technology will improve capability by a large margin, both by making computers much cheaper and by allowing the production of supercomputers with much larger memories. This will contribute to rapid continued growth of the importance of computers in fluid mechanics in general and in turbulent flow computation in particular.

Turbulence modeling is the area in which progress seems to be the slowest. It seems likely that a variety of models will continue to be used and that models based on partial differential equations for turbulence quantities will increase in importance and popularity. However, evidence seems to be accumulating that present models are not accurate enough for many engineering purposes and that models of the future will need to be tuned to the physics of particular regions of the flow, i.e., they will be zonal in nature.

Methods used to generate grids, especially for three-dimensional flows, need considerable further development. In this area, finite elements are currently ahead of finite difference methods. It seems likely that finite difference methods will adopt some of the concepts used for finite elements, so that methods of the future may be hybrids. For example, methods in which finite-element ideas are used to map the considered domain into a set of rectangles which can be treated with finite differences that allow fast solution techniques may come to the fore.

A persistent problem in the computation of turbulent flows has been the lack of a method for estimating numerical errors. This has led to blaming turbulence models for errors that are caused by inadequate grid resolution, finite difference approximations, or incomplete convergence. Techniques which detect these errors

and correct them by refining the grid where necessary are known as adaptive grid methods and are likely to become important in the future.

The optimum choice of finite difference approximation is not settled at present. As we have seen, the dominant physical phenomena in a particular region of a flow govern the appropriate choice of a difference approximation. Zone-dependent differencing in which the method is tailored to the nature of each particular region may be a method of the future. Automation of grid-generation procedures is very desirable but will require some time to develop. The trend toward use of methods of at least second-order accuracy will probably also continue.

Present solution techniques leave much room for improvement. The directions are not entirely clear, but implicit or direct methods offer the best possibilities. Techniques in which different time steps are used in different zones of the flow or, perhaps, are applied to different terms in the equation, which have recently shown promise in compressible flows, may be developed.

ACKNOWLEDGMENTS

The author is indebted to people, too numerous to mention, for pointing out various references, methods, and other useful information. Despite their help, I am sure that a number of important works have not been cited, and the author apologizes to the authors for the oversight. The author is particularly indebted to Professor J. Oliger for valuable discussions and to a number of people who read and commented on earlier drafts of this paper. The latter include (in alphabetical order): R. Avva, S. C. Caruso, Dr. R. C. Y. Chin, Professor S. J. Kline, Dr. M. Y. Hussaini, Professor R. L. Street, and Professor M. Wolfshtein.

REFERENCES

1. D. A. ANDERSON, *Advances in Grid Generation, FED-5* (ASME, 1983).
2. A. ARAKAWA, *J. Phys.* **1**, 119 (1966).
3. R. AVVA, Ph. D. thesis, Department of Mechanical Engineering, Stanford University, in preparation.
4. A. J. BAKER, *Finite Element Computational Fluid Mechanics* (McGraw-Hill, New York, 1983).
5. B. R. BALDWIN AND H. LOMAX, "Thin Layer Approximation and Algebraic Model for Separated Turbulent Flows," AIAA Paper No. 78-257, 1978.
6. J. BARDINA, A. A. LYRIO, J. H. FERZIGER, S. J. KLINE, AND J. P. JOHNSTON, *J. Fluids Eng.* **103**, 315 (1981).
7. R. BEAM, R. W. WARMING, AND M. BERGER, Report STAN-CS-82-924, Department of Computer Science, Stanford Univ., Stanford, CA, 1982.
8. R. M. BEAM AND R. F. WARMING, *J. Comput. Phys.* **22**, 87 (1976).
9. M. BERGER, Report STAN-CS-82-924, Department of Computer Science, Stanford Univ., Stanford, CA, 1982.
10. R. BRACEWELL, *The Fourier Transform and Its Applications* (McGraw-Hill, New York, 1978).
11. E. O. BRIGHAM, *The Fast Fourier Transform* (Prentice-Hall, Englewood Cliffs, NJ, 1974).
12. L. S. CARETTO, A. D. GOSMAN, S. V. PATANKAR, AND D. B. SPALDING, *in Proceeding, Third Intl. Conf. Num. Meth. Fluid Dyn., Paris, 1972.*
13. T. CEBECI AND P. BRADSHAW, *Momentum Transfer in Boundary Layers* (McGraw-Hill, New York, 1977).
14. J. L. C. CHANG AND D. KWAK, AIAA Paper No. 84-0252, 1984.

15. R. C. Y. Chin, J. A. Manteuffel, and J. de Pillis, Report No. UCRL-85941, Lawrence Livermore Natl. Lab., 1983.
16. A. J. Chorin, *J. Comput. Phys.* **2**, 12 (1967).
17. A. J. Chorin, *Math. Comput.* **22**, 745 (1968).
18. J. W. Cooley and J. W. Tukey, *Math. Comput.* **19**, 297 (1965).
19. G. de Vahl Davis and G. D. Mallinson, U. New So. Wales Sch. Mech. Ind. Engr. Report No. 1972|FM/1, 1972.
20. H. A. Dwyer, R. J. Kee, and B. R. Sanders, *AIAA J.* **18**, 1205 (1980).
21. P. R. Eiseman and R. E. Smith, NASA Conf. Publ. 2166, NASA, 1980.
22. S. C. Eisenstat, M. C. Gursky, M. H. Schultz, and A. H. Sherman, Research Report No. 114, Yale University, 1983.
23. J. H. Ferziger, J. Bardina, and G. Allen, *in Proceedings, AFOSR-HTTM-Stanford Conference on Complex Turbulent Flows*, edited by Kline *et al.* (Stanford Univ., Stanford, CA, 1981), Vol. 2.
24. J. H. Ferziger, "Higher Level Simulations of Turbulent Flow," *Computational Methods for Turbulent, Transonic, and Viscous Flows*, edited by J.-A. Essers (Hemisphere, 1983).
25. D. M. Gottlieb and S. A. Orszag, SIAM *J. Sci. Stat. Comput.* **19** (1977).
26. F. H. Harlow and J. E. Welsh, *Phys. Fluids* **8**, 2182 (1965).
27. P. W. Hemker, Dissertation, Univ. of Amsterdam, 1977.
28. K. A. Hessenius and M. M. Rai, "Applications of a Conservative Scheme to Transient and Geometrically Complex Problems," AIAA Paper No. 84-1532, 1984.
29. T. J. R. Hughes, W. K. Liu, and A. Brooks, *J. Comput. Phys.* **30**, 1 (1979).
30. W. P. Jones and B. E. Launder, *Int. J. Heat Mass Transfer* **15**, 301 (1972).
31. W. M. Kays and M. E. Crawford, *Convective Heat and Mass Transfer*, (McGraw-Hill, New York, 1978), 2nd ed.
32. H. B. Keller, "A New Difference Scheme for Parabolic Problems," *Numerical Solution of Partial Differential Equations*, edited by J. Bramble (Academic Press, New York, 1970), Vol. II.
33. J. Kim and P. Moin, *J. Comput. Phys.* **35**, 31 (1980).
34. S. J. Kline, 3 vols., *in Proceedings, AFOSR-HTTM-Stanford Conference on Complex Turbulent Flows* (Stanford Univ., Stanford, CA, 1981).
35. S. J. Kline, G. M. Lilley, and B. J. Cantwell, *in Prodeedings of the 1980–1981 AFOSR-HTTM-Stanford Conference on Complex Turbulent Flows* (Department of Mechanical Engineering, Stanford Univ., Stanford, CA, 1981).
36. M. J. Lee and W. C. Reynolds, Report No. TF-22, Department of Mechanical Engineering, Stanford Univ., 1985.
37. A. Leonard, *J. Comput. Phys.* **37**, 289 (1980).
38. A. Leonard, *Ann. Rev. Fluid Mech.* **17**, 523 (1985).
39. B. P. Leonard, *in Proceedings, Second GAMM Conf. on Num. Meth. in Fluid Mech., Cologne, 1977.*
40. M. A. Leschziner, *Comput. Methods Appl. Mech. Eng.* **23**, 293 (1980).
41. N. Levinson, *Ann. Math.* **51**, 428 (1950).
42. R. W. MacCormack, AIAA Paper No. 69-354, 1969.
43. N. N. Mansour, P. Moin, W. C. Reynolds, and J. H. Ferziger, *in Proceeding, First Symp. Turb. Shear Flows* (Penn. St. Univ. Press, University Park, 1977).
44. G. M. Marchuk, *Methods of Numerical Mathematics* (Springer, New York, 1975).
45. C. W. Mastin and H. V. McConnaughey, AIAA Paper No. 84-1611, 1984 (unpublished).
46. R. D. Moser, Dissertation, Report No. TF-22, Department of Mechanical Engineering, Stanford Univ., Stanford, CA, 1984.
47. D. Naot, A. Shavit, and M. Wolfshtein, *Warme- und Stoffübertragung* **7**, 151 (1974).
48. S. V. Patankar, *Numerical Heat Transfer and Fluid Flow* (McGraw-Hill, New York, 1980).
49. N. R. Patel and J. F. Thompson, AIAA Paper No. 84-1534, 1984 (unpublished).
50. M. M. Rai, AIAA Paper No. 84-0164, 1984 (unpublished).
51. G. D. Raithby, *Comput. Methods Appl. Mech. Eng.* **9**, 75 (1976).
52. W. C. Reynolds, *Ann. Rev. Fluid Mech.* **8**, 183 (1976).

53. W. RODI, *Z. Angew. Math. Mech.* **T219** (1976).
54. W. RODI, *Turbulence Models and their Applications in Hydraulics* (Intl. Assn. for Hydraulics Research, Delft, 1980).
55. R. S. ROGALLO AND P. MOIN, *Ann. Rev. Fluid Mech.* **16**, 99 (1984).
56. R. L. SORENSON AND J. L. STEGER, "Use of Partial Differential Equations to Generate Body-Fitted Coordinates in Numerical Grid-Generation Techniques," NASA Conf. Publ. No. 2166, NASA, 1980.
57. J. L. STEGER AND P. KUTLER, *AIAA J.* **15**, 581 (1977).
58. J. C. STRIKWERDA, MRC Techn Summary Report No. 2584, Math. Res. Ctr., Univ. of Wisc., 1983.
59. H. TENNEKES AND J. L. LUMLEY, *A First Course in Turbulence* (MIT Press, Cambridge, MA, 1972).
60. J. F. THOMPSON AND C. W. MASTIN, *in* "Numerical Grid Generattion Techniques," NASA Conf. Publ. No. 2166, NASA, 1980.
61. J. F. THOMPSON, F. THAMES, AND C. W. MASTIN, *J. Comput. Phys.* **15**, 299 (1974).
62. L. N. TREFETHEN, *SIAM Rev.* **24**, 113 (1982).
63. K. L. TZUOO, J. H. FERZIGER, AND S. J. KLINE, Internal report, Dept. of Mech. Engr., Stanford Univ., 1984.
64. S. P. VANKA AND G. K. LEAF, Report ANL-83-73, Argonne Nat'l Lab., 1983.
65. S. P. VANKA, Report ANL-83-87, Argonne Nat'l Lab., 1983.
66. C. T. WU, J. H. FERZIGER, AND D. R. CHAPMAN, Report TF-21, Dept. of Mech. Engrg., Stanford Univ., 1985.
67. R. AVVA, S. J. KLINE, AND J. H. FERZIGER, Report, Dept. of Mech. Engr., Stanford University, in preparation.
68. R. BRILEY AND H. MCDONALD, in *Proceedings, 4th Int. Conf. Num. Meth. Fluid Dynamics* (Springer, New York, 1975).
69. G. L. BROWN AND A. ROSHKO, *J. Fluid Mech.* **64**, 775 (1974).
70. T. CEBECI AND H. B. KELLER, in *Proceedings, 3th Int. Conf. Num. Meth. Fluid Dynamics* (Springer, New York, 1973).
71. R. C. Y. CHIN AND R. KRASNY, *SIAM J. Sci. Stat. Comput.* **4**, 229 (1983).
72. R. GLOWINSKI, Q. V. DINH, AND J. PERIAUX, *Comp. Meth. Appl. Mech. Engr.*, in press.
73. F. H. HARLOW AND C. W. HIRT, Los Alamos Sci. Lab. Rep. LA 4086, 1969.
74. G. W. HEDSTROM AND A. OSTERHELD, *J. Comput. Phys.* **37**, 399 (1980).
75. E. ISAACSON AND H. B. KELLER, *Analysis of Numerical Methods*, (Wiley, New York, 1966).
76. R. D. MOSER AND P. MOIN, Rep. TF-20, Dept. of Mech. Engr., Stanford Univ., 1984.
77. W. R. VAN DALSEM, Dissertation, Dept. of Aeronautics and Astronautics, Stanford Univ., 1984.