

Labels can be used in any visualization, both as a tool for showing features of the visual representations and as a means to help clarify potentially confusing or unclear items. By adding labels, the viewer can be shown the size of the domain, the range of the values, the coarseness of the simulation, etc. Labels make the visualization more clear, understandable, and, therefore, a more useful means of communication.

27.3.2.4 Human Perception

One filter that will always have an effect on how data are viewed is the human perceptual system [Weintraub and Walker 1969]. Our perception of the world does not exactly match with physical reality. In fact, there are many elements of the real world that we cannot directly perceive at all. For example, human vision covers only a small range of the electromagnetic spectrum; human hearing perceives only air pressure changes within a specific range of frequencies; human olfactory nerves have limited precision and no real ability to determine directionality of smells. Each sense has its own perceptual anomalies.

It is likely that some animals are able to perceive phenomena that humans cannot consciously perceive. For example, many species of birds can sense the earth's magnetic field and use it as a navigational aid (magnetoperception). Fortunately, we are able to use instruments that can sense elements of the physical world that we are unable to sense. These instruments then translate the sensory input into a display that humans are able to interpret. Visualization often involves the mapping of information from imperceptible forms to something we can interpret and analyze.

The field of *human factors* studied the relationship between people and machines and has a deep body of research to draw upon. It will be beneficial for anyone creating visualizations to familiarize themselves with the work of this field [Wickens et al. 1994].

All human perception, however, suffers from the additional problem that each of our brains interprets the incoming signals differently. Our experiences through life have trained our perceptual systems uniquely. Many biases are fairly consistent throughout a culture because the individuals within that culture will have had many similar experiences. Symbols in a culture lead to an understanding of what lies ahead or within; a skull and crossbones has a specific meaning in western culture, as does a railroad crossing sign. Colors are also culturally biased. Red can mean danger or stop; yellow, caution; and green, go. But for some, green also implies money or envy. In science, colors often are used to represent a scale of information, but this might not be understood by the rest of the culture. Which colors should represent high and low values, and which should signify the interesting data?

It is important to take human perceptual biases into account when designing an information display, but one also must recognize that these biases are not necessarily universal. Thus, a scale or legend should always be used to illustrate how information is being mapped.

27.4 The Practice of Scientific Visualization

In this section, we examine the process and visual components of visualization, we discuss several types of tools available for producing visualization, and we look at examples of how visualization has been applied to particular sciences and evolved over the course of several years.

27.4.1 Representation Techniques

One of the most important elements in creating scientific visualizations is the choice of visual representation, or **visual idioms**. This section surveys a variety of commonly used visual idioms, each of which is appropriate in different situations. The visual representation is created by combining the elements of form, color, and motion that together show features of the data. This representation can range from very realistic renderings of real physical objects to abstract glyphs used to combine many pieces of information into a single idiom.

The traditional forms of data display should not be dismissed. Quantitative data can readily be retrieved from such representations as the XY plot, the contour map, and the bar chart (Figure 27.5).

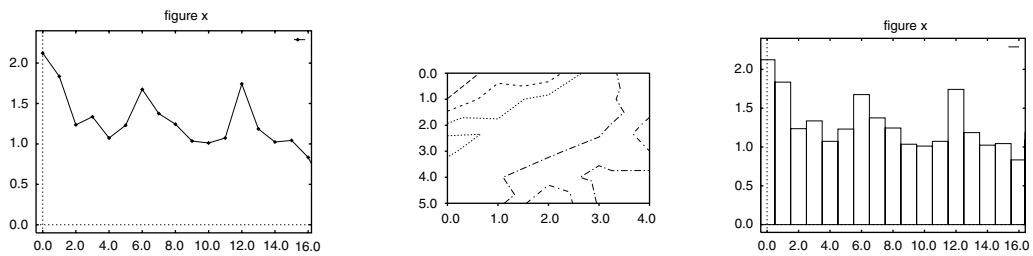


FIGURE 27.5 Simple 2-D graph representations convey both quantitative and qualitative information.



FIGURE 27.6 A realistic graphic representation of a front-end loader rendered from CAD data. (Courtesy of Caterpillar; Mark Bajuk; NCSA.)

27.4.1.1 Realism Continuum

Accurate representation of information does not necessitate that the display be realistic. Information can be represented in a very abstract form, and its contents can be read accurately by someone that knows what the form symbolizes. Sometimes it is useful, though not always possible, to show a realistic representation of what the data symbolize, as in many architectural visualizations and prototype designs (Figure 27.6).

In contrast, it is often more useful to create an abstract representation. This is especially true when there is not a direct physical counterpart to the concept being displayed. This is a convenient way of representing many variables simultaneously. The drawback with using very abstract representational schemes is that it takes time for the viewer to learn and become fluent with the symbols. In [Figure 27.7](#), symbols (or glyphs) represent different aspects of current weather conditions.

27.4.1.2 Color

Color adds to the appeal of a visualization. More importantly, color is used to convey information about data. Some fields, including atmospheric sciences, chemistry, and seismic analysis, have developed common conventions about color use within their application areas. Colors must be chosen carefully, with a view to the goal of the visual analysis. For example, visual displays often are used to identify the quantitative value at a point. Color can be used for this task by assigning colors to specific data values. The number of colors should be limited to about seven, and the colors should be easily distinguished from each other. Alternatively, the visualization task might be to determine the overall structure of the data slice, so that a color map that is continuously varying can be more effective.

[Figure 27.8](#) shows the use of a variety of color palettes to color the same 2-D slice of data. The top two color maps use a wide range of hues — these are variations of a “rainbow” palette. These color maps

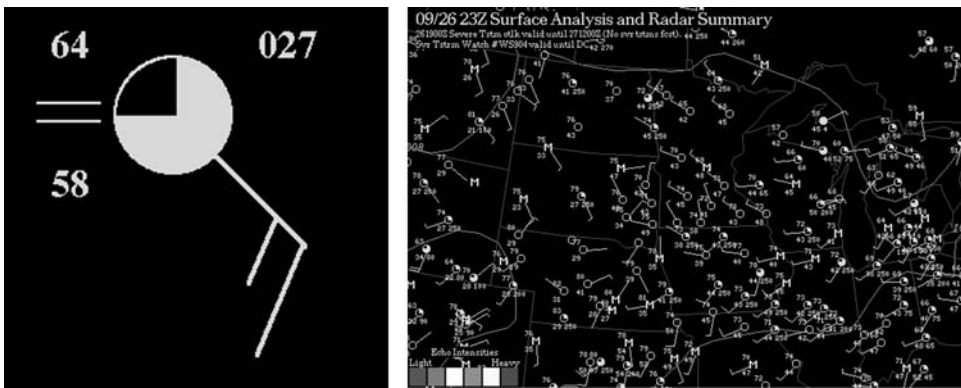


FIGURE 27.7 Weather maps often include an abstract symbol which depicts wind speed and direction, cloud coverage, etc.

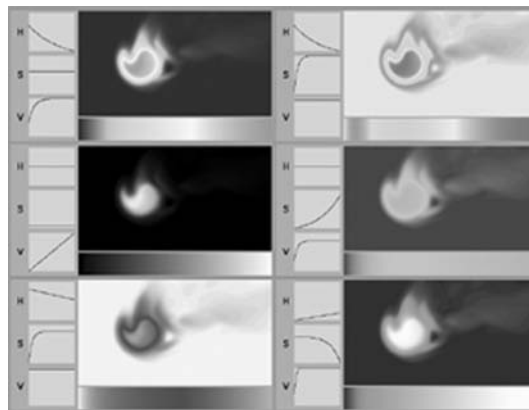


FIGURE 27.8 (See Plate 27.8 in the color insert following page 29-22.) A variety of color palettes used to color the the same two-dimensional slice of data. The top two color maps use a wide range of hues and introduce discontinuities into the image that may not be present in the data. If we are looking for overall structure, a smoothly changing palette with a restricted hue - such as the “fire” palette shown in the lower right - is more appropriate. (Courtesy of NCSA, Yale University.)

introduce discontinuities into the image that may not be present in the data. For example, the edge from yellow to green or yellow to red suggests a distinct change in the data, which may or may not be there. The distinct bands may be useful if we’re trying to isolate all the data values within a particular range. However, if we’re looking for overall structure, a smoothly changing palette with a restricted hue, such as the “fire” palette shown in the lower right, is more appropriate.

27.4.1.3 Form

27.4.1.3.1 Two-Dimensional Plane

The simplest form to present on a sheet of paper or a computer screen is a 2-D plane. Depending on the dimensionality of the data, there can be tradeoffs between representing data purely in two dimensions and giving a 3-D view of the data. In a 2-D representation, we do not have to deal with perspective projection or other potentially confusing 3-D cues, thus making it much easier to make quantitative determinations from the rendering. Of course, this works best for data which inherently lie on a 2-D grid. Higher-dimensional datasets can be examined a slice at a time by cutting 2-D planes through it.

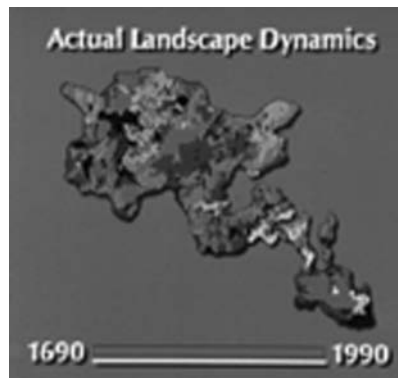


FIGURE 27.9 (See Plate 27.9 in the color insert following page 29-22.) Discrete color mapping is used to depict the age of the forest in Yellowstone National Park [Kovacic et al. 1990]. (Courtesy of D. Kovacic, A. Craig, and R. Patterson; UIUC, NCSA.)

One can represent data on a 2-D raster image (also known as a false-color map). An example of this is the visualization of a 2-D fluid flow problem involving a jet of high-density material into a lower-density material.

The range of values is mapped to a range of colors. The choice of colors requires considerable thought in order to produce a meaningful image. Choosing colors arbitrarily may lead to false impressions of what the data indicate. It is not always necessary to employ a continuous range of color values. Sometimes a few colors chosen to indicate discrete states can convey information more effectively. In Figure 27.9, discrete shades of yellow and green represent the age of the forest since the last fire in the region, red indicates a current fire, and black represents areas of nonvegetation.

There are several other idioms for representing information on a 2-D plane, many of which extend into 3-D forms. Contour lines and vector plots are two examples. Contour lines can be drawn on a 2-D plane of data indicating where specific values of the data are located. Typically, multiple contours are drawn in a single visualization to show the range of values. Vector values on a 2-D plane of data often are represented as arrows that point in the direction of the vector, with a length proportional to the magnitude. To view the entire domain, arrows are placed at the location of each data element.

27.4.1.3.2 Height

If a goal is to see correlations between two variables in the dataset, one can add the element of height to the aforementioned color mapping idiom. One variable is mapped to the color, and the other is used to determine the height of the surface (e.g., Figure 27.10). One must be careful in viewing such representations so as to avoid inaccuracies due to poor viewpoint selection. For example, in the extreme case, viewing a height-color rendering from directly above causes the height information to be lost.

27.4.1.3.3 Volume Rendering

The technique of color-mapping a single variable of a 2-D dataset can be extended to a 3-D dataset using a computer graphics technique called volume rendering [Drebin et al. 1988]. In looking at 3-D data, a viewpoint must be selected, and then an image is rendered by traversing through the elements of the dataset and assigning color and transparency values. A common usage is to volume-render medical data, such as the dog heart in Figure 27.11. In this example, the less dense material is assigned a high transparency value, so we can see through to the higher density material of the muscle and bone [Moran and Potter 1992].

27.4.1.3.4 Vectors: Arrows, Tracers, Streamlines, etc.

There are several common idioms used for displaying vector data. A standard representation of a 2-D vector field is to simply draw an arrow from each cell in the data in the direction of the vector, with the length proportional to the magnitude (Figure 27.12).

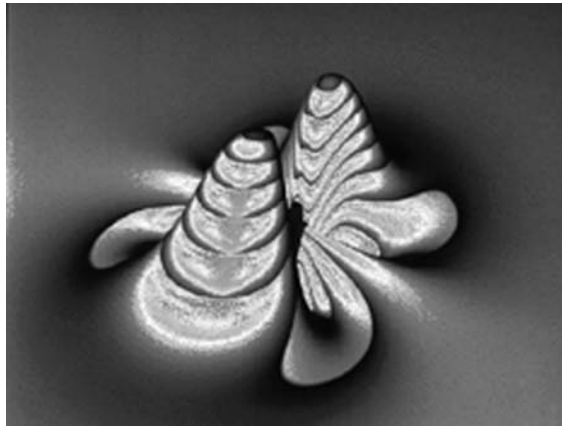


FIGURE 27.10 Visualization of gravitational effects of colliding black holes. (Courtesy of Mark Bajuk, Edward Seidel; NCSA [Anninos et al. 1993].)

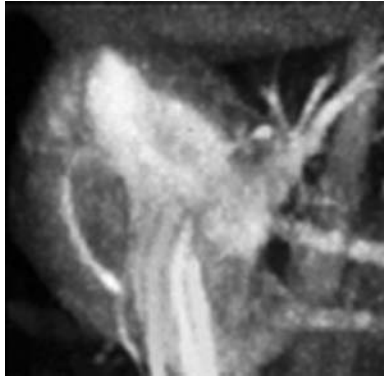


FIGURE 27.11 (See Plate 27.11 in the color insert following page 29-22.) Volume rendering of a dog's heart. (Courtesy of E. Hoffman, P. Moran, C. Potter; NCSA.)

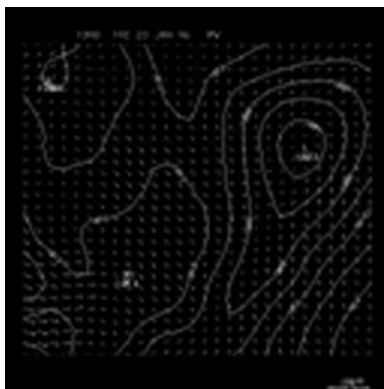


FIGURE 27.12 Wind vector field and pressure contours.

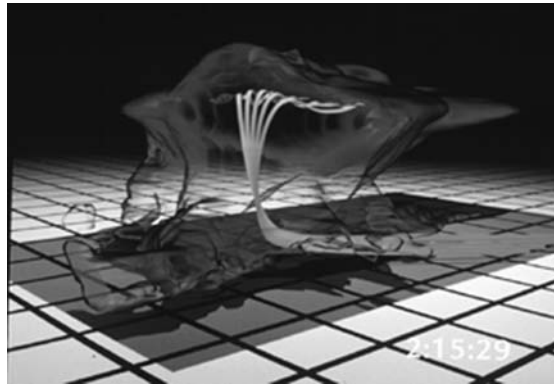


FIGURE 27.13 (See Plate 27.13 in the color insert following page 29-22.) Tracers clarify the wind flow within a simulation of a severe thunderstorm. (Courtesy of Robert Wilhelmson et al. NCSA.)

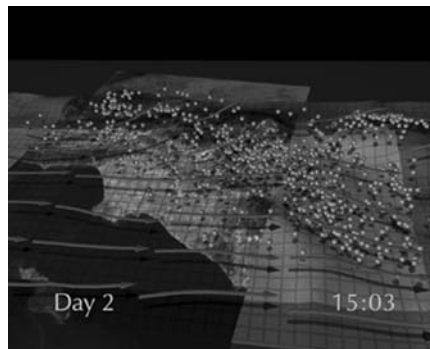


FIGURE 27.14 (See Plate 27.14 in the color insert following page 29-22.) Streaklines and particles depict smog in Los Angeles. Each green particle represents 10 tons of reactive organic gases; each yellow particle, carbon monoxide; and each red particle, nitrogen oxide. (Courtesy of W. Sherman, M. McNeill et al. NCSA.)

Just as techniques such as colored smoke are used to visualize fluid flows in the laboratory, computer graphics equivalents of smoke may be used to display flowfields in computational simulations [Merzkirch 1987]. There are several idioms that give an effective qualitative view of the data. The simplest utilizes particles to indicate the flow. By adding lines showing the history of the paths taken by the particles, we derive a different technique often called **tracers** (Figure 27.13). These are also referred to as **streamlines**, especially when the particle itself is not represented. A similar idiom, the **streakline**, shows a path through the vector field for a static moment of time (Figure 27.14). In other words, it shows a portion of the path a particle would take if it were traveling infinitely fast.

As with many geometric representations, additional information (age, velocity, temperature, etc.) can be mapped onto a tracer, streamline, or streakline. This information can be represented with color, transparency, texture, or twist. Figure 27.15 shows a good example of representing the streamwise vorticity of a fluid flow simulation. In this representation, the twist along a streamline is proportional to the amount of vorticity at the particular location of the domain.

27.4.1.3.5 Isosurfaces

Extending the isoline (or contour line) to three dimensions gives us the isosurface [Lorensen and Cline 1987]. An **isosurface** is a surface of constant value in three dimensions. All points inside the surface have values below the threshold level, and all outside have values above the threshold level, or vice versa. For

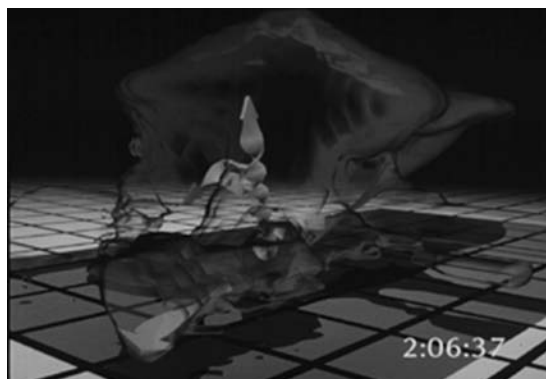


FIGURE 27.15 (See Plate 27.15 in the color insert following page 29-22.) A ribbon's rate of twist indicates the streamwise vorticity. (Courtesy of Robert Wilhelmson et al. NCSA.)

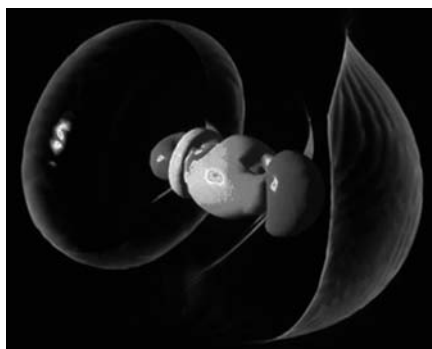


FIGURE 27.16 Isosurfaces depict regions of electron-density change. (Courtesy of Jeffrey Thingvold; NCSA.)

example, the isosurfaces shown in the cloud surface shown in Figure 27.16 represent changes in electron density.

27.4.1.3.6 Forms from Traditional Science

It often is desirable to choose representational idioms with which the audience is already familiar. For example, chemists are familiar with representing molecular structure with ball-and-stick models. These can be replicated in computer graphics. Using computer-generated imagery, however, removes certain physical constraints, and additional information may be given. For example, Figure 27.17 shows a picture of a leukotriene molecule in which colored, orthographic shadow views are projected onto a three-walled stage to give the viewer a better clue to the 3-D shape.

27.4.1.3.7 Cutting Plane

Sometimes it is useful when looking at a 3-D dataset to cut the 3-D cube with what is called a cutting plane, revealing data along a single 2-D slice of that cube. Figure 27.18 shows a cutting plane, together with the 3-D isosurface. The black line in the isosurface indicates where the plane intersects it. The scale alongside the cutting place shows the mapping between the level of water content and the colors.

27.4.1.3.8 Alpha Shapes

A technique called **alpha shapes** allows one to represent the concept of shape applied to a collection of points in space [Edelsbrunner and Mücke 1994]. Alpha is the name of the parameter that, when varied,

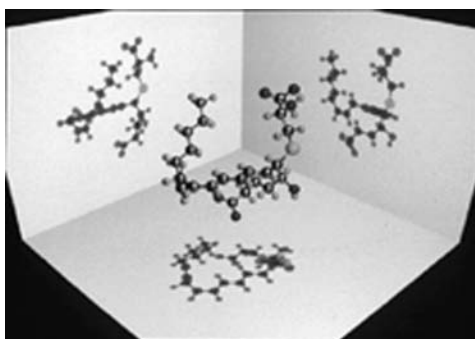


FIGURE 27.17 (See Plate 27.17 in the color insert following page 29-22.) A computer-generated ball-and-stick model of a leukotriene molecule. Colored shadows aid in perceiving the 3-D shape. (Courtesy of D. Herron, Eli Lilly & Co.; J. Thingvold, W. Sherman, NCSA.)

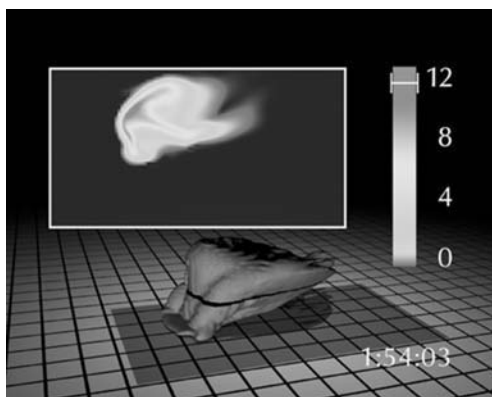


FIGURE 27.18 (See Plate 27.18 in the color insert following page 29-22.) A cutting plane displays additional detail of a particular slice of a 3-D dataset. (Courtesy of Robert Wilhelmson et al. NCSA.)

affects the complexity of the resultant shape. Large values of alpha produce a simple representation of the shape — a convex hull. As the value of alpha is continuously decreased, more complex shapes are created, revealing cavities, tunnels, and voids inherent in the dataset. For example, [Figure 27.19](#) shows a geometric representation of a molecule of gramicidin A, the first clinically used antibiotic (predating penicillin).

27.4.1.4 Motion

Motion can be a very important cue to the viewer for understanding the data being portrayed. It provides a qualitative overview of dynamic data by showing how the system evolves over time. Of course, the medium in which the imagery is presented affects the ability to indicate motion. In a book, for example, motion must be indicated by techniques such as motion blur, tracers, or series of still images from a time sequence (small multiples) [Tuft 1996a] ([Figure 27.31](#)).

Motion also can be used to aid the viewer in discerning the three-dimensionality of the object. [Figure 27.20](#) shows an example of a 3-D set of molecules. The image in this book looks flat. However, when the molecules are rotating on a screen, a strong sense of the 3-D structure is conveyed. This representation also demonstrates how motion through the simulation can be represented in a single static geometry. In this visualization, the collection of small spheres represents the positions the atoms take over the course of the entire simulation.

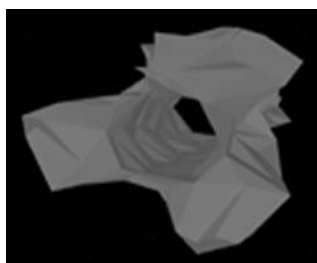


FIGURE 27.19 Alpha-shape representation of gramicidin A molecule. (Courtesy of H. Edelsbrunner, P. Fu, UIUC, NCSA.)

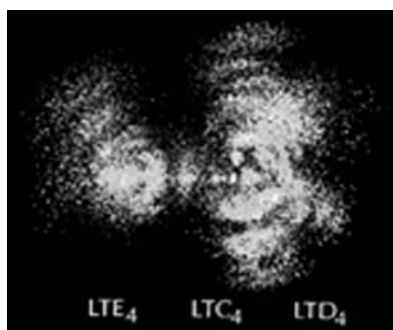


FIGURE 27.20 This image shows all of simulation time in a single geometric form and uses motion in the animation to aid the viewer in discerning the 3-D structure. (Courtesy of D. Herron, Eli Lilly & Co.; J. Thingvold, W. Sherman, NCSA.)

27.4.1.5 Transparency

Sometimes it is desirable to show something that's inside or behind something else in the scene. By increasing the transparency of the geometry, the visualizer can allow internal structure or hidden objects to be viewed. When using transparency, often many of the shadings and other cues that indicate shape are less compelling. In Figure 27.15, a transparency technique was used to allow the twisting ribbons inside the cloud structure to be seen. If the cloud had been made uniformly transparent, it would be difficult to see its shape. Here, the transparency is dependent on how nearly perpendicular the surface is to the viewpoint. This makes it easy to see the twisting ribbons in the center, while maintaining a well-defined edge to the surface.

27.4.1.6 Combining Techniques

Any of the above idioms may be combined to illustrate different aspects of the dataset (e.g., see [Figure 27.21](#)). The benefit of this is the ability to show correlations between various parameters. When combining idioms, however, care must be taken not to overload the display or occlude important information.

27.4.2 The Visualization Process

This subsection provides an overview of the process of creating various kinds of visualizations. We begin by looking at basic still images. We then add features such as choreography, real-time interaction, and, finally, simulation control from within the visualization process.

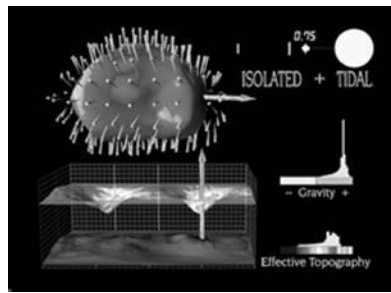


FIGURE 27.21 Multiple techniques combine to show gravity pull on the surface of Mars's moon Phobos. (Courtesy of Wayne Lytle, CTC.)

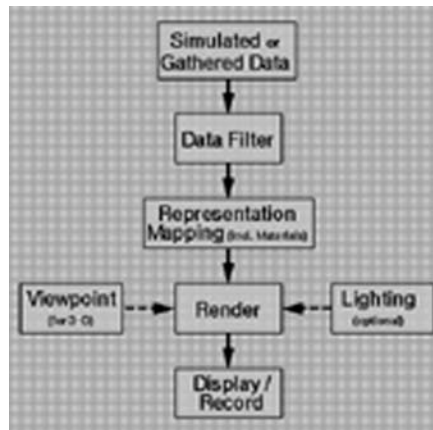


FIGURE 27.22 The basic visualization process. (Courtesy of W. Sherman.)

27.4.2.1 Still Imagery

The basic method for transforming numeric data into visual output is depicted in Figure 27.22. In this diagram, information flow follows the arrows, with dashed lines indicating optional parameters. At the top, data are provided from some numeric source such as a mathematically based computational model, a conceptually based computational model, or a collection of observed values. Data filtering involves a wide range of operations. Data must be translated into a form that is required by the visualization software. Additionally, the filtering process is used to sift out the most relevant aspects and discard unnecessary values.

After the data are filtered, they are mapped to some geometric form. At that point, decisions must be made about the materials and the characteristics of what these objects will look like — are they shiny, are they smooth, are they rough, are they dull, are they red, are they green, are they semitransparent? These parameters can also be driven by the computational model within the constraints imposed by the visualization software. Lighting information — i.e., how many lights, how diffuse they are, their relative location, etc. — is then combined with the viewpoint and information about the geometry by a computer program called the renderer. The renderer takes this information and computes the 2-D image which the eyes see. For a stereoscopic view, two images are rendered, from the point of view of each eye. The resultant images are then either recorded on film or video tape or displayed on a computer screen or other output device.

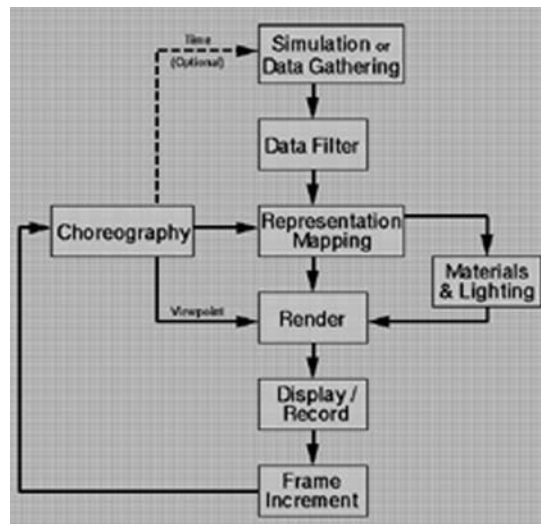


FIGURE 27.23 Process of creating a scientific visualization animation. (Courtesy of W. Sherman.)

27.4.2.2 Animation

The basic methodology for creating a computer animation is an extension of the process used to create the still image. The major extension is the concept of choreography. Now, the same steps are followed as in Figure 27.22, but the viewpoint information can be choreographed, along with the objects, their representations, and the passage of simulation time itself, resulting in the diagram in Figure 27.23. Animation is the result of rendering the scene repeatedly while varying the choreographic information and then displaying the images in rapid succession.

Choreography is that part of the process which controls the viewpoint of the scene and the movement of objects within the scene. Choreography can also be used to control how representations of the data change during an animation, such as the transparency of a material of a particular idiom.

Another important element of the dataset that can be controlled by the choreographer is *time*. Usually, time is considered to be constantly moving, at least as long as the animation is playing. However, there are two notions of time in a visualization animation. One is the passage of time of the viewer while watching the animation. The other, for visualizations of time-varying datasets, is the notion of time within the data. There is no requirement that data time move at a steady pace through the animation. It often can be insightful to hold time constant while other parameters of the representation are choreographed, revealing additional information. Moving time at different rates may reveal features that otherwise would be hidden.

27.4.2.3 Interaction

Animated sequences are made of several still images displayed in rapid succession. Often these images take longer to create than the fraction of a second that they will be displayed. This results in a situation where the choreography for all the action is planned ahead of time and is then unchangeable once the animation is complete. This is not generally the optimal method for a researcher who wants to probe a dataset interactively.

To have a more insightful experience with the data, scientists want to be able to spend more time looking at specific pieces. They want to be able to look at them from different angles, change representation parameters, and watch them at different rates of speed. To allow this kind of interaction, the images that are animated on the screen must be created as fast as they are displayed (i.e., in *real time*). Thus, the rendering stage in our visualization figure becomes a real-time (RT) process, and the display–record stage is now display only, with the option of recording (Figure 27.24).

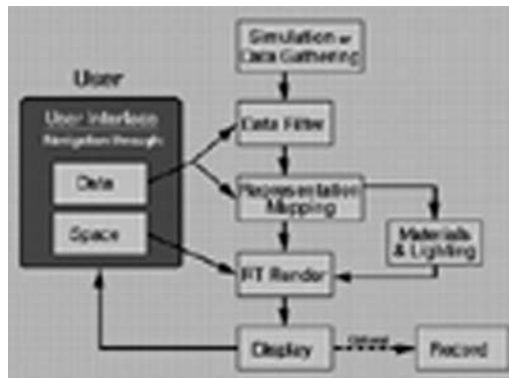


FIGURE 27.24 Interactive visualization allows the user to control the data filter, the representation, and the viewpoint. (Courtesy of W. Sherman.)

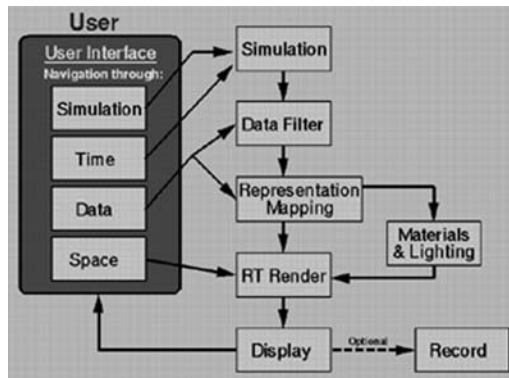


FIGURE 27.25 Interactive steering allows the user to manipulate the simulation in addition to the visualization. (Courtesy of W. Sherman.)

The most significant change in the interactive visualization process is that the choreography now happens in real time too; instead of an animator creating the choreography, the scientists themselves choreograph the scene through the package's user interface (UI). At this level of interaction, users are able to guide themselves through space and time and control the representation of the data (Figure 27.24).

27.4.2.4 Interactive Steering

Interactive steering is the ability for the user to alter the course of the simulation in real time [Haber 1989]. Figure 27.25 further extends the visualization flow chart to include user control of the simulation — including direct control over passage of time. It is vital that both the simulation and graphics system provide real-time performance and allow for user input. The user can then interactively steer the simulation by modifying variables and data in the simulation, as shown in the figure. The user can still steer the processing of the data into graphical representations and steer the viewpoint of the scene.

27.4.3 Visualization Tools

A number of tools are available for creating visualizations of information. The choices have improved with the evolution of computer technology. Visualization tools provide varying elements of the visualization processes shown in the figures. They can be categorized as follows: plotting libraries, turnkey packages,

dataflow packages, and animation packages. There is also the option of creating the visualization by writing the software in-house.

Each package (and type of package) has certain advantages and limitations that may make one more suitable for a particular application or situation. We will examine some examples of each type of visualization tool.

27.4.3.1 Plotting Libraries

Historically, computer-graphics-based visualization originated with the use of alphanumeric printouts to mimic a bar graph, a shaded “color” map, or values from which contours can be hand-plotted. Computer visualization began to take shape with the flexibility brought about by new output devices, such as pen plotters and vector displays. These new devices made the creation of computer-generated contour and vector plots more feasible.

Software libraries were developed that enabled researchers to generate charts, graphs, and plots without the need for reinventing the graphics themselves. These packages have evolved as the computers and computer I/O have improved and are still available today. Since these packages are basically programming interfaces, they do not typically involve any form of interaction other than through programming and (perhaps) a command-line interface.

The limits on the interactivity of these tools suggest that they are of limited functionality. Although this is partially true, there are also some benefits. Because scientists often want to work with quantitative representations of their data, plotting libraries can provide the easiest means for producing the appropriate output. Print output has much higher resolution than screen output, and these libraries can produce high-resolution plots specifically for output to such printers. Screen-based interactive packages often limit their resolution to what is available on the screen, thus limiting their ability to produce quantitative visualization.

27.4.3.2 Turnkey Visualization Packages

A turnkey visualization package is a program designed specifically for doing visualization and contains controls (widgets) for most options a user would want to exercise when visualizing the data. This is accomplished through the use of pulldown menus or popup windows with control panels.

A number of commercial and no-cost visualization software packages are available. Most packages are available for Unix workstations, and many are also available for PC compatibles and Apple Macintoshes. Commercial packages include Fluent’s CFD software package [Fluent], NASA/Sterling Software’s FAST [FAST], and Fortner Research’s visualization suite [Fortner]. Freely available packages include VIS-5D from the University of Wisconsin [Vis5D], Sci-An from Florida State University [Sci-An], and Gnu-Plot from Dartmouth College [Gnuplot].

There are definite advantages to using turnkey visualization software. The primary advantage is that one can often read one’s data directly into the packages and immediately begin manipulating the controls to look at various aspects of them. This assumes that the data are initially in a format readable by the package and that the controls are intuitive enough for someone to experiment with and get good results.

These packages are designed as interactive tools; their ability to manipulate the view and the visualization parameters is key to putting the visualization process into the hands of individual researchers. Also, because the target user is a scientist, the user interface is usually designed for ease of learning and use.

Off-the-shelf packages also have some limitations, however. Foremost is that their flexibility is often limited. For instance, the user won’t be able to add visualization representations that are not provided by the software. Also, if too many representations are included, then the user interface may suffer from an overwhelming number of control selections, making it hard for the user to find the control needed to perform a specific operation.

The desire for interactivity also imposes limits on these packages. To be interactive, data must be easily retrievable, and the representational forms should not be rapidly changing. This means that displaying features of time-dependent data may be too slow to be very interactive or may not be allowed by the package at all. To have interactive real-time rendering, many constraints are put on the amount of complexity the

images can have. Generally, very beautiful imagery requires more complexity than can be achieved in real time. One solution for this is to allow the user to experiment with representations and choreography of the data; when a satisfactory situation is selected, information can be sent to a noninteractive rendering system to produce nicer images.

27.4.3.3 Dataflow Packages

Like the turnkey visualization packages above, dataflow visualization packages are designed as tools to be used directly by the scientist. However, dataflow packages are much more modular, with each stage of the visualization process represented as an independent unit. These units are then connected in the appropriate manner to allow data to be passed (or *flow*) from one unit to the next. This style of interface is inherently more flexible and also provides a map of the visualization process the system is using. These packages also are designed to be more extensible, allowing the user to add features and functionality that are not provided off the shelf.

Dataflow software currently requires the availability of a Unix workstation. Most packages now run across different brands of workstations. Despite the narrowing gap between PC and workstation, not many run on PCs. Three factors that contribute to this situation are the requirement for doing a considerable amount of processing of the data, the use of large amounts of memory, and the need for large-screen displays with reasonable graphics performance.

The dataflow concept consists in breaking down tasks into small programs, each of which does one thing. Each task is represented as a *module* (a software building block, or “black box”) that performs the specified operation. Each module has a defined set of required inputs and outputs for passing information between modules. Figure 27.26 shows a simple connection network of some modules. In this example, data are retrieved with the Read HDF module and flow to the Isosurface module, which creates a geometric representation of the data, passing the new information to the Render module, which renders a pixel map, which in turn is passed to the Display module, which puts the image on the screen.

Though the dataflow concept had been described for doing other tasks (such as image manipulation), AVS (Application Visualization System) was probably the first package to apply the dataflow concept to the task of doing visualization [Upson et al. 1989, AVS]. AVS is available on most Unix workstations. IRIS Explorer [IRIS Explorer], originally developed for Silicon Graphics workstations, is also now available on other workstations. IBM's Data Explorer [IBM Data Explorer] is another example of commercially available software. Khoros [Khoros] is one freely available package of this nature. SCIRun, a powerful new implementation of the dataflow paradigm, is also freely available to some institutions [Johnson and Parker 1995]. The user interfaces for most of these programs are surprisingly similar, and familiarity with one package makes learning another easier.

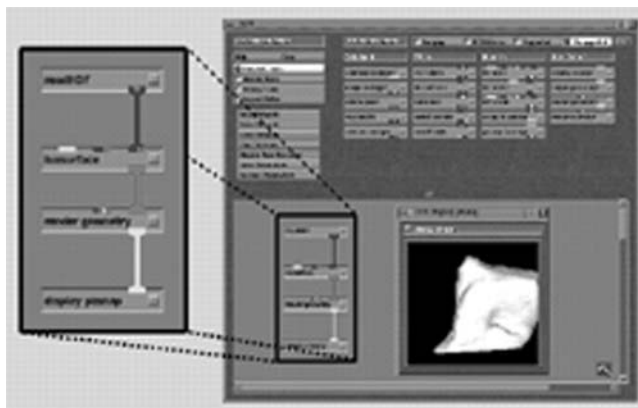


FIGURE 27.26 (See Plate 27.26 in the color insert following page 29-22.) Dataflow packages such as AVS connect modules to customize a visualization. (Courtesy of W. Sherman, NCSA.)

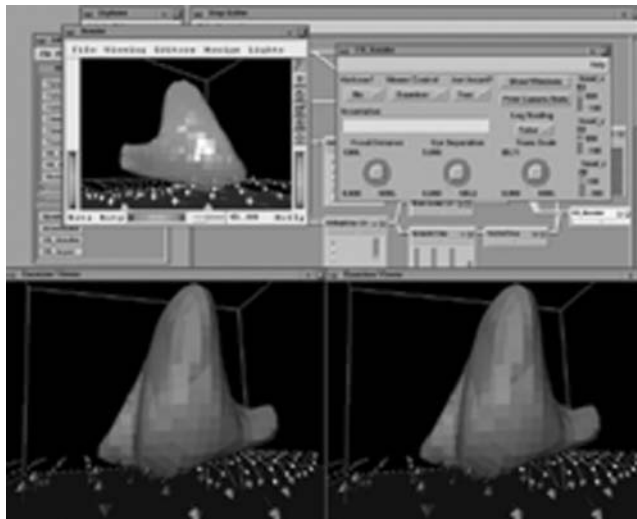


FIGURE 27.27 (See Plate 27.27 in the color insert following page 29-22.) Customization capabilities allow the package to be extended to new display paradigms, such as in this stereo image linked to a virtual-reality viewing device. (Courtesy of W. Sherman, NCSA.)

Advantages of systems designed around the dataflow paradigm are numerous. As with turnkey packages, real-time rendering coupled with user control of the viewpoint and visualization parameters allows researchers to experience their data first hand. Flexibility is enhanced by the user's ability to connect modules together, creating a network of modules specific to the task at hand.

Flexibility is raised another degree by having the option to create new modules that perform operations not provided by the base collection of modules provided with the system. New modules may be written in common programming languages, such as C or Fortran. By utilizing routines provided with the system, the new modules will exhibit the same user interface and data handling as the standard modules. Since many people typically share the same desire for specific extensions, sharing of modules is common practice, thus making these packages even more valuable.

There are some disadvantages to this form of visualization package. For many, the primary disadvantage is the fact that most dataflow packages are available only on Unix workstations. Another is the difficulty with which this type of package handles data that change over time as part of a real-time visualization. Typically, data flowing down the network is for a particular time step of the computational simulation, and the modules that provide the representation of particle flow through a vector field will not be able to get the data fast enough to give smooth-flowing animation of those particles in a changing dataset (though they generally can handle particles flowing through a time-static dataset).

And last, the dataflow interface *is* a type of programming; thus, the end user at one level must learn how to program the system. While easier than writing code in C or Fortran, this process is more complicated than in the turnkey approach. The degree of difficulty increases as the networks become complex and the application has greater requirements, such as looping time-varying data that are flowing through the system.

The modularity of the dataflow paradigm also allows for the addition of a new style of interfacing with the user. New input devices such as Spaceballs™ and 3-D trackers can allow the system to control the viewpoint of the environment. New output devices can provide a head-tracked, stereo view, allowing *virtual reality* to become a potential method of interacting with a visualization [Sherman 1990] (Figure 27.27).

27.4.3.4 Animation

Today, presentation-quality visualization animations typically use standard computer animation packages [Fangmeier 1988]. Until recently, computer animation packages were the only off-the-shelf software

available that provided 3-D rendering of any sort. Thus, these were often used to provide the rendering stage of the visualization process.

Computer animation packages are designed for all types of computer animation, not just for use in scientific visualization. However, it can be said that whatever they are used for, they are used to show (or visualize) the world the animator has conceptualized. For scientific visualization, the objects in the scene are typically derived from data. An animation package typically has separate components that allow the modeling of objects; the choreography of the objects, lights, and viewpoint; and the ability to design visual material qualities for each of the objects. These are all tied together into an image or series of images by the rendering component.

The primary advantage of using animation packages is that they are extremely flexible. The user is able to create objects of any type or size to represent the subject of the visualization and any supporting objects that might be required to give context to the representations. In a visualization environment, most of the objects are created by writing specialized programs that convert data from their native format to a format suitable for the animation package. This flexibility does have limits, however, such as when there is a need to represent the data using a method not available in the package. For example, a volume rendering technique may be required, and the renderer for the package may support only polygonal techniques.

Another advantage of animation packages is the ability to create very high-quality imagery. It is important to note, though, that the image quality is also dependent on the work done by the visualization creator.

The flexibility and complexity of animation packages also can be a disadvantage. These tools can be sufficiently complex that to use them as a visualization tool requires an outside expert to bring the visualization to fruition. Such an expert would be able to write the custom programs that convert data from their original state to an appropriate format. The fact that the consultant doing the visualization work is usually not an expert in the field of study being represented also can be a disadvantage.

Because each image in an animation can be very detailed, rendered using computationally expensive techniques that give rise to high-quality scenes, the time required to create the image is unlikely to be short enough to allow for what is referred to as real-time rendering. In fact, it can take several minutes to several hours to produce an image that will be viewed for one-thirtieth of a second (at video rates). The time required is a disadvantage even in situations where months are available to produce the animation, because of the delay between the time a small change is made and the time the scientist can view the resulting work.

Another disadvantage of using an animation package is the increased potential loss of data integrity that can occur in the process of translating the data from their original form to a form suitable for the animation package. The likelihood of data integrity problems is higher in this environment for two reasons: more custom work is done to translate the data, and the animation package may have options to make the output look good at the expense of giving an accurate view of the object being rendered. In an off-the-shelf visualization package, the techniques used to bring data to the screen are often well-known ones that a user can research and learn how they may affect the resulting image. Custom code used with animation packages may not be as heavily scrutinized if it is used only once for a particular application. Also, animation packages have features such as automatic smoothing that take the rough spots out of the data. This changed representation probably will not give an accurate view of the original information.

27.4.3.5 WYOS: Write Your Own Software

Before visualization and animation packages were available, tools to look at data were custom programmed for the task at hand. This is still done for certain visualizations — especially ones that handle large amounts of time-varying data or require new computer graphics techniques that have not been implemented in off-the-shelf software (e.g., certain volume visualization techniques). Now, however, users generally choose to use off-the-shelf software, perhaps with a few modifications.

Some examples of visualizations requiring custom renderers include *L.A. — The Movie* [Hussey et al. 1986], which used special techniques for overlaying satellite imagery onto a topological map of the Los Angeles region, and *The Deluge* [Sims 1989], which brings a still image to life by using a renderer written specifically for that task.

The only advantage to writing your own application software is in situations where that is the only way you can obtain the visualization you need.

27.4.3.6 Tools Summary

In summary, many tools have been developed which aid in visualizing scientific data. These tools are improving, making visualization an easier task to perform. However, even with a very easy user interface, the skill of the visualization developer is the overriding issue. No tool, no matter how easy to use, can replace the skill and insight of a visualization expert.

27.4.4 Examples of Scientific Visualizations

We have explained the components of scientific visualization and examined how these are used to create output that the researcher can use. In order to have a better understanding of how visualization is typically used in practice, and to look at how representational idioms have evolved, we will describe a selection of visualizations used as a means for scientific study.

27.4.4.1 The Study of a Severe Thunderstorm

The primary example of scientific visualization we will examine in this chapter is the computational simulation of severe thunderstorms. The simulation of meteorological processes has a long history in the computational sciences and consequently can be used to demonstrate the evolution of many visualization techniques. In particular, we will look at the research performed by Robert Wilhelmson's research group at NCSA and the Department of Atmospheric Sciences at UIUC and the visualizations done by his group and the NCSA Visualization Group.

27.4.4.1.1 Evolution of Atmospheric Simulations and Visualization

Wilhelmson's group has been studying the process of thunderstorm development computationally for 25 years. Over this period, several developments have been made in scientific visualization, and these are reflected in the many visualizations made of his work.

In the 1950s, the simulations run on digital computers were so simple that the researchers could literally examine every number produced. In the 1960s, digital computers had become fast enough to be used to execute 2-D simulations of atmospheric conditions over time. At first, the data were examined simply by printing the numbers in a 2-D array on paper. The researcher could then actually draw contours through the data by hand based on the printed numbers. An alternative early technique was to print a 2-D array of letters, with each letter representing a range of data values (this was often referred to as "gray-shading," with each character effectively representing a different level of gray) (Figure 27.28). About this time, a package for producing standard visualizations of 2-D data was released by the National Center for Atmospheric Research (NCAR). This package, known as *NCAR-graphics* [NCAR] could produce representations such as the image of contour levels within the developing storm front shown in Figure 27.29. In general, all the visual output of the simulations were of a static slice of time.

The 1970s saw the true advent of 3-D simulation models for all areas of atmospheric science. With the expansion into three dimensions, it became more difficult to interpret and explain the results. By this time, *NCAR-graphics* was the primary visualization package used by atmospheric sciences. With the addition of the ability to produce 3-D isocontours (i.e., isosurfaces) (Figure 27.30), Wilhelmson was able to view storms from a real-world perspective and from a radar perspective, among others.

Other techniques that began to be used in the late 1970s include the use of color to visually separate different representations, giving the ability to view them simultaneously. Color also was used to draw different layers of isosurfaces within a 3-D structure. In addition to color, Wilhelmson began to look at the change of the system over time; initially, this was limited to an alternation between two images on a raster-frame storage device.

In the early 1980s, the use of flow tracers (particles) to visualize flow through a field began in earnest — though some had been done in the late 1970s. Computing particle location was a new way to represent

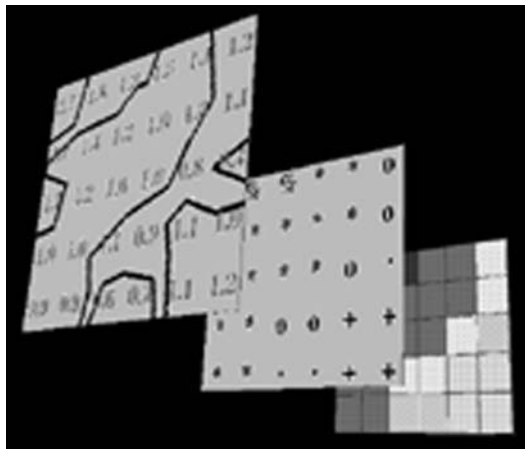


FIGURE 27.28 Early visualization attempts included hand-drawn contours and shaded images constructed from a judicious choice of ASCII characters.

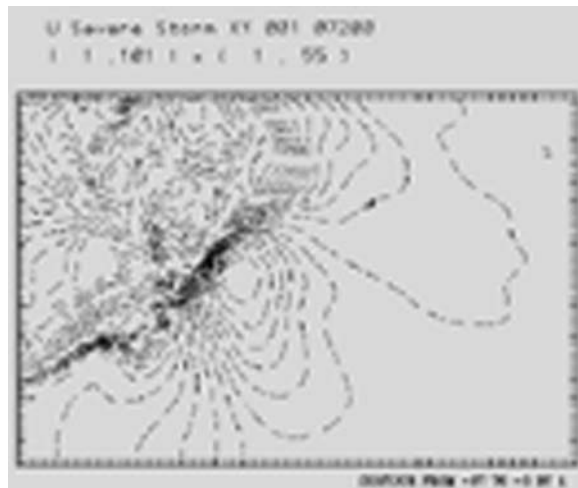


FIGURE 27.29 Developing storm front.



FIGURE 27.30 An early isosurface representing the late stage of the storm.

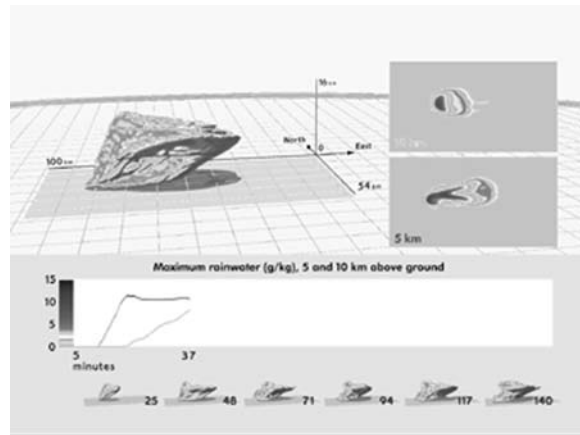


FIGURE 27.31 The thunderstorm visualization was redesigned with less contrast between the grid and the ground and small multiples added to represent changes in time. (Courtesy of Yale University, NCSA.)

flow through a fluid substance, providing the ability to view a large region of the simulation at one time. In practice, the visualization creator experiments with where the particles should be released to show specific aspects of the simulation. Several examples of the storm visualization produced in 1989 can be seen in [Figures 27.13](#), [27.15](#), and [Figure 27.18](#).

In line with the development of visualization technology is the continued exploration of effective information design. The goal of visualization is to communicate aspects of the data clearly. To achieve this clarity, one must be aware of basic design principles and have the ability to effectively critique the design quality of visualization. Issues of visual hierarchy, composition, choreography, color theory, and typography cannot be disregarded. Attempts to demonstrate scale, provide a sense of context, combine qualitative and quantitative information simultaneously, direct the viewer's eye, create multiple representations, and point out intentional manipulations in the data are tasks that the creator must work through in order to produce quality information design [Tufte 1996a].

The storm visualization shown in [Figure 27.18](#) was redesigned to illustrate several issues of information design. In the new design ([Figure 27.31](#)), the contrast between the elements of the scene was adjusted so that visual emphasis is placed on the important details of the visualization. The grid floor in the original animation is a strong visual element and distracts from the dynamics of the cloud development. Reference information was added to the scene, such as the dimensions of the computational domain and the diagrammatic clock at the bottom. This clock includes a stripe of small images that illustrate the entire duration of the visualization. This gives contextual information by indicating which portion of the animation one is viewing and in addition shows the numerical time step for specific reference. The two-dimensional cutting planes in the scene in conjunction with the three-dimensional cloud, and the quantitative scale at the bottom, provide multiple representations of the data for comparison and clarification [Baker and Bushell 1995, Tufte 1996b].

Today, we are reaching the point where the flow of large numbers of particles can be computed and visualized. Too many particles, however, can make it hard to see what is going on, so the technique of selectively choosing to display specific particles is being explored. In the picture in [Figure 27.32](#), particles are used to represent the flow of a tornado. To highlight just the tornado, only particles with high vertical vorticity, within a certain region, are shown. The amount of data and objects required to represent it also brings about a logistical problem of where all this information is stored. Large disk drives are required to create visualizations of massive numbers of particles, and files must be carefully managed to stay within the constraints of the storage.

Virtual reality (VR) also is being experimented with as a means of gaining insights from simulated scientific experiments [Baker and Wickens 1995]. VR experiences have been designed for the VROOM

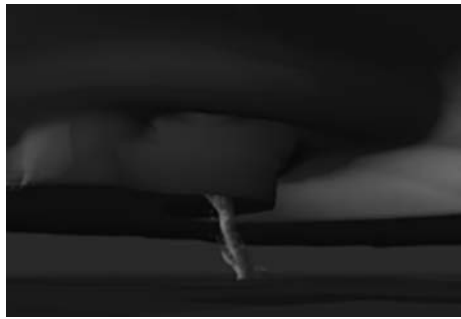


FIGURE 27.32 Thousands of particles show the shape of a tornado. (Courtesy of Matthew Arrott.)

[Brown 1994] and SuperVROOM [Korab and Brown 1995] venues at SIGGRAPH '94 and SuperComputing '95, respectively. In these experiences, the domain of the data is represented by a bounding box, with various tools available to represent the data, including isosurfaces, and particle flow. In a CAVETM [Cruz-Neira et al. 1992] VR system, the user can easily navigate the data space and has the ability to control the representations. In the particle flow representation, the user can control the release of particles with a hand-held device. As the user releases the particles, a flow simulation program is begun that updates the particle positions. This program (and any of the representation modules) can execute on a separate larger supercomputer for better interactivity.

It is hard to judge how much additional insight can be gained by immersive interactive visualizations (i.e., virtual reality). This stems from the deep knowledge atmospheric scientists already have of their simulations. Major concerns of the scientist in using virtual reality include: Would I use a device that requires me to be standing or walking for an extended period? Analyzing the data requires knowing where one is with respect to the cloud, which is easier when viewing the cloud from the outside — in which case, what is the advantage over the desktop? To be truly immersed, the user should feel the wind, hear the thunder, see the lightning, none of which is accurately representable with current VR technology.

Scientists have become proficient at deciphering the 3-D content from flat 2-D displays from years of experience. Because of this, they see less need for the improved 3-D cues that VR provides. However, they are continuing to explore the use of virtual reality as a visualization tool. Since it is a new medium, there are many technological and user interface improvements ahead; VR still may prove to provide unique insight not obtainable through other means.

The World Wide Web (Figure 27.33) is viewed as a very useful tool for disseminating research information. It can be used by the public to access weather forecast information or by other atmospheric researchers to directly examine the simulation data themselves. It can be the complete tool for a researcher — the notebook that contains all the diary entries over the course of a study, including explanations, data, images, animations, and interactive visualization tools. It can be made available not only to the researcher but also to colleagues, advisors, and (if desired) the scientific community [Jewett and Wilhelmson 1995].

27.4.4.1.2 What Does the Scientist Use Today?

With all that is available to the scientist today, what tools are actually used? To be specific, everything. Line graphs, 2-D contours, and 2-D vector plots still provide the basis for closely investigating the results of a simulation. Three-dimensional stills and animations are excellent tools for illustrating concepts. Interactive visualization tools help to find out what's in the dataset and to locate interesting regions that demand close inspection.

Animation, while a good tool for presenting the overall contents of a research study, poses problems when it comes to publishing the study. Scientific publishing still exists primarily as a print medium, and

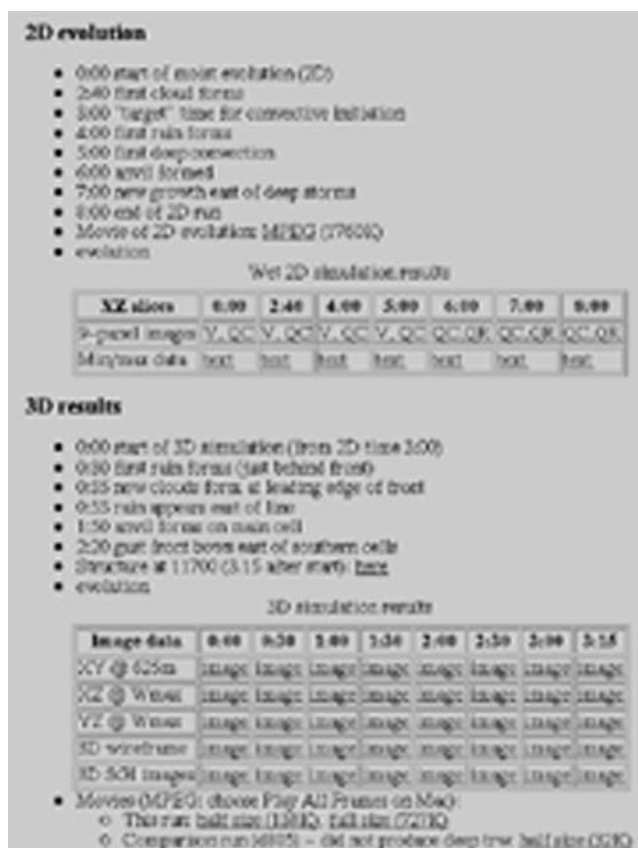


FIGURE 27.33 Portion of a WWW page used to document a research project. (Courtesy of Brian Jewett; NCSA.)

thus it is difficult to include animation. Indeed, even submitting an animation for inclusion in the review process is difficult. The World Wide Web is beginning to provide a solution to these problems. The entire research submission, including animations and interactive tools, can be included as one comprehensive document for the review. Once reviewed, such a submission can be published on CD-ROM, while still remaining available online where it can be periodically updated.

27.4.5 Visualizing Smog

Smog: Visualizing the Components is another animation visualization in the domain of atmospheric science. However, there are some interesting information presentation features that are highlighted in this animation.

One segment of the *Smog* animation has particles that represent different forms of pollutants flowing through the atmosphere, based on air movement data. Unlike most particle flow animations where particles are released from locations within the simulation to highlight specific aspects of flow, the particles in this animation emanate from actual locations of pollution sources, at a rate based on measured releases. This animation also demonstrates a slightly different form of the wind-tracing ribbons known as streaklines (Figure 27.14).

An additional representation of the data in this animation was made through the use of sound. A sirenlike tone reinforces the information displayed in the XY plot (Figure 27.34). It also experimented with mapping the ozone level to the repetition rate of digital recordings of coughs.

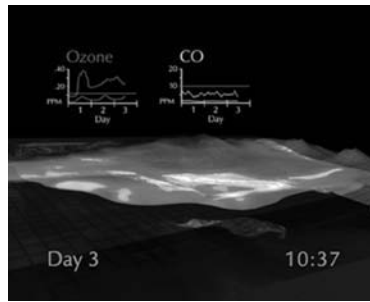


FIGURE 27.34 Los Angeles smog visualization. (Courtesy of NCSA.)

27.5 Research Issues and Summary

Though great strides have been taken in developing scientific visualization as a tool in computational science, a multitude of issues remain for further research and development. As computer power increases, the need for effective visualization tools increases due to the larger volume of information being produced. Fortunately, the increase in computer power also allows us to develop more compelling visualization tools with more flexibility, better performance, and higher fidelity.

When studying the field of visualization, it is important to recognize that visualization is being done effectively in a variety of fields outside of the realm of computational science. Medical imaging, factory automation, and financial researchers, to name a few, are taking advantage of the power of visual representation of information.

Additionally, there is a growing wealth of knowledge about the visual representation of information, and how we perceive and interpret visual information, being published by researchers in fields such as cognitive psychology, perceptual psychology, human factors, etc. As visualization researchers and tool developers incorporate new findings, we may take better advantage of visualization as a powerful analysis and communication tool. The art and design community also is demonstrating new ways of presenting and representing information. The use of sound in representing scientific data is becoming more important. Research is being done to find what types of information are best represented in audio.

In order to move beyond the current level of visualization tools, we must continue to develop the basic technologies and algorithms involved in the image-generation aspects of the process — for example, improved rendering algorithms, improved user interface, improved representations, etc.

In the ideal world, the computational scientist would not be required to think about human visual perception, optical illusions, how cultural biases affect the interpretation of imagery, or how color is understood. If research in the appropriate areas can be integrated into the tools which scientists use, we can raise the level of effectiveness of the visual analysis and presentation of science and, indeed, the science itself. Research is currently under way [Baker 1994] to develop tools which do automatic idiom selection. Based on input from the scientists regarding the type of data, the insight they are trying to achieve, etc., the tool will generate representations that have a high likelihood of being useful.

Visualization is rapidly moving beyond the researcher's workstation and videotape presentations. New media and communications systems are allowing the scientist to collaborate remotely over networks and share data, imagery, etc. The Internet and the World Wide Web are allowing the scientist to publish findings, including visual representations, to a worldwide audience. These new media bring a variety of new concerns and research issues. Some examples include the size and resolution of imagery that it is possible to share. Current network speeds render it impractical to share multiple large, high-resolution images. Animations also are just beginning to be practical over the network. As data compression techniques and sharing of executable code over the network improve, the scientist will truly be able to explore this new medium.

Current work in VR techniques is enabling scientists to see and interact with their research in many new ways. Stereoscopic imagery, wide fields of view, and tracking of the viewer's head enable the scientist

to become immersed within the simulation and become a part of the system. Current systems suffer from a lack of resolution, tracker lag, etc. As these improve, virtual reality promises to be a powerful tool for computational scientists. Many scientists who have currently abandoned virtual reality systems with their clumsy, low-resolution head-mounted displays are finding that the less intrusive, higher-resolution projection-based displays such as the CAVETM are transforming virtual reality from a novelty to a useful tool for analysis and display.

In summary, the field of scientific visualization is sufficiently mature to allow the scientist to harness the power of the eye–brain connection for data analysis and presentation. At the same time, it is breaking new ground with respect to online collaborative computing, worldwide publication, and fully immersive interaction through virtual reality.

Acknowledgments

We would like to thank Robert Wilhelmson for sharing his wealth of experience and historical perspective on visualization in computational science; also, from the Atmospheric Science Group, Crystal Shaw and Brian Jewett for additional support. Thanks also to the reviewers and editors for helpful comments and suggestions to improve this chapter.

Many of the ideas in this chapter were the result of our interactions with a variety of scientists. We have had the opportunity to work on a wide variety of challenging visualization problems with numerous world-class researchers.

We gained a much greater insight into the field of scientific visualization as a direct result of interaction with the participants of the Representation Project, including the guest speakers from a wide variety of disciplines and institutions. In particular, the members of the former Visualization Group at NCSA including Boss Dan Brady, Matthew Arrott, Mark Bajuk, Ingrid Kallick, Mike Krogh, Mike McNeill, Gautam Mehrotra, Jeff Thingvold, and Deanna Spivey. This group was intellectually stimulating, an endless source of talent and ideas, and most of all, our friends. We feel honored to have been part of such a magical team.

Defining Terms

Alpha shapes: A technique that allows one to represent the concept of shape applied to a collection of points in space.

Choreography: In computer animation, the timing and sequencing of activity and representation.

Computer graphics: The medium in which modern visualization is practiced.

Dataset: A value or set of values that is the input or output of a computational simulation.

Glyph: Generally, a symbol used to represent information. For example, in visualizing vector fields, an arrow often is used to show the direction and magnitude of the vector value at each location.

Interactive steering: The practice of dynamically modifying the parameters of a running simulation, guided by a real-time visualization of the simulation's progress.

Isosurface: The shape defined within a volume of scalar values on which all the values are equal to some constant.

Parallax: The difference in the apparent position of an object caused by a change in the point of observation.

Perceptualization: A term perhaps more suitable than “visualization,” in recognition of the efficacy of using auditory and tactile techniques for representing and communicating about scientific data.

Scientific illustration: The traditional use of graphics created by an artist to show scientific concepts.

Scientific visualization: The use of computer-generated graphics, often animated, interactive, and three-dimensional, to represent scientific data and concepts.

Simulation: A computer model of natural phenomena.

Streakline: A line showing the path taken by all particles that pass through a given location in a vector field.

- Streamline:** A line drawn in a vector field such that, at any instant, the tangent to the line at any point on the line is the direction of the flow. Often restricted to fields with steady flow, in which case the streamline shows the path of a tracer particle.
- Tracer:** An animated symbol, usually a sphere, showing the path that would be taken by a particle in a vector field.
- Vector field:** An n -dimensional collection of vector values arranged in space, such as the wind velocity over a two-dimensional surface.
- Virtual reality:** A medium composed of highly interactive computer simulations that utilizes data about the participant's position and replaces or augments one or more of their senses — giving the feeling of being immersed, or being present, in the simulation [Sherman and Craig 1996].
- Visual idiom:** A technique for representing scientific data that has a commonly accepted interpretation.

References

- Anninos, P., Bajuk, M., Bernstein, Seidel, E., Smarr, L., and Hobill, D. 1993. The evolution of distorted black holes. *Physics Computing* '93.
- Arrott, M. and Latta, S. 1992. Perspectives on visualization, pp. 61–65. *IEEE Spectrum*, Sept.
- AVS. AVS Home Page, <http://www.avs.com/>.
- Bajuk, M. 1992. Camera evidence: visibility analysis through a multi-camera viewpoint.
- Baker, M. P. 1994. KnowVis: an experiment in automating visualization, p. 456. In *Proc. Decision-Support 2001*, Toronto, Ontario, Sept.
- Baker, M. P. and Bushell, C. 1995. After the storm: considerations for information visualization. *IEEE Trans. Comput. Graphics Appl.* 15(3):12–15 (May).
- Baker, M. P. and Wickens, C. D. 1995. Human factors in virtual environments for the visual analysis of scientific data. *NCSA Tech. Rep.* 032, Aug.
- Brooks, F. P., Jr., Ming O.-Y., Batter, J. J., and Kilpatrick, P. J. 1990. Project GROPE — haptic displays for scientific visualization. *Comput. Graphics (Proc. SIGGRAPH)* 24(4):177–185 (Aug.).
- Brown, M., ed. 1994. *Comput. Graphics, SIGGRAPH '94 Visual Proceedings*, Aug., Orlando, FL.
- Cruz-Neira, C., Sandin, D., DeFanti, T., Kenyon, R., and Hart, J. 1992. The CAVE audio visual experience automatic virtual environment. *Comm. ACM* 35(6):64–72. (June). URL: <http://www.ncsa.uiuc.edu/EVL/docs/html/CAVE.html>.
- Drebin, R. A., Carpenter, L., and Hanrahan, P. 1988. Volume rendering. *Comput. Graphics (Proc. SIGGRAPH)* 22(4):64–75 (Aug.).
- Edelsbrunner, H. and Mucke, E. P. 1994. Three-dimensional alpha shapes. *ACM Trans. Graphics* 13:43–72.
- Fangmeier, S. M. 1988. The scientific visualization process, pp. 26–38. *SIGGRAPH '88 Course Notes*, Course 20: Computer Graphics in Science.
- FAST. Flow Analysis Software Toolkit (FAST) home page. <http://www.nas.nasa.gov/NAS/FAST/fast.html>. See also Sterling Software WWW Home Page. <http://www.sterling.com/>.
- Fluent. Fluent Incorporated Home Menu Page. Fluent CFD software. <http://www.fluent.com/>.
- Fortner. Fortner Research LLC. <http://www.langsys.com/langsys/>.
- Gnuplot. Gnuplot. http://www.cs.dartmouth.edu/gnuplot_info.html.
- Haber, R. B. 1989. Scientific visualization and the rivers project at the National Center for Supercomputing Applications. *Computer* 22(8):84–89 (Aug.).
- Herron, D. K., Bollinger, N. G., Chaney, M. O., Varshavsky, A. D., Yost, J. B., Sherman, W. R., and Thingvold, J. A. 1995. Visualization and comparison of molecular dynamics simulations of leukotriene C(4), leukotriene D(4), and leukotriene E(4). *J. Mol. Graphics* 13:337–341, Elsevier Science, New York.
- Hussey, K., Mortensen, B., and Hall, J. 1986. Jet Propulsion Lab Animation: L.A. — *The Movie*. Visualization in scientific computing. *ACM SIGGRAPH Video Review*, No. 28.
- IBM Data Explorer. IBM Visualization Data Explorer (DX). <http://www-1.almaden.ibm.com/dx/>.

- IRIS Explorer. IRIS Explorer Center. http://www.nag.co.uk:70/1h/Welcome_IEC.
- Jewett, B. F. and Wilhelmson, R. B. 1995. Use of HTML and web tools in atmospheric sciences research. URL: <http://redrock.ncsa.uiuc.edu/AOS/publications/IIPS96/web-atmossci.html>.
- Johnson, C. R. and Parker, S. G. 1995. Applications in computational medicine using SCIRun: a computational steering programming environment, H. W. Meuer, ed., pp. 2–19. In *Supercomputing '95*. URL: <http://www.cs.utah.edu/~sci/>.
- Kaufmann, W. J., III, and Smarr, L. L. 1993. *Supercomputing and Science*. Scientific American Library, New York.
- Keates, J. S. 1996. *Understanding Maps*, Hallstead Press. p. 39.
- Khoros. The Khoros Research Home Page. <http://www.khoros.unm.edu/home.html>.
- Korab, H. and Brown, M., eds. 1995. Virtual environments and distributed computing at SC '95. ACM/IEEE allstead Press. Supercomputing '95, Dec., San Diego, CA.
- Kovacic, D., Craig, A., Patterson, R., Romme, W., and Despain, D. 1990. *Fire Dynamics in the Yellowstone Landscape, 1690–1990: An Animation*. Model driven visual simulation, Proc. Resource Technology 90, Second International Symposium on Advanced Technology in Natural Resources Management.
- Lorensen, W. E., and Cline, H. 1987. Marching cubes: a high resolution 3-D surface construction algorithm, *Comput. Graphics (Proc. SIGGRAPH)* 21(4):163–169 (July).
- Lytle, W. 1993. The dangers of glitziness and other visualization faux pas. Animation, Cornell Theory Center, *SIGGRAPH 93 Electronic Theater*. No. 91.
- Marshall, R., Kempf, J., and Dyer, S. 1990. Visualization methods and simulation steering for a 3-D turbulence model of Lake Erie. Symposium on Interactive 3-D Graphics. *ACM SIGGRAPH* 24(2):89–97 (Mar.).
- Merzkirch, W. 1987. *Flow Visualization*, 2nd ed. Academic Press, Orlando, FL.
- Moran, P. J. and Potter, C. S. 1992. Tiller: a tool for analyzing 4-d data. In *Visual Data Interpretation*, Proc. SPIE, 1668:124–128.
- NCAR. NCAR Graphics Home Page. <http://ngwww.ucar.edu/>.
- Scaletti, C. and Craig, A. B. 1991. Using sound to extract meaning from complex data. In *Extracting Meaning from Complex Data: Processing, Display, Interaction II*, Proc. SPIE, 1459:207–219.
- SciAn. SciAn — Scientific Visualization and Animation Package. <http://www.scri.fsu.edu/~mimi/scian.html>.
- Sherman, W., McNeill, M., Arrott, M., Bajuk, M., and Corson, A. 1990. Animation: Smog: Visualizing the Components. SIGGRAPH '90 Film & Video Theater. *ACM SIGGRAPH Video Rev.*, No. 62.
- Sherman, W. R. 1990. Integrating virtual environments into the dataflow paradigm. Fourth Eurographics Workshop on Visualization in Scientific Computing, Abingdon, U.K., Apr.
- Sherman, W. R. and Craig, A. B. 1997. *Working with Virtual Reality*. Morgan Kaufmann, San Francisco.
- Sims, C. 1989. Animation: Leonardo's Deluge SIGGRAPH '89 Computer Graphics Theater. *ACM SIGGRAPH Video Rev.*, No. 52.
- Tiede, U., Schiemann, T., and Hohne, K. H. 1996. Visualizing the visible human. In *IEEE Computer Graphics & Applications* 16(1).
- Tufte, E. R. 1996a. *Envisioning Information*. Graphics Press, Cheshire, CT.
- Tufte, E. R. 1996b. *Visual Explanations*. Graphics Press, Cheshire, CT.
- Upson, C., Faulhaber, T., Kamins, D., Laidlaw, D., Vroom, J., Gurwitz, R., and van Dam, A. 1989. The application visualization system: a computational environment for scientific visualization. *IEEE Trans. Comput. Graphics Appl.* 9(4):30–42 (July).
- Vis5D. Vis5D Home Page. <http://www.ssec.wisc.edu/~billh/vis5d.html>.
- Weintraub, D. J. and Walker, E. H. 1969. *Perception*. Brooks/Cole, Belmont, CA.
- Wickens, C., Merwin, D., and Lin, E. 1994. The human factors implications of graphics enhancements for the visualization of scientific data. In *1994 Human Factors*, Vol. 36. Also in *Proceedings of the 38th Annual Meeting of the Human Factors Society (1994)*, pp. 44–61.
- Wickens, C. and Seidler, K. 1995. Information access and utilization. In *Emerging Needs and Opportunities for Human Factors Research*, R. Nickerson, Ed., National Academy Press, Washington, D.C.

Wilhelmson, R., 2nd. 1993. PATHFINDER: Probing ATmospHeric Flows in an INteractive and Distributed EnviRonment. In *Proc. 9th Conf. on Interactive Information and Processing Systems for Meteorology, Oceanography, and Hydrology*, Anaheim, CA, Jan.

Further Information

Brown, J. R., Earnshaw, R., Jern, M., and Vince, J. 1995. *Visualization: Using Computer Graphics to Explore Data and Present Information*. Wiley, New York.

Chambers, J., Cleveland, W., Kleiner, B., and Tukey, P. 1983. *Graphical Methods for Data Analysis*. Wadsworth International Group, Belmont, CA.

Dent, B. D. 1990. *Cartography: Thematic Map Design*. William C. Brown, New York.

Dondis, D. A. 1973. *A Primer of Visual Literacy*. MIT Press, Cambridge, MA.

Friedhoff, R. M. and Benzoni, W. 1989. *Visualization*. W. H. Freeman, New York.

Gallager, R. S., ed. 1995. *Computer Visualization*. CRC Press.

Hearn, D. and Baker, M. P. 1994. *Computer Graphics*, 2nd ed. Addison–Wesley.

Huff, D. 1954. *How to Lie with Statistics*. Norton, New York.

Kaufmann, W. J., III, and Smarr, L. L. 1993. *Supercomputing and Science*. Scientific American Library, New York.

Keates, J. F. *Understanding Maps*.

Keller, P. R. and Keller, M. M. 1993. *Visual Cues*. IEEE Press.

Lauer, D. A. 1990. *Design Basics*. Harcourt Brace Jovanovich.

MacEachren, A. M. 1995. *How Maps Work*. Guilford Press, New York.

McCormick, B. H., DeFanti, T. A., and Brown, M. D. 1987. Visualization in scientific computing. *Comput. Graphics* 21(6): (Nov.).

Tufte, E. R. 1983. *The Visual Display of Quantitative Information*. Graphics Press, Cheshire, CT.

Wickens, C. 1992. *Engineering Psychology and Human Performance*, 2nd ed. HarperCollins, New York.

28

Computational Structural Mechanics

- 28.1 Introduction
- 28.2 Classification of Structural Mechanics Problems
 - Structural Characteristics and Source Variables
 - Different Classes of Structural Mechanics Problems
 - Deterministic and Nondeterministic Methods
- 28.3 Formulation of Structural Mechanics Problems
 - Different Formulations of Structural Mechanics Problems
 - Description of the Motion of a Structure
- 28.4 Steps Involved in the Application of Computational Structural Mechanics to Practical Engineering Problems
 - Major Steps in the Application of Computational Structural Mechanics • Selection of the Mathematical Models
 - Discretization Techniques • Model and Mesh Generation
 - Quality Assessment and Control of Numerical Solutions
- 28.5 Overview of Static, Stability, and Dynamic Analysis
 - Static Analysis • Dynamic Analysis • Energy Balance in Transient Dynamic Analysis • Stability Analysis • Eigenvalue Problems • Sensitivity Analysis • Strategies and Numerical Algorithms for New Computing Systems
- 28.6 Brief History of the Development of Computational Structural Mechanics Software
- 28.7 Characteristics of Future Engineering Systems and Their Implications on Computational Structural Mechanics
- 28.8 Primary Pacing Items and Research Issues
 - High-Fidelity Modeling of the Structure • Failure and Life Prediction Methodologies • Hierarchical, Integrated Multiple Methods and Adaptive Modeling Techniques
 - Nondeterministic Analysis, Modeling, and Risk Assessment
 - Validation of Numerical Simulations • Multidisciplinary Analysis and Design Optimization • Related Tasks

Ahmed K. Noor
Old Dominion University

28.1 Introduction

Structural mechanics deals with (1) the idealization of actual structures and their environments and (2) prediction of response, failure, life, and performance of structures. In the last three decades the discipline of computational structural mechanics (CSM) has emerged as an insightful blend between structural mechanics, on the one hand, and other disciplines such as computer science, numerical analysis, and

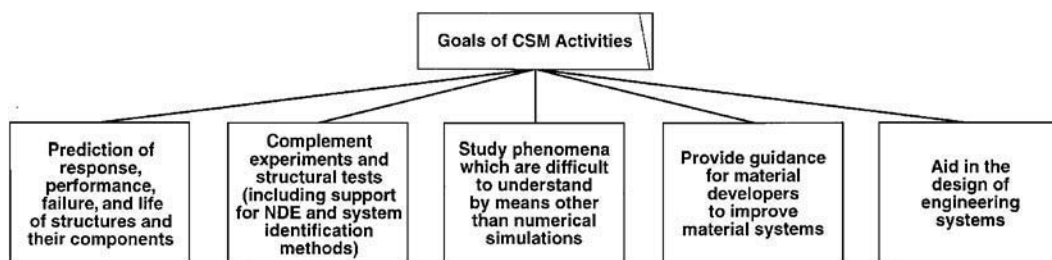


FIGURE 28.1 Five major goals of CSM activities (NDE refers to nondestructive evaluation techniques).

approximation theory, on the other hand. This rapidly evolving discipline is having a major impact on the development of structures technology, as well as on its application to various engineering systems. Development of the modern finite-element method during the 1950s marks the beginning of CSM. Finite-element technology is the backbone of many structural analysis software systems which are widely used by government, academia, and industry to solve complex structures problems.

The five major goals of CSM activities are shown in Figure 28.1. In support of these goals, current activities of CSM cover the study of phenomena, through numerical simulations, at a wide range of length scales ranging from the microscopic level to the structural level. Today, no important design can be completed without CSM, nor can any new theory be validated without it.

A number of survey papers and monographs have been written on various aspects of CSM (see, for example, Noor and Atluri [1987], Noor and Venneri [1990, 1995], and Ju [1995]). Also, a number of workshops and symposia have been devoted to CSM and proceedings have been published (for example, Grandhi et al. [1989], Noor et al. [1992], Onaté et al. [1992], Ladevéze and Zienkiewicz [1992], Stein [1993], and Storaasli and Housner [1994]). The present chapter attempts to present, in a concise manner, the broad spectrum of problems covered by CSM along with the basic principles, formulations, and solution techniques for these problems. A brief history is given of the development of software systems used for the modeling and analysis of structures. The research areas in CSM, which have high potential for meeting future technological needs, are identified.

28.2 Classification of Structural Mechanics Problems

28.2.1 Structural Characteristics and Source Variables

The functions which govern the response, failure, life, and performance of structures can be grouped into four categories, namely:

Kinematic variables: e.g., displacements, velocities, strains, and strain rates

Kinetic variables: e.g., stresses and internal forces (or stress resultants)

Material characteristics: e.g., material stiffness and compliance coefficients

Source variables: which include environmental effects and external forces (e.g., mechanical, aerodynamic, thermal, optical, and electromagnetic forces)

The relations between the external forces and response quantities are shown in [Table 28.1](#).

28.2.2 Different Classes of Structural Mechanics Problems

A number of classifications can be made for structural mechanics problems depending on: (1) presence or absence of uncertainties about the structural characteristics and source variables, (2) nature of the functions which govern the behavior of the structure (e.g., time dependence or independence), (3) the functional

TABLE 28.1 Relations Between External Forces and Response Quantities

Quantities	Relation	Type of Relation
External forces	Balance equations	Physical (conservation) laws
Stresses	Constitutive relations	Semi-empirical based on experiments
Strains Displacements	Strain-displacement relations	Geometric-based on logic

form of the relations between the source variables and response quantities (e.g., linear or nonlinear), and (4) the geometric characteristics of the structure and its components. A general classification, incorporating the aforementioned factors, is shown in [Figure 28.2](#). Additional classifications can be made based on: (1) material response (e.g., homogeneous or nonhomogeneous, isotropic or anisotropic), (2) nature of source variables (e.g., conservative or nonconservative), and (3) coupling or noncoupling between source variables and response quantities (e.g., whether the changes in the source variables with structural deformations are pronounced or not).

28.2.3 Deterministic and Nondeterministic Methods

Deterministic methods of structural mechanics have become quite elaborate and include sophisticated mathematical models, highly refined computational methods, and optimization techniques. However, there is a growing realization among engineers during the past 25 years that unavoidable uncertainties in geometry, material properties, boundary conditions, loading, and operational environment must be taken into account to produce meaningful designs.

Three possible approaches for handling uncertainty can be identified, depending on the type of uncertainty and the amount of information available about the structural characteristics and the operational environment. The three approaches are: *probabilistic analysis*, *fuzzy-logic approach*, and *set theoretical, convex* (or antioptimization) *approach*. A discussion of the three approaches and their combinations is given in Elishakoff [1995]. In the probabilistic analysis, the structural characteristics and/or the source variables are assumed to be random variables (or functions), and the joint probability density functions of these variables are selected. The main objective of the analysis is the determination of the reliability of the system. Reliability is defined as the probability that the structure will adequately perform its intended mission for a specified interval of time, when operating under specified environmental conditions.

If the uncertainty is due to vague definition of structural and/or operational characteristics, imprecision of data and subjectivity of opinion or judgment, then fuzzy logic-based treatment is appropriate. The distinction between randomness and fuzziness is the fact that whereas randomness describes the uncertainty in the occurrence of an event (e.g., damage or failure of the structure), fuzziness describes the ambiguity of the event (e.g., imprecisely defined criteria for failure or damage — see Ross [1995]).

When the information about the structural and/or operational characteristics is fragmentary (e.g., only a bound on a maximum possible response function is known), then convex modeling can be used. Convex modeling produces the maximum or least favorable response, and the minimum or most favorable response, of the structure under the constraints within the set-theoretical description.

28.3 Formulation of Structural Mechanics Problems

28.3.1 Different Formulations of Structural Mechanics Problems

Several classifications can be made of the different formulations used for structural mechanics problems. Two of the major classifications are discussed next.

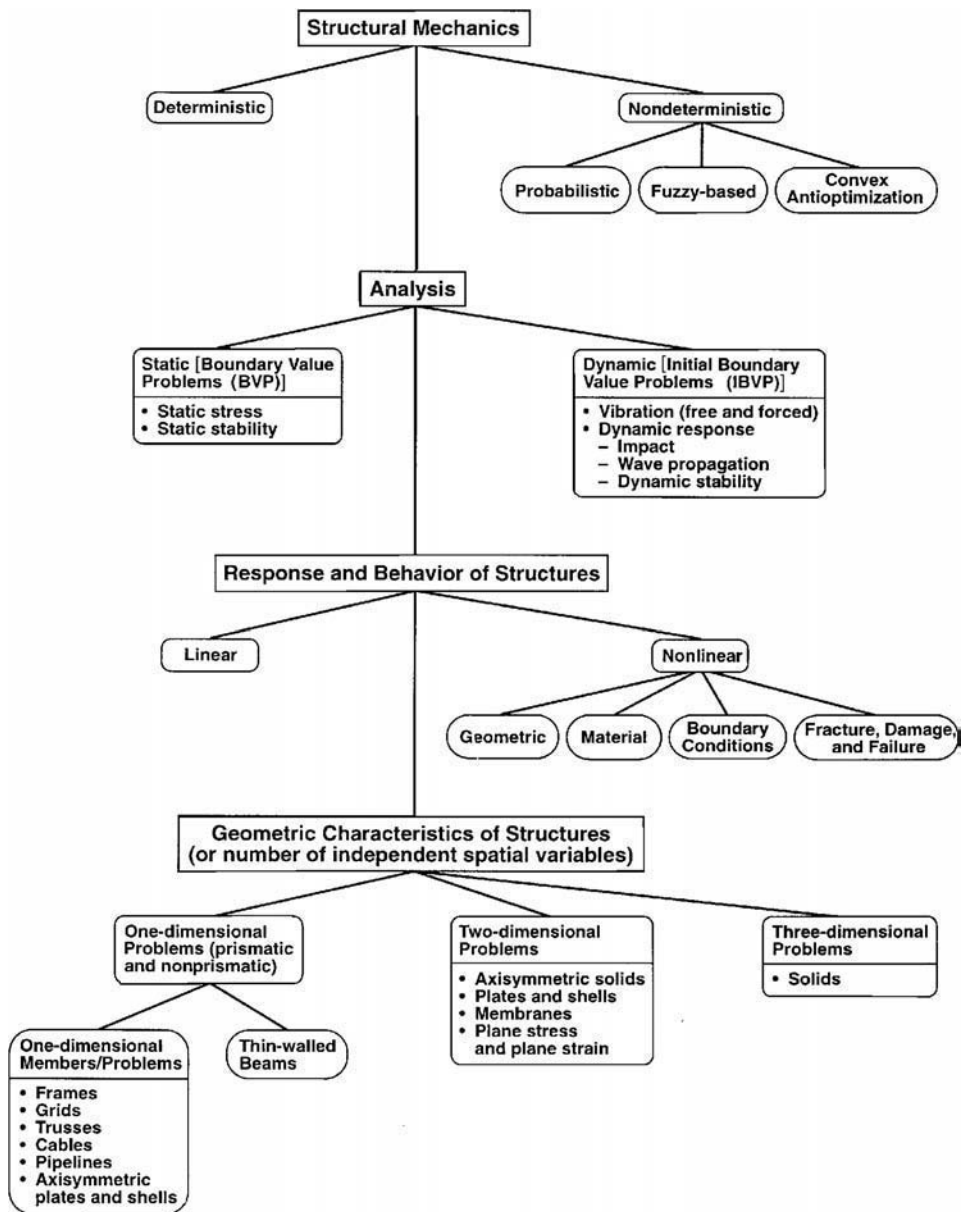


FIGURE 28.2 Classification of structural mechanics problems.

First, with respect to the approach used in deriving governing equations:

1. Differential equation formulation. This is the classical approach which is referred to as the *strong statement* of the problem.
2. Variational formulation. This is based on application of the principle of virtual displacements, and is referred to as the *weak statement* of the problem. A variety of variational principles for static and dynamic problems can be derived from the principle of virtual displacements (see, for example, Washizu [1982] and Geradin [1980]).
3. Integral, integro-differential or boundary integral equation formulation.

Some of the discretization techniques can be applied directly to the strong statement of the problem. The most notable example is the finite-difference method. Other techniques, such as finite elements, are typically used with the weak statement. Boundary element methods are generally used with the third formulation. Second, with respect to fundamental unknowns in the governing equations or the governing functional:

1. Single-field formulation in which the fundamental unknowns belong to one field. The most commonly used formulation is the displacement formulation.
2. Multifield formulation in which the fundamental unknowns belong to more than one field, i.e., stresses and displacements. A detailed discussion of multifield formulations is given in Noor [1980].

28.3.2 Description of the Motion of a Structure

For nonlinear structural problems, three basic choices need to be made, namely, the approach used for describing the motion, the kinematic description (i.e., deformation measure), and the kinetic description (i.e., stress measure). Typically, the choice of the deformation measure determines the stress measure since the two have to be work conjugates. The two most commonly used approaches for describing the motion are as follows.

First is a *total Lagrangian description*, in which all the kinematic and kinetic variables are referred to the initial (undeformed) configuration. The Green–Lagrange strain tensor is used as the measure of deformation, and the associated second Piola–Kirchhoff stress tensor is used as the stress measure. For large-rotation problems, it is useful to separate the rigid-body movement from stretching by using *local corotational frames* and the polar decomposition method. This eliminates the problems associated with approximating finite rotations using trigonometric functions (or series expansions thereof).

Second is an *updated Lagrangian description*, in which the kinematic and kinetic variables are referred to the current configuration. For small-strain problems, the Almansi strain is used as the measure of deformation and the associated Cauchy stress is used as the stress measure. For large-strain problems, it is convenient to use the velocity strain (or rate of deformation) and the Jaumann stress rate as the deformation and stress measures.

A detailed discussion of the aforementioned formulations and their combinations, along with the appropriate deformation and stress measures is given in Bathe [1996] and Cescotto et al. [1979].

28.4 Steps Involved in the Application of Computational Structural Mechanics to Practical Engineering Problems

28.4.1 Major Steps in the Application of Computational Structural Mechanics

The application of CSM to contemporary structural problems involves a sequence of five steps, namely (Figure 28.3):

- Observation of response phenomena of interest.
- Development of computational models for the numerical simulation of these phenomena. This in turn includes: (1) selecting mathematical models which represent the environmental effects and external forces and describe the phenomena, analyzing the models to ensure that the problem is properly formulated, and testing the range of validity of the models; and (2) development of a discrete model, computational strategy, and numerical algorithms to approximate the mathematical model. Successful computational models for structures are based on a thorough familiarity with the response phenomena being simulated and a good understanding of the mathematical models available to describe them.
- Development and assembly of software and/or hardware to implement the computational model.
- Postprocessing and interpretation of the predictions of the computational model.
- Utilization of the computational model in the analysis and design of engineering structures.

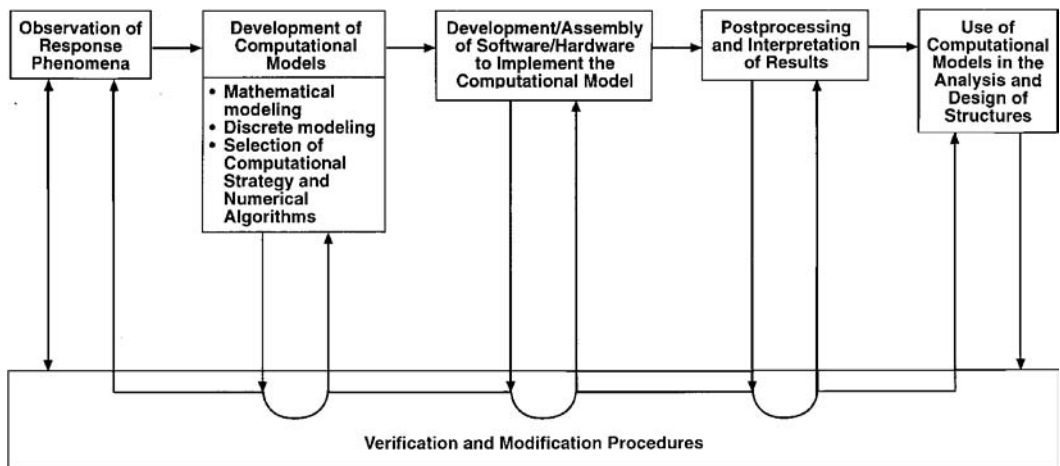


FIGURE 28.3 Application of CSM to practical structural problems.

Within this general framework, CSM is being used today in a broad range of practical applications. To date, large structural calculations are performed which account for complicated geometry, complex loading history, and material behavior. The applications span several industries, including aerospace, automotive, naval, nuclear, and microelectronics.

28.4.2 Selection of the Mathematical Models

The mathematical models are idealized representations of the real structure and its environment. Proper selection of the mathematical models is strongly influenced by the goals of the computation. The models will represent reality only if they take into consideration all factors which affect the conclusions drawn from them. The mathematical models are described by their governing equations (in one of the forms described in the previous section).

It is useful to view a particular mathematical model as one in a sequence (or hierarchy) of models of progressively increasing complexity. The effect of the model selection on the accuracy of the response predictions for simple structures, boundary conditions, and joints is discussed in Szabo and Babuska [1991]. A number of comparisons can be made between the predictions of the model being formulated and more elaborate models, i.e., models which account for a greater number of potentially relevant phenomena.

28.4.3 Discretization Techniques

Because of the complexity of the governing equations of the mathematical models used in representing real structures, exact solutions can only be obtained in very few cases, and one has to resort to approximate or numerical discretization techniques. A variety of approximate and numerical discretization techniques have been applied to structural mechanics problems. Two possible classifications of these techniques are shown in Figure 28.4 and are based on: (1) the formulation used, namely, differential equation, variational, or integral-equation formulation; and (2) modifications made in the form of the governing equations (replacement of the governing equations by an equivalent form).

The finite-element method is the most commonly used discretization technique to date. Extensive literature exists on various aspects of the finite-element method. A list of monographs, books, and conference proceedings on the method is given in Noor [1991]. The boundary element method is a computational tool for the boundary integral equation formulation. The method works with values of the dependent

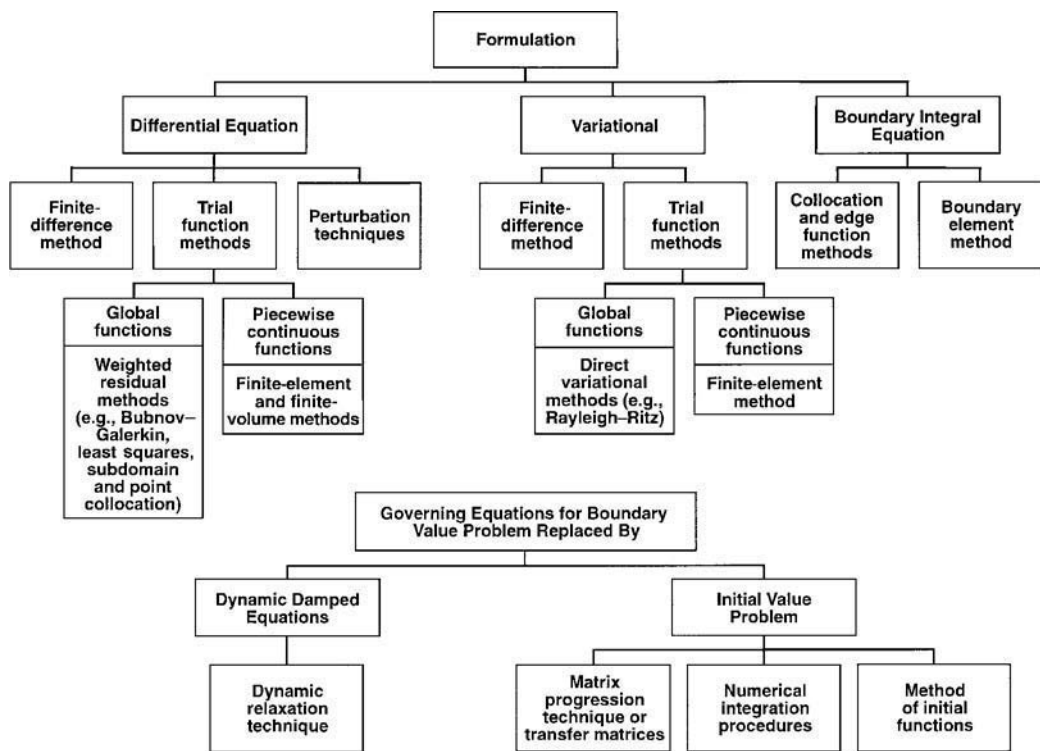


FIGURE 28.4 Approximate and numerical discretization techniques for structural mechanics problem.

variables on the boundary only, and therefore, is well suited to problems involving a large volume to surface ratio (Kane et al. [1993], Banerjee [1994]). In a number of applications, hybrid combinations of analytical and numerical discretization techniques were shown to be more effective than individual techniques, and resulted in dramatic savings in the computational effort (see, for example, Noor and Andersen [1992] and Noor [1994]).

28.4.4 Model and Mesh Generation

The reliability of the predictions of numerical discretization techniques (e.g., finite differences, finite elements, finite volumes, and boundary elements), and the computational effort involved in obtaining them, is very much influenced by the selection of the mesh and the procedure for generating it. Considerable effort has been devoted to the development of robust mesh generation procedures capable of producing controlled meshes over domains of arbitrary complex geometry. Finite-element mesh generation activities have been approached from both geometric modeling and adaptive finite-element viewpoints. Overviews and classifications of finite-element mesh generation methods are given in Shephard and Wentorf [1994], Shephard [1988], Ho-Le [1988], George [1991], Sabin [1991], and Mackerle [1993].

Among the recent activities on model generation are: (1) application of knowledge-based analysis-assistance tools, which allow a simple description of the analysis objectives and generate the corresponding discrete models appropriate for these objectives (Turkiyyah and Fenves [1996]); and (2) the development of paving and plastering techniques for automated generation of quadrilateral and hexahedral finite-element grids. These techniques generate well-formed elements (with reasonably small distortion metric), and are based on iteratively layering or paving rows of elements to the interior of a region's boundary (see Blacker and Stephenson [1991]).

28.4.5 Quality Assessment and Control of Numerical Solutions

Assessment of the reliability of computational models has been the focus of intense efforts in recent years. These efforts can be grouped into three general categories (see, for example, Noor and Babuska [1987], Noor [1992], Demkowicz et al. [1992], Krizek and Neittaanmaki [1987]): a posteriori error estimation, superconvergent recovery techniques, and adaptive improvement strategies. A posteriori error estimates use information generated during the solution process to assess the discretization errors. In superconvergent recovery techniques, more accurate values of certain response quantities (e.g., derivatives of fundamental unknowns) are calculated than those obtained by direct finite-element calculations. Adaptive strategies attempt to improve the reliability of the computational model by controlling the discretization error.

28.4.5.1 Error Estimation

Two broad classes of error estimation schemes are currently used: residual methods and interpolation methods. Residual methods involve the use of local residuals, usually as data in a local auxiliary problem designed to generate the local error to an acceptable accuracy. A significant amount of computation may be required in implementing these methods. In interpolation methods the available approximate solution for a given mesh (or time step) is used to estimate higher derivatives locally (e.g., local gradients or second derivatives). The higher derivatives are used in turn to determine the local error. Although these error estimates can be very crude, they are portable: a subroutine for computing local estimates can be added to virtually any existing code that operates on unstructured meshes with some effort.

Although significant progress has been made in developing a posteriori error estimates for linear elliptic problems, the error estimates for nonlinear and time-dependent problems are considerably less developed. This is particularly true for bifurcation problems, problems with multiple scales, and problems with resonance. Work on error estimation for highly nonlinear problems has mainly been a subject of ad hoc experimentation.

28.4.5.2 Superconvergent Recovery Techniques

Superconvergent recovery techniques refer to simple postprocessing techniques which provide increased accuracy of the sought quantities at some isolated points (e.g., Gauss-Legendre, Jacobi, or Lobatto); in a subdomain; or even in the whole domain (Krizek and Neittaanmaki [1987]). In the latter two cases, the techniques are referred to as local- and global-superconvergent recovery techniques, respectively. Recent work included development of local-superconvergent patch derivative techniques for both interior and boundary (or material interface) points (Babuska and Miller [1984a, 1984b], Zienkiewicz and Zhu [1992], Zienkiewicz et al. [1993], Tabbara et al. [1994]). It was shown in Zienkiewicz and Zhu [1992] and Zienkiewicz et al. [1993] that the superconvergent recovery technique can be used to obtain a posteriori error estimates for the finite-element solution.

28.4.5.3 Adaptive Strategies

Different strategies have been used for adaptive improvement of the numerical solutions, including: (1) mesh refinement (or derefinement) schemes, h methods; (2) moving mesh (node redistribution) schemes, r methods; (3) subspace enrichment schemes (selection of the local order of approximation), p methods; (4) mesh superposition schemes (overlapping local finite-element meshes on the global one), s methods (see, for example, Fish and Markolefas [1993, 1994]); and (5) hybrid (or combined) schemes. Examples of these schemes are: (1) simultaneous selection of the meshes and local order of approximation, h - p methods; recent theoretical results have shown that the fastest possible convergence rates can be attained by optimally decreasing the mesh size h and increasing the degree of the polynomial degree p in a special way; and (2) simultaneous selection of the meshes and node redistribution, h - r method. These methods can be effective in shock problems since an r -method might align the mesh along discontinuities prior to a mesh refinement.

28.5 Overview of Static, Stability, and Dynamic Analysis

Flow charts for the basic components of the solution methods for static stability and dynamic problems are given in Figure 28.5 and Figure 28.6. In this section a brief summary is given of the fundamental equations and solution techniques used in static, stability, and dynamic analysis. A single-field displacement formulation is used, and the spatial discretization of the structure is assumed to have been performed. The external load vector and associated displacement vector will henceforth be denoted by $\{\mathbf{F}^{\text{ext}}\}$ and $\{\mathbf{X}\}$, respectively. For linear problems $\{\mathbf{X}\}$ is a linear function of the components of $\{\mathbf{F}^{\text{ext}}\}$, and for nonlinear problems $\{\mathbf{X}\}$ is a nonlinear function of these components.

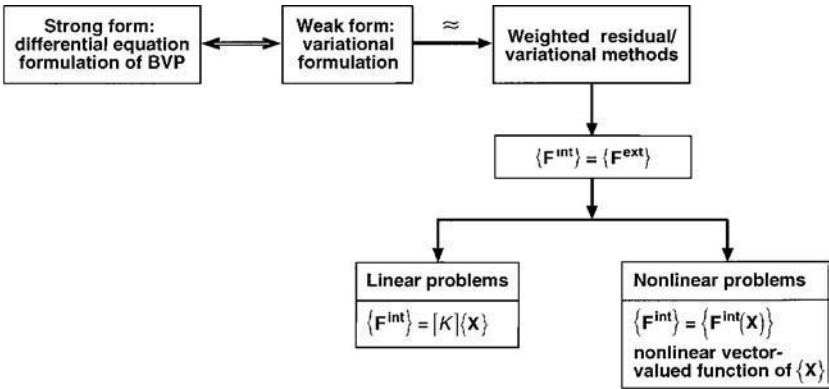


FIGURE 28.5 Basic components of solution methods for static problems.

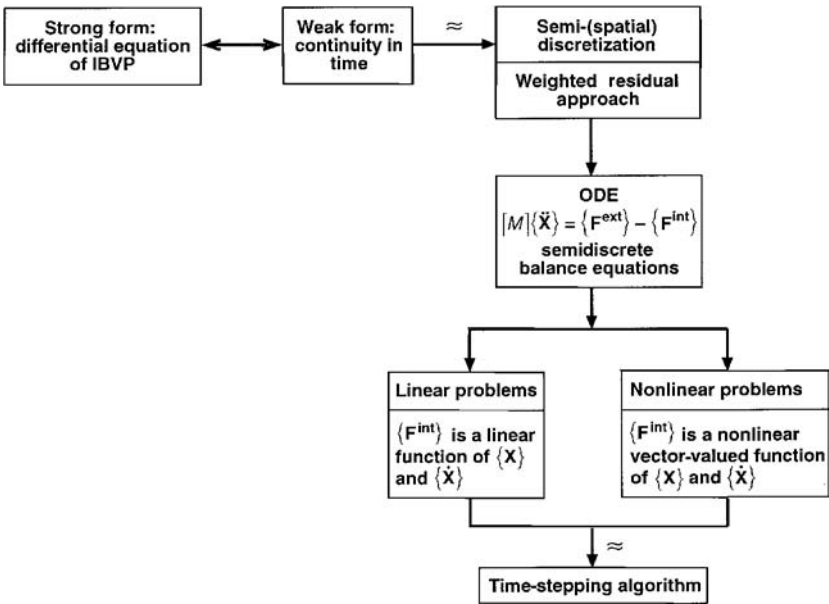


FIGURE 28.6 Basic components of solution methods for dynamic problems.

28.5.1 Static Analysis

The equilibrium equations for the discretized structure can be written in the following form:

$$\{\mathbf{F}^{\text{int}}(\mathbf{X})\} = \{\mathbf{F}^{\text{ext}}\} \quad (28.1)$$

where $\{\mathbf{F}^{\text{int}}(\mathbf{X})\}$ is the vector of internal forces, which is a vector-valued function of the displacements $\{\mathbf{X}\}$. For conservative loading $\{\mathbf{F}^{\text{ext}}\}$ is independent of $\{\mathbf{X}\}$, and for nonconservative loading it is a function of $\{\mathbf{X}\}$.

The equilibrium path is usually expressed in terms of one or more variable path parameters (typically taken as load, displacement, or arc-length parameters in the solution space). For simplicity, in subsequent discussion the loading is assumed to be conservative and proportional to a single path parameter p . The displacement vector is therefore a function of p , i.e., $\{\mathbf{X}\} = \{\mathbf{X}(p)\}$.

For linear problems

$$\{\mathbf{F}^{\text{int}}(\mathbf{X})\} = [K]\{\mathbf{X}\} \quad (28.2)$$

where $[K]$ is the stiffness matrix of the discretized structure.

For nonlinear problems the solution of Equation 28.1 can be obtained by using an incremental or an incremental/iterative procedure. The two procedures are described next.

28.5.1.1 Incremental Procedure

If a purely incremental procedure (without iteration) is used, the equilibrium equations associated with the n th increment of the path parameter can be written in the following form:

$$\{\mathbf{F}^{\text{int}}\}_{n+1} = \{\mathbf{F}^{\text{int}}\}_n + \{\Delta\mathbf{F}^{\text{int}}\}_n \quad (28.3)$$

$$\{\mathbf{X}\}_{n+1} = \{\mathbf{X}\}_n + \{\Delta\mathbf{X}\}_n \quad (28.4)$$

where

$$\{\Delta\mathbf{F}^{\text{int}}\}_n = \{\mathbf{F}^{\text{ext}}\}_{n+1} - \{\mathbf{F}^{\text{int}}\}_n \quad (28.5)$$

$$\cong [K]_n\{\Delta\mathbf{X}\}_n \quad (28.6)$$

and

$$[K]_n = \left[\frac{\partial\{\mathbf{F}^{\text{int}}\}}{\partial\{\mathbf{X}\}} \right]_n \quad (28.7)$$

In Equation 28.3, Equation 28.4, and Equation 28.6, $\{\mathbf{F}^{\text{int}}\}_n$, $[K]_n$ and $\{\mathbf{X}\}_n$ refer to the vector of internal forces, tangent stiffness matrix, and displacement vector at the beginning of the n th increment.

Note that Equation 28.6 is approximate, and therefore, a purely incremental approach should be used with a sufficiently small step of the path parameter, so that the departure from the equilibrium position is small.

28.5.1.2 Incremental-Iterative Procedures

Incremental-iterative procedures are predictor-corrector continuation methods. For any increment of the path parameter, the displacement vector at the beginning of the increment is used to calculate suitable approximations (predictors) for the displacement vectors and the internal force vectors at the end of that increment. The approximations are then chosen as initial estimates for $\{\mathbf{X}\}$ and $\{\mathbf{F}^{\text{int}}\}$ in a corrective-iterative scheme. The process is described by the following recursive equations for the i th iteration cycle of the n th increment.

$$\{\mathbf{R}\}^{(i)} = \{\mathbf{F}^{\text{ext}}\}_{n+1} - \{\mathbf{F}^{\text{int}}\}_{n+1}^{(i-1)} \quad (28.8)$$

$$= [K]\{\Delta\mathbf{X}\}_n^{(i)} \quad (28.9)$$

and

$$\{\mathbf{X}\}_{n+1}^{(i)} = \{\mathbf{X}\}_{n+1}^{(i-1)} + \{\Delta\mathbf{X}\}_n^{(i)} \quad (28.10)$$

where $\{\mathbf{R}\}$ is the residual force vector.

The iterative process is continued until convergence. As a test for convergence, a number of error norms can be used. Two of the error norms are described next. First is the modified Euclidean (spectral) norm,

$$|e| = \frac{1}{n} |\Delta\mathbf{X}|/|\mathbf{X}| \leq \text{tolerance} \quad (28.11)$$

where n is the total number of displacement parameters in the model. Second is the energy norm,

$$|e| = \frac{[\{\Delta\mathbf{X}\}^{(i)}]^T [\{\mathbf{F}^{\text{ext}}\}_{n+1} - \{\mathbf{F}^{\text{int}}\}_{n+1}^{(i-1)}]}{[\{\Delta\mathbf{X}\}^{(i)}]^T [\{\mathbf{F}^{\text{ext}}\}_{n+1} - \{\mathbf{F}^{\text{int}}\}_n]} \leq \text{tolerance} \quad (28.12)$$

In Equation 28.8 to Equation 28.10 superscripts $(i-1)$ and (i) refer to the values of the vectors at the beginning and end of the i th iteration cycle and $[K]$ is an approximation to the tangent stiffness matrix. A number of different iterative processes are distinguished by the choice of $[K]$.

28.5.1.3 Newton–Raphson Technique

The matrix $[K]$ is selected to be the tangent stiffness matrix based on the solution at the end of iteration cycle $(i-1)$, i.e.,

$$[K] = \left[\frac{\partial \{\mathbf{F}^{\text{int}}\}}{\partial \{\mathbf{X}\}} \right]_{n+1}^{(i-1)} = [K]_{n+1}^{(i-1)} \quad (28.13)$$

28.5.1.4 Modified Newton Method

The matrix $[K]$ is selected to be the tangent stiffness matrix associated with increment m of the path parameter, where $m \leq n$, i.e.,

$$[K] = [K]_m \quad (28.14)$$

28.5.1.5 Broyden–Fletcher–Goldfarb–Shanno (BFGS) Method

A secant approximation to the stiffness matrix is used in successive iterations, through updating the inverse of the stiffness matrix using vector products. This is equivalent to updating the stiffness matrix, based on iteration history, as follows:

$$[K] = [K]_{n+1}^{(i-1)} \quad (28.15)$$

28.5.2 Dynamic Analysis

The balance equations of the discretized structure can be written in the form of a system of ordinary differential equations in time as follows:

$$[M]\{\ddot{\mathbf{X}}\} = \{\mathbf{F}^{\text{ext}}\} - \{\mathbf{F}^{\text{int}}\} \quad (28.16)$$

where $[M]$ is the global mass matrix and $\{\ddot{\mathbf{X}}\}$ is the acceleration vector. The explicit characterization of $[M]$, $\{\ddot{\mathbf{X}}\}$, and $\{\mathbf{F}^{\text{int}}\}$ depends on the particular structure under consideration. When damping or viscous effects exist, the vector $\{\mathbf{F}^{\text{int}}\}$ is a function of both the displacement vector and the velocity vector, i.e.,

$$\{\mathbf{F}^{\text{int}}\} = \{\mathbf{F}^{\text{int}}(\mathbf{X}, \dot{\mathbf{X}})\} \quad (28.17)$$

For linear problems, $\{\mathbf{F}^{\text{int}}\}$ is a linear function of both $\{\mathbf{X}\}$ and $\{\dot{\mathbf{X}}\}$, i.e.,

$$\{\mathbf{F}^{\text{int}}\} = [\mathbf{K}]\{\mathbf{X}\} + [\mathbf{C}]\{\dot{\mathbf{X}}\} \quad (28.18)$$

where $[\mathbf{C}]$ is the global damping matrix.

For nonlinear problems $\{\mathbf{F}^{\text{int}}\}$ is a nonlinear function of $\{\mathbf{X}\}$ and $\{\dot{\mathbf{X}}\}$. At any time instant, the tangent stiffness and tangent damping matrices are given by

$$[\mathbf{K}] = \left[\frac{\partial \{\mathbf{F}^{\text{int}}\}}{\partial \{\mathbf{X}\}} \right] \quad (28.19)$$

$$[\mathbf{C}] = \left[\frac{\partial \{\mathbf{F}^{\text{int}}\}}{\partial \{\dot{\mathbf{X}}\}} \right] \quad (28.20)$$

The approaches used for obtaining the response-time history of the structure [solution of Equation 28.16] can be divided into two general categories, namely, direct integration techniques and modal superposition methods. The application of the two approaches to nonlinear dynamic analysis is described next.

28.5.2.1 Direct Integration Techniques

Direct temporal integration techniques are time-stepping (or step-by-step) strategies in which the response vectors at an initial time are used to generate the corresponding vectors at subsequent times. The techniques are based on: (1) satisfying the balance equations only at discrete time intervals and (2) assuming the functional dependence of the response vectors within each time interval. A variety of approximations for the response vectors within each time interval have been applied to structural dynamics problems. The approximations for the velocity and displacement vectors in the n th time step can be expressed as follows:

$$\{\dot{\mathbf{X}}\}_{n+1} = \frac{\alpha_1}{\Delta t} \{\ddot{\mathbf{X}}\}_{n+1} + L(\{\dot{\mathbf{X}}\}_n, \{\ddot{\mathbf{X}}\}_n, \dots) \quad (28.21)$$

$$\{\mathbf{X}\}_{n+1} = \frac{\alpha_2}{(\Delta t)^2} \{\ddot{\mathbf{X}}\}_{n+1} + M(\{\mathbf{X}\}_n, \{\dot{\mathbf{X}}\}_n, \{\ddot{\mathbf{X}}\}_n, \dots) \quad (28.22)$$

where Δt is the time-step size, α_1 and α_2 are coefficients used in the approximation, L and M are functions of the response vectors, and subscripts n and $n + 1$ refer to the values of the vectors at the beginning and end of the n th time step.

Temporal integration techniques can be classified into two general categories, namely, explicit and implicit techniques. In explicit methods the response at the end of a time step is evaluated using the balance equations at the beginning of the time step [α_1 and α_2 in Equation 28.21 and Equation 28.22 are both zero]. By contrast, implicit methods use the balance equations at the end of the time step, with either α_1 and/or α_2 in Equation 28.21 and Equation 28.22 as nonzero.

Explicit techniques generally require fewer computations per time step than implicit techniques and can easily handle complex nonlinearities. However, the time-step size must often be very small to ensure numerical stability. By contrast, for linear problems, the time-step size in implicit techniques is only restricted by accuracy requirements. However, based on available information, unconditional stability of implicit methods in linear problems does not necessarily extend to nonlinear problems. Explicit and implicit techniques can also be classified into *single-step* and *multistep methods* according to whether the response at any instant is related to the response at one or more previous times. Detailed discussions and assessments of different explicit and implicit techniques are given in Belytschko and Hughes [1983] and Belytschko et al. [1987]. Mixed explicit-implicit techniques are described in Belytschko and Hughes [1983].

Herein, the application of two of the widely used temporal integration techniques, central difference explicit scheme and the Newmark implicit scheme is described. Both are single-step methods.

28.5.2.1.1 Central Difference Explicit Scheme

The central difference scheme is based on the following approximations for the velocity and the acceleration vectors:

$$\{\dot{\mathbf{X}}\}_{n+\frac{1}{2}} = \frac{1}{\Delta t} (\{\mathbf{X}\}_{n+1} - \{\mathbf{X}\}_n) \quad (28.23)$$

$$\{\ddot{\mathbf{X}}\}_n = \frac{1}{(\Delta t)^2} [\{\mathbf{X}\}_{n-1} - 2\{\mathbf{X}\}_n + \{\mathbf{X}\}_{n+1}] \quad (28.24)$$

where Δt is the time-step size, and subscripts n , $n + \frac{1}{2}$, and $n + 1$ refer to the values of the vectors at the beginning, middle, and end of the n th time step.

Based on Equation 28.25 and Equation 28.26, the update formulas for the velocity and displacement vectors are

$$\{\dot{\mathbf{X}}\}_{n+\frac{1}{2}} = \{\dot{\mathbf{X}}\}_{n-\frac{1}{2}} + \Delta t [M]^{-1} (\{\mathbf{F}^{\text{ext}}\}_n - \{\mathbf{F}^{\text{int}}\}_n) \quad (28.25)$$

$$\{\mathbf{X}\}_{n+1} = \{\mathbf{X}\}_n + \Delta t \{\dot{\mathbf{X}}\}_{n+\frac{1}{2}} \quad (28.26)$$

If a lumped (diagonal) mass matrix is used, then Equation 28.25 uncouples.

Initially (at $t = 0$), Equation 28.25 and Equation 28.26 are replaced by

$$\{\dot{\mathbf{X}}\}_{\frac{\Delta t}{2}} = \{\dot{\mathbf{X}}\}_0 + \frac{\Delta t}{2} [M]^{-1} (\{\mathbf{F}^{\text{ext}}\}_0 - \{\mathbf{F}^{\text{int}}\}_0) \quad (28.27)$$

and

$$\{\mathbf{X}\}_{\Delta t} = \{\mathbf{X}\}_0 + \frac{\Delta t}{2} \{\dot{\mathbf{X}}\}_{\frac{\Delta t}{2}} \quad (28.28)$$

where subscript 0 in Equation 28.27 and Equation 28.28 refers to the value of the vector at $t = 0$.

Note that the central difference scheme is only conditionally stable. Therefore, the time step Δt must be smaller than a critical value Δt_{cr} , which for linear problems is calculated from the smallest period of vibration T_{min} . Specifically,

$$\Delta t \leq \Delta t_{\text{cr}} = \frac{T_{\text{min}}}{\pi} = \frac{2}{\omega_{\text{max}}}$$

For nonlinear problems experience has shown that a 10–20% reduction in the time step is usually sufficient to maintain stability.

28.5.2.1.2 Newmark's Method

Newmark's method is based on the following approximations for the displacement and velocity vectors:

$$\{\mathbf{X}\}_{n+1} = \{\mathbf{X}\}_n + \Delta t \{\dot{\mathbf{X}}\}_n + (\Delta t)^2 \left[\left(\frac{1}{2} - \beta \right) \{\ddot{\mathbf{X}}\}_n + \beta \{\ddot{\mathbf{X}}\}_{n+1} \right] \quad (28.29)$$

$$\{\dot{\mathbf{X}}\}_{n+1} = \{\dot{\mathbf{X}}\}_n + \Delta t [(1 - \gamma) \{\ddot{\mathbf{X}}\}_n + \gamma \{\ddot{\mathbf{X}}\}_{n+1}] \quad (28.30)$$

where β and γ are free parameters of Newmark's method. The particular choice $\beta = 1/6$, $\gamma = 1/2$ corresponds to a linear approximation of the acceleration over the n th time step. The choice $\beta = 1/4$, $\gamma = 1/2$ corresponds to a constant average acceleration (trapezoidal rule), which is unconditionally stable for linear systems. The central difference scheme corresponds to $\beta = 0$, $\gamma = 1/2$.

If Equation 28.29 and Equation 28.30 are combined with the balance equations, Equation 28.16, a set of simultaneous algebraic equations result at each time step. For nonlinear problems, the equations are

nonlinear and are expressed by the following process for the n th time step:

1. Predict velocities and displacements using the explicit approximations

$$\{\tilde{\dot{\mathbf{X}}}\}_{n+1} = \{\dot{\mathbf{X}}\}_n + \Delta t(1 - \gamma)\{\ddot{\mathbf{X}}\}_n \quad (28.31)$$

$$\{\tilde{\mathbf{X}}\}_{n+1} = \{\mathbf{X}\}_n + \Delta t\{\dot{\mathbf{X}}\}_n + (\Delta t)^2\left(\frac{1}{2} - \beta\right)\{\ddot{\mathbf{X}}\}_n \quad (28.32)$$

where a tilde (\sim) refers to the predicted value of the vector.

2. Obtain corrections to the displacements, velocities, and accelerations by the following iterative process:

$$[K^*]\{\Delta\mathbf{X}\}_{n+1}^{(i)} = \{\mathbf{R}\}_{n+1}^{(i)} \quad (28.33)$$

$$\{\mathbf{X}\}_{n+1}^{(i+1)} = \{\mathbf{X}\}_{n+1}^{(i)} + \{\Delta\mathbf{X}\}_{n+1}^{(i)} \quad (28.34)$$

$$\{\dot{\mathbf{X}}\}_{n+1}^{(i+1)} = \{\dot{\mathbf{X}}\}_{n+1}^{(i)} + \gamma\Delta t\{\ddot{\mathbf{X}}\}_{n+1}^{(i+1)} \quad (28.35)$$

$$\{\ddot{\mathbf{X}}\}_{n+1}^{(i+1)} = \frac{1}{\beta(\Delta t)^2} \left[\{\mathbf{X}\}_{n+1}^{(i+1)} - \{\tilde{\mathbf{X}}\}_{n+1} \right] \quad (28.36)$$

For $i = 0$, the following values are used for the displacement, velocity, and acceleration vectors

$$\{\mathbf{X}\}_{n+1}^{(0)} = \{\tilde{\mathbf{X}}\}_{n+1}$$

$$\{\dot{\mathbf{X}}\}_{n+1}^{(0)} = \{\tilde{\dot{\mathbf{X}}}\}_{n+1}$$

$$\{\ddot{\mathbf{X}}\}_{n+1}^{(0)} = 0$$

where superscript (0) refers to the value of the vector associated with $i = 0$, $[K^*]$ and $\{\mathbf{R}\}$ in Equation 28.33 are the effective stiffness matrix and the residual vector given by

$$[K^*] = [K] + \frac{\gamma}{\beta\Delta t}[C] + \frac{1}{\beta(\Delta t)^2}[M] \quad (28.37)$$

$$\{\mathbf{R}\}_{n+1}^{(i)} = \{\mathbf{F}^{\text{ext}}\}_{n+1} - [M]\{\ddot{\mathbf{X}}\}_{n+1}^{(i)} - \{\mathbf{F}^{\text{int}}\}_{n+1}^{(i)} \quad (28.38)$$

Superscripts i and $i + 1$ refer to the values of the vectors at the beginning and end of the i th iteration cycle. As for nonlinear static problems, the choice of $[K^*]$ depends on the iterative procedure used.

28.5.2.1.3 Modal Superposition Method

In this method the structural response at any time is expressed as a linear combination of a number of preselected modes (or basis vectors). This is expressed by the following transformation:

$$\{\mathbf{X}\} = [\Gamma]\{\boldsymbol{\psi}\} \quad (28.39)$$

where $[\Gamma]$ is the transformation matrix whose columns are the modes (or basis vectors) and $\{\boldsymbol{\psi}\}$ is a vector of unknown coefficients (amplitudes of the preselected modes).

The number of modes (or basis vectors) is usually selected to be much smaller than the number of degrees of freedom of the discretized structure (components of the vector $\{\mathbf{X}\}$). A Bubnov–Galerkin technique is used to approximate the balance equations of the discretized structure, Equation 28.16, by a much smaller system of equations in $\{\boldsymbol{\psi}\}$. The resulting equations have the following form:

$$[\Gamma]^T[M][\Gamma]\{\ddot{\boldsymbol{\psi}}\} = [\Gamma]^T\{\mathbf{F}^{\text{ext}}\} - [\Gamma]^T\{\mathbf{F}^{\text{int}}(\boldsymbol{\psi})\} \quad (28.40)$$

For linear problems, if the basis vectors are selected to be the free vibration modes of the structure, Equation 28.40 uncouples in the components of $\{\Psi\}$. For nonlinear problems the free vibration modes and frequencies of the structure change with time.

The mode superposition technique is effective when the response can be adequately represented by few modes and the time integration has to be carried out over many time steps (e.g., earthquake loading), and the cost of calculating the required modes is reasonable.

28.5.3 Energy Balance in Transient Dynamic Analysis

Energy balance (conservation) in transient analysis is a reflection of the accuracy of the time integrator. Furthermore, conservation properties are intimately related to stability. The construction of stable integrators is often approached by enforcing conservation laws.

In linear problems, instability can be easily detected by the spurious growth in velocities. On the other hand, in large structural problems with material nonlinearities the energy generated by an instability may be rapidly dissipated as the material becomes inelastic and the erroneous nature of the results does not become obvious to the user. This kind of instability has been termed *arrested instability* in Belytschko and Hughes [1983] and the energy balance check described subsequently was suggested to detect it.

Let $\{\Delta \mathbf{X}\}_n$ be the increment of the displacement vector from time t_n to time t_{n+1} , and let the internal energy, and the increment of external work be approximated by the trapezoidal rule as follows:

$$U_j = \sum_{l=1}^{j-1} \frac{1}{2} \{\Delta \mathbf{X}\}_l^T \left(\{\mathbf{F}^{\text{int}}\}_l + \{\mathbf{F}^{\text{int}}\}_{l+1} \right) \quad (28.41)$$

$$\Delta W_j = \frac{1}{2} \{\Delta \mathbf{X}\}_j^T \left(\{\mathbf{F}^{\text{ext}}\}_j + \{\mathbf{F}^{\text{ext}}\}_{j+1} \right) \quad (28.42)$$

The kinetic energy T_j at time t_j is given by

$$T_j = \frac{1}{2} \{\dot{\mathbf{X}}\}_j^T [M] \{\dot{\mathbf{X}}\}_j \quad (28.43)$$

If the mass matrix is positive definite, then the energy and the displacements of the structure at time t_{n+1} are bounded if the following inequality is satisfied:

$$T_{n+1} + U_{n+1} \leq (1 + \epsilon)(T_n + U_n) + \Delta W_n \quad (28.44)$$

where ϵ is a small number. Equation 28.44 provides an energy balance check provided $U_n \geq 0$.

Detailed discussion of the stability, convergence, and decay of energy is given in Hughes [1976]. Examples of the construction of integrators through energy conservation are given in Hughes et al. [1978] and Haug et al. [1977].

28.5.4 Stability Analysis

A number of instability phenomena can occur in structures depending on the structural characteristics and operational environment. These instabilities can be either time dependent or time invariant. Time-dependent instability analysis can be performed by using Lyapunov's method (see Kratzig and Eller [1992] and Kounadis and Kratzig [1995]). Time-invariant instabilities can be in the form of bifurcation points (simple or multiple) or limit points in the solution path. The *critical loads* associated with these points are referred to as bifurcation buckling and limit loads, respectively.

The algorithmic tools for *time-invariant instability analysis* encompass three distinct aspects: (1) non-linear (or linear) analysis of the prebuckling state, using the nonlinear equations — Equation 28.1 [or the linearized version, Equation 28.2]; (2) determination of the critical points (e.g., bifurcation and/or limit points) on the equilibrium path; and (3) tracing the postcritical response. Critical points can be detected by the singularity of the tangent stiffness matrix, Equation 28.19. Bifurcation loads associated with linear

prebuckling state can be determined by solving a linear eigenvalue problem as described in the succeeding subsection. Special numerical algorithms are available for overcoming the difficulties associated with commonly used iterative techniques near critical points (see, for example, Riks [1984] and Crisfield [1983]).

28.5.5 Eigenvalue Problems

Undamped free vibration and linear (bifurcation) buckling problems of structures can be represented by the general linear matrix eigenvalue equation

$$[K]\{\mathbf{X}\} = \lambda[B]\{\mathbf{X}\} \quad (28.45)$$

where $[K]$ is the stiffness matrix of the discretized structure, $\{\mathbf{X}\}$ is the displacement vector, $[B]$ is the mass matrix for free vibration problems or the negative of the geometric stiffness matrix for buckling problems, and λ is the eigenvalue square of vibration frequency or buckling load parameter. Typically, most of the elements of $[K]$ and $[B]$ are zero and only a few pairs $(\lambda, \{\mathbf{X}\})$ are wanted.

Although the governing equations for both the free vibration and linear buckling problems are similar, the properties of the two matrices $[K]$ and $[B]$ are different. For vibration problems, the stiffness matrix can be positive definite, positive semidefinite, or indefinite. For an unsupported structure the stiffness matrix is indefinite. If the structure is stable (except for rigid motions), the stiffness matrix will be positive semidefinite. Also, some lumping procedures can produce mass matrices, which are positive semidefinite because they have zero mass elements on the diagonal. The mass matrix can be positive definite or semidefinite (if some of the diagonal mass elements are zero, due to the lumping procedure used).

For buckling problems, the stiffness matrix is positive definite, provided the deformation state is stable (which may be assessed by a buckling analysis). The geometric stiffness matrix may be indefinite.

The preferred eigenvalue extraction techniques in use to date are sampling techniques. They create a linear operator (or matrix) and apply it to a sequence of carefully constructed vectors. From these transformed vectors the dominant eigenvectors and their eigenvalues can be approximated. Examples of these techniques are subspace iteration (or simultaneous iteration) and Lanczos techniques. The details of these methods are described in Bathe [1996], Parlett [1987], and Hughes [1987].

The eigensolver algorithms typically generate a set of eigenpairs (eigenvalues and associated eigenvectors) for the lowest eigenstates of the system, i.e., for the smallest eigenvalues (in absolute order). Often, it is necessary to compute eigenpairs for cases other than the set of smallest eigenvalues. This may be performed by introducing a shift α , which defines the shifted eigenvalue

$$\lambda = \bar{\lambda} + \alpha$$

The shifted eigenvalue problem is

$$[[K] - \alpha[B]]\{\mathbf{X}\} = \bar{\lambda}[B]\{\mathbf{X}\} \quad (28.46)$$

Equation 28.46, with a nonzero α , can be used to compute the eigenvalues for an unsupported structure (one for which $[K]$ is singular).

28.5.6 Sensitivity Analysis

Sensitivity analysis refers to methods of calculating the rates of change of: (1) response quantities (e.g., displacements, stresses, vibration frequencies, and buckling loads) with respect to changes in the structure characteristics (e.g., geometric and material parameters of the structure) and (2) the optimum design variable values with respect to changes in the structure parameters (e.g., applied loads and allowable stresses). The two types of calculations are usually designated by response and optimum design sensitivity analysis, and the rates of change are referred to as sensitivity coefficients.

A number of techniques have been developed for evaluating the sensitivity coefficients of response quantities using either the governing discrete equations or continuum equations of the structure. The

techniques used with the discrete equations can be grouped into three categories: *analytical*, *direct differentiation methods*; *finite difference methods*; and *semianalytical or quasianalytical methods* (see Kleiber and Hisada [1993] and Hinton and Sienz [1994]).

Methods for computing sensitivity coefficients for linear structural response have been developed for over 20 years (see, for example, Haftka and Adelman [1989], Haber et al. [1993], and Choi [1993]). However, only recently have attempts been made to extend the domain of sensitivity analysis to (1) nonlinear structural response and to path-dependent problems, for which the sensitivity coefficients depend also on the deformation history (e.g., viscoplastic response and frictional contact — see, for example, Kleiber et al. [1994], Kulkarni and Noor [1995], and Karaoglan and Noor [1995]); and (2) structural systems exhibiting probabilistic uncertainties.

Because of the importance of its role in structural optimization and in assessing the effect of uncertainties in the input parameters on the structural response, some commercial software systems have incorporated response sensitivity analysis into their systems. Also, an automatic differentiation facility has been developed for evaluating the derivatives of functions defined by computer programs exactly to within machine precision. The facility, automatic differentiation of Fortran (ADIFOR), is described in Chinchalkar [1994] and Carle et al. [1994]. The use of ADIFOR to evaluate the sensitivity coefficients from incremental/iterative forms of three-dimensional fluid flow problems is discussed in Sherman et al. [1994], and the additional facilities needed for ADIFOR to become competitive with hand-differentiated codes are listed in Carle et al. [1994].

28.5.7 Strategies and Numerical Algorithms for New Computing Systems

In recent years, intense efforts have been devoted to the development of efficient computational strategies and numerical algorithms which exploit the capabilities of new computing systems (in particular, the vector and parallel processing of the powerful high-performance computers — see, for example, Onaté et al. [1992]). Efficient direct and iterative numerical algorithms have been developed for solution of large sparse linear systems of equations (see Wang and Bruch [1993], Law and Mackay [1993], Qin and Mackay [1993], Vaughan [1991], and Papadrakakis [1993]).

Most parallel strategies are related to the *divide-and-conquer* paradigm based on breaking a large problem into a number of smaller subproblems which may be solved separately on individual processors. The degree of independence of the subproblems is a measure of the effectiveness of the algorithm since it determines the amount and frequency of communication and synchronization. The numerical algorithms developed for structural analysis can be classified into three major categories: namely, *elementwise algorithms*, *nodewise algorithms*, and *domainwise algorithms*.

The elementwise parallel algorithms include element-by-element equation solvers and parallel frontal equation solvers. The nodewise parallel equation solvers include node-by-node iterative solvers as well as column-oriented direct solvers. The domainwise algorithms include nested dissection-based (substructuring) techniques and domain-decomposition methods. The first two categories of numerical algorithms allow only small granularity of the parallel tasks and require frequent communication among the processors. By contrast, the third category allows a larger granularity, which can result in improved performance for the algorithm.

Nested dissection ordering schemes have been found to be effective in reducing both the storage requirements and the total computational effort required of direct factorization [Noor et al. 1978]. The performance of nested dissection-based linear solvers depends on balancing the computational load across processors in a way that minimizes interprocessor communication. Several nested dissection ordering schemes have been developed which differ in the strategies used in partitioning the structure and selecting the separators. Among the proposed partitioning strategies are: recursive bisection strategies (e.g., spectral graph bisection, recursive coordinate bisection, and recursive graph bisection [Pothen et al. 1990, Hendrickson and Leland 1992]); combinatorial and design-optimization based strategies (e.g., simulated annealing algorithm, genetic algorithm and neural-network-based techniques [Khan and Topping 1993]); and heuristic strategies (e.g., methods based on geometric projections and mappings; and algorithms based

on embedding the problem in Euclidean space [Bui and Jones 1993]). For highly irregular and/or three-dimensional structures the effectiveness of nested dissection-based schemes may be reduced. However, this is also true for most other parallel numerical algorithms. Scalable parallel computational strategies for nonlinear, postbuckling, and dynamic contact/impact problems are presented in Watson and Noor [1996a, 1996b].

28.6 Brief History of the Development of Computational Structural Mechanics Software

Development of CSM software spans a period of less than 40 years, and may be divided into four stages, each stage lasting approximately 8–10 years. In the first stage (during the 1950s and 1960s), the aircraft industry pioneered development of in-house finite-element programs. These programs were generally based on the force method of analysis and were used to automate analysis of highly redundant structural components. Subsequent efforts in industry and academia led to development of simple two- and three-dimensional finite elements based on the displacement formulation. The variational process for formulating the elemental matrices was also introduced in this period.

In the second stage, general-purpose finite-element programs, such as NASTRAN, ASKA, ANSYS, STARDYNE, MARC, SAP, SESAM, and SAMCEF, were released for public use in the U.S. and Europe. These programs brought a significant technology base that led to development of numerous commercial finite-element software systems. This development included mixed and hybrid finite-element models with the fundamental unknowns consisting of stress and displacement parameters, efficient numerical algorithms for the solution of algebraic equations and extraction of eigenvalues, and substructuring and modal synthesis techniques for handling large problems. The finite-element method's success in linear static problems has encouraged bolder applications to nonlinear and transient response problems.

The third stage involved refining the commercial software codes and expanding their technology base. Design optimization techniques were also developed in this stage, as were pre- and postprocessing software and computer-aided design systems. The technology development included singular elements for fracture mechanics applications, boundary-element techniques, coupling of finite elements with other discretization techniques such as boundary elements, and quality assurance methods for both software and finite-element models.

The fourth stage included in the adaptation of CSM software to new computing systems (vector, multiprocessor, and massively parallel machines), development of efficient computational strategies and numerical algorithms for these machines, widespread availability of CSM software on personal computers and workstations, and the addition of substantial nonstructural modeling and analysis capabilities to CSM software systems such as MSC/NASTRAN and ANSYS. The latter capabilities were added because future flight vehicles and high-performance engineering systems (e.g., health monitoring aircraft and micro-sized spacecraft) will require significant interactions between CSM and other disciplines such as aerodynamics, controls, acoustics, electromagnetics, and optics.

Technology development in the fourth stage included introduction of advanced material models, development of stochastic models and probabilistic methods for structural analysis and design, and development of facilities for quality assessment and control of numerical simulations.

The four stages of CSM software development parallel the four stages of the computing environment's evolution: noncentralized mainframes, centralized mainframes, mainframe computing with timesharing, and distributed computing and networking. A summary of the major characteristics of currently used finite-element systems is given in Mackerle and Fredriksson [1988], and a guide to information sources on finite-element methods is given in Mackerle [1991]. Commercial finite-element programs for structural analysis have a rich variety of elements, and are widely used for performing structural calculations on large components and/or entire structures (see, for example, [Figure 28.7](#) to [Figure 28.9](#)).

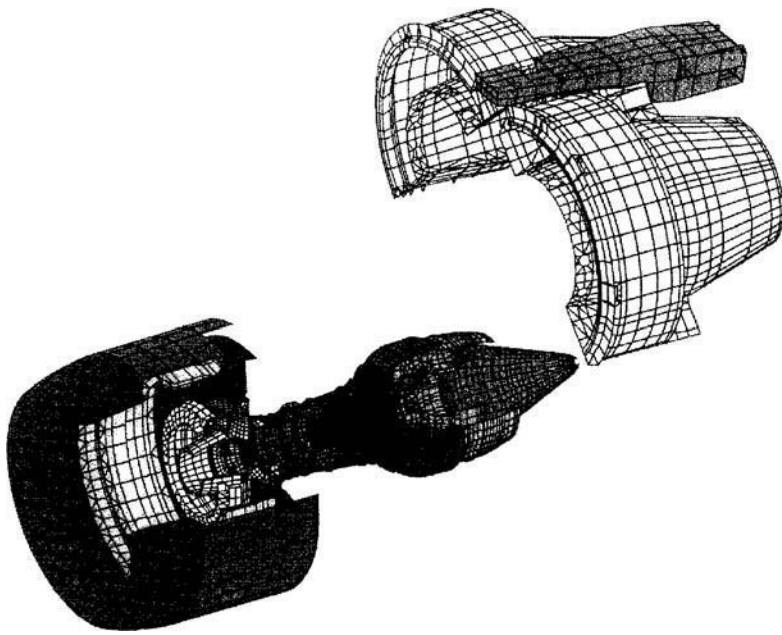


FIGURE 28.7 MSC/NASTRAN finite-element model of a G.E. Engine: 180,000 degrees of freedom. (Courtesy of GE Aircraft Engines.)

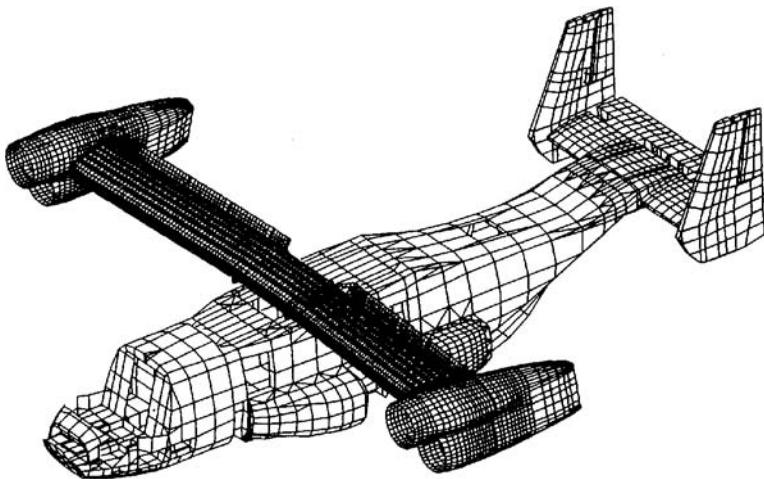


FIGURE 28.8 MSC/NASTRAN finite-element dynamics model of the V-22 Osprey Tiltrotor: 134,982 degrees of freedom (22,497 grid points), 44,006 elements. (Courtesy of Bell-Boeing.)

28.7 Characteristics of Future Engineering Systems and Their Implications on Computational Structural Mechanics

The demands that future high-performance engineering systems place on CSM differ somewhat from those of current systems. The radically different and more unpredictable operational environments for many of the systems (e.g., future flight vehicles) are one reason for this difference. Another is the stringency

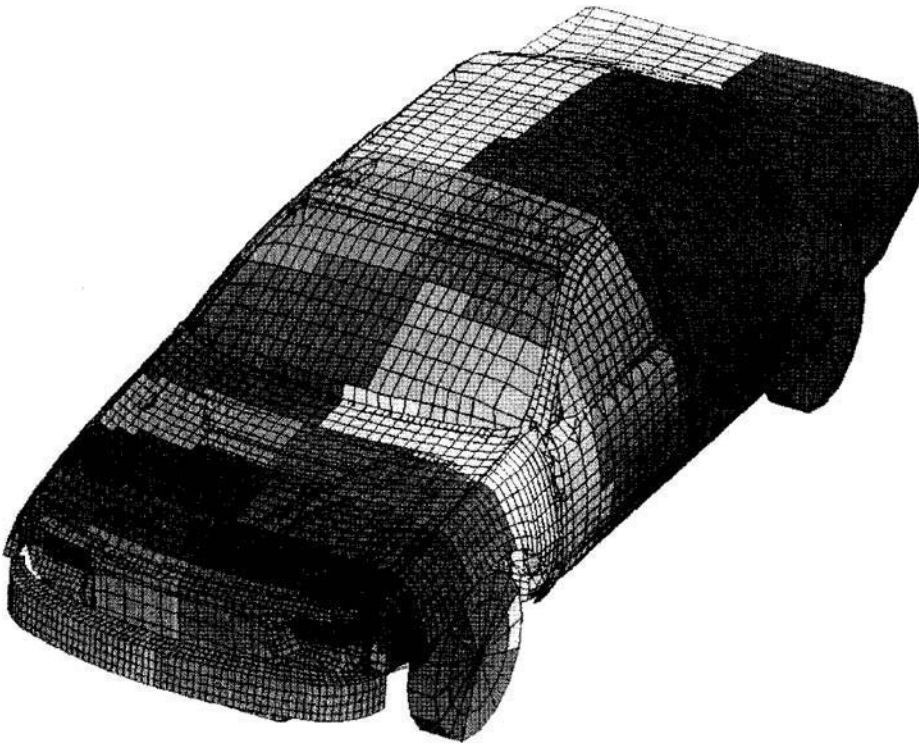


FIGURE 28.9 DYNA 3-D finite element car model used in crash simulation: 27,000 shell elements, 162,000 degrees of freedom. (Courtesy of Lawrence Livermore.)

of design requirements for high performance, light weight, and economy. The technical needs for future high-performance engineering systems include:

1. Development of new high-performance material systems, such as smart/intelligent material systems
2. Development of novel structural concepts, such as structural tailoring and smart/intelligent structures, with active and/or passive adaptive control of dynamic deformations
3. Investigation of more complex phenomena and interdisciplinary couplings such as fluid flow/acoustics/thermal/control/electromagnetic/optics, and structural couplings.

28.8 Primary Pacing Items and Research Issues

The primary pacing items for CSM are:

1. High-fidelity modeling of the structure and its components
2. Failure and life prediction methodologies
3. Hierarchical, integrated methods and adaptive modeling techniques
4. Nondeterministic analysis, modeling methods, and risk assessment
5. Validation of numerical simulations
6. Multidisciplinary analysis and design optimization

For each of the aforementioned items, attempts should be made to exploit the major characteristics of high-performance computing technologies, as well as the future computing environment. The six primary pacing items are described subsequently. Note that some of the tasks within the pacing items are of a generic nature. Others are specific to certain components of future engineering systems (e.g., propulsion systems or airframes).

28.8.1 High-Fidelity Modeling of the Structure

The reliability of the predictions of response, failure, and life of structures is critically dependent on: (1) the accurate characterization and modeling of material behavior and (2) high-fidelity modeling of the critical details of the structure and its components (e.g., joints, damping, and for large deformations, frictional contact between the different parts of the structure). The simple material models used to date are inadequate for many of the future applications, especially those involving severe environment (e.g., high temperatures). Needed work on material modeling can be grouped in two general areas: (1) modeling the response and damage of advanced material systems in the actual operating environment of future engineering systems and (2) numerical simulation of manufacturing (fabrication) processes.

Advanced material systems include new polymer composites, metal-matrix composites, ceramic composites, carbon/carbon, and advanced metallics. The length scale selected in the model must be adequate for capturing the response phenomena of interest (e.g., micromechanics, mesomechanics, and macromechanics). For materials used in high-temperature applications, work is needed on the modeling of damage accumulation and propagation to fracture; modeling of thermoviscoplastic response, thermal-mechanical cycling, and ratcheting; and prediction of long-term material behavior from short-term data, which are particularly important.

28.8.2 Failure and Life Prediction Methodologies

Practical numerical techniques are needed for predicting the life, as well as the failure initiation and propagation in structural components made of new, high-performance materials in terms of measurable and controllable parameters. Examples of these materials are high-temperature materials; piezoelectric composites; and electronic, optical, and smart materials. For some of the materials, accurate constitutive descriptions, failure criteria, damage theories, and fatigue data are needed, along with more realistic characterization of interface phenomena (such as contact and friction). The constitutive descriptions may require investigations at the microstructure level or even the atomic level, as well as carefully designed and conducted experiments. Failure and life prediction of structures made of these materials is difficult and numerical models often constructed under restricting assumptions may not capture the dominant and underlying physical failure mechanisms. Moreover, material failure and structural response (such as instability) often couple in the failure mechanism.

28.8.3 Hierarchical, Integrated Multiple Methods and Adaptive Modeling Techniques

The effective use of numerical simulations for predicting the response, life, performance, and failure of future engineering systems requires strategies for treating phenomena occurring at disparate spatial and time scales, using reasonable computer resources. The strategies are based on using multiple mathematical models in different regions of the structure to take advantage of efficiencies gained by matching the model to the expected response in each region. To achieve the full potential of hierarchical modeling, there should be minimal reliance on a priori assumptions about the response. This is accomplished by adding adaptivity to the strategy. The key tasks of the research in this area are: (1) simple design-oriented models for use in the early stages of the design process; (2) rational selection of a set of nested mathematical models for different regions, and discretization techniques for use in conjunction with the mathematical models, which in turn requires the availability of a capability for holistic modeling from micro to structural response with varying degrees of accuracy; (3) simulation of local phenomena through global/local methodologies; (4) automated (or semiautomated) coupling of different mathematical/discrete models; (5) error estimation and adaptive modeling strategies; (6) efficient methods for coupling different components (e.g., engine airframe and rotor/engine frame); and (7) sensitivity analysis to assess the sensitivity of the response to each of the parameters neglected in the current mathematical model.

28.8.4 Nondeterministic Analysis, Modeling, and Risk Assessment

The new methodology developed for treating different types of uncertainties in geometry, material properties, boundary conditions, loading, and operational environment in the structural analysis formulation of structural components needs to be extended to the design and risk assessment of engineering systems. The ability to quantify inherent uncertainties in the response of the structure is obviously of great advantage. However, the principal benefit of using any nondeterministic method consists of the insights into engineering, safety, and economics that are gained in the process of arriving at those quantitative results and carrying out reliability analyses. As future engineering structures become more complicated, modeling of failure mechanisms will account for uncertainties from the beginning of the design process, and potential design improvements will be evaluated to assess their effects on reducing overall risk. The results combined with economic considerations will be used in systematic cost-benefit analyses (perhaps also done on a non-deterministic basis) to determine the structural design with the most acceptable balance of cost and risk.

28.8.5 Validation of Numerical Simulations

In addition to selecting a benchmark set of structures for assessing new computational strategies and numerical algorithms, a high degree of interaction and communication is needed between computational modelers and experimentalists. This is done on four different levels, namely:

1. Laboratory tests on small specimens to obtain material data
2. Component tests to validate computational models
3. Full-scale tests to validate the modeling of details, and for flight vehicles
4. Flight tests to validate the entire modeling process

28.8.6 Multidisciplinary Analysis and Design Optimization

The realization of new complex engineering systems requires integration between the structures discipline and other traditionally separate disciplines such as aerodynamics, propulsion, and control. This is mandated by significant interdisciplinary interactions and couplings which need to be accounted for in predicting response, as well as in the optimal design of these structures. Examples are the couplings between the aerodynamic flow field, structural heat transfer, and structural response of high-speed aircraft and propulsion systems and the couplings between the control system and structural response in control-configured aircraft and spacecraft. This activity also includes design optimization with multiobjective functions (e.g., performance, durability, integrity, reliability, and cost), and multiscale structural tailoring (micro, local, and global levels). For propulsion systems, it also includes design with damping for high-cycle fatigue, low-cycle fatigue, and acoustic fatigue.

Typically, in the design process questions arise regarding influence of design variable changes on system behavior. Answers to these questions, quantified by the derivatives of behavior with respect to the design variables or by parametric studies, guide design improvements toward a better overall system. In large applications, this improvement process is executed by numerical optimization, combined with symbolic/artificial intelligence (AI) techniques, and human judgment aided by data visualization. Efficiency of the computations that provide data for such a process is decisive for the depth, breadth, and rate of progress achievable, and hence, ultimately, is critical for the final product quality.

28.8.7 Related Tasks

For CSM to impact the design process, the following tasks need to be addressed by the research community: (1) development of automated or semiautomated model (and mesh) generation facilities; (2) pre- and postdata processing and use of advanced visualization technology, including multimedia and virtual reality facilities; and (3) adaptation of object-oriented technology and AI tools (knowledge-based expert systems and neural networks) to CSM.

CSM's wide acceptance can affect the design and operation of future engineering systems and structures in three ways. It can provide a better understanding of the phenomena associated with response, failure, and life, thereby identifying desirable structural design attributes. CSM can verify and certify designs and also allow low-cost modifications to be made during the design process. Finally, it can improve the design team's productivity and allow major improvements and innovations during the design phase, enabling fully integrated design in an integrated product and process development (IPPD) environment. Such an environment allows computer simulation of the entire life cycle of the engineering system, including material selection and processing, multidisciplinary design, automated manufacturing and fabrication, quality assurance, certification, operation, health monitoring and control, retirement, and disposal.

Acknowledgment

The present work is partially supported by NASA Cooperative Agreement NCCW-0011 and by Air Force Office of Scientific Research Grant AFOSR-F49620-93-1-0184.

References

- Babuska, I. and Miller, A. 1984a. The postprocessing approach in the finite element method — part 1: calculation of displacements, stresses and other higher derivatives of the displacements. *Int. J. Num. Methods. Eng.* 20:1085–1109.
- Babuska, I. and Miller, A. 1984b. The postprocessing approach in the finite element method — part 2: the calculation of stress intensity factors. *Int. J. Num. Methods. Eng.* 20:1110–1129.
- Banerjee, P. K. 1994. *The Boundary Element Methods in Engineering*, 2nd ed. McGraw-Hill, New York.
- Bathe, K. J. 1996. *Finite Element Procedures*. Prentice-Hall, Englewood Cliffs, NJ.
- Belytschko, T., Engelmann, B. E., and Liu, W. K. 1987. A review of recent developments in time integration. In *State-of-the-Art Surveys on Computational Mechanics*, A. K. Noor and J. T. Oden, Eds., pp. 185–199. American Society of Mechanical Engineers, New York.
- Belytschko, T. and Hughes, T. J. R. 1983. *Computational Methods for Transient Analysis*. Elsevier Science, Amsterdam.
- Blacker, T. D. and Stephenson, M. B. 1991. Pacing: a new approach to automated quadrilateral mesh generation. *Int. J. Num. Methods. Eng.* 32:811–847.
- Bui, T. N. and Jones, C. 1993. A heuristic for reducing fill-in in sparse matrix factorization. *Proc. 6th SIAM Conf. Parallel Process. Sci. Comput.*
- Carle, A., Green, L. L., Bischof, C. H., and Newman, P. A. 1994. Application of automatic differentiation in CFD. *Proc. 25th AIAA Fluid Dynamics Conf.* Colorado Springs, CO, June 20–23, AIAA Paper 94-2197.
- Cescotto, S., Frey, F., and Fonder, G. 1979. Total and updated Lagrangian descriptions in nonlinear structural analysis: a unified approach. In *Energy Methods in Finite Element Analysis*, R. Glowinski, E. Y. Rodin, and O. C. Zienkiewicz, Eds., pp. 283–296. Wiley, New York.
- Chinchalkar, S. 1994. The application of automatic differentiation to problems in engineering analysis. *Comp. Methods. Appl. Mech. Eng.* 118:197–207.
- Choi, K. K. 1993. Design sensitivity of nonlinear structures — II. In *Structural Optimization: Status and Promise*, M. P. Kamat, Ed., pp. 407–446. American Institute of Aeronautics and Astronautics, Washington, DC.
- Crisfield, M. A. 1983. An arc-length method including line searches and accelerations. *Int. J. Num. Methods Eng.* 19:1269–1289.
- Demkowicz, L., Oden, J. T., and Babuska, I. 1992. *Reliability in Computational Mechanics*. *Comput. Methods Appl. Mech. Eng.* Spec. issue 101(1–3).
- Elishakoff, I. 1995. Essay on uncertainties in elastic and viscoelastic structures: from A. M. Freudenthal's criticisms to modern convex modeling. *Comput. Struct.* 56(6):871–895.
- Fish, J. and Markolefas, S. 1993. Adaptive s-method for linear elastostatics. *Comput. Methods. Appl. Mech. Eng.* 104:363–396.

- Fish, J. and Markolefas, S. 1994. Adaptive global-local refinement strategy based on the interior error estimates of the h-method. *Int. J. Num. Methods Eng.* 37:827–838.
- George, P. L. 1991. *Automatic Mesh Generation. Application to Finite Element Methods*. Wiley, New York.
- Geradin, M. 1980. Variational methods of structural dynamics and their finite element implementation. In *Advanced Structural Dynamics*, J. Donea, Ed., pp. 1–41. Applied Science, Ltd., London.
- Grandhi, R. V., Stroud, W. J., and Venkayya, V. B., eds. 1989. *Computational Structural Mechanics and Multidisciplinary Optimization*, AD Vol. 16, American Society of Mechanical Engineers, New York.
- Haber, R. B., Tortorelli, D. A., and Vidal, C. A. 1993. Design sensitivity analysis of nonlinear structures — I: large deformation hyperelasticity and history dependent material response. In *Structural Optimization: Status and Promise*, M. P. Kamat, Ed., pp. 369–406. American Institute of Aeronautics and Astronautics, Washington, DC.
- Haftka, R. T. and Adelman, H. M. 1989. Recent developments in structural sensitivity analysis. *Struct. Optimization* 1(3):137–151.
- Haug, E., Nguyen, Q. S., and de Rouvray, A. L. 1977. An improved energy conserving implicit time integration algorithm for nonlinear dynamic structural analysis. In *Trans. 4th Conf. Struct. Mech. Reactor Tech.*, T. A. Jaeger and B. A. Boley, Eds. IASMiRT, Paper No. M-5/3.
- Hendrickson, B. and Leland, R. 1992. An Improved Spectral Graph Partitioning Algorithm for Mapping Parallel Computations. *Sandia National Lab.* TR SAND92-1460.
- Hinton, E. and Sienz, J. 1994. Aspects of adaptive finite-element analysis and structural optimization. In *Advances in Structural Optimization*, B. H. V. Topping and M. Papadrakakis, Eds., pp. 1–25. Civil-Comp, Ltd., Edinburgh, Scotland.
- Ho-Le, K. 1988. Finite element mesh generation methods: review and classification. *Comput. Aided Design* 1(20):27–38.
- Hughes, T. J. R. 1976. Stability, convergence and growth and decay of energy of the average acceleration method in nonlinear structural dynamics. *Comput. Struct.* 6:313–324.
- Hughes, T. J. R. 1987. *The Finite Element Method*. Prentice-Hall, Englewood Cliffs, NJ.
- Hughes, T. J. R., Caughey, T. K., and Liu, W. K. 1978. Finite element methods for nonlinear elastodynamics which conserve energy. *J. Appl. Mech.* 45:366–370.
- Ju, J. W., ed. 1995. *Numerical Methods in Structural Mechanics*, ASME Press, New York.
- Kane, J. H., Maier, G., Tosaka, N., and Atluri, S. N., Eds. 1993. *Advances in Boundary Element Techniques*. Springer-Verlag, New York.
- Karaoglan, L. and Noor, A. K. 1995. Dynamic sensitivity analysis of frictional contact/impact response of axisymmetric composite structures. *Comp. Methods Appl. Mech. Eng.* 128(1–2):169–190.
- Khan, A. I. and Topping, B. H. V. 1993. Subdomain generation for parallel finite element analysis. *Comput. Syst. Eng.* 4(4–6):473–488.
- Kleiber, M., Hien, T. D., and Postek, E. 1994. Incremental finite-element sensitivity analysis for nonlinear mechanics applications. *Int. J. Num. Methods. Eng.* 37(19):3291–3308.
- Kleiber, M. and Hisada, T., eds. 1993. *Design-Sensitivity Analysis*. Atlanta Technology, Atlanta, GA.
- Kounadis, A. N. and Kratzig, W. B. 1995. *Nonlinear Stability of Structures: Theory and Computational Techniques*. Springer-Verlag, Vienna.
- Kratzig, W. B. and Eller, C. 1992. Numerical algorithms for unstable dynamic shell responses. *Comput. Struct.* 44(1–2):263–271.
- Krizek, M. and Neittaanmaki, P. 1987. On superconvergence techniques. *Acta Applicandae Mathematicae* 9:175–198.
- Kulkarni, M. and Noor, A. K. 1995. Sensitivity analysis of the nonlinear dynamic viscoplastic response of two-dimensional structures with respect to material parameters. *Int. J. Num. Methods. Eng.* 38(2):183–198.
- Ladevèze, P. and Zienkiewicz, O. C., eds. 1992. *New Advances in Computational Structural Mechanics (Proc. Eur. Conf. New Advances Comput. Struc. Mechanics)*. Giens, France, April 2–5, 1991. Elsevier, Amsterdam.

- Law, K. H. and Mackay, D. R. 1993. A parallel row-oriented sparse solution method for finite element structural analysis. *Int. J. Num. Methods. Eng.* 36:2895–2919.
- Mackerle, J. 1991. *Finite Element Methods, A Guide to Information Sources*. Elsevier, Amsterdam.
- Mackerle, J. 1993. Mesh generation and refinement for FEM and BEM — A bibliography. *Finite Elements Anal. Design* 15:177–188.
- Mackerle, J. and Fredriksson, B. 1988. *Handbook of Finite Element Software*. Studentlitteratur, Lund, Sweden.
- Noor, A. K. 1980. Mixed methods of analysis. In *Structural Mechanics Software Series*, N. Perrone and W. Pilkey, Eds., Vol. III, pp. 263–305. University Press of Virginia, Charlottesville.
- Noor, A. K. 1991. Bibliography of books and monographs on finite element technology. *Appl. Mech. Rev.*, 44(6):307–317.
- Noor, A. K., ed. 1992. *Adaptive, Multilevel and Hierarchical Computational Strategies*. Proc. Symp. ASME Winter Annu. Meeting, Nov. 8–13, Anaheim, CA. AMD Vol. 157, American Society of Mechanical Engineers, New York.
- Noor, A. K. 1994. Recent advances and applications of reduction methods. *Appl. Mech. Rev.* 47(5): 125–146.
- Noor, A. K. and Andersen, C. M. 1992. Hybrid analytical techniques for the nonlinear analysis of curved beams. *Comput. Struct.* 43(5):823–830.
- Noor, A. K. and Atluri, S. N. 1987. Advances and trends in computational structural mechanics. *AIAA J.* 25(7):977–995.
- Noor, A. K. and Babuska, I. 1987. Quality assessment and control of finite element solutions. *Finite Elements Anal. Design* 3:1–26.
- Noor, A. K., Housner, J. M., Starnes, J. H., and Hopkins, D. A. compilers. 1992. *Computational Structures Technology for Airframes and Propulsion Systems*. NASA CP-3142.
- Noor, A. K., Kamel, H. A., and Fulton, R. E. 1978. Substructuring techniques — status and projections. *Comput. Struct.* 8:621–632.
- Noor, A. K. and Venneri, S. L. 1990. Advances and trends in computational structural technology. *Comput. Syst. Eng.* 1(1):23–36.
- Noor, A. K. and Venneri, S. L., Eds. 1995. Computational structures technology. *Flight-Vehicle Materials, Structures and Dynamics*, Vol. 6. American Society of Mechanical Engineers, New York.
- Onaté, E., Periaux, J., and Samuelsson, A., eds. 1992. *The Finite Element Method in the Nineteen Ninety's: A Book Dedicated to O. C. Zienkiewicz*. Springer-Verlag, New York.
- Papadrakakis, M., ed. 1993. *Solving Large-Scale Problems in Mechanics: The Development and Application of Computational Solution Methods*. Wiley, Chichester, UK.
- Parlett, B. N. 1987. The state-of-the-art in extracting eigenvalues and eigenvectors in structural mechanics. In *State-of-the-Art Surveys on Computational Mechanics*. A. K. Noor and J. T. Oden, Eds., pp. 201–218. American Society of Mechanical Engineers, New York.
- Pothen, A., Simon, H. D., and Liou, K. 1990. Partitioning sparse matrices with eigenvectors of graphs. *SIAM J. Matrix Anal. App.* 11(3):430–452.
- Qin, J. and Mackay, D. R. 1993. A new parallel-vector finite element analysis software on distributed memory computers. *Proc. 34th AIAA/ASME/ASCE/AHS/ASC Struct. Structural Dynamics Mater. Conf.* La Jolla, CA, April 15–22.
- Riks, E. 1984. Bifurcation and stability. A numerical approach. In *Proc. Int. Conf. Innovative Methods Nonlinear Probl.* W. K. Liu, T. Belytschko, and K. C. Park, Eds., pp. 313–344. Pineridge Press, Swansea, UK.
- Ross, T. 1995. *Fuzzy Logic with Engineering Applications*. McGraw-Hill, New York.
- Sabin, M. 1991. Criteria for comparison of automatic mesh generation methods. *Advances Eng. Software & Workstations* 13(5–6):220–225.
- Shephard, M. S. 1988. Approaches to the automatic generation and control of finite element meshes. *Appl. Mech. Rev.* 41:169–185.
- Shephard, M. S. and Wentorf, R. 1994. Toward the implementation of automated analysis idealization control. *Appl. Num. Math.* 14(1–3).

- Sherman, L. L., Taylor, A. C., III, Green, L. L., Newman, P. A., Hou, G. J.-W., and Korivi, V. M. 1994. First- and second-order aerodynamic sensitivity derivatives via automatic differentiation with incremental iterative methods. *5th AIAA/USAF/NASA/ISSMO Symp. Multidisciplinary Anal. Optimization*, AIAA Paper 94-4262. Panama City Beach, FL. Sept. 7–9.
- Stein, E., ed. 1993. *Progress in Computational Analysis of Inelastic Structures*. Springer-Verlag, Vienna.
- Storaasli, O. O. and Housner, J. M., Eds. 1994. *Large-Scale Structural Analysis for High-Performance Computers and Workstations*, *Comput. Syst. Eng. Spec. issue* 5(4–6).
- Szabo, B. and Babuska, I. 1991. *Finite Element Analysis*. Wiley, New York.
- Tabbara, M., Blacker, T., and Belytschko, T. 1994. Finite element derivative recovery by moving least square interpolants. *Comput. Meth. Appl. Mech. Eng.* 117(1–2):211–223.
- Turkiyyah, G. M. and Fenves, S. J. 1996. Knowledge-based assistance for finite element modeling. *IEEE Expert Intell. Syst. Their Appl.* (June)11(3):23–32.
- Vaughan, C. T. 1991. Structural analysis on massively parallel computers. *Comput. Systems Eng.* 2(2/3):261–267.
- Wang, K. P. and Bruch, J. C. 1993. A highly efficient iterative parallel computational method for finite element systems. *Eng. Comput.* 10:195–204.
- Washizu, K. 1982. *Variational Methods in Elasticity and Plasticity*, 3rd ed. Pergamon Press, Oxford, UK.
- Watson, B. C. and Noor, A. K. 1996a. Large-scale contact/impact simulation and sensitivity analysis on distributed-memory computers. *Comp. Methods Appl. Mech. Eng.* 65:6.
- Watson, B. C. and Noor, A. K. 1996b. Sensitivity analysis for large-deflection and postbuckling responses on distributed-memory computers. *Comp. Methods. Appl. Mech. Eng.* 129:393–409.
- Zienkiewicz, O. C. and Zhu, J. Z. 1992. The superconvergent patch recovery and *a posteriori* error estimates — part 1: the recovery technique. *Int. J. Num. Methods. Eng.* 33:1331–1364.
- Zienkiewicz, O. C., Zhu, J. Z., and Wu, J. 1993. Superconvergent patch recovery techniques — some further tests. *Commun. Num. Methods Eng.* 9(3):251–258.

Further Information

Information about CSM software is available on the Internet including structural analysis and design programs, and commercial finite-element programs. A number of publications on finite-element practice is available from the National Agency for Finite Element Methods and Standards (NAFEMS), Department of Trade and Industry, National Engineering Laboratory, East Kilbride, Glasgow G75 0QU, United Kingdom, including *A Finite Element Primer* (1986) and *Guidelines to Finite Element Practice* (1984). A finite element bibliography, including books and conference proceedings published since 1967, is available on the World Wide Web. The WWW address is: <http://ohio.ikp.liu.se/fe/index.html>.

29

Computational Electromagnetics

29.1	Introduction
29.2	Governing Equations
29.3	Characteristic-Based Formulation
29.4	Maxwell Equations in a Curvilinear Frame
29.5	Eigenvalues and Eigenvectors
29.6	Flux-Vector Splitting
29.7	Finite-Difference Approximation
29.8	Finite-Volume Approximation
29.9	Summary and Research Issues

J. S. Shang
Air Force Research

Nomenclature

\mathbf{B}	= Magnetic flux density
\mathbf{C}	= Coefficient matrix of flux-vector formulation
\mathbf{D}	= Electric displacement
\mathbf{E}	= Electric field strength
\mathbf{F}	= Flux vector component
\mathbf{H}	= Magnetic flux intensity
i, j, k	= Index of discretization
\mathbf{J}	= Electric current density
n	= Index of temporal level of solution
\mathbf{S}	= Similar matrix of diagonalization
t	= Time
\mathbf{U}	= Dependent variables
V	= Elementary-cell volume
x, y, z	= Cartesian coordinates
Δ	= Forward difference operator
λ	= Eigenvalue
ξ, η, ζ	= Transformed coordinates
∇	= Gradient, backward difference operator

29.1 Introduction

Computational electromagnetics (CEM) is a natural extension of the analytical approach in solving the Maxwell equations. In spite of the fundamental difference between representing the solution in a continuum and in a discretized space, both approaches satisfy all pertaining theorems rigorously. The analytic approach

to electromagnetics is elegant, and the results can describe the specific behavior as well as the general patterns of a physical phenomenon in a given regime. However, exact solutions to the Maxwell equations are usually unavailable. Some of the closed-form results that exist have restrictive underlying assumptions that limit their range of validity. Solutions of CEM generate only a point value for a specific simulation, but complexity of the physics or of the field configuration is no longer a limiting factor. The numerical accuracy of CEM is an issue to be addressed. Nevertheless, with the advent of high-performance computing systems, CEM is becoming a mainstay for engineering applications.

CEM in the present context is focused on simulation methods for solving the Maxwell equations in the time domain. First of all, time dependence is the most general form of the Maxwell equations, and the dynamic electromagnetic field is not confined to a time-harmonic phenomenon. Therefore, CEM in the time domain has the widest range of engineering applications. In addition, several new numerical algorithms for solving the first-order hyperbolic partial differential equations, as well as coordinate transformation techniques, were introduced recently to the CEM community. These finite-difference and finite-volume numerical algorithms were devised specifically to mimic the physics involving directional wave propagation. Meanwhile, very complex shapes associated with the field can be easily accommodated by incorporating a coordinate transformation technique. These methodologies have the potential to radically change future research in electromagnetics.

In order to use CEM effectively, it will be beneficial to understand the fundamentals of numerical simulation and its limitations. The inaccuracy incurred by a numerical simulation is attributable to the mathematical model for the physics, the numerical algorithm, and the computational accuracy. In general, differential equations in computational electromagnetics consist of two categories: the first-order divergence-curl equations and the second-order curl-curl equations [Elliott 1966, Harrington 1961, 1968]. In specific applications, further simplifications into the frequency-domain or the Helmholtz equations and the potential formulation have been accomplished. Poor numerical approximations to physical phenomena can result, however, from solving overly simplified governing equations. Under these circumstances, no meaningful quantification of errors for the numerical procedure can be achieved. Physically incorrect values and inappropriate implementation of initial and/or boundary conditions are another major source of error. The placement of the far-field boundary and the type of initial or boundary conditions have also played an important role. These concerns are easily appreciated in the light of the fact that the governing equations are identical, but the different initial/boundary conditions generate different solutions.

Numerical accuracy is also controlled by the algorithm and computing system adopted. The error induced by the discretization consists of the roundoff and the truncation error. The roundoff error is contributed by the computing system and is problem-size-dependent. Since this error is random, it is the most difficult to evaluate. One anticipates that this type of error will be a concern for solution procedures involving large-scale matrix manipulation such as the method of moments and the implicit numerical algorithm for finite-difference or finite-volume methods. The truncation error for time-dependent calculations appears as dissipation and dispersion, which can be assessed and alleviated by mesh-system refinements.

Finally, numerical error can be the consequence of a specific formulation. The error becomes pronounced when a special phenomenon is investigated or when a discontinuous and distinctive stratification of the field is encountered, such as a wave propagating through the interface between media of different characteristic impedances, for which the solution is piecewise continuous. Only in a strongly conservative formulation can the discontinuous phenomenon be adequately resolved. Another example is encountered in radar cross-section simulation, where the scattered-field formulation has been shown to be superior to the total-field formulation.

The Maxwell equations in the time domain consist of a first-order divergence-curl system and are difficult to solve by conventional numerical methods. Nevertheless, the pioneering efforts by Yee and others have attained impressive achievements [Yee 1966, Taflovie 1992]. Recently, numerical techniques in CEM have been further enriched by development in the field of computational fluid dynamics (CFD). In CFD, the Euler equations, which are a subset of the Navier-Stokes equations, have the same partial-differential-system

classification as that of the time-dependent Maxwell equations. Both are hyperbolic systems and constitute initial-value problems [Sommerfeld 1949]. For hyperbolic partial differential equations, the solutions need not be analytic functions. More importantly, the initial values together with any possible discontinuities are continued along a time–space trajectory, which is commonly referred to as the characteristic. A series of numerical schemes have been devised in the CFD community to duplicate the directional information-propagation feature. These numerical procedures are collectively designated as the characteristic-based method, which in its most elementary form is identical to the Riemann problem [Roe 1986]. The characteristic-based method when applied to solve the Maxwell equations in the time domain has exhibited many attractive attributes. A synergism of the new numerical procedures and scalable parallel-computing capability will open up a new frontier in electromagnetics research. For this reason, a major portion of the present chapter will be focused on introducing the characteristic-based finite-volume and finite-difference methods [Shang 1993, Shang and Gaitonde 1993, Shang and Fithen 1994].

29.2 Governing Equations

The time-dependent Maxwell equations for the electromagnetic field can be written as [Elliott 1966, Harrington 1961]:

$$\frac{\partial \mathbf{B}}{\partial t} + \nabla \times \mathbf{E} = \mathbf{0} \quad (29.1)$$

$$\frac{\partial \mathbf{D}}{\partial t} - \nabla \times \mathbf{H} = -\mathbf{J} \quad (29.2)$$

$$\nabla \cdot \mathbf{B} = 0 \quad (29.3)$$

$$\nabla \cdot \mathbf{D} = \rho$$

The only conservation law for electric charge and current densities is

$$\frac{\partial \rho}{\partial t} + \nabla \cdot \mathbf{J} = 0 \quad (29.4)$$

where ρ and \mathbf{J} are the charge and current density, respectively, and represent the source of the field. The constitutive relations between the magnetic flux density and intensity and between the electric displacement and field strength are $\mathbf{B} = \mu\mathbf{H}$ and $\mathbf{D} = \epsilon\mathbf{E}$. Equation 29.4 is regarded as a fundamental law of electromagnetics, derived from the generalized Ampere's circuit law and Gauss's law. Since Equation 29.1 and Equation 29.2 contain the information on the propagation of the electromagnetic field, they constitute the basic equations of CEM.

The above partial differential equations also can be expressed as a system of integral equations. The following expression is obtained by using Stoke's law and the divergence theorem to reduce the surface and volume integrals to a circuital line and surface integrals, respectively [Elliott 1966]. These integral relationships hold only if the first derivatives of the electric displacement \mathbf{D} and the magnetic flux density \mathbf{B} are continuous throughout the control volume:

$$\oint \mathbf{E} \cdot d\mathbf{L} = - \iint \frac{\partial \mathbf{B}}{\partial t} \cdot d\mathbf{S} \quad (29.5)$$

$$\oint \mathbf{H} \cdot d\mathbf{L} = \iint \left(\mathbf{J} + \frac{\partial \mathbf{D}}{\partial t} \right) \cdot d\mathbf{S} \quad (29.6)$$

$$\iiint \mathbf{D} \cdot d\mathbf{S} = \iiint \rho \, dV \quad (29.7)$$

$$\iint \mathbf{B} \cdot d\mathbf{S} = 0 \quad (29.8)$$

The integral form of the Maxwell equations is rarely used in CEM. They are, however, invaluable as a validation tool for checking the global behavior of field computations.

The second-order curl–curl form of the Maxwell equations is derived by applying the curl operator to get

$$\nabla \times \nabla \times \mathbf{E} + \frac{1}{c^2} \frac{\partial^2 \mathbf{E}}{\partial t^2} = -\frac{\partial(\mu \mathbf{J})}{\partial t} \quad (29.9)$$

$$\nabla \times \nabla \times \mathbf{B} + \frac{1}{c^2} \frac{\partial^2 \mathbf{B}}{\partial t^2} = \nabla \times (\mu \mathbf{J}) \quad (29.10)$$

The outstanding feature of the curl–curl formulation of the Maxwell equations is that the electric and magnetic fields are decoupled. The second-order equations can be further simplified for harmonic fields. If the time-dependent behavior can be represented by a harmonic function $e^{i\omega t}$, the separation-of-variables technique will transform the Maxwell equations into the frequency domain [Elliott 1966, Harrington 1961]. The resultant partial differential equations in spatial variables become elliptic:

$$\begin{aligned} \nabla \times \nabla \times \mathbf{E} - k^2 \mathbf{E} &= -i\omega(\mu \mathbf{J}) \\ \nabla \times \nabla \times \mathbf{B} - k^2 \mathbf{B} &= \nabla \times (\mu \mathbf{J}) \end{aligned} \quad (29.11)$$

where $k = \omega/c$ is called the propagation constant or the wave number, and ω is the angular frequency of a component of a Fourier series or a Fourier integral [Elliott 1966, Harrington 1961]. The above equations are frequently the basis for finite-element approaches [Rahman et al. 1991].

In order to complete the description of the differential system, initial and/or boundary values are required. For Maxwell equations, only the source of the field and a few physical boundary conditions at the media interfaces are pertinent [Elliott 1966, Harrington 1961]:

$$\begin{aligned} \mathbf{n} \times (\mathbf{E}_1 - \mathbf{E}_2) &= \mathbf{0} \\ \mathbf{n} \times (\mathbf{H}_1 - \mathbf{H}_2) &= \mathbf{J}_s \\ \mathbf{n} \cdot (\mathbf{D}_1 - \mathbf{D}_2) &= \rho_s \\ \mathbf{n} \cdot (\mathbf{B}_1 - \mathbf{B}_2) &= 0 \end{aligned} \quad (29.12)$$

where the subscripts 1 and 2 refer to media on two sides of the interface, and \mathbf{J}_s and ρ_s are the surface current and charge densities, respectively.

Since all computing systems have finite memory, all CEM computations in the time domain must be conducted on a truncated computational domain. This intrinsic constraint requires a numerical far-field condition at the truncated boundary to mimic the behavior of an unbounded field. This numerical boundary unavoidably induces a reflected wave to contaminate the simulated field. In the past, absorbing boundary conditions at the far-field boundary have been developed from the radiation condition [Sommerfeld 1949, Enquist and Majda 1977, Higdon 1986, Mur 1981]. In general, a progressive order-of-accuracy procedure can be used to implement the numerical boundary conditions with increasing accuracy [Enquist and Majda 1977, Higdon 1986]. On the other hand, the characteristic-based methods which satisfy the physical domain of influence requirement can specify the numerical boundary condition readily. For this formulation, the reflected wave can be suppressed by eliminating the undesirable incoming numerical data. Although the accuracy of the numerical far-field boundary condition depends on the coordinate system, in principle this formulation under ideal circumstances can effectively suppress artificial wave reflections.

29.3 Characteristic-Based Formulation

The fundamental idea of the characteristic-based method for solving the hyperbolic system of equations is derived from the eigenvalue–eigenvector analyses of the governing equations. For Maxwell equations in the time domain, every eigenvalue is real but not all of them are distinct [Shang 1993, Shang and Gaitonde 1993, Shang and Fithen 1994]. In a time–space plane, the eigenvalue actually defines the slope of the characteristic or the phase velocity of the wave motion. All dependent variables within the time–space domain bounded by two intersecting characteristics are completely determined by the values along these characteristics and by their compatibility relationship. The direction of information propagation is also clearly described by these two characteristics [Sommerfeld 1949]. In numerical simulation, the well-posedness requirement on initial or boundary conditions and the stability of a numerical approximation are also ultimately linked to the eigenvalues of the governing equation [Anderson et al. 1984, Richtmyer and Morton 1967]. Therefore, characteristic-based methods have demonstrated superior numerical stability and accuracy to other schemes [Roe 1986, Shang 1993]. However, characteristic-based algorithms also have an inherent limitation in that the governing equation can be diagonalized only in one spatial dimension at a time. The multidimensional equations are required to split into multiple one-dimensional formulations. This limitation is not unusual for numerical algorithms, such as the approximate factored and the fractional-step schemes [Shang 1993, Anderson et al. 1984]. A consequence of this restriction is that solutions of the characteristic-based procedure may exhibit some degree of sensitivity to the orientation of the coordinate selected. This numerical behavior is consistent with the concept of optimal coordinates.

In the characteristic formulation, data on the wave motion are first split according to the direction of phase velocity and then transmitted in each orientation. In each one-dimensional time–space domain, the direction of the phase velocity degenerates into either a positive or a negative orientation. They are commonly referred to as the right-running and the left-running wave components [Sommerfeld 1949, Roe 1986]. The sign of the eigenvalue is thus an indicator of the direction of signal transmission. The corresponding eigenvectors are the essential elements for diagonalizing the coefficient matrices and for formulating the approximate Riemann problem [Roe 1986]. In essence, knowledge of eigenvalues and eigenvectors of the Maxwell equations in the time domain becomes the first prerequisite of the present formulation.

The system of governing equations cast in the flux-vector form in the Cartesian frame becomes [Shang 1993, Shang and Gaitonde 1993, Shang and Fithen 1994]

$$\frac{\partial U}{\partial t} + \frac{\partial F_x}{\partial x} + \frac{\partial F_y}{\partial y} + \frac{\partial F_z}{\partial z} = -J \quad (29.13)$$

where U is the vector of dependent variables. The flux vectors are formed by the inner product of the coefficient matrix and the dependent variable: $F_x = C_x U$, $F_y = C_y U$, and $F_z = C_z U$, with

$$U = \{B_x, B_y, B_z, D_x, D_y, D_z\}^T \quad (29.14)$$

and

$$\begin{aligned} F_x &= \{0, -D_z/\epsilon, D_y/\epsilon, 0, B_z/\mu, -B_y/\mu\}^T \\ F_y &= \{D_z/\epsilon, 0, -D_x/\epsilon, -B_z/\mu, 0, B_x/\mu\}^T \\ F_z &= \{-D_y/\epsilon, D_x/\epsilon, 0, B_y/\mu, -B_x/\mu, 0\}^T \end{aligned} \quad (29.15)$$

The coefficient matrices, or the Jacobians of the flux vectors C_x , C_y , and C_z are [Shang 1993]:

$$C_x = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -\frac{1}{\epsilon} \\ 0 & 0 & 0 & 0 & \frac{1}{\epsilon} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{\mu} & 0 & 0 & 0 \\ 0 & -\frac{1}{\mu} & 0 & 0 & 0 & 0 \end{bmatrix} \quad (29.16a)$$

$$C_y = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & \frac{1}{\epsilon} \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -\frac{1}{\epsilon} & 0 & 0 \\ 0 & 0 & -\frac{1}{\mu} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{1}{\mu} & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (29.16b)$$

$$C_z = \begin{bmatrix} 0 & 0 & 0 & 0 & -\frac{1}{\epsilon} & 0 \\ 0 & 0 & 0 & \frac{1}{\epsilon} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{\mu} & 0 & 0 & 0 & 0 \\ -\frac{1}{\mu} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (29.16c)$$

where ϵ and μ are the permittivity and permeability, which relate the electric displacement to the electric field intensity and the magnetic flux density to the magnetic field intensity, respectively.

The eigenvalues of the coefficient matrices C_x , C_y , and C_z in the Cartesian frame are identical and contain multiplicities [Shang 1993, Shang and Gaitonde 1993]. Care must be exercised to ensure that all associated eigenvectors

$$\lambda = \left\{ +\frac{1}{\sqrt{\epsilon\mu}}, -\frac{1}{\sqrt{\epsilon\mu}}, 0, +\frac{1}{\sqrt{\epsilon\mu}}, -\frac{1}{\sqrt{\epsilon\mu}}, 0 \right\} \quad (29.17)$$

are linearly independent. The linearly independent eigenvectors associated with each eigenvalue are found by reducing the matrix equation, $(C - \bar{\lambda}I)U = 0$, to the Jordan normal form [Shang 1993, Shang and Fithen 1994].

Since the coefficient matrices C_x , C_y , and C_z can be diagonalized, there exist nonsingular similar matrices S_x , S_y , and S_z such that

$$\begin{aligned} \Lambda_x &= S_x^{-1} C_x S_x \\ \Lambda_y &= S_y^{-1} C_y S_y \\ \Lambda_z &= S_z^{-1} C_z S_z \end{aligned} \quad (29.18)$$

where the Λ s are the diagonalized coefficient matrices. The columns of the similar matrices S_x , S_y , and S_z are simply the linearly independent eigenvectors of the coefficient matrices C_x , C_y , and C_z , respectively.

The fundamental relationship between the characteristic-based formulation and the Riemann problem can be best demonstrated in the Cartesian frame of reference. For the Maxwell equations in this frame of reference and for an isotropic medium, all the similar matrices are invariant with respect to temporal and spatial independent variables. In each time-space plane ($x-t$, $y-t$, and $z-t$), the one-dimensional governing equation can be given just as in the $x-t$ plane:

$$\frac{\partial U}{\partial t} + C_x \frac{\partial U}{\partial x} = 0 \quad (29.19)$$

Substitute the diagonalized coefficient matrix to get

$$\frac{\partial U}{\partial t} + S_x \Lambda_x S_x^{-1} \frac{\partial U}{\partial x} = 0 \quad (29.20)$$

Since the similar matrix in the present consideration is invariant with respect to time and space, it can be brought into the differential operator. Multiplying the above equation by the left-hand inverse of S_x , S_x^{-1} , we have

$$\frac{\partial (S_x^{-1} U)}{\partial t} + \Lambda_x \frac{\partial (S_x^{-1} U)}{\partial x} = 0 \quad (29.21)$$

One immediately recognizes the group of variables $S_x^{-1} U$ as the characteristic, and the system of equations is decoupled [Shang 1993]. In scalar-variable form and with appropriate initial values, this is the Riemann problem [Sommerfeld 1949, Courant and Hilbert 1965]. This differential system is specialized to study the breakup of a single discontinuity. The piecewise continuous solutions separated by the singular point are also invariant along the characteristics. Equally important, stable numerical operators can now be easily devised to solve the split equations according to the sign of the eigenvalue. In practice it has been found if the multidimensional problem can be split into a sequence of one-dimensional equations, this numerical technique is applicable to those one-dimensional equations [Roe 1986, Shang 1993].

The gist of the characteristic-based formulation is also clearly revealed by the decomposition of the flux vector into positive and negative components corresponding to the sign of the eigenvalue:

$$\Lambda = \Lambda^+ + \Lambda^-, \quad F = F^+ + F^- \quad (29.22)$$

$$F^+ = S \Lambda^+ S^{-1}, \quad F^- = S \Lambda^- S^{-1} \quad (29.23)$$

where the superscripts $+$ and $-$ denote the split vectors associated with positive and negative eigenvalues, respectively.

The characteristic-based algorithms have a deep-rooted theoretical basis for describing the wave dynamics. They also have however an inherent limitation in that the diagonalized formulation is achievable only in one dimension at a time. All multidimensional equations are required to be split into multiple one-dimensional formulations. The approach yields accurate results so long as discontinuous waves remain aligned with the computational grid. This limitation is also the state-of-the-art constraint in solving partial differential equations [Roe 1986, Shang 1993, Anderson et al. 1984].

29.4 Maxwell Equations in a Curvilinear Frame

In order to develop a versatile numerical tool for computational electromagnetics in a wide range of applications, the Maxwell equations can be cast in a general curvilinear frame of reference [Shang and Gaitonde 1993, Shang and Fithen 1994]. For efficient simulation of complex electromagnetic field configurations, the adoption of a general curvilinear mesh system becomes necessary. The system of equations in general curvilinear coordinates is derived by a coordinate transformation [Anderson et al. 1984, Thompson 1982]. The mesh system in the transformed space can be obtained by numerous grid generation procedures [Thompson 1982]. Computational advantages in the transformed space are also realizable. For a body-oriented coordinate system, the interface between two different media is easily defined by one of the coordinate surfaces. Along this coordinate parametric plane, all discretized nodes on the interface are precisely prescribed without the need for an interpolating procedure. The outward normal to the interface, which is essential for boundary-value implementation, can be computed easily by $n = \nabla S / \|\nabla S\|$. In the transformed space, computations are performed on a uniform mesh space, but the corresponding physical spacing can be highly clustered to enhance the numerical resolution.

As an illustration of the numerical advantage of solving the Maxwell equations on nonorthogonal curvilinear, body-oriented coordinates, a simulation of the scattered electromagnetic field from a re-entry vehicle has been performed [Shang and Gaitonde 1994]. The aerospace vehicle, X24C-10D, has



FIGURE 29.1 Radar-wave fringes on X24C-10D, grid $181 \times 59 \times 162$, TE excitation, $L/\lambda = 9.2$.

a complex geometrical shape (Figure 29.1). In addition to a blunt leading-edge spherical nose and a relatively flat delta-shaped underbody, the aft portion of the vehicle consists of five control surfaces — a central fin, two middle fins, and two strakes. A body-oriented, single-block mesh system enveloping the configuration is adopted. The numerical grid system is generated by using a hyperbolic grid generator for the near-field mesh adjacent to the solid surface, and a transfinite technique for the far field. The two mesh systems are merged by the Poisson averaging technique [Thompson 1982, Shang and Gaitonde 1994]. In this manner, the composite grid system is orthogonal in the near field but less restrictive in the far field. All solid surfaces of the X24C-10D are mapped onto a parametric surface in the transformed space, defined by $\eta = 0$. The entire computational domain is supported by a $181 \times 59 \times 162$ grid system, where the first coordinate index denotes the number of cross sections in the numerical domain. The second index describes the number of cells between the body surface and the far-field boundary, while the third index gives the cells used to circumscribe each cross-sectional plane. The electromagnetic excitation is introduced by a harmonic incident wave traveling along the x -coordinate. The fringe pattern of the scattered electromagnetic waves on the X24C-10D is presented in Figure 29.1 for a characteristic-length-to-wavelength ratio $L/\lambda = 9.2$. A salient feature of the scattered field is brought out by the surface curvature: the smaller the radius of surface curvature, the broader the diffraction pattern. The numerical result exhibits highly concentrated contours at the chine (the line of intersection between upper and lower vehicle surfaces) of the forebody and the leading edges of strakes and fins.

For the most general coordinate transformation of the Maxwell equations in the time domain, a one-to-one relationship between two sets of temporal and spatial independent variables is required. However, for most practical applications, the spatial coordinate transformation is sufficient:

$$\begin{aligned}\xi &= \xi(x, y, z) \\ \eta &= \eta(x, y, z) \\ \zeta &= \zeta(x, y, z)\end{aligned}\tag{29.24}$$

The governing equation in the strong conservation form is obtained by dividing the chain-rule-differentiated equations by the Jacobian of coordinate transformation and by invoking metric identities [Shang and Gaitonde 1993, Anderson et al. 1984]. The time-dependent Maxwell equations on a general

curvilinear frame of reference and in the strong conservative form are

$$\frac{\partial U}{\partial t} + \frac{\partial F_\xi}{\partial \xi} + \frac{\partial F_\eta}{\partial \eta} + \frac{\partial F_\zeta}{\partial \zeta} = -J \quad (29.25)$$

where the dependent variables are now defined as

$$U = U(B_x V, B_y V, B_z V, D_x V, D_y V, D_z V) \quad (29.26)$$

Here V is the Jacobian of the coordinate transformation and is also the inverse local cell volume. If the Jacobian has nonzero values in the computational domain, the correspondence between the physical and the transformed space is uniquely defined [Anderson et al. 1984, Thompson 1982]. Since systematic procedures have been developed to ensure this property of coordinate transformations, detailed information on this point is not repeated here [Anderson et al. 1984, Thompson 1982]. We have

$$V = \begin{bmatrix} \xi_x & \eta_x & \zeta_x \\ \xi_y & \eta_y & \zeta_y \\ \xi_z & \eta_z & \zeta_z \end{bmatrix} \quad (29.27)$$

and ξ_x, η_x, ζ_x , etc. are the metrics of coordinate transformation and can be computed easily from the definition given by Equation 29.24. The flux-vector components in the transformed space have the following form:

$$F_\xi = \begin{bmatrix} 0 & 0 & 0 & 0 & -\frac{\xi_z}{\epsilon V} & \frac{\xi_y}{\epsilon V} \\ 0 & 0 & 0 & \frac{\xi_z}{\epsilon V} & 0 & -\frac{\xi_x}{\epsilon V} \\ 0 & 0 & 0 & -\frac{\xi_y}{\epsilon V} & \frac{\xi_x}{\epsilon V} & 0 \\ 0 & \frac{\xi_z}{V\mu} & -\frac{\xi_y}{V\mu} & 0 & 0 & 0 \\ -\frac{\xi_z}{V\mu} & 0 & \frac{\xi_x}{V\mu} & 0 & 0 & 0 \\ \frac{\xi_y}{V\mu} & -\frac{\xi_x}{V\mu} & 0 & 0 & 0 & 0 \end{bmatrix} \begin{Bmatrix} B_x \\ B_y \\ B_z \\ D_x \\ D_y \\ D_z \end{Bmatrix} \quad (29.28)$$

$$F_\eta = \begin{bmatrix} 0 & 0 & 0 & 0 & -\frac{\eta_z}{\epsilon V} & \frac{\eta_y}{\epsilon V} \\ 0 & 0 & 0 & \frac{\eta_z}{\epsilon V} & 0 & -\frac{\eta_x}{\epsilon V} \\ 0 & 0 & 0 & -\frac{\eta_y}{\epsilon V} & \frac{\eta_x}{\epsilon V} & 0 \\ 0 & \frac{\eta_z}{V\mu} & -\frac{\eta_y}{V\mu} & 0 & 0 & 0 \\ -\frac{\eta_z}{V\mu} & 0 & \frac{\eta_x}{V\mu} & 0 & 0 & 0 \\ \frac{\eta_y}{V\mu} & -\frac{\eta_x}{V\mu} & 0 & 0 & 0 & 0 \end{bmatrix} \begin{Bmatrix} B_x \\ B_y \\ B_z \\ D_x \\ D_y \\ D_z \end{Bmatrix} \quad (29.29)$$

$$F_\zeta = \begin{bmatrix} 0 & 0 & 0 & 0 & -\frac{\zeta_z}{\epsilon V} & \frac{\zeta_y}{\epsilon V} \\ 0 & 0 & 0 & \frac{\zeta_z}{\epsilon V} & 0 & -\frac{\zeta_x}{\epsilon V} \\ 0 & 0 & 0 & -\frac{\zeta_y}{\epsilon V} & \frac{\zeta_x}{\epsilon V} & 0 \\ 0 & \frac{\zeta_z}{V\mu} & -\frac{\zeta_y}{V\mu} & 0 & 0 & 0 \\ -\frac{\zeta_z}{V\mu} & 0 & \frac{\zeta_x}{V\mu} & 0 & 0 & 0 \\ \frac{\zeta_y}{V\mu} & -\frac{\zeta_x}{V\mu} & 0 & 0 & 0 & 0 \end{bmatrix} \begin{Bmatrix} B_x \\ B_y \\ B_z \\ D_x \\ D_y \\ D_z \end{Bmatrix} \quad (29.30)$$

After the coordinate transformation, all coefficient matrices now contain metrics which are position-dependent, and the system of equations in the most general frame of reference possesses variable coefficients. This added complexity of the characteristic formulation of the Maxwell equations no longer permits the system of one-dimensional equations to be decoupled into six scalar equations and reduced to the true Riemann problem [Shang 1993, Shang and Gaitonde 1993, Shang and Fithen 1994] like that on the Cartesian form.

29.5 Eigenvalues and Eigenvectors

As previously mentioned, eigenvalue and the eigenvector analyses are the prerequisites for characteristic-based algorithms. The analytic process to obtain the eigenvalues and the corresponding eigenvectors of the Maxwell equations in general curvilinear coordinates is identical to that in the Cartesian frame. In each of the temporal-spatial planes t - ξ , t - η , and t - ζ , the eigenvalues are easily found by solving the sixth-degree characteristic equation associated with the coefficient matrices [Sommerfeld 1949, Courant and Hilbert 1965]

$$\lambda_\xi = \left\{ -\frac{\alpha}{V\sqrt{\epsilon\mu}}, -\frac{\alpha}{V\sqrt{\epsilon\mu}}, \frac{\alpha}{V\sqrt{\epsilon\mu}}, \frac{\alpha}{V\sqrt{\epsilon\mu}}, 0, 0 \right\} \quad (29.31)$$

$$\lambda_\eta = \left\{ -\frac{\beta}{V\sqrt{\epsilon\mu}}, -\frac{\beta}{V\sqrt{\epsilon\mu}}, \frac{\beta}{V\sqrt{\epsilon\mu}}, \frac{\beta}{V\sqrt{\epsilon\mu}}, 0, 0 \right\} \quad (29.32)$$

$$\lambda_\zeta = \left\{ -\frac{\gamma}{V\sqrt{\epsilon\mu}}, -\frac{\gamma}{V\sqrt{\epsilon\mu}}, \frac{\gamma}{V\sqrt{\epsilon\mu}}, \frac{\gamma}{V\sqrt{\epsilon\mu}}, 0, 0 \right\} \quad (29.33)$$

where $\alpha = \sqrt{\xi_z^2 + \xi_y^2 + \xi_x^2}$, $\beta = \sqrt{\eta_z^2 + \eta_y^2 + \eta_x^2}$, and $\gamma = \sqrt{\zeta_z^2 + \zeta_y^2 + \zeta_x^2}$.

One recognizes that the eigenvalues in each one-dimensional time-space plane contain multiplicities, and hence the eigenvectors do not necessarily have unique elements [Shang 1993, Courant and Hilbert 1965]. Nevertheless, linearly independent eigenvectors associated with each eigenvalue still have been found by reducing the coefficient matrix to the Jordan normal form [Shang 1993, Shang and Fithen 1994]. For reasons of wide applicability and internal consistency, the eigenvectors are selected in such a fashion that the similar matrices of diagonalization reduce to the same form as in the Cartesian frame. Furthermore, in order to accommodate a wide range of electromagnetic field configurations such as antennas, wave guides, and scatterers, the eigenvalues are no longer identical in the three time-space planes. This complexity of formulation is essential to facilitate boundary-condition implementation on the interfaces of media with different characteristic impedances.

From the eigenvector analysis, the similarity transformation matrices for diagonalization in each time-space plane are formed by using eigenvectors as the column arrays as shown in the following equations. For an example, the first column of the similar matrix of diagonalization,

$$\left[-\frac{\sqrt{\mu}\xi_y}{\sqrt{\epsilon\alpha}}, \frac{\sqrt{\mu}(\xi_x^2 + \xi_z^2)}{\sqrt{\epsilon\xi_x\alpha}}, \frac{\sqrt{\mu}\xi_y\xi_z}{\sqrt{\epsilon\xi_x\alpha}}, \frac{\xi_y}{\xi_x}, 0, 1 \right]$$

in the t - ξ plane is the eigenvector corresponding to the eigenvalue $\lambda_\xi = -\alpha/V\sqrt{\epsilon\mu}$. We have

$$S_\xi = \begin{bmatrix} -\frac{\sqrt{\mu}\xi_y}{\sqrt{\epsilon\alpha}} & \frac{\sqrt{\mu}\xi_z}{\sqrt{\epsilon\alpha}} & \frac{\sqrt{\mu}\xi_y}{\sqrt{\epsilon\alpha}} & -\frac{\sqrt{\mu}\xi_z}{\sqrt{\epsilon\alpha}} & 1 & 0 \\ \frac{\sqrt{\mu}(\xi_x^2 + \xi_z^2)}{\sqrt{\epsilon\xi_x\alpha}} & \frac{\sqrt{\mu}\xi_y\xi_z}{\sqrt{\epsilon\xi_x\alpha}} & -\frac{\sqrt{\mu}(\xi_x^2 + \xi_z^2)}{\sqrt{\epsilon\xi_x\alpha}} & -\frac{\sqrt{\mu}\xi_y\xi_z}{\sqrt{\epsilon\xi_x\alpha}} & \frac{\xi_y}{\xi_x} & 0 \\ -\frac{\sqrt{\mu}\xi_y\xi_z}{\sqrt{\epsilon\xi_x\alpha}} & -\frac{\sqrt{\mu}(\xi_x^2 + \xi_z^2)}{\sqrt{\epsilon\xi_x\alpha}} & \frac{\sqrt{\mu}\xi_y\xi_z}{\sqrt{\epsilon\xi_x\alpha}} & \frac{\sqrt{\mu}(\xi_x^2 + \xi_z^2)}{\sqrt{\epsilon\xi_x\alpha}} & \frac{\xi_z}{\xi_x} & 0 \\ -\frac{\xi_z}{\xi_x} & -\frac{\xi_y}{\xi_x} & -\frac{\xi_z}{\xi_x} & -\frac{\xi_y}{\xi_x} & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & \frac{\xi_y}{\xi_x} \\ 1 & 0 & 1 & 0 & 0 & \frac{\xi_z}{\xi_x} \end{bmatrix} \quad (29.34)$$

$$S_\eta = \begin{bmatrix} -\frac{(\eta_y^2 + \eta_z^2)\sqrt{\mu}}{\sqrt{\epsilon}\eta_y\beta} & -\frac{\eta_x\eta_z\sqrt{\mu}}{\sqrt{\epsilon}\eta_y\beta} & \frac{(\eta_y^2 + \eta_z^2)\sqrt{\mu}}{\sqrt{\epsilon}\eta_y\beta} & \frac{\eta_x\eta_z\sqrt{\mu}}{\sqrt{\epsilon}\eta_y\beta} & \frac{\eta_x}{\eta_y} & 0 \\ \frac{\eta_x\sqrt{\mu}}{\sqrt{\epsilon}\beta} & -\frac{\eta_z\sqrt{\mu}}{\sqrt{\epsilon}\beta} & -\frac{\eta_x\sqrt{\mu}}{\sqrt{\epsilon}\beta} & \frac{\eta_z\sqrt{\mu}}{\sqrt{\epsilon}\beta} & 1 & 0 \\ \frac{\eta_x\eta_z\sqrt{\mu}}{\sqrt{\epsilon}\eta_y\beta} & \frac{(\eta_x^2 + \eta_y^2)\sqrt{\mu}}{\sqrt{\epsilon}\eta_y\beta} & -\frac{\eta_x\eta_z\sqrt{\mu}}{\sqrt{\epsilon}\eta_y\beta} & -\frac{(\eta_x^2 + \eta_y^2)\sqrt{\mu}}{\sqrt{\epsilon}\eta_y\beta} & \frac{\eta_x}{\eta_y} & 0 \\ 0 & 1 & 0 & 1 & 0 & \frac{\eta_x}{\eta_y} \\ -\frac{\eta_z}{\eta_y} & -\frac{\eta_x}{\eta_y} & -\frac{\eta_z}{\eta_y} & -\frac{\eta_x}{\eta_y} & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & \frac{\eta_z}{\eta_y} \end{bmatrix} \quad (29.35)$$

$$S_\xi = \begin{bmatrix} \frac{\sqrt{\mu}(\xi_y^2 + \xi_z^2)}{\sqrt{\epsilon}\xi_z\gamma} & \frac{\sqrt{\mu}\xi_x\xi_y}{\sqrt{\epsilon}\xi_z\gamma} & -\frac{\sqrt{\mu}(\xi_y^2 + \xi_z^2)}{\sqrt{\epsilon}\xi_z\gamma} & -\frac{\sqrt{\mu}\xi_x\xi_y}{\sqrt{\epsilon}\xi_z\gamma} & \frac{\xi_x}{\xi_z} & 0 \\ -\frac{\sqrt{\mu}\xi_x\xi_y}{\sqrt{\epsilon}\xi_z\gamma} & -\frac{\sqrt{\mu}(\xi_x^2 + \xi_z^2)}{\sqrt{\epsilon}\xi_z\gamma} & \frac{\sqrt{\mu}\xi_x\xi_y}{\sqrt{\epsilon}\xi_z\gamma} & \frac{\sqrt{\mu}(\xi_x^2 + \xi_z^2)}{\sqrt{\epsilon}\xi_z\gamma} & \frac{\xi_y}{\xi_z} & 0 \\ -\frac{\sqrt{\mu}\xi_x}{\sqrt{\epsilon}\gamma} & \frac{\sqrt{\mu}\xi_y}{\sqrt{\epsilon}\gamma} & \frac{\sqrt{\mu}\xi_x}{\sqrt{\epsilon}\gamma} & -\frac{\sqrt{\mu}\xi_y}{\sqrt{\epsilon}\gamma} & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & \frac{\xi_x}{\xi_z} \\ 1 & 0 & 1 & 0 & 0 & \frac{\xi_y}{\xi_z} \\ -\frac{\xi_y}{\xi_z} & -\frac{\xi_x}{\xi_z} & -\frac{\xi_y}{\xi_z} & -\frac{\xi_x}{\xi_z} & 0 & 1 \end{bmatrix} \quad (29.36)$$

Since the similar matrices of diagonalization, S_ξ , S_η , and S_ζ , are nonsingular, the left-hand inverse matrices S_ξ^{-1} , S_η^{-1} , and S_ζ^{-1} are easily found. Although these left-hand inverse matrices are essential to the diagonalization process, they provide little insight for the following flux-vector splitting procedure. The rather involved results are omitted here, but they can be found in Shang and Fithen [1994].

29.6 Flux-Vector Splitting

An efficient flux-vector splitting algorithm for solving the Euler equations was developed by Steger and Warming [1987]. The basic concept is equally applicable to any hyperbolic differential system for which the solution need not be analytic [Sommerfeld 1949, Courant and Hilbert 1965]. In most CEM applications, discontinuous behavior of the solution is associated only with the wave across an interface between different media, a piecewise continuous solution. Even if a jump condition exists, the magnitude of the finite jump across the interface is much less drastic than the shock waves encountered in supersonic flows. Nevertheless, the salient feature of the piecewise continuous solution domains of the hyperbolic partial differential equation stands out: The coefficient matrices of the time-dependent, three-dimensional Maxwell equations cast in the general curvilinear frame of reference contain metrics of coordinate transformation. Therefore, the equation system no longer has constant coefficients even in an isotropic and homogeneous medium. Under this circumstance, eigenvalues can change sign at any given field location due to the metric variations of the coordinate transformation. Numerical oscillations have appeared in results calculated using the flux-vector splitting technique when eigenvalues change sign. A refined flux-difference splitting algorithm has been developed to resolve fields with jump conditions [Van Leer 1982, Anderson et al. 1985]. The newer flux-difference splitting algorithm is particularly effective at locations where the eigenvalues vanish. Perhaps more crucial for electromagnetics, the polarization of the medium, making the basic equations become nonlinear, occurs only in the extremely high-frequency range [Elliott 1966, Harrington 1961]. In general the governing equations are linear; at most, the coefficients of the differential system are dependent on physical location and phase velocity. For this reason, the difference between the flux-vector splitting [Steger and Warming 1987] and flux-difference splitting [Van Leer 1982, Anderson et al. 1985] schemes, when applied to the time-dependent Maxwell equations, is not of great importance.

The basic idea of the flux-vector splitting of Steger and Warming is to process data according to the direction of information propagation. Since diagonalization is achievable only in each time-space plane,

the direction of wave propagation degenerates into either the positive or the negative orientation. This designation is consistent with the notion of the right-running and the left-running wave components. The flux vectors are computed from the point value, including the metrics at the node of interest. This formulation for solving hyperbolic partial differential equations not only ensures the well-posedness of the differential system but also enhances the stability of the numerical procedure [Roe 1986, Shang 1993, Anderson et al. 1984, Richtmyer and Morton 1967]. Specifically, the flux vectors F_ξ , F_η , and F_ζ will be split according to the sign of their corresponding eigenvalues. The split fluxes are differenced by an upwind algorithm to allow for the zone of dependence of an initial-value problem [Roe 1986, Shang 1993, Shang and Gaitonde 1993, Shang and Fithen 1994].

From the previous analysis, it is clear that the eigenvalues contain multiplicities, and hence the split flux of the three-dimensional Maxwell equations is not unique [Shang and Gaitonde 1993, Shang and Fithen 1994]. All flux vectors in each time-space plane are split according to the signs of the local eigenvalues:

$$\begin{aligned} F_\xi &= F_\xi^+ + F_\xi^- \\ F_\eta &= F_\eta^+ + F_\eta^- \\ F_\zeta &= F_\zeta^+ + F_\zeta^- \end{aligned} \quad (29.37)$$

The flux-vector components associated with the positive and negative eigenvalues are obtainable by a straightforward matrix multiplication:

$$\begin{aligned} F_\xi^+ &= S_\xi \lambda_\xi^+ S_\xi^{-1} U \\ F_\xi^- &= S_\xi \lambda_\xi^- S_\xi^{-1} U \\ F_\eta^+ &= S_\eta \lambda_\eta^+ S_\eta^{-1} U \\ F_\eta^- &= S_\eta \lambda_\eta^- S_\eta^{-1} U \\ F_\zeta^+ &= S_\zeta \lambda_\zeta^+ S_\zeta^{-1} U \\ F_\zeta^- &= S_\zeta \lambda_\zeta^- S_\zeta^{-1} U \end{aligned} \quad (29.38)$$

It is also important to recognize that even if the split flux vectors in each time-space plane are non-unique, the sum of the split components must be unambiguously identical to the flux vector of the governing Equation 29.25. This fact is easily verifiable by performing the addition of the split matrices to reach the identities in Equation 29.28, Equation 29.29, and Equation 29.30. In addition, if one sets the diagonal elements of metrics, ξ_x , η_y , and ζ_z equal to unity and the off-diagonal elements equal to zero, the coefficient matrices will recover the Cartesian form:

$$F_\xi^+ = \begin{bmatrix} \frac{\xi_y^2 + \xi_z^2}{2\sqrt{\epsilon\mu}V\alpha} & \frac{-\xi_x\xi_y}{2\sqrt{\epsilon\mu}V\alpha} & \frac{-\xi_x\xi_z}{2\sqrt{\epsilon\mu}V\alpha} & 0 & \frac{-\xi_x}{2\epsilon V} & \frac{\xi_y}{2\epsilon V} \\ \frac{-\xi_x\xi_y}{2\sqrt{\epsilon\mu}V\alpha} & \frac{\xi_x^2 + \xi_z^2}{2\sqrt{\epsilon\mu}V\alpha} & \frac{-\xi_y\xi_z}{2\sqrt{\epsilon\mu}V\alpha} & \frac{\xi_z}{2\epsilon V} & 0 & \frac{-\xi_x}{2\epsilon V} \\ \frac{-\xi_x\xi_z}{2\sqrt{\epsilon\mu}V\alpha} & \frac{-\xi_y\xi_z}{2\sqrt{\epsilon\mu}V\alpha} & \frac{\xi_x^2 + \xi_y^2}{2\sqrt{\epsilon\mu}V\alpha} & \frac{-\xi_y}{2\epsilon V} & \frac{\xi_x}{2\epsilon V} & 0 \\ 0 & \frac{\xi_z}{2V\mu} & \frac{-\xi_y}{2V\mu} & \frac{\xi_y^2 + \xi_z^2}{2\sqrt{\epsilon\mu}V\alpha} & \frac{-\xi_x\xi_y}{2\sqrt{\epsilon\mu}V\alpha} & \frac{-\xi_x\xi_z}{2\sqrt{\epsilon\mu}V\alpha} \\ \frac{-\xi_z}{2V\mu} & 0 & \frac{\xi_y}{2V\mu} & \frac{-\xi_x\xi_y}{2\sqrt{\epsilon\mu}V\alpha} & \frac{\xi_x^2 + \xi_z^2}{2\sqrt{\epsilon\mu}V\alpha} & \frac{-\xi_y\xi_z}{2\sqrt{\epsilon\mu}V\alpha} \\ \frac{\xi_y}{2V\mu} & \frac{-\xi_x}{2V\mu} & 0 & \frac{-\xi_x\xi_z}{2\sqrt{\epsilon\mu}V\alpha} & \frac{-\xi_y\xi_z}{2\sqrt{\epsilon\mu}V\alpha} & \frac{\xi_x^2 + \xi_y^2}{2\sqrt{\epsilon\mu}V\alpha} \end{bmatrix} \begin{Bmatrix} B_x \\ B_y \\ B_z \\ D_x \\ D_y \\ D_z \end{Bmatrix} \quad (29.39)$$

$$F_{\xi}^{-} = \begin{bmatrix} \frac{-(\xi_y^2 + \xi_z^2)}{2\sqrt{\epsilon\mu}V\alpha} & \frac{\xi_x\xi_y}{2\sqrt{\epsilon\mu}V\alpha} & \frac{\xi_x\xi_z}{2\sqrt{\epsilon\mu}V\alpha} & 0 & \frac{-\xi_z}{2\epsilon V} & \frac{\xi_y}{2\epsilon V} \\ \frac{\xi_x\xi_y}{2\sqrt{\epsilon\mu}V\alpha} & \frac{-(\xi_x^2 + \xi_z^2)}{2\sqrt{\epsilon\mu}V\alpha} & \frac{\xi_y\xi_z}{2\sqrt{\epsilon\mu}V\alpha} & \frac{\xi_z}{2\epsilon V} & 0 & \frac{-\xi_x}{2\epsilon V} \\ \frac{\xi_x\xi_z}{2\sqrt{\epsilon\mu}V\alpha} & \frac{\xi_y\xi_z}{2\sqrt{\epsilon\mu}V\alpha} & \frac{-(\xi_x^2 + \xi_y^2)}{2\sqrt{\epsilon\mu}V\alpha} & \frac{-\xi_y}{2\epsilon V} & \frac{\xi_x}{2\epsilon V} & 0 \\ 0 & \frac{\xi_z}{2V\mu} & \frac{-\xi_y}{2V\mu} & \frac{-(\xi_y^2 + \xi_z^2)}{2\sqrt{\epsilon\mu}V\alpha} & \frac{\xi_x\xi_y}{2\sqrt{\epsilon\mu}V\alpha} & \frac{\xi_x\xi_z}{2\sqrt{\epsilon\mu}V\alpha} \\ \frac{-\xi_x}{2V\mu} & 0 & \frac{\xi_x}{2V\mu} & \frac{\xi_x\xi_y}{2\sqrt{\epsilon\mu}V\alpha} & \frac{-(\xi_x^2 + \xi_z^2)}{2\sqrt{\epsilon\mu}V\alpha} & \frac{\xi_y\xi_z}{2\sqrt{\epsilon\mu}V\alpha} \\ \frac{\xi_y}{2V\mu} & \frac{-\xi_x}{2V\mu} & 0 & \frac{\xi_x\xi_z}{2\sqrt{\epsilon\mu}V\alpha} & \frac{\xi_y\xi_z}{2\sqrt{\epsilon\mu}V\alpha} & \frac{-(\xi_x^2 + \xi_y^2)}{2\sqrt{\epsilon\mu}V\alpha} \end{bmatrix} \begin{Bmatrix} B_x \\ B_y \\ B_z \\ D_x \\ D_y \\ D_z \end{Bmatrix} \quad (29.40)$$

$$F_{\eta}^{+} = \begin{bmatrix} \frac{\eta_y^2 + \eta_z^2}{2\sqrt{\epsilon\mu}\beta V} & \frac{-\eta_x\eta_y}{2\sqrt{\epsilon\mu}\beta V} & \frac{-\eta_x\eta_z}{2\sqrt{\epsilon\mu}\beta V} & 0 & \frac{-\eta_z}{2\epsilon V} & \frac{\eta_y}{2\epsilon V} \\ \frac{-\eta_x\eta_y}{2\sqrt{\epsilon\mu}\beta V} & \frac{\eta_x^2 + \eta_z^2}{2\sqrt{\epsilon\mu}\beta V} & \frac{-\eta_y\eta_z}{2\sqrt{\epsilon\mu}\beta V} & \frac{\eta_z}{2\epsilon V} & 0 & \frac{-\eta_x}{2\epsilon V} \\ \frac{-\eta_x\eta_z}{2\sqrt{\epsilon\mu}\beta V} & \frac{-\eta_y\eta_z}{2\sqrt{\epsilon\mu}\beta V} & \frac{\eta_x^2 + \eta_y^2}{2\sqrt{\epsilon\mu}\beta V} & \frac{-\eta_y}{2\epsilon V} & \frac{\eta_x}{2\epsilon V} & 0 \\ 0 & \frac{\eta_z}{2V\mu} & \frac{-\eta_y}{2V\mu} & \frac{\eta_y^2 + \eta_z^2}{2\sqrt{\epsilon\mu}\beta V} & \frac{-\eta_x\eta_y}{2\sqrt{\epsilon\mu}\beta V} & \frac{-\eta_x\eta_z}{2\sqrt{\epsilon\mu}\beta V} \\ \frac{-\eta_z}{2V\mu} & 0 & \frac{\eta_x}{2V\mu} & \frac{-\eta_x\eta_y}{2\sqrt{\epsilon\mu}\beta V} & \frac{\eta_x^2 + \eta_z^2}{2\sqrt{\epsilon\mu}\beta V} & \frac{-\eta_y\eta_z}{2\sqrt{\epsilon\mu}\beta V} \\ \frac{\eta_y}{2V\mu} & \frac{-\eta_x}{2V\mu} & 0 & \frac{-\eta_x\eta_z}{2\sqrt{\epsilon\mu}\beta V} & \frac{-\eta_y\eta_z}{2\sqrt{\epsilon\mu}\beta V} & \frac{\eta_x^2 + \eta_y^2}{2\sqrt{\epsilon\mu}\beta V} \end{bmatrix} \begin{Bmatrix} B_x \\ B_y \\ B_z \\ D_x \\ D_y \\ D_z \end{Bmatrix} \quad (29.41)$$

$$F_{\eta}^{-} = \begin{bmatrix} \frac{-(\eta_y^2 + \eta_z^2)}{2\sqrt{\epsilon\mu}\beta V} & \frac{\eta_x\eta_y}{2\sqrt{\epsilon\mu}\beta V} & \frac{\eta_x\eta_z}{2\sqrt{\epsilon\mu}\beta V} & 0 & \frac{-\eta_z}{2\epsilon V} & \frac{\eta_y}{2\epsilon V} \\ \frac{\eta_x\eta_y}{2\sqrt{\epsilon\mu}\beta V} & \frac{-(\eta_x^2 + \eta_z^2)}{2\sqrt{\epsilon\mu}\beta V} & \frac{\eta_y\eta_z}{2\sqrt{\epsilon\mu}\beta V} & \frac{\eta_z}{2\epsilon V} & 0 & \frac{-\eta_x}{2\epsilon V} \\ \frac{\eta_x\eta_z}{2\sqrt{\epsilon\mu}\beta V} & \frac{\eta_y\eta_z}{2\sqrt{\epsilon\mu}\beta V} & \frac{-(\eta_x^2 + \eta_y^2)}{2\sqrt{\epsilon\mu}\beta V} & \frac{-\eta_y}{2\epsilon V} & \frac{\eta_x}{2\epsilon V} & 0 \\ 0 & \frac{\eta_z}{2V\mu} & \frac{-\eta_y}{2V\mu} & \frac{-(\eta_y^2 + \eta_z^2)}{2\sqrt{\epsilon\mu}\beta V} & \frac{\eta_x\eta_y}{2\sqrt{\epsilon\mu}\beta V} & \frac{\eta_x\eta_z}{2\sqrt{\epsilon\mu}\beta V} \\ \frac{-\eta_z}{2V\mu} & 0 & \frac{\eta_x}{2V\mu} & \frac{\eta_x\eta_y}{2\sqrt{\epsilon\mu}\beta V} & \frac{-(\eta_x^2 + \eta_z^2)}{2\sqrt{\epsilon\mu}\beta V} & \frac{\eta_y\eta_z}{2\sqrt{\epsilon\mu}\beta V} \\ \frac{\eta_y}{2V\mu} & \frac{-\eta_x}{2V\mu} & 0 & \frac{\eta_x\eta_z}{2\sqrt{\epsilon\mu}\beta V} & \frac{\eta_y\eta_z}{2\sqrt{\epsilon\mu}\beta V} & \frac{-(\eta_x^2 + \eta_y^2)}{2\sqrt{\epsilon\mu}\beta V} \end{bmatrix} \begin{Bmatrix} B_x \\ B_y \\ B_z \\ D_x \\ D_y \\ D_z \end{Bmatrix} \quad (29.42)$$

$$F_{\zeta}^{+} = \begin{bmatrix} \frac{\zeta_y^2 + \zeta_z^2}{2\sqrt{\epsilon\mu}V\gamma} & \frac{-\zeta_x\zeta_y}{2\sqrt{\epsilon\mu}V\gamma} & \frac{-\zeta_x\zeta_z}{2\sqrt{\epsilon\mu}V\gamma} & 0 & \frac{-\zeta_z}{2\epsilon V} & \frac{\zeta_y}{2\epsilon V} \\ \frac{-\zeta_x\zeta_y}{2\sqrt{\epsilon\mu}V\gamma} & \frac{\zeta_x^2 + \zeta_z^2}{2\sqrt{\epsilon\mu}V\gamma} & \frac{-\zeta_y\zeta_z}{2\sqrt{\epsilon\mu}V\gamma} & \frac{\zeta_z}{2\epsilon V} & 0 & \frac{-\zeta_x}{2\epsilon V} \\ \frac{-\zeta_x\zeta_z}{2\sqrt{\epsilon\mu}V\gamma} & \frac{-\zeta_y\zeta_z}{2\sqrt{\epsilon\mu}V\gamma} & \frac{\zeta_x^2 + \zeta_y^2}{2\sqrt{\epsilon\mu}V\gamma} & \frac{-\zeta_y}{2\epsilon V} & \frac{\zeta_x}{2\epsilon V} & 0 \\ 0 & \frac{\zeta_z}{2V\mu} & \frac{-\zeta_y}{2V\mu} & \frac{\zeta_y^2 + \zeta_z^2}{2\sqrt{\epsilon\mu}V\gamma} & \frac{-\zeta_x\zeta_y}{2\sqrt{\epsilon\mu}V\gamma} & \frac{-\zeta_x\zeta_z}{2\sqrt{\epsilon\mu}V\gamma} \\ \frac{-\zeta_z}{2V\mu} & 0 & \frac{\zeta_x}{2V\mu} & \frac{-\zeta_x\zeta_y}{2\sqrt{\epsilon\mu}V\gamma} & \frac{\zeta_x^2 + \zeta_z^2}{2\sqrt{\epsilon\mu}V\gamma} & \frac{-\zeta_y\zeta_z}{2\sqrt{\epsilon\mu}V\gamma} \\ \frac{\zeta_y}{2V\mu} & \frac{-\zeta_x}{2V\mu} & 0 & \frac{-\zeta_x\zeta_z}{2\sqrt{\epsilon\mu}V\gamma} & \frac{-\zeta_y\zeta_z}{2\sqrt{\epsilon\mu}V\gamma} & \frac{\zeta_x^2 + \zeta_y^2}{2\sqrt{\epsilon\mu}V\gamma} \end{bmatrix} \begin{Bmatrix} B_x \\ B_y \\ B_z \\ D_x \\ D_y \\ D_z \end{Bmatrix} \quad (29.43)$$

$$F_{\zeta}^{-} = \begin{bmatrix} \frac{-(\zeta_y^2 + \zeta_z^2)}{2\sqrt{\epsilon\mu}V\gamma} & \frac{\zeta_x\zeta_y}{2\sqrt{\epsilon\mu}V\gamma} & \frac{\zeta_x\zeta_z}{2\sqrt{\epsilon\mu}V\gamma} & 0 & \frac{-\zeta_z}{2\epsilon V} & \frac{\zeta_y}{2\epsilon V} \\ \frac{\zeta_x\zeta_y}{2\sqrt{\epsilon\mu}V\gamma} & \frac{-(\zeta_x^2 + \zeta_z^2)}{2\sqrt{\epsilon\mu}V\gamma} & \frac{\zeta_y\zeta_z}{2\sqrt{\epsilon\mu}V\gamma} & \frac{\zeta_x}{2\epsilon V} & 0 & \frac{-\zeta_x}{2\epsilon V} \\ \frac{\zeta_x\zeta_z}{2\sqrt{\epsilon\mu}V\gamma} & \frac{\zeta_y\zeta_z}{2\sqrt{\epsilon\mu}V\gamma} & \frac{-(\zeta_x^2 + \zeta_y^2)}{2\sqrt{\epsilon\mu}V\gamma} & \frac{-\zeta_y}{2\epsilon V} & \frac{\zeta_x}{2\epsilon V} & 0 \\ 0 & \frac{\zeta_x}{2V\mu} & \frac{-\zeta_y}{2V\mu} & \frac{-(\zeta_y^2 + \zeta_z^2)}{2\sqrt{\epsilon\mu}V\gamma} & \frac{\zeta_x\zeta_y}{2\sqrt{\epsilon\mu}V\gamma} & \frac{\zeta_x\zeta_z}{2\sqrt{\epsilon\mu}V\gamma} \\ \frac{-\zeta_z}{2V\mu} & 0 & \frac{\zeta_x}{2V\mu} & \frac{\zeta_x\zeta_y}{2\sqrt{\epsilon\mu}V\gamma} & \frac{-(\zeta_x^2 + \zeta_z^2)}{2\sqrt{\epsilon\mu}V\gamma} & \frac{\zeta_y\zeta_z}{2\sqrt{\epsilon\mu}V\gamma} \\ \frac{\zeta_y}{2V\mu} & \frac{-\zeta_x}{2V\mu} & 0 & \frac{\zeta_x\zeta_z}{2\sqrt{\epsilon\mu}V\gamma} & \frac{\zeta_y\zeta_z}{2\sqrt{\epsilon\mu}V\gamma} & \frac{-(\zeta_x^2 + \zeta_y^2)}{2\sqrt{\epsilon\mu}V\gamma} \end{bmatrix} \begin{Bmatrix} B_x \\ B_y \\ B_z \\ D_x \\ D_y \\ D_z \end{Bmatrix} \quad (29.44)$$

29.7 Finite-Difference Approximation

Once the detailed split fluxes are known, formulation of the finite-difference approximation is straightforward. From the sign of an eigenvalue, the stencil of a spatially second- or higher-order-accurate windward differencing can be easily constructed to form multiple one-dimensional difference operators [Shang 1993, Anderson et al. 1984, Richtmyer and Morton 1967]. In this regard, the forward difference and the backward difference approximations are used for the negative and the positive eigenvalues, respectively. The split flux vectors are evaluated at each discretized point of the field according to the signs of the eigenvalues. For the present purpose, a second-order accurate procedure is given:

$$\begin{aligned} \text{If } \lambda < 0, \quad \Delta U_i &= [-3U_i + 4U_{i+1} - U_{i+2}] / 2 \\ \text{If } \lambda > 0, \quad \nabla U_i &= [3U_i - 4U_{i-1} + U_{i-2}] / 2 \end{aligned} \quad (29.45)$$

The necessary metrics of the coordinate transformation are calculated by central-differencing, except at the edges of computational domain, where one-sided differences are used. Although the fractional-step or the time-splitting algorithm [Shang 1993, Anderson et al. 1984, Richtmyer and Morton 1967] has demonstrated greater efficiency in data storage and a higher data-processing rate than predictor–corrector time integration procedures [Shang 1993, Shang and Gaitonde 1993, Shang and Fithen 1994], it is limited to second-order accuracy in time. With respect to the fractional-step method, the temporal second-order result is obtained by a sequence of symmetrically cyclic operators [Shang 1993, Richtmyer and Morton 1967]:

$$U^{n+2} = L_{\xi} L_{\eta} L_{\zeta} L_{\xi} L_{\eta} L_{\xi} U^n \quad (29.46)$$

where L_{ξ} , L_{η} , and L_{ζ} are the difference operators for one-dimensional equations in the ξ , η , and ζ coordinates, respectively.

In general, second-order and higher temporal resolution is achievable through multiple-time-step schemes [Anderson et al. 1984, Richtmyer and Morton 1967]. However, one-step schemes are more attractive because they have less memory requirements and don't need special startup procedures [Shang and Gaitonde 1993, Shang and Fithen 1994]. For future higher-order accurate solution development potential, the Runge–Kutta family of single-step, multistage procedures is recommended. This choice is also consistent with the accompanying characteristic-based finite-volume method [Shang and Gaitonde 1993].

In the present effort, the two-stage, formally second-order accurate scheme is used:

$$\begin{aligned}
 U_0 &= U_n \\
 U_1 &= U_0 - \Delta U(U_0) \\
 U_2 &= U_0 - 0.5(\Delta U(U_1) + \Delta U(U_0)) \\
 U_{n+1} &= U_2
 \end{aligned} \tag{29.47}$$

where ΔU comprises the incremental values of dependent variables during each temporal sweep. The resultant characteristic-based finite-difference scheme for solving the three-dimensional Maxwell equations in the time domain is second-order accurate in both time and space.

The most significant feature of the flux-vector splitting scheme lies in its ability to easily suppress reflected waves from the truncated computational domain. In wave motion, the compatibility condition at any point in space is described by the split flux vector [Shang 1993, Shang and Gaitonde 1993, Shang and Fithen 1994]. In the present formulation, an approximated no-reflection condition can be achieved by setting the incoming flux component equal to zero:

$$\text{either } \lim_{r \rightarrow \infty} F^+(\xi, \eta, \zeta) = 0 \quad \text{or} \quad \lim_{r \rightarrow \infty} F^-(\xi, \eta, \zeta) = 0 \tag{29.48}$$

The one-dimensional compatibility condition is exact when the wave motion is aligned with one of the coordinates [Shang 1993]. This unique attribute of the characteristic-based numerical procedure in removing a fundamental dilemma in computational electromagnetics will be demonstrated in detail later.

29.8 Finite-Volume Approximation

The finite-volume approximation solves the governing equation by discretizing the physical space into contiguous cells and balancing the flux-vectors on the cell surfaces. Thus in discretized form, the integration procedure reduces to evaluation of the sum of all fluxes aligned with surface-area vectors

$$\frac{\Delta U}{\Delta t} + \frac{\Delta F}{\Delta \xi} + \frac{\Delta G}{\Delta \eta} + \frac{\Delta H}{\Delta \zeta} - J = 0 \tag{29.49}$$

In the present approach, the continuous differential operators have been replaced by discrete operators. In essence, the numerical procedure needs only to evaluate the sum of all flux vectors aligned with surface-area vectors [Shang and Gaitonde 1993, Shang and Fithen 1994, Van Leer 1982, Anderson et al. 1985]. Only one of the vectors is required to coincide with the outward normal to the cell surface, and the rest of the orthogonal triad can be made to lie on the same surface. The metrics, or more appropriately the direction cosines, on the cell surface are uniquely determined by the nodes and edges of the elementary volume. This feature is distinct from the finite-difference approximation. The shape of the cell under consideration and the stretching ratio of neighbor cells can lead to a significant deterioration of the accuracy of finite-volume schemes [Leonard 1988].

The most outstanding aspect of finite-volume schemes is the elegance of its flux-splitting process. The flux-difference splitting for Equation 29.25 is greatly facilitated by a locally orthogonal system in the transformed space [Van Leer 1982, Anderson et al. 1985]. In this new frame of reference, eigenvalues and eigenvectors as well as metrics of the coordinate transformation between two orthogonal systems are well known [Shang 1993, Shang and Gaitonde 1993]. The inverse transformation is simply the transpose of the forward mapping. In particular, the flux vectors in the transformed space have the same functional form as that in the Cartesian frame. The difference between the flux vectors in the transformed and the Cartesian coordinates is a known quantity and is given by the product of the surface outward normal and the cell volume, $V(\nabla S / \|\nabla S\|)$ [Shang and Gaitonde 1993]. Therefore, the flux vectors can be split in the transformed space according to the signs of the eigenvalues but without detailed knowledge of the associated eigenvectors in the transformed space. This feature of the finite-volume approach

provides a tremendous advantage over the finite-difference approximation in solving complex problems in physics.

The present formulation adopts Van Leer's kappa scheme in which solution vectors are reconstructed on the cell surface from the piecewise data of neighboring cells [Van Leer 1982, Anderson et al. 1985]. The spatial accuracy of this scheme spans a range from first-order to third-order upwind biased approximations,

$$\begin{aligned} U_{i+\frac{1}{2}}^+ &= U_i + \frac{\phi}{4} [(1 - \kappa)\nabla + (1 + \kappa)\Delta] U_i \\ U_{i+\frac{1}{2}}^- &= U_i - \frac{\phi}{4} [(1 + \kappa)\nabla + (1 - \kappa)\Delta] U_{i+1} \end{aligned} \quad (29.50)$$

where $\Delta U_i = U_i - U_{i-1}$ and $\nabla U_i = U_{i+1} - U_i$ are the forward and backward differencing discretizations. The parameters ϕ and κ control the accuracy of the numerical results. For $\phi = 1$, $\kappa = -1$ a two-point windward scheme is obtained. This method has an odd-order leading truncation-error term; the dispersive error is expected to dominate. If $\kappa = \frac{1}{3}$, a third-order upwind-biased scheme will emerge. In fact both upwind procedures have discernible leading phase error. This behavior is a consequence of using the two-stage time integration algorithm, and the dispersive error can be alleviated by increasing the temporal resolution. For $\phi = 1$, $\kappa = 0$ the formulation recovers the Fromm scheme [Van Leer 1982, Anderson et al. 1985]. If $\kappa = 1$, the formulation yields the spatially central scheme. Since the fourth-order dissipative term is suppressed, the central scheme is susceptible to parasitic odd-even point decoupling [Anderson et al. 1984, 1985].

The time integration is carried out by the same two-stage Runge–Kutta method as in the present finite-difference procedure [Shang 1993, Shang and Gaitonde 1993]. The finite-volume procedure is therefore second-order accurate in time and up to third-order accurate in space [Shang and Gaitonde 1993, Shang and Fithen 1994]. For the present purpose, only the second-order upwinding and the third-order upwind biased options are exercised. The second-order windward schemes in the form of the flux-vector splitting finite-difference and the flux-difference splitting finite-volume scheme are formally equivalent [Shang and Gaitonde 1993, Shang and Fithen 1994, Van Leer 1982, Anderson et al. 1985, Leonard 1988].

29.9 Summary and Research Issues

The technical merits of the characteristic-based methods for solving the time-dependent, three-dimensional Maxwell equations can best be illustrated by the following two illustrations. In [Figure 29.2](#), the exact electrical field of a traveling wave is compared with numerical results. The numerical results were generated at the maximum allowable time-step size defined by the Courant–Friedrichs–Lewy (CFL) number of 2 ($\lambda \Delta x / \Delta t = 2$) [Anderson et al. 1984, Richtmyer and Morton 1967]. The numerical solutions presented are at instants when a right-running wave reaches the midpoint of the computational domain and exits the numerical boundary respectively. For this one-dimensional simulation, the characteristic-based scheme using the single-step upwind explicit algorithm exhibits the shift property, which indicates a perfect translation of the initial value in space [Anderson et al. 1984]. As the impulse wave moves through the initially quiescent environment, the numerical result duplicates the exact solution at each and every discretized point, including the discontinuous incoming wavefront. Although this highly desirable property of a numerical solution is achievable only under very restrictive conditions and is not preserved for multidimensional problems [Anderson et al. 1984, Richtmyer and Morton 1967], the ability to simulate the nonanalytic solution behavior in the limit is clearly illustrated.

In [Figure 29.3](#), another outstanding feature of the characteristic-based method is highlighted by simulating the oscillating electric dipole. For the radiating electric dipole, the depicted temporal calculations are sampled at the instant when the initial pulse has traveled a distance of 2.24 wavelengths from the dipole. The numerical results are generated on a $48 \times 48 \times 96$ mesh system with the second-order scheme. Under that condition each wavelength is resolved by 15 mesh points and the difference between numerical results by the finite-volume and the finite-difference method is negligible. However, on an irregular

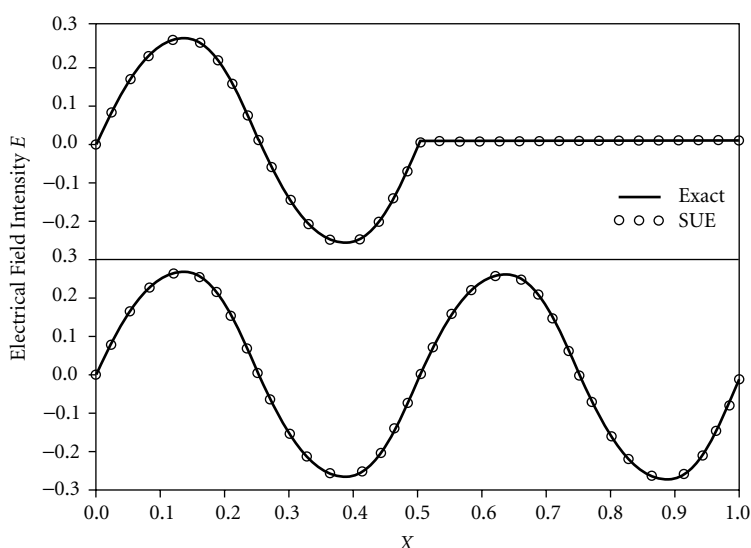


FIGURE 29.2 Perfect-shift property of a one-dimensional wave computation, CFL = 2.

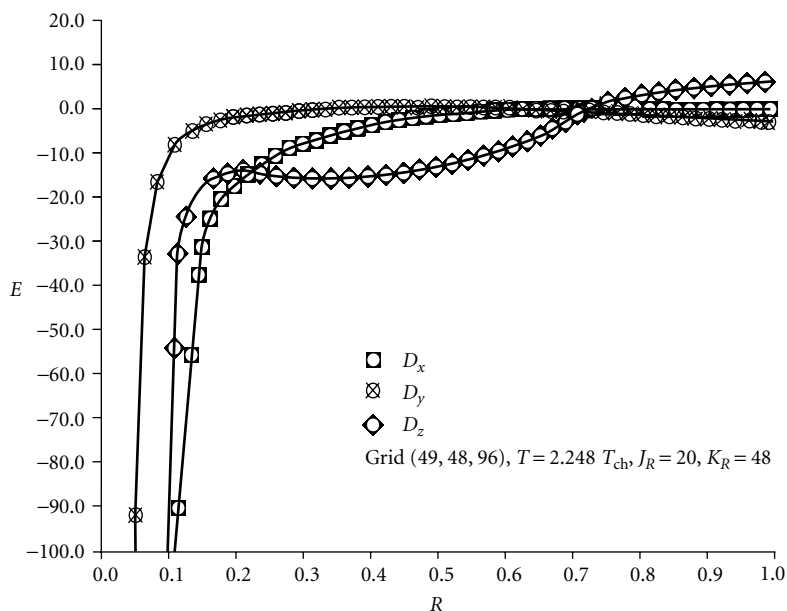


FIGURE 29.3 Instantaneous distributions of oscillating-dipole electric field.

mesh like the present spherical polar system, the finite-volume procedure has shown a greater numerical error than the finite-difference procedure on highly stretched grid systems [Shang and Fithen 1994, Anderson et al. 1985]. Under the present computational conditions, both numerical procedures yield uniformly excellent comparison with the theoretical result. Significant error appeared only in the immediate region of the coordinate origin. At that location, the solution has a singularity that behaves as the inverse cube of the radial distance. The most interesting numerical behavior, however, is revealed in the truncated far field. The no-reflection condition at the numerical boundary is observed to be satisfied

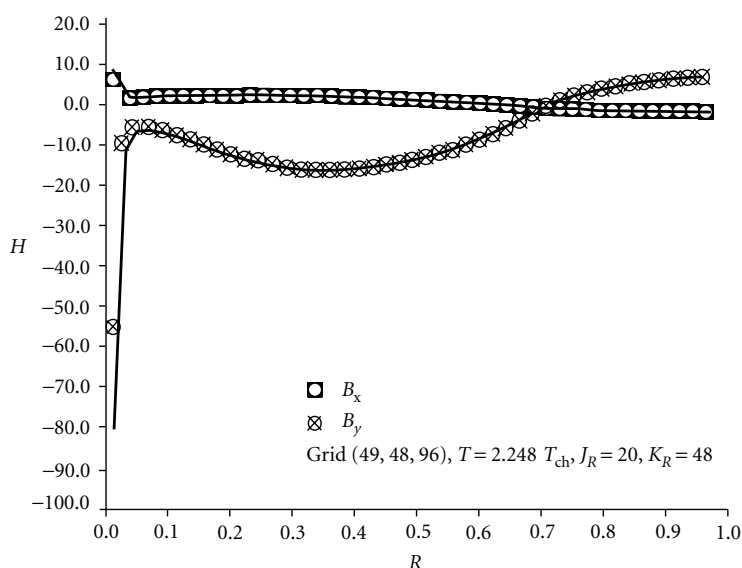


FIGURE 29.4 Instantaneous distributions of oscillating-dipole magnetic field.

within the order of the truncation error. For a spherically symmetric radiating field, the orientation of the wave is aligned with the radial coordinate, and the suppression of the reflected wave within the numerical domain is the best achievable by the characteristic formulation.

The corresponding magnetic field intensity computed by both the second-order accurate finite-difference and the finite-volume procedure is given in Figure 29.4. Again the difference in solution between the two distinct numerical procedures is indiscernible. For the oscillating electric dipole, only the x and y components of the magnetic field exist. Numerical results attain excellent agreement with theoretical values [Shang and Gaitonde 1993, Shang and Fithen 1994]. The third-order accurate finite-volume scheme also produces a similar result on the same mesh but at a greater allowable time-step size (a CFL value of 0.87 is used vs. 0.5 for the second-order method). A numerically more efficient and higher-order accurate simulation is obtainable in theory. However, at the present, the third-order windward biased algorithm cannot reinforce rigorously the zone-of-dependence requirement; therefore the reflected-wave suppression is incomplete in the truncated numerical domain. For this reason, the third-order accurate results are not included here.

Numerical accuracy and efficiency are closely related issues in computational electromagnetics. A high-accuracy requirement of a simulation is supportable only by efficient numerical procedures. The inaccuracies incurred by numerical simulations are attributable to the mathematical formulation of the problem, to the algorithm, to the numerical procedure, and to computational inaccuracy. A basic approach to relieve the accuracy limitation must be derived from using high-order schemes or spectral methods. The numerical efficiency of CEM can be enhanced substantially by using scalable multicomputers [Shang et al. 1993]. The effective use of a distributed-memory, message-passing homogeneous multicomputer still requires a judicious tradeoff between a balanced work load and interprocessor communication. A characteristic-based finite-volume computer program has been successfully mapped onto distributed-memory systems by a rudimentary domain decomposition strategy [Shang et al. 1993]. For example, a square waveguide, at five different frequencies up to the cutoff, was simulated.

Figure 29.5 displays the x -component of the magnetic field intensity within the waveguide. The simulated transverse electric mode, $TE_{1,1}$, $E_x = 0$, which has a half period of π along the x and y coordinates, is generated on a $24 \times 24 \times 128$ mesh system. Since the entire field is described by simple harmonic functions, the remaining field components are similar and only half the solution domain along the z -coordinate is

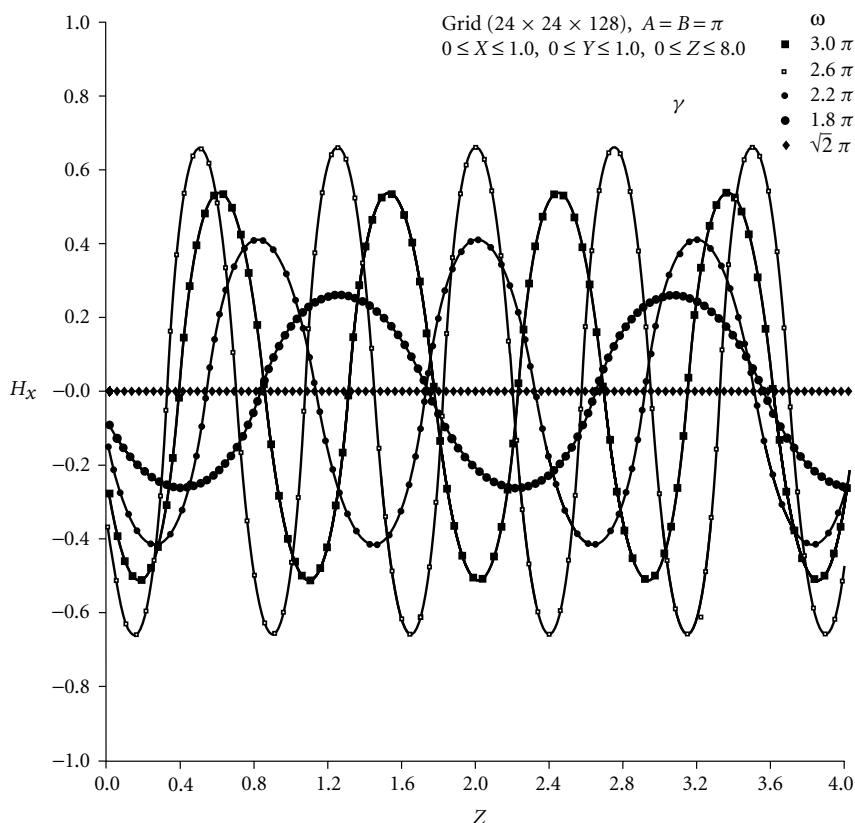


FIGURE 29.5 Cutoff frequency of a square waveguide, $TE_{1,1}$.

presented to minimize repetition. In short, the agreement between the closed-form and numerical solutions is excellent at each frequency. In addition, the numerical simulations duplicate the physical phenomenon at the cutoff frequency, below which there is no phase shift along the waveguide and the wave motion ceases [Elliott 1966, Harrington 1961]. For simple harmonic wave motion in an isotropic medium, the numerical accuracy can be quantified. At a grid-point density of 12 nodes per wavelength, the L2 norm [Richtmyer and Morton 1967] has a nearly uniform magnitude of order 10^{-4} . The numerical results are fully validated by comparison with theory. However, further efforts are still required to substantially improve the parallel and scalable numerical efficiency. In fact, this is the most promising area in CEM research.

The pioneering efforts in CEM usually employed the total-field formulation on staggered-mesh systems [Yee 1966, Taflov 1992]. That particular combination of numerical algorithm and procedure has been proven to be very effective. In earlier RCS calculations using the present numerical procedure, the total-field formulation was also utilized [Shang and Gaitonde 1994, Shang et al. 1993]. However, for three-dimensional scatterer simulation, the numerical accuracy requirement for RCS evaluations becomes extremely stringent. In the total-field formulation, the dynamic range of the field variables has a substantial difference from the exposed and the shadow region, and the incident wave must also traverse the entire computation domain. Both requirements impose severe demands on the numerical accuracy of simulation. In addition, the total field often contains only the residual of partial cancellations of the incident and the diffracted waves. The far-field electromagnetic energy distribution becomes a secular problem — a small difference between two variables of large magnitude. An alternative approach via the scattered-field formulation for RCS calculations appears to be very attractive. Particularly in this formulation, the numerical dissipation of the

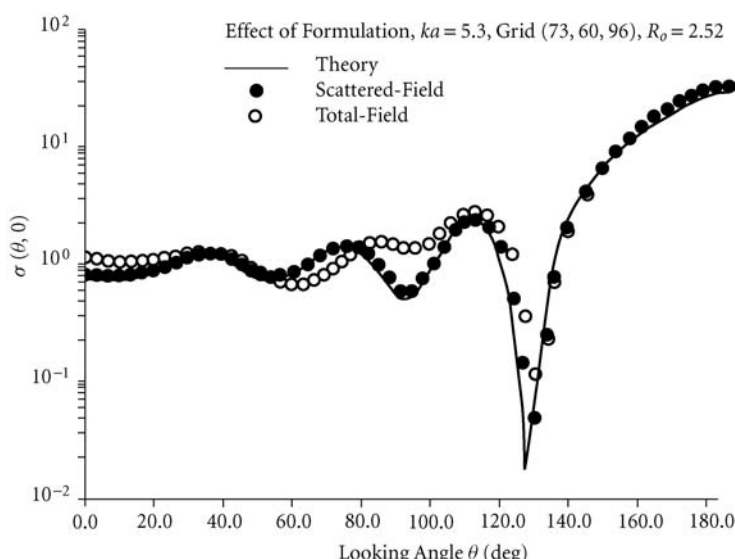


FIGURE 29.6 Comparison of total-field and scattered-field RCS calculations of $\sigma(\theta, 0^\circ)$, $ka = 5.3$.

incident wave that must propagate from the far-field boundary to the scatterer is completely eliminated from the computations. In short, the incident field can be directly specified on the scatterer's surface. The numerical advantage over the total-field formulation is substantial.

The total-field formulation can be cast in the scattered-field form by replacing the total field with scattered field variables [Elliott 1966, Harrington 1961]:

$$U_s = U_t - U_i \quad (29.51)$$

Since the incident field U_i must satisfy the Maxwell equations identically, the equations of the scattered field remain unaltered from the total-field formulation. Thus, the scattered-field formulation can be considered as a dependent-variable transform of the total-field equations. In the present approach, both formulations are solved by a characteristic-based finite-volume scheme.

The comparison of horizontal polarized RCS of a perfect electrically conducting (PEC) sphere, $\sigma(\theta, 0.0)$, from the total-field and scattered-field formulations at $ka = 5.3$ (where k = wave number and a = diameter of the sphere) is presented in Figure 29.6. The validating datum is the exact solution for the scattering of a plane electromagnetic wave by a PEC sphere, which is commonly referred to as the Mie series [Elliott 1966]. Both numerical results are generated under identical computational conditions. The location of the truncated far-field boundary is prescribed at 2.5 wavelengths from the center of the PEC sphere. Numerical results of the total-field formulation reveal far greater error than for the scattered-field formulation. The additional source of error is incurred when the incident wave must propagate from the far-field boundary to the scatterer. In the scattered-field formulation, the incident field data are described precisely by the boundary condition on the scatterer surface. Since the far-field electromagnetic energy distribution is derived from the near-field parameters [Taflov 1992, Sommerfeld 1949, Shang and Gaitonde 1993], the advantage of describing the data incident on a scatterer without error is tremendous. Numerical errors of the total-field calculations are evident in the exaggerated peaks and troughs over the entire viewing-angle displacement.

In Figure 29.7, the vertically polarized RCS $\sigma(\theta, 90.0^\circ)$ of the $ka = 5.3$ case substantiates the previous observation. In fact, the numerical error of the total-field calculation is excessive in comparison with the result of the scattered-field formulation. Since the results are not obtained for the optimal far-field placement for RCS computation, the results of the scattered-field formulation overpredict the theoretical values by 2.7 percent. The deviation of the total-field result from the theory, however, exceeds

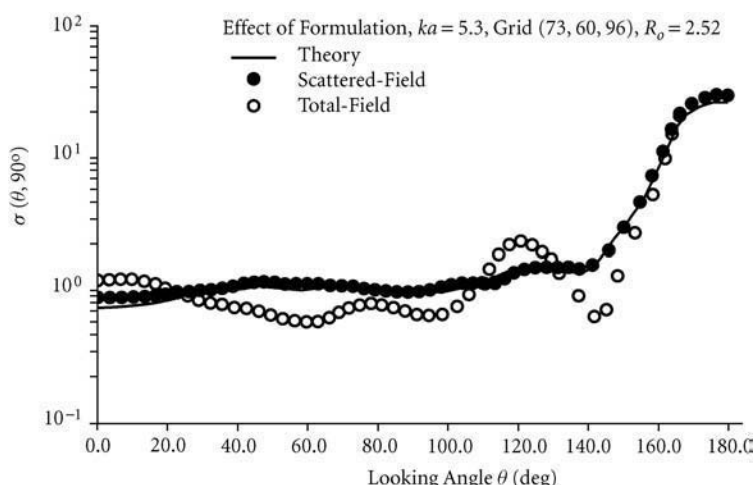


FIGURE 29.7 Comparison of total-field and scattered-field RCS calculations of $\sigma(\theta, 90^\circ)$, $ka = 5.3$.

25.6 percent and becomes unacceptable. In addition, computations by the total-field formulation exhibit a strong sensitivity to placement of the far-field boundary. A small perturbation of the far-field boundary placement leads to a drastic change in the RCS prediction: a feature resembling the ill-posedness condition, which is highly undesirable for numerical simulation. Since there is very little difference in computer coding for the two formulations, the difference in computing time required for an identical simulation is insignificant. On the Cray C90, 1,505.3 s at a data-processing rate of 528.8 Mflops and an average vector length of 62.9 is needed to complete a sampling period. At the present, the most efficient calculation on a distributed memory system, Cray T3-D, has reduced the processing time to 1,204.2 s using 76 computing nodes.

In summary, recent progress in solving the three-dimensional Maxwell equations in the time domain has opened a new frontier in electromagnetics, plasmadynamics, and optics, as well as the interface between electrodynamics and quantum mechanics [Taflove 1992]. The progress in microchip and interconnect network technology has led to a host of high-performance distributed memory, message-passing parallel computer systems. The synergism of efficient and accurate numerical algorithms for solving the Maxwell equations in the time domain with high-performance multicomputers will propel the new interdisciplinary simulation technique to practical and productive applications.

References

- Anderson, D. A., Tannehill, J. C., and Pletcher, R. H. 1984. *Computational Fluid Mechanics and Heat Transfer*. Hemisphere, New York.
- Anderson, W. K., Thomas, J. L., and Van Leer, B. 1985. A comparison of finite-volume flux splittings for the Euler equations., AIAA 23rd Aerospace Science Meeting, Reno NV, Jan. *AIAA Paper* 85-0122.
- Courant, R. and Hilbert, D. 1965. *Methods of Mathematical Physics*, Vol. II. Interscience, New York.
- Elliott, R. A. 1966. *Electromagnetics*, Ch. 5. McGraw-Hill, New York.
- Enquist, B. and Majda, A. 1977. Absorbing boundary conditions for the numerical simulation of waves. *Math Comp.* 31:629-651, July.
- Harrington, R. F. 1961. *Time-Harmonic Electromagnetic Fields*. McGraw-Hill, New York.
- Harrington, R. F. 1968. *Field Computation by Moment Methods*, 4th ed. Robert E. Krieger, Malabar, FL.
- Higdon, R. 1986. Absorbing boundary conditions for difference approximation to multidimensional wave equation. *Math Comp.* 47(175):437-459.

- Leonard, B. P. 1988. Simple high-accuracy resolution program for convective modeling of discontinuities. *Int. J. Numer. Methods Fluids* 8:1291–1318.
- Mur, G. 1981. Absorbing boundary conditions for the finite-difference approximation of the time-domain electromagnetic-field equations. *IEEE Trans. Elect. Compat.* EMC-23(4):377–382, Nov.
- Rahman, B. M. A., Fernandez, F. A., and Davies, J. B. 1991. Review of finite element methods for microwave and optical waveguide. *Proc. IEEE* 79:1442, 1448.
- Richtmyer, R. D. and Morton, K. W. 1967. *Difference Methods for Initial-Value Problem*. Interscience, New York.
- Roe, P. L. 1986. Characteristic-based schemes for the Euler equations. *Ann. Rev. Fluid Mech.* 18:337–365.
- Shang, J. S. 1993. A fractional-step method for solving 3-D, time-domain Maxwell equations. AIAA 31st Aerospace Science Meeting, Reno NV, Jan. *AIAA Paper* 93-0461; *J. Comput. Phys.* Vol. 118(1):109–119, Apr. 1995.
- Shang, J. S. and Fithen, R. M. 1994. A comparative study of numerical algorithms for computational electromagnetics. AIAA 25th Plasmadynamics and Laser Conference, Colorado Springs, June 20–23, *AIAA Paper* 94-2410.
- Shang, J. S. and Gaitonde, D. 1993. Characteristic-based, time-dependent Maxwell equation solvers on a general curvilinear frame. AIAA 24th Fluid Dynamics, Plasmadynamics, and Laser Conference, Orlando FL, July. *AIAA Paper* 93-3178; *AIAA J.* 33(3):491–498, Mar. 1995.
- Shang, J. S. and Gaitonde, D. 1994. Scattered electromagnetic field of a reentry vehicle. AIAA 32nd Aerospace Science Meeting, Reno NV, Jan. *AIAA Paper* 94-0231; *J. Spacecraft and Rockets* 32(2):294–301, Mar.–Apr. 1995.
- Shang, J. S., Hill, K. C., and Calahan, D. 1993. Performance of a characteristic-based, 3-D, time-domain Maxwell equations solver on a massively parallel computer. AIAA 24th Plasmadynamics & Lasers Conference, Orlando FL, July 6–9 *AIAA Paper* 3179; *Appl. Comput. Elect. Soc.* 10(1):52–62, Mar. 1995.
- Sommerfeld, A. 1949. *Partial Differential Equations in Physics*, Ch. 2. Academic Press, New York.
- Steger, J. L. and Warming, R. F. 1987. Flux vector splitting of the inviscid gas dynamics equations with application to finite difference methods. *J. Comput. Phys.* 20(2):263–293, Feb.
- Taflove, A. 1992. Re-inventing electromagnetics: supercomputing solution of Maxwell's equations via direct time integration on space grids. *Comput. Systems Eng.* 3(1–4):153–168.
- Thompson, J. F. 1982. *Numerical Grid Generation*. Elsevier Science, New York.
- Van Leer, B. 1982. Flux-vector splitting for the Euler equations. *TR* 82-30, ICASE, Sept., pp. 507–512, *Lecture Notes in Physics*, Vol. 170.
- Yee, K. S. 1966. Numerical solution of initial boundary value problems involving Maxwell's equations. In *Isotropic Media*, *IEEE Trans. Ant. Prop.* 14(3):302–307.

30

Computational Fluid Dynamics

- 30.1 Introduction
- 30.2 Underlying Principles
 - Fluid-Dynamical Background • Treatment of Geometry
- 30.3 Best Practices
 - Panel Methods • Nonlinear Methods • Navier–Stokes Equation Methods
- 30.4 Research Issues and Summary

David A. Caughey
Cornell University

30.1 Introduction

The use of computer-based methods for the prediction of fluid flows has seen tremendous growth in the past several decades. Fluid dynamics has been one of the earliest, and most active fields for the application of numerical techniques. This is due to the essential nonlinearity of most fluid flow problems of practical interest — which makes analytical, or closed-form, solutions virtually impossible to obtain — combined with the geometrical complexity of these problems. In fact, the history of computational fluid dynamics can be traced back virtually to the birth of the digital computer itself, with the pioneering work of John von Neumann and others in this area. Von Neumann was interested in using the computer not only to solve engineering problems, but to understand the fundamental nature of fluid flows themselves. This is possible because the complexity of fluid flows arises, in many instances, not from complicated or poorly understood formulations, but from the nonlinearity of partial differential equations that have been known for more than a century. A famous paragraph written by von Neumann in 1946 serves to illustrate this point. He wrote [Goldstine and von Neumann 1963]:

Indeed, to a great extent, experimentation in fluid mechanics is carried out under conditions where the underlying physical principles are not in doubt, where the quantities to be observed are completely determined by known equations. The purpose of the experiment is not to verify a proposed theory but to replace a computation from an unquestioned theory by direct measurements. Thus, wind tunnels are, for example, used at present, at least in part, as computing devices of the so-called analogy type . . . to integrate the nonlinear partial differential equations of fluid dynamics.

The present article provides some of the basic background in fluid dynamics required to understand the issues involved in numerical solution of fluid flow problems, then outlines the approaches that have been successful in attacking problems of practical interest.

30.2 Underlying Principles

In this section, we will provide the background in fluid dynamics required to understand the principles involved in the numerical solution of the governing equations. The formulation of the equations in generalized, curvilinear coordinates and the geometrical issues involved in the construction of suitable grids also will be discussed.

30.2.1 Fluid-Dynamical Background

As can be inferred from the quotation of von Neumann presented in the introduction, fluid dynamics is fortunate to have a generally accepted mathematical framework for describing most problems of practical interest. Such diverse problems as the high-speed flow past an aircraft wing, the motions of the atmosphere responsible for our weather, and the unsteady air currents produced by the flapping wings of a housefly all can be described as solutions to a set of partial differential equations known as the Navier–Stokes equations. These equations express the physical laws corresponding to conservation of mass, Newton’s second law of motion relating the acceleration of fluid elements to the imposed forces, and conservation of energy, under the assumption that the stresses in the fluid are linearly related to the local rates of strain of the fluid elements. This latter assumption is generally regarded as an excellent approximation for everyday fluids such as water and air — the two most common fluids of engineering and scientific interest.

We will describe the equations for problems in two space dimensions, for the sake of economy of notation; here, and elsewhere, the extension to problems in three dimensions will be straightforward unless otherwise noted. The Navier–Stokes equations can be written in the form

$$\frac{\partial \mathbf{w}}{\partial t} + \frac{\partial \mathbf{f}}{\partial x} + \frac{\partial \mathbf{g}}{\partial y} = \frac{\partial \mathbf{R}}{\partial x} + \frac{\partial \mathbf{S}}{\partial y} \quad (30.1)$$

where \mathbf{w} is the vector of conserved quantities

$$\mathbf{w} = \{\rho, \rho u, \rho v, e\}^T \quad (30.2)$$

where ρ is the fluid density, u and v are the fluid velocity components in the x and y directions, respectively, and e is the total energy per unit volume. The inviscid flux vectors \mathbf{f} and \mathbf{g} are given by

$$\mathbf{f} = \{\rho u, \rho u^2 + p, \rho uv, (e + p)u\}^T \quad (30.3)$$

$$\mathbf{g} = \{\rho v, \rho uv, \rho v^2 + p, (e + p)v\}^T \quad (30.4)$$

where p is the fluid pressure, and the flux vectors describing the effects of viscosity are

$$\mathbf{R} = \{0, \tau_{xx}, \tau_{xy}, u\tau_{xx} + v\tau_{xy} - q_x\}^T \quad (30.5)$$

$$\mathbf{S} = \{0, \tau_{xy}, \tau_{yy}, u\tau_{xy} + v\tau_{yy} - q_y\}^T \quad (30.6)$$

The viscous stresses appearing in these terms are related to the derivatives of the components of the velocity vector by

$$\tau_{xx} = 2\mu \frac{\partial u}{\partial x} - \frac{2}{3}\mu \left\{ \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} \right\} \quad (30.7)$$

$$\tau_{xy} = \mu \left\{ \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right\} \quad (30.8)$$

$$\tau_{yy} = 2\mu \frac{\partial v}{\partial y} - \frac{2}{3}\mu \left\{ \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} \right\} \quad (30.9)$$

and

$$q_x = -k \frac{\partial T}{\partial x} \quad (30.10)$$

$$q_y = -k \frac{\partial T}{\partial y} \quad (30.11)$$

represent the x and y components of the heat flux vector, according to Fourier's law. In these equations μ and k represent the coefficients of viscosity and thermal conductivity, respectively.

If the Navier–Stokes equations are nondimensionalized by normalizing lengths with respect to a representative length L and velocities by a representative velocity V_∞ , and normalizing the fluid properties (such as density and coefficient of viscosity) by their values in the freestream, an important nondimensional parameter, the *Reynolds number*

$$\text{Re} = \frac{\rho_\infty V_\infty L}{\mu_\infty} \quad (30.12)$$

appears as a parameter in the equations. In particular, the viscous stress terms on the right-hand side of Equation 30.1 are multiplied by the reciprocal of the Reynolds number. Physically, the Reynolds number can be interpreted as an inverse measure of the relative importance of the contributions of the viscous stresses to the dynamics of the flow; i.e., when the Reynolds number is large, the viscous stresses are small almost everywhere in the flowfield.

The computational resources required to solve the complete Navier–Stokes equations are enormous, particularly when the Reynolds number is large and regions of **turbulent flow** must be resolved. In 1970 Howard Emmons of Harvard University estimated the computer time required to solve a simple turbulent pipe-flow problem, including direct computation of all turbulent eddies containing significant energy [Emmons 1970]. For a computational domain consisting of approximately 12 diameters of a pipe of circular cross section, the computation of the solution at a Reynolds number based on the pipe diameter of $\text{Re}_d = 10^7$ would require approximately 10^{17} seconds on a 1970s mainframe computer. Of course, much faster computers are now available, but even at a computational speed of 100 gigaflops — i.e., 10^{11} floating-point operations per second — such a calculation would require more than 3000 *years* to complete.

Because the resources required to solve the complete Navier–Stokes equations are so large, it is common to make approximations to bring the required computational resources to a more modest level for problems of practical interest. Expanding slightly on a classification introduced by Chapman [1979], the sequence of fluid-mechanical equations can be organized into the hierarchy shown in Table 30.1.

Table 30.1 summarizes the physical assumptions and time periods of most intense development for each of the stages in the fluid-mechanical hierarchy. Stage IV represents an approximation to the Navier–Stokes equations in which only the largest, presumably least isotropic scales of turbulence are resolved; the subgrid scales are modeled. Stage III represents an approximation in which the solution is decomposed into time-averaged or ensemble-averaged and fluctuating components for each variable. For example, the

TABLE 30.1 The Hierarchy of Fluid-Mechanical Approximations, Following Chapman [1979]

Stage	Model	Equations	Time Frame
I	Linear potential	Laplace's equation	1960s
IIa	Nonlinear potential	Full potential equation	1970s
IIb	Nonlinear inviscid	Euler equations	1980s
III	Modeled turbulence	Reynolds-averaged Navier–Stokes equations	1990s
IV	Large-eddy simulation (LES)	Navier–Stokes equations with subgrid turbulence model	1980s–1990s
V	Direct numerical simulation (DNS)	Fully resolved Navier–Stokes equations	1980s–1990s

velocity components and pressure can be decomposed into

$$u = U + u' \quad (30.13)$$

$$v = V + v' \quad (30.14)$$

$$p = P + p' \quad (30.15)$$

where U , V , and P are the average values of u , v , and p , respectively, taken over a time interval that is long compared to the turbulence time scales, but short compared to the time scales of any nonsteadiness of the averaged flowfield. If we let $\langle u \rangle$ denote such a time average of the u -component of the velocity, then, e.g.,

$$\langle u \rangle = U \quad (30.16)$$

When a decomposition of the form of Equation 30.13 to Equation 30.15 is substituted into the Navier–Stokes equations and the equations are averaged as described above, the resulting equations describe the evolution of the mean-flow quantities (such as U , V , P , etc.). These equations, called the **Reynolds-averaged Navier–Stokes equations**, are nearly identical to the original Navier–Stokes equations written for the mean-flow variables because terms that are linear in $\langle u' \rangle$, $\langle v' \rangle$, $\langle p' \rangle$, etc., are identically zero. The nonlinearity of the equations, however, introduces terms that depend upon the fluctuating components. In particular, terms proportional to $\langle \rho u' v' \rangle$, $\langle \rho u'^2 \rangle$, and $\langle \rho v'^2 \rangle$ appear in the averaged equations. Dimensionally, these terms are equivalent to stresses; in fact, these quantities are called the Reynolds stresses. Physically, the Reynolds stresses are the turbulent counterparts of the molecular viscous stresses, and appear as a result of the transport of momentum by the turbulent fluctuations. The appearance of these terms in the equations describing the mean flow means that the mean flow cannot be determined without some knowledge of these fluctuating components.

In order to solve the Reynolds-averaged Navier–Stokes equations, the Reynolds stresses must be related to the mean flow at some level of approximation using a phenomenological model. The simplest procedure is to try to relate the Reynolds stresses to the local mean-flow properties. Since the turbulence that is responsible for these stresses is a function not only of the local mean-flow state, but also of the flow history as well, such an approximation cannot have broad generality, but such local, or algebraic, turbulence models, based on the analogy of the mixing length from the kinetic theory of gases, are useful for many flows of engineering interest, especially for flows in which boundary-layer separation does not occur.

A more general procedure is to develop partial differential equations for the Reynolds stresses themselves, or for quantities that can be used to determine the scales of the turbulence, such as the turbulent kinetic energy k and the dissipation rate ϵ . The latter, so-called k – ϵ model is widely used in engineering analyses of turbulent flows. These differential equations can be derived by taking higher-order moments of the Navier–Stokes equations. For example, if the x -momentum equation is multiplied by v and then averaged, the result will be an equation for the Reynolds-stress component $\langle \rho u' v' \rangle$. Again because of the nonlinearity of the equations, however, yet higher moments of the fluctuating components (e.g., terms proportional to $\langle \rho u'^2 v' \rangle$, $\langle \rho u' v'^2 \rangle$, etc.) will appear in the equations for the Reynolds stresses. This is an example of the problem of closure of the equations for the Reynolds stresses; i.e., to solve the equations for the Reynolds stresses, these third-order quantities must be known. Equations for the third-order quantities can be derived by taking yet higher-order moments of the Navier–Stokes equations, but these equations will involve fourth-order moments. Thus, at some point the taking of moments must be terminated and models must be developed for the unknown higher-order quantities. It is hoped that more nearly universal models may be developed for these higher-order quantities, but the superiority of second-order models, in which the equations for the Reynolds-stress components are solved, subject to modeling assumptions for the required third-order quantities, has yet to be established for most practical problems.

For many design purposes, especially in aerodynamics, it is sufficient to represent the flow as that of an inviscid, or ideal, fluid. This is appropriate for flows at high Reynolds numbers that contain only negligibly small regions of separated flow. The equations describing inviscid flows can be obtained as a simplification of the Navier–Stokes equations in which the viscous terms are neglected altogether. This results in the

Euler equations of inviscid, compressible flow,

$$\frac{\partial \mathbf{w}}{\partial t} + \frac{\partial \mathbf{f}}{\partial x} + \frac{\partial \mathbf{g}}{\partial y} = 0 \quad (30.17)$$

This approximation corresponds to stage IIb in the hierarchy described in [Table 30.1](#).

The Euler equations constitute a hyperbolic system of partial differential equations, and their solutions contain features that are absent from the Navier–Stokes equations. In particular, while the viscous diffusion terms appearing in the Navier–Stokes equations guarantee that solutions will remain smooth, the absence of these dissipative terms from the Euler equations allows them to have solutions that are discontinuous across surfaces in the flow. Solutions to the Euler equations must be interpreted within the context of generalized (or weak) solutions, and this theory provides the mathematical framework for developing the properties of any discontinuities that may appear. In particular, jumps in dependent variables (such as density, pressure, and velocity) across such surfaces must be consistent with the original conservation laws upon which the differential equations are based. For the Euler equations, these jump conditions are called the Rankine–Hugoniot conditions.

Solutions of the Euler equations for flows containing shock waves can be computed using either **shock fitting** or **shock capturing** methods. In the former, the shock surfaces must be located and the Rankine–Hugoniot jump conditions enforced explicitly. In shock capturing methods, **artificial viscosity** terms are added in the numerical approximation to provide enough dissipation to allow the shocks to be captured automatically by the scheme, with no special treatment in the vicinity of the shock waves. The numerical viscosity terms usually act to smear out the discontinuity over several grid cells. Numerical viscosity also is used when solving the Navier–Stokes equations for flows containing shock waves, because usually it is impractical to resolve the shock structure defined by the physical dissipative mechanisms.

In many cases, the flow can further be approximated as steady and irrotational. In these cases, it is possible to define a velocity potential Φ such that the velocity vector \mathbf{V} is given by

$$\mathbf{V} = \nabla \Phi \quad (30.18)$$

and the fluid density is given by the isentropic relation

$$\rho = \left(1 + \frac{\gamma - 1}{2} \mathbf{M}_\infty^2 [1 - (u^2 + v^2)] \right)^{\frac{1}{\gamma - 1}} \quad (30.19)$$

where

$$u = \frac{\partial \Phi}{\partial x} \quad (30.20)$$

$$v = \frac{\partial \Phi}{\partial y} \quad (30.21)$$

and \mathbf{M}_∞ is a reference Mach number corresponding to the freestream state in which $\rho = 1$ and $u^2 + v^2 = 1$. The steady form of the Euler equations then reduces to the single equation

$$\frac{\partial \rho u}{\partial x} + \frac{\partial \rho v}{\partial y} = 0 \quad (30.22)$$

Equation (30.19) can be used to eliminate the density from Eq. (30.22), which then can be expanded to the form

$$(a^2 - u^2) \frac{\partial^2 \Phi}{\partial x^2} - 2uv \frac{\partial^2 \Phi}{\partial x \partial y} + (a^2 - v^2) \frac{\partial^2 \Phi}{\partial y^2} = 0 \quad (30.23)$$

where a is the local speed of sound, which is a function only of the fluid speed $V = \sqrt{u^2 + v^2}$. Thus, Equation 30.23 is a single equation that can be solved for the velocity potential Φ .

Equation 30.23 is a second-order quasilinear partial differential equation whose type depends on the sign of the discriminant $1 - \mathbf{M}^2$, where \mathbf{M} is the *local* Mach number. The equation is elliptic or hyperbolic according as the Mach number is less than or greater than unity. Thus, the nonlinear potential equation contains a mathematical description of the physics necessary to predict the important features of transonic flows. It is capable of changing type, and the conservation form of Equation 30.22 allows surfaces of discontinuity, or shock waves, to be computed. Solutions at this level of approximation, corresponding to stage IIa, are considerably less expensive to compute than solutions of the full Euler equations, since only one dependent variable need be computed and stored. The jump relation corresponding to weak solutions of the potential equation is different than the Rankine–Hugoniot relations, but is a good approximation to the latter when the shocks are not too strong.

Finally, if the flow can be approximated by small perturbations to some uniform reference state, Equation 30.23 can further be simplified to

$$(1 - \mathbf{M}_\infty^2) \frac{\partial^2 \phi}{\partial x^2} + \frac{\partial^2 \phi}{\partial y^2} = 0 \quad (30.24)$$

where ϕ is the *perturbation* velocity potential defined according to

$$\Phi = x + \phi \quad (30.25)$$

if the uniform velocity in the reference state is assumed to be parallel to the x -axis and be normalized to have unit magnitude. Further, if the flow can be approximated as incompressible, i.e., in the limit as $\mathbf{M}_\infty \rightarrow 0$, Equation 30.23 reduces to

$$\frac{\partial^2 \Phi}{\partial x^2} + \frac{\partial^2 \Phi}{\partial y^2} = 0 \quad (30.26)$$

Since Equation 30.24 and Equation 30.26 are linear, superposition of elementary solutions can be used to construct solutions of arbitrary complexity. Numerical methods to determine the singularity strengths for aerodynamic problems are called **panel methods**, and constitute stage I in the hierarchy of approximations.

It is important to realize that, even though the time periods for development of some of the different models overlap, the applications of the various models may be for problems of significantly differing complexity. For example, DNS calculations were performed as early as the 1970s, but only for the simplest flows — homogeneous turbulence — at very low Reynolds numbers. Flows at higher Reynolds numbers and nonhomogeneous flows are being performed only now, whereas calculations for three-dimensional flows with modeled turbulence were performed as early as the mid 1980s.

30.2.2 Treatment of Geometry

For all numerical solutions to partial differential equations, including those of fluid mechanics, it is necessary to discretize the boundaries of the flow domain. For the panel method of stage I, this is all that is required, since the problem is linear and superposition can be used to construct solutions satisfying the boundary conditions. For nonlinear problems, it is necessary to discretize the entire flow domain. This can be done using either structured or unstructured grid systems. *Structured grids* are those in which the grid points can be ordered in a regular Cartesian structure; i.e., the points can be given indices (i, j) such that the nearest neighbors of the (i, j) point are identified by the indices $(i \pm 1, j \pm 1)$. The grid cells for these meshes are thus quadrilateral in two dimensions and hexahedral in three dimensions. *Unstructured grids* have no regular ordering of points or cells, and a connectivity table must be maintained to identify which points and edges belong to which cells. Unstructured grids most often consist of triangles (in two dimensions), tetrahedra (in three dimensions), or combinations of these and quadrilateral and hexahedral cells, respectively. In addition, grids having purely quadrilateral cells may also be unstructured, even though they have a locally Cartesian structure — e.g., when multilevel grids are used for adaptive refinement.

Implementations on structured grids are generally more efficient than those on unstructured grids, since indirect addressing is required for the latter, and efficient implicit methods often can be constructed that take advantage of the regular ordering of points (or cells) in structured grids. A great deal of effort has been expended to generate both structured and unstructured grid systems, much of which is closely related to the field of computational geometry.

30.2.2.1 Structured Grids

A variety of techniques are used to generate structured grids for use in fluid-mechanical calculations. These include relatively fast algebraic methods, including those based on transfinite interpolation and conformal mapping, as well as more costly methods based on the solution of either elliptic or hyperbolic systems of partial differential equations for the grid coordinates. These techniques are discussed in a review article by Eiseman [1985].

For complex geometries, it often is not possible to generate a single grid that conforms to all the boundaries. Even if it is possible to generate such a grid, it may have undesirable properties, such as excessive skewness or a poor distribution of cells, which could lead to poor stability or accuracy in the numerical algorithm. Thus, structured grids for complex geometries are generally constructed as combinations of simpler grid blocks for various parts of the domain. These grids may be allowed to overlap, in which case they are referred to as *Chimera* grids, or be required to share common surfaces of intersection, in which case they are identified as *multiblock* grids. In the latter case, grid lines may have varying degrees of continuity across the interblock boundaries, and these variations have implications for the construction and behavior of the numerical algorithm in the vicinity of those boundaries. Grid generation techniques based on the solution of systems of partial differential equations are described by Thompson et al. [1985].

Numerical methods to solve the equations of fluid mechanics on structured grid systems are implemented most efficiently by taking advantage of the ease with which the equations can be transformed to a generalized coordinate system. The expression of the system of conservation laws in the new, body-fitted coordinate system reduces the problem to one effectively of Cartesian geometry. The transformation will be described here for the Euler equations.

Consider the transformation of Equation 30.17 to a new coordinate system (ξ, η) . The local properties of the transformation at any point are contained in the Jacobian matrix of the transformation, which can be defined as

$$J = \begin{Bmatrix} x_\xi & x_\eta \\ y_\xi & y_\eta \end{Bmatrix} \quad (30.27)$$

for which the inverse is given by

$$J^{-1} = \begin{Bmatrix} \xi_x & \xi_y \\ \eta_x & \eta_y \end{Bmatrix} = \frac{1}{h} \begin{Bmatrix} y_\eta & -x_\eta \\ -y_\xi & x_\xi \end{Bmatrix} \quad (30.28)$$

where $h = x_\xi y_\eta - y_\xi x_\eta$ is the determinant of the Jacobian matrix. Subscripts in Equation 30.27 and Equation 30.28 denote partial differentiation.

It is natural to express the fluxes in conservation laws in terms of their contravariant components. Thus, if we define

$$\{\mathbf{F}, \mathbf{G}\}^T = J^{-1} \{\mathbf{f}, \mathbf{g}\}^T \quad (30.29)$$

then the transformed Euler equations can be written in the compact form

$$\frac{\partial h\mathbf{w}}{\partial t} + \frac{\partial h\mathbf{F}}{\partial \xi} + \frac{\partial h\mathbf{G}}{\partial \eta} = 0 \quad (30.30)$$

if the transformation is independent of time (i.e., if the grid is not moving). If the grid is moving or deforming with time, the equations can be written in a similar form, but additional terms must be included that allow for the fluxes induced by the motion of the mesh.

The Navier–Stokes equations can be transformed in a similar manner, although the transformation of the viscous contributions is somewhat more complicated and will not be included here. Since the nonlinear potential equation is simply the continuity equation (the first of the equations that comprise the Euler equations), the transformed potential equation can be written as

$$\frac{\partial}{\partial \xi}(\rho h U) + \frac{\partial}{\partial \eta}(\rho h V) = 0 \quad (30.31)$$

where

$$\begin{Bmatrix} U \\ V \end{Bmatrix} = J^{-1} \begin{Bmatrix} u \\ v \end{Bmatrix} \quad (30.32)$$

are the contravariant components of the velocity vector.

30.2.2.2 Unstructured Grids

Unstructured grids generally have greater geometric flexibility than structured grids, because of the relative ease of generating triangular or tetrahedral tessellations of two- and three-dimensional domains. Advancing-front methods and Delaunay triangulations are the most frequently used techniques for generating triangular/tetrahedral grids. Unstructured grids also are easier to adapt locally so as to better resolve localized features of the solution.

30.3 Best Practices

30.3.1 Panel Methods

The earliest numerical methods used widely for making fluid-dynamical computations were developed to solve linear potential problems, described as stage I calculations in the previous section. Mathematically, panel methods are based upon the fact that Equation 30.26 can be recast as an integral equation giving the solution at any point (x, y) in terms of the freestream speed U (here assumed unity) and angle of attack α and the line integrals of singularities distributed along the curve \mathcal{C} representing the body surface:

$$\Phi(x, y) = x \cos \alpha + y \sin \alpha + \int_{\mathcal{C}} \sigma \ln \left(\frac{r}{2\pi} \right) ds + \int_{\mathcal{C}} \delta (\partial / \partial n) \ln \left(\frac{r}{2\pi} \right) ds \quad (30.33)$$

In this equation, σ and δ represent the source and doublet strengths distributed along the body contour, respectively, r is the distance from the point (x, y) to the boundary point, and n is the direction of the outward normal to the body surface. When the point (x, y) is chosen to lie on the body contour \mathcal{C} , Equation 30.33 can be interpreted as giving the solution Φ at any point on the body in terms of the singularities distributed along the surface. This effectively reduces the dimension of the problem by one (i.e., the two-dimensional problem considered here becomes essentially one-dimensional, and the analogous procedure applied to a three-dimensional problem results in a two-dimensional equation requiring the evaluation only of integrals over the body surface).

Equation 30.33 is approximated numerically by discretizing the boundary \mathcal{C} into a collection of panels (line segments in this two-dimensional example) on which the singularity distribution is assumed to be of some known functional form, but of an as yet unknown magnitude. For example, for a simple nonlifting body, the doublet strength δ might be assumed to be zero, while the source strength σ is assumed to be constant on each segment. The second integral in Equation 30.33 is then zero, while the first integral can be written as a sum over all the elements of integrals that can be evaluated analytically as

$$\Phi(x, y) = x \cos \alpha + y \sin \alpha + \sum_{i=1}^N \sigma_i \int_{\mathcal{C}_i} \ln \left(\frac{r}{2\pi} \right) ds \quad (30.34)$$

where C_i is the portion of the body surface corresponding to the i th segment and N is the number of segments, or panels, used.

The source strengths σ_i must be determined by enforcing the boundary condition

$$\frac{\partial \Phi}{\partial n} = 0 \quad (30.35)$$

which specifies that the component of velocity normal to the surface be zero (i.e., that there be no flux of fluid across the surface). This is implemented by requiring that Equation 30.35 be satisfied at a selected number of control points. For the example of constant-strength source panels, if one control point is chosen on each panel, the requirement that Equation 30.35 be satisfied at each of the control points will result in N equations of the form

$$\sum_{i=1}^N A_{i,j} \sigma_i = \mathbf{U} \cdot \hat{\mathbf{n}}, \quad j = 1, 2, \dots, N \quad (30.36)$$

where $A_{i,j}$ are the elements of the influence-coefficient matrix that give the normal velocity at control point j due to sources of unit strength on panel i , and \mathbf{U} is a unit vector in the direction of the freestream. Equation 30.35 and Equation 30.36 constitute a system of N linear equations that can be solved for the unknown source strengths σ_i . Once the source strengths have been determined, the velocity potential, or the velocity itself, can be computed directly at any point in the flowfield using Equation 30.34. A review of the development and application of panel methods is provided by Hess [1990].

A major advantage of panel methods, relative to the more advanced methods required to solve the nonlinear problems of stages II–V, is that it is necessary to describe (and to discretize into panels) only the surface of the body. While linearity is a great advantage in this regard, it is not clear that the method is computationally more efficient than the more advanced nonlinear field methods. This results from the fact that the influence-coefficient matrix in the system of equations that must be solved to give the source strengths for each panel is not sparse; i.e., the velocities at each control point are affected by the sources on all the panels. In contrast, the solution at each mesh point in a finite-difference formulation (or each mesh cell in a finite-volume formulation) is related to the values of the solution at only a few neighbors, resulting in a very sparse matrix of influence coefficients that can be solved very efficiently using iterative methods. Thus, the primary advantage of the panel method is the geometrical one associated with the reduction in dimension of the problem. For nonlinear problems the use of finite-difference, finite-element, or finite-volume methods requires discretization of the entire flowfield, and the associated mesh generation task has been a major pacing item limiting the application of such methods.

30.3.2 Nonlinear Methods

For nonlinear equations, superposition of elementary solutions is no longer a valid technique for constructing solutions, and it becomes necessary to discretize the solution throughout the entire domain, not just the boundary surface. In addition, since the equations are nonlinear, some sort of iteration must be used to compute successively better approximations to the solution. This iterative process may approximate the physics of an unsteady flow process or may be chosen to provide more rapid convergence to the solution for steady-state problems. Solutions for both the nonlinear potential and Euler equations are generally determined using a finite-difference, finite-volume, or finite-element method. In any of these techniques a grid, or network of points and cells, is distributed throughout the flowfield. In a finite-difference method the derivatives appearing in the original differential equation are approximated by discrete differences in the values of the solution at neighboring points (and times, if the solution is unsteady), and substitution of these into the differential equation yields a system of algebraic equations relating the values of the solution at neighboring grid points. In a finite-volume method, the unknowns are taken to be representative of the values of the solution in the control volumes formed by the intersecting grid surfaces, and the equations are constructed by balancing fluxes across the bounding surfaces of each control volume with the rate of change of the solution within the control volume. In a finite-element method, the solution is represented

using simple interpolating functions within each mesh cell, or element, and equations for the nodal values are obtained by integrating a variational or residual formulation of the equations over the elements. The algebraic equations relating the values of the solution in neighboring cells can be very similar (or even identical) in appearance for all three methods, and the choice of method often is primarily a matter of taste. Stable and efficient finite-difference and finite-volume methods were developed earlier than finite-element methods for compressible flow problems, but finite-element methods capable of treating very complex compressible flows now are available. A review of recent progress in the development of finite-element methods for compressible flows and remaining issues is given by Glowinski and Pironneau [1992].

Since the nonlinear potential, Euler, and Navier–Stokes equations all are nonlinear, the algebraic equations resulting from these discretization procedures also are nonlinear, and a scheme of successive approximation usually is required to solve them. As mentioned earlier, however, these equations tend to be highly local in nature, and efficient iterative methods have been developed to solve them in many cases.

30.3.2.1 Nonlinear Potential-Equation Methods

The primary advantage of solving the potential equation rather than the Euler (or Navier–Stokes) equations derives from the fact that the flowfield can be described completely in terms of a single scalar function, the velocity potential Φ . The formulation of numerical schemes to solve Equation 30.22 is complicated by the fact that, as noted earlier, the equation changes type according to whether the local Mach number is subsonic or supersonic. Differencing schemes for the potential equation must, therefore, be type-dependent — i.e., they must change their form depending on whether the local Mach number is less than or greater than unity. These methods usually are based upon central, or symmetric, differencing formulas that are appropriate for the elliptic case (corresponding to subsonic flows); they are then modified by adding an upwind bias to maintain stability in hyperbolic regions (where the flow is supersonic). This directional bias can be introduced into the difference equations either by adding an artificial viscosity proportional to the third derivative of the velocity potential Φ in the streamwise direction, or by replacing the density at each point in supersonic zones with its value at a point slightly upstream of the actual point. Mathematically, these two approaches can be seen to be equivalent, since the upwinding of the density evaluation also effectively introduces a correction proportional to the third derivative of the potential in the flow direction.

It is important to introduce such artificial viscosity (or compressibility) terms in such a way that their effect vanishes in the limit as the mesh is refined. In this way, the numerical approximation will approach the differential equation in the limit of zero mesh spacing, and the method is said to be consistent with the original differential equation. In addition, for flows with shock waves, it is important that the numerical approximation be **conservative**; this guarantees that the properties of discontinuous solutions will be consistent with the jump relations of the original conservation laws. The shock jump relations corresponding to Equation 30.22, however, are different from the Rankine–Hugoniot conditions describing shocks within the framework of the Euler equations. Since entropy is everywhere conserved in the potential theory, and since there is a finite entropy jump across a Rankine–Hugoniot shock, it is clear that the jump relations must be different. For weak shocks, however, the differences are small and the economies afforded by the potential formulation make computations based on this approximation attractive for many transonic problems.

Perturbation techniques can be used to demonstrate that the effect of entropy jump across the shocks is more important than the rotationality introduced by these weak shocks. Thus, it makes sense to develop techniques that allow for the entropy jump, but are still within the potential formulation. Such techniques have been developed (see, e.g., Hafez [1985]), but have been relatively little used, as developments in techniques to solve the Euler equations have overtaken these approaches.

The nonlinear difference equations resulting from discrete approximations to the potential equation generally are solved using iterative, or relaxation, techniques. The equations are linearized by computing approximations to all but the highest (second) derivatives from the preceding solution in an iterative sequence, and a correction is computed at each mesh point in such a way that the equations are identically satisfied. It is useful in developing these iterative techniques to think of the iterative process as a discrete approximation to a continuous time-dependent process [Garabedian 1956]. Thus, the iterative process

approximates an equation of the form

$$\beta_0 \frac{\partial \Phi}{\partial t} + \beta_1 \frac{\partial^2 \Phi}{\partial \xi \partial t} + \beta_2 \frac{\partial^2 \Phi}{\partial \eta \partial t} = \frac{a^2}{\rho} \left\{ \frac{\partial}{\partial \xi} (\rho h U) + \frac{\partial}{\partial \eta} (\rho h V) \right\} \quad (30.37)$$

The parameters β_0 , β_1 , and β_2 , which are related to the mix of old and updated values of the solution used in the difference equations, can then be chosen to ensure that the time-dependent process converges to a steady state in both subsonic and supersonic regions. The formulation of transonic potential flow problems and their solution is described by Caughey [1982].

Even when the values of the parameters are chosen to provide rapid convergence, many hundreds of iterations may be necessary to achieve convergence to acceptable levels, especially when the mesh is very fine. This slow convergence is a characteristic of virtually all iterative schemes, and is a result of the fact that the representation of the difference equations must be highly local if the scheme is to be computationally efficient. As a result of this locality, the reduction of the low-wave-number component of the error to acceptable levels often requires many iterations. A powerful technique for circumventing this difficulty with the iterative solution of numerical approximations to partial differential equations, called the multigrid technique, has been applied with great success to problems in fluid mechanics.

The multigrid method relies for its success on the fact that after a few cycles of any good iterative technique, the error remaining in the solution is relatively smooth, and can be represented accurately on a coarser grid. Application of the same iterative technique on the coarser grid soon makes the error on this grid smooth as well, and the grid-coarsening process can be repeated until the grid contains only a few cells in each coordinate direction. The corrections that have been computed on all coarser levels are then interpolated back to the finest grid, and the process is repeated. The accuracy of the converged solution on the fine grid is determined by the accuracy of the approximation on that grid, since the coarser levels are used only to effect a more rapid convergence of the iterative process.

A particularly efficient multigrid technique for steady transonic potential-flow problems has been developed by Jameson [1979]. It uses a generalized alternating-direction implicit smoothing algorithm in conjunction with a full-approximation-scheme multigrid algorithm. In theory, for a wide class of problems, the work (per mesh point) required to solve the equations using a multigrid approach is independent of the number of mesh cells. In many practical calculations, 10 or fewer multigrid cycles may be required even on very fine grids.

30.3.2.2 Euler-Equation Methods

As noted earlier, the Euler equations constitute a hyperbolic system of partial differential equations, and numerical methods for solving them rely heavily upon the rather well-developed mathematical theory of such systems. As for the nonlinear potential equation, discontinuous solutions corresponding to flows with shock waves play an important role. Shock-capturing methods are much more widely used than shock-fitting methods, and for these methods it is important to use schemes that are conservative.

As mentioned earlier, it is necessary to add artificial, or numerical, dissipation to stabilize the Euler equations. This can be done by adding dissipative terms explicitly to an otherwise nondissipative central difference scheme or by introducing upwind approximations in the flux evaluations. Both approaches are highly developed. The most widely used central difference methods are those modeled after the approach of Jameson et al. [1981]. This approach introduces an adaptive blend of second and fourth differences of the solution in each coordinate direction; a local switching function, usually based on a second difference of the pressure, is used to reduce the order of accuracy of the approximation to first order locally in the vicinity of shock waves and to turn off the fourth differences there. The second-order terms are small in smooth regions where the fourth-difference terms are sufficient to stabilize the solution and ensure convergence to the steady state. More recently, Jameson [1992] has developed improved symmetric limited positive (SLIP) and upstream limited positive (USLIP) versions of these blended schemes (see also Tatsumi et al. [1995]).

Much effort has been directed toward developing numerical approximations for the Euler equations that capture discontinuous solutions as sharply as possible without overshoots in the vicinity of the discontinuity. For purposes of exposition of these methods, we consider the one-dimensional form of the

Euler equations, which can be written

$$\frac{\partial \mathbf{w}}{\partial t} + \frac{\partial \mathbf{f}}{\partial x} = 0 \quad (30.38)$$

where $\mathbf{w} = \{\rho, \rho u, e\}^T$ and $\mathbf{f} = \{\rho u, \rho u^2 + p, (e + p)u\}^T$. Not only is the exposition clearer for the one-dimensional form of the equations, but the implementation of these schemes for multidimensional problems also generally is done by dimensional splitting in which one-dimensional operators are used to treat variations in each of the mesh directions.

For smooth solutions, Equation 30.38 are equivalent to the quasilinear form

$$\frac{\partial \mathbf{w}}{\partial t} + \mathbf{A} \frac{\partial \mathbf{w}}{\partial x} = 0 \quad (30.39)$$

where $\mathbf{A} = \{\partial \mathbf{f} / \partial \mathbf{w}\}$ is the Jacobian of the flux vector with respect to the solution vector. The eigenvalues of \mathbf{A} are given by $\lambda = u, u + a, u - a$, where $a = \sqrt{\gamma p / \rho}$ is the speed of sound. Thus, for subsonic flows, one of the eigenvalues will have a different sign than the other two. For example, if $0 < u < a$, then $u - a < 0 < u < u + a$. The fact that various eigenvalues have different signs in subsonic flows means that simple one-sided difference methods cannot be stable. One way around this difficulty is to split the flux vector into two parts, the Jacobian of each of which has eigenvalues of only one sign. Such an approach has been developed by Steger and Warming [1981]. They used a relatively simple splitting that has discontinuous derivatives whenever an eigenvalue changes sign; an improved splitting has been developed by van Leer [1982] that has smooth derivatives at the transition points.

Each of the characteristic speeds can be identified with the propagation of a wave. If a mesh surface is considered to represent a discontinuity between two constant states, these waves constitute the solution to a Riemann (or shock-tube) problem. A scheme developed by Godunov [1959] assumes the solution to be piecewise constant over each mesh cell, and uses the fact that the solution to the Riemann problem can be given in terms of the solution of algebraic (but nonlinear) equations to advance the solution in time. Because of the assumption of piecewise constancy of the solution, Godunov's scheme is only first-order accurate. Van Leer [1979] has shown how it is possible to extend these ideas to a second-order monotonicity-preserving scheme using the so-called monotone upwind scheme for systems of conservation laws (MUSCL) formulation. The efficiency of schemes requiring the solution of Riemann problems at each cell interface for each time step can be improved by the use of approximate solutions to the Riemann problem [Roe 1986].

More recent ideas to control oscillation of the solution in the vicinity of shock waves include the concept of total-variation-diminishing (TVD) schemes, first introduced by Harten (see, e.g., Harten [1983, 1984]), and essentially nonoscillatory (ENO) schemes, introduced by Osher and his coworkers (see, e.g., Harten et al. [1987] and Shu and Osher [1988]).

Hyperbolic systems describe the evolution in time of physical systems undergoing unsteady processes governed by the propagation of waves. This feature frequently is used in fluid mechanics, even when the flow to be studied is steady. In this case, the unsteady equations are solved for long enough times that the steady state is reached asymptotically — often to within roundoff error. To maintain the hyperbolic character of the equations, and to keep the numerical method consistent with the physics it is trying to predict, it is necessary to determine the solution at a number of intermediate time levels between the initial state and the final steady state. Such a sequential process is said to be a time marching of the solution.

The simplest practical methods for solving hyperbolic systems are **explicit** in time. The size of the time step that can be used to solve hyperbolic systems using an explicit method is limited by a constraint known as the Courant–Friedrichs–Lewy or **CFL condition**. Broadly interpreted, the CFL condition states that the time step must be smaller than the time required for a signal to propagate across a single mesh cell. Thus, if the mesh is very fine, the allowable time step also must be very small, with the result that many time steps must be taken to reach an asymptotic steady state.

Multistage, or Runge–Kutta, methods have become extremely popular for use as explicit time-stepping schemes. After discretization of the spatial operators, using finite-difference, finite-volume, or finite-element approximations, the Euler equations can be written in the form

$$\frac{d\mathbf{w}_i}{dt} + Q(\mathbf{w}_i) + D(\mathbf{w}_i) = 0 \quad (30.40)$$

where \mathbf{w}_i represents the solution at the i th mesh point, or in the i th mesh cell, and Q and D are discrete operators representing the contributions of the Euler fluxes and numerical dissipation, respectively. An m -stage time integration scheme for these equations can be written in the form

$$\mathbf{w}_i^{(k)} = \mathbf{w}_i^{(0)} - \alpha_k \Delta t \left[Q(\mathbf{w}_i^{(k-1)}) + D(\mathbf{w}_i^{(k-1)}) \right], \quad k = 1, 2, \dots, m \quad (30.41)$$

with $\mathbf{w}_i^{(0)} = \mathbf{w}_i^n$, $\mathbf{w}_i^{(m)} = \mathbf{w}_i^{n+1}$, and $\alpha_m = 1$. The dissipative and dispersive properties of the scheme can be tailored by the sequence of α_i chosen; note that, for nonlinear problems, this method may be only first-order accurate in time, but this is not necessarily a disadvantage if one is interested only in the steady-state solution. The principal advantage of this formulation, relative to versions that may have better time accuracy, is that only two levels of storage are required regardless of the number of stages used. This approach was first introduced for problems in fluid mechanics by Graves and Johnson [1978], and has been further developed by Jameson et al. [1981]. In particular, Jameson and his group have shown how to tailor the stage coefficients so that the method is an effective smoothing algorithm for use with multigrid (see, e.g., Jameson and Baker [1983]).

Implicit techniques also are highly developed, especially when structured grids are used. Approximate factorization of the implicit operator usually is required to reduce the computational burden of solving a system of linear equations for each time step. Methods based on alternating-direction implicit (ADI) techniques date back to the pioneering work of Briley and McDonald [1974] and Beam and Warming [1976]. An efficient diagonalized ADI method has been developed within the context of the multigrid method by Caughey [1987], and a lower-upper symmetric Gauss–Seidel method has been developed by Yoon and Kwak [1994]. The multigrid implementations of these methods are based on the work of Jameson [1983].

30.3.3 Navier–Stokes Equation Methods

As described earlier, the relative importance of viscous effects is characterized by the value of the Reynolds number. If the Reynolds number is not too large, the flow remains smooth, and adjacent layers (or laminae) of fluid slide smoothly past one another. When this is the case, the solution of the Navier–Stokes equations is not too much more difficult than solution of the Euler equations. Greater resolution is required to resolve the large gradients in the boundary layers near solid boundaries, especially as the Reynolds number becomes large, so more mesh cells are required to achieve acceptable accuracy. In most of the flowfield, however, the flow behaves as if it were nearly inviscid, so methods developed for the Euler equations are appropriate and effective. The equations must, of course, be modified to include the additional terms resulting from the viscous stresses, and care must be taken to ensure that any artificial dissipative effects are small relative to the physical viscous dissipation in regions where the latter is important. The solution of the Navier–Stokes equations for laminar flows, then, is somewhat more costly in terms of computer resources, but not significantly more difficult from an algorithmic point of view, than solution of the Euler equations. Unfortunately, most flows of engineering interest occur at large enough Reynolds numbers that the flow in the boundary layers near solid boundaries becomes turbulent.

30.3.3.1 Turbulence Models

Solution of the Reynolds-averaged Navier–Stokes equations requires modeling of the Reynolds stress terms. The simplest models, based on the original mixing-length hypothesis of Prandtl, relate the Reynolds stresses to the local properties of the mean flow. The Baldwin–Lomax model [Baldwin and Lomax 1978] is the

most widely used model of this type and gives good correlation with experimental measurements for wall-bounded shear layers so long as there are no significant regions of separated flow.

The most complete commonly used turbulence models include two additional partial differential equations that determine characteristic length and time scales for the turbulence. The most widely used of these techniques is called the k - ϵ model, since it is based on equations for the turbulence kinetic energy (usually given the symbol k) and the turbulence dissipation rate (usually given the symbol ϵ). This method has grown out of work by Launder and Spaulding [1972]. Another variant, based on a formulation by Kolmogorov, calculates a turbulence frequency ω instead of the dissipation rate (and hence is called a k - ω model). More complete models that compute all elements of the Reynolds stress tensor are in an active state of development. The common base for most of these models is the work of Launder et al. [1975], with more recent contributions by Lumley [1978], Speziale [1987], and Reynolds [1987]. These models, and their implementation within the context of CFD, are described in the book by Wilcox [1993].

More limited models, based on a single equation for a turbulence scale, also have been developed. These include the models of Baldwin and Barth [1991] and of Spalart and Allmaras [1992]. These models are applicable principally to boundary-layer flows of aerodynamic interest.

30.3.3.2 Large-Eddy Simulations

The difficulty of developing generally applicable phenomenological turbulence models on the one hand, and the enormous computational resources required to resolve all scales in turbulent flows at large Reynolds number on the other, have led to the development of large-eddy simulation (LES) techniques. In this approach, the largest length and time scales of the turbulent motions are resolved, but the smaller (subgrid) scales are modeled. This is an attractive approach because it is the largest scales that contain the preponderance of turbulent kinetic energy and that are responsible for most of the mixing. At the same time, the smaller scales are believed to be more nearly isotropic and independent of the larger scales, and thus are less likely to behave in problem-specific ways — i.e., it should be easier to develop universal models for these smaller scales.

A filtering technique is applied to the Navier–Stokes equations which results in equations having a form similar to the original equations, but with additional terms representing a subgrid-scale tensor that describes the effect of the modeled terms on the larger scales. The solution of these equations is not well posed if there is no initial knowledge of the subgrid scales; the correct statistics are predicted for the flow, but it is impossible to reproduce a specific realization of the flow without this detailed initial condition. Fortunately, for most engineering problems, it is only the statistics that are of interest.

LES techniques date back to the pioneering work of Smagorinsky [1963], who developed an eddy-viscosity subgrid model for use in meteorological problems. The model turned out to be too dissipative for large-scale meteorological problems in which large-scale, predominantly two-dimensional motions are affected by three-dimensional turbulence. Smagorinsky's model finds wide application in engineering problems, however. Details of the LES approach are discussed by Rogallo and Moin [1984], and more recent developments and applications are described by Lesieur and Métais [1996].

30.3.3.3 Direct Numerical Simulations

Direct numerical simulations generally use spectral or pseudospectral approximations for the spatial discretization of the Navier–Stokes equations (see, e.g., Gottlieb and Orszag [1977] or Hussaini and Zang [1987]). The difference between spectral and pseudospectral methods is in the way that products are computed; the advantage of spectral methods, which are more expensive computationally, is that aliasing errors are removed exactly [Orszag 1972].

A description of the issues involved and some results are given by Rogallo and Moin [1984]. Direct numerical simulations are particularly valuable for the insight that they provide into the fundamental nature of turbulent flows. The computational resources required for DNS calculations grow so rapidly with increasing Reynolds number that they are unlikely to be directly useful for engineering predictions, but they will remain an invaluable tool providing insight needed to construct better turbulence models for use in LES and with the Reynolds-averaged equations.

30.4 Research Issues and Summary

Computational fluid dynamics continues to be a field of intense research activity. The development of accurate algorithms based on unstructured grids for problems involving complex geometries, and the increasing application of CFD techniques to unsteady problems, including aeroelasticity and acoustics, are examples of areas of great current interest. Algorithmically, there continues to be fruitful work on the incorporation of adaptive grids that automatically increase resolution in regions where it is required to maintain accuracy, and on the development of inherently multidimensional high-resolution schemes. Finally, the continued expansion of computational capability allows the application of DNS and LES methods to problems of higher Reynolds number and increasingly realistic flow geometries.

Defining Terms

Artificial viscosity: Terms added to a numerical approximation that provide artificial — i.e., nonphysical — dissipative mechanisms to stabilize a solution or to allow shock waves to be captured automatically by the numerical scheme.

CFL condition: The Courant–Friedrichs–Lewy (CFL) condition is a stability criterion that limits the time step of an explicit time-marching scheme for hyperbolic systems of differential equations. In the simplest one-dimensional case, if Δx is the physical mesh spacing and $\rho(\mathbf{A})$ is the spectral radius of the Jacobian matrix \mathbf{A} corresponding to the fastest wave speed for the problem, then the time step for explicit schemes must satisfy the constraint $\Delta t \leq K \Delta x / \rho(\mathbf{A})$, where K is a constant. For the simplest explicit schemes, $K = 1$, which implies that the time step must be less than that required for the fastest wave to cross the cell.

Conservative: A numerical approximation is said to be conservative if it is based on the conservation (or divergence) form of the differential equations, and the net flux across a cell interface is the same when computed from either direction; in this way the properties of discontinuous solutions are guaranteed to be consistent with the jump relations for the original integral form of the conservation laws.

Direct numerical simulation (DNS): A solution of the complete Navier–Stokes equations in which all length and time scales, down to those describing the smallest eddies containing significant turbulent kinetic energy, are fully resolved.

Euler equations: The equations describing the inviscid flow of a compressible fluid. These equations constitute a hyperbolic system of differential equations; the Euler equations are nondissipative, and weak solutions containing discontinuities (which can be viewed as approximations to shock waves) must be allowed for many practical problems.

Explicit method: A method in which the solution at each point for the new time level is given explicitly in terms of values of the solution at the previous time level; to be contrasted with an *implicit method*, in which the solution at each point at the new time level also depends on the solution at one or more neighboring points at the new time level, so that an algebraic system of equations must be solved at each time step.

Finite-difference method: A numerical method in which the solution is computed at a finite number of points in the domain; the solution is determined from equations that relate the solution at each point to its values at selected neighboring points.

Finite-element method: A numerical method for solving partial differential equations in which the solution is approximated by simple functions within each of a number of small elements into which the domain has been divided.

Finite-volume method: A numerical method for solving partial differential equations, especially those arising from systems of conservation laws, in which the rate of change of quantities within each mesh volume is related to fluxes across the boundaries of the volume.

Implicit method: See **explicit method**.

Large-eddy simulation: A numerical solution of the Navier–Stokes equations in which the largest, energy-carrying eddies are completely resolved, but the effects of the smaller eddies, which are more nearly isotropic, are accounted for by a subgrid model.

- Mach number:** The ratio $M = V/a$ of the fluid velocity V to the speed of sound a . This nondimensional parameter characterizes the importance of compressibility in the dynamics of the fluid motion.
- Mesh generation:** The generation of mesh systems suitable for the accurate representation of solutions to partial differential equations.
- Panel method:** A numerical method to solve Laplace's equation for the velocity potential of a fluid flow. The boundary of the flow domain is discretized into a set of nonoverlapping facets, or panels, on each of which the strength of some elementary solution is assumed constant. Equations for the normal velocity at control points on each panel can be solved for the unknown singularity strengths to give the solution. In some disciplines this approach is called the *boundary integral element method* (BIEM).
- Reynolds-averaged Navier–Stokes equations:** Equations for the mean quantities in a turbulent flow obtained by decomposing the fields into mean and fluctuating components and averaging the Navier–Stokes equations. Solution of these equations for the mean properties of the flow requires knowledge of various correlations (the Reynolds stresses), of the fluctuating components.
- Shock capturing:** A numerical method in which shock waves are treated by smearing them out with artificial dissipative terms in a manner such that no special treatment is required in the vicinity of the shocks; to be contrasted with *shock fitting* methods in which shock waves are treated as discontinuities with the jump conditions explicitly enforced across them.
- Shock fitting:** See **shock capturing**.
- Shock wave:** Region in a compressible flow across which the flow properties change almost discontinuously; unless the density of the fluid is extremely small, the shock region is so thin relative to other significant dimensions in most practical problems that it is a good approximation to represent the shock as a surface of discontinuity.
- Turbulent flow:** Flow in which unsteady fluctuations play a major role in determining the effective mean stresses in the field; regions in which turbulent fluctuations are important inevitably appear in fluid flow at large Reynolds numbers.
- Upwind method:** A numerical method for CFD in which upwinding of the difference stencil is used to introduce dissipation into the approximation, thus stabilizing the scheme. This is a popular mechanism for the Euler equations, which have no natural dissipation, but is also used for Navier–Stokes algorithms, especially those designed to be used at high Reynolds number.
- Visualization:** The use of computer graphics to display features of solutions to CFD problems.

References

- Abid, R., Vatsa, V. N., Johnson, D. A., and Wedan, B. W. 1990. Prediction of separated transonic wing flows with non-equilibrium algebraic turbulence models. *AIAA J.* 28:1426–1431.
- Baldwin, B. S. and Barth, T. J. 1991. A one-equation turbulence transport model for high Reynolds number wall-bounded flows. *AIAA Paper* 91-0610, 29th Aerospace Sciences Meeting, Reno, NV.
- Baldwin, B. S. and Lomax, H. 1978. Thin layer approximation and algebraic model for separated turbulent flows. *AIAA Paper* 78-257, 16th Aerospace Sciences Meeting, Huntsville, AL.
- Beam, R. M. and Warming, R. F. 1976. An implicit finite-difference algorithm for hyperbolic systems in conservation law form. *J. Comput. Phys.* 22:87–110.
- Briley, W. R. and McDonald, H. 1974. Solution of the three-dimensional compressible Navier–Stokes equations by an implicit technique. In *Lecture Notes in Physics*, Vol. 35, pp. 105–110. Springer-Verlag, New York.
- Caughey, D. A. 1982. The computation of transonic potential flows. *Ann. Rev. Fluid Mech.* 14:261–283.
- Caughey, D. A. 1987. Diagonal implicit multigrid solution of the Euler equations. *AIAA J.* 26:841–851.
- Chapman, D. R. 1979. Computational aerodynamics: review and outlook. *AIAA J.* 17:1293–1313.
- Eiseman, P. R. 1985. Grid generation for fluid mechanics computations. *Ann. Rev. Fluid Mech.* 17:487–522.
- Emmons, H. W. 1970. Critique of numerical modeling of fluid-mechanics phenomena. *Ann. Rev. Fluid Mech.* 2:15–36.

- Garabedian, P. R. 1956. Estimation of the relaxation factor for small mesh size. *Math. Tables Aids Comput.* 10:183–185.
- Glowinski, R. and Pironneau, O. 1992. Finite element methods for Navier–Stokes equations. *Ann. Rev. Fluid Mech.* 24:167–204.
- Godunov, S. K. 1959. A finite-difference method for the numerical computation of discontinuous solutions of the equations of fluid dynamics. *Mat. Sb.* 47:357–393.
- Goldstine, H. H. and von Neumann, J. 1963. On the principles of large scale computing machines. In *John von Neumann, Collected Works*, A. H. Taub, ed. Vol. 5, p. 4. Pergamon Press, New York.
- Gottlieb, D. and Orszag, S. A. 1977. *Numerical Analysis of Spectral Methods: Theory and Application*, CBMS-NSF Reg. Conf. Ser. Appl. Math. 26. SIAM, Philadelphia.
- Graves, R. A. and Johnson, N. E. 1978. Navier–Stokes solutions using Stetter’s method. *AIAA J.* 16:1013–1015.
- Hafez, M. M. 1985. Numerical algorithms for transonic, inviscid flow calculations. In *Advances in Computational Transonics*, W. G. Habashi, Ed., pp. 23–58. Pineridge Press, Swansea.
- Harten, A. 1983. High resolution schemes for hyperbolic conservation laws. *J. Comput. Phys.* 49:357–393.
- Harten, A. 1984. On a class of total-variation stable finite-difference schemes. *SIAM J. Numer. Anal.* 21:1–23.
- Harten, A., Engquist, B., Osher, S., and Chakravarthy, S. 1987. Uniformly high order accurate, essentially non-oscillatory schemes III. *J. Comput. Phys.* 71:231–323.
- Hess, J. L. 1990. Panel methods in computational fluid dynamics. *Ann. Rev. Fluid Mech.* 22:255–274.
- Hussaini, M. Y. and Zang, T. A. 1987. Spectral methods in fluid dynamics. *Ann. Rev. Fluid Mech.* 19:339–367.
- Jameson, A. 1979. A multi-grid scheme for transonic potential calculations on arbitrary grids, pp. 122–146. In *Proc. AIAA 4th Comput. Fluid Dynamics Conf.* Williamsburg, VA.
- Jameson, A. 1983. Solution of the Euler Equations by a Multigrid Method. *MAE Rep.* 1613, Department of Mechanical and Aerospace Engineering, Princeton University.
- Jameson, A. 1992. *Computational Algorithms for Aerodynamic Analysis and Design*. Tech. Rep. INRIA 25th Anniversary Conference on Computer Science and Control, Paris. Princeton University Rep. MAE 1966, Princeton, NJ, Dec.
- Jameson, A. and Baker, T. J. 1983. Solution of the Euler equations for complex configurations, pp. 293–302. In *Proc. AIAA Comput. Fluid Dynamics Conf.*
- Jameson, A., Schmidt, W., and Turkel, E. 1981. Numerical solution of the Euler equations by finite volume methods using Runge–Kutta time stepping schemes. *AIAA Paper* 81-1259, AIAA Fluid and Plasma Dynamics Conf., Palo Alto, CA.
- Launder, B. E., Reese, G. J., and Rodi, W. 1975. Progress in the development of a Reynolds-stress turbulence closure. *J. Fluid Mech.* 68(3):537–566.
- Launder, B. E. and Spaulding, D. B. 1972. *Mathematical Models of Turbulence*. Academic Press, London.
- Lesieur, M. and Métais, O. 1996. New trends in large-eddy simulations of turbulence. *Ann. Rev. Fluid Mech.* 28:45–82.
- Lumley, J. L. 1978. Computational modeling of turbulent flows. *Adv. Appl. Mech.* 18:123–176.
- Orszag, S. A. 1972. Comparison of pseudo-spectral and spectral approximation. *Stud. Appl. Math.* 51:253–259.
- Reynolds, W. C. 1987. Fundamentals of turbulence for turbulence modeling and simulation, pp. 1–66. In *Lecture Notes for von Karman Institute*, AGARD Lecture Series No. 86, NATO, New York.
- Roe, P. L. 1986. Characteristic-based schemes for the Euler equations. *Ann. Rev. Fluid Mech.* 18:337–365.
- Rogallo, R. S. and Moin, P. 1984. Numerical simulation of turbulent flows. *Ann. Rev. Fluid Mech.* 16:99–137.
- Shu, C. and Osher, S. 1988. Efficient implementation of essentially non-oscillatory shock-capturing schemes. *J. Comp. Phys.* 77:439–471.
- Smagorinsky, J. 1963. General circulation experiments with the primitive equations. *Mon. Weather Rev.* 91:99–164.
- Spalart, P. R. and Allmaras, S. R. 1992. A one-equation turbulence model for aerodynamic flows. *AIAA Paper* 92-0439, 30th Aerospace Sciences Meeting, Reno, NV.

- Speziale, C. G. 1987. On nonlinear $k-\ell$ and $k-\epsilon$ models of turbulence. *J. Fluid Mech.* 178:459–475.
- Steger, J. L. and Warming, R. F. 1981. Flux vector splitting of the inviscid gasdynamic equations with application to finite-difference methods. *J. Comput. Phys.* 40:263–293.
- Tatsumi, S., Martinelli, L., and Jameson, A. 1995. Flux-limited schemes for the compressible Navier–Stokes equations. *AIAA J.* 33:252–261.
- Thompson, J. F., Warsi, Z. U. A., and Mastin, C. W. 1985. *Numerical Grid Generation*. North Holland, New York.
- van Leer, B. 1974. Towards the ultimate conservative difference scheme, II. Monotonicity and conservation combined in a second-order accurate scheme. *J. Comput. Phys.* 14:361–376.
- van Leer, B. 1979. Towards the ultimate conservative difference scheme, V. A second-order sequel to Godunov's scheme. *J. Comput. Phys.* 32:101–136.
- van Leer, B. 1982. Flux-vector splitting for the Euler equations. In *Lecture Notes in Phys.* 170:507–512.
- Wilcox, D. C. 1993. *Turbulence Modeling for CFD*. DCW Industries, La Cañada, CA.
- Yoon, S. and Kwak, D. 1994. Multigrid convergence of an LU scheme, pp. 319–338. In *Frontiers of Computational Fluid Dynamics — 1994*, D. A. Caughey and M. M. Hafez, eds. Wiley-Interscience, Chichester, U.K.

Further Information

Several organizations sponsor regular conferences devoted completely, or in large part, to computational fluid dynamics. The American Institute of Aeronautics and Astronautics (AIAA) sponsors the AIAA Computational Fluid Dynamics Conferences in odd-numbered years, usually in July; the proceedings of this conference are published by AIAA. In addition, there typically are many sessions on CFD and its applications at the AIAA Aerospace Sciences Meeting, held every January, and the AIAA Fluid and Plasma Dynamics Conference, which is held every summer, in conjunction with the AIAA CFD Conference in those years when the latter is held. The Fluids Engineering Conference of the American Society of Mechanical Engineers, held every summer, also contains sessions devoted to CFD. In even-numbered years, the International Conference on Numerical Methods in Fluid Dynamics is held, alternating between Europe and America; the proceedings of this conference are published in the *Lecture Notes in Physics* series by Springer-Verlag. The International Symposium on Computational Fluid Dynamics, sponsored by the CFD Society of Japan, is held in odd-numbered years, alternating between the U.S. and Asia; in September 1997 this meeting will be held in Beijing, China.

The *Journal of Computational Physics* contains many articles on CFD, especially covering algorithmic issues. The *AIAA Journal* also has many articles on CFD, including aerospace applications. The *International Journal for Numerical Methods in Fluids* contains articles emphasizing the finite-element method applied to problems in fluid mechanics. The journals *Computers and Fluids* and *Theoretical and Computational Fluid Dynamics* are devoted exclusively to CFD, the latter journal emphasizing the use of CFD to elucidate basic fluid-mechanical phenomena. The first issue of the *CFD Review*, which attempts to review important developments in CFD, was published in 1995. The *Annual Review of Fluid Mechanics* also contains a number of review articles on topics in CFD.

The following textbooks provide excellent coverage of many aspects of CFD:

- Anderson, D. A., Tannehill, J. C., and Pletcher, R. H. 1984. *Computational Fluid Mechanics and Heat Transfer*. Hemisphere, Washington.
- Hirsch, C. 1988, 1990. *Numerical Computation of Internal and External Flows, Vol. I: Fundamentals of Numerical Discretization, Vol. II: Computational Methods for Inviscid and Viscous Flows*. Wiley, Chichester.
- Wilcox, D. C. 1993. *Turbulence Modeling for CFD*. DCW Industries, La Cañada, CA.

An up-to-date summary on algorithms and applications for high-Reynolds-number aerodynamics is found in Caughey, D. A. and Hafez, M. M., eds. *Frontiers of Computational Fluid Dynamics — 1994*. J. Wiley Chichester, U.K.

Computational Ocean Modeling

- 31.1 Introduction
- 31.2 Underlying Principles
 - Global or Regional • Deep Basin or Shallow Coastal
 - Rigid Lid or Free Surface • Comprehensive or Purely
 - Dynamical • With Applications to Short-Term Simulations or
 - Long-Term Climate Studies • Quasigeostrophic (QG) or
 - Primitive-Equation-Based • Barotropic or Baroclinic
 - Purely Physical or Physical–Chemical–Biological
 - Process-Studies-Oriented or Application-Oriented
 - With and Without Coupling to Sea Ice • Coupled to the
 - Atmosphere or Uncoupled
- 31.3 Best Practices
 - Barotropic Models • *z*-Level Models • Sigma-Coordinate
 - Models • Layered Models • Isopycnal Models
 - Data Assimilation • Computational Issues
- 31.4 Nowcast/Forecast in the Gulf of Mexico
(a Case Study)
- 31.5 Research Issues and Summary

Lakshmi Kantha
University of Colorado

Steve Piacsek
Naval Research Laboratory

31.1 Introduction

Oceanography is a relatively young field, barely a century old; major discoveries — such as the reason for the western intensification of currents such as the Gulf Stream and Kuroshio, and the existence of a deep sound channel in which acoustic energy can travel for thousands of kilometers with little attenuation — were not made till the 1940s. Even today, our knowledge of the circulation in the global oceans is rather sketchy. Computational ocean modeling is even younger; the very first comprehensive numerical global ocean model was formulated by Kirk Bryan [1969] in the late 1960s. However, the advent of supercomputers has led to a phenomenal growth in the field, especially in the last decade. In a brief chapter such as this, it is impossible to provide a detailed account of the many different versions of the ocean models that exist at present. Instead we will attempt to provide a bird's-eye view of the field and a detailed account of a selected few. The objective is to provide a road map that enables an interested reader to consult appropriate sources for details of a particular approach.

Oceans act as thermal flywheels and moderate our long-term weather. They are also huge reservoirs of CO₂ and have long memory and therefore play a crucial role in determining the climatic conditions on our planet on a variety of time scales. A better understanding of the oceans is also important for other reasons, including defense and commerce needs of nations. They are a source of protein, and might be able to supply part of our energy and mineral needs in the coming century. However, the oceans are data-poor

in general. It is only in the last decade or so that satellite-borne sensors such as infrared radiometers, microwave imagers, and altimeters have begun to fill in the data gaps, especially in the poorly explored southern-hemisphere oceans. Since collection of in situ data in the oceans is quite expensive, and since satellite-borne sensors provide information mostly on the near-surface layers of the ocean, it is often thought that ocean models are central to understanding the way the oceans function. The hope is that comprehensive ocean models in combination with sparse in situ and relatively abundant remotely sensed data will provide the best means of studying and monitoring the oceans. Herein lies the importance of ocean modeling. For prediction purposes, of course, numerical ocean models are quite indispensable.

31.2 Underlying Principles

The choice of a particular ocean model or modeling approach depends very much on the intended application and on the computational and pre- and postprocessing capabilities available. A judicious compromise is essential for success. With this in mind, numerical ocean models can be classified in many different ways.

31.2.1 Global or Regional

The former necessarily requires supercomputing capabilities, whereas it may be possible to run the latter on powerful modern workstations. Even then, the resolution demanded (grid sizes in the horizontal and vertical) is critical. A doubling of the resolution in a three-dimensional model often requires an order of magnitude increase in computing (and analysis) resources. It is therefore quite easy to overwhelm even the most modern supercomputer (or workstation), whether it be a coarse-grained multiple CPU vector processor such as a Cray C-90 or a modern massively parallel machine such as a CM5 or Cray T3-D, irrespective of whether the model is global or regional. Regional models have to contend with the problem of how to inform the model about the state of the rest of the ocean, in other words, of prescribing suitable conditions along the open boundaries. Often the best solution is to nest the fine-resolution regional model in a coarse-resolution model of the basin.

31.2.2 Deep Basin or Shallow Coastal

The prevailing physical processes and the underlying driving mechanisms are essentially different for the two cases. Circulation in shallow coastal regions is highly variable, driven primarily by synoptic wind and other rapidly changing surface forcing (and, near river outflows, by buoyancy differences between the fresh river water and saline ambient shelf water). Wind mixing at the surface and benthic processes are important, and a numerical model that has reliable mixing physics and that resolves the benthic boundary layer is therefore better suited for coastal applications. A model such as the one developed at Princeton University, which employs a bottom-following, sigma-coordinate vertical grid and incorporates an advanced turbulence closure [Blumberg and Mellor 1987; Kantha and Piacsek 1993], may be essential for such applications.

Deep basins, on the other hand, are comparatively sluggish, and the horizontal density gradients, especially below the wind-mixed upper layers, are a dominant factor in the circulation. The upper mixed layer can often be modeled less rigorously, especially for applications that do not require consideration of air-sea interaction processes. The popular z-level Geophysical Fluid Dynamics Laboratory Modular Ocean Model (MOM), with or without an upper mixed layer, is a good candidate for modeling the ocean basins (and deep marginal and semienclosed seas) on a variety of time scales ranging from synoptic to climate. The very first global ocean model [Bryan 1969], on which many modern global models are based [Semtner and Chervin 1992], was a z-level model without an upper mixed layer.

In a z-level model, a number of horizontal levels are defined in the water column, and the equations are written for the oceanic variables at each level and each point on the model grid in the horizontal and solved. This is an Eulerian approach. Another equally viable approach is a semi-Lagrangian approach [Hurlburt and Thompson 1980] that divides the ocean vertically into a number of layers and models the variation in

properties such as the thickness and density of each layer at each grid point on the horizontal grid. More recently developed isopycnal models [Oberhuber 1993, Bleck and Smith 1990] belong to this category. Since mixing in the deep oceans is primarily along **isopycnals** (density surfaces), isopycnal models perform a better job of depicting interior mixing and are ideal for long-term simulations of circulation in the global oceans.

31.2.3 Rigid Lid or Free Surface

Oceanic response to surface forcing can often be divided into two parts: fast barotropic response mediated by external Kelvin and gravity waves on the sea surface, and relatively slow baroclinic adjustment via internal gravity and other waves. On long time scales, it is the internal adjustment that is important to model and it is possible to suppress the external gravity waves by imposing a *rigid lid* on the free surface. This permits larger time-stepping of the model, and models used for climatic-type simulations are usually the rigid-lid kind. The very first global ocean model [Bryan 1969] was a rigid-lid model. At each time step an elliptic Poisson equation for the stream function has to be solved. This is difficult to carry out efficiently on vector and parallel processors and for complicated basin shapes (including islands). Also, under synoptic forcing, the convergence of the iterative solver slows down. For these reasons, free-surface models are becoming more popular for nonclimatic simulations. A mode-splitting technique must then be employed to circumvent the severe limitation on time-stepping that would otherwise be imposed. For shallow-water applications, such as storm-surge and tide modeling, free-surface dynamics must be retained. To diminish the drawbacks of a rigid-lid model, Dietrich et al. [1987] and Dukowicz and Smith [1994] have developed versions in which one works with the pressure on the rigid lid, rather than the barotropic stream function, leaving the domain multiply connected and with better matrix inversion characteristics.

31.2.4 Comprehensive or Purely Dynamical

Since the density gradients are overwhelmingly important and the density below the upper layers in the global oceans changes very slowly, it is often possible to ignore completely the changes in density with time. The model then becomes purely dynamical and can be used to explore the consequences of changing wind forcing at the surface. Purely dynamical layered models, originally developed at Florida State University [Metzger et al. 1992; Wallcraft 1991], belong to this category and are essentially isopycnal models without the thermodynamic component [Hurlburt and Thompson 1980]. Their principal advantage is that it is often possible to select a limited number of layers in the vertical (as few as two) and still include salient dynamical processes. This enables very high horizontal resolutions necessary for resolving mesoscale eddies in the oceans to be afforded. The highest-resolution global model at present is the Naval Research Laboratory $\frac{1}{8}^\circ$ eddy-resolving global model [Metzger et al. 1992] that needs a 16-processor Cray C-90 for multiyear simulations.

Even with modern computing power, it is necessary to sacrifice either vertical or horizontal resolution for many (especially global) simulations. The layered (and isopycnal) models sacrifice vertical resolution, whereas the z -level models, employing large numbers of levels in the vertical, are necessarily comparatively coarse-grained in the horizontal. The highest-resolution dynamical–thermodynamic z -level model at present is the $\frac{1}{6}^\circ$ Semtner model at Los Alamos (Dukowicz and Smith 1994; see Semtner and Chervin 1992 for a description of the basic model) and it stretches the capability of a 256-processor CM5 to the limit.

31.2.5 With Applications to Short-Term Simulations or Long-Term Climate Studies

On climatic time scales, it is extremely important to model correctly the thermohaline circulation driven by the formation of dense deep-water masses during strong wintertime cooling in subpolar seas, especially in the Atlantic. These water masses flow along the ocean bottom, and several centuries are needed for a water particle that sank (say) in the north Atlantic to surface again in the Indian or the Pacific Ocean.

Because of this long memory of the deep oceans, it is necessary to make multicentury simulations, and, irrespective of whether isopycnal or z -level models are employed, the horizontal and vertical resolutions that can be afforded are necessarily coarse. Accurate ocean simulations on climatic time scales belong to the category of grand challenge problems requiring a teraflop (10^{12} floating-point operations per second) computing capability that has been the holy grail of the computer industry.

31.2.6 Quasigeostrophic (QG) or Primitive-Equation-Based

In the 1970s and early 1980s, the limited computing power available led some to explore simplifications to the governing equations to be solved. QG models assume that there is a near-balance between the Coriolis acceleration and pressure gradient in the dynamical equations in the rotating coordinate frame of reference in which most ocean models are formulated. The resulting simplification enables higher vertical and horizontal resolutions to be achieved. QG models have strong limitations with respect to the accuracy with which physical processes are depicted and are becoming obsolete in the modern high-computing-power environment. We will not discuss QG models in this article, but instead refer the reader to Holland [1986]. Neither shall we discuss intermediate models, which are in between QG and primitive equation (PE) models in complexity.

31.2.7 Barotropic or Baroclinic

In the former, the density gradients are neglected and therefore the currents become independent of the depth in the water column. Many phenomena, such as tidal sea-surface elevation fluctuations and storm surges, can be simulated quite adequately by a **barotropic** model, which is a two-dimensional (in the horizontal) model based on the vertically integrated equations of motion. The advantage is that a barotropic model requires an order of magnitude less computing resources than a comparable baroclinic model. However, when it is important to model the vertical structure of currents, or the density field, a fully three-dimensional **baroclinic** model is necessary.

31.2.8 Purely Physical or Physical–Chemical–Biological

Often there is a need to model the fate of chemical and biological constituents in the ocean, and to do so it is essential to include not only the dynamical equations governing the circulation and other physical variables, but also the conservation equations for chemical and biological variables. Modeling the fate of inorganic CO_2 in the oceans (a problem germane to global warming) and modeling the primary production in the upper layers of the ocean are two such examples. The former requires solving for at least two more variables, the total CO_2 and alkalinity, whereas the simplest biological model must solve for at least three additional quantities, the nutrient, phytoplankton, and zooplankton concentrations (the so-called NPZ model). The governing equations are transport equations with appropriate source and sink terms, whose parametrization is not always quite straightforward. This not only implies additional complexity but also requires considerably more computing (and data) resources.

31.2.9 Process-Studies-Oriented or Application-Oriented

Models that are used to study some salient processes (for example western boundary currents and gyre circulation) can often be considerably simplified and hence made less computationally (and observational data-) intensive, since it is often possible to isolate and retain only the relevant physical process and ignore the rest in the model. Also, such models may not require extensive observational data for model initialization or forcing. They are most often run free in a predictive (or prognostic) mode. On the other hand, application-oriented models such as those used for ocean prediction purposes require extensive observational data for realistic initialization, forcing, and data assimilation. Assimilation into the model by one means or other of observational data is indispensable for **nowcast**, forecast, and **hindcast** applications,

and data-assimilative models often employ approaches very much similar to those employed by numerical weather prediction (NWP) models in the atmosphere.

31.2.10 With and Without Coupling to Sea Ice

Global ocean models do not at present include sea ice. Ice-ocean-coupled basin models of the Arctic exist, however; for coupling to a z -level ocean model, see Hibler and Bryan [1987], and to an isopycnal model, see Oberhuber [1993]. Sea ice insulates the ocean from the cold atmosphere and mediates the exchange of heat and momentum between the two, and therefore such models involve solving dynamical and thermodynamic equations for the sea ice cover and its coupling to the underlying ocean.

31.2.11 Coupled to the Atmosphere or Uncoupled

Finally, for accurate simulation of long-time-scale processes, it is essential to couple ocean models with atmospheric models. Such coupled models are being increasingly used for such things as forecasting **El Niño** events. Most often, either the atmosphere or the ocean is highly simplified in such models, although modern supercomputers are enabling comprehensive atmospheric general circulation models to be coupled to global ocean models, at least for simulations of interannual variability. Truly comprehensive coupled models with applications to long-term climate studies require teraflop computing capability that might be routinely available in the coming century.

All numerical ocean models solve one form or other of the same governing equations for oceanic motions, written in a rotating coordinate frame of reference. These equations are essentially Navier–Stokes equations (or more appropriately **Reynolds-averaged** equations for mean term quantities, since the flow is invariably turbulent), but with the buoyancy and Coriolis force terms (fictitious accelerations due to the noninertial nature of the rotating coordinate frame) prominent in the dynamical balance. In addition, an equation of state relating the density of seawater to its temperature and salinity, and conservation equations for temperature and salinity, are also solved. In those models which model turbulence explicitly, equations for turbulence quantities such as the turbulence velocity scale (or equivalently turbulence kinetic energy) and turbulence macroscale are also solved. If chemical or biological components are included, conservation equations for relevant species with appropriate source and sink terms are solved as well.

Global and basin-scale models are formulated in spherical coordinates, but regional models are usually cast in rectangular coordinates instead. For simplicity we present the governing equations in rectangular Cartesian coordinates (spherical coordinate version can be found for example in Semtner [1986]). The x_1 -axis is usually taken to be in the zonal direction (positive to the east), the x_2 -axis is in the meridional direction (positive to the north), and the z -axis is in the vertical direction, positive upwards, with the origin located at the sea surface. Using tensorial notation and treating the horizontal coordinates separately from the vertical (indices take values 1 and 2 only), the governing equations consist of the continuity equation

$$\frac{\partial U_k}{\partial x_k} + \frac{\partial W}{\partial z} = 0 \quad (31.1)$$

where U_k denotes the horizontal components of velocity and W the vertical, and the momentum equations

$$\frac{\partial U_j}{\partial t} + \frac{\partial}{\partial x_k}(U_k U_j) + \frac{\partial}{\partial z}(W U_j) + f \epsilon_{j3k} U_k = -\frac{\partial P}{\partial x_j} + \frac{\partial \Phi}{\partial x_j} + \frac{\partial}{\partial z} \left(K_M \frac{\partial U_j}{\partial z} \right) + F_j \quad (31.2)$$

$$\frac{\partial P}{\partial z} = -\frac{\rho}{\rho_0} g \quad (31.3)$$

where f is the Coriolis parameter ($2 \Omega \cos \phi$), Ω the earth's angular velocity, ϕ the latitude, P the kinematic pressure, K_M the vertical mixing coefficient, ρ the in situ density, ρ_0 the reference density, g the gravitational acceleration, and Φ the gravitational potential.

The transport equations for **potential temperature** Θ and salinity S are

$$\frac{\partial \Theta}{\partial t} + \frac{\partial}{\partial x_k}(U_k \Theta) + \frac{\partial}{\partial z}(W \Theta) = \frac{\partial}{\partial z} \left(K_H \frac{\partial \Theta}{\partial z} \right) + S_\Theta + F_\Theta \quad (31.4)$$

$$\frac{\partial S}{\partial t} + \frac{\partial}{\partial x_k}(U_k S) + \frac{\partial}{\partial z}(W S) = \frac{\partial}{\partial z} \left(K_H \frac{\partial S}{\partial z} \right) + F_S \quad (31.5)$$

where K_H is the vertical mixing coefficient for scalar quantities, and S_Θ denotes a volumetric heat source such as due to penetrative solar heating. The equation of state is given by

$$\rho = \rho(\Theta, S) \quad (31.6)$$

Various simplifications have been made in deriving the above equations. The ocean is considered incompressible, a very good approximation for most applications. It is also considered to be in hydrostatic balance in the vertical, and hence the only terms remaining in the vertical component of the momentum equations are the buoyancy and pressure gradient terms. The hydrostatic approximation involves neglecting the vertical acceleration and regarding the fluid as essentially motionless in the vertical. In addition, the Boussinesq approximation has been used. This involves considering the ocean to be of constant density except when buoyancy forces are computed, thus assuming $\rho = \rho_0$ in all except the terms involving gravitation. Also, terms containing the horizontal component of rotation are neglected. The resulting equations are sufficiently accurate for most ocean circulation modeling. The equation of state employed is the so-called UNESCO equation [Pond and Pickard 1979] and is in the form of polynomial expansions in temperature and salinity. The pressure P is given by

$$P = g\eta + \frac{g}{\rho_0} \int_z^0 \rho(x_j, z', t) dz' \quad (31.7)$$

where η is the sea surface height. The terms F_j , F_Θ , F_S are horizontal mixing terms corresponding to unresolved subgrid-scale processes and are most often parametrized simply as Laplacian diffusion terms:

$$F_{j,\Theta,S} = A_{M,H} \frac{\partial^2}{\partial x_k \partial x_k} (U_j, \Theta, S) \quad (31.8)$$

where $A_{M,H}$ are horizontal mixing coefficients. A more rigorous form for these terms can be found in Blumberg and Mellor [1987]. The values for these coefficients are most often chosen as constants in a rather ad hoc manner based on purely numerical considerations. While the vertical mixing coefficients K_M and K_H can be rigorously modeled by turbulence closure theories [for example, Kantha and Clayson 1994, Mellor and Yamada 1982], there does not exist a similar approach for these terms. One approach, widely used in atmospheric modeling, is that due to Smagorinsky [1963], which is similar to the classical mixing-length theory of turbulence. Here the mixing coefficient is assumed to be proportional to the mean strain rate, so that

$$A_M = C(\Delta x_1)(\Delta x_2) \left[\left(\frac{\partial U_i}{\partial x_j} + \frac{\partial U_j}{\partial x_i} \right) \left(\frac{\partial U_i}{\partial x_j} + \frac{\partial U_j}{\partial x_i} \right) \right]^{1/2}, \quad A_H \sim A_M \quad (31.9)$$

where C is the Smagorinsky coefficient, with a value of around 0.04, and Δx_1 and Δx_2 are the grid sizes. This approach assumes that the subgrid scales fall within the Kolmogoroff inertial subrange, an assumption not always satisfied. A practical consequence of using this model is that strong horizontal shear is accompanied by strong horizontal mixing, which tends to smear out thermal fronts. A more general approach is to assume that the mixing coefficients are a sum of a constant background value and the Smagorinsky value given by Equation 31.9 and to choose the values assigned to each appropriately [for example, Kantha 1995]. Another approach is to assign a constant cell Reynolds number $R_N = |U_j| |\Delta x_j| / A_M$ and determine the

value of the mixing coefficient thus [Choi and Kantha 1995]. Some modelers have used biharmonic form

$$\frac{\partial^2}{\partial x_k \partial x_k} \frac{\partial^2}{\partial x_k \partial x_k}$$

to model these terms [O'Brien 1985]. In this form, the terms serve principally to control the so-called $2 \Delta x$ noise in the numerical solutions. Suffice it to say that modeling horizontal diffusion terms is still rather ad hoc.

The oceans are driven by momentum, heat, and salt fluxes at the air–sea interface. The boundary conditions at the sea surface ($z = \eta$) are therefore

$$\begin{aligned} K_M \left(\frac{\partial U_j}{\partial z} \right) &= \tau_{0j} \\ K_H \left(\frac{\partial}{\partial z} (\Theta, S) \right) &= (q_H, q_S) \\ W &= \frac{\partial \eta}{\partial t} + U_j \frac{\partial \eta}{\partial x_j} \end{aligned} \quad (31.10)$$

where τ_{0j} is the kinematic shear stress acting at the free surface due to the action of winds and waves (taken mostly as equal to the kinematic wind stress) and $q_{H,S}$ are the kinematic heat and salt fluxes. The value of q_H is determined by the net heat balance at the air–sea interface due to short-wave and long-wave solar heating, back radiation by the ocean surface, and the turbulent sensible and latent heat exchanges, and that of q_S is determined by the difference between evaporation and precipitation. Accurate parametrization of these air–sea fluxes has been the subject of intense research for several decades (for example, the 1992 multinational Tropical Ocean Global Atmosphere/Coupled Ocean Atmosphere Response Experiment).

The conditions at the ocean bottom ($z = -H$) are of no mass transfer through the bottom

$$W = -U_j \frac{\partial H}{\partial x_j}$$

and

$$\begin{aligned} K_M \left(\frac{\partial U_j}{\partial z} \right) &= \tau_{bj} \\ K_H \left(\frac{\partial}{\partial z} (\Theta, S) \right) &= (0, 0) \end{aligned} \quad (31.11)$$

The last of the above conditions implies no heat or salt transfer through the ocean bottom. The bottom stresses are usually parametrized using a quadratic drag law with $c_d \sim 0.0025$:

$$\tau_{bj} = c_d |U_{bj}| U_{bj} \quad (31.12)$$

or by assuming that the lowest model grid point falls within the logarithmic-law region and using the well-known logarithmic relationship between the mean velocity and the friction velocity to derive the drag coefficient [Blumberg and Mellor 1987, Kantha and Piacsek 1993, for example].

When additional quantities such as turbulence velocity and length scales are modeled explicitly, corresponding conservation equations need to be solved along with appropriate boundary conditions at the ocean surface and bottom [Blumberg and Mellor 1987, Kantha and Clayson 1994, Kantha and Piacsek 1993]. The same holds for modeling of chemical and biological quantities.

If the lateral boundary is closed, then it is straightforward to apply the lateral boundary condition; the component of velocity perpendicular to the boundary is zero, and there is no lateral mass, heat, or salt flux through the boundary. If it is open, on the other hand, as is usual in regional models of coastal and marginal seas, then it is necessary to prescribe open boundary conditions. This is a difficult problem, since

complete information on various flow properties must be specified, and this depends to a large extent on how well the flow at the boundary is known. The best strategy is to nest the model in a coarser-resolution model of the basin. In many cases, this is not feasible, and hence it is not possible to inform the model about what the rest of the ocean is doing. The best under these conditions is some form of Sommerfeld radiation boundary condition on dynamical quantities, which ensures that disturbances approaching the boundary from the inside are radiated out and not bottled up [Blumberg and Kantha 1985, Kantha et al. 1990, Roed and Cooper 1986]. This is usually of the form

$$\frac{\partial \zeta}{\partial t} + C \frac{\partial \zeta}{\partial x_n} = 0 \quad (31.13)$$

where ζ is a variable such as the sea surface height, n denotes direction normal to the boundary, and C is the phase speed of the approaching disturbance. Proper prescription of C is important to the success of the radiative boundary condition and has been the subject of much research [Blumberg and Kantha 1985, Orlonski 1976].

If there is inflow at the lateral boundary, temperature and salinity of the incoming flow must be prescribed. If there is outflow, on the other hand, these quantities are simply advected out:

$$\frac{\partial(\Theta, S)}{\partial t} + U_n \frac{\partial(\Theta, S)}{\partial x_n} = 0 \quad (31.14)$$

31.3 Best Practices

In many applications, ocean simulation is an initial-value problem and therefore the initial state of the modeled ocean must be prescribed as accurately as possible. This is difficult to do in practice, since observational data are insufficient to specify the state of the ocean at any given point in time. The best alternative is to prescribe some sort of a climatological average as the initial state and spin up the ocean from that state under prescribed surface forcing. Here appeal is made to databases such as [Levitus 1982] (updated in 1995) and the U.S. Navy GDEM that contain distributions of climatological average temperature and salinity in the global oceans derived from historical archives of in situ observations.

Prescription of surface forcing is itself a major problem. Both momentum and buoyancy fluxes need to be specified at the sea surface, and both are determined by processes in the adjacent atmospheric boundary layer. For determining the long-term average state of the oceans, it is once again possible to appeal to climatological databases such as that due to Hellerman and Rosenstein [1983] and the Comprehensive Ocean-Atmosphere Data Set (COADS) database (the database has been updated to 1990s recently), which provide gridded monthly average values of wind stress over the global oceans derived from historical marine surface observations. However, for many applications, including ocean prediction, it is necessary to provide surface forcing on a daily or even a multihourly basis. This is impossible to do from observations alone (although satellite-sensed air-sea fluxes may help in the future), and one has little choice but to appeal to six-hourly analyses and predictions of NWP centers, even though the accuracy of the surface forcing so derived depends very much on the skill of the particular NWP model and methodology.

Finally, the governing equations are discretized and the resulting finite-difference equations (or a set of algebraic equations) are solved to determine the evolution with time of the various oceanic variables such as temperature, salinity, and velocity at each model grid point. There are two principal means of discretization in the horizontal direction: finite differences [for example, Blumberg and Mellor 1987, Choi and Kantha 1995, Semtner and Chervin 1992] and finite elements [Le Provost et al. 1994], the latter being quite popular with civil engineers. The principal advantage of the latter is the flexibility the finite-element grid provides in assigning localized high resolution where needed. The reader is referred to Kantha [1995] for an example of a finite-difference approach to modeling the barotropic equations for global tides and to Le Provost et al. [1994] for a finite-element approach to the same problem. Advanced discretization techniques such as adaptive grids and nonorthogonal coordinate systems that are quite routinely used in