by slowly changing A to B over the course of the simulation, performing a computer experiment that is impossible to conduct in the laboratory in order to derive experimentally measurable quantities.

Another simulation method, quantum dynamics, treats the atoms of particular interest with quantum-mechanical methods, and the rest of the system by using a force-field model. This technique will allow the simulation of such phenomena as proton exchange and enzymatic activity, impossible to do with force fields alone. *See* FREE ENERGY; SIMULATION.

Interactive computer graphics. Although not strictly a branch of theory, interactive graphics has become so prevalent in the chemical community that it is almost impossible to conceive of calculations without associated molecular manipulation and visualization. The chemist is essentially working with a sophisticated set of molecular models that are stored in the computer memory and displayed on a screen. Moving a mouse or turning a dial moves atoms about on the screen as easily as tangible models can be moved physically. Stereoscopic views are produced either by placing left- and right-eye images next to each other on the screen and using a viewer to merge the images, or by alternately blinking left- and right-eye images and looking through a viewer that presents the correct image to each eye.

Molecular orbitals, electron densities, and normal-mode vibrations also can be viewed from any direction. The sequence of conformations from computer simulations, displayed one frame after another, in effect creates a movie of the thermal motion of the molecules.

These techniques are most useful when studying interactions between large molecules, and they are often used in the design of potential drug molecules. The intuition of the scientist is used to position new molecules into an active site of an enzyme or to make alterations to the molecule that may improve its effectiveness as a drug. These structures are then used as starting geometries for further computational studies. *See* COMPUTER GRAPHICS; CONFORMATIONAL ANALYSIS; STEREOCHEMISTRY. Zelda R. Wasserman

Bibliography. M. P. Allen and D. J. Tildesley, *Computer Simulation of Liquids*, 1991; U. Burkert and N. Allinger, *Molecular Mechanics*, 1982; T. Clark, *A Handbook of Computational Chemistry*, 1985; D. M. Hirst, *A Computational Approach to Chemistry*, 1990; K. B. Lipkowitz and D. B. Boyd (eds.), *Reviews in Computational Chemistry*, vols. 1–6, 1990–1995.

Computational fluid dynamics

The numerical approximation to the solution of mathematical models of fluid flow and heat transfer. Computational fluid dynamics is one of the tools (in addition to experimental and theoretical methods) available to solve fluid-dynamic problems. With the advent of modern computers, computational fluid dynamics evolved from potential-flow and boundary-layer methods and is now used in many diverse fields, including engineering, physics, chemistry, meteorology, and geology.

Computational methods. The fundamental model of fluid flow, known as the Navier-Stokes equations, is derived from the conservation of mass, momentum, and energy. For example, if the fluid is incompressible, the Navier-Stokes equations can be written as Eqs. (1) and (2), where u, v, and w are the

$$\text{Continuity}: \quad \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} = 0 \tag{1}$$

Momentum :

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} + w\frac{\partial u}{\partial z} = -\frac{1}{\rho}\frac{\partial p}{\partial x} + \frac{1}{\mu}\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2}\right) \tag{2a}$$

$$\frac{\partial v}{\partial t} + u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y} + w\frac{\partial v}{\partial z} = -\frac{1}{\rho}\frac{\partial p}{\partial y} + \frac{1}{\mu}\left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} + \frac{\partial^2 v}{\partial z^2}\right) \tag{2b}$$

$$\frac{\partial w}{\partial t} + u\frac{\partial w}{\partial x} + v\frac{\partial w}{\partial y} + w\frac{\partial w}{\partial z} = -\frac{1}{\rho}\frac{\partial p}{\partial z} + \frac{1}{\mu}\left(\frac{\partial^2 w}{\partial x^2} + \frac{\partial^2 w}{\partial y^2} + \frac{\partial^2 w}{\partial z^2}\right) \tag{2c}$$

cartesian components of velocity, ρ the fluid density, p the pressure, and μ the fluid viscosity. The compressible form of the Navier-Stokes equations, which is of slightly different form because of the variable fluid properties, additionally contains the energy equation, which couples the temperature, velocity, and pressure fields. The complete formulation requires initial and boundary conditions and thermodynamic relations for fluid properties. Numerical solution of this set of nonlinear coupled partial differential equations presents many difficulties, including application of boundary conditions, particularly for free and moving surfaces; grid generation for complex geometries; turbulence modeling; pressure-velocity coupling for incompressible flows; and resolution of shock waves in supersonic flows. A hierarchy of approaches is available for simplifying Eqs. (1) and (2) [see **table**]. *See* DIFFERENTIAL EQUATION; EULER'S MOMENTUM THEOREM; NAVIER-STOKES EQUATIONS.

Computational fluid dynamics is associated with field-equation solutions of the Euler and Navier-Stokes equations instead of other computational techniques such as boundary-element methods for inviscid flows and momentum-integral methods for boundary-layer flows. The crucial elements of computational fluid dynamics are discretization, grid generation and coordinate transformation, solution of the coupled algebraic equations, turbulence modeling, and visualization.

Hierarchy of approaches to fluid-flow problems*

Mathematical basis	Type of partial differential equation	Physical problems
Laplace equation	Elliptic	Inviscid and irrotational flow
Euler equation	Hyperbolic	Inviscid flow
Boundary-layer equations	Parabolic	Thin viscous layer
Navier-Stokes equations	Mixed	Laminar flow
Reynolds-averaged Navier-Stokes (RaNS) equations + turbulence model	Mixed	Turbulent flow
Large-eddy simulation (Navier-Stokes equations + subgrid model)	Mixed	Turbulent flow
Direct numerical solution (Navier-Stokes equations + grid resolving all length scales)	Mixed	Turbulent flow

*Listed in order of increasing complexity of numerical solution.

Discretization. Numerical solution of partial differential equations requires representing the continuous nature of the equations in a discrete form. Discretization of the equations consists of a process where the domain is subdivided into cells or elements (that is, grid generation) and the equations are expressed in discrete form at each point in the grid by using finite difference, finite volume, or finite element methods. The finite difference method requires a structured grid arrangement (that is, an organized set of points formed by the intersections of the lines of a boundary-conforming curvilinear coordinate system), while the finite element and finite volume methods are more flexible and can be formulated to use both structured and unstructured grids (that is, a collection of triangular elements or a random distribution of points). Complex shapes necessitate nonuniform, boundary-conforming grids with grid points concentrated in regions of high gradients, such as inside boundary layers or near shock waves. Structured grids are usually generated by using algebraic interpolation functions or elliptic partial differential equations. For finite difference methods, the equations are rewritten in generalized nonorthogonal coordinates and evaluated on a so-called computational domain, which is defined by the transformation between the boundary-conforming curvilinear-coordinate system in the so-called physical domain and an orthogonal grid with uniform spacing (**Fig. 1**). *See* FINITE ELEMENT METHOD.

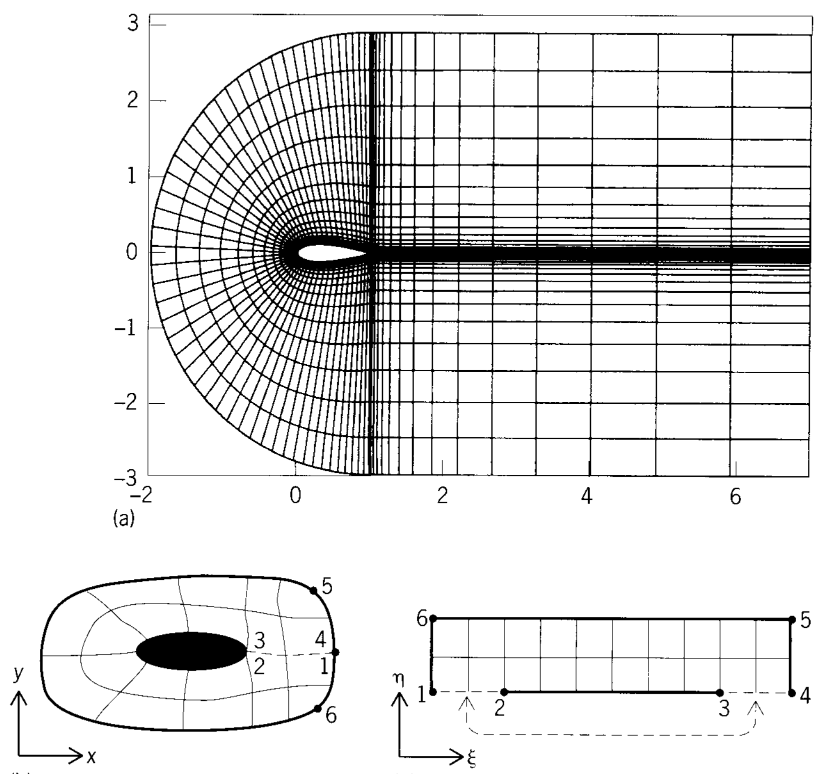


Fig. 1. Computational grid for numerical analysis of flow about a hydrofoil. (*a*) Structured grid. (*b*) Schematic of physical domain. (*c*) Schematic of computational domain.

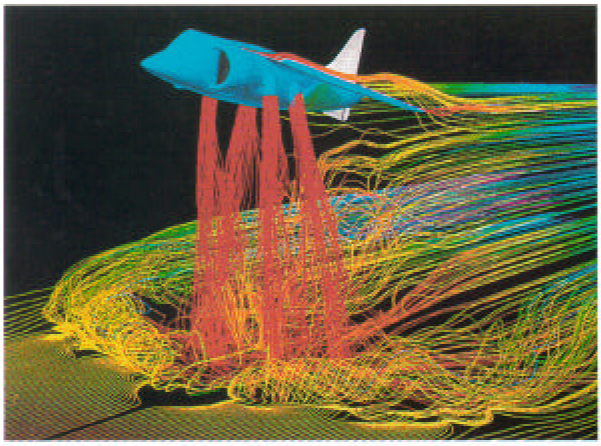
Finite difference discretization of Eqs. (1) and (2) is based on the algebraic representation of derivatives by using Taylor-series expansions. For example, the temporal derivative in Eq. (2*a*) expressed as a first-order backward difference is Eq. (3), where (i, j, k) is the location of the grid

$$\frac{\partial u}{\partial t} = \frac{u^{n}_{i,j,k} - u^{n-1}_{i,j,k}}{\Delta t} + \vartheta[(\Delta t)] \qquad (3)$$

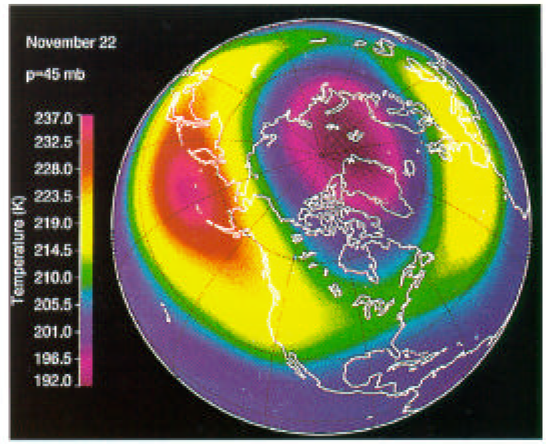
point, (n) is the current time level, $(n-1)$ is the previous time level, Δt is the time step between (n) and $(n-1)$, and $\vartheta(t)$ represents the truncated terms in the Taylor-series approximation of the partial derivative $\partial u/\partial t$. (Dropping these terms introduces a source of error, known as a truncation error, which is of an order of magnitude equal to that of the time step Δt.) Due to advantageous stability characteristics, in comparison to central differencing, an upstream-backward difference is typically used for the convective terms. Assuming that the grid is uniform and orthogonal, the first-order upstream difference for the x-component convective term of Eq. (2*a*) is given by Eq. (4),

$$u\frac{\partial u}{\partial x} = u^{n-1}_{i,j,k}\frac{u^{n}_{i,j,k} - u^{n}_{i-1,j,k}}{\Delta x} + \vartheta[(\Delta x)] \qquad (4)$$

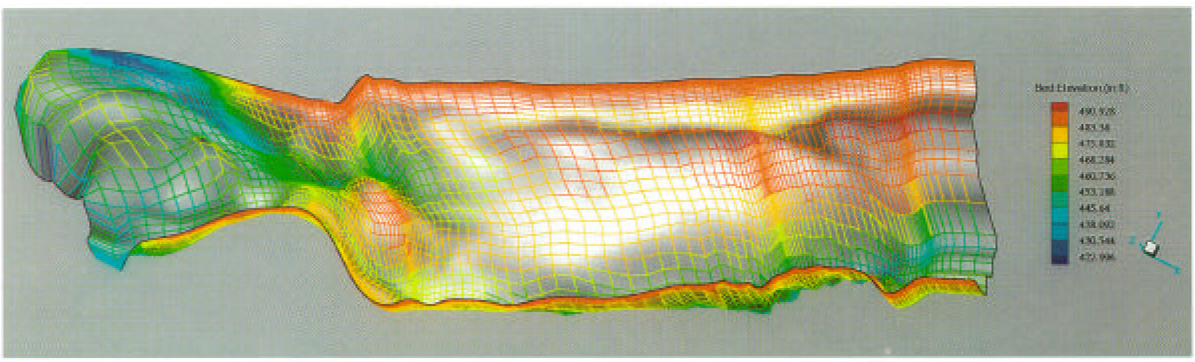
where Δx or the grid spacing, and the grid point locations (i, j, k) are given by a numerical molecule (**Fig. 2**). Usually, the viscous terms in the momentum equations are evaluated by using a central difference. For example, the second-order central difference of the second partial derivative



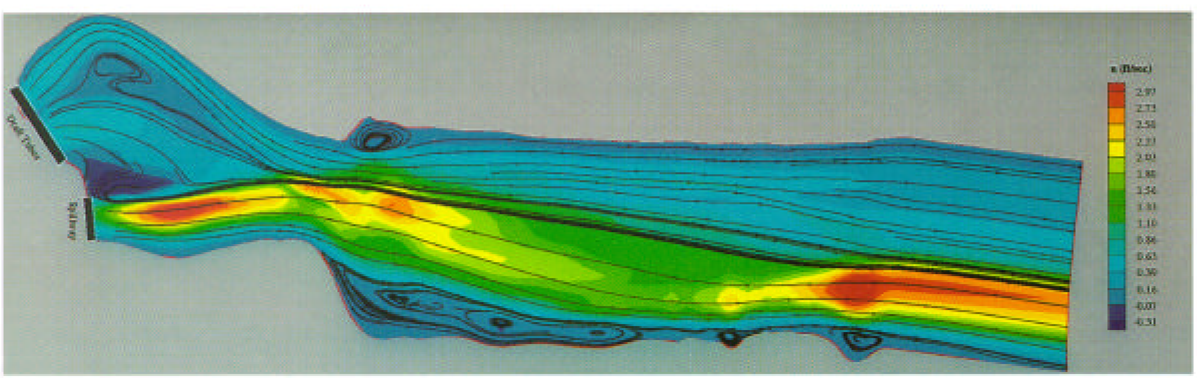
(a)



(b)



(c)



(d)

Computer-graphic visualizations of the results of computational fluid-dynamic simulations. *(a)* Flow about a low-flying aircraft, shown by particle traces colored according to time since release (*M. Smith, K. Chawla, and W. Van Dalsem, NASA Ames Research Center*). (*b*) Global atmospheric temperature prediction (*W. Grose et al., NASA Langley Research Center*). (*c*) River flow downstream of a dam, showing bed topography and surface grid, and (*d*) distribution of streamwise velocity (*S.K. Sinha, F. Sotiropoulos, and A. Odgaard, Iowa Institute of Hydrodynamic Research*).

of u in the x direction [from Eq. (2a)] is given by Eq. (5).

$$\frac{\partial^2 u}{\partial x^2} = \frac{u^n_{i+1,j,k} - 2u^n_{i,j,k} + u^n_{i-1,j,k}}{(\Delta x)^2} + \vartheta[(\Delta x)^2] \quad (5)$$

Solution of equations. The assembly of the finite difference equations for each grid point results in a large implicit system of algebraic equations for each of the velocity components (u, v, w). For example, the equation for the u component of velocity at the point (i, j, k) is Eq. (6), where A

$$\begin{aligned} &A_{i,j,k}u^n_{i,j,k} + A_{i+1,j,k}u^n_{i+1,j,k} + A_{i-1,j,k}u^n_{i-1,j,k} \\ &\quad + A_{i,j+1,k}u^n_{i,j+1,k} + A_{i,j-1,k}u^n_{i,j-1,k} \\ &\qquad + A_{i,j,k+1}u^n_{i,j,k+1} + A_{i,j,k-1}u^n_{i,j,k-1} = C^{n-1} \quad (6) \end{aligned}$$

is a matrix of coefficients related to the grid and the transformation relations (between the physical and computational domains) and C is a known column vector including pressure, body-force, and velocity terms from the previous time step. Equation (6) is usually solved by iterative schemes, such as the alternating direction implicit (ADI) method, which is based upon splitting Eq. (6) into three tridiagonal systems that correspond to sweeps along each of the index (i, j, k) directions. If Eq. (4) and Eq. (5) are evaluated at time level $(n - 1)$ instead of (n), the system of finite difference equations is explicit and the solution at each point can be directly evaluated. However, explicit formulations have restrictions on the ratio of the time step to the grid spacing, which are significant for viscous flows because of the fine near-wall grid spacing required to resolve the boundary layer. *See* MATRIX THEORY.

For either implicit or explicit approaches, efficient use of computers is required. Current-generation supercomputers are based on vector processors, which are most efficient with large arrays (vectors) of numbers. Optimizing a program, or vectorization, is accomplished by following programming rules so that the compiler can translate the code into vector instructions. *See* SUPERCOMPUTER.

Complete solution of the Navier-Stokes equations requires, for compressible flow, an equation of state such as the ideal gas equation, which relates temperature and density. In contrast, the solution of the incompressible equations presents the problem that the system of equations is lacking an equation for the direct solution of the pressure field. Typically, an equation for pressure is derived which satisfies continuity, and an algorithm must be incorporated which iteratively couples the pressure and velocity fields.

Turbulence modeling. There are a variety of approaches for resolving the phenomena of fluid turbulence. The Reynolds-averaged Navier-Stokes (RANS) equations, which are derived by decomposing the velocity into mean and fluctuating components, contain additional turbulent Reynolds-stress terms which are unknown. Closure of the RANS equations can be accomplished by using the Boussinesq assumption, which relates the Reynolds

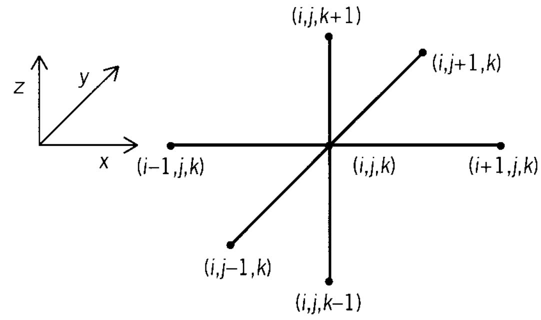


Fig. 2. Numerical molecule, showing nomenclature for neighboring grid points in finite difference discretization.

stresses to the mean rate of strain or the Reynolds-stress equations, which can be solved directly with additional modeling assumptions. An alternative to the RANS equations is large-eddy simulation, which solves the Navier-Stokes equations in conjunction with a subgrid turbulence model. The most direct approach to solving turbulent flows is direct numerical simulation, which solves the Navier-Stokes equations on a mesh that is fine enough to resolve all length scales in the turbulent flow. Unfortunately, direct numerical simulation is limited to simple geometries and low-Reynolds-number flows because of the limited capacity of even the most sophisticated supercomputers. *See* TURBULENT FLOW.

Visualization. The final step is to visualize the results of the simulation. Powerful graphics workstations and visualization software permit generation of velocity vectors, pressure and velocity contours, streamline generation, calculation of secondary quantities (such as vorticity), and animation of unsteady calculations. Despite the sophisticated hardware, visualization of three-dimensional and unsteady flows is still particularly difficult. Moreover, many advanced visualization techniques tend to be qualitative, and the most valuable visualization often consists of simple x-y plots comparing the numerical solution to theory or experimental data. *See* COMPUTER GRAPHICS.

Applications. Computational fluid dynamics has wide applicability in such areas as aerodynamics, hydraulics, and geophysical flows, with length and time scales of the physical processes ranging from millimeters (fractions of an inch) and seconds to kilometers (miles) and years.

Vehicle aerodynamics and hydrodyamics, which have provided much of the impetus in the development of computational fluid dynamics, are primarily concerned with the flow around aircraft, automobiles, and ships where the goal is the determination of the lift and drag forces and the moments resulting from these forces. Furthermore, computational fluid dynamics is being applied to explicate basic physics and to examine complete vehicle configurations. *See* AERODYNAMIC FORCE; AERODYNAMICS; HYDRODYNAMICS.

Hydraulics and environmental fluid dynamics also find many applications of computational fluid dynamics. The prediction of pollutant dispersion in rivers, lakes, and estuaries and riverine morphology, including river-bed formation and sediment deposition, have been studied by using computational fluid dynamics. Detailed analysis of hydraulic structures by using computational fluid dynamics can provide data which are useful in new designs and in the modernization of aging facilities. *See* HYDRAULICS; RIVER; WATER POLLUTION.

The study of atmospheric and oceanic dynamics finds prolific use of computational fluid dynamics. Although long-term prediction of weather is not possible, because of its random nature and the multiplicity of length and time scales, the study of smaller problems (such as thunderstorms and ocean circulation) and low-resolution global-scale phenomena (for example, global climate models) is possible. *See* CLIMATE MODELING; DYNAMIC METEOROLOGY; FLUID FLOW; FLUID-FLOW PRINCIPLES; NUMERICAL ANALYSIS; OCEAN CIRCULATION; SIMULATION; WEATHER FORECASTING AND PREDICTION. **Eric Paterson; Fred Stern**

Bibliography. M. Abbott and D. R. Basco, *Computational Fluid Dynamics: An Introduction for Engineers*, 1990; K. A. Hoffmann and S. T. Chiang, *Computational Fluid Dynamics for Engineers*, 2 vols., 2d ed., 1993; Y. Jaluria, *Computer Methods for Engineering*, 1988; J. Von Wendt (ed.), *Computational Fluid Dynamics: An Introduction*, 1992.

Computer

A device that receives, processes, and presents information. The two basic types of computers are analog and digital. Although generally not regarded as such, the most prevalent computer is the simple mechanical analog computer, in which gears, levers, ratchets, and pawls perform mathematical operations—for example, the speedometer and the watt-hour meter (used to measure accumulated electrical usage). The general public has become much more aware of the digital computer with the rapid proliferation of the hand-held calculator and a large variety of intelligent devices, ranging from typewriters to washing machines.

Analog computer. An analog computer uses inputs that are proportional to the instantaneous value of variable quantities, combines these inputs in a predetermined way, and produces outputs that are a continuously varying function of the inputs and the processing. These outputs are then displayed or connected to another device to cause action, as in the case of a speed governor or other control device.

The electronic analog computer is often used for the solution of complex dynamic problems. Electrical circuits, usually transistorized, perform the processing. Electronic amplifiers allow signals to be impressed upon cascaded circuits without significant electrical loss of attenuation through loading of prior stages, a feature absent in purely mechanical computers. Friction in a mechanical analog computer builds up and limits the complexity of the device.

Small electronic analog computers are frequently used as components in control systems. Inputs come from measuring devices which output an electrical signal (transducers). These electrical signals are presented to the analog computer, which processes them and provides a series of electronic outputs that are then displayed on a meter for observation by a human operator or connected to an electrical action device to ring a bell, flash a light, or adjust a remotely controlled valve to change the flow in a pipeline system. If the analog computer is built solely for one purpose, it is termed a special-purpose electronic analog computer. *See* CONTROL SYSTEMS.

General-purpose electronic analog computers are used by scientists and engineers for analyzing dynamic problems. A general-purpose analog computer receives its degree of flexibility through the use of removable control panels, each of which carries a series of mating plugs. Outputs from one component are routed to the input of another component by connecting an electrical conductor from one mating plug on the removable board (output) to another plug on the removable board (input). This process is called patching, and the removable panel is frequently called a patch board.

Thus, in any analog computer the key concepts involve special versus general-purpose computer designs, and the technology utilized to construct the computer itself, mechanical or electronic. In any case, an analog computer receives inputs that are instantaneous representations of variable quantities and produces output results dynamically to a graphical display device, a visual display device, or in the case of a control system, a device which causes mechanical motion. *See* ANALOG COMPUTER.

Digital computer. In contrast, a digital computer uses symbolic representations of its variables. The arithmetic unit is constructed to follow the rules of one (or more) number systems. Further, the digital computer uses individual discrete states to represent the digits of the number system chosen.

Electronic versus mechanical computers. The most prevalent special-purpose mechanical digital computers have been the supermarket cash register, the office adding machine, and the desk calculator. Each of these is being widely replaced by electronic devices allowing much greater logical decision making and greatly increased speed. For example, most products now carry a bar code, the Universal Product Code (UPC); in suitably equipped supermarkets, the code is scanned by a light-sensitive device, bringing information about each product into the point-of-sale (POS) terminal that has replaced the mechanical cash register. The POS terminal then computes total charges and provides a receipt for the customer. It may also communicate with a centralized computer system that controls inventory, accounts payable, salaries and commissions, and so on. While a mechanical cash register