A Perspective of Theoretical and Applied Computational Fluid Dynamics

Paul Kutler
NASA Ames Research Center, Moffett Field, California

Introduction

TO heighten his understanding of the fluid dynamics and aerodynamics pertinent to the early stages of the atmospheric flight-vehicle design process, the aerodynamicist has at his disposal three standard tools: analytical methods, computational procedures, and experimentation. Some of the advantages and disadvantages of each design tool are summarized in Fig. 1.

Analytical methods provide quick, closed-form solutions, but they require unduly restrictive assumptions, can treat only simple configurations, and capture only the idealized aerodynamics. Through experimentation, representative or actual configurations can be tested and representative and complete aerodynamic data can be produced. Experimentation is costly, however, both in terms of the model and actual test time. In addition, the limited conditions that can be attained in wind tunnels restrict the scope of experimental programs.

Relative to analytical techniques, computational procedures require very few restrictive assumptions and can be used to treat complicated configurations. In addition, they have few Mach number or Reynolds number limitations, they result in complete surface and exterior flowfield definition, and, most important, they are far more cost effective than wind-tunnel testing. On the other hand, when the Reynold-averaged Navier-Stokes equations are used, computations require an adequate turbulence model to correctly simulate viscous-dominated flows, as well as a powerful, that is, large and fast, mainframe computer on which to perform the simulation. The desirability of numerical simulations is enhanced when one considers that the cost of wind-tunnel experimentation is continually increasing because of model, labor, and energy overhead, whereas the cost of computer simulations is continually decreasing as a result of improved numerical procedures and advances in computer technology.

It is such a comparative weighting of all advantages and disadvantages that underlies the growth of computational fluid dynamics (CFD) and, especially, its increasing role in the design of new flight vehicles. In Fig. 2, the "push-pull" concept is used to amplify the factors affecting the popularity of CFD. The need for a cost-effective alternative to experimentation is "pushing" the growth of CFD and factors such as the significant improvements in computer design, enhancements of CFD solution methods, and the advances in computer graphics facilities for data manipulation and display are all undergoing significant growth that tend to "pull" CFD to even higher levels of desirability. In addition to aerodynamic applications, CFD plays a major role in various other disciplines (Fig. 3), for example, in meteorology, civil engineering, and nuclear science. This wide range of CFD applications is yet another "pulling" factor that further increases its popularity.

Flight-vehicle design visionaries who are spearheading applied CFD developments have as an ultimate goal the capability for designing a complete flight vehicle (e.g., commercial transport, fighter aircraft, and re-entry vehicle) using CFD technology (see Fig. 4). The flow codes for such technology will include 1) flow solvers for determining the realistic flow about a particular configuration (i.e., an inviscid solver coupled with a viscous correction procedure for testing a scenario of design configurations, as well as a complete viscous Navier-Stokes solver for verifying a preliminary computational design); and 2) optimization packages with multidisciplinary integration for refining the design based on aerodynamic, structural, acoustic, or other constraints established by the vehicle designers.

The discipline of CFD has progressed toward satisfying the stated utopian objective, but there is still a considerable amount of work to be accomplished before a computer-designed flight vehicle takes to the air. To date the only realistic computational flow simulations that have been obtained involve either simplified configurations or components of complicated configurations. In many instances, CFD tools have been successfully employed in the partial design of relatively simple flight vehicles, and, in some special instances, coupled flow solver/optimization programs have resulted in improved designs of aircraft components. The

Dr. Paul Kutler is Chief of the Applied Computational Aerodynamics Branch at the Ames Research Center, NASA. In this position, Dr. Kutler leads a group of research scientists who develop computer programs for numerically simulating the realistic flow about aerospace vehicles. Dr. Kutler joined Ames as an aerospace engineer in 1969 after spending a year on a NASA work-study program performing the research for his dissertation. He received his B.S., M.S., and Ph.D. degrees from Iowa State University in 1965, 1967, and 1969, respectively, and was partially supported for three years by a NASA Fellowship. He made his fundamental contributions in the area of computational fluid dynamics and, in particular, the development of shock-capturing procedures for predicting complicated supersonic flowfields about realistic configurations. In 1975 Dr. Kutler received the H. Julian Allen Award of the Ames Research Center for his work on the shock-on-shock interaction problem and was selected by Iowa State University for the Outstanding Young Alumnus Recognition Award. In 1977 he received the AIAA Lawrence Sperry Award for his contributions to the discipline of computational fluid dynamics. Dr. Kutler is author of numerous technical publications and has lectured at both the national and international levels. He is active in the AIAA and has served as the Chairman of the San Francisco Section. He is an Associate Fellow of the AIAA.

Presented as Paper 83-0037 at the AIAA 21st Aerospace Sciences Meeting, Reno, Nev., Jan. 10-13, 1983; received March 24, 1983; revision submitted Jan. 24, 1984.

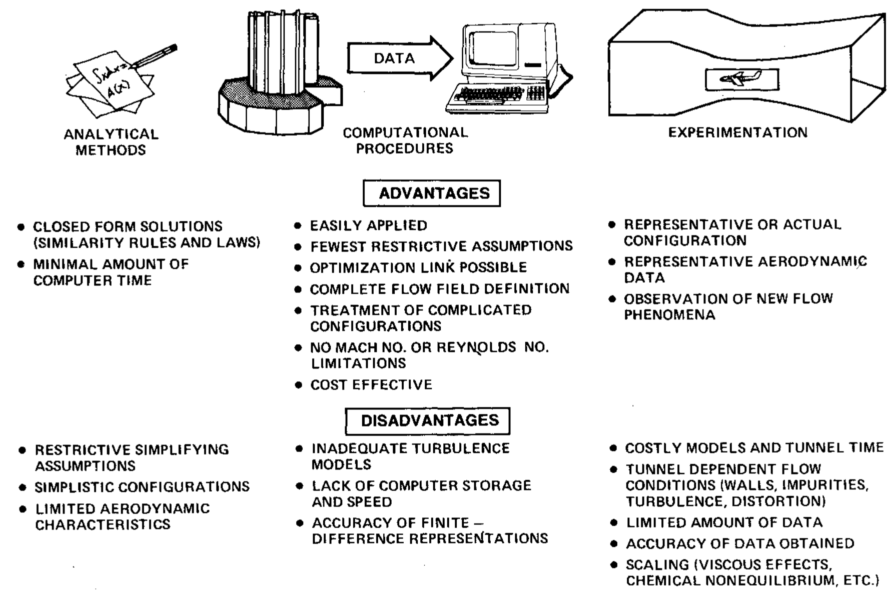

Fig. 1 Tools for aerodynamic prediction.

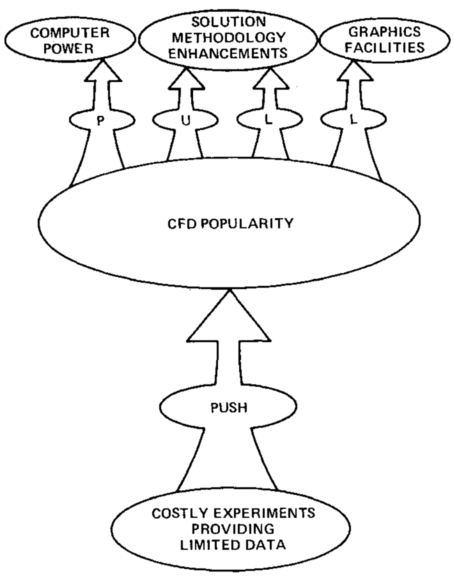

Fig. 2 Push-pull theory for CFD popularity.

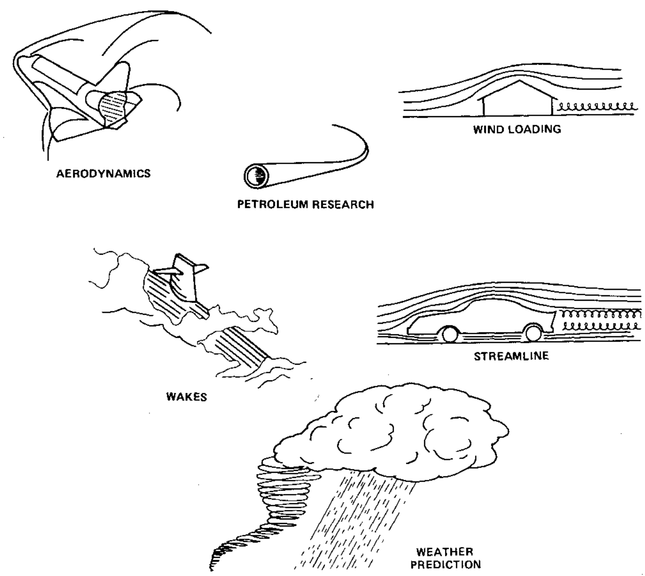

Fig. 3 Computational fluid dynamic applications.

HiMAT wing, the Rockwell forward-swept wing, and the Saab transonic fighter wing were all designed using, first, a small-disturbance procedure to survey potential designs and, second, a full-potential method with boundary-layer correction to computationally verify the design.[1] Models of all these wings were built and tested, and the results of the experiments satisfied the design goals and, hence, verified the computationally designed wings. In all cases, the use of CFD tools reduced the amount of experimental development work from what would have been required without its use.

To accomplish the ideal stated above will require a major advance in the state of the art of computational methods for treating three-dimensional configurations. To this end there are both primary and secondary pacing items that must and are being addressed by the CFD research community. The primary pacing items listed by Chapman[2] for solving the Reynolds-averaged Navier-Stokes equations include 1) three-dimensional grid generation, 2) turbulence modeling, 3) algorithm development, and 4) computer mainframe design advances. One of these, algorithm development, is replaced in this paper as a primary pacing item by solution methodology development for reasons to be explained later. It is, however, still listed as a secondary pacing item in CFD. The secondary pacing items include 1) algorithm development, 2) complex geometry definition, and 3) input and output data processing. Most of this paper is devoted to a discussion of these pacing items. In addition, user demands on computational tools for simulations are also discussed.

Primary Pacing Items in Computational Fluid Dynamics

Although CFD and its value as an analytical tool have progressed tremendously in the past decade, there are still important technological developments that must be made before the complicated configurations encountered in the "real world" can be confidently and successfully treated. As in any area of research, there are items that pace the progress of CFD. Chapman[2] pointed out that, as these pacing items are developed, CFD will reach a new level of computational capability and thus play a more important role in future aerodynamic applications.

The primary pacing items to be discussed in this section are depicted in Fig. 5. They include three-dimensional grid generation, turbulence modeling, solution methodology development, and mainframe computer methodology and architectural advances. In the following section, the secondary pacing items, which play a subordinate but yet important role in the advancement and acceptability of CFD, are discussed.

Three-Dimensional Grid Generation

One of the most important steps required to accurately solve a three-dimensional CFD problem using finite difference procedures involves the proper location of the nodal points in the flow region disturbed by the body. During the past two years this area has received considerable attention. A good summary of this technology is given by Carr and Forsey.[3]

There are basically two decision stages involved in the discretization of a three-dimensional flowfield. The first involves a decision about the grid-generation concept or topology to be used, and the second involves a decision about the grid generation scheme to be employed. The various grid-generation concepts and schemes are summarized in Fig. 6 and discussed next.

Most discretization methods can be classified as either one or a combination of the concepts listed in Fig. 6. These include single-module, multiblock, interfering (or nonaligning), component-adaptive overlapping, and component-adaptive interfacing concepts, all of which are described below. Probably the easiest and most popular concept is the single module, in which the discretized flow region is transformed into a single computational cube in three dimensions. The example in Fig. 7a shows the discretized flow region about a supersonic blunt body in which the boundaries consist of the body, shock, axis, and outflow plane. For this case a coordinate singularity exists at the axis.

The multiblock concept results when more than one of these blocks are linked together. An example of this concept is shown in Fig. 7b (from Lee and Ruppert[4]) which illustrates the discretized flow region about a commercial aircraft configuration. This approach results in singular lines and planes where special treatment at these boundaries is required. Another effective multiblock approach is that used by Thomas[5] for discretizing a wing fuselage combination and termed by him the "composite subregion approach."

A grid-generation concept that has received some attention is the body-interfering or nonaligning procedure. It is a nonconforming, overlapping method, that is, no single coordinate line coincides with the surface of the body, and the mesh is permitted to extend inside the body. A typical example of the interfering concept is shown in Fig. 7c. This concept has been successfully used by Wedan[6] in computing the flow about several complicated configurations at transonic speeds. The basic advantage of the interfering concept is that it can treat complicated configurations. The disadvantages are that the surface boundary condition procedure is difficult to apply, and the capability for fine resolution of the flowfield near the surface is difficult to obtain.

Two other grid-generation concepts which show promise of being applicable to three-dimensional flow regions are component-adaptive overlapping and component-adaptive interfacing. Both are body-conforming procedures. In the overlapping concept (Fig. 7d), the body-conforming grid overlaps the inner region of the outer grid and requires an interpolation procedure to transfer information from one grid to the other. This approach has been applied successfully by Atta[7] for the full-potential formulation in both two- and three-dimensional problems, and a slight variation for both Euler and streamfunction formulations currently is being explored by Steger et al.[8] in two dimensions. This shows promise of being applicable for treating the multiple moving body problem.

In the component-adaptive interfacing concept (Fig. 7e), the body-conforming grid is embedded in the outer grid such that at the mesh boundaries there is no overlap of the two grids. Coordinate lines crossing the boundaries from one grid to the other do so in one of three ways. These are shown in Figs. 7f-h. In the first method (Fig. 7f), the coordinate lines from one grid match in both location and slope with the coordinate lines of the other grid. In the second method (Fig. 7g), the crossing coordinate lines match in only location and not slope. In the third method (Fig. 7h), the coordinate lines from one grid do not line up with the coordinate lines from the other grid. Simple conservative interpolation procedures are only required for the last two methods. However, at the patch boundary, grid singularities are possible. These singularities, however, are placed in the flowfield away from the body surface and, thus, are not affected by the body boundary conditions and severe flow gradients. The component-adaptive interfacing concept has been successfully applied by Lasinski et al.[9] in computing the viscous flow about a tri-element airfoil configuration. The version of this concept shown in Fig. 7h shows great promise of being able to treat complicated configurations. The boundary condition

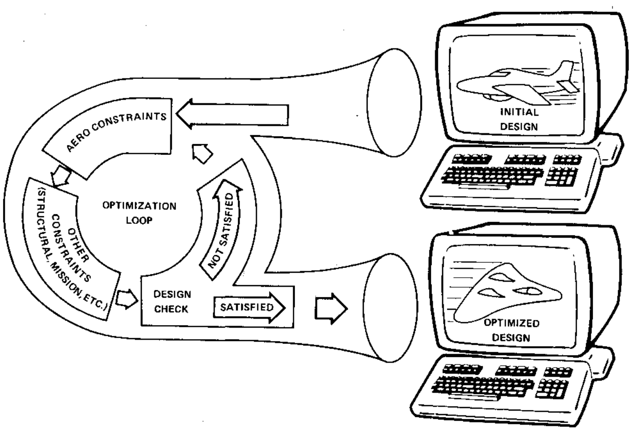

Fig. 4 Computational design optimization cycle: the ultimate goal.

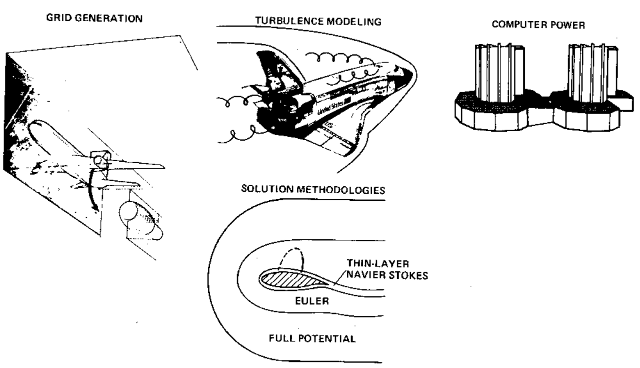

Fig. 5 Primary pacing items in computational fluid dynamics.

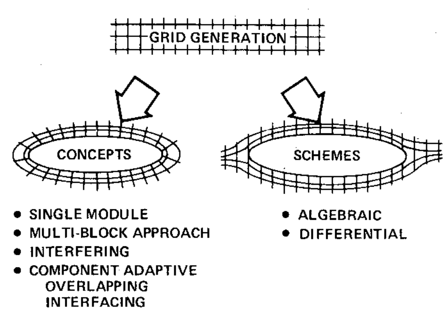

Fig. 6 Discretization methods.

procedures for accurately transferring information across the interfacing boundaries of this particular method are being actively developed by CFD researchers.

The choice of the grid-generation concept to be used in discretizing a flow region is based on many factors, one of which depends on whether a viscous calculation is to be performed. Optimization of grid-point use for a viscous Navier-Stokes procedure normally requires a body oriented or component-adaptive grid. This permits one array of coordinate lines to intersect the body in a nearly normal way while the other array of coordinate lines wrap around the body and are clustered near it. This resolves the thin viscous region near the surface. Both the component-adaptive overlapping and interfacing concepts show promise of being able to satisfactorily treat viscous flows about three-dimensional configurations.

The grid-generation concepts illustrated in Fig. 7 have been successfully applied to actual flow problems, including those that yield a grid singularity on the body surface. For example, Jou and Huynh[10] and Kwak[11] demonstrate that the full-potential equation can be solved in the presence of an "H-grid" singularity for the cascade problem.

Once the desired grid-generation concept has been decided, the next step is to determine the actual location of the grid points. To do this, a variety of grid-generation schemes are available. Basically, they are classified into two categories: algebraic and diffferential. Each scheme has variations, some of which are depicted in Fig. 8. In this paper, emphasis is on differential grid-generation schemes, although algebraic methods are briefly discussed and reference information is provided. The proceedings for a symposium on numerical grid generation[12] is an excellent publication that covers the state of the art in the technology of grid generation and acts as a handy reference guide for both algebraic and differential grid-generation schemes.

Algebraic schemes may be classified as either conformal or nonconformal according to Mehta and Lomax.[13] Conformal procedures, according to Ives,[14] can be classified as either classical or contemporary. Some nonconformal procedures include multisurface transformations, transfinite interpolations, and isoparametric mappings. A descriptive summary of some of these algebraic procedures can be found in papers by Ives,[14] Eiseman,[15] and Carr and Forsey.[3]

Differential grid-generation schemes can be classified as elliptic, hyperbolic, or parabolic. They have received widespread attention recently because of their versatility and the ease with which they can be applied. The underlying characteristics of each scheme are depicted in Fig. 9. Elliptic schemes require the specification of data at all boundaries of the computational line, plane, or volume (i.e., the x, y, and z locations of the boundary grid points). The location of the grid points between these boundaries is then determined iteratively by solving a set of elliptic partial differential equations (PDE's). Elliptic procedures ensure monotonicity of the coordinate lines, that is, they are not multivalued. Mesh point control, that is, coordinate line clustering and inclination, is possible with the use of forcing functions. This procedure was popularized by Thompson et al.[16] as an extension of the work by Barfield,[17] Godunov and Prokopov,[18] and Amsden and Hirt.[19]

A quasi-three-dimensional "O-grid" (i.e., two-dimensional at each spanwise station) generated about a wing-fuselage combination using the elliptic procedure is shown in Fig. 10. The program that generated this grid is called GRAPE; it was designed by Sorenson.[20] GRAPE can control the grid-point spacing and inclination at all computational boundaries. The elliptic procedure has been extended to three-dimensions by Mastin and Thompson[21] and Lee et al.[22]

Hyperbolic schemes require only the specification of initial data at one boundary (Fig. 9). The grid is generated by marching outward from this boundary until the far field is reached. The procedure is noniterative but provides no control of the position and shape of the outer boundary. A hyperbolic grid-generation scheme was presented by Steger and Chaussee[23] for airfoil applications. A typical example of the hyperbolic grid-generation procedure applied to a cross section of the flowfield about the Space Shuttle orbiter is shown in Fig. 11. This same procedure has been applied in three dimensions for simple wings in a transonic flow by Bridgeman et al.[24]

The parabolic grid-generation scheme is noniterative in that no iterations are required in the three-dimensional sense (Fig. 9). The grid is generated by marching from one coordinate surface to the next beginning at the inner boundary and terminating at a prescribed outer boundary, both of which are discretized initially. An iteration scheme is used to generate

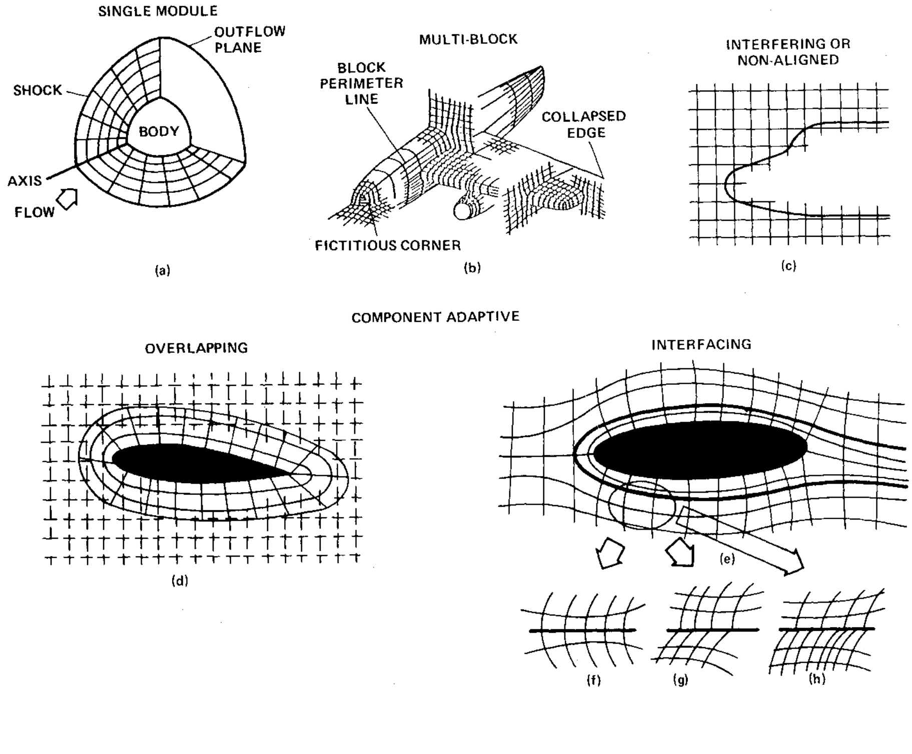

Fig. 7 Grid generation concepts: a) single module; b) multiblock; c) interfering or nonaligned; d) component-adaptive overlapping; e) component-adaptive interfacing; f) continuity of function and slope; g) continuity of function; h) discontinuous function and slope.

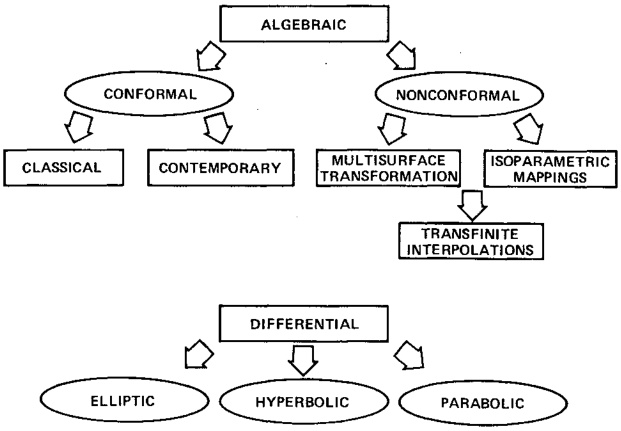

Fig. 8 Grid generation schemes.

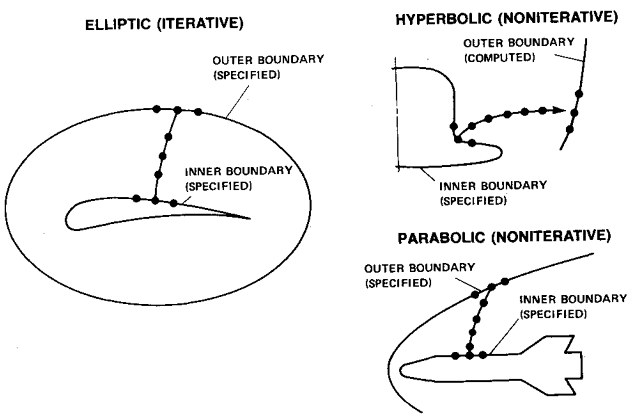

Fig. 9 Differential grid generation schemes.

each grid surface during the outward marching process, but its convergence rate is quite fast (i.e., requires at most three or four iterations). The parabolic grid-generation scheme was devised by Nakamura[25] and has been applied successfully to the simple wing-body configuration shown in Fig. 12. Coordinate line spacing and orientation is possible with this procedure. This scheme combines the attributes of both the elliptic and hyperbolic grid-generation procedures and, hence, has the potential of providing efficient and adequate grids for most aerodynamic applications.

The preceding discussion was concerned with the discretization of a given flowfield, but one of the most important overlooked aspects of the grid-generation process is the "quality" of the grid generated. The grid is one of three factors that have a profound effect on the accuracy of the numerical solution (see also Ref. 13); the other two being the numerical algorithm and the flowfield itself. In most applications today, aesthetics is the dominating criterion for judging the suitability of a given mesh for a particular problem. Klopfer[26] tried to quantify the criteria for generating acceptable finite difference grids (including assessment of grid smoothness, skewness, and cell aspect ratio), and showed that it is not possible to do so independently of the numerical solution obtained on the mesh. However, as a first-order estimate of the grid quality, variations of the metrics and Jacobian can be studied. This subject is deserving of more attention.

There has been great progress in the discipline of three-dimensional grid generation since it was identified as a pacing item by Chapman in June 1980. There is not and likely will not be a universally accepted concept or scheme for discretizing three-dimensional flowfields. Combinations of the procedures summarized earlier will likely be used to satisfy the requirements of the user community. Certainly as

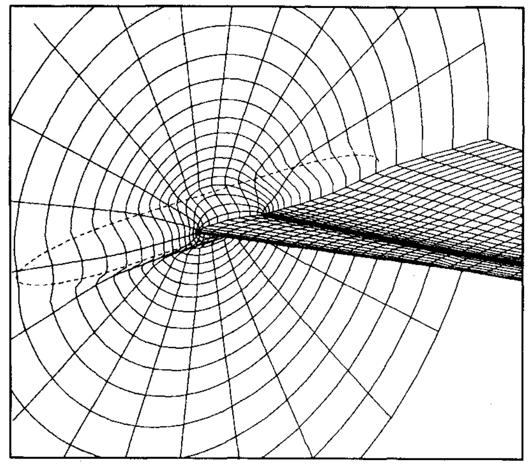

Fig. 10 Three-dimensional grid for wing generated by elliptic solver.

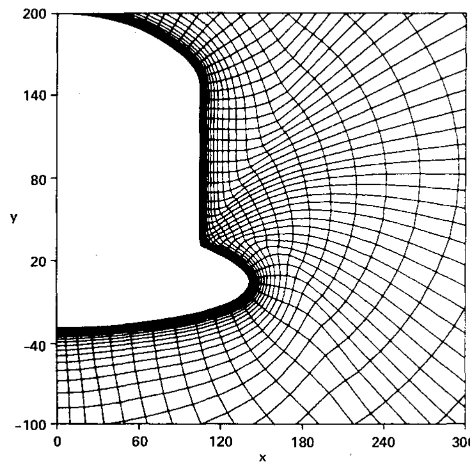

Fig. 11 Three-dimensional grid for shuttle cross section generated by hyperbolic solver.

the flow problems become more and more complicated, computer graphics will play a more important role in the discretization process to facilitate use and implementation.

Turbulence Modeling

It is realistic to state that the simulation of the dynamics of turbulence is today unattainable because of the limited available computational power. That is, available computer speed and storage preclude significant resolution of a sufficiently broad range of interacting turbulent scales (both spatially and temporally) associated with the aerodynamic flowfield (Fig. 13). As a result, modeling has taken the approach of single-point closure of the Reynolds-averaged Navier-Stokes equations, thereby eliminating much of the physical information necessary for simulating the behavior of the turbulence. As of today, no single turbulence model exists that can be applied to a general variety of flows. Limited computer resources also constrain finite difference simulations of the Reynolds-averaged Navier-Stokes calculations to extreme coordinate distortion in one direction, that is, a very fine grid can only be introduced in the direction nearly normal to the body (thin-layer approximation). This permits simulation of the viscous effects in thin layers near the body but at high Reynolds numbers in that direction only. The resulting computational process can qualitatively simulate separated flows and flows with large-scale unsteady behaviors, but the accuracy of such simulations is still controversial.

In 1980, Chapman[2] listed turbulence modeling as a primary pacing item for Reynolds-averaged Navier-Stokes computations. He suggested that existing models are adequate for flows with small regions of separation but that significant improvements in the turbulence models are needed for more realistic simulation of flows with 1) large regions of separation, 2) transitional-type separated flows involving transition between separation and reattachment, and 3) hypersonic flows.

In recent survey papers on turbulence modeling for computational aerodynamics, Marvin[27] and Mehta and Lomax[13] provided valuable insights as to the current state of the art in this area, as well as what will be required in the future. Since no universal model exists, most researchers are now focusing their attention on creating a catalog of models based on fundamental building-block experiments; each model being carefully tested computationally to determine its capabilities and limitations (Fig. 14). In essence, the currently used turbulence model in conjunction with the numerical procedure is "tuned" for a specific class of flow problems. This is a little disconcerting considering the large class of flows that exist in the real world.

Finally, it must be emphasized that the grid clustering, the metric evaluation, the intrinsic parameters (such as mixing length, turbulent energy, and stress), and the numerical algorithm with its associated dissipation, in fact, all together make up the turbulence model. Thus, as Mehta and Lomax[13] point out, the contributions from all of these factors contribute to the computational solution, and its comparison with the experiment depends, in part, on the choice of any one of these factors.

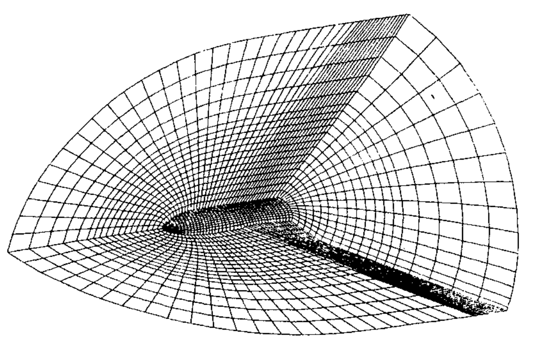

Fig. 12 Three-dimensional grid for wing-body configuration generated by parabolic solver.

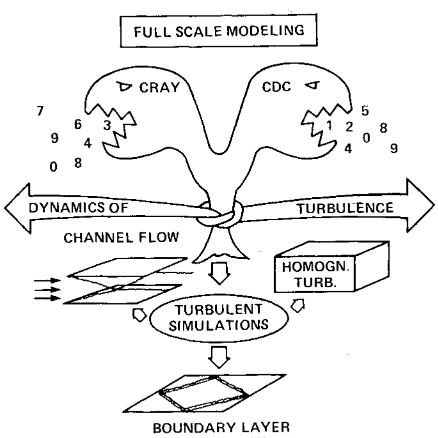

Fig. 13 Turbulence simulation.

Significant progress has been made in the area of turbulence modeling in view of the fact that the Reynolds-averaged Navier-Stokes equations started receiving serious attention only about eight years ago. It goes without saying that a considerable amount of research remains to be done in the discipline of turbulence modeling, and perhaps the greatest advancements are awaiting the availability of future supercomputers with their increased speeds and memories.

Solution Methodology Development

The computational efficiency of a given computer code is dependent on a number of factors. These include the numerical algorithm used to solve the governing equations, the computer system used to solve the finite difference representation of the equations, special programming techniques for efficient coding on the particular computer system, special treatment at the computational boundaries, local time-step advancements, convergence extrapolation procedures, and optimal nodal point distribution (Fig. 15). All of these factors are being addressed by the CFD research community, and more efficient computational simulations are being developed as a result. Several of these factors are categorized as primary pacing items and will be discussed here; others will be addressed in the section on secondary pacing items.

The term solution methodology in this context is defined as any scheme or procedure that, when coupled with the numerical method, enhances the accuracy and convergence properties of the global solution procedure. The development of this area of CFD technology offers the computational scientist a chance to exhibit a new thrust in originality and creativity. The computational efficiency and accuracy to be gained by improvements in numerical algorithm development is limited. But there are other means available to accomplish these objectives. Some, designated as computational efficiency factors, were mentioned earlier. Included are a few solution methodologies that currently are receiving considerable attention within the CFD community because of their potential for considerably enhancing the efficiency and accuracy of the overall solution procedure.

Because of this slowing trend in the creation of new numerical methods, enhanced convergence acceleration and solution accuracy via this mode have dampened. To progress more rapidly, it will be necessary to devise other means of accelerating convergence and enhancing the accuracy of the overall solution procedure, thus, the creation of new solution methodologies is a primary pacing item in CFD. Some of the more attractive solution methods include 1) zonal procedures, 2) hybrid methods (i.e., those that operate off a base algorithm, e.g., multigrid methods, grid embedding panel methods, and correction methods), and 3) solution adaptive techniques.

A zonal method is defined as a solution methodology that numerically simulates a complex flow region by utilizing different equation sets and associated solution algorithms in

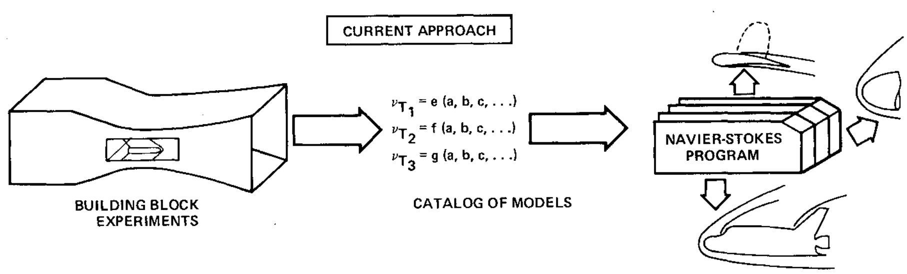

Fig. 14 Turbulence modeling.

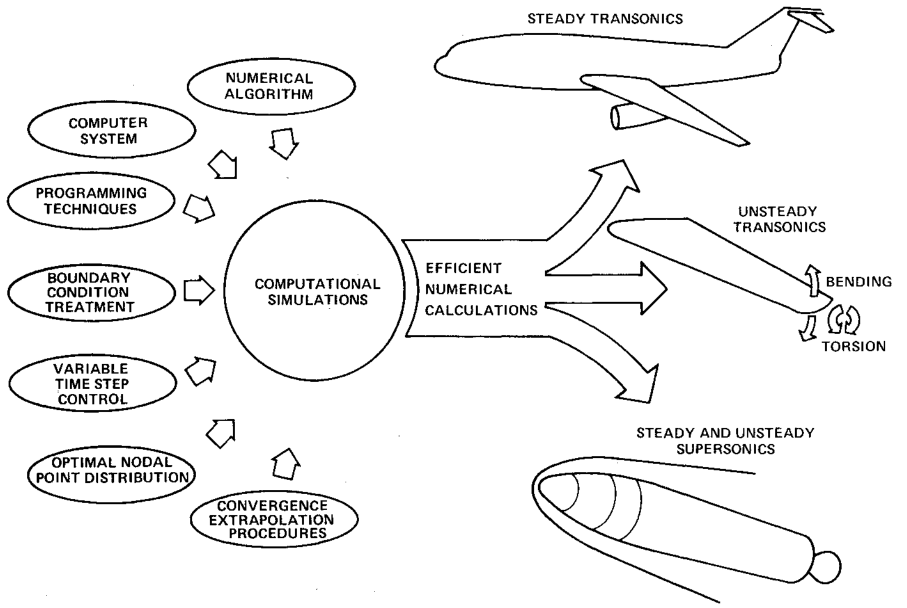

Fig. 15 Computational efficiency factors.

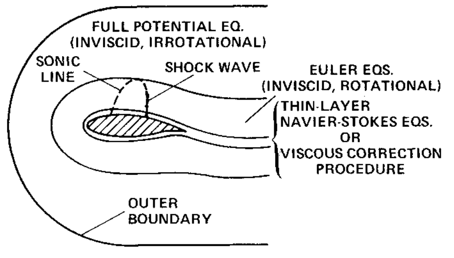

Fig. 16 Zonal method for real flow about airfoil.

their respective flow regions of applicability. A simple example which demonstrates a zonal procedure is shown in Fig. 16. In this problem, the real flow about an airfoil is desired. The flow region is divided into three zones: a thin viscous region near the body, an intervening larger inviscid rotational region, and, finally, outside an inviscid irrotational region which extends to the outer boundary. In the first region the thin-layer Navier-Stokes equations are used, in the second region the Euler equations are used, and in the third region the full-potential equations are used.

Zonal methods are also directionally independent, that is, flow regions can be blocked off in any of the coordinate directions, and the appropriate equation sets applied. Figure 17 shows an example in which the flowfield about a commercial aircraft is subdivided into regions, each of which requires a different equation set to adequately simulate its associated flow behavior. For example, the flow in the canopy region (where flow separation might occur) and in the wing-body juncture region (where viscous interference flow might dominate) requires use of the Reynolds-averaged Navier-Stokes equations to accurately model the flow. Attached to the Navier-Stokes regions are either regions governed by the full-potential equation with viscous corrections or regions adequately described by the small-disturbance equations.

The use of zonal methods is advantageous because 1) the regions over which the complex equation sets are required are minimized, 2) the simpler equation sets have faster solution algorithms which results in improved computational efficiency, and 3) the computer storage requirements for the overall procedure are not as great. Some of the research topics that must be studied before widespread acceptance of formidable zonal methods will occur are 1) requisite boundary and initial condition procedures at interface boundaries, 2) stability and convergence properties of the overall procedure, and 3) minimization of computer program complexity.

Two examples which demonstrate a variant of the zonal method are shown in Figs. 18 and 19. In the first example, a complex of computer codes has been assembled by researchers at Lewis Research Center (see Ref. 28) to analyze the flow in a nonaxisymmetric nozzle. A brief description of each program and where it is applied in the nozzle region is shown in the figure. In the second example, three computer codes were linked by researchers at Ames Research Center (see Ref. 29) to compute the viscous supersonic flow about realistic flight vehicles. They include an unsteady Navier-Stokes solver for computing the flow near the blunt nose of the vehicle, a viscous unsteady continuation code for computing the flow in regions where streamwise separation occurs or where embedded subsonic flow exists, and, finally, a parabolized Navier-Stokes program for determining the viscous flow where the flow is attached and supersonic.

Another solution methodology that is receiving widespread attention is the procedure referred to in this paper as a hybrid method. It is a procedure that utilizes a very successful numerical algorithm as a basis to which is added another sequence of numerical recipes, the combination of which usually enhances the convergence acceleration, the accuracy of the global method, or both.

One of the most popular and successful hybrid methods is called the multigrid method. In essence, it is based on a process of residual reduction resulting from coarse-grid/fine-grid sequencing. To date, this method has had considerable success when applied to simplified equation sets such as the full-potential equation (see Ref. 30). However, no similarly outstanding results have been obtained when the method has been applied to either the Euler or Navier-Stokes equations. It does, however, have the potential for accelerating the convergence of those equation sets when the transient solution is of no interest; thus, the method is being exploited by researchers. A symposium held at Ames Research Center in October 1981 was devoted to multigrid methods. The published proceedings[31] contain an international sampling of the most recent developments in multigrid procedures.

Another new hybrid method that has the potential of computing the transonic flow about a complete aircraft configuration was introduced by Johnson et al.[32] The base program is a subsonic panel-method procedure. The method employs the geometry that already exists for the panel program and uses rectangular grids and subgrids together with embedded surface panels on which boundary conditions are applied. A least-squares procedure is used to reduce the solution of either the Euler or full-potential equations to the solution of a sequence of Poisson problems. Fast Fourier transforms and panel-influence-coefficient techniques are used to solve the Poisson problems. In Ref. 32 it is shown that Poisson iterations can solve mixed-type fluid-flow problems. The scheme has been tried on some simple two-dimensional problems.

The computation of three-dimensional flows containing discontinuities such as shock waves and slip surfaces is still the subject of much research. Scientists are still searching for the ultimate shock-capturing technique. Noteworthy work in this area is being performed by Saltzman and Brackbill[33] at the Los Alamos Scientific Laboratory, Colella and Glaz[34] at the Lawrence Berkeley Laboratory and Naval Surface Weapons Center, respectively, and by Harten[35] at the Courant Institute of New York University. Some of these workers employ the hybrid schemes referred to in this paper as correction methods, that is, procedures that apply corrective terms to an existing numerical scheme to enhance its shock-capturing properties. The success of such schemes is exemplified by the results shown in the reports just mentioned. Typical of such results is that shown in Fig. 20 (from Ref. 36). The figure shows density contours of the complex Mach reflection region that results from the deflection of a shock wave striking an inclined surface. The shock and slip surface structure, as well as the continuous regions of the flow, is resolved quite well by the method.

Regardless of the size of the computational facility available to the user of CFD he will always require more storage. In this vein, he constantly searches for ways to use efficiently the storage capacity of the computer and, hence, the maximum number of grid points available to him. This is one of the main reasons for the popularity of solution adaptive procedures, another being the better accuracy to be gained. Solution-adaptive procedures are schemes that readjust the location of the grid points as the solution evolves to cluster points in regions of rapid change of the flow variable gradients. A survey paper on this area is presented by Thompson.[37] A number of papers on adaptive systems are given in Ref. 12: a noteworthy one was written by Saltzman

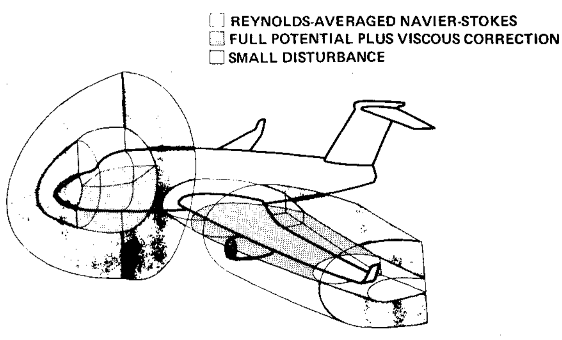

Fig. 17 Directional independence of zonal method.

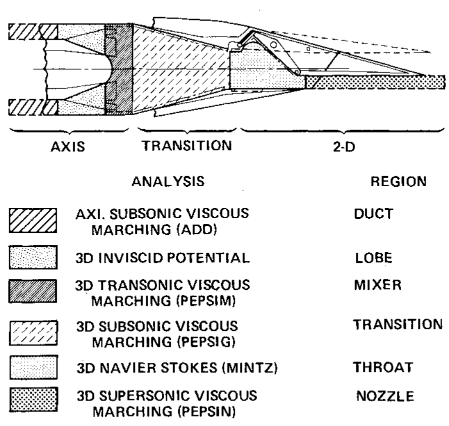

Fig. 18 Zonal method for nonaxisymmetric nozzle system.

and Brackbill.[33] A typical result from their paper is shown in Fig. 21, which displays pressure contours for the supersonic flow over a forward-facing step constrained by an upper wall. Clearly the solution-adaptive mesh conforms to the shock structure.

Advances in Mainframe Computer Methods and Architecture

Because of the demands for more speed and storage placed on a computational system for simulating turbulence dynamics, technological advances in computer design was also listed as a pacing item by Chapman[2] for turbulent eddy simulations. Today, however, for the same reasons, it is also a pacing item for large-scale, Reynolds-averaged Navier-Stokes simulations. To attain the goal of computing the realistic flow about complex three-dimensional configurations, more computational power (i.e., greater speed and storage) than is currently available or planned for the near future will be required. In striving to get the most for one's computing dollar, not only is a large effort being expanded to reduce the number of operations required by the numerical method, but also computer architecture methods are being explored to maximize the number of operations that can be carried out concurrently on the computer. The task of obtaining the most from a computing dollar will require the collaborative effort of both the algorithm developers and the computer architects (see Fig. 22).

Estimates of the computational requirements for simulating the realistic flow about a complete flight vehicle specify a machine with a speed of at least 1000 MFLOPS (million floating point operations per second) sustained on real codes and a memory of 288×10^6 words (32×10^6 words of main memory, and 256×10^6 words of secondary memory). These estimates, arrived at by experienced users of CFD were originally selected as the criteria for design specifications for the NAS (a supercomputer planned by NASA). The largest currently available machines (class VI) such as the CYBER 205 (8×10^6 words of main memory, 32×10^6 words of secondary memory, 80 MFLOPS sustained) and the Cray 1S (4×10^6 words of main memory, 32×10^6 words of secondary memory, 45 MFLOPS sustained) are not satisfactory. The Illiac IV (0.13×10^6 words of main memory, 12×10^6 words of secondary memory, 35 MFLOPS sustained), which is now extinct, was also unable to satisfy the criteria (see Fig. 23). A detailed discussion which covers the characteristics and architectural features of these machines and others can be found in a very comprehensive article written by Levine.[38]

In the past, the economic incentive for manufacturers of large-scale scientific computers was lacking (compared with the demand for business and personal computers), a direct result of the basic law of supply and demand. However, in today's market, the demand for these big "number crunchers" is increasing, a direct result of the realization that progress in a wide range of technical disciplines (e.g., computational chemistry, meteorology, seismology, and nuclear and plasma physics) is dependent on those machines (see Ref. 39). As such, American as well as Japanese scientific computer manufacturers are designing and producing new machines for use not only in national research laboratories but also in universities and industrial organizations around the world. The success of their efforts will obviously depend on how quickly they can design new machines to take advantage of the new technologies and methods that are affecting the scientific computer industry.

NASA is planning to develop a computational system known as the Numerical Aerodynamic Simulator (NAS); it will eventually satisfy the 1 GFLOP (Gigaflop, 10^3 MFLOPS) requirement. The primary objectives of the NAS project are to design and develop a unique large-scale, high-performance, computational resource for solving viscous, three-dimensional, fluid-flow equations specially oriented toward the solution of aerodynamic and fluid dynamic

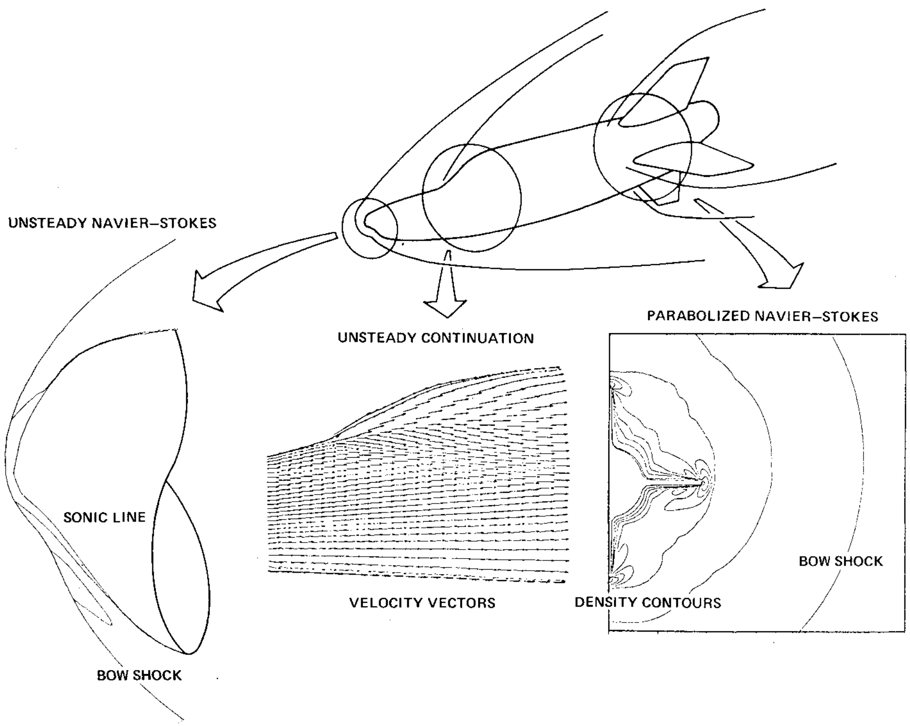

Fig. 19 Zonal method for viscous supersonic simulations.

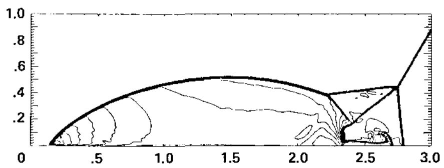

Fig. 20 Density contours for planar shock diffraction resulting in complex Mach reflection.

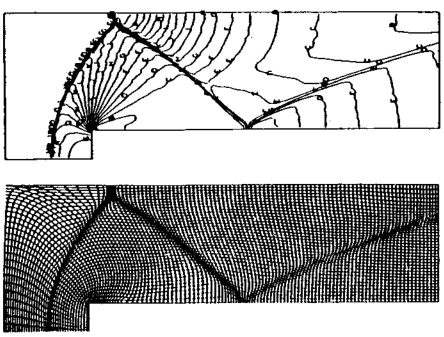

Fig. 21 Pressure contours and grid plot from solution adaptive procedure.

problems. The resource will be a complete facility, including ancillary support such as user and graphics work stations and satellite communications. A secondary objective of NAS is to generalize the computational resource for application to a broader scope of problems of interest to NASA. The present approach envisioned for satisfying the NAS objectives is based on a phased evolutionary development utilizing advanced state-of-the-art hardware and software. Such an approach will provide for an early operational capability. This plan also permits sufficient flexibility to implement the newest in capability in scientific computing machines and, thus, accommodates multiple vendors. The system is also designed to permit nationwide user access to the NAS.

The demand by the computational scientist for bigger and faster supercomputers for attacking and solving the problems at the farthermost limits of their disciplines will never cease. As mentioned earlier, several computer manufacturers are cognizant of this and are continuing to produce machines in an attempt to meet these needs. A summary of existing as well as soon-to-be-available scientific computing systems is shown in Table 1. For comparison purposes, the capabilities of the CDC 7600 and Illiac IV are listed. At the bottom row of the table are listed the known desired statistical characteristics for the NAS. The statistics of the Cray 1 and CYBER 205 were discussed earlier and are not repeated here.

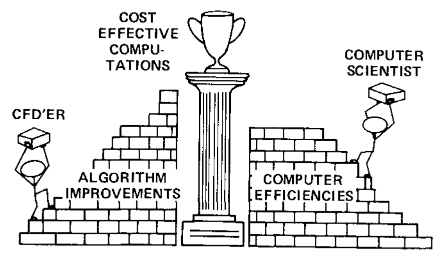

Fig. 22 Optimization of the computing dollar.

An important near-term advancement is that of Cray Research, which introduced, in the second quarter of 1983, the Cray X-MP. It is a dual-processor machine with a two-pipe capability per processor operating on a common memory. It has a maximum vector rate of 470 MFLOPS and a main memory size of 4×10^6 words. Cray is also building the Cray 2, which should run 6 times faster for scalar operations and 12 times faster for vector operations than the Cray 1. It has four processors and, to dissipate the heat, each is immersed in a cooling tank filled with an inert liquid. The size of the computer (38-in. diameter by 26 in.) is about one-third that of the Cray 1.

Control Data Corporation plans to introduce, in the third quarter of 1985, the GF-10 (CYBER 2XX). Its performance statistics are summarized in Table 1. It will have two to four processors, an 8-ns clock cycle, and a 1 GFLOP sustained capability.

Two Japanese computer manufacturers, Fujitsu and Hitachi, introduced their machines in the last quarter of 1983.

The statistics available on these machines (see Table 1) show them to be very competitive with the Cray 2 and CDC CYBER 205. In the future, Japanese computer engineers hope to build a supercomputer with a 10 GFLOP capability by the 1990's. The Japanese government's Ministry of International Technology and Industry, as well as individual Japanese companies, has pledged $700 million for its development.

The speed of scientific computers and the years in which they were introduced are shown graphically in Fig. 24. The associated range bars for each machine indicate sustained and maximum obtainable rates for a given computer. Also plotted is an estimate created by the Los Alamos Scientific Laboratory in 1976. Based on existing and near-term computers, the Los Alamos projection is holding true to form. The NAS, a 1 GFLOP machine for satisfying computational aerodynamic requirements, is shown in Fig. 24 as a darkened circle; it is projected to become operational in 1988. The current trend in computer progress seems to indicate that the NAS goal is realistic.

As user demands for supercomputers increase, the computer architects will devise new ideas for satisfying those demands (including the use of artificial intelligence, in some instances to actually have computers design computers). With the advent of these new machines, the computational scientist will have to 1) be aware of the machine architecture and use it effectively in the design of his software, and 2) be prepared to reduce the enormous amounts of data produced by such machines.

Secondary Pacing Items in Computational Fluid Dynamics

Of importance to the users of CFD are three secondary pacing items which are currently receiving attention within the research community. They are (Fig. 25) 1) algorithm development, 2) complex geometry definition, and 3) predata and postdata processing.

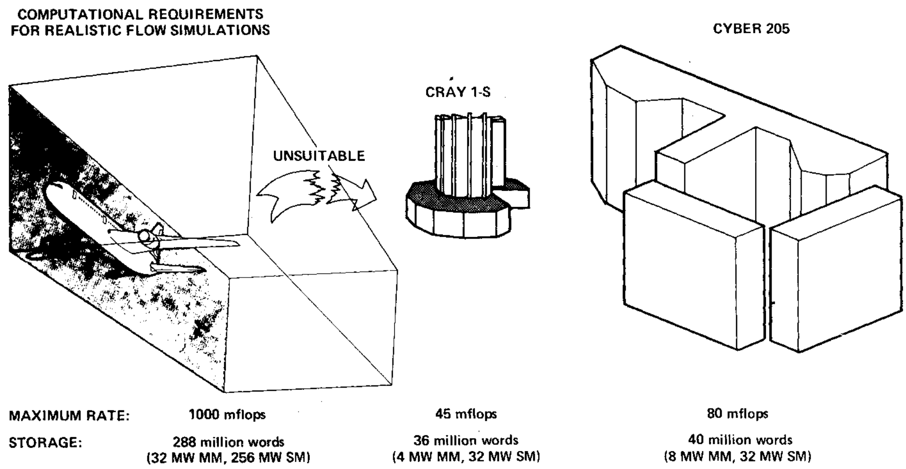

Fig. 23 Required power vs existing capability.

Table 1 Existing and proposed mainframe computing engine characteristic

Processor	Clock cycle, ns (10^{-9} s)	Memory size (megabytes/64 bit megawords) Main	Secondary	Maximum/ sustained vector rate, MFLOPS	No. of processors	No. of pipelines	Date available
CDC 7600	27.5	0.45 mb/0.06 mw	4 mb/0.5 mw	12/4	1	1	Currently
ILLIAC IV	80.0	1 mb/0.13 mw	96mb/12 mw	NA/35	64		Extinct
Cray 1	12.5	32 mb/4 mw	256 mb/32 mw	160/45	1	1	Currently
Cray X-MP	9.5	32 mb/4 mw	256 mb/32 mw	470/100	2	1×2	2nd quarter 1983
Cray 2	4-6	NA	NA	2000/400	4	NA	4th quarter 1983
Cyber 205	20.0	64 mb/8 mw	256 mb/32 mw	800/160	1	4	Currently
Cyber 2XX[a]	8	256 mb/32 mw	2048 mb/256 mw	2000/1000[c]	2-4	8/Processor	3rd quarter 1985
Fujitsu VP 200	15	256 mb/32 mw	None	500/NA	1	4[b]	4th quarter 1983
Hitachi S-810-20	NA	256 mb/32 mw	NA	630/NA	NA	NA	4th quarter 1983
NAS goal	?	256 mb/32 mw	2048 mb/256 mw	?/1000	?	?	1988

[a] System specifications are related to NAS flow model processor studies.
[b] Two pipelines clocked at 7.5 ns giving the effect of four.
[c] Per processor.

Algorithm Development

A prominent pacing item in computational simulations (and one originally listed by Chapman[2] as primary) is algorithm development. In the past 15 years, algorithm efficiency for the inviscid potential equation has increased at an average rate of about an order of magnitude every 10 years and for the Reynolds-averaged Navier-Stokes equations about two orders of magnitude every 10 years. According to Chapman,[2] this has nearly kept pace with the improvements obtained in computer hardware efficiency. The efficiency of a numerical algorithm for a particular equation set is bounded, however, but that limit has not as yet been reached. If one were to predict the improvement to be gained in algorithm efficiency over the next 15 years based on past history and considering the fundamental limits of finite difference representations, one and possibly two orders of magnitude improvement can be expected with the most dramatic improvements occurring for steady-state Navier-Stokes solvers. However, the greatest potential for performing efficient computational simulations lies in the expected advancements to be made in computer technology, as was pointed out in the last subsection.

Table 2 Execution speed comparison for ONERA M6 wing using TWING computer program; $M_\infty = 0.84$, $\alpha = 3.06$ deg

Code	FLO28	TWING	VTWING	VTWING
Computer	CDC 7600	CDC 7600	CDC 7600	Cray 1-S
Field points	53,760	40,050	40,050	40,050
Surface points	1,350	1,513	1,513	1,513
Time for 98% lift, s	742	64	53	4.8
Speed factor	155	13	11	1

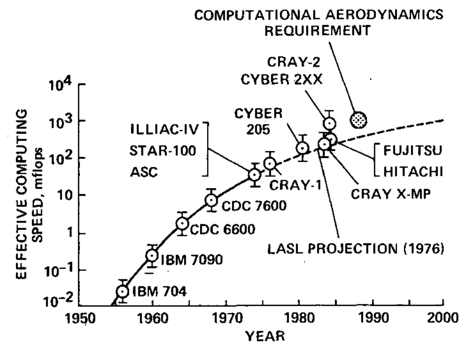

Fig. 24 Trend of effective speed of general-purpose computers.

The detailed history of numerical algorithm development is too lengthy to cover completely in this paper. Worthy of note, however, are a few major milestones that have significantly affected the progress of CFD. For time-accurate, finite difference or finite volume calculations, Lax's first-order method[40] offered the user of CFD the ability to compute accurately flows with shocks. The second-order Lax-Wendroff schemes,[41] of which MacCormack's scheme[42] is a variant, permitted increased resolution of the entire flowfield, especially in the vicinity of the shock wave. An attribute of MacCormack's method was that it permitted the use of non-Cartesian coordinate systems without the need for complicated correction factors to maintain the freestream. A recent algorithm advancement for solving the Euler equations is a procedure developed by Jameson et al.[43] which employs a fourth-order Runge-Kutta technique with both second- and fourth-order stability-enhancing smoothing terms. Solution of the Euler equations is regaining its popularity over relaxation schemes and approximate-factorization procedures for solving the full-potential equation because the assumption of irrotational flow is eliminated. This results in a more physically realistic flowfield simulation and does not require the code designer to specifiy a priori the location of all slip surfaces and lines of separation as is required with a full-potential formulation. In addition, Euler solutions are becoming competitive with full-potential solutions when computer times required to reach steady-state solutions are compared.

As interest grew in solving the Reynolds-averaged Navier-Stokes equations, there was increased need for a new algorithm to eliminate the stiffness introduced as a result of the fine grid required near the surface of the body to resolve the viscous effects. Implicit methods were popularized by such researchers as Briley and McDonald[44] and Beam and Warming.[45] Recently, MacCormack[46] introduced a new composite implicit-explicit method which reverts back to his explicit procedure if the local Courant number (the Courant number is a measure of the time step) becomes smaller than 1 at any mesh point in the flowfield. Implicit methods have also been successfully implemented for solving inviscid flow problems made stiff by the necessity for a fine grid. The use of implicit methods has resulted in increases in convergence rates of one to two orders of magnitude over those of the explicit methods.

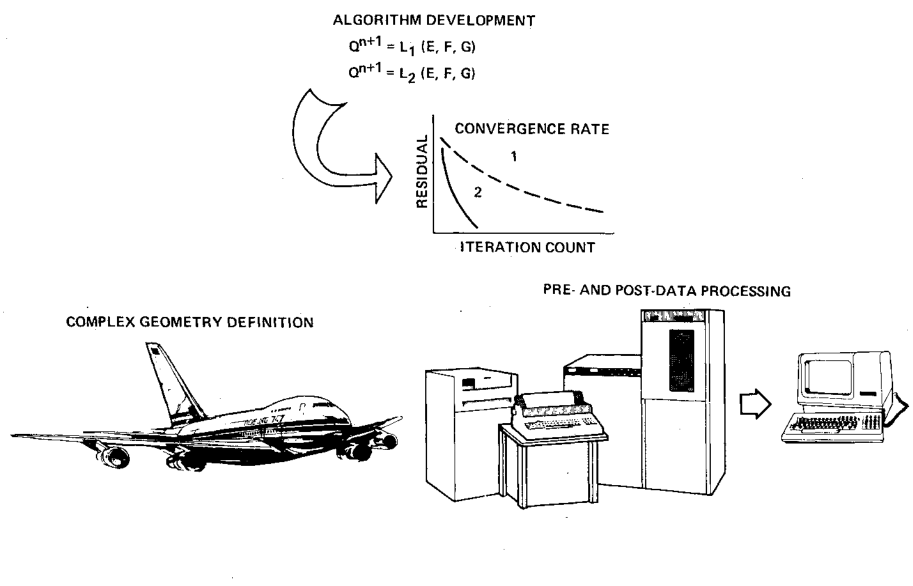

Fig. 25 Secondary pacing items in computational fluid dynamics.

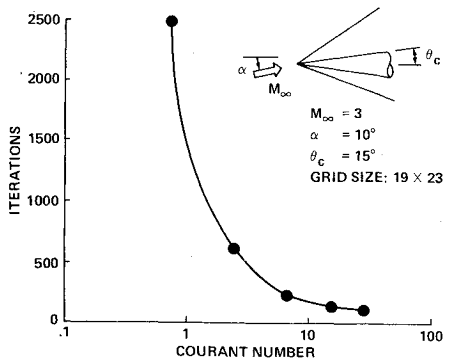

Fig. 26 Convergence acceleration for inviscid supersonic cone flow using PNS program: $M_\infty = 3$, $\alpha = 10$ deg, $\delta = 15$ deg.

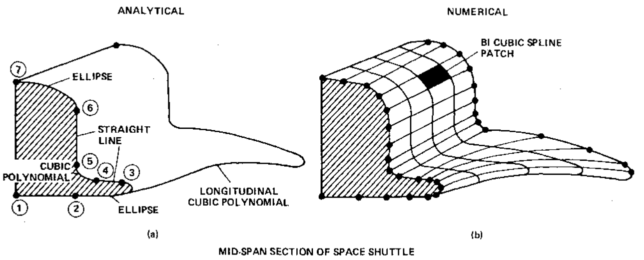

Fig. 27 Complex geometry definition: a) analytical procedure; and b) numerical procedure.

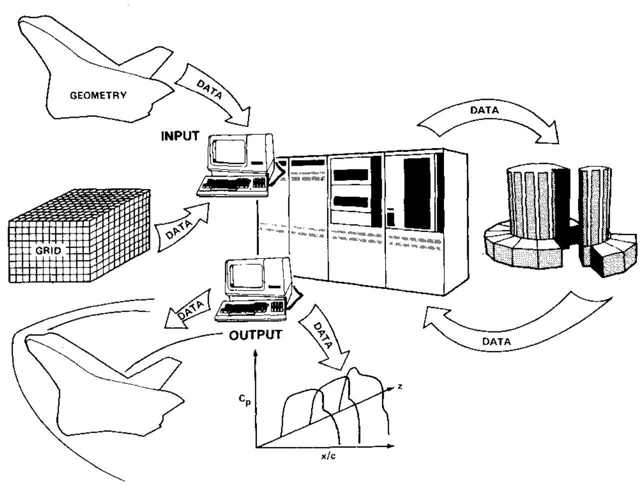

Fig. 28 Predata and postdata processing.

Two excellent examples that demonstrate the dramatic increases in computational capability gained by improvements in some of the factors depicted in Fig. 15 are the transonic flow about a three-dimensional wing and the supersonic flow about a pointed cone. In the first example, the full-potential equation was solved by Thomas and Holst,[47] using an approximate-factorization scheme (AF2). The configuration was an ONERA M6 wing in Mach 0.84 flow at an angle of attack of 3.06 deg. This case has been used by a number of researchers for testing new methods. The equations were originally programmed on the CDC 7600 (resulting in a program named TWING) and then programmed for the Cray 1S computer system (program named VTWING), taking advantage of the vector architecture. The results (see Ref. 47), both numerical solution and convergence history, were compared with a computer code (named FLO28) based on the SLOR approach of Caughey and Jameson[48] for solving the same problem. The execution speed comparison is shown in Table 2. From this chart it can be seen that more than an order-of-magnitude speed increase resulted from utilization of the AF2 algorithm. Another order-of-magnitude increase in speed was obtained by vectorizing the program and running it on the Cray 1S. Because this program converges so quickly, it has become an attractive tool for coupling with a numerical optimization routine for performing three-dimensional wing design studies.

In the second example, the inviscid supersonic flow over a pointed cone at angle of attack was obtained by Rai and Chaussee,[49] using a fully implicit spatially marching procedure. The flow conditions consisted of a 15-deg cone in Mach 3 flow at an angle of attack of 10 deg. Figure 26 shows the number of iterations vs Courant number for the solution to reach a steady state for this case. Operating the program at a Courant number of 23 required slightly over 100 iterations; nearly 2500 iterations were required at a Courant number of 0.7 (the typical Courant number required by explicit procedures).

There has been remarkable progress in numerical algorithms suitable for the solution of both the inviscid and viscous flow equations in the last 30 years. A discussion on the potential for a new generation of numerical methods that will accompany the advances in computer technology, as well as a short history of existing procedures, can be found in a paper by Lomax.[50] In addition, Mehta and Lomax[13] surveyed existing procedures for solving the Reynolds-averaged Navier-Stokes equations for transonic flows. It will probably not be possible to devise a numerical algorithm efficient enough by itself that will encourage the design aerodynamicist to use Navier-Stokes solvers in their routine day-to-day design process. However, advances in numerical algorithms collectively with the expected improvements in all of the other computational efficiency factors depicted in Fig. 15 should result in Navier-Stokes solvers being regarded as a designer's most valued tool.

Complex Geometry Definition

As the users of CFD begin to solve more and more complicated flow problems involving complex configurations, the demand grows for sophisticated geometry definition routines that are easy to use and require a minimal amount of computer time. Considerable work in this area has been done in the past but not necessarily with emphasis for applications in CFD. It is, thus, mandatory that the users of CFD not atempt to reinvent the wheel, but whenever possible utilize existing technology in the form of both software packages and hardware systems to simplify their jobs. However, if that existing technology does not satisfy their particular sets of criteria, and it can be accomplished in a more efficient way (which is the case in most instances), then that technology will have to be created. Thus, complex geometry definition becomes a secondary pacing item for CFD simulations.

Basically, there are two computer-compatible procedures for describing the geometry of any particular flight vehicle: analytical or numerical. Analytical procedures utilize various combinations of algebraic line segment formulas (such as straight lines, ellipses, polynomials, etc.) to approximate the configuration shape as closely as possible (while maintaining continuity of the first- and, in some instances, second-order or higher derivatives of the functions). Once the analytic formulas have been linked, they can be differentiated to obtain the necessary derivative expressions required by flow codes. An example of such a procedure can be found in a paper by Kutler et al.[51] in which the Space Shuttle geometry was modeled analytically (see Fig. 27a).

Numerical procedures for defining geometries rely on discrete data describing the body to be modeled (see Fig. 27b). This can be in the form of digitized cross-sectional data at discrete longitudinal stations or in the form of blueprints or drawings from which the required data can be obtained. Once the digitized data exist, they are made continuous using various surface-fitting approaches, for example, parametric cubics. Such sophisticated procedures are used routinely in conjunction with panel methods or structural finite element procedures. Use of these same procedures for finite difference applications simply requires a transition program to transform the existing coordinate system into the desired coordinate system if different from the first. Numerical procedures are probably more attractive than analytical schemes because of the ease with which they can be used to define complicated configurations. In addition, the surface definition is created interactively from a graphics work station and can, thus, be monitored as the development proceeds.

Without a doubt, computer graphics has played and will continue to play an integral part in the geometry definition process. Color graphics terminals capable of manipulating the image on the screen can rotate, translate, and zoom in on the image to reveal flaws or glitches in the geometric representation of the flight vehicle.

Predata and Postdata Processing

The present decade is proving to be one of three-dimensional flow computations—both inviscid and viscous. The solution of these three-dimensional problems on supercomputers involves the management of enormous amounts of not only input but, primarily, output data. This can significantly affect the time of the researcher who has to handle that data, make decisions about it, and formulate new avenues of approach to solve his given problem. As a result of the need to efficiently manage this quantity of data, the development of optimal predata and postdata processing procedures is a secondary pacing item in CFD. Preprocessing, intermediate processing, and postprocessing of bulk data can only be done effectively using high-resolution, high-throughput computer graphics devices. Thus, the efficient utilization of onsite supercomputers will necessitate networking them with peripheral minicomputers linked with sophisticated interactive graphics work stations (Fig. 28).

Generation of three-dimensional data poses yet another significant problem for the scientist, namely, how to display it for optimum understanding and analysis. This is where the CFD user requires either an artistic ability not normally acquired in a typical college engineering curriculum or a graphics display engineer with a creative imagination.

Graphical display devices and supporting software are also available to generate three-dimensional color movies (film and video) which are invaluable for visualizing an evolving sequence of fluid-dynamic events and for the effective presentation or description of those events to the technical community. In addition, machines are also available for instantaneously producing hard copy directly from the graphics device for a single color copy or for permitting instant viewing of a color movie from video disk or tape. However, the production of such color movies is expensive. Full-color, three-dimensional hidden line movies will require significant computer time (minutes on a Cray), but probably an insignificant amount compared to the computer time required to generate the data from which the movie is made. However, the benefits to be gained from such a movie justify its production cost.

User Demands on CFD Computer Codes

The sophistication of complicated computer codes to be developed in the future will require implementation of software development standards and management of the software development process. Until recently, fluid-dynamic computer codes were generally developed by a single researcher, were very difficult to modify, and were often not reliable (i.e., when applied to slightly different problems or considerably different flow parameters). It is expected that computer codes to be designed and utilized on supercomputers will be constructed not by a single researcher but by a team of researchers and programmers possibly utilizing concepts from the discipline of artificial intelligence such as expert systems and computer vision devices. The constraints (e.g., usability, affordability, maintainability, reliability, flexibility, portability, and cost-effective development will still dominate the code development process (these assume the proposed program adequately simulates the true physical behavior of the flow). It is, thus, mandatory that before these codes are exported to the user community they be carefully tested, well documented, and certified by in-house user groups. That is, the true capabilities and limitations of the code must be exhibited, and their operational cost and performance must be specified.

Miranda[52] presents a formula that describes the effectiveness of CFD software in the flight vehicle design process as a function of a computer code's characteristics. The formula defines effectiveness as follows.

$$\text{Effectiveness} = \text{Quality} \times \text{Acceptance}$$

The quality factors are solution accuracy and physical realism. The acceptance factors are applicability; usability, i.e., simple, reliable, flexible; affordability (manpower and computer); and maintainability. As Miranda points out, changes affecting these factors should be carefully weighed to determine if the resulting trade-offs hinder or enhance the overall effectiveness of the resulting software.

Associated with the development of the modern CFD software operating on today's supercomputers and in some instances being designed to take advantage of a given machine's architecture, there arises the problem of transportability. It can be expected that potential customers of this desired CFD technology will not have available for their use the supercomputer on which these codes were originally designed and first operated. Aware of this fact, the modern software designer should provide the user with an option to permit him to designate the machine on which the code is to be executed. An example of this, in which such an option is available, occurs in the TWING code (some numerical results from this code were mentioned earlier), which determines the three-dimensional transonic flow about wings using a full-potential analysis. This program was designed to operate at the user's discretion on one of three computer systems located at Ames Research Center: the VAX 11/780, CDC 7600, or Cray 1S (see Ref. 47). Software flexibilities such as these will facilitate the problem of transportability.

Concluding Remarks

The increasing popularity and growth of the discipline of computational fluid dynamics over the past decade has been truly remarkable. The future of CFD for the next decade also looks bright, although there will be some rather strong challenges to meet. With the improvements in computational techniques and advances in computer technology, CFD researchers must and will find more effective ways of applying computational tools in the design and analysis process. With the growing popularity of new design concepts, such as aeroelastic tailoring and swept forward wings, and with multidisciplinary constraints (e.g., aerodynamic, acoustical, structural, and performance) being placed on a proposed configuration, synergistic computational simulations will begin to play a more dominant role in the design cycle.

References

[1] Shakar, V. S. V., private communication, Rockwell International Science Center, Thousand Oaks, Calif., Aug. 1982.

[2] Chapman, D. R., "Trends and Pacing Items in Computational Aerodynamics," *Lecture Notes in Physics*, Vol. 141, edited by W. C. Reynolds and R. W. MacCormack, Springer-Verlag, New York, June 1980, pp. 1-12.

[3] Carr, M. P. and Forsey, C. R., "Development in Coordinate Systems for Flow Field Problems," Aircraft Research Association Ltd., Memo No. 231, March 1981.

[4]Lee, K. D. and Ruppert, P. E., "Transonic Flow Computations Using Grid Systems with Block Structure," *Lecture Notes in Physics*, Vol. 141, edited by W. C. Reynolds and R. W. MacCormack, Springer-Verlag, New York, June 1980, pp. 266-271.

[5]Thomas, P. D., "Numerical Generation of Composite Three-Dimensional Grids by Quasi-linear Elliptic Systems," *Numerical Grid Generation*, edited by J. F. Thompson, North-Holland Publishing Co., New York, 1982, pp. 667-686.

[6]Wedan, B., "A Method for Solving the Transonic Full-Potential Equation for General Configurations," AIAA Paper 83-1889, July 1983.

[7]Atta, E., "Component-Adaptive Grid Interfacing," AIAA Paper 81-0382, Jan. 1981.

[8]Steger, J. L., Benek, J., and Dougherty, F. C., private communication, Stanford Univ., Stanford, Calif., Oct. 1982.

[9]Lasinski, T. A., Andrews, A. E., Sorenson, R. L., Chaussee, D. S., Pulliam, T. H., and Kutler, P., "Computation of the Steady Viscous Flow over a Tri-Element 'Augmentor Wing' Airfoil," AIAA Paper 82-0021, Jan. 1982.

[10]Jou, W.-H. and Huynh, H., "Singularity Embedding Method in Potential Flow Calculations," NASA CR-166387, 1982.

[11]Kwak, D., "An Implicit Transonic Full Potential Code for Cascade Flow on H-Grid Topology," AIAA Paper 83-0506, Reno, Nev., Jan. 1983.

[12]Thompson, J. F., ed., *Numerical Grid Generation*, North-Holland Publishing Co., New York 1982.

[13]Mehta, U. and Lomax, H., "Reynolds Averaged Navier-Stokes Computations of Transonic Flows: the State-of-the-Art," *Progress in Astronautics and Aeronautics, Transonic Aerodynamics*, edited by D. Nixon, Vol. 81, AIAA, New York, 1982, pp. 297-375.

[14]Ives, D. C., "Conformal Grid Generation," *Numerical Grid Generation*, edited by J. F. Thompson, North-Holland Publishing Co., New York, 1982.

[15]Eiseman, P. R., "Geometric Methods in Computational Fluid Dynamics," ICASE Rept. 80-11, NASA Langley Research Center, April 1980.

[16]Thompson, J. F., Thames, F. C., and Mastin, C. W., "Automatic Numerical Generation of Body-Fitted Curvilinear Coordinate System for Field Containing Any Number of Arbitrary Two-Dimensional Bodies," *Journal of Computational Physics*, Vol. 15, July 1974, pp. 299-319.

[17]Barfield, W. D., "An Optimal Mesh Generator for Lagrangian Hydrodynamic Calculations in Two Space Dimensions," *Journal of Computational Physics*, Vol. 6, Dec. 1970, pp. 417-429.

[18]Godunov, S. K. and Prokopov, G. P., "The Use of Moving Meshes in Gas-Dynamical Computations," *USSR Computational Mathematics and Mathematical Physics*, Vol. 12, 1972.

[19]Amsden, A. A. and Hird, C. W., "A Simple Scheme for Generating General Curvilinear Grids," *Journal of Computational Physics*, Vol. 11, March 1973, pp. 348-359.

[20]Sorenson, R. L., "A Computer Program to Generate Two-Dimensional Grids About Airfoils and Other Shapes by Use of Poisson's Equation," NASA TM-81198, 1980.

[21]Mastin, C. W. and Thompson, J. F., "Transformation of Three-Dimensional Regions onto Rectangular Regions by Elliptic Systems," *Numerische Mathematik*, Vol. 29, 1978.

[22]Lee, K. D., Huang, N. J. Y., and Rubbert, P. E., "Grid Generation for General Three-Dimensional Configurations," NASA CP-2166, 1980.

[23]Steger, J. L. and Chaussee, D. S., "Generation of Body Fitted Coordinates Using Hyperbolic Partial Differential Equations," *SIAM Journal of Scientific and Statistical Computing*, Vol. 1, Dec. 1980, pp. 431-437.

[24]Bridgeman, J. O., Steger, J. L., and Caradona, F. X., "A Conservative Finite-Difference Algorithm for the Unsteady Transonic Potential Equation in Generalized Coordinates," AIAA Paper 82-1388, Aug. 1982.

[25]Nakamura, S., "Noniterative Grid Generation Using Parabolic Difference Equations for Fuselage-Wing Flow Calculations," Eighth International Conference on Numerical Methods in Fluid Dynamics, Aachen, Germany, June 1982.

[26]Klopfer, G. H., "Analysis of a Finite-Difference Grid," Nielsen Engineering and Research, Mountain View, Calif., TR 267, Feb. 1982.

[27]Marvin, J. G., "Turbulence Modeling for Computational Aerodynamics," AIAA Paper 82-0164, Jan. 1982.

[28]Bowditch, D. N., private communication, NASA Lewis Research Center, Cleveland, Ohio, Nov. 1982.

[29]Chaussee, D. S., private communication, NASA Ames Research Center, Moffett Field, Calif., Oct. 1982.

[30]Jameson, A., "Acceleration of Transonic Potential Flow Calculations on Arbitrary Meshes by the Multiple Grid Method," AIAA Paper 79-1458, July 1979.

[31]Lomax, H., ed., "Multigrid Methods," NASA CP-2202, 1982.

[32]Johnson, F. T., James, R. M., Bussoletti, J. E., and Woo, A. C., "A Transonic Rectangular Grid Embedded Panel Method," AIAA Paper 82-0953, June 1982.

[33]Saltzman, J. and Brackbill, J., "Applications and Generalizations of Variational Methods for Generating Adaptive Meshes," *Numerical Grid Generation*, edited by J. F. Thompson, North-Holland Publishing Co., New York, 1982.

[34]Colella, P. and Glaz, H. M., "Numerical Modeling of Inviscid Shocked Flows of Real Gases," LBL Rept. 14573, Lawrence Berkeley Laboratory, Berkeley, Calif., June 1982.

[35]Harten, A., "The Artificial Compression Method for Computation of Shocks and Contact Discontinuities: III. Self-Adjusting Hybrid Schemes," *Mathematics of Computation*, Vol. 32, No. 142, April 1978, pp. 363-390.

[36]Woodward, P. and Colella, P., "The Numerical Simulation of Two-Dimensional Fluid Flow with Strong Shocks," in preparation.

[37]Thompson, J. F., "A Survey of Adaptive Grid Techniques in Computational Fluid Dynamics," AIAA Paper 83-0447, Jan. 1983.

[38]Levine, R. D., "Supercomputers," *Scientific American*, Vol. 246, Jan. 1982, pp. 118-136.

[39]Ballhaus, W. F., "Computational Aerodynamics and Design," NASA TM-84257, 1982.

[40]Lax, P. D., "Weak Solutions of Nonlinear Hyperbolic Equations and Their Numerical Computation," *Communications on Pure and Applied Mathematics*, Vol. 7, Feb. 1954, pp. 159-193.

[41]Lax, P. D. and Wendroff, B., "Difference Schemes for Hyperbolic Equations with High Order of Accuracy," *Communications on Pure and Applied Mathematics*, Vol. 17, Aug. 1964, pp. 381-398.

[42]MacCormack, R. W., "The Effect of Viscosity in Hypervelocity Impact Cratering," AIAA Paper 69-354, April 1969.

[43]Jameson, A., Schmidt, W., and Turkel, E., "Numerical Solutions of the Euler Equations by Finite Volume Methods Using Runge-Kutta Time Stepping Schemes," AIAA Paper 81-1259, June 1981.

[44]Briley, W. R. and McDonald, H., "Solution of the Multidimensional Compressible Navier-Stokes Equations by a Generalized Implicit Method," *Journal of Computational Physics*, Vol. 24, Aug. 1977, pp. 372-397.

[45]Beam, R. and Warming, R. F., "An Implicit Factored Scheme for the Compressible Navier-Stokes Equations," *AIAA Journal*, Vol. 16, April 1978, pp. 393-402.

[46]MacCormack, R. W., "A Numerical Method for Solving the Equations of Compressible Viscous Flow," AIAA Paper 81-0110, Jan. 1981.

[47]Thomas, S. D. and Holst, T. L., "Numerical Solutions of Transonic Wing Flow Fields," AIAA Paper 83-0499, Jan. 1983.

[48]Caughey, D. A. and Jameson, A., "Numerical Calculation of Transonic Potential Flow about Wing-Body Combinations," *AIAA Journal*, Vol. 17, Feb. 1979, pp. 175-181.

[49]Rai, M. M. and Chaussee, D. S., "New Implicit Boundary Procedures: Theory and Application," AIAA Paper 83-0123, Jan. 1983.

[50]Lomax, H., "Some Prospects for the Future of Computational Fluid Dynamics," AIAA Paper 81-994, June 1981.

[51]Kutler, P., Reinhardt, W. A., and Warming, R. F., "Multishocked, Three-Dimensional Supersonic Flowfields with Real Gas Effects," *AIAA Journal*, Vol. 11, May 1973, pp. 657-664.

[52]Miranda, L. R., "A Perspective of Computational Aerodynamics from the Viewpoint of Airplane Design Applications," AIAA Paper 82-0018, Jan. 1982.