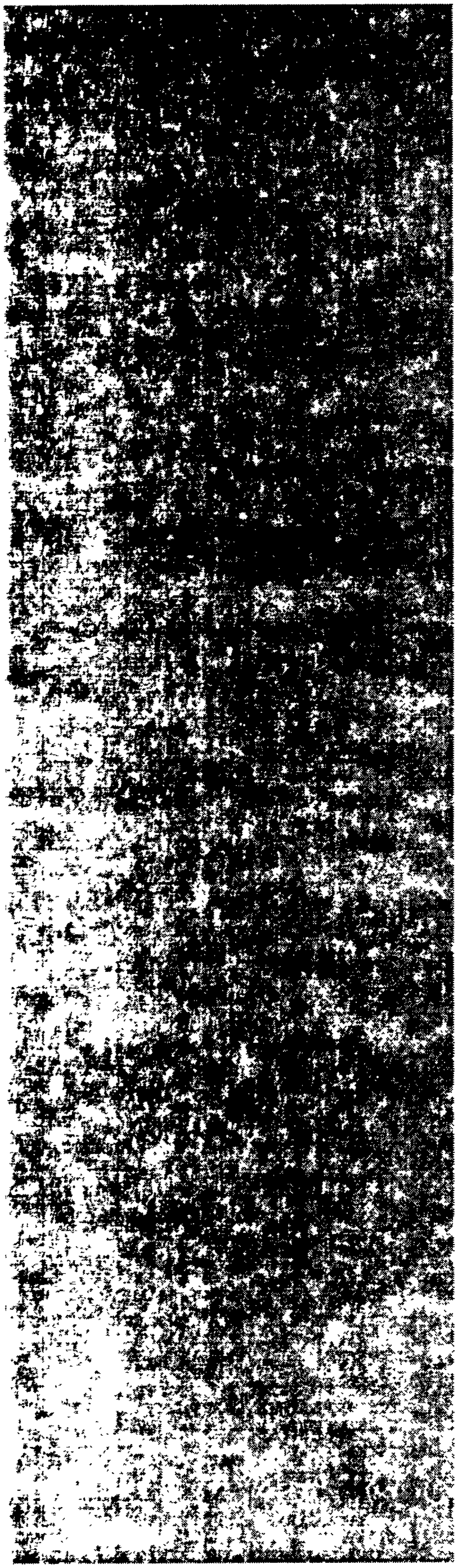
*LA-13427-MS*



# *Three-Dimensional Computational Fluid Dynamics*

**Los Alamos**
NATIONAL LABORATORY

*Los Alamos National Laboratory is operated by the University of California for the United States Department of Energy under contract W-7405-ENG-36.*

*This work was supported by the US Department of Energy, Office of Transportation Technologies (EE-30).*

*Reprinted from ASM Handbook, Volume 20, Materials Selection and Design, pp. 186–203, ASM International (1997), with permission of ASM.*

*This report was prepared as an account of work sponsored by an agency of the United States Government. Neither The Regents of the University of California, the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by The Regents of the University of California, the United States Government, or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of The Regents of the University of California, the United States Government, or any agency thereof. Los Alamos National Laboratory strongly supports academic freedom and a researcher's right to publish; as an institution, however, the Laboratory does not endorse the viewpoint of a publication or guarantee its technical correctness.*

## DISCLAIMER

Portions of this document may be illegible electronic image products. Images are produced from the best available original document.

LA-13427-MS

UC-1500
Issued: September1998

# *Three-Dimensional Computational Fluid Dynamics*

*Peter J. O'Rourke*
*Daniel C. Haworth**
*Raj Ranganathan**

**General Motors Corporation*

Los Alamos
NATIONAL LABORATORY
Los Alamos, New Mexico 87545

## CONTENTS

ABSTRACT . . . . . 1
I. OVERVIEW . . . . . 1
II. FUNDAMENTALS OF COMPUTATIONAL FLUID DYNAMICS . . . . . 5
A. The Governing Equations of Fluid Dynamics . . . . . 5
1. The Equations of Continuous, Compressible Media . . . . . 5
2. Constitutive Relations of Fluid Flow . . . . . 7
3. Simplifications of the Fluid-Flow Equations . . . . . 8
4. Turbulence and Other Models . . . . . 10
B. Numerical Solution of the Fluid-Flow Equations . . . . . 13
1. Discretization of the Fluid Equations . . . . . 13
a. Finite-Difference Methods . . . . . 13
b. Finite-Volume Methods . . . . . 16
c. Finite-Element Methods . . . . . 17
d. Spectral Methods . . . . . 20
e. Computational Particle Methods . . . . . 20
2. Solution of Implicit Equations . . . . . 21
C. Grid Generation for Complex Geometries . . . . . 23
1. Unstructured Meshes . . . . . 23
2. Specialized Differencing Techniques . . . . . 24
III. COMPUTATIONAL FLUID DYNAMICS FOR ENGINEERING DESIGN . . . 25
A. The CFD Process . . . . . 26
1. Geometry Acquisition . . . . . 28
2. Grid Generation and Problem Specification . . . . . 30
3. Flow Solution . . . . . 31
4. Post-Processing and Synthesis . . . . . 32
B. Examples of Engineering CFD . . . . . 32
1. Internal Duct Flow . . . . . 33
2. External Aerodynamics . . . . . 33
3. Manufacturing Processes . . . . . 33
4. Building Interior . . . . . 36
5. Environmental Flow . . . . . 38
6. Internal Combustion Engine . . . . . 38
IV. ISSUES AND DIRECTIONS FOR ENGINEERING CFD . . . . . 39
A. Geometric Fidelity . . . . . 39
B. Numerical Inaccuracy . . . . . 42
C. Physical Models . . . . . 43
D. User Expertise . . . . . 43
E. CFD and Experimental Measurements . . . . . 44
F. Interdisciplinary Analysis . . . . . 44
G. Future of Engineering CFD . . . . . 44
V. SOURCES FOR FURTHER INFORMATION . . . . . 45
ACKNOWLEDGEMENTS . . . . . 47
REFERENCES . . . . . 47

# THREE-DIMENSIONAL COMPUTATIONAL FLUID DYNAMICS

by

Peter J. O'Rourke, Daniel C. Haworth, Raj Ranganathan

## ABSTRACT

Computational fluid dynamics (CFD) is one discipline falling under the broad heading of computer-aided engineering (CAE). CAE, together with computer-aided design (CAD) and computer-aided manufacturing (CAM), comprise a mathematical-based approach to engineering product and process design, analysis, and fabrication. In this overview of CFD for the design engineer, our purposes are three-fold: (1) to define the scope of CFD and motivate its utility for engineering, (2) to provide a basic technical foundation for CFD, and (3) to convey how CFD is incorporated into engineering product and process design.

## I. OVERVIEW

The objective of CFD is the numerical solution of fluid-flow equations. The calculus problem of solving a coupled system of nonlinear partial differential equations (PDEs) for the variables of interest (e.g., velocity, pressure, and temperature) is transformed into an algebra problem of solving a large system of simultaneous linear equations for discrete unknowns that represent the state of a thermal-fluids system; the latter is amenable to numerical solution on a digital computer.

The above is a somewhat abstract description of CFD, but one necessarily must speak in general terms to introduce a subject that encompasses such a wide variety of solution techniques. In this overview we will talk about finite difference, finite volume, finite element, spectral, and some computational particle methods. The emphasis will be on the first three in this list, as these are the methods that are primarily used in contemporary CFD codes for engineering design.

Here we reserve the terminology "computational fluid dynamics" for computationally intensive three-dimensional simulations of thermal-fluids systems where nonlinear momentum transport plays an important role. It does not encompass all branches of numerical analysis as applied to fluid dynamics problems. In particular, we exclude consideration of zero- or quasi-dimensional analysis of fluids systems[1,2] and linear heat conduction or potential flow problems.[3,4]

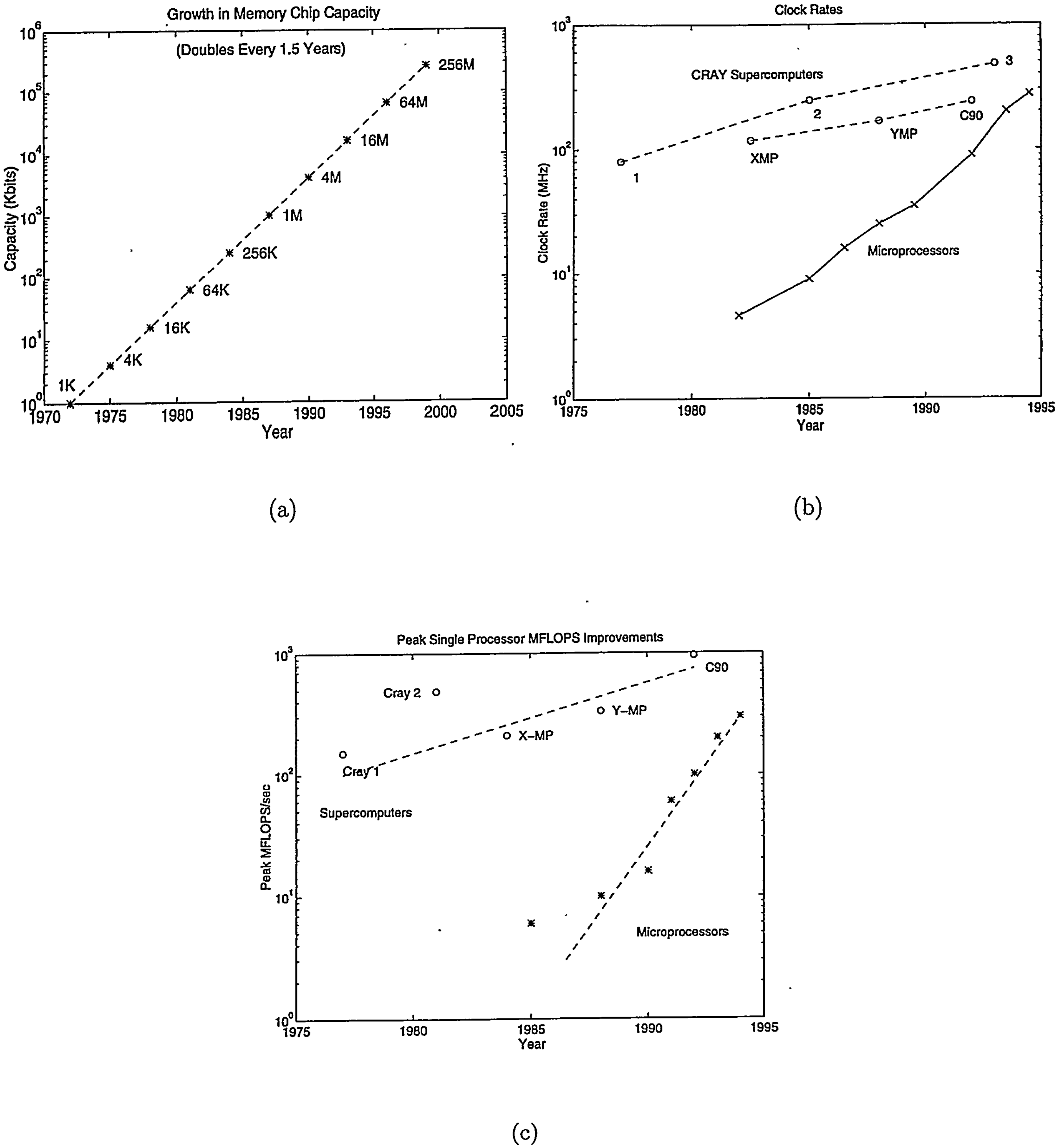
The practice of CFD began with the advent of computers; indeed, the first computer was developed, in part, to solve fluid-flow equations. It was recognized by the developers of the atomic bomb at Los Alamos that many fluid dynamics problems were impossible to solve by analytic means. What was needed was a machine that could perform the massive number of calculations necessary to solve the flow equations by simple finite-difference methods. The ENIAC computer began operating shortly after World War II, and its first calculations were to test various configurations for a hydrogen bomb.[5] Ensuring the safety and reliability of today's nuclear weapons remains a major impetus for the development of more powerful computers and more efficient numerical techniques for solving the fluid-flow equations.[6]

Initially most numerical solutions were limited to flows that could be approximated as spatially one- or two-dimensional; the time and expense of performing three-dimensional calculations remained prohibitive. Over the last 15 years, however, CFD calculations of three-dimensional flows have become more common. This has heightened enormously the interest in CFD among engineers, as most real flows are three dimensional. In fact, most fluid-flow problems encountered in industry are so complex that the *only* method of analysis to which they are amenable is CFD. Thus although CFD was born only 50 years ago, it is difficult to find problems in fluid dynamics to which computer solution has not been brought to bear.

The capability to perform three-dimensional CFD has resulted primarily from the availability of faster computers with larger memories (Fig. 1).[7] The development of parallel and massively parallel computers promises to further improve the speed and extend the applicability of CFD. A recent simulation of the oceans[8,9] serves to illustrate the problem size and computational requirements that have been realized in modern applications. This problem was run on a 512-node massively parallel computer, requiring 10 gigabytes of memory (giga = billion, one byte = eight bits). It ran at a computational speed of four gigaflops ("flops" = floating point operations per second), and required 80 days of computer time. A plot of ocean surface temperature obtained in this simulation is shown in Fig. 2. Computers with maximum performance at one teraflop (tera = trillion) now are becoming available, and petaflop computers (peta = 1000 trillion) are being planned for the next decade.[10]

At the same time, improved numerical methods have yielded higher computational efficiency: that is, fewer operations and/or less memory for a given accuracy. Among the most important of these advances has been the development of faster methods for solving implicit difference approximations (see the following section). A third enabler for three-dimensional CFD has been the formulation of improved finite-volume and finite-element methods that better accommodate the complex geometrical boundaries that characterize engineering flows. Examples of engineering applications are given later in this report.

The advent of three-dimensional calculations has increased the engineering relevance of CFD, but many obstacles remain to be overcome before CFD realizes its full potential



(a)

(b)

(c)

**Fig. 1.** Growth in computer hardware performance, 1970 to 1995.[7] (a) Memory chip capacity doubles every 1.5 years. (b) Clock rate. (c) Peak single-processor megaflops.

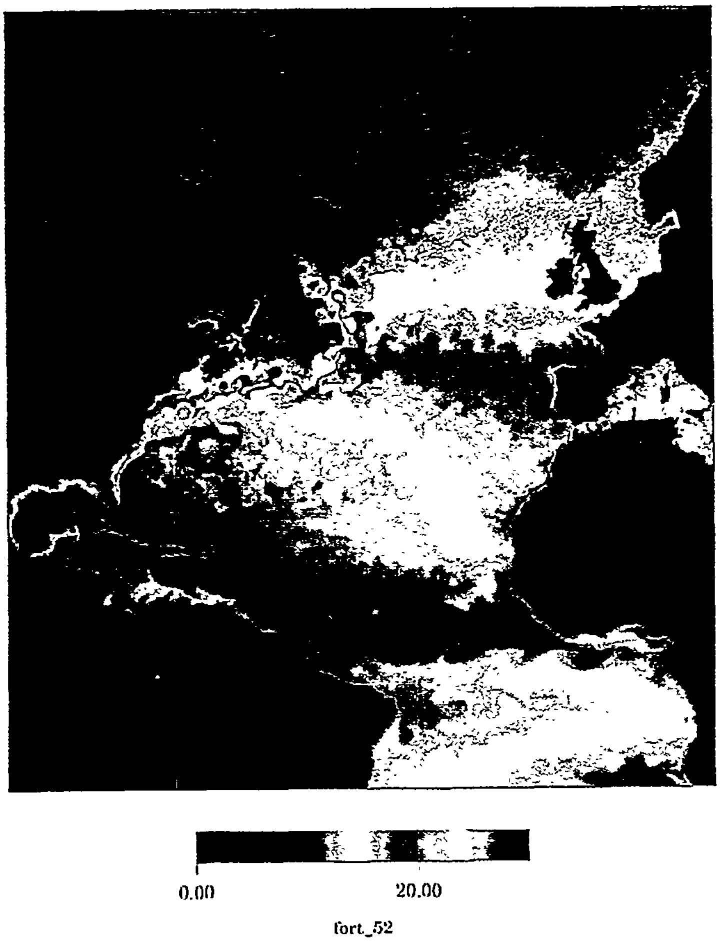


**Fig. 2.** Ocean surface temperatures from a recent CFD simulation of the north Atlantic Ocean.[8,9]

as an engineering design tool. Foremost among these is spatial resolution. Most flows of practical interest have features whose relevant spatial and temporal scales span many orders of magnitude. For example, in an automotive four-stroke-cycle, spark-ignited internal-combustion engine operating at 2000 rpm, hydrodynamic scales range from about 0.01 mm (the turbulence microscale) to 100 mm (the bore diameter); flame thicknesses (stoichiometric, undiluted reactants) are in the range 0.01–0.10 mm; and, spray droplets issuing from a typical port-fuel injector have diameters as small as 0.10 mm.[11] Computers do not exist, and will not exist in the foreseeable future, that can store all the numbers required to fully resolve these phenomena. Thus, the effects of small-scale, unresolvable features on the large-scale, average flow features of interest are "modeled" through modifications to the governing PDEs. Examples of models include turbulence models, combustion models, and multiphase flow models. All models necessarily introduce imprecision, and an ongoing goal of research is to improve the accuracy of these models.

Other issues for three-dimensional engineering CFD include geometry acquisition and grid generation, numerical accuracy, and diagnostics to extract the physical information of interest from the computations. Modeling and other issues are discussed further in subsequent sections.

## II. FUNDAMENTALS OF COMPUTATIONAL FLUID DYNAMICS

We now give the governing equations of fluid dynamics and introduce the CFD techniques for their solution. We also introduce some basic terminology used by practitioners of CFD. Readers who are not interested in the technical foundation of CFD may proceed to the next major section on how CFD is incorporated into the engineering design process.

### A. The Governing Equations of Fluid Dynamics

The equations of fluid dynamics can be derived from kinetic theory or continuum points of view,[12,13,14] each of which complements the other. Kinetic theory regards the fluid as made up of molecules subject to collisions and inter-molecular forces. Kinetic theory derivations are valid only for dilute gases but give detailed information about how transport phenomena, such as stresses and heat fluxes, arise from molecular fluctuations, which in turn are related to the average molecular properties for which the fluid equations solve. Continuum derivations regard the fluid as a continuous medium, show the applicability of the fluid equations to a much broader class of media than dilute gases, but do not give detailed information about transport phenomena.

**1. The Equations of Continuous, Compressible Media.** Three basic physical principles, applicable to any continiuous medium, are used in continuum derivations:

1. conservation of mass,
2. Newton's second law that force equals mass times acceleration, and
3. the first law of thermodynamics that total energy, in all its forms, must be conserved.

These three principles lead to the following three equations of motion.

the mass, or continuity, equation

$$\frac{\partial \rho}{\partial t} + \frac{\partial \rho u_i}{\partial x_i} = 0\,, \qquad (1)$$

the momentum equation

$$\frac{\partial \rho u_i}{\partial t} + \frac{\partial\left(\rho u_i u_j\right)}{\partial x_j} = \frac{\partial \Sigma_{ij}}{\partial x_j} + \rho F_i\,, \quad \text{and} \qquad (2)$$

the total energy equation

$$\frac{\partial \rho E}{\partial t} + \frac{\partial\left(\rho E u_j\right)}{\partial x_j} = \frac{\partial\left(\Sigma_{ij} u_j\right)}{\partial x_i} - \frac{\partial Q_j}{\partial x_j} + \rho F_i u_i\,. \qquad (3)$$

We have written these equations in Cartesian tensor notation,[15] according to which the subscripts $i$ and $j$ take the values 1, 2, or 3 corresponding to the three Cartesian coordinate directions. A subscript that appears just once in a term takes on one of the three values 1, 2, or 3; repeated subscripts in a term denote a summation of that term over all three coordinate directions. The other notation in Eqs. (1), (2), and (3) is defined in Table 1.

Table 1. Nomenclature

| Symbol | Definition |
|---|---|
| $C_p$ | specific heat at constant pressure |
| $C_v$ | specific heat at constant volume |
| $e$ | internal energy |
| $E$ | total energy |
| $F$ | body force per unit mass |
| $h$ | enthalpy |
| $K$ | turbulent kinetic energy |
| $p$ | pressure |
| $Q$ | heat flux |
| $R$ | universal gas constant |
| $S$ | rate of deformation |
| $t$ | time |
| $T$ | temperature |
| $u$ | velocity |
| $W$ | molecular weight |
| $x$ | spatial location |
| $\delta$ | Kronecker delta function |
| $\epsilon$ | dissipation rate of turbulent kinetic energy |
| $\kappa$ | heat conductivity |
| $\lambda$ | second coefficient of viscosity |
| $\lambda'$ | bulk viscosity |
| $\mu$ | first coefficient of viscosity |
| $\rho$ | mass density |
| $\Sigma$ | stress |

We note here that the total energy, $E$, is the sum of the local flow kinetic energy and its internal energy $e$

$$E = e + \frac{1}{2}u_i^2 \, . \tag{4}$$

Alternative energy equations for $e$ and for enthalpy $h = e + p/\rho$, where $p$ is the pressure, may easily be derived using Eqs. (1), (2), and (3). CFD codes often solve internal energy or enthalpy equations, in place of Eq. (3), when calculating compressible flows.

The above equations are expressed in Eulerian form, by which we mean that the time derivative is taken at a fixed point in space. In contrast, in Lagrangian form, the time derivative is taken following a fluid element.[14] Although the Eulerian form of the equations is most often used in CFD, there are CFD methods that approximate the Lagrangian equations.

When completed with constitutive relations appropriate for fluids, these are the basic equations of compressible fluid dynamics. In practice one often encounters applications in which extensions of these equations are necessary. Among the most common are extensions to multicomponent and chemically reactive flows,[16] to magnetohydrodynamic flows,[17] and to flows with radiative heat transfer.[18] It is beyond the scope of this modest overview of CFD to give these extended equations, and the reader is referred to the cited references for this information.

**2. Constitutive Relations of Fluid Flow.** To complete these equations we need to express the stress $\Sigma_{ij}$ and heat flux $Q_i$ in terms of known fluid variables and their derivatives. These expressions are known as constitutive relations. A fluid is a medium for which the nonhydrostatic part of its stress depends only on its *rate* of deformation $S_{ij}$, or what is usually referred to as the rate of strain, and not on its deformation. The quantity $S_{ij}$ is given by

$$S_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) \, . \tag{5}$$

Thus, a fluid has no memory of its previous configurations. This fact, together with the assumption of an isotropic medium in terms of its microstructure, allow us[14] to express the full stress as

$$\Sigma_{ij} = 2\mu S_{ij} + \lambda S_{kk}\delta_{ij} - p\delta_{ij} \, , \tag{6}$$

where $\delta_{ij}$ is the Kronecker delta function.[15] Fluids with this form of stress tensor are called Newtonian fluids. The thermodynamic pressure $p$ and the first and second coefficients of viscosity $\mu$ and $\lambda$, depend only on the local thermodynamic state of the fluid. Often the second coefficient of viscosity in Eq. (6) is replaced by the bulk viscosity $\lambda'$, defined by

$$\lambda' = \lambda + \frac{2}{3}\mu \, . \tag{7}$$

The heat flux $Q_i$ depends on gradients in temperature. Again assuming an isotropic fluid $Q_i$ may be written

$$Q_i = -\kappa \frac{\partial T}{\partial x_i} . \tag{8}$$

This is Fourier's heat conduction law, and $\kappa$ is called the heat conduction coefficient or simply the heat conductivity. Its value depends on the local thermodynamic state of the gas. When one substitutes Eq. (6) for the stress tensor and Eq. (8) for the heat flux vector into Eqs. (1), (2), and (3), the resulting equations are called the compressible Navier-Stokes equations.

The fluid equations are completed by the specification of the quantities $p$, $e$, $\mu$, $\lambda$, and $\kappa$ in terms of the local fluid temperature $T$ and density $\rho$. The equations specifying $p$ and $e$ are referred to as thermal and caloric equations of state, respectively. For a so-called ideal gas these are given by

$$p = \rho \frac{R}{W} T , \tag{9}$$

and

$$e = \int^T C_v(T') \, dT' \tag{10}$$

where the specific heat at constant volume $C_v$ is a function of temperature. Alternatively, the enthalpy $h$ is given by

$$h = \int^T C_p(T') \, dT' \tag{11}$$

where, from the definition of $h$ and the thermal equation of state Eq. (9), the specific heat at constant pressure $C_p$ is given by

$$C_p = C_v + \frac{R}{W} . \tag{12}$$

Values of $C_v$ and $C_p$ versus temperature are given in the references.[19,20]

The quantities $\mu$, $\lambda$, and $\kappa$ are called transport coefficients. How they are related to the local thermodynamic state of the fluid and its molecular properties are given in another reference.[21] Given an expression for the viscosity $\mu$, the heat conductivity $\kappa$ can frequently be approximated by

$$\kappa = \frac{\mu C_p}{Pr} \tag{13}$$

where $Pr$ is the Prandtl number, whose value is nearly constant and often of order unity.

**3. Simplifications of the Fluid-Flow Equations.** For certain flow situations considerable computer time can be saved by solving simplified forms of the compressible flow equations. In this section we introduce the steady-state, inviscid, and incompressible approximations, and we describe the circumstances under which they may be used.

The steady-state approximation is obtained simply by dropping the time derivative terms in Eqs. (1), (2), and (3). While solving the steady-state equations can often save computer time, sometimes CFD solution techniques for the steady-state equations have what are called convergence difficulties, and steady fluid flow solutions are more reliably obtained by calculating the long-time limits of solutions to the unsteady equations.[22]

The inviscid, or Euler, equations are obtained by neglecting the viscosity and heat conduction terms in the preceding equations. A necessary condition for the applicability of the Euler equations is that the Reynolds number, *Re*, be much greater than one, where *Re* is defined by

$$Re = \frac{\rho u L}{\mu} \, . \tag{14}$$

In Eq. (14) $\rho$, $u$, and $\mu$ are characteristic values of the density, velocity, and viscosity, respectively, of the fluid, and $L$ is a characteristic distance over which the velocity changes appreciably, also called a gradient length. In a flow to which the Euler equations apply, $L$ is typically the dimension of the apparatus that bounds the flow. The Reynolds number is approximately the ratio of the magnitude of the advective terms to that of the viscous terms in the fluid momentum equation, Eq. (2). Thus when *Re* is large, the viscous terms may sometimes be neglected. When fluid Prandtl numbers are of order unity, smallness of the viscous terms also implies smallness of the heat conduction terms relative to the advection terms in the energy equation.

There are many high-Reynolds number flows, however, where neglect of the viscous and heat conduction terms is not justified. Sometimes fluid flows have broad regions over which the inviscid equations apply, coupled with thin regions (e.g., boundary layers and shocks) in which the viscous and heat conduction terms are important. In addition, as *Re* is increased, many flows become turbulent, and the velocity then varies over a range of length scales, $L$. At the smallest of these length scales, *Re* is of order unity, and viscosity is important because it is responsible for the dissipation of turbulent kinetic energy into heat. (See "turbulence," below.)

An incompressible flow is one in which the divergence of the velocity field is identically equal to zero

$$\frac{\partial u_i}{\partial x_i} = S_{ii} = 0 \, . \tag{15}$$

A necessary, but not sufficient,[14] condition that a flow be incompressible is that the Mach number, $M$, be much less than one, where $M$ is defined by

$$M = \frac{u}{c} \, . \tag{16}$$

In Eq. (16) $u$ and $c$ are characteristic values of the velocity and sound speed of the fluid.

In combination with the continuity equation Eq. (1), Eq. (15) implies that

$$\frac{D\rho}{Dt} = \frac{\partial \rho}{\partial t} + u_i \frac{\partial \rho}{\partial x_i} = 0 \, . \tag{17}$$

$D/Dt$ is the time derivative following a fluid element, and Eq. (17) states that the density of each element of fluid remains a constant along its trajectory. Commonly, a more restrictive assumption is made that the density of the whole fluid is equal to a constant $\rho_0$. In this case, the momentum equation, Eq. (2), becomes

$$\frac{\partial u_i}{\partial t} + \frac{\partial (u_i u_j)}{\partial x_j} = \frac{1}{\rho_0} \frac{\partial \Sigma_{ij}}{\partial x_j} + F_i \, . \tag{18}$$

The great simplification of the incompressible flow equations is that the energy equation is decoupled from the momentum equation and need not be solved. What we have given here is the so-called primitive-variable form of the incompressible flow equations. Another formulation that is used in CFD calculations of two-dimensional, incompressible flows is the stream function and vorticity formulation.[3]

**4. Turbulence and Other Models.** As stated in the introduction, there are many flow situations in which flows have changes in their properties, such as their velocities, with superimposed size scales or time scales that differ by many orders of magnitude. Examples are the seemingly chaotic motions in a turbulent flow or in a multiphase flow, such as a liquid spraying into a gas. Classical theories of turbulence predict that the ratios of the largest to the smallest fluctuation length scales of turbulent flows are approximately equal to $Re^{0.75}$, where here the Reynolds number $Re$ is based on the velocity and size scales of the largest turbulent eddies.[23] Even a low value of $Re = 10\,000$ gives fluctuation length scales varying over 3 orders of magnitude. For such cases it is impossible to resolve the detailed flow fluctuations with CFD methods, and fortunately we are not concerned with predicting these detailed fluctuations. We are concerned with average flow behavior, however, and it is important to account for the effects of the fluctuations on average flow variables.

There are many ways to define averaged, or filtered, flow variables. In general, space- and time-averages can be defined using a filter function $K(x_i, t)$ whose integral overall space and time is unity. In terms of $K$, the average of a fluid variable $q$, denoted by $\overline{q}$, is defined by

$$\overline{q}(x_i, t) = \iiiint q(y_i, t')K\left(y_i - x_i, t' - t\right)\, dy_i\, dt' \ . \tag{19}$$

For example, for pure time-averaging one can take $K(x_i, t) = \delta(x_i)\Psi_T(t)/T$, where

$$\Psi_T(t) = \begin{cases} 1 & \text{if } |t| < T/2 \\ 0 & \text{otherwise ,} \end{cases} \tag{20}$$

and $\delta(x_i)$ is the Dirac delta function. Then the average $q$ is defined by

$$\overline{q}(x_i, t) = \frac{1}{T}\int_{t-T/2}^{t+T/2} q(x_i, t')\, dt' \tag{21}$$

and the filter size is said to be $T$. In addition to space- or time-averaging, one can also use ensemble averaging. This is defined by averaging over an imagined large set of realizations of a fluid experiment. Sometimes ensemble-averaging is combined with space- or time-averaging. In any case, the fluctuation of quantity $q$ from its mean value is denoted by $q'$

$$q'(x_i, t) = q(x_i, t) - \overline{q}(x_i, t) \ . \tag{22}$$

There are two approaches to calculating average flow fields. In the first, called Reynolds averaging because it was first proposed by O. Reynolds,[24] one is interested in predicting the average flow field and uses ensemble averaging, or a filter size that is large compared with the scales of fluctuations. Thus the average of the fluctuating part of $q$ is zero

$$\overline{(q')} = 0 \qquad \text{(Reynolds averaging).} \tag{23}$$

In contrast, subgrid-scale turbulence models use filters with as small a size as possible, typically comparable to the grid size in one's CFD calculation. Thus, one attempts to calculate flow fluctuations with scales larger than the filter, or grid size, and to model only subgrid-scale fluctuations. In a subgrid-scale model the average of the fluctuating part of $q$ is, in general, nonzero

$$\overline{(q')} \neq 0 \qquad \text{(Subgrid model filtering).} \tag{24}$$

Once the method of averaging is chosen, then equations for the averaged flow variables can be obtained by averaging the equations of the preceding sections, or simplified forms of these. In deriving the averaged equations, one finds that the rates of change of average flow variables depend upon averages of the products of two fluctuating quantities, also called second-order correlations. The values of these are unknown; and if one tries to close the system of equations by deriving transport equations for the second-order correlations, it is found that these depend on third-order correlations, or averages of three fluctuating quantities. Continuing in this way one finds that it is impossible to obtain a closed system of equations using either Reynolds averaging or subgrid-scale averaging.[25] By using physical and dimensional reasoning and empirical information, the unknown correlations must be expressed (the word "modeled" is also used here) in terms of average flow variables that are known.

A very important example of a second-order correlation and its modeling arises when averaging the incompressible flow momentum equation, Eq. (18). The Reynolds-averaged form of this equation is

$$\rho_0 \left( \frac{\partial \overline{u}_i}{\partial t} + \frac{\partial \overline{u}_i \overline{u}_j}{\partial x_j} \right) = \frac{\partial \overline{\Sigma}_{ij}}{\partial x_j} - \frac{\partial}{\partial x_j} \left( \rho_0 \overline{u_i' u_j'} \right) + \rho_0 \overline{F}_i \, . \tag{25}$$

The second-order correlation, $-\rho_0 \overline{u_i' u_j'}$, on the right-hand side of this equation is called the Reynolds stress. The most popular turbulence models in engineering design calculations are the so-called two-equation models, in which the Reynolds stress is, with some theoretical justification,[25,26] taken to have the form

$$-\rho_0 \overline{u_i' u_j'} = 2\mu_T \overline{S}_{ij} - \frac{2}{3} \rho_0 K \delta_{ij} \, . \tag{26}$$

In this expression $\mu_T$ is the turbulent viscosity and $K$ is the turbulent kinetic energy

$$K = \frac{1}{2} \overline{(u_i')^2} \, . \tag{27}$$

By substituting for the Reynolds stress in Eq. (25) using Eq. (26), one finds that the momentum equation for turbulent flow closely resembles the momentum equation for laminar, or nonturbulent, flow. This is also the case for other averaged fluid equations, and this resemblance allows the same numerical techniques for CFD to be applied to both laminar and turbulent flows.

In two-equation turbulence models, transport equations are solved for the turbulent kinetic energy, $K$, and one other scalar that gives a local length or time scale of the turbulence. A popular choice for this second turbulence quantity is the turbulence kinetic energy dissipation rate, $\epsilon$. In terms of $K$ and $\epsilon$ the turbulent viscosity is given by

$$\mu_T = c_\mu \rho_0 \frac{K^2}{\epsilon} \, , \tag{28}$$

where $c_\mu$ is a dimensionless constant. Launder and Spalding[26] describe the $K/\epsilon$ turbulence model in more detail in Ref. 26, and the use of wall functions to calculate wall heat and momentum losses in conjunction with the $K/\epsilon$ model. Descriptions of many two-equation turbulence models, and their relative advantages, may be found in Wilcox.[27]

### B. Numerical Solution of the Fluid-Flow Equations

We now introduce some common techniques for discretizing the fluid-flow equations and methods for solving the discrete equations.

**1. Discretization of the Fluid Equations.** In the process of discretization we represent a continuously varying fluid-flow field, which has an infinite number of degrees of freedom, by a finite set of data. In this section we introduce the discretization techniques used by finite difference, finite volume, finite element, spectral, and some particle methods, and associated concepts of numerical stability and accuracy. The discrete equations of the finite-difference, finite-volume, and finite-element techniques all look similar and are referred to generically as "difference approximations." In this introduction to CFD we have only time to "scratch the surface" of each method. For more in-depth information, the reader should consult one of several standard books on the subject.[3,28,29,30,31,32]

*a. Finite-Difference Methods (FDMs).* In FDMs we subdivide all the fluid region of interest into nonoverlapping cells and store approximate values of the fluid variables in each cell. This subdivision is called a grid or a mesh. Derivatives are approximated by taking differences between the variable values in neighboring cells, using the idea of a Taylor-series expansion. Let us consider the simple one-dimensional example of finite-difference solution of the linear advection equation

$$\frac{\partial q}{\partial t} + u\frac{\partial q}{\partial x} = 0 \tag{29}$$

in the spatial interval $a \leq x \leq b$. We subdivide this interval into cells of equal size $\Delta x = (b-a)/N$, where $N$ is the total number of cells, and denote by $q_i^n$ the approximate value of $q$ at the center of cell $i$, which lies at the location, or grid point, $x_i = a+(i-1/2)\Delta x$, at time $t = n\Delta t$, where $\Delta t$ is the computational timestep. (In this section, the subscript $i$ will represent a cell number, rather than a coordinate direction.) We have stored in computer memory all the values of $q_i^n$, $1 \leq i \leq N$, for a particular time $t = n\Delta t$, and we wish to compute values at time $t = (n+1)\Delta t$ by using a finite-difference approximation to Eq. (29).

To approximate the spatial derivative in Eq. (29), we consider the $q_i^n$ to be the values at $x_i$ of a differentiable function $q(x,t)$ that can be expanded in a Taylor series about any grid point. Thus the value of $q$ at a neighboring grid point can be expressed in terms of the value of $q$ and its derivatives at grid point $i$ by

$$q_{i+k}^n = q_i^n + \left(\frac{\partial q}{\partial x}\right)_i^n k\Delta x + \left(\frac{\partial^2 q}{\partial x^2}\right)_i^n \frac{(k\Delta x)^2}{2} + O(\Delta x^3)\,, \tag{30}$$

where $O(\Delta x^m)$ represents the fact that the remaining terms in this expansion have as their lowest-order term one in which $\Delta x$ is raised to the power $m$. Now the value of the spatial derivative in Eq. (29) can be approximated by any finite combination that satisfies

$$\sum_k a_k q_{i+k}^n = \left(\frac{\partial q}{\partial x}\right)_i^n + O(\Delta x^m)\,, \tag{31}$$

when one substitutes from Eq. (30) for the $q_{i+k}$, where the $a_k$ are coefficients that depend on $\Delta x$. In the approximation Eq. (31), '$m$' is said to be the order of accuracy of the approximation and all terms containing $\Delta x$ to some power are said to be truncation errors. As long as $m > 0$, the approximation is said to be consistent. Examples of consistent approximations are the centered-difference approximation

$$\frac{q_{i+1}^n - q_{i-1}^n}{2\Delta x} = \left(\frac{\partial q}{\partial x}\right)_i^n + O(\Delta x^2)\,, \tag{32}$$

which is second-order accurate, and the one-sided approximations

$$\frac{q_i^n - q_{i-1}^n}{\Delta x} = \left(\frac{\partial q}{\partial x}\right)_i^n + O(\Delta x)\,, \tag{33a}$$

and

$$\frac{q_{i+1}^n - q_i^n}{\Delta x} = \left(\frac{\partial q}{\partial x_i}\right)_i^n + O(\Delta x)\,, \tag{33b}$$

which are first-order accurate. If the advection speed $u$ in Eq. (29) is positive, then the approximation Eq. (33a) is called an upwind approximation and Eq. (33b) a downwind approximation.

Order of accuracy is one measure of the accuracy of a finite-difference method. To test the accuracy of a finite-difference solution one can refine the grid by reducing the cell size $\Delta x$. When $\Delta x$ is reduced by a factor of 2, numerical errors will be reduced approximately by a factor of 4 when using a second-order method, but only by a factor of 2 with a first-order method. It may thus seem to be desirable to use only methods with a very high order of accuracy. In practice, however, it is difficult to define high-order methods near boundaries, and often numerical solutions using high-order methods have oscillations in regions of steep gradients. Because of these difficulties, most modern finite-difference methods have second- to fourth-order accuracy and sometimes drop to first-order accuracy in regions of steep gradients.

Returning to the example of the linear advection equation, the time derivative can be approximated in much the same way as the spatial derivative. Since one usually only stores the values of $q_i$ at a single time-level in order to save computer storage, the time derivative is most often approximated by the one-sided finite-difference formula

$$\frac{q_i^{n+1} - q_i^n}{\Delta t} = \left(\frac{\partial q}{\partial t}\right)_i^n + O(\Delta t)\,. \tag{34}$$

When Eq. (34)—and one of the finite-difference formulas Eq. (32), Eq. (33a), or Eq. (33b)—are used to approximate the time- and space-derivatives in Eq. (29), one obtains a consistent approximation to the linear advection equation that is first-order accurate in time.

When these finite-difference equations are used to advance the numerical solution for $q$ in time, one finds that, in contrast to solutions to the differential equation, solutions to the finite-difference equations using Eq. (32) or the downwind approximation Eq. (33b) are subject to catastrophic numerical instabilities, and solutions using Eq. (33a) are only stable if a certain condition is met. This condition, the so-called Courant condition, is that the Courant number $C = (u\Delta t)/\Delta x$ be less than one. The origin of these numerical instabilities was first discovered by J. von Neumann,[33] who devised a method for analyzing the stability of linear finite-difference equations based on examining the behavior of each Fourier component of the solution.

The finite-difference approximations we have presented so far are explicit in the sense that the solution for $q_i^{n+1}$ can be explicitly found by solving only the finite-difference equation at grid point $i$. All explicit methods, if they are stable, are subject to Courant conditions to ensure their numerical stability. Intuitively, this condition arises because when using an explicit method, information can only propagate at a speed proportional to $\Delta x/\Delta t$. In order for the numerical solution to approximate the physical solution, the numerical propagation speed must be at least as great as the physical speed. For the simple advection Eq. (29), the only physical propagation speed is $u$. For the fluid equations there are several physical, or characteristic, speeds. The largest of these is $u + c$, where $c$ is the fluid speed of sound, and the Courant condition in explicit CFD calculations is based on the speed $u + c$. To overcome the Courant condition one uses implicit finite-difference methods, in which solution for the value of $q_i^{n+1}$ is implicitly coupled to the solution for

$q^{n+1}$ at other grid points. An example of an implicit finite-difference approximation to the linear advection Eq. (29) is

$$\frac{q_i^{n+1} - q_i^n}{\Delta t} = -u \frac{q_{i+1}^{n+1} - q_{i-1}^{n+1}}{2\Delta x} \,, \tag{35}$$

which can be shown to be unconditionally stable. The disadvantage of implicit methods is that they usually require costly iterative solution. Some iterative solution techniques for implicit equations will be introduced below.

*b. Finite-Volume Methods (FVMs).* As in FDMs, FVMs subdivide the computational region into a mesh of cells; but finite-volume cells can be arbitrary quadrilaterals in two dimensions, hexahedra in three dimensions, or indeed any shape enclosed by a set of corner points. In contrast, FDMs are defined on grids that are obtained using orthogonal curvilinear coordinate systems. Thus, FVMs have much more geometric flexibility than FDMs.

Finite-volume methods approximate forms of the fluid equations that are integrated over these cells, which are also called control volumes. As an example, we consider finite-volume approximation of the integrated form of the mass equation, Eq. (1). After integrating Eq. (1) over control volume $V$ and applying the Reynolds transport and divergence theorems[14] one obtains

$$\frac{d}{dt} \iiint\limits_V \rho \, dv + \iint\limits_S \rho u_i n_i \, da = 0 \,. \tag{36}$$

The quantity $\rho u_i n_i$ is the mass flux (mass per unit area and time) through surface $S$ with unit normal $n_i$, and Eq. (36) is a statement that the time-rate-of-change of the total mass in volume $V$ is equal to the sum of the fluxes, times the areas, through the surface $S$ of the volume. Thus mass is conserved in the sense that there are no internal mass sources. Commonly, the time derivative term in Eq. (36) is approximated by

$$\frac{d}{dt} \iiint\limits_v \rho \, dv \approx V_\nu \frac{\rho_\nu^{n+1} - \rho_\nu^n}{\Delta t} \,, \tag{37}$$

where $\nu$ is the index of a finite-volume cell and $V_\nu$ is its volume, and the surface integral is approximated by

$$\iint \rho u_i n_i \, da \approx \sum_\alpha \rho_\alpha (u_i)_\alpha (n_i)_\alpha A_\alpha \,, \tag{38}$$

where the sum is over all faces $\alpha$ of control volume $\nu$; $\rho_\alpha$ and $(u_i)_\alpha$ are approximations to $\rho$ and $u_i$, respectively, on face $\alpha$; $(n_i)_\alpha$ is an average unit normal vector to face $\alpha$ pointing out of volume $V$; and $A_\alpha$ is the area of face $\alpha$.

Using FVMs one can easily construct discrete approximations that have the conservative property; that is, the discrete approximations can mimic the physical laws from which the fluid equations were derived by conserving properties such as computed mass, momentum, and energy. To be more precise, consider the approximation to the mass equation above. A conservative approximation has the property that if $\nu$ and $\mu$ are two cells that share face $\alpha$, then when one sums the finite-volume approximations to the change of mass in cells $\nu$ and $\mu$, the contributions due to fluxes through common face $\alpha$ cancel each other. This will be true if $\rho_\alpha$ and $(u_i)_\alpha$ are defined the same way in the finite-volume approximations at nodes $\nu$ and $\mu$, since the unit outward normal to face $\alpha$ relative to cell $\nu$ is minus the outward normal relative to cell $\mu$. Conservative difference approximations have many desirable accuracy properties. For example, it can be shown that difference approximations that conserve mass, momentum, and energy will calculate the correct jump conditions across shocks without having to resolve shock structure.[3]

A problem with FVMs is that it is difficult to formulate higher order FVMs. When a FVM is specialized to a finite-difference grid, the difference approximations look very much like finite-difference approximations, and one can perform Taylor-series expansions and determine the order of accuracy of the method. When more general meshes are used, however, it is unclear whether the same accuracy can be expected.

*c. Finite-Element Methods (FEMs).* FEMs[28] use a consistent spatial interpolation when evaluating all the spatial derivative terms in the fluid dynamics equations. These methods have long been popular in stress analysis problems and have recently been gaining popularity in CFD problems because of advances in the methodology. As in FVMs the computational domain is subdivided into nonoverlapping cells that in three dimensions are either arbitrary hexahedra or tetrahedra (Figs. 3, 4). Finite-element terminology is different, however, in that the cells are called elements, and the vertices of the cells are called nodes. A function $q(x_i, t)$ is represented by an expansion of the form

$$q(x_i, t) = \sum_\nu q_\nu(t) b_\nu(x_i) \;, \tag{39}$$

where the sum is over all the nodes $\nu$ in the computational domain. The $b_\nu(x_i)$ are called basis functions and have finite support, meaning that they vanish outside of some neighborhood of the node $\nu$ location $(x_i)_\nu$. They also have the properties that

$$b_\nu\left((x_i)_\mu\right) = \delta_{\nu\mu} \;, \tag{40}$$

where $\delta_{\nu\mu}$ is the Kronecker delta function, and

$$\sum_\nu b_\nu(x_i) = 1 \tag{41}$$

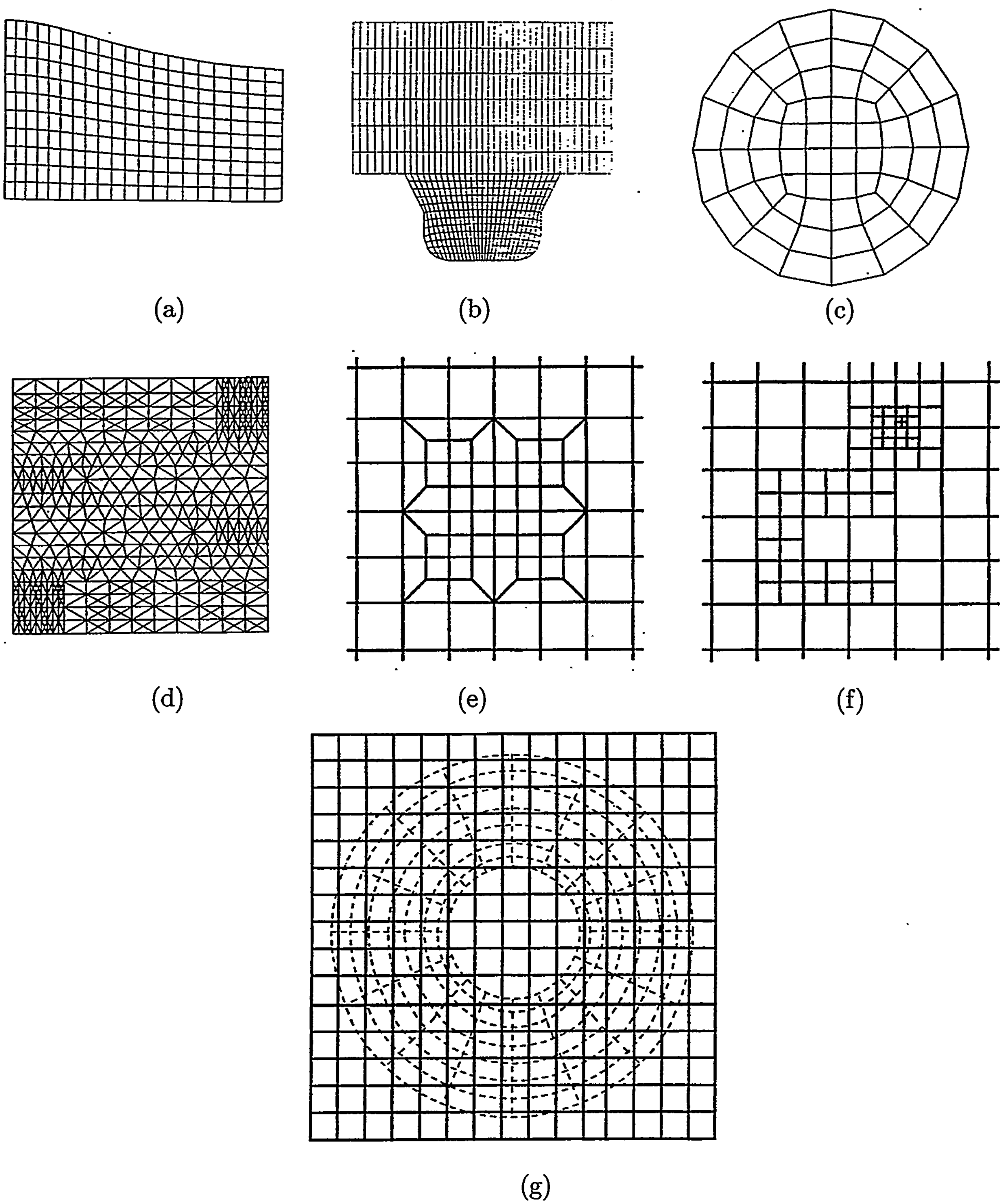


**Fig. 3.** Examples of grids used in CFD calculations. Two-dimensional examples are shown for clarity. (a) A structured grid. (b) A block-structured grid. (c) An unstructured hexahedral (quadrilateral) grid. (d) An unstructured tetrahedral (triangular) grid. (e) Local mesh refinement via a transition region on an unstructured hexahedral grid. (f) Local mesh refinement via cell splitting on an unstructured hexahedral grid. (g) A chimera grid.

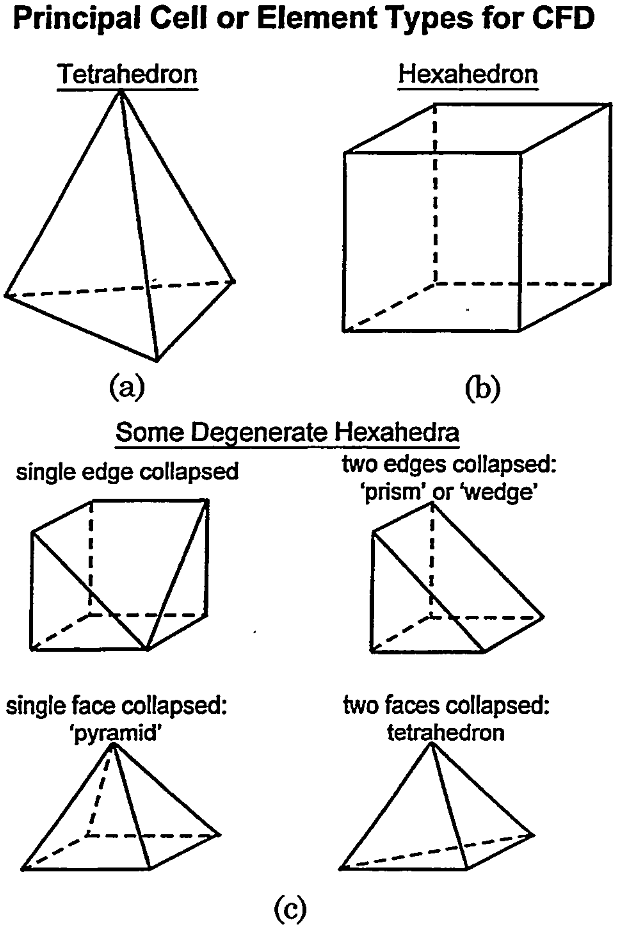

Principal Cell or Element Types for CFD
Tetrahedron
Hexahedron
(a)
(b)
Some Degenerate Hexahedra
single edge collapsed
two edges collapsed: 'prism' or 'wedge'
single face collapsed: 'pyramid'
two faces collapsed: tetrahedron
(c)

**Fig. 4.** Principal cell or element types for CFD. (a) Tetrahedron: there are four vertices or nodes, four faces, and six edges for each element. (b) Hexahedron: there are eight vertices or nodes, six faces, and twelve edges for each element. Hexahedral elements generally must remain convex (angles formed by edge and face intersections must remain smaller than 180°). (c) A sampling of possible edge and/or face degeneracies for hexahedral elements.

for all $x_i$. Linear (for tetrahedra) or trilinear (for hexahedra) basis functions give rise to second-order numerical methods in the following sense: when the finite-element grid is refined in such a way that the dimensions of the elements are reduced by a factor of 2, then the difference between the computed and exact solutions, as measured by a global integral of this difference, is reduced by a factor of 4. Higher order FEMs may be constructed by adding mid-side nodes to the elements and using nonlinear basis functions that have properties Eq. (40) and Eq. (41).[28] Because of Eq. (40) the coefficient $q_\nu(t)$ is the value of $q$ at location $(x_i)_\nu$ at time $t$.

Although there are many possibilities for determining $q_\nu(t)$, the most common method is that of the Galerkin finite element method (GFEM). In GFEM one substitutes expansions of the form Eq. (39) for each function in the fluid equations. To obtain the discrete equations associated with node $\nu$, one multiplies the resulting expanded equations by basis function $b_\nu(x_i)$ and integrates over the entire computational domain. This gives rise to a coupled system of ordinary differential equations for the functions $q_\nu(t)$. Standard numerical methods for ordinary differential equations may then be used to solve for the $q_\nu(t)$. These ordinary differential equations involve coefficients that are integrals of products of the basis functions and their derivatives. Evaluating these coefficients can be a costly step in obtaining a GFEM solution.

*d. Spectral Methods.* Like FEMs, spectral methods[28] represent a function $q(x_i, t)$ by a finite sum

$$q(x_i, t) = \sum_n c_n(t) b_n(x_i) \ , \tag{42}$$

but unlike FEMs the basis functions $b_n(x_i)$ are typically orthogonal functions with respect to some weighting function $W(x_i)$; that is

$$\iiint b_n(x_i) b_m(x_i) W(x_i) dx_i = \delta_{nm} \ . \tag{43}$$

There is no grid in a spectral method. The $c_n(t)$ are no longer the values of $q$ at nodes, but simply the coefficients of the function $q$ in an orthogonal function expansion. Ordinary differential equations for the $c_n(t)$ are obtained by a method that is similar to that of GFEM: one substitutes the expansion Eq. (42) into the fluid equations, then multiplies the resulting expanded equation by $b_n(x_i)W(x_i)$, and integrates over the computational domain.

Spectral methods are most often used in situations where suitable basis functions can be found that satisfy the boundary conditions of one's problem. When this is the case, spectral methods are very efficient for solving fluid dynamics problems. For example, direct simulations of turbulence with periodic boundary conditions invariably use Fourier series expansions[34] because of their high accuracy. Because of the difficulty of finding suitable basis functions that satisfy boundary conditions in complex geometries, spectral methods are usually used only for simple geometries.

*e. Computational Particle Methods.* Computational particles have long been used for many purposes in CFD calculations.[35] At the simplest level they are used to follow the motion of Lagrangian fluid elements for flow visualization purposes. At the other extreme, in some particle methods the fluid is completely represented by particles, each of which is endowed with a certain amount of mass, momentum, and energy. This is the case for Particle-in-Cell (PIC) methods,[36,37] and for the newer Smoothed-Particle-Hydrodynamics (SPH) methods.[38] The great advantage of the latter two methods is their Lagrangian nature. Because the Lagrangian equations are solved, numerical truncation errors are avoided that arise from finite-difference approximations to the advection terms. These are often the largest errors in approximations of the Eulerian equations. When carefully formulated, PIC and SPH method solutions can also be Galilean invariant and conserve angular momentum.[37]

A disadvantage of particle methods lies in the difficulty of calculating interactions among fluid particles—which give rise, for example, to the pressure gradient terms in the momentum equation. This difficulty manifests itself, particularly in low Mach number calculations, in particle bunching, and consequent fluctuations in advective transport. Possibly because of this difficulty, very few commercially available CFD codes use particle methods. An exception is a class of commonly used fluid/particle methods for calculating dispersed, two-phase flows,[39,40] such as occur when a liquid sprays into a gas. In these methods, computational particles represent the dispersed phase entities and only interact with each other weakly, if they do so at all.

**2. Solution of Implicit Equations.** When solving difference approximations to the steady fluid equations, or when solving implicit approximations to the unsteady equations, one must solve a large number of coupled algebraic equations for the unknown values of the fluid variables. When the equations are linear, an equation corresponding to the $i^{\text{th}}$ cell or node can be written in the form

$$\sum_j a_{ij} q_j = s_i \ , \tag{44}$$

where $a_{ij}$ are constant coefficients and $s_i$ is a known source term. The $q_j$ are the unknowns for which we wish to solve; in an unsteady problem $q_j = q_j^{n+1}$. For example, for the implicit approximation Eq. (35) to the one-dimensional linear advection equation, one can take $a_{ii} = 1.0$, $a_{ii+1} = u\Delta t/(2\Delta x)$, $a_{ii-1} = -u\Delta t/(2\Delta x)$, and $s_i = q_i^n$. Equation (44) is usually written

$$\mathbf{Aq} = \mathbf{s} \ , \tag{45}$$

where $\mathbf{A} = (a_{ij})$ is an $N \times N$ matrix of coefficients, $N$ being the number of unknowns, $\mathbf{s} = (s_i)$ is a known source vector, and $\mathbf{q} = (q_j)$ is the vector of unknowns. Because the difference approximation in cell $i$ only depends on the values of $q$ in cell $i$ and a small number of neighbors of cell $i$, only a small number of the elements of the $i^{\text{th}}$ row of matrix $\mathbf{A}$ will be nonzero, and for this reason $\mathbf{A}$ is referred to as a sparse matrix. The basic problem of implicit fluid dynamics is to solve Eq. (45) for the vector of unknowns, given a sparse matrix $\mathbf{A}$ and source vector s.

Only for problems with small $N$ can the matrix problem Eq. (45) be solved directly by Gaussian elimination. This is because although the matrix $\mathbf{A}$ is sparse, and therefore does not require much computer storage for its nonzero elements, when Gaussian elimination is used to solve Eq. (45), one finds that it is generally necessary to store in computer memory approximately $N^2$ nonzero coefficients, which is impossible in problems with a large number of cells.

Thus, iterative methods are usually used to solve the matrix problem Eq. (45). Iterative solution methods calculate a sequence of approximations $\mathbf{q}^k$ that converge to the solution $\mathbf{q}$. The exact solution is not obtained, but one stops calculating $\mathbf{q}^k$ when either the difference between successive iterates $\mathbf{q}^{k+1} - \mathbf{q}^k$, or the residual $\mathbf{A}\mathbf{q}^k - \mathbf{s}$, is acceptably small. In the past, popular iterative methods have been point-successive relaxation, line-successive relaxation, and methods based on approximate decomposition of matrix $\mathbf{A}$ into a product of lower and upper triangular matrices that can each be easily inverted.[30] Recently, these methods have largely been supplanted by two methods that have greatly reduced the computer time to solve implicit equations and thereby have made implicit methods more attractive. These more recent methods are conjugate-gradient methods[41] and multi-grid methods.[42]

When nonlinear finite difference equations are solved, the above iterative methods can be used in conjunction with Newton's method.[43] A nonlinear difference approximation can be written

$$\mathbf{F}(\mathbf{q}) = \mathbf{O} \ , \tag{46}$$

where $\mathbf{F}$ is a vector-valued function of the vector of unknowns $\mathbf{q}$. If $\mathbf{q}^k$ is the approximation to the solution $\mathbf{q}$ after $k$ Newton-iteration steps, then $\mathbf{q}^{k+1} = \mathbf{q}^k + \delta\mathbf{q}$ is obtained by solving the matrix equation

$$\frac{\partial \mathbf{F}}{\partial \mathbf{q}} \delta\mathbf{q} = -\mathbf{F}(\mathbf{q}^k) \ . \tag{47}$$

The matrix $\frac{\partial \mathbf{F}}{\partial \mathbf{q}}$ is called the Jacobian matrix. Equation (47) is of the form of Eq. (45) and can be solved by one of the iterative methods for linear equations. Thus, solution for $\mathbf{q}$ involves using an iteration within an iteration. As in the solution of nonlinear equations for single variables, convergence is sometimes accelerated by under-relaxation; that is, one takes $\mathbf{q}^{k+1} = \mathbf{q}^k + \lambda\delta\mathbf{q}$ where $\delta\mathbf{q}$ is the solution to Eq. (47) and $\lambda$ is an under-relaxation factor whose value lies between zero and one.

Newton's method is sometimes used to solve systems of coupled difference equations arising in CFD,[44] but it is often more economical for this purpose to use the Simple-Implicit Method for Pressure-Linked Equations (SIMPLE) method.[45] In the SIMPLE method, a system of coupled implicit equations is solved by associating with each equation an independent solution variable and solving implicitly for the value of the associated solution variable that satisfies the equation, while keeping the other solution variables fixed. As is implied by the acronym SIMPLE, pressure is chosen as an independent variable, and a special treatment is used to solve for pressure.[45] The equations are solved sequentially, and repeatedly, until convergence of all the equations is obtained. The SIMPLE method is more efficient if the difference equations are loosely coupled, or if some independent linear combinations of the equations can be found that have little coupling.

### C. Grid Generation for Complex Geometries

Before applying most of the CFD methods outlined above, a computational grid must be generated that fills the flow domain and conforms to its boundaries. For complex domains with curved or moving boundaries, or with embedded sub-regions that require higher resolution than the remainder of the flow field, grid generation can be a formidable task requiring more time than the flow solution itself. Two general approaches are available to deal with complex geometries: use of unstructured grids and use of special differencing methods on structured grids.

**1. Unstructured Meshes.** Figure 3 shows examples (in two dimensions) of several possible grid arrangements for CFD. In a structured three-dimensional grid (Fig. 3a) one can associate with each computational cell an ordered triple of indices $(i, j, k)$, where each index varies over a fixed range, independently of the values of the other indices, and where neighboring cells have associated indices that differ by plus or minus one. Thus, if $N_i$, $N_j$, and $N_k$ are the number of cells in the $i$-, $j$-, and $k$-index directions, respectively, then the number of cells in the entire mesh is $N_i \, N_j \, N_k$. Additionally, it is seen that each interior vertex in a structured grid is a vertex of exactly eight neighboring cells.

In an unstructured grid (Figs. 3c and 3d), on the other hand, a vertex is shared by an arbitrary number of cells. Unstructured grids are further classified according to the allowed cell or element shapes (Fig. 4). In the case of FVMs in particular, an unstructured CFD code may require a mesh of strictly hexahedral cells (Fig. 4b), hexahedral cells with degeneracies (Fig. 4c), strictly tetrahedral cells (Fig. 4a), or may allow for multiple cell types. In any case, the cells cannot be associated with an ordered triple of indices as in a structured mesh.

Intermediate between structured and unstructured meshes are block-structured meshes (Fig. 3b), in which "blocks" of structured grid are pieced together to fill the computational domain.

There are three advantages of unstructured meshes over structured and block-structured meshes. First, unstructured meshes do not require that the computational domain or sub-domains be topologically cubic. This flexibility allows one to construct unstructured grids in which the cells are less distorted, and therefore give rise to less numerical inaccuracy, compared with a structured grid. Second, local adaptive mesh refinement (AMR) is naturally accommodated in unstructured meshes by subdividing cells in flow regions where more numerical resolution is required (Fig. 3e). Such subdivision cannot be performed in structured meshes without destroying the logical $(i, j, k)$ indexing. Third, in some cases, particularly when the cells are tetrahedra, unstructured grid generation can be automated with little or no user intervention.[46] Thus, generating unstructured grids can be much faster than generating block-structured grids.

On the other hand, unstructured-mesh CFD codes generally demand higher computational resources. Additional memory is needed to store cell-to-cell and vertex-to-cell pointers on unstructured meshes, while this information is implicit for a structured mesh. And, the implied connectivity of structured meshes reduces the number of numerical operations and memory accesses needed to implement a given solution algorithm compared with the indirect addressing required with unstructured meshes.

The relative advantages of hexahedral versus tetrahedral element shapes remain subjects of debate in the CFD community. Tetrahedra have an advantage in grid generation, as any arbitrary three-dimensional domain can be filled with tetrahedra using well-established methodologies.[46] By contrast, it is not mathematically possible to tessellate an arbitrary three-dimensional domain with nondegenerate six-faced convex volume elements. Thus, each of the various automatic hexahedral grid-generation approaches that have been proposed either yields occasional degeneracies or shifts the location of boundary nodes, thus compromising the geometry.[47,48]

**2. Specialized Differencing Techniques.** In a second general approach to computing flows in complex geometric configurations, the onus of work is shifted from complexity in grid generation to complexity in the differencing scheme.[49,50,51] Structured and block-structured grids are used, but one of three numerical strategies is used to extend the applicability of these grids. The first strategy is to use so-called chimera grids[49] that can overlap in a fairly arbitrary manner (Fig. 3g). Solutions on the multiple grids are coupled by interpolating the solution from each grid to provide the boundary conditions for the grid that overlaps it. This is a very powerful strategy that handles naturally problems in which two flow regions meet at a boundary with a complicated shape or where one object moves relative to another. The second numerical strategy is to use so-called embedded boundaries.[50] Again, structured meshes are used, but the complicated boundary of the computational domain is allowed to cut through computational cells. Special numerical methods are then used in the partial cells that are intersected by the boundary. In the third strategy, local AMR is allowed by using a nested hierarchy of grids.[51] The different grids in the hierarchy are structured and have different cell sizes, but the cells in the more finely resolved grids must subdivide those of the coarser grids (Fig. 3f).

Although the second general approach affords simplicity in grid generation, it generally is less mature than the various unstructured-mesh approaches. Much development remains before these specialized differencing techniques have the robustness, generality, and efficiency to deal with the variety of problems presented in engineering applications. For the near future, then, the use of various unstructured-mesh approaches is expected to dominate in engineering applications of CFD.

## III. COMPUTATIONAL FLUID DYNAMICS FOR ENGINEERING DESIGN

We next relate the process by which the above formalisms are utilized by the industrial design engineer. Because the use of CFD in engineering design is proliferating rapidly in the 1990s, some of this information, particularly that citing specific software, will rapidly and unavoidably become dated. We believe that the benefits of providing concrete examples to the reader outweigh our concern of premature obsolescence.

CFD is one of the tools available to the engineer to understand and predict the performance of thermal-fluids systems. It is used to provide insight into thermal-fluids processes, to interpret experimental measurements, to identify controlling parameters, and to optimize product and process designs. It is the use of CFD as a design tool that is the principal focus here. In the course of a design program, an engineer typically will perform multiple CFD computations to explore the influence of geometry (hardware shape), operating conditions (initial and boundary conditions), and fluid properties. For CFD to be fully integrated into the design process, it must satisfy ever-tightening demands for functionality, accuracy, robustness, speed, and cost.

At present, most engineering CFD using commercially available software can be characterized as having high geometric complexity (domain boundaries are complex three-dimensional surfaces) and moderate physical complexity. The majority of flows considered are steady, incompressible, single-phase, and nonreacting. A common physical complexity encountered in engineering situations is turbulence, as engineering flows typically are characterized by a high Reynolds number. Turbulence is modeled using a two-equation model (standard $K/\epsilon$ or variants[27]) in most cases. Applications to transient flows with additional physical complexity and/or more sophisticated models (e.g., compressibility, multiphase, reacting, higher-order turbulence models) are increasing.

## A. The CFD Process

Let us consider the idealized component design processes shown schematically in Fig. 5. There the left-hand-side flowchart depicts a hardware-based design process, while the right-hand side represents an analysis- or math-based process. Although CFD is the single analysis tool under consideration here, the right-hand side applies equally well to other mathematical/computational tools (e.g., finite-element structural analysis) that together fall under the heading of CAE.

Both the hardware- and analysis-based processes require the generation or acquisition of geometric data, and the specification of design requirements. Here it is assumed that a three-dimensional CAD geometry model is the preferred method for geometric representation. A hardware approach then proceeds with fabrication of prototypes, followed by testing of prototypes, and evaluation of test results. Design iterations are accomplished either by direct changes to the hardware or by modification of the CAD dataset and refabrication, until the design requirements are satisfied. At that point the original CAD data must be updated (in the case of direct hardware iterations), and the design proceeds to the next component or system level where a similar process is repeated.

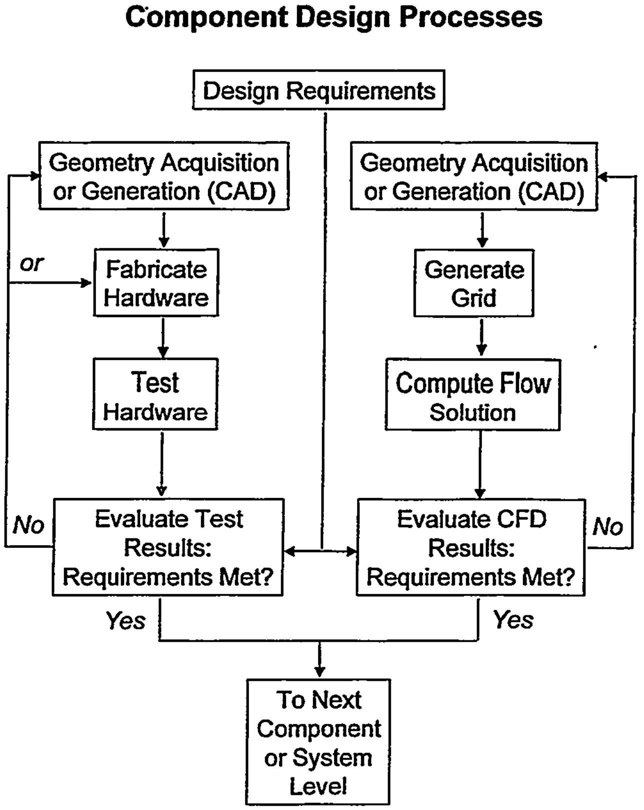

Component Design Processes
Design Requirements
Geometry Acquisition or Generation (CAD)
Geometry Acquisition or Generation (CAD)
or
Fabricate Hardware
Generate Grid
Test Hardware
Compute Flow Solution
No
Evaluate Test Results: Requirements Met?
Evaluate CFD Results: Requirements Met?
No
Yes
Yes
To Next Component or System Level

Fig. 5. Engineering component design processes. Left-hand side depicts a hardware-based approach; right-hand side is an analysis- (CFD-) based approach.

Analysis-based design (here, CFD) is not fundamentally different. Mesh generation replaces hardware fabrication, computer simulation substitutes for experimental measurement, and postprocessing diagnostics are needed to extract relevant physical information from the vast quantity of numerical data. To the extent that relatively simple design criteria are available and the component lends itself to a parametric representation, the design-iteration loop can be automated using numerical optimization techniques.[52] Automated computer optimization with three-dimensional CFD remains a subject of research; in most engineering applications, determination of the next design iteration remains largely a subjective, experience-based exercise.

Analysis-based design can be faster and less costly compared with hardware build-and-test. If this is not yet the case in a particular application, it most likely will be true at some point in the future. Thus, analysis affords the opportunity to explore more design possibilities within specified time or budget constraints. Advances in rapid prototyping systems[53] and other fabrication technology mitigate this advantage to some extent.

A second advantage of analysis is that more extensive information can be extracted compared with experimental measurements. CFD yields values of the computed dependent variables (e.g., velocity, pressure, temperature) at literally thousands or even millions of discrete points in space and (in time-dependent problems) in time. From this high density of information can be extracted qualitative and quantitative pictures of flow streamlines and three-dimensional isopleths of any computed dependent variable. For time-dependent problems, animation or "movies" reveal the time evolution of physical processes. Application-specific "figures-of-merit," including total drag force, wall heat flux, or overall pressure drop or rise, can be computed. Examples are given in the case studies that follow. Experimental measurements, on the other hand, traditionally have been limited to global quantities or to values of flow variables at a small number of points in space and/or time. Thus in principal, much more complete information is available from CFD to guide the next design iteration. An important caveat is that this additional information is useful only to the extent that it accurately and reliably represents the actual hardware under the desired operating conditions. In most applications of CFD today, there are sufficient sources of uncertainty that abandonment of experimentation is unwarranted. Recent progress in two- and three-dimensional experimental diagnostics (e.g., particle-image velocimetry for velocity fields,[54] laser-induced fluorescence for species concentrations[55]) is enabling higher spatial and/or temporal measurement densities in many applications.

In Fig. 6, the CFD process is modeled as a four-step procedure: (1) geometry acquisition, (2) grid generation and problem specification, (3) flow solution, and (4) postprocessing and synthesis. Depending on the level of integration in the software selected, four (or

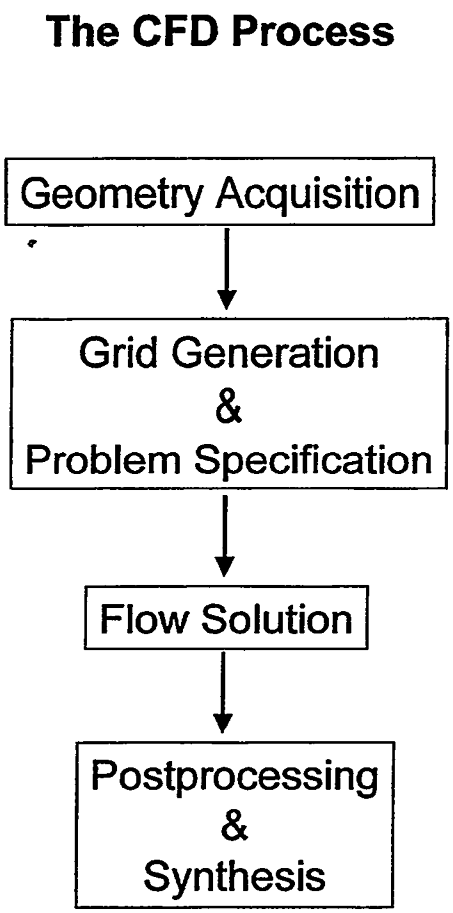

The CFD Process
Geometry Acquisition
Grid Generation & Problem Specification
Flow Solution
Postprocessing & Synthesis

**Fig. 6.** The CFD process. Examples of available software are given in Table 2.

more) distinct codes may be needed to accomplish these tasks. Some vendors offer fully integrated systems. For the purpose of exposition, we treat each step separately.

**1. Geometry Acquisition (CAD).** The principal role of CAD software in the CFD process is to provide geometric definition of the bounding surfaces of the three-dimensional computational domain. The computational domain of interest in CFD generally is everything *external* to the solid material; this conveniently might be thought of as the negative of a finite-element structural solid model. Several CAD packages are available commercially; examples are listed in Table 2. These codes are designed primarily with the design and fabrication of three-dimensional solids in mind, and have considerable functionality that is not of direct relevance for CFD.[56]

The various CAD packages use different internal representations for curves (one-dimensional objects), surfaces (two-dimensional objects), and solids (three-dimensional objects). The surfaces needed for CFD, for example, may be represented using one of several tensor-product polynomial or spline representations in a two-dimensional parametric space.[57,58] Any of these representations generally suffice for CFD; most FDM, FVM, and FEM solution methodologies in today's engineering CFD codes require at most

Table 2. Examples of CFD software available in the United States. This partial listing was extracted from information maintained by several computer hardware and software companies on the Internet early in 1997. Further information on each company and/or code can be found by initiating a network keyword search. Additional information is provided for some companies in Table 3.

**Geometry Acquisition**
ICEM CFD
Unigraphics
CATIA
CADDS
I-DEAS
IEMS
Pro-Engineer
Patran
AutoCAD

**Grid Generation**
ICEM CFD
GridGen
Patran
Hexar
CFD-GEOM

**Postprocessing (3D Visualization)**
ICEM
Patran
Fieldview
Application Visualization System - AVS
DATA VISUALIZER
EnSight
FAST
PLOT3D/TURB3D
MPGS
CFD-VIEW

linear interpolation between the discrete points (nodes or vertices) representing the surface. However, spectral-element methods[59] and some other high-order orthogonal basis function expansions require a level of surface definition that generally is not available from current commercial CAD systems; this presently limits the application of such methods to simple geometric configurations.

The need to move geometry models among different CAD systems having different internal representations led to the establishment of standards for external geometric data exchange. An early standard supported by most CAD software is the Initial Graphics Exchange Specification (IGES).[60] Most CAD to CFD interfaces now operate by extracting the outer surfaces and writing an IGES file of "trimmed" B-spline surfaces. Newer standards such as Standard for the Exchange of Product model data (STEP) are merging with IGES and supplanting it; existing standards are evolving rapidly and new standards are developed as needed. Other external data formats commonly used in the CAD/CAE arena include Stereo Lithography (STL), where surfaces are processed into a set of triangular facets, cloud-of-points (a set of random points in three-dimensional space), and DES (a set of piece-wise linear curves Describing a Surface).

The set of raw surfaces extracted from the CAD model usually requires additional processing before it is suitable for CFD grid generation. The extracted surfaces may not define a closed three-dimensional domain (gaps), there may be more than one surface at a physical location (overlaps), and there simply may be too much geometric detail to be practical for CFD. Modern CAD and grid-generation systems provide fault tolerance and a variety of tools to "clean up" the extracted surfaces prior to grid generation. This cleanup step is labor intensive, and often is the single most time-consuming element of the CFD process.

**2. Grid Generation and Problem Specification.** The second step in the CFD process is to generate a computational mesh. This might be accomplished using the same software as for geometry acquisition, or a separate code. The grid must satisfy three general requirements: (1) it must be compatible with the selected flow solver; (2) it must be sufficiently fine to satisfy accuracy requirements; and (3) it must be sufficiently coarse to satisfy computational resource limitations.

For an unstructured mesh, the minimum information that must be provided from the grid-generation step is the location of each node or vertex, and a description of connectivity among the vertices. A complete problem prescription for CFD requires, in addition, the specification of initial and boundary conditions for all flow variables (e.g., velocity, pressure, temperature), fluid properties, and any model and numerical parameters. Other code- and application-specific information also may be needed. Because both geometry and grid information are available at the grid-generation stage, this is the most natural time to tag volumes for initial conditions and material properties and surfaces for boundary conditions (e.g., specify which surfaces represent walls, inflow boundaries, etc.). Specific initial values for each dependent variable at each interior cell or vertex, boundary values for each boundary element face or vertex, and fluid properties may be set either in the grid-generation software itself or in a separate "pre-processor" provided for the specific CFD code. For present purposes, the pre-processor is considered to be part of the flow solver. Model constants and numerical parameters are specified to the flow solver directly.

Fully automatic tetrahedral-mesh generation is available in a number of commercial and public-domain codes.[46] (See Table 2.) Early generations of automated hexahedral, hexahedral-with-degeneracies, and hybrid hexahedral/tetrahedral strategies (requiring varying levels of manual intervention) also are available at the time of this writing.[47,48] (See Table 2.) However, a high level of manual intervention still is required to generate high-quality meshes for CFD. This is particularly true in the case of tetrahedral meshes in the vicinity of solid walls. Here we define a "high-quality" mesh as one that yields high numerical accuracy for low computational effort (memory and CPU time). This is quantified by performing multiple computations of a single-flow configuration using different meshes, and computing the error in each with respect to a benchmark numerical or experimental solution. Fletcher[32] and Sengupta et al.[61] discuss modern mesh generation techniques for CFD.

Regardless of the specific methodology used to generate the mesh, it is important that any grid-generation software for CFD maintain separate data structures for geometry definition and for the computational mesh. This ensures that design changes (modifications to CAD surfaces) can be made without redoing the domain decomposition, that boundary conditions can be reset without regenerating the grid, and that mesh density and distribution can be changed independently of the geometry.

**3. Flow Solution.** Most contemporary CFD solvers available to the industrial design engineer use either finite-volume or finite-element discretization, with SIMPLE-like iterative pressure-based implicit solution algorithms. Unstructured meshes of primarily hexahedral elements (with limited degeneracies) have been prevalent in most finite-volume formulations to date, although the grid-generation advantages of tetrahedra are leading to an increase in the usage of that element type.

Default or recommended values of numerical parameters are provided by each flow solver. New and/or unusual applications often require experimentation in selecting values of numerical parameters to obtain a stable, converged solution. For the solution methodologies commonly used today, parameters include choice of advection scheme (e.g., the degree of upwinding), convergence criteria for linear equation solvers and pressure iterations, time-step control (for transient problems), mesh adaptation (where available), and other method-specific controls. For this reason, the CFD practitioner needs to have a working knowledge of the information covered in the "Fundamentals" section above. With these caveats, flow solution is the step requiring the least manual intervention. The engineer can monitor the solution as it progresses using the available diagnostics, which are discussed next.

**4. Post-Processing and Synthesis.** Viewing and making sense of the vast quantities of three-dimensional data that are generated in CFD is a challenging task. Many software packages have been developed for this purpose, both for structured and unstructured meshes (Table 2). All provide considerable flexibility in setting model orientation, in passing cutting planes and/or lines through the computed solution, and in displaying the computed vector and scalar fields. Postprocessors have varying levels of "calculator" capability for computing quantities not supplied directly from the CFD solution, such as vorticity or total pressure. Many allow transient animation to accommodate time-dependent data. Most modern packages provide both a graphics-user interface (GUI) and a save file/read file capability, the latter to allow the user to replicate a particular view of interest for multiple data sets.

Such direct inspection of the computed fields provides detailed insight into flow structure in the same sense as a high-resolution flow visualization experiment. In this respect and others, it had been argued that CFD is more akin to experiment than to theory. Features such as an undesirable flow separation, for example, might provide the engineer with sufficient information to guide a modification to the device geometry for the next design iteration. The connection between device performance or design requirements and the full three-dimensional flow field often is not obvious, however; considerable effort may be required to extract meaningful figures-of-merit from the numerical solution.

A judicious development of diagnostics is necessary to advance CFD from sophisticated flow visualization tool to scientifically based design tool. Quantitative information of direct relevance to the design is needed to drive design changes towards satisfaction of the design requirements. Such diagnostics are application-specific, and have received relatively little attention by CFD researchers and code developers. Examples of diagnostics to extract physical insight and to assess numerical accuracy can be found in Haworth, El Tahry, and Huebler.[62]

## B. Examples of Engineering CFD

Application areas that have been particularly active in their use of CFD include aircraft and ship design, geophysical fluid flows, and flows in industrial devices that involve energy conversion and utilization. A comprehensive list of the applications of CFD would be difficult to compile, and no attempt to do so is made here. Instead, specific case studies are cited with several purposes: (1) to illustrate the scope and state-of-the-art in engineering CFD, (2) to highlight issues that arise in engineering applications of CFD, and (3) to introduce some specific CFD software that is widely used in industry.

1. **Internal Duct Flow.** Many internal flows of engineering interest can be broadly categorized as complex duct flows. The principal physical complexity is turbulence, particularly as it influences flow separation. A related numerical issue is mesh resolution, especially in the vicinity of walls. Flow losses (pressure drop and separations), flow distribution among multiple branches, mixing, and heat transfer may be important in such configurations.

Two examples of steady, incompressible CFD simulations are given in Fig. 7.[63,64] Figures 7a and 7b show a simplified automotive heating, ventilation, and air-conditioning (HVAC) duct. This is taken from a validation study, which also contains experimental measurements.[63] Results of this kind have allowed engineers to identify flow separations and poor flow distribution among branches; optimized designs for lower pressure drop and more favorable flow distribution are identified using CFD prior to hardware fabrication.

A second internal flow configuration (Fig. 7c) illustrates the geometric complexity that often arises in engineering applications. Figure 7c shows surface heat transfer coefficients from computations of flow in the coolant passages of a production automotive engine block. Such results are used to identify potential "hot spots" and to modify flow passages for more uniform cooling.

2. **External Aerodynamics.** External flows comprise a second broad category of engineering interest. This includes flows around immersed bodies such as aircraft, ships, submarines, and automobiles. Bluff-body aerodynamics is particularly challenging; the accurate computation of separation, which may be highly unsteady, is key to predicting lift and drag.

Examples of computations and measurements for idealized three-dimensional bluff bodies are shown in Figs. 8(a) and (b).[65,66,67,68] A computational challenge is to capture the sudden drop in drag coefficient at a slant angle of about 30° (Fig. 8b). Computations of flow over realistic vehicle shapes also are feasible using modern CAD/grid generation tools (Fig. 8c).[69] In all cases shown here, the flows have been computed as steady and incompressible using standard Reynolds-averaged turbulence models to account for unsteadiness.

3. **Manufacturing Processes.** Increasing attention is being focused on the design and analysis of engineering processes. Heat transfer accompanied by melting

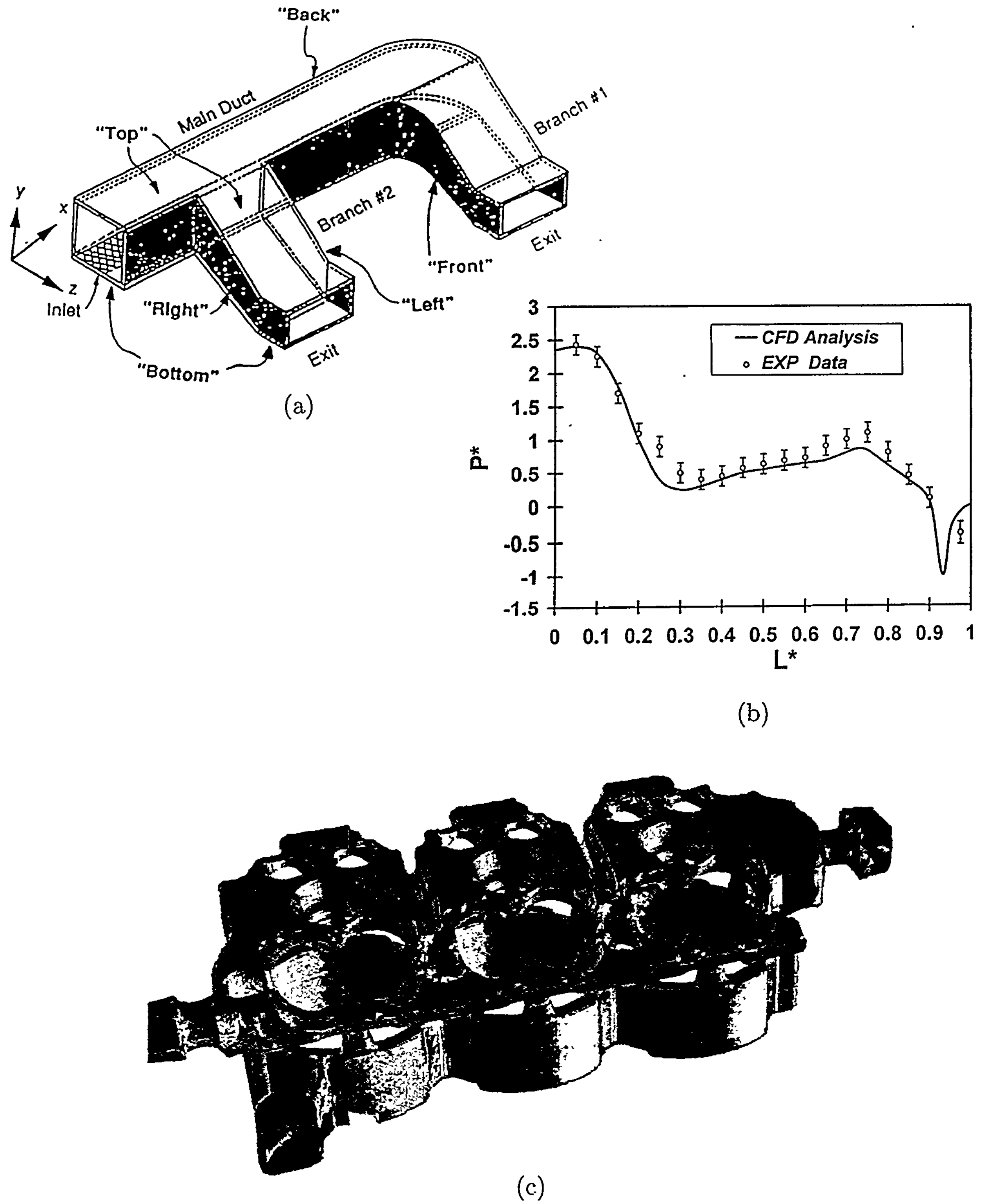
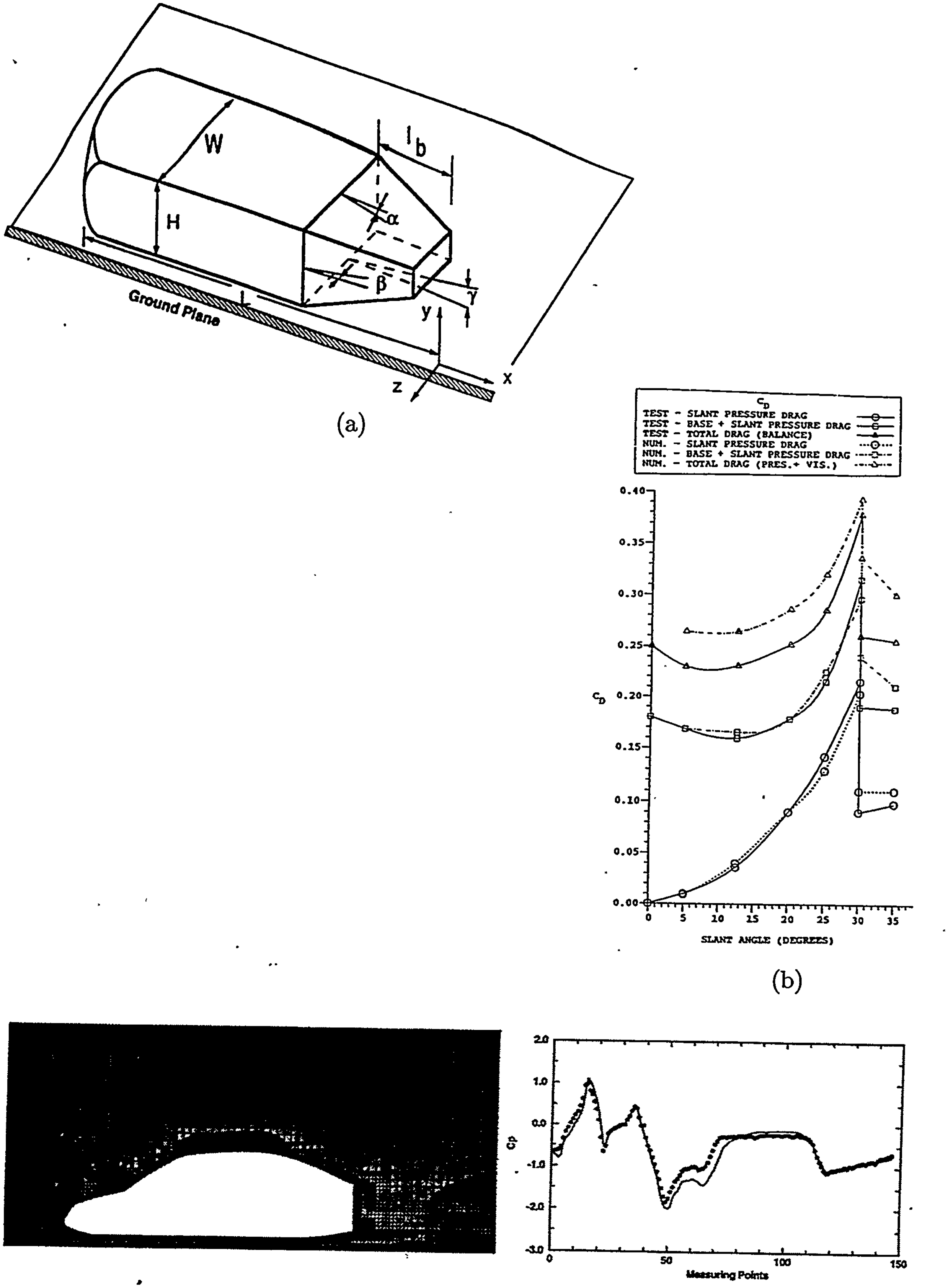


Fig. 7. Examples of internal flow CFD. (a) A simplified automotive HVAC duct.[63] (b) Measured and computed static pressure distributions along the 'Top' surface of the Main Duct and Branch #1.[63] (c) Computed surface heat transfer coefficients for a production automotive engine block.[64]



(a)

(b)

(c)

**Fig. 8.** Examples of external flow CFD. (a) Generic three-dimensional bluff body for validation studies.[65] (b) Computed drag coefficient versus slant angle (angle $\alpha$).[66] (c) Measured and computed pressure coefficients along a production car body.[69]

and solidification occurs in manufacturing processes including casting, injection molding, welding, and crystal growth. In such applications, heat conduction in the solid is coupled to convective heat transfer in the fluid. The solid–liquid interface moves with time, and its location needs to be tracked as a propagating three-dimensional surface in a CFD solution. Also, fluid properties may be highly temperature dependent and non-Newtonian, including phase changes. Here we cite metal casting as one example of such an application.

Casting is a process in which parts are produced by pouring molten metal into a cavity having the shape of the desired product. Figure 9(a) is a schematic of a typical sand casting configuration.[70] Once the two halves of the mold have been made, they are carefully aligned, one over the other, with the aid of pins and bushings in the sides of the molding boxes, to create the complete mold. Aside from the casting cavity itself, other features are also incorporated into the finished mold, such as the pouring basin, downsprue, runners, and ingates that conduct the molten metal into the casting cavity. Risers, or reservoirs of molten metal that remain molten longer than the casting, are needed with most metals and alloys that undergo liquid shrinkage as the casting solidifies. These are placed at critical locations in the mold, generally at heavier sections and areas remote from the ingates. Once the casting has been poured and allowed to cool, and after it has been withdrawn from the sand mold, these appendages are removed before the casting undergoes various finishing operations.

Fluid flow plays two important roles in the casting process. First, and most obviously, the flow of molten metal is necessary to fill the mold. Second, and less obvious, are the effects of convective fluid flow during solidification of the casting. It is the task of the foundry engineer to design gating and riser systems (Fig. 9a) that ensure proper filling and solidification, and CFD is playing an increasingly important role in this field. Proper designs result in less scrap and less casting repair at the foundry. An example of a computational mesh and computed solidification times is given in Figs. 9b and 9c.[71] One CFD package that has been developed specifically for the modeling of flow and thermal phenomena in casting applications is MAGMASOFT$^{TM}$.[72] Recent references from the literature give ample evidence of the vast amount of CFD activity that is taking place in this area.[73,74]

**4. Building Interior.** Figure 10 shows an example of CFD applied to building HVAC design. In this case, the geometric configuration is relatively straightforward. The computational domain represents the interior of the Sistine Chapel at the Vatican. The purpose of the analysis was to determine the placement and angles of air-conditioning ducts to minimize deposition of contaminants on the newly-restored surfaces of Michelanglo's frescos. The creation of two separate recirculation cells for the configuration shown in

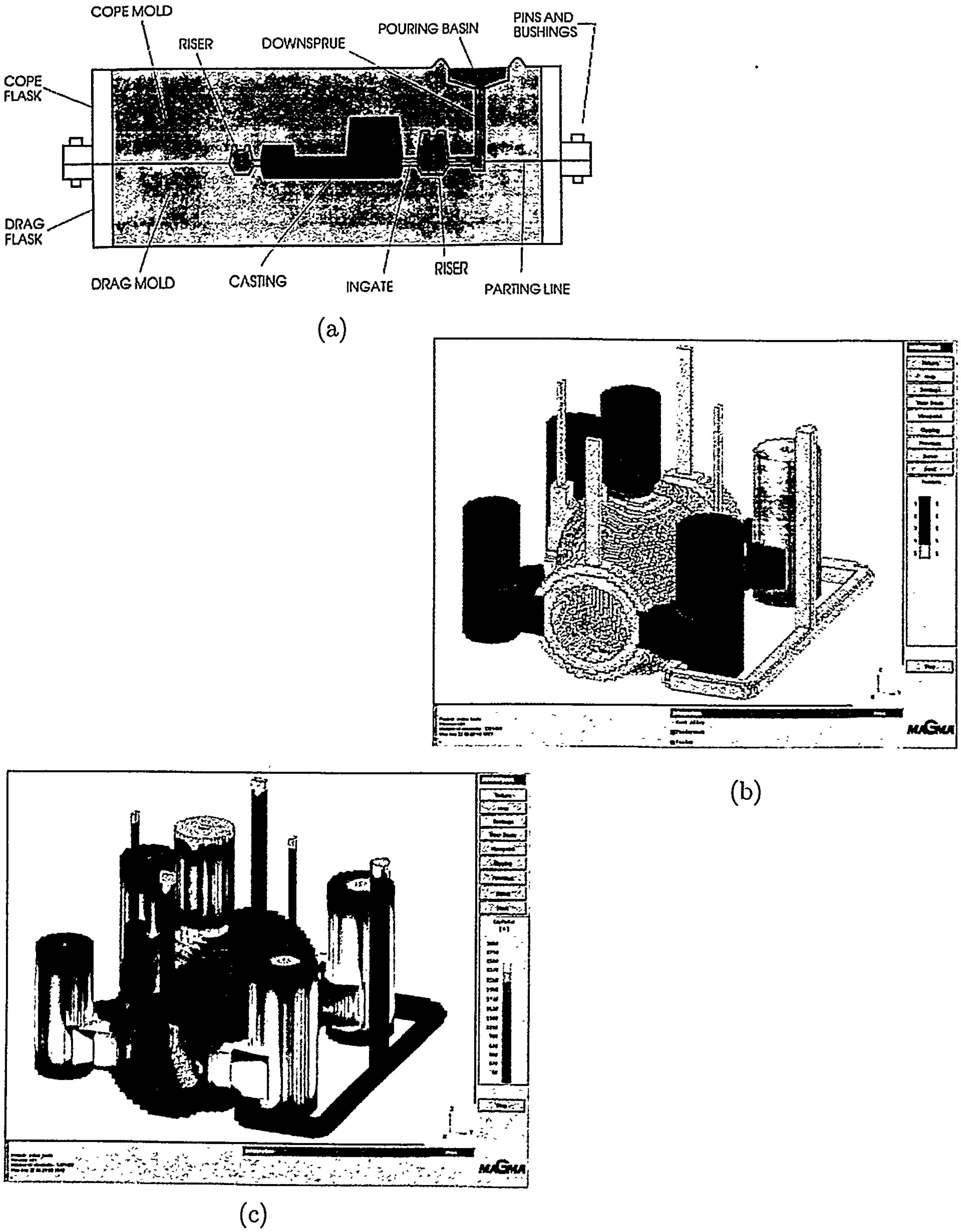


Fig. 9. A metal casting simulation. (a) A typical sand casting configuration.[70] (b) Automatically generated mesh (five million elements) for casting and cooling channels.[71] (c) Computed local solidification times, which range from 1 to 3000 seconds.[71]

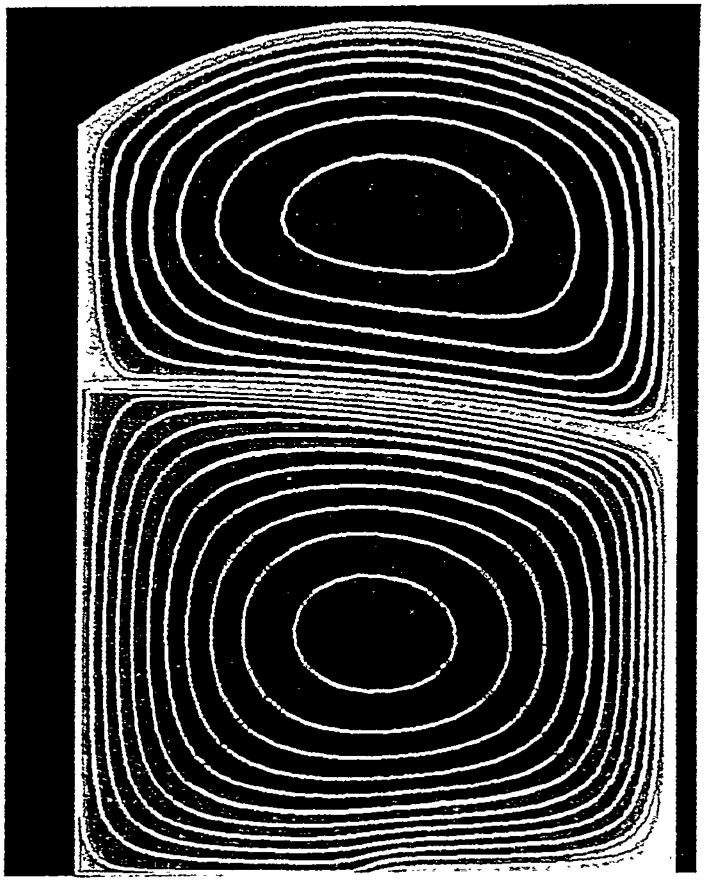


**Fig. 10.** Flow in the interior of the Sistine Chapel for one possible air-conditioning system configuration. Calculations were done using the FIDAP finite-element CFD code.[75]

Fig. 10 was deemed to be favorable for isolating traffic-borne particles created by chapel visitors in the lower half, from the fresco surfaces along the upper walls and ceiling.

**5. Environmental Flow.** Environmental flows include natural phenomena, such as atmospheric weather patterns and ocean currents (Fig. 2), and flows of molten rock beneath the earth's crust. Engineering design issues arise in the extraction of fossil fuels and other materials from the earth, in bridge and building design, and in the treatment and dispersal of wastes from electrical utilities, transportation systems and vehicles, and industrial manufacturing plants. Such problems typically are characterized by a coupling of natural convection (resulting from temperature and/or concentration gradients) with other forces, in many cases including the earth's rotation.

**6. Internal Combustion (IC) Engine.** For our final example, we show a few results from transient computations of flow, fuel spray, and combustion in a reciprocating internal combustion engine (Fig. 11).[76,77] This application includes geometric complexity (complex internal flow passages, moving boundaries—piston and valves), physical complexity (turbulence, combustion, multiphase flow), and numerical challenges (deforming mesh, large density and fluid property variations, coupled Eulerian/Lagrangian algorithms). This represents an application area of CFD that lies at the frontier between research and engineering application.

Of particular interest in a homogeneous-charge spark-ignited engine is the tradeoff between flow losses and in-cylinder flow "structure." Flow losses (induction system pressure drop) reduce the quantity of air that can be drawn into the cylinder, lowering engine peak power. A coherent large-scale in-cylinder flow structure tends to yield higher combustion efficiency, but generation of highly structured flow (e.g., a large-scale swirl about the cylinder axis) generally implies a pressure-drop penalty. These tradeoffs can be quantified and optimized using CFD.[76] The computations of Figs. 11(a) and (b) were performed on unstructured meshes of up to 250 000 predominantly hexahedral cells; computation through one crankshaft revolution required about 150 equivalent single-processor Cray Y-MP CPU hours.

Flame propagation for a production four-valve-per-cylinder automotive engine is shown in Fig. 11(c). Flame shapes and burn rates are tailored by changing the intake port, intake valve, and combustion chamber geometry. A good design generally is one having favorable spark-gap conditions and a flame that propagates uniformly outward to reach all solid walls at the same instant.

Direct-injection diesel and gasoline engines, wherein liquid fuel is injected directly into the combustion chamber, are of interest for their high fuel economy potential. Here mixing and fuel stratification are key issues affecting combustion performance; CFD is one tool that is being used to explore the influence of flow structure, injector placement, and injection characteristics on engine combustion performance (Figs. 11d, 11e).[77]

## IV. ISSUES AND DIRECTIONS FOR ENGINEERING CFD

### A. Geometric Fidelity

Geometric fidelity between hardware and the computational mesh is crucial to obtaining accurate results. It is characteristic of the highly nonlinear flow equations that small geometric perturbations can result in large changes to the flow field. One example is shown in Fig. 12.[78] Significantly different flow structure and mixing result when the fraction-of-a-millimeter gap between piston and cylinder liner (the "top-ring-land crevice") is included in the mesh compared with when it is ignored. With a top-ring-land-crevice, the flow entering the cylinder attaches to the cylinder wall and flows parallel to the wall for an extended time; in the absence of this crevice, the entering flow quickly adopts the port angle on entering the cylinder. This highlights the importance of maintaining a consistent three-dimensional representation of the hardware at all stages of design, analysis,

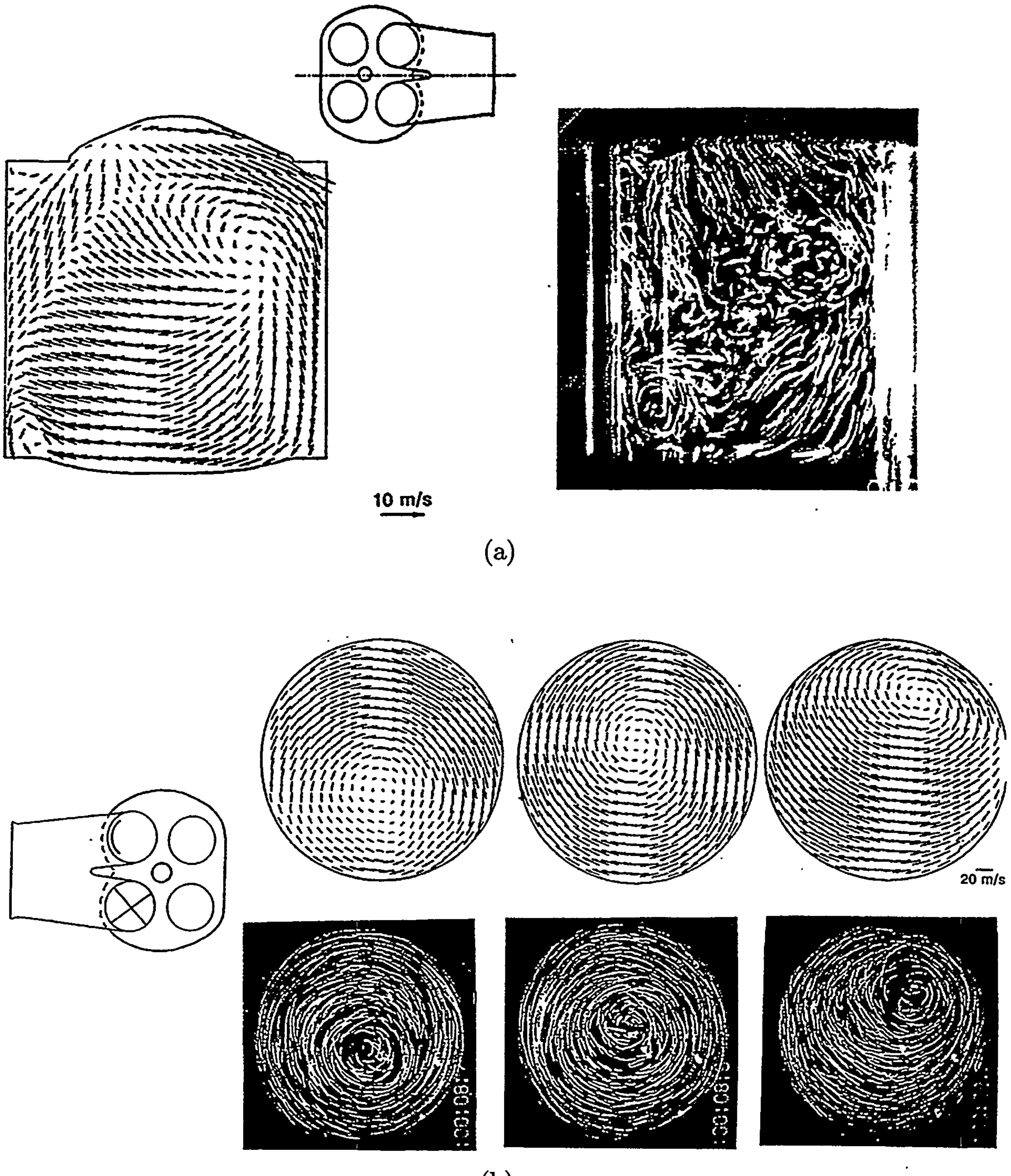


**Fig. 11.** Examples of CFD· for in-cylinder processes in reciprocating IC engines. (a) Instantaneous computed and measured induction flow at piston bottom-dead-center for a port and chamber configuration yielding weakly structured in-cylinder flow.[76] (b) Instantaneous computed and measured induction flow at piston bottom-dead-center for a port and chamber configuration yielding a highly structured in-cylinder flow.[76]

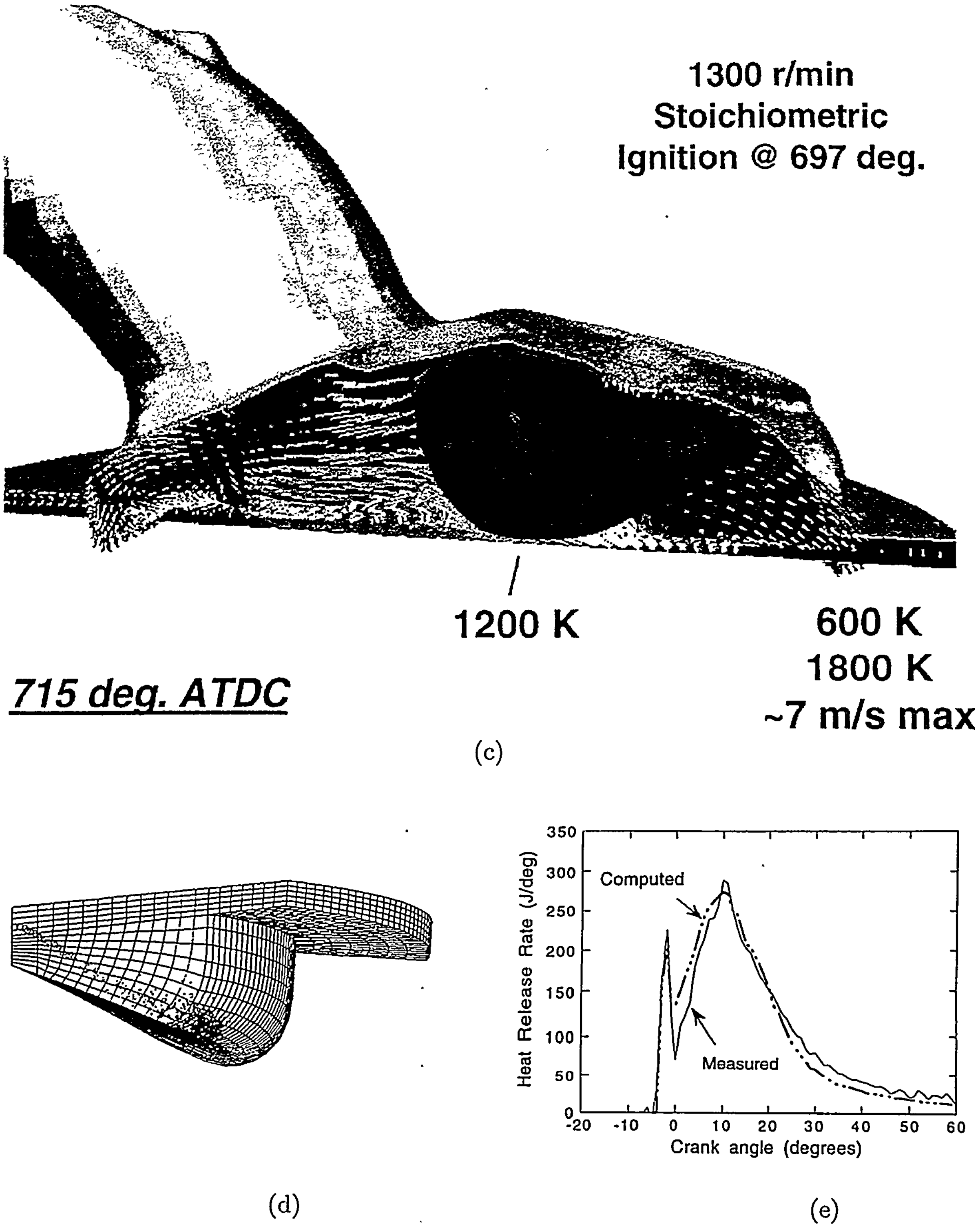


Fig. 11 (continued). Examples of CFD for in-cylinder processes in reciprocating IC engines. (c) Instantaneous computed velocity field and flame propagation near piston top-dead-center for a production four-valve-per-cylinder engine. (d) Instantaneous computed fuel spray for a direct-injection diesel engine.[77] (e) Computed and measured heat release for a direct-injection diesel engine.[77]

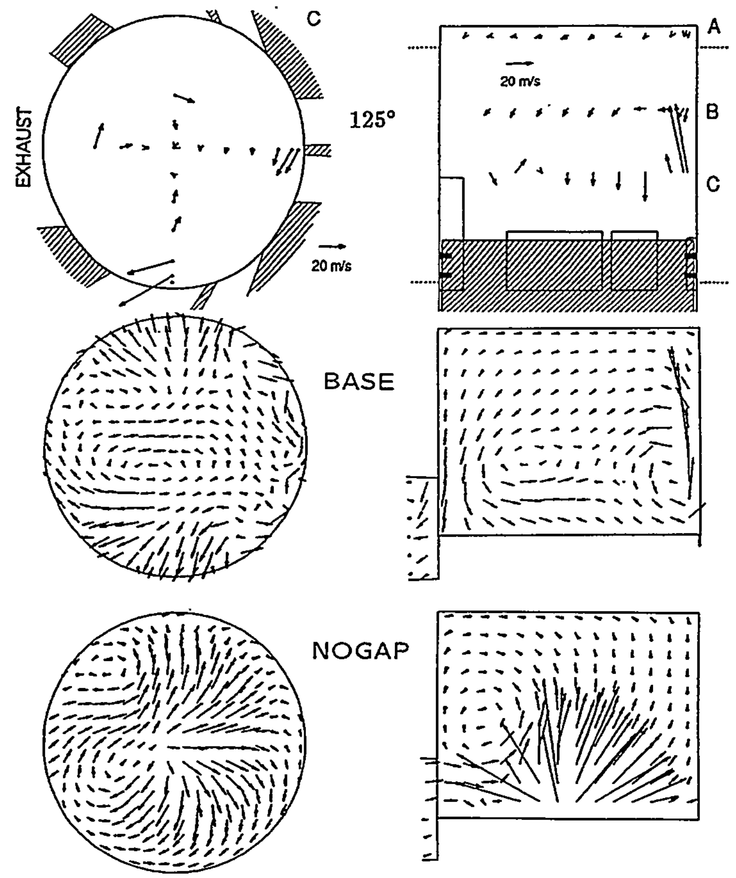


Fig. 12. Computed and measured ensemble-mean velocity fields on two-dimensional cutting planes at 125° after piston top-dead-center for a ported two-stroke-cycle engine.[78] Computational results with and without a top-ring-land crevice are shown. (a) Measured (top). (b) CFD with top-ring-land crevice (middle). (c) CFD without top-ring-land crevice (bottom).

and fabrication. The CFD practitioner should be wary of compromising the geometry in favor of grid-generation expediency, particularly in applications where he or she has little previous experience.

### B. Numerical Inaccuracy

Meshes of hundreds-of-thousands of computational cells are common in transient engineering applications of CFD today, and several millions of cells are being used in steady-state computations. Even so, numerical inaccuracy remains an issue for three-dimensional CFD. A mesh of 1 000 000 cells corresponds to just 100 nodes in each coordinate direction in a three-dimensional calculation. With the low-order numerics that characterize engineering CFD, this is sufficient to resolve a dynamic range of about 1 order of magnitude (a factor of 10) in flow scales.

Rapid progress is being made both in discretization schemes for tetrahedral meshes, and in automated grid generation for (primarily) hexahedral meshes; it is unclear at this time which will become dominant in engineering CFD.

### C. Physical Models

The physical models used to represent turbulence, combustion, sprays, and other unresolvable phenomena are a third source of uncertainty in CFD. Turbulence modeling, in particular, is an issue that affects nearly all engineering applications. Research towards improved models continues. Much new physical insight into turbulence is itself being derived from large-scale numerical simulations.[79]

In many high Reynolds-number engineering applications in which the instantaneous flow is highly transient and three-dimensional, turbulence models can be used to reduce the problem to one of steady flow, provided that the *mean* quantities of interest are time-independent. This reduces the computational requirements considerably, and provides results of acceptable accuracy in many cases. However, as engineering design requirements tighten, there is an increasing number of problems that demand a full three-dimensional transient treatment. Models still are needed to account for scales smaller than those that can be resolved numerically, but subgrid-scale turbulence models are used instead of Reynolds-averaged models. The resulting three-dimensional time-dependent simulations in this case are referred to as large-eddy simulations (LES).[80] The use of LES in engineering design is expected to proliferate rapidly. Examples of current applications of interest include acoustics and aerodynamic noise[81] and in-cylinder flows in engines.[82]

In principle each of these three sources of uncertainty can be isolated and quantified in simple configurations where a second source of data (e.g., experimental measurements) is available. It is more difficult in engineering applications of CFD to isolate and to quantify these errors to obtain meaningful estimates of error bounds. Early in the history of three-dimensional CFD, discrepancies between CFD and experiments generally were attributed to the turbulence model. The importance of the other sources of uncertainty, and numerical inaccuracy in particular, has been more widely acknowledged recently.[62,83,84] In our experience, most discrepancies between computations and measurements for single-phase nonreacting flows in complex configurations are traced to geometric infidelity or to inadequate mesh resolution (in cases where they have been traced at all).

### D. User Expertise

CFD codes generally require more experience on the part of the user than other, more mature, CAE tools (e.g., linear FEM structural analysis). "General purpose" CFD software provides a large number of numerical parameters and problem specification options. In steady-flow problems, results should be independent of the choice of initial conditions, but different initial conditions may lead to different steady solutions when time-marching to the steady state. The choice of computational domain and specification of boundary conditions always are important, both for steady and time-dependent flows. Minimal user experience may suffice to obtain a reliable solution for steady incompressible flow in a benign geometric configuration, but considerable expertise is needed in problem specification and in results interpretation for complex flows.

### E. CFD and Experimental Measurements

The engineering and scientific community typically accepts measurements from experiments as being more reliable than similar information generated by a CFD calculation. This is the reason for the strong emphasis placed by the profession on "validating" CFD results. While it is true that there are many sources of uncertainty in CFD, the same is true of experiments, particularly for complex systems (e.g., the in-cylinder flow of our last example). When comparing CFD results with measurements for such complex engineering problems, it is more appropriate to approach the exercise as a "reconciliation" rather than a "validation," as the latter implies that the experiment provides the "correct" value.

### F. Interdisciplinary Analysis

In this overview, CFD has been considered as an isolated analysis tool. This is satisfactory only to the extent that one can reasonably prescribe boundary conditions that are independent of the flow solution itself.

For example, in the coolant-flow analysis of Fig. 7(c) temperature boundary conditions might be prescribed from a separate finite-element structural analysis, but the temperature field in the solid depends on the coolant flow itself. One can alternate through a sequence of CFD and thermal structural analyses, taking the most recent boundary conditions available at each step, to obtain a solution that effectively is coupled. A single direct computation of the coupled solution would be more satisfactory, however. In this case, a coupled fluid/heat conduction analysis is feasible because many CFD codes provide a so-called conjugate heat transfer capability.

More difficult are cases where fluids and solids interact in a manner that changes the shape of the flow domain. Flow/structure interactions including deformations are important, for example, in some aircraft design problems or in applications where there is significant thermal distortion. Interdisciplinary analysis tools are becoming available for these problems and will see more widespread use in the future.

### G. Future of Engineering CFD

Most contemporary commercial CFD codes start from a discretization of the continuum equations of fluid mechanics and require a computational mesh of discrete cells or elements. An alternative is to approach CFD from a kinetic theory point of view. For example, an (essentially) grid-free Lagrangian-particle method has been developed and implemented.[85] It is too early, at the time of this writing, to speculate on the future of this approach for engineering design. Computations have been reported for configurations, including external flow over simplified and realistic vehicles.

Active research areas for CFD include automated mesh generation, numerical algorithms for parallel computer architectures, linear equation solvers, more accurate and stable discretization schemes, automatic numerical error assessment and correction, improved solution algorithms for coupled nonlinear systems, new and enhanced physical models, more sophisticated diagnostics, interdisciplinary coupled structures/fluids analysis, optimization algorithms, and coupling of three-dimensional CFD into systems-level models.

In the ideal math-based design process, CFD is one part of a multidisciplinary CAE approach, and the full system (versus isolated component) is considered. Grid generation is fully automated to ensure a high-quality (initial) mesh. The flow solver selects all numerical parameters, and provides automated solution-adaptive mesh refinement to a specified level of error or allowable computational resource (time or cost). Solution diagnostics provide information of direct relevance to the design requirements. And, automated design optimization through modifications to the geometry and/or operating conditions proceeds until design requirements are met.

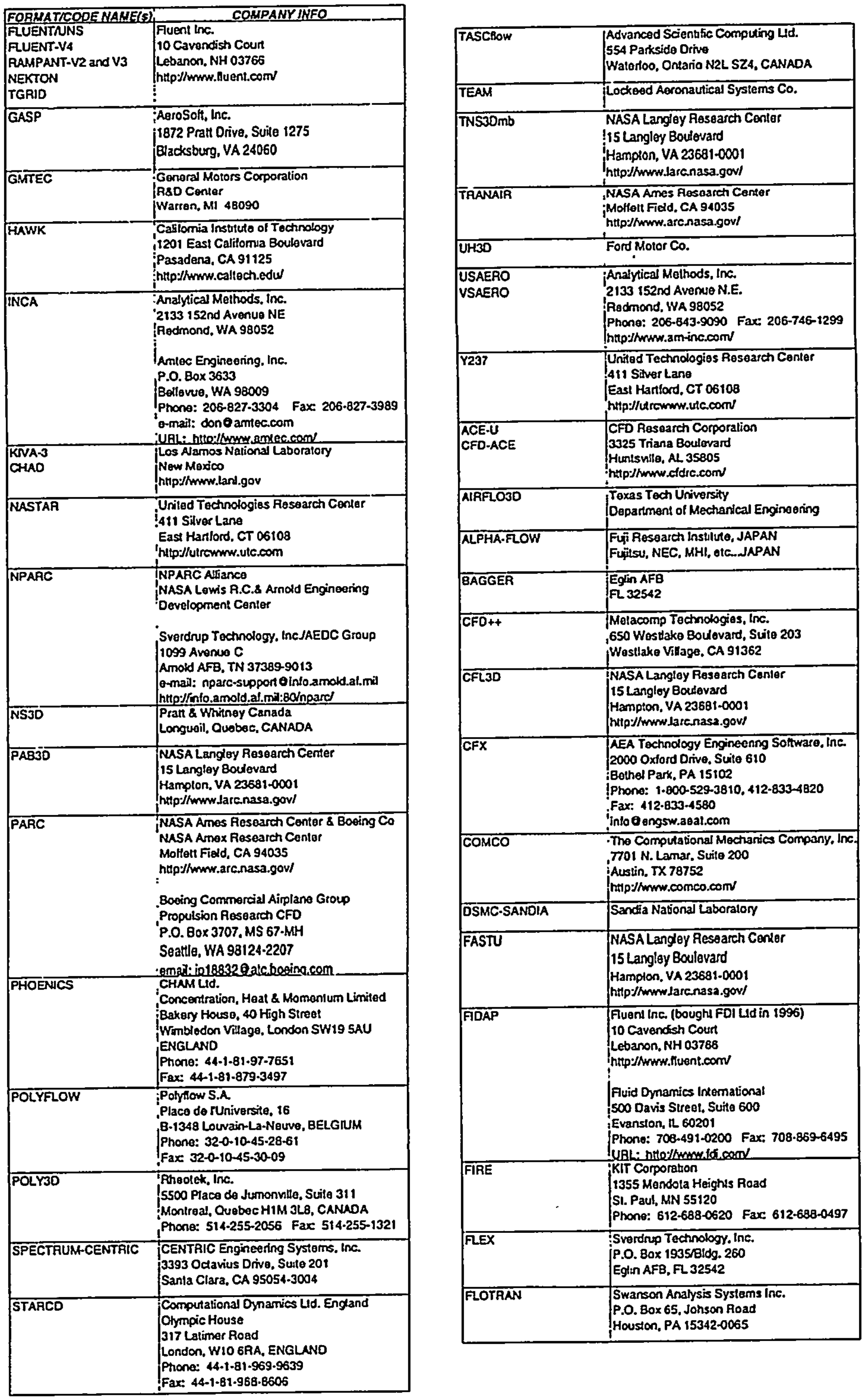
While much work remains to realize this ideal, CFD already is being used with considerable success in engineering design. Its utility and applicability will increase as the outstanding issues are resolved.

## V. SOURCES FOR FURTHER INFORMATION

Many references to specific topics have been cited throughout this chapter. For general information, one of several CFD texts can be consulted.[3,28,31,32]

At all stages of the CFD process (geometry acquisition, grid generation, flow solution, and postprocessing), a broad array of commercial, public-domain, and in-house proprietary codes are being used in engineering design. A small sampling of the software currently available to the design engineer has been mentioned herein. For more comprehensive and up-to-date listings, the reader can consult several sources. Computer hardware companies maintain lists of software that have been ported to their platforms; software vendors maintain lists of codes with which their own product is compatible. Table 3 has been extracted from one such list.[86] General[87] and industry-specific engineering periodicals often provide reviews of available software. And, a wealth of timely information can be found on the Internet.[88] Given the rapid pace at which CFD technology is evolving, this last source

Table 3. This provides a snapshot in time (late 1996) of the wide variety of commercially available, public domain, and proprietary CFD software used for engineering design and analysis.[86]



| FORMAT/CODE NAME(s) | COMPANY INFO |
|---|---|
| FLUENT/UNS<br>FLUENT-V4<br>RAMPANT-V2 and V3<br>NEKTON<br>TGRID | Fluent Inc.<br>10 Cavendish Court<br>Lebanon, NH 03766<br>http://www.fluent.com/ |
| GASP | AeroSoft, Inc.<br>1872 Pratt Drive, Suite 1275<br>Blacksburg, VA 24060 |
| GMTEC | General Motors Corporation<br>R&D Center<br>Warren, MI 48090 |
| HAWK | California Institute of Technology<br>1201 East California Boulevard<br>Pasadena, CA 91125<br>http://www.caltech.edu/ |
| INCA | Analytical Methods, Inc.<br>2133 152nd Avenue NE<br>Redmond, WA 98052<br><br>Amtec Engineering, Inc.<br>P.O. Box 3633<br>Bellevue, WA 98009<br>Phone: 206-827-3304 Fax: 206-827-3989<br>e-mail: don@amtec.com<br>URL: http://www.amtec.com/ |
| KIVA-3<br>CHAD | Los Alamos National Laboratory<br>New Mexico<br>http://www.lanl.gov |
| NASTAR | United Technologies Research Center<br>411 Silver Lane<br>East Hartford, CT 06108<br>http://utrcwww.utc.com |
| NPARC | NPARC Alliance<br>NASA Lewis R.C.& Arnold Engineering<br>Development Center<br><br>Sverdrup Technology, Inc./AEDC Group<br>1099 Avenue C<br>Arnold AFB, TN 37389-9013<br>e-mail: nparc-support@info.arnold.af.mil<br>http://info.arnold.af.mil:80/nparc/ |
| NS3D | Pratt & Whitney Canada<br>Longueil, Quebec, CANADA |
| PAB3D | NASA Langley Research Center<br>15 Langley Boulevard<br>Hampton, VA 23681-0001<br>http://www.larc.nasa.gov/ |
| PARC | NASA Ames Research Center & Boeing Co<br>NASA Amex Research Center<br>Moffett Field, CA 94035<br>http://www.arc.nasa.gov/<br><br>Boeing Commercial Airplane Group<br>Propulsion Research CFD<br>P.O. Box 3707, MS 67-MH<br>Seattle, WA 98124-2207<br>email: jp18832@atc.boeing.com |
| PHOENICS | CHAM Ltd.<br>Concentration, Heat & Momentum Limited<br>Bakery House, 40 High Street<br>Wimbledon Village, London SW19 5AU<br>ENGLAND<br>Phone: 44-1-81-97-7651<br>Fax: 44-1-81-879-3497 |
| POLYFLOW | Polyflow S.A.<br>Place de l'Universite, 16<br>B-1348 Louvain-La-Neuve, BELGIUM<br>Phone: 32-0-10-45-28-61<br>Fax: 32-0-10-45-30-09 |
| POLY3D | Rheotek, Inc.<br>5500 Place de Jumonville, Suite 311<br>Montreal, Quebec H1M 3L8, CANADA<br>Phone: 514-255-2056 Fax: 514-255-1321 |
| SPECTRUM-CENTRIC | CENTRIC Engineering Systems, Inc.<br>3393 Octavius Drive, Suite 201<br>Santa Clara, CA 95054-3004 |
| STARCD | Computational Dynamics Ltd. England<br>Olympic House<br>317 Latimer Road<br>London, W10 6RA, ENGLAND<br>Phone: 44-1-81-969-9639<br>Fax: 44-1-81-968-8606 |
| TASCflow | Advanced Scientific Computing Ltd.<br>554 Parkside Drive<br>Waterloo, Ontario N2L SZ4, CANADA |
| TEAM | Lockeed Aeronautical Systems Co. |
| TNS3Dmb | NASA Langley Research Center<br>15 Langley Boulevard<br>Hampton, VA 23681-0001<br>http://www.larc.nasa.gov/ |
| TRANAIR | NASA Ames Research Center<br>Moffett Field, CA 94035<br>http://www.arc.nasa.gov/ |
| UH3D | Ford Motor Co. |
| USAERO<br>VSAERO | Analytical Methods, Inc.<br>2133 152nd Avenue N.E.<br>Redmond, WA 98052<br>Phone: 206-643-9090 Fax: 206-746-1299<br>http://www.am-inc.com/ |
| Y237 | United Technologies Research Center<br>411 Silver Lane<br>East Hartford, CT 06108<br>http://utrcwww.utc.com/ |
| ACE-U<br>CFD-ACE | CFD Research Corporation<br>3325 Triana Boulevard<br>Huntsville, AL 35805<br>http://www.cfdrc.com/ |
| AIRFLO3D | Texas Tech University<br>Department of Mechanical Engineering |
| ALPHA-FLOW | Fuji Research Institute, JAPAN<br>Fujitsu, NEC, MHI, etc...JAPAN |
| BAGGER | Eglin AFB<br>FL 32542 |
| CFD++ | Metacomp Technologies, Inc.<br>650 Westlake Boulevard, Suite 203<br>Westlake Village, CA 91362 |
| CFL3D | NASA Langley Research Center<br>15 Langley Boulevard<br>Hampton, VA 23681-0001<br>http://www.larc.nasa.gov/ |
| CFX | AEA Technology Engineering Software, Inc.<br>2000 Oxford Drive, Suite 610<br>Bethel Park, PA 15102<br>Phone: 1-800-529-3810, 412-833-4820<br>Fax: 412-833-4580<br>info@engsw.aeat.com |
| COMCO | The Computational Mechanics Company, Inc.<br>7701 N. Lamar, Suite 200<br>Austin, TX 78752<br>http://www.comco.com/ |
| DSMC-SANDIA | Sandia National Laboratory |
| FASTU | NASA Langley Research Center<br>15 Langley Boulevard<br>Hampton, VA 23681-0001<br>http://www.larc.nasa.gov/ |
| FIDAP | Fluent Inc. (bought FDI Ltd in 1996)<br>10 Cavendish Court<br>Lebanon, NH 03766<br>http://www.fluent.com/<br><br>Fluid Dynamics International<br>500 Davis Street, Suite 600<br>Evanston, IL 60201<br>Phone: 708-491-0200 Fax: 708-869-6495<br>URL: http://www.fdi.com/ |
| FIRE | KIT Corporation<br>1355 Mendota Heights Road<br>St. Paul, MN 55120<br>Phone: 612-688-0620 Fax: 612-688-0497 |
| FLEX | Sverdrup Technology, Inc.<br>P.O. Box 1935/Bldg. 260<br>Eglin AFB, FL 32542 |
| FLOTRAN | Swanson Analysis Systems Inc.<br>P.O. Box 65, Johson Road<br>Houston, PA 15342-0065 |

is particularly valuable. In addition to lists and descriptions of the available software, user evaluations and direct comparisons of alternative codes and methodologies can be found there.

CFD is at a relatively early stage of development compared with other areas of CAE, such as linear FEM structural analysis. No single code covers all areas of application equally well. While "general purpose" CFD has been emphasized here, specialized application-specific numerical methods and software often are needed. Specialized experience and expertise can be found within university engineering departments, U.S. national laboratories, and engineering consulting firms; again, the Internet provides a good vehicle for exploring these possibilities.

## ACKNOWLEDGEMENTS

The authors thank Dr. Sherif El Tahry of the GM R&D Center for several discussions that helped to shape this report. We are grateful to Ms. Diane Poirier of ICEM CFD Engineering for compiling the list of CFD solver software provided in Table 3.[86] We also thank Mathew Maltrud and Richard Smith of Los Alamos National Laboratory for supplying Fig. 2, Norm Carter of GM's Powertrain Group for supplying Fig. 9a, and Christopher Rosbrook of MAGMA Foundry Technologies for providing Figs. 9b and 9c and the related discussion.

## REFERENCES

1. FLOWMASTER USA, Inc., co/Fluid Dynamics International, 500 Davis Street, Suite 600, Evanston, IL 60201.
2. *WAVE Basic Manual: Documentation/User's Manual, Version 3.4* (Ricardo Software, 7850 Grant Street, Burr Ridge, IL 60521, October 1996).
3. P. J. Roache, *Computational Fluid Dynamics* (Hermosa Publishers, Albuquerque, 1982).
4. N. Grün, "Simulating External Vehicle Aerodynamics with Carflow," SAE Paper No. 960679 (1996).
5. W. Aspray, *John von Neumann and the Origins of Modern Computing* (MIT Press, Cambridge, 1990).
6. "Accelerated Strategic Computing Initiative (ASCI): Draft Program Plan," prepared by Department of Energy Defense Programs (May 1996).
7. L. H. Turcotte, ICASE/LaRC Industry Roundtable, October 3–4, 1994; original source for some data is F. Baskett and J. L Hennessy, "Microprocessors: From Desktops to Supercomputers," *Science* **261**, 864–871 (13 August 1993).

8. J. K. Dukowicz and R. D. Smith, *Journal of Geophysical Research* **99**, 7991–8014 (1994).
9. R. D. Smith, J. K. Dukowicz, and R. C. Malone, *Physica D* **60**, 38–61 (1992).
10. G. Taubes, "Redefining the Supercomputer," *Science* **273**, 655–1657 (September 1996).
11. J. B. Heywood, *Internal Combustion Engine Fundamentals* (McGraw Hill, New York, 1988).
12. F. H. Harlow and A. A. Amsden, "Fluid Dynamics," Los Alamos Scientific Laboratory report LA-4700 (June 1971).
13. W. G. Vincenti and C. H. Kruger, *Introduction to Physical Gas Dynamics* (Robert E. Krieger Publishing Co, Huntington, NY, 1975).
14. P. A. Thompson, *Compressible-Fluid Dynamics* (McGraw Hill, New York, 1972).
15. H. Jeffreys, *Cartesian Tensors* (Cambridge University Press, Cambridge, UK, 1977).
16. F. A. Williams, *Combustion Theory*, 2nd ed. (Benjamin/Cummings, Menlo Park, CA, 1985).
17. T. G. Cowling, *Magnetohydrodynamics* (Interscience Tracts on Physics and Astronomy, No. 4, 1957).
18. S. Chandrasekhar, *Radiative Transfer* (Dover, New York, 1960).
19. D. R. Stull and H. Prophet, *JANAF Thermochemical Tables*, 2nd ed. (National Bureau of Standards, NSRDS-NBS 37, 1971).
20. B. McBride and S. Gordon, "Computer Program for Calculating and Fitting Thermodynamic Functions," NASA-RP-1271, National Aeronautics and Space Administration (1992).
21. R. B. Bird, W. E. Stewart, and E. N. Lightfoot, *Transport Phenomena* (Wiley, New York, 1960).
22. L. Crocco, "A Suggestion for the Numerical Solution of the Steady Navier-Stokes Equations," *AIAA Journal* **3**, (No. 10), 1824–1832 (1965).
23. L. D. Landau and E. M. Lifshitz, *Fluid Mechanics* (Pergamon Press, London, 1959).
24. O. Reynolds, "On the Dynamical Theory of Incompressible Viscous Fluids and the Determination of the Critèrion," *Philosophical Transactions of the Royal Society of London*, Series A **186**, 123 (1895).
25. H. Tennekes and J. L. Lumley, *A First Course in Turbulence* (MIT Press, Cambridge, MA, 1972).
26. B. E. Launder and D. B. Spalding, *Mathematical Models of Turbulence* (Academic Press, New York, 1972).
27. D. C. Wilcox, *Turbulence Modeling for CFD* (DCW Industries, La Canada, CA, 1993).

28. R. Peyret and T. D. Taylor, *Computational Methods for Fluid Flow* (Springer-Verlag, New York, 1983).

29. G. D. Smith, *Numerical Solution of Partial Differential Equations* (2nd ed., Oxford University Press, Oxford, 1978).

30. R. D. Richtmyer and K. W. Morton, *Difference Methods for Initial-Value Problems* (2nd ed., Interscience Publishers, New York, 1967).

31. C. A. J. Fletcher, *Computational Techniques for Fluid Dynamics, Volume I: Fundamental and General Techniques* (2nd Edition, Springer-Verlag, Berlin, 1991).

32. C. A. J. Fletcher, *Computational Techniques for Fluid Dynamics, Volume II: Specific Techniques for Different Flow Categories* (2nd Edition, Springer-Verlag, Berlin, 1991).

33. G. G. O'Brien, M. A. Hyman, and S. Kaplan, "A Study of the Numerical Solution of Partial Differential Equations," *Journal of Mathematics and Physics* **29**, 223–251 (1950).

34. M. J. Lee and W. C. Reynolds, "Numerical Experiments on the Structure of Homogeneous Turbulence," Report TF-24, Dept. of Mechanical Engineering, Stanford University (1985).

35. "Proceedings of the Workshop on Particle Methods in Fluid Dynamics and Plasma Physics," in *Computer Physics Communications*, J. U. Brackbill and J. J. Monaghan, Eds. (1988), Vol. 48 (No. 1).

36. F. H. Harlow, "The Particle-in-Cell Computing Method for Fluid Dynamics," in *Fundamental Methods in Hydrodynamics.* B. Alder, S. Fernbach, and M. Rotenberg, Eds. (Academic Press, New York, 1964).

37. J. U. Brackbill and H. M. Ruppel, "FLIP: A Method for Adaptively Zoned, Particle-in-Cell Calculations of Fluid Flows in Two Dimensions," *Journal of Computational Physics* **65**, 314 (1986).

38. J. J. Monaghan, "Particle Methods for Hydrodynamics," *Computer Physics Reports* **3**, 71–124 (1985).

39. J. K. Dukowicz, "A Particle-Fluid Numerical Model for Liquid Sprays," *Journal of Computational Physics* **35** (No. 2), 229–253 (1980).

40. P. J. O'Rourke, "Collective Drop Effects in Vaporizing Liquid Sprays," Ph.D. thesis, Princeton University (1981).

41. Y. Sahd and M. Schultz, "Conjugate Gradient-like Algorithms for Solving Non-symmetric Linear Systems," *Mathematics and Computation* **44**, 417–424 (1985).

42. W. L. Briggs, *A Multigrid Tutorial* (Society for Industrial and Applied Mathematics, Philadelphia, 1987).

43. W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vettering, *Numerical Recipes: The Art of Scientific Computing* (Cambridge University Press, Cambridge, 1987).

44. D. A. Knoll and P. R. McHugh, "Newton-Krylov Methods Applied to a System of Convection-Diffusion-Reaction Equations," *Computer Physics Communications* **88**, 141–160 (1995).

45. S. V. Patankar, *Numerical Heat Transfer and Fluid Flow* (McGraw-Hill, New York, 1980).

46. M. C. Cline, J. K. Dukowicz, and F. L. Addessio, "CAVEAT-GT: A General Topology Version of the CAVEAT Code," Los Alamos National Laboratory report LA-11812-MS (June 1990).

47. *HEXAR* (Cray Research Inc., Mendota Heights, MN, 1994).

48. G. D. Sjaardeam et al., "CUBIT Mesh Generation Environment, Vols. 1 & 2," Sandia National Laboratories report SAND94-1100/–1101(1994).

49. W. D. Henshaw, "A Fourth-Order Accurate Method for the Incompressible Navier-Stokes Equations on Overlapping Grids," *Journal of Computational Physics* **133**, 13–25 (1994).

50. R.B. Pember et al., "An Embedded Boundary Method for the Modeling of Unsteady Combustion in an Industrial Gas-Fired Furnace," Lawrence Livermore National Laboratory report UCRL-JC-122177 (October 1995).

51. J. P. Jessee et al., "An Adaptive Mesh Refinement Algorithm for the Discrete Ordinates Method," Lawrence Berkeley National Laboratory report LBNL-38800 (March 1996).

52. M. Landon and R. Johnson, Idaho National Engineering Laboratory, personal communication, 1995.

53. S. Ashley, "Rapid Concept Modelers," *Mechanical Engineering* **118**, No. 1, 64–66 (January 1996).

54. D. L. Reuss, R. J. Adrian, C. C. Landreth, D. T. French, and T. D. Fansler, "Instantaneous Planar Measurements of Velocity and Large-Scale Vorticity and Strain Rate Using Particle Image Velocimetry," SAE Paper No. 890616 (1989).

55. M. C. Drake, T. D. Fansler, and D. T. French, "Crevice Flow and Combustion Visualization in a Direct-Injection Spark-Ignition Engine Using Laser Imaging Techniques," SAE Paper No. 952454 (1995).

56. D. Deitz, "Next-Generation CAD Systems," *Mechanical Engineering* **118**, No. 8, 68–72 (August 1996).

57. G. Farin, *Curves and Surfaces for Computer-Aided Geometric Design* (Academic Press, San Diego, 1990).