NASA/TM–2002–211747

Opportunities for Breakthroughs in Large-Scale Computational Simulation and Design

Natalia Alexandrov, Stephen J. Alter, Harold L. Atkins, Kim S. Bey, Karen L. Bibb, Robert T. Biedron, Mark H. Carpenter, F. McNeil Cheatwood, Philip J. Drummond, Peter A. Gnoffo, William T. Jones, William L. Kleb, Elizabeth M. Lee-Rausch, N. Ronald Merski, Raymond E. Mineck, Eric J. Nielsen, Michael A. Park, Shahyar Z. Pirzadeh, Thomas W. Roberts, Jamshid A. Samareh, Roy C. Swanson, James L. Thomas, Veer N. Vatsa, K. James Weilmuenster, Jeffery A. White, William A. Wood, and Long P. Yip
Langley Research Center, Hampton, Virginia

June 2002

The NASA STI Program Office . . . in Profile

Since its founding, NASA has been dedicated to the advancement of aeronautics and space science. The NASA Scientific and Technical Information (STI) Program Office plays a key part in helping NASA maintain this important role.

The NASA STI Program Office is operated by Langley Research Center, the lead center for NASA's scientific and technical information. The NASA STI Program Office provides access to the NASA STI Database, the largest collection of aeronautical and space science STI in the world. The Program Office is also NASA's institutional mechanism for disseminating the results of its research and development activities. These results are published by NASA in the NASA STI Report Series, which includes the following report types:

- TECHNICAL PUBLICATION. Reports of completed research or a major significant phase of research that present the results of NASA programs and include extensive data or theoretical analysis. Includes compilations of significant scientific and technical data and information deemed to be of continuing reference value. NASA counterpart of peer-reviewed formal professional papers, but having less stringent limitations on manuscript length and extent of graphic presentations.
- TECHNICAL MEMORANDUM. Scientific and technical findings that are preliminary or of specialized interest, e.g., quick release reports, working papers, and bibliographies that contain minimal annotation. Does not contain extensive analysis.
- CONTRACTOR REPORT. Scientific and technical findings by NASA-sponsored contractors and grantees.
- CONFERENCE PUBLICATION. Collected papers from scientific and technical conferences, symposia, seminars, or other meetings sponsored or co-sponsored by NASA.
- SPECIAL PUBLICATION. Scientific, technical, or historical information from NASA programs, projects, and missions, often concerned with subjects having substantial public interest.
- TECHNICAL TRANSLATION. English-language translations of foreign scientific and technical material pertinent to NASA's mission.

Specialized services that complement the STI Program Office's diverse offerings include creating custom thesauri, building customized databases, organizing and publishing research results. . . even providing videos.

For more information about the NASA STI Program Office, see the following:

- Access the NASA STI Program Home Page at ***http://www.sti.nasa.gov***
- E-mail your question via the Internet to help@sti.nasa.gov
- Fax your question to the NASA STI Help Desk at (301) 621–0134
- Phone the NASA STI Help Desk at (301) 621–0390
- Write to:
  NASA STI Help Desk
  NASA Center for AeroSpace Information
  7121 Standard Drive
  Hanover, MD 21076–1320

NASA/TM–2002–211747

Opportunities for Breakthroughs in Large-Scale Computational Simulation and Design

Natalia Alexandrov, Stephen J. Alter, Harold L. Atkins, Kim S. Bey, Karen L. Bibb, Robert T. Biedron, Mark H. Carpenter, F. McNeil Cheatwood, Philip J. Drummond, Peter A. Gnoffo, William T. Jones, William L. Kleb, Elizabeth M. Lee-Rausch, N. Ronald Merski, Raymond E. Mineck, Eric J. Nielsen, Michael A. Park, Shahyar Z. Pirzadeh, Thomas W. Roberts, Jamshid A. Samareh, Roy C. Swanson, James L. Thomas, Veer N. Vatsa, K. James Weilmuenster, Jeffery A. White, William A. Wood, and Long P. Yip
Langley Research Center, Hampton, Virginia

National Aeronautics and
Space Administration

Langley Research Center
Hampton, Virginia 23681–2199

June 2002

Available from:

NASA Center for AeroSpace Information (CASI)
7121 Standard Drive
Hanover, MD 21076–1320
(301) 621–0390

National Technical Information Service (NTIS)
5285 Port Royal Road
Springfield, VA 22161–2171
(703) 605–6000

Abstract

Opportunities for breakthroughs in the large-scale computational simulation and design of aerospace vehicles are presented. Computational fluid dynamics tools to be used within multidisciplinary analysis and design methods are emphasized. The opportunities stem from speedups and robustness improvements in the underlying unit operations associated with simulation (geometry modeling, grid generation, physical modeling, analysis, etc.). Further, an improved programming environment can synergistically integrate these unit operations to leverage the gains. The speedups result from reducing the problem setup time through geometry modeling and grid generation operations, and reducing the solution time through the operation counts associated with solving the discretized equations to a sufficient accuracy. The opportunities are addressed only at a general level here, but an extensive list of references containing further details is included. The opportunities discussed are being addressed through the Fast Adaptive Aerospace Tools (FAAST) element of the Advanced Systems Concept to Test (ASCoT) and the 3rd Generation Reusable Launch Vehicles (RLV) projects at NASA Langley Research Center. The overall goal is to enable greater inroads into the design process with large-scale simulations.

1 Introduction

A wide range of NASA missions require aerospace vehicle engineering assessments, which are accomplished by a combination of theoretical, experimental (including wind tunnel and flight test), and computational techniques. Each tool has limitations which preclude it from exclusively providing sufficient information for the intended mission; a judicious interplay among the three areas has yielded impressive results in aeronautics. Spurred by the decrease in computing costs relative to other costs, a continually enlarging fraction of mission analyses rely on computation. For example, large-scale computational fluid dynamics (CFD) solvers for the Euler and Navier-Stokes equations are routinely used to simulate the cruise shapes of transport aircraft through complex-geometry simulations involving the solution of 25–100 million equations; in this arena, the number of wind-tunnel tests for a new cruise wing design has been substantially reduced. However, simulations of the entire flight envelope of the vehicle, including maximum lift, buffet onset, flutter, and control effectiveness, have not successfully eliminated the reliance on wind-tunnel testing. These simulations generally involve more complex flow physics, such as unsteady flows with extensive separation and strong shock waves, and a much wider range of configuration parameters. Thus the total number of wind-tunnel hours associated with new aircraft design has remained virtually constant.

Three major roadblocks limit further inroads of CFD into the design process: (1) the lack of reliability of physical models (e.g., transition, turbulence, or gas kinetic models), (2) the long turnaround time of the numerical simulation, and (3) the lack of reliably automated functions (objectives and constraints) and derivatives for use in design optimization. Because of the prohibitive resolution requirements of direct simulations at high Reynolds numbers, transition and turbulence modeling are expected to remain concerns for the near term. However, there is increasing need for time-accurate simulations intended to compute, rather than model, as many of the larger scale interactions as possible (ref. 1). This paper focuses on addressing the latter two problems: long turnaround time and the current lack of reliably automated (i.e., robust) simulations. It is recognized from experience that improvements in these two areas benefit the reliability of physical models since they enable a much greater range of models to be developed and assessed through comparisons with experiments.

In addressing the current deficiencies in large-scale simulation, the simulation process should be considered a system composed of the following unit operations: geometry modeling, grid generation and movement, physical modeling, analysis, visualization, sensitivity analysis, and optimization. Experience has shown that to have value in an industrial analysis and design environment, the answers must be turned around in a day or, at most, overnight. Individual bottlenecks in the process need to be addressed to a lesser or greater degree depending on the application. Two classes of applications are typically encountered, termed market-driven and technology-driven by P. Rubbert of Boeing (remarks presented at the NASA Administrator's Forum on Supercomputing, Washington, DC, July 1993). Market-driven applications are those in which the physical processes are relatively well understood and the predominant need is to obtain the simulation result as rapidly or as accurately as possible, in order to have a greater impact on the cost and schedule of the project.

Technology-driven applications are those in which the physical processes are not well understood and the predominant need is to gain understanding by using computation in connection with theory and experiment. Examples of the two applications are the cruise design of a subsonic transport aircraft and the heat load calculation for a reusable launch vehicle, respectively.

Alongside the growth of CFD in both these areas in the government, industry, and university sectors, there has been a growth of commercial CFD companies. These CFD companies have concentrated on robust general-purpose analysis methods, which include quite sophisticated physical models to simulate a wide variety of applications. Historically, the accuracy of commercial codes has not been as high as codes developed by NASA for high Reynolds number applications to aerospace vehicles, but the gap is closing and will narrow in the future. Thus, many of the routine application analyses of today may well be contracted to these CFD companies.

NASA predominantly has technology-driven applications, for example to design radically new aerospace concepts, to develop improved physical models, or to design new experimental facilities to supplement or validate CFD. NASA should continue to develop CFD methods targeted at these technology-driven applications. However, because the underlying software for these applications has become quite complex, the software development process needs to change—improvements are needed in modularity, supportability, and the use of open standards such as the CFD General Notation System (CGNS) (refs. 2–4). These improvements, discussed in more detail in the next sections, should create a system with a high degree of interchangeability of components, including usage of third-party, proprietary modules, in order to meet three needs: (1) pioneering development of new aerospace concepts, (2) developing and validating new physical models, and (3) assessing the state of the art and development of new algorithms. As algorithms and physical models are developed, this technology should be transferred to a wide audience with an emphasis on dissemination to the commercial CFD companies. A very successful model of this is the visualization capability in CFD: early technology was spurred by NASA development and now nearly all of the visualization tools are supplied by the commercial sector.

In both market- and technology-driven applications, the most dominant factor is how rapidly the complex geometry, defined from a diverse set of computer-aided design (CAD) standards, can be addressed. For market-driven applications, the next dominant factor is usually the speed of the calculation.

For technology-driven applications, the next dominant consideration is usually the fidelity of the physical model. Ever more powerful algorithms and computers are needed to meet the goal of reducing the overall process time. A key element cutting across many areas is the increased usage of adjoint methods; these methods have been extensively used for many years in the structures and controls disciplines. The adjoint methods have value not only in design optimization but also in error estimation, stopping criteria, and adaptive-grid criteria. Finally, in both market- and technology-driven applications, grid manipulation schemes and simulations that may be robust for the purposes of analysis are usually not robust for the purposes of design optimization. A significant obstacle to design optimization is that the simulations and mesh manipulation schemes are frequently unable to provide objective functions, constraints, and the attendant derivatives over the entire design space in an automated fashion usually required by optimization algorithms.

The paper is organized as follows. The roadblocks, opportunities for breakthroughs, and long-term goals are presented in the selected areas of geometry modeling and grid generation, error assessment and grid adaptation, convergence acceleration, physical models and model synthesis, and design optimization. To realize the opportunities discussed, a much more adaptive approach, in terms of grids, solvers, and physical models, needs to be taken; the software development process is key to a successful implementation and is discussed in the next section. A future scenario is sketched in the next section showing how these elements could interplay in a NASA application. The final section summarizes the impacts of attaining the breakthroughs addressed here.

2 Geometry Modeling and Grid Generation

Both structured- and unstructured-grid methods have been advanced to handle complex geometries. The unstructured-grid methods have a decided edge in two areas: (1) the time to generate grids for new configurations, and (2) grid adaptation. The structured-grid methods are still in wide use because of the faster speed of computation once a grid has been generated. This aspect is especially important for time-dependent applications, design applications, or parametric evaluations of geometry-independent variations, such as Mach number or angle of attack. The cycle time to generate grids using overset structured grids about very complex geometries, such as a transport aircraft in the landing configuration, has been reduced substantially (ref. 5) and practical applications are encountered in which this approach is quite viable in the time to generate grids for new configurations. One of the advantages of structured-grid methods is the ability to easily stretch the grid normal to the body to efficiently resolve viscous flows at a high Reynolds number. This aspect is actually being exploited in unstructured-grid methods with a combination of advancing-layer methods near the surface and advancing-front methods elsewhere. A long-term approach is to exploit hybrid methods, characterized by very regular isotropic cells in the regions away from the surface and a transition to more specialized element types near the surface. For inviscid flows, the use of Cartesian grids with irregular boundary cells is already common. Using mixed elements benefits the construction of accurate and efficient solvers but requires the solvers to handle a richer variety of element types.

The unstructured grid generator VGRIDns (ref. 6) has provided high quality viscous grids for complex geometries in applications at NASA Langley, including high-lift systems, subsonic transports, military aircraft, and the Space Shuttle. The ability to generate grids with smoothly varying element sizes is available through a spacing function obtained by solving an elliptic partial differential equation on a background Cartesian mesh. Anisotropic grid stretching has been demonstrated to be essential in reducing the number of grid points in 3-D applications. The VGRIDns methodology can carve out regions and remesh locally, improving grid quality and enabling an adaptive-grid enrichment capability. The methodology is quite robust for inviscid simulations and viscous simulations utilizing wall functions to model the turbulence equations. The method is not completely robust for viscous simulations integrating turbulence models to the wall and can fail near small scale regions of complex geometries, such as gaps, corners, or thin surfaces. This issue is becoming increasingly important as the demand for grids increases.

Irrespective of the grid generation method, defining an appropriate geometry model from a CAD definition for the configuration can take weeks. Diverse CAD standards are used within the industry; NASA, in its role of assessing technology or problem solving, must contend with this diversity. Typically, the CAD model has been designed to build or construct the configuration rather than from the standpoint of a CFD analysis. For example, the domains may not be watertight, or some geometric features may need to be eliminated from the analyzed geometry.

An open development framework for geometry modeling and grid generation is currently being pursued (ref. 7) within the NASA Langley Geometry Laboratory (GEOLAB). The goal of the framework is to seamlessly integrate alternate geometry modeling and grid generation algorithms by developing an Application Programming Interface (API). The API is a set of rules (standard software interfaces) that define the interaction of different component algorithms. The structure of the API is defined by decoupling and encapsulating the system tasks into separate, largely independent component processes; in this case, geometry (including the connectivity of the surface regions), grid metrics (including grid spacing), and meshing (including edge, surface, and volume) processes. The approach makes available a much richer palette of techniques for complex geometry meshings; this interchangeability is important given that the current methods are not sufficiently robust for viscous applications.

The interface draws heavily from the Computational Analysis PRogramming Interface (CAPRI) model (ref. 8). CAPRI is a CAD-vendor-neutral API that accesses computational solid geometry related information from the kernel of the originating CAD system. This allows a layer of abstraction from the specific (and often proprietary) methods of a given CAD standard and thereby efficient access, without loss of information, to the underlying CAD model by the grid generation processes. This aspect is especially important with adaptation procedures to ensure new surface grid points correctly represent the intended geometry. The CAPRI model is limited to the manifold solid geometries used by modern CAD vendors. By taking advantage of the inherent topological information of the solid model, the grid generation can be automated and much of the setup time required by other descriptions of the domain is avoided. Support of legacy systems requires that other descriptions be accommodated, such as through IGES specification, in which case topology information needs to be provided. A prototype derivative application of this framework, GridEx, is under development (ref. 7), using the algorithms found in VGRID (ref. 6), FELISA (refs. 9–12), and AFLR3 (ref. 13) interchangeably in meshing applications.

The grid generation development relies heavily on the development of commercial CAD systems. Until a few years ago, most commercial grid generators had a very weak link to the commercial CAD systems. Today, most commercial grid generators have a direct interface to most CAD systems.

In the design environment of today, geometry models are quite complex: a CAD model often uses thousands of curves and surfaces to represent an aircraft. This level of complexity highlights the importance of automation. Today, the commercial grid generation companies have shifted their focus to the important issue of process automation. For example, CFD Research (`http://www.cfdrc.com`) has added a journaling capability, allowing parameter and grid/geometry changes to be saved and modified at a later date. Also, Gridgen (`http://www.pointwise.com`) has a new scripting feature that can automate the grid generation process. These are promising directions for the creation of tools that will enable an automated multidisciplinary shape optimization. Currently, the commercial unstructured grid generators lack the important capability of generating high Reynolds number viscous (highly stretched) grids.

The sensitivity analysis is an essential building block of the gradient-based aerodynamic shape optimization. The sensitivity is defined as the partial derivative of the grid-point coordinates with respect to a design variable. Despite recent advances in sensitivity analysis, very few grid generation tools currently provide analytical grid-point sensitivity. It is possible to use finite differences to calculate the sensitivity derivatives, but certain feasibility and accuracy issues must be considered. Using the finite-differences approximations for sensitivity calculation is feasible as long as the perturbed geometry grids have the same topology as the unperturbed geometry grids. Commercial CAD systems and unstructured grid generation techniques do not guarantee to maintain the same topology for the grids of the perturbed and unperturbed geometries.

The long-term goal is automation and self-adaptation, proceeding directly from the CAD model without human intervention. Some degree of human involvement at the outset is desirable in specifying (1) the intended outcomes of the simulation, and (2) a set of "best practice" measures (such as minimum wall spacing, curvature resolution, etc.) gained from experience with past analyses to construct a reasonable starting grid. The grid self-adapts from that point on, interacting with the primal and adjoint flow solvers to give the most accurate resolution for a given cost or the cheapest solution for a given accuracy. It is clear that development frameworks that encapsulate individual processes to a high degree are necessary for a general, extensible capability; this adaptation process is discussed further in the next section.

3 Error Assessment and Grid Adaptation

Adaptive-grid methods have long promised to provide more accurate solutions or reduced costs for a given application. The necessary software to refine and derefine the grid can be quite extensive. For hypersonic flow applications, these methods have proven useful in allowing the robust capture of strong shocks, although recent studies have indicated that showing a direct benefit in efficiency for these applications can be problematic (refs. 14, 15).

Historically, the criteria for adaptation in adaptive-grid methods has been feature based; a recent innovation is to adapt the grid based on the solution of the adjoint equation. The feature-based methods cluster the grid based on some heuristic, such as pressure gradient or curvature, that clusters the grid points intuitively. Controls must be exercised near discontinuities (shocks) because all the added grid points can go to this region to the exclusion of other regions; thus, seemingly grid-converged but inaccurate solutions can be found (ref. 16). Pirzadeh used entropy as a feature-based adaptation criteria in vortex-dominated flows over delta wings and showed an improvement in feature resolution (ref. 17). Vortical flows are known to diffuse rapidly because of the artificial viscosity of standard schemes; the adaptive-grid method allows interacting vortices to be computed accurately and enables the simulation of such flows.

The adjoint-based adaptation procedure stems from the recent progress in developing *a posteriori* error estimates for the fluid dynamic equations (refs. 18–20). The adjoint developments use the fact that the error in functionals, defined as integral quantities such as lift or drag, are the most important features in an engineering simulation. The adjoint solution reflects the linearized change in this functional with respect to the primal residual error measure, corresponding to the extent that the discrete solution differs from the differential solution. This is used as a rationale for grid adaptation by Müller and Giles (ref. 18), since wherever the adjoint variable is large, even small primal residual errors can have a significant effect on the functional. An alternative refinement strategy is used by Venditti and Darmofal (ref. 19, 20) and is designed to improve the accuracy of the computable error estimate. This method has a more conservative criterion for adaptation that is based on reducing both the primal and the adjoint residual errors. This methodology has been applied to multielement subsonic and supersonic inviscid flow test cases and compares favorably in terms of accuracy and robustness with gradient-based methods. In some cases, at equivalent accuracies in drag, a tenfold reduction in the number of grid points over a uniform refinement procedure was found. Also, unlike the feature-based adaption, the adjoint-based adaptation has to-date always converged the functional to the correct (i.e., uniform-refinement) result.

The error estimates obtained through the use of adjoints provide a natural stopping criterion; the adaptation simply stops when the computable estimate of the error reaches the desired goal. This aspect is especially important in 3-D analysis and design methods where the solution times are directly related to the number of grid points. Both the error estimate and grid adaptation procedures have been extended to three dimensions by Park (ref. 21), including linkage in adaptation to the underlying CAD surface definition. The initial results demonstrated significant efficiency gains with the adaptive mesh procedure compared with uniform refinement for the drag error.

To show the payoff of adjoint based adaptation procedures in practical applications, we cite an example. In the summer of 2001, the AIAA Applied Aerodynamics Technical Committee conducted the Drag Prediction Workshop to determine the state of the art of prediction in a focused area, namely the transonic cruise drag prediction for subsonic transports. There was a large spread among the 35 contributed solutions in drag. Hemsch statistically analyzed the contributions (ref. 22) and, even after discarding many of the outlying solutions, found a standard deviation of more than 20 drag counts, significantly more than the desired error levels of 1 count in absolute drag and 1/2 count in incremental drag. The turbulence models varied, but the most significant contribution to uncertainty was the variability of the unknown discretization error associated with a given grid. As is typical of 3-D complex-geometry applications, obtaining an error estimate by refining the grid is prohibitively expensive; the grid can be coarsened but usually it becomes too coarse to give a meaningful result (ref. 23). This variability could be substantially reduced by incorporating "best practice" process improvements. However, viable 3-D error estimates would enable a much more precise assessment of the numerical versus the physical modeling errors. An adaptive-grid method in concert with error prediction promises to efficiently compute the simulation to a prescribed error tolerance much closer to the desired error levels.

4 Convergence Acceleration

Over the past thirty years, considerable progress has been made in developing large-scale CFD solvers for the Euler and Navier-Stokes equations. Existing solvers can accommodate a broad range of Mach numbers and are quite robust. Much of the robustness stems from the realization of nonlinear schemes that have roots in the numerical solution of the time-dependent Riemann problem, extended to multiple dimensions on a dimension-by-dimension basis. The flux-difference-splitting scheme is an example of this approach; central differencing schemes with scalar or matrix artificial viscosity are closely related to this approach. At low Mach numbers, preconditioning is typically required for accuracy and efficiency. The equations are solved using a time-dependent (multistage time-stepping methods) or a quasi-Newton framework (approximate implicit or residual minimization schemes), often embedded in a Full Approximation Scheme (FAS) multigrid framework.

Existing solvers are typically able to converge lift and drag values for cruise configurations within approximately 1000 residual evaluations, where a residual is defined as the numerical evaluation of the discretized system of equations. More complex geometry or physics generally requires many more residual evaluations to converge. This efficiency can be compared with an optimally convergent method, defined as solving the governing system of equations in a computational work which is a small (less than 10) multiple of the operation counts in a single residual evaluation. Such a method is said to possess textbook multigrid efficiency (TME) (refs. 24–27); the descriptor TME derives from the efficiencies that have been demonstrated for elliptic equations with multigrid methods, already available in many textbooks. The two essential ingredients of this efficiency are that the discrete equations be solved only to the level of discretization error (not to machine zero) and that the convergence rate of the algebraic errors be independent of the grid. Extending this efficiency to a nonlinear system of conservation equations with discontinuities (shocks, slip lines, etc.) and singularities (flow- or grid-induced) is a formidable task. Each of the difficulties needs to be isolated, analyzed, and solved systematically using a carefully constructed series of model problems. Nonetheless, a potential gain of more than two orders of magnitude in operation count reduction is theoretically possible.

This potential gain in efficiency is being pursued by many groups through the construction of general-purpose solvers for linearized systems of equations. Newton's method is the prototype solver in this regard, although a direct solver is not currently used in practice beyond one dimension. An alternate approach for the fluid dynamics equations promises to reach greater optimal efficiency. The approach exploits a special property of the governing equations that in smooth regions (i.e., neglecting shocks), the determinant of the matrix of operators consists of separable factors. This special property of the differential equations is termed factorizability. Exploiting this property in discrete computations reduces the problem of relaxing a complicated system of discretized coupled differential equations to relaxing scalar factors constituting the system determinant.

Constructing a discretely factorizable scheme (i.e., one that mimics discretely the differential property of factorizability) is difficult, since the usual discretizations do not satisfy this property. There are some notable exceptions, such as staggered-grid discretizations or incompressible pressure-based discretizations (ref. 28), but the widely used flux-difference-splitting scheme is not factorizable. Several approaches are being developed at NASA Langley to exploit this property through the development of discretely factorizable schemes, and some encouraging results have been obtained. This progress was reported in a special session on TME methods for the fluid dynamics equations held at the 2001 AIAA CFD conference (refs. 29–33) .

The idea to exploit factorizability dates back nearly two decades (refs. 24,25). It has not been used much in mainstream CFD development for many reasons. Among them is that one of the important problems to overcome was accuracy and robustness for strong shocks, and this has been approached mainly through time-dependent methods. Another reason is that it is difficult to attain TME for non-elliptic factors and mixed elliptic-hyperbolic systems. Recent interest in factorizability was renewed partly due to the demonstration by Ta'asan (ref. 34) of TME for the Euler equations. Brandt summarized the barriers, along with possible solutions, to attaining TME for fluid dynamics in a recent ICASE paper (ref. 26) much of which appeared in an addendum to a recent textbook on multigrid (ref. 27).

In order to translate the idea of factorizability into practice, the solvers should be restructured to handle a progressively more difficult hierarchy of equations: scalar (e.g., hyperbolic, elliptic, convection-diffusion, or full potential), constant coefficient systems, variable coefficient systems, nonlinear nonconservative systems, and nonlinear conservative systems. The scalar equations represent each of the factors of the system determinant, for which TME has to be demonstrated before expecting TME for the system. Most of the quantitative theoretical analysis is limited to the constant coefficient systems of equations. Thus, this stage is especially important in demonstrating that the implementation correctly recovers the theoretically expected behavior. The methodology is also applicable to the adjoint (dual) equations that represent a set of nonconservative variable-coefficient system of equations.

For conservative systems of fluid dynamics, the conservative variables are corrected according to each of the variables associated with the determinant factors in a process known as distributed relaxation. The variables associated with the determinant factors are quite often familiar to aerodynamicists, for instance as the potential function or the entropy function. Thus, the original coupled (conservative) equations are always used to compute residuals.

Four attributes differentiate special-purpose methods from the general-purpose large-scale simulation methods of today, namely treatment of: (1) complex geometry, (2) viscous flow, including turbulence models, (3) compressible flow, and (4) discontinuities, including shocks and contacts. Considerable investment exists in these general-purpose solvers, and new algorithms have to show a quantum increase in capability before they become part of the mainstream usage. Much, but not all, of the groundwork, including demonstrations (refs. 35–38), is already in place for the development of general-purpose solvers that can attain TME for the nonlinear nonconservative equations of fluid dynamics, i.e., factors 1–3. The treatment of discontinuities remains a hurdle. A combination of two approaches is needed: (1) globally distributed relaxation for the smoothly varying part of the solution, and (2) local quasi-Newton relaxation for the regions near discontinuities (shocks, contacts) and, possibly, near boundaries. Some initial results using this approach have been demonstrated in one spatial dimension (ref. 39). Also, a factorized scheme has been demonstrated for the subsonic and transonic flow over airfoils, including shocks (ref. 31).

To attain a possible hundredfold increase in efficiency, the underlying physical characteristics of the simulation, namely the determinant factors (elliptic or hyperbolic) of smooth regions and the discontinuities (shocks or contacts), must be approached separately with specialized techniques. This is distinctly different from the global pseudo-time-marching approaches taken with current methods. The new approach is also applicable to time-dependent flows for which solvers are conceptually simpler to develop than those for steady-state flows. It is also well suited for efficient solution of the adjoint equation used in design and error estimation procedures.

5 Physical Models and Module Synthesis

The wide breadth of applications for analysis and design studies supporting NASA missions requires a wide variety of physical models, including transition and turbulence models and high-energy gas models. The fidelity of the physical model is likely to be the single most important discriminating factor between alternative software systems of the future. Thus, it is appropriate to develop standards that enable software systems to access a wide variety of physical models.

At NASA Langley, the range of physical models supported by different codes has largely evolved in response to programmatic needs. In late 1999, physical models appropriate for hypersonic flow, i.e., thermochemical nonequilibrium high-energy flow, were available in two structured-grid solvers, LAURA (ref. 40) and VULCAN (ref. 41). The unstructured-grid solver FELISA was available for inviscid flows using perfect gas and equilibrium chemistry models. All of the viscous unstructured-grid solvers were restricted to perfect-gas simulations. Since there was a high demand for complex geometry simulations, a project was proposed to synthesize these modules into an existing unstructured-grid code, FUN3D (refs. 42, 43). The project, entitled High-Energy Flow Solver Synthesis (HEFSS), was intended to include modern practices, such as modular, object-like code structure, and run on distributed-memory commodity clusters and shared-memory supercomputers.

The first argument to be settled by the HEFSS team was to choose the software language of modern scientific computing. Although there are many scientific software languages, the most evolved object-oriented is C++, especially as regards the use of templates. There is, however, a large base of legacy scientific code written in Fortran 77; Fortran 90 (supporting dynamic memory allocation, subroutine/function recursion, and pointers) is now available, and Fortran 2000, not yet available, should contain more object-oriented syntax. There is also an argument that Mathematica is the most appropriate choice because its symbolic manipulation, graphics, text processing, and computing capabilities are bundled interchangeably in one platform (private communication from R. Walters of Virginia Polytechnic Institute and State University, 2001). The team could reach no clear consensus, and Fortran 90 was adopted as a reasonable choice for the project. Yet the debate over the most appropriate scientific computing language continues today in the software industry.

One interesting finding of the HEFSS team was that Fortran 90 implementations varied widely across machines. Some coding styles that were very useful for modularity had to be discarded because either one or more of the intended compilers did not fully implement the Fortran 90 standard or the execution of the syntax was slower by a factor as high as ten. An example in this regard was the implementation of `use modules` rather than argument lists to pass information between routines; the `use modules` were preferred on the basis of modularity but were two times slower in execution. Thus, a practical compromise was struck between the language and the implementation on current machines.

Through weekly team meetings and user/developer workshops, the HEFSS team developed a configuration management plan, a list of attributes to be embodied by the software, and coding standards similar to the standards developed by the European weather prediction community (`http://www.meto.uk.gov/sec5/NWP/TOVS_monitoring/NWP_F90Standards.html`). The documentation (`http://hefss.larc.nasa.gov`) is web-based. Software engineering support was established for maintenance and eventual dissemination of the system. A great deal of software changes made the existing unstructured-grid software more streamlined and modular, especially regarding grid processing, boundary conditions, and physical modeling. The physics modules for thermodynamics, transport properties, chemical kinetics, and thermal relaxation were first developed and tested within the LAURA code; the generic data structure was modeled after the VULCAN code and recovers all of the current hypersonic applications. Nightly builds and test case execution of the system across different computer platforms uncovered many of the implementation difficulties mentioned previously and many software inconsistencies. The initial version was completed in the fall of 2001 and is undergoing testing on a series of benchmark cases. The code retains the capabilities of the original perfect-gas code but also allows high-energy simulations, including scramjet and planetary entry flowfields.

6 Design Optimization

High-fidelity computational models have traditionally been used in the analysis of physical systems. They have recently begun to be used for design optimization as well. Given a set of design variables, the analysis problem computes the corresponding state variables. The attendant design optimization problem is then to optimize a set of objectives, subject to design constraints. The objectives and constraints are quantities of engineering interest, derived from the state variables that describe the physical behavior of the system. While design optimization problems with a few variables may be approached with heuristic and intuitive schemes, large-scale (i.e., computationally intensive), nonlinear problems generally require rigorous, derivative based optimization methods. Iterative use of simulations within design optimization algorithms presents a difficult challenge to developers, because simulation-based design places much more stringent requirements on simulations in terms of robustness and automation than does the use of simulations strictly for the purposes of analysis.

Simulation-based optimization methods are enabling tools for advanced concept design, promising to determine new shapes within properly formulated constrained design problems. The optimization environment on parallel computers requires the integration of many processes, including domain decomposition, flow and adjoint (if required) solvers, gradient evaluation, mesh movement and adaptation, and parameterization. FUN3D (refs. 42,43) is the unstructured-grid code that forms the basis of the HEFSS effort to extend high-energy physics models to applications with more complex geometries. FUN3D has a unique capability for use in viscous design optimization on unstructured grids (ref. 44) since it has been completely hand-differentiated to enable sensitivity analyses that are discretely consistent with the computed flow solution. This section describes the approach to design optimization.

The need for derivatives in large-scale optimization is well known. Very attractive derivative-free methods cannot currently be used for problems with many design variables or even for problems with relatively few design variables but expensive functions because of the large number of required function evaluations. A sensitivity derivative is the derivative of an output quantity with respect to an input quantity; a simple example for a CFD code is the derivative of the lift with respect to the thickness of a particular airfoil section. Traditionally, sensitivities are computed with the finite-difference approach. Since this approach scales with the number of design variables, it can be expensive and, moreover, it can be inaccurate.

Automatic code differentiation tools form a more advanced approach to computing sensitivities. Tools such as ADIFOR (Automatic Differentiation of Fortran 77, refs. 45–48) and ADIC (Automatic Differentiation of C, ref. 49) function as preprocessors. For instance, ADIFOR accepts as input a Fortran code along with specifications of the input and output variables, and it produces as output an augmented Fortran code that contains the original analysis capability plus the capability for computing the derivatives of all the specified output quantities with respect to all the specified input quantities. Alternatively, one may use the complex variable technique (refs. 50–52) to produce sensitivities. These methods are attractive when the number of outputs is large compared to the number of design variables. Derivatives produced via the complex variable technique are available with the FUN3D solver (refs. 53–55).

For design problems where the number of outputs is significantly lower than the number of design variables, computing sensitivities via the adjoint technique is an efficient and versatile alternative. The approach not only provides analytic derivatives, but the error estimation and grid adaptation criteria described previously are an inherent by-product of adjoint-based design. Hand-differentiated code is used to construct an adjoint system within FUN3D. The system's solution yields sensitivities discretely consistent with the analysis problem. For a single output, sensitivities computed in this fashion are produced at a cost of a single additional linear solve and matrix-vector product.

The three approaches—adjoint-based sensitivities, automatic differentiation, and the complex variable technique—place different implementation and memory requirements on the system and exhibit different, problem dependent performances and each method has its range of applicability. The present effort focuses on adjoint-based sensitivities because aerodynamic optimization problems typically contain few outputs and many design variables.

Obtaining sensitivities by any method constitutes the most expensive computational component in design optimization. In particular, flow equations for outputs and the adjoint equations for sensitivities in the adjoint-based approach are usually solved separately, as in the FUN3D code, where equations are solved with separate residual minimization methods. Extending the ideas of Giles (refs. 56, 57), Darmofal (private communication, Massachusetts Institute of Technology, August 2001) has conducted promising research in combining the flow and adjoint equations in these residual minimization methods. Darmofal has shown that superconvergence of the solution occurs in the sense that the algebraic errors in the functional to which the adjoint pertains is reduced as the product of the primal and dual residuals is reduced. This has been confirmed through numerical computations for two-dimensional model equations and inviscid flow.

In a design environment, grid generation is a critical issue, both from the standpoint of robustness and the quality of the grid. Mesh deformation is an important element in the analysis of moving bodies and shape optimization. The lack of robust and efficient mesh deformation tools is still a major barrier to routine application of CFD tools in single-discipline and multidisciplinary design optimization. CFD application for shape optimization requires a robust, automatic, and efficient tool to propagate the boundary deformation into the field mesh. For gradient-based optimization, the efficiency is particularly crucial because, in addition to the boundary deformation, the sensitivity of the boundary coordinates must be propagated into the field mesh. The current procedure uses the solution to an elasticity equation for grid movement to overcome earlier problems encountered with a spring-analogy method. Further improvements in reliable mesh movement are needed; a potential approach uses free-form deformation techniques adapted from the movie industry to move the entire volume grid as the design changes. This technique has already proven useful in surface parameterizations of existing configurations (ref. 58).

In traditional optimization methods for high-fidelity problems, optimization software interacts directly with high-fidelity analysis and derivative codes to compute objectives, constraints, and their sensitivities. For example, in a drag minimization procedure, an optimizer would interact directly with a Navier-Stokes (high-fidelity) code to evaluate drag and the design constraints on, say, lift. A considerable gain in efficiency is attained by using variable-fidelity models in a rigorous optimization process termed Approximation Model Management Optimization (AMMO). AMMO interacts with lower-fidelity models most of the time and infrequently with the higher-fidelity code (refs. 59–61) to provide systematic corrections to the lower-fidelity model. AMMO guarantees convergence to high-fidelity critical points; that is, convergence occurs to designs of the same accuracy that would be obtained if the high-fidelity code were used throughout. Design studies of two-dimensional multi-element airfoils have demonstrated a savings factor of five with this approach, using an Euler code as the lower fidelity code and the Navier-Stokes code as the higher fidelity code. A greater savings should be possible in three dimensions since the difference in the relative expenses of the two approximations is greater. AMMO allows for the broadest range of low-fidelity models, including data-fitting (response surfaces, kriging), reduced-order, variable-resolution, variable-accuracy, and variable-fidelity physics models.

The longer term vision for design has several major features.

First, AMMO must design with variable-fidelity models of optimal quality. "Optimal" in this context means that the low-fidelity model is significantly cheaper than the high-fidelity model but, when corrected, maintains the descent characteristics necessary to make rapid progress with the high-fidelity model. Such optimal models have been considered in a more narrow sense in general unconstrained optimization (ref. 62). Constrained optimization, and specifically aerodynamic optimization, is much more difficult. A decision on optimality of a low-fidelity model can be obtained from the error estimates of the objective and constraint gradients. Such error estimates for variable-resolution models should come at a small overhead, given that the adjoint equation is already being used in the evaluation of derivatives.

Second, AMMO must yield robust designs. In this context robustness is twofold: resulting designs should be insensitive with respect to uncertainties in the design parameters (ref. 63), such as geometry variations; they should also be acceptable for a wide range of operating conditions. Methodologically, the former requirement implies the incorporation of stochastic optimization techniques (ref. 64). To attain the latter, multicriteria decision making (ref. 65) will be part of the design process.

Finally, given that the design of realistic vehicles is always multidisciplinary, rigorous techniques of multidisciplinary design optimization (ref. 66) will be brought into the design problem formulation and solution strategies.

7 Programming Environment

The large-scale simulation codes in use at NASA Langley are quite complex, typically averaging about 100,000 lines of code and supported by one or two researchers. At one point, the previously referenced unstructured-grid design code had nearly 500,000 lines of code and was supported by one person. The complexity of the software makes modifications and maintenance difficult, especially in the light of the evolving hardware computing environment of today and the need for maintaining discrete consistency between the analysis and design codes. Though mathematical libraries exist for solving linear algebra problems, there is no similar library for CFD. Although some code is exchanged between the various developers (e.g., turbulence models and block solvers), the software and its subsequent maintenance are duplicated. Thus, development of a mathematical library for CFD, relying on standard interfaces such as those described further, is one way to reduce the burden of this large software.

CGNS (refs. 2–4) originated in 1994 as a joint effort between Boeing and NASA, and has since evolved into an ISO standardization effort for fluid dynamics data (refs. 2, 4) with participation from many contributing organizations worldwide. It represents an effort to standardize CFD input and output, including grids, flow solution, connectivity, boundary conditions, and auxiliary information. It is designed to be extensible and allows for file-stamping and user-inserted commenting. The CGNS system creates binary files that are portable across computer platforms. The system is a software library that includes a second layer of software, known as a mid-level library or an API, which eases the implementation of the standard into existing CFD codes. Further information, documentation, and the open-source software itself is available on the World Wide Web (`http://www.cgns.org`).

The CGNS software is distributed openly; a widely known example of a very successful open source development is the Linux operating system. Although there are examples of other less successful open-source developments, several quite sophisticated software systems have evolved for source code version control, such as the Concurrent Versions Systems (CVS) (`http://www.cvshome.org`) and its Apache Web interface (`http://stud.fh-heilbronn.de/zeller/cgi/cvsweb.cgi`). The CVS software, in particular, is an integral tool in the collaborative software development discussed in this paper.

Other organizations have been struggling with the complexity of software systems for simulations. B. Kleb and W. Wood, under a Langley Creativity and Innovation project, experimented with what they term "Object-Oriented Designer CFD" and sponsored an ICASE lecture series entitled "Modern Programming Practices." They propose to contend with software complexity by introducing at the very outset qualities that ensure modular and extendible agile software that can be supported in an open team environment. The radically different ideas springing from practices developed by successful software teams has come to be termed "eXtreme Programming." The ideas include test-first and acceptance coding (including frameworks for coding), refactoring, agile methods, and pair programming. Many of the ideas are a cultural change to established programming practice. As an example, properly functioning software puts a high mental strain on the developers and encourages slow, methodical, and incremental changes to existing software. Much time can be spent finding seemingly small errors in code (e.g., a "1" instead of a "l" in column 37 of line 258,711). Successful programming teams using the newer methodology feel comfortable about maintaining and radically changing a million lines of code in the knowledge that the code successfully passes around ten thousand unit tests of the software at every build. These nightly (or more often if possible) builds and test case executions are proving their worth in the entire software development process discussed in this paper. They are ideal for checking that discrete consistency is maintained with working versions as the software evolves.

8 A Future Scenario

The following scenario illustrates the potential benefits of applying multidisciplinary design and analysis methods to computational simulation.

Sometime in the not too distant future, NASA has designed a new planet exploratory vehicle with preliminary design methods, composed principally of linear methods with nonlinear corrections; the mission includes aerobraking and atmospheric flyover segments. Twenty points over the mission profile, including reentry, deployment, and flyover, are selected for detailed CFD analysis in the morning. Starting from the CAD description and "best practice" guidelines, an initial grid is constructed in one hour. The twenty viscous CFD simulations are started independently with tolerances specified for a weighted combination of the lift, drag, and pitching moment errors at deployment and flyover conditions; at reentry conditions, the tolerances include a term for the heating error. Different thermal and chemical models are automatically selected as the solution evolves; the aeroelastic structural deflections arising from the load are computed as the solution evolves, using structural codes linked to the CFD codes through extensions to the CGNS system. The solutions are rapidly converged to within acceptable discretization errors with fast multigrid solvers. The grids are adapted to minimize the predicted errors until the prescribed error tolerances are met. In several areas judged most critical, a single uniform refinement is used and verifies that the solution is within the required accuracy. The twenty solutions are used to refine mission performance obtained from previous estimates. By late afternoon, results indicate that the flyover drag is too high to allow the intended flyover duration; visualization of the solution indicates a separation on the inboard wing section. The next morning, the CAD model in that region is quickly parameterized. The parameters of a multipoint design optimization that includes the drag at flyover and the heat load at reentry are formulated. A design optimization is initiated and a solution is returned in the afternoon. The wing leading-edge radius is relofted to eliminate the separation. The twenty solutions are restarted with the new geometry and the flyover duration is now predicted to meet the mission requirements.

9 Concluding Remarks

Opportunities for breakthroughs in large-scale computational simulation and design have been addressed. The breakthroughs offer significant improvements in cost, speed, and accuracy of the current bottlenecks in the simulation and design process. In order to obtain the full impact of these improvements to the technology-driven applications needed by NASA, the software programming environment must be improved as well. An attainable goal is a hundredfold improvement in flow time from a computer-aided design (CAD) definition to a performance prediction with known levels of discretization error. This improvement is in addition to improvement from computing hardware. The payoffs will enable many simulations not possible today. Among them are the use of higher fidelity tools earlier in advanced concept development: for example, exploring overland supersonic flight or distributed flow control for separated flow, or using a larger envelope of configuration parameters in design studies. These improvement enable the increased use of simulation as a cost-effective alternative to other methods for evaluation of the entire flight envelope for aerospace vehicles.

This payoff presumes the improvement and validation of physical models, although the speedups considered here to the underlying simulation methods are much-needed improvements, especially for time-dependent high-Reynolds applications. This paper emphasizes computational fluid dynamics (CFD), but many benefits accrue in the multidisciplinary environment. For example, the development process itself requires the encapsulation and standardization of the individual processes, a key element of communication across disciplines. Also, many of the needs in CFD (e.g., geometry modeling, grid generation, and solution of large systems of equations) are shared by other field simulation disciplines, such as acoustics or electromagnetics; these disciplines potentially benefit from the breakthroughs described herein.

10 Acknowledgments

The ideas presented here first came together in the spring of 2000 at a request from the Advanced Vehicles System program to the research competencies for new initiatives in basic research. Many people contributed to the inception and coordination of the response, including all of the eventual FAAST team. The new initiative was incorporated into the FAAST element of ASCOT and 3rd Generation RLV, led by L. Yip and R. Merski, respectively. The project planning and integration was led by J. Thomas and the FAAST element leads, including P. Gnoffo (HEFSS), J. Weilmuenster and W. Jones (CAD-Grid Methods), T. Roberts (Optimally Convergent Algorithms), and E. Nielsen and N. Alexandrov (Efficient Adjoint Design Methods). The FAAST development team members included S. Alter, H. Atkins, K. Bey, K. Bibb, R. Biedron, M. Carpenter, F. Cheatwood, P. Drummond, B. Kleb, R. Mineck, E. Lee-Rausch, M. Park, S. Pirzadeh, J. Samareh, C. Swanson, V. Vatsa, J. White, and W. Wood.

References

1. Rubinstein, R.; Rumsey, C. L.; Salas, M. D.; and Thomas, J. L.: *Turbulence Modeling Workshop*. ICASE Rep. No. 2001-37, NASA/CR-2001-210841, 2001.

2. Poirier, D. M.; Bush, R. H.; Cosner, R. R.; Rumsey, C. L.; and McCarthy, D. R.: Advances in the CGNS Database Standard for Aerodynamics and CFD. AIAA-2000-0681, Jan. 2000.

3. Rumsey, C. L.; Poirier, D. M.; Bush, R. H.; and Towne, C. E.: *User's Guide to CGNS*. NASA/TM-2001-211236, 2001.

4. Legensky, S. M.; Edwards, D. E.; Bush, R. H.; Poirier, D. M. A.; Rumsey, C. L.; Cosner, R. R.; and Towne, C. E.: CFD General Notation System (CGNS): Status and Future Directions. AIAA-2002-0752, Jan. 2002.

5. Rogers, S. E.; Roth, K.; Nash, S. M.; Baker, D. M.; Slotnick, J. P.; Whitlock, M.; and Cao, H. V.: Advances in Overset CFD Processes Applied to Subsonic High-Lift Aircraft. AIAA-2000-4216, Aug. 2000.

6. Pirzadeh, S.: Progress Toward a User-Oriented Unstructured Viscous Grid Generator. AIAA-96-0031, Jan. 1996.

7. Jones, W. T.: An Open Framework for Unstructured Grid Generation. AIAA-2002-3192, June 2002.

8. Haimes, R.: CAPRI: Computational Analysis PRogramming Interface. Massachusetts Institute of Technology, 2002.

9. Peraire, J.; Peiro, J.; and Morgan, K.: Multigrid Solution of the 3D Compressible Euler Equations on Unstructured Tetrahedral Grids. *International J. Numer. Methods Eng.*, vol. 36, no. 6, 1993, pp. 1029–1044.

10. Morgan, K.; and Peraire, J.: Unstructured Grid Finite Element Methods for Fluid Mechanics. *Inst. Phys. Rev.*, vol. 61, no. 6, 1998, pp. 569–638.

11. Bibb, K. L.; Peraire, J.; and Riley, C. J.: Hypersonic Flow Computations on Unstructured Meshes. AIAA-97-0625, Jan. 1997.

12. Prabhu, R. K.: *An Inviscid Computational Study of an X-33 Configuration at Hypersonic Speeds*. NASA/CR-1999-209366, 1999.

13. Marcum, D. L.: Efficient Generation of High-Quality Unstructured Surface and Volume Grids. *Eng. Comput.*, vol. 17, no. 3, 2001, pp. 211–233.

14. Yamaleev, N. K.; and Carpenter, M. H.: On Accuracy of Adaptive Grid Methods for Captured Shocks. Submitted to *Journal of Computational Physics*.

15. Wood, W. A.: *Multi-Dimensional Upwind Fluctuation Splitting Scheme With Mesh Adaption for Hypersonic Viscous Flow*. NASA/TP-2002-211640, 2002.

16. Warren, G. P.; Anderson, W.; Thomas, J. L.; and Krist, S. L.: Grid Convergence for Adaptive Methods. AIAA-91-1592, June 1991.

17. Pirzadeh, S.: Vortical Flow Prediction Using an Adaptive Unstructured Grid Method. RTO Applied Vehicle Technology (AVT) Panel, Loen, Norway, May 2001.

18. Müller, J.-D.; and Giles, M. B.: Solution Adaptive Mesh Refinement Using Adjoint Error Analysis. AIAA-2001-2550, June 2001.

19. Venditti, D. A.; and Darmofal, D. L.: Adjoint Error Estimation and Grid Adaptation for Functional Outputs: Application to Quasi-One-Dimensional Flow. *J. Comput. Phys.*, vol. 164, 2000, pp. 204–227.

20. Venditti, D. A.; and Darmofal, D. L.: Grid Adaptation for Functional Outputs: Application to Two-Dimensional Inviscid Flows. *J. Comput. Phys.*, vol. 176, 2002, pp. 40–69.

21. Park, M. A.: Adjoint-Based, Three-Dimensional Error Prediction and Grid Adaptation. AIAA-2002-3286, June 2002.

22. Hemsch, M.: Statistical Analysis of CFD Solutions From the Drag Prediction Workshop. AIAA-2002-0842, Jan. 2002.

23. Rumsey, C. L.; and Biedron, R. T.: *Computation of Flow Over a Drag Prediction Workshop Wing/Body Transport Configuration Using CFL3D*. NASA/TM-2001-211262, 2001.

24. Brandt, A.: Guide to Multigrid Development. *Lecture Notes in Math 960: Multigrid Methods*, W. Hackbusch and U. Trottenberg, eds., Springer-Verlag, 1982, pp. 220–312.

25. Brandt, A.: Multigrid Techniques: 1984 Guide With Applications to Fluid Dynamics. *Lecture Series at the Von-Karman Institute for Fluid Dynamics: Lecture Notes for Computational Fluid Dynamics*, St. Augustin, 1984.

26. Brandt, A.: *Barriers to Achieving Textbook Multigrid Efficiency in CFD*. ICASE Rep. No. 98-32, NASA/CR-1998-207647, 1998.

27. Brandt, A.: Recent Developments in Multigrid Efficiency in Computational Fluid Dynamics. *Multigrid*, U. Trottenberg, C. W. Oosterlee, and A. Schüller, eds., Academic Press, 2001, pp. 573–589.

28. Sidilkover, D.; and Ascher, U.: A Multigrid Solver for the Steady-State Navier-Stokes Equations using the Pressure-Poisson Formulation. *Matematica Aplicada e Computational*, vol. 14, 1995, pp. 21–35.

29. Brandt, A.; Diskin, B.; and Thomas, J. L.: Textbook Multigrid Efficiency for Computational Fluid Dynamics Simulations. AIAA-2001-2570, June 2001.

30. Diskin, B.; and Thomas, J. L.: Distributed Relaxation for Conservative Discretizations. AIAA-2001-2571, June 2001.

31. Roberts, T. W.: The Development of a Factorizable Multigrid Algorithm for Subsonic and Transonic Flow. AIAA-2001-2572, June 2001.

32. Swanson, R. C.: Towards Optimal Multigrid Efficiency for the Navier-Stokes Equations. AIAA-2001-2574, June 2001.

33. Sidilkover, D.; and Nielsen, E. J.: Factorizable Upwind Schemes—The Triangular Unstructured Grid Formulation. AIAA-2001-2575, June 2001.

34. Ta'asan, S.: *Canonical-Variables Multigrid Method for Steady-state Euler Equations*. ICASE Rep. No. 94-14, NASA CR-194888, 1994.

35. Brandt, A.; and Yavneh, I.: On Multigrid Solution of High-Reynolds Incompressible Entering Flow. *J. Comput. Phys.*, vol. 101, no. 1, 1992, pp. 151–164.

36. Diskin, B.: Efficient Multigrid Solvers for the Linearized Transonic Full Potential Equation. Ph.D. Thesis, Weizmann Institute of Science, June 1998.

37. Thomas, J. L.; Diskin, B.; and Brandt, A.: Distributed Relaxation Multigrid and Defect Correction Applied to the Compressible Navier-Stokes Equations. AIAA-99-3334, July 1999.

38. Thomas, J. L.; Diskin, B.; and Brandt, A.: Textbook Multigrid Efficiency for the Incompressible Navier-Stokes Equations: High Reynolds Number Wakes and Boundary Layers. *Comp. & Fluids*, vol. 30, no. 7, 2001, pp. 853–874.

39. Thomas, J. L.; Diskin, B.; Brandt, A.; and South, J. C.: General Framework for Achieving Textbook Multigrid Efficiency: Quasi-One-Dimensional Euler Example. *Frontiers of Computational Fluid Dynamics—2000*, D. A. Caughey and M. M. Hafez, eds., World Scientific Publ., 2000. (Also available as ICASE Rep. No. 2000-30, NASA/CR-2000-210320, 2000).

40. Cheatwood, F. M.; and Gnoffo, P.: *User's Manual for the Langley Aerothermodynamic Upwind Relaxation Algorithm (LAURA)*. NASA TM-4674, 1996.

41. White, J. A.; and Morrison, J. H.: A Pseudo-Temporal Multi-Grid Relaxation Scheme for Solving the Parabolized Navier-Stokes Equations. AIAA-99-3360, June 1999.

42. Anderson, W. K.; and Bonhaus, D. L.: An Implicit Upwind Algorithm for Computing Turbulent Flow on Unstructured Grids. *Comp. & Fluids*, vol. 23, no. 1, 1994, pp. 1–21.

43. Anderson, W. K.; Rausch, R. D.; and Bonhaus, D. L.: Implicit/Multigrid Algorithms for Incompressible Turbulent Flows on Unstructured Grids. *J. Comput. Phys.*, vol. 128, no. 2, 1996, pp. 391–408.

44. Nielsen, E. J.; and Anderson, W. K.: Recent Improvements in Aerodynamic Design Optimization on Unstructured Meshes. *AIAA J.*, vol. 40, no. 6, 2002, pp. 1–9.

45. Bischof, C.; Carle, A.; Khademi, P.; and Mauer, A.: ADIFOR 2.0: Automatic Differentiation of Fortran 77 Programs. *IEEE Comput. Sci. & Eng.*, vol. 3, no. 3, 1996, pp. 18–32.

46. Carle, A.; Fagan, M.; and Green, L. L.: Preliminary Results From the Application of Automated Adjoint Code Generation to CFL3D. AIAA-98-4807, Sept. 1998.

47. Park, M. A.; Green, L. L.; Montgomery, R. C.; and Raney, D. L.: Determination of Stability and Control Derivatives Using Computational Fluid Dynamics and Automatic Differentiation. AIAA-99-3136, July 1999.

48. Bischof, C.; Corliss, G.; Green, L. L.; Griewank, A.; Haigler, K.; and Newman, P. A.: Automatic Differentiation of Advanced CFD Codes for Multidisciplinary Design. *J. Comput. Sys. Eng.*, vol. 3, no. 6, 1992, pp. 625–637.

49. Bischof, C.; Roh, L.; and Mauer, A.: ADIC: An Extensible Automatic Differentiation Tool for ANSIC. *Software—Practice & Experience*, vol. 27, no. 12, 1997, pp. 1427–1456.

50. Lyness, J. N.: Numerical Algorithms Based on the Theory of Complex Variables. *Proceedings of the ACM 22nd National Conference*, Thomas Book Co., 1967, pp. 124–134.

51. Lyness, J. N.; and Moler, C. B.: Numerical Differentiation of Analytic Functions. *J. Numer. Anal.*, vol. 4, 1967, pp. 202–210.

52. Squire, W.; and Trapp, G.: Using Complex Variables to Estimate Derivatives of Real Functions. *SIAM Rev.*, vol. 10, no. 1, 1998, pp. 110–112.

53. Anderson, W. K.; Newman, J. C.; Whitfield, D. L.; and Nielsen, E. J.: Sensitivity Analysis for Navier-Stokes Equations on Unstructured Meshes Using Complex Variables. AIAA-99-3294, Nov. 1999.

54. Newman, J. C.; Anderson, W. K.; and Whitfield, D. L.: Multidisciplinary Sensitivity Derivatives Using Complex Variables. Engineering Research Center Rep. MSSU-COE-ERC-98-09, Mississippi State University, July 1998.

55. Newman, J. C.; Whitfield, D. L.; and Anderson, W. K.: A Step-Size Independent Approach for Multidisciplinary Sensitivity Analysis and Design Optimization. AIAA-99-3101, June 1999.

56. Giles, M. B.: Defect and Adjoint Error Correction. *Proceedings of the International Conference on Computational Fluid Dynamics*, Kyoto Institute of Technology, July 2000.

57. Giles, M. B.; Duta, M. C.; and Müller, J.: Adjoint Code Developments Using the Exact Discrete Approach. AIAA-2001-2596, June 2001.

58. Samareh, J.: Novel Multidisciplinary Shape Parameterization Approach. *J. Aircr.*, vol. 38, no. 6, 2001, pp. 1015–1024.

59. Alexandrov, N. M.: Robustness Properties of a Trust Region Framework for Managing Approximations in Engineering Optimization. AIAA-96-4102, Sept. 1996.

60. Alexandrov, N. M.; Nielsen, E. J.; Lewis, R. M.; and Anderson, W. K.: First-Order Model Management With Variable-Fidelity Physics Applied to Multi-Element Airfoil Optimization. AIAA-2000-4886, Sept. 2000.

61. Alexandrov, N. M.; Lewis, R. M.; Gumbert, C. R.; Green, L. L.; and Newman, P. A.: Approximation and Model Management in Aerodynamic Optimization With Variable-Fidelity Models. *J. Aircr.*, vol. 38, no. 6, 2001, pp. 1093–1101.

62. Carter, R. G.: Numerical Experience With a Class of Algorithms for Nonlinear Optimization Using Inexact Function and Gradient Information. *SIAM J. Sci. Comp.*, vol. 14, no. 2, 1993, pp. 368–388.

63. Huyse, L.: *Free-Form Airfoil Shape Optimization Under Uncertainty Using Maximum Expected Value and Second-Order Second-Moment Strategies*. ICASE Rep. No. 2001-18, NASA/CR-2001-211020, 2001.

64. Ermoliev, Y.; and Wets, R. J.-B., eds.: *Numerical Techniques for Stochastic Optimization*. Springer-Verlag, 1988.

65. Haimes, Y. Y.; Tarvainen, K.; Shima, T.; and Thadathil, J.: *Hierarchical Multiobjective Analysis of Large-Scale Systems*. Hemisphere Publ. Corp., 1990.

66. Alexandrov, N. M.; and Hussaini, M. Y., eds.: *Multidisciplinary Design Optimization: State of the Art*. SIAM, 1997.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704–0188

1. AGENCY USE ONLY ***(Leave blank)***

2. REPORT DATE
June 2002

3. REPORT TYPE AND DATES COVERED
Technical Memorandum

4. TITLE AND SUBTITLE
Opportunities for Breakthroughs in Large-Scale Computational Simulation and Design

5. FUNDING NUMBERS
706-31-21-80

6. AUTHOR(S)
Natalia Alexandrov, Stephen J. Alter, Harold L. Atkins, Kim S. Bey, Karen L. Bibb, Robert T. Biedron, Mark H. Carpenter, F. McNeil Cheatwood, Philip J. Drummond, Peter A. Gnoffo, William T. Jones, William L. Kleb, Elizabeth M. Lee-Rausch, N. Ronald Merski, Raymond E. Mineck, Eric J. Nielsen, Michael A. Park, Shahyar Z. Pirzadeh, Thomas W. Roberts, Jamshid A. Samareh, Roy C. Swanson, James L. Thomas, Veer N. Vatsa, K. James Weilmuenster, Jeffery A. White, William A. Wood, and Long P. Yip

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
NASA Langley Research Center
Hampton, VA 23681–2199

8. PERFORMING ORGANIZATION REPORT NUMBER
L–18196

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
National Aeronautics and Space Administration
Washington, DC 20546–0001

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
NASA/TM–2002–211747

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Unclassified-Unlimited
Subject Category 02 Distribution: Standard
Availability: NASA CASI (301) 621–0390

12b. DISTRIBUTION CODE

13. ABSTRACT ***(Maximum 200 words)***
Opportunities for breakthroughs in the large-scale computational simulation and design of aerospace vehicles are presented. Computational fluid dynamics tools to be used within multidisciplinary analysis and design methods are emphasized. The opportunities stem from speedups and robustness improvements in the underlying unit operations associated with simulation (geometry modeling, grid generation, physical modeling, analysis, etc.). Further, an improved programming environment can synergistically integrate these unit operations to leverage the gains. The speedups result from reducing the problem setup time through geometry modeling and grid generation operations, and reducing the solution time through the operation counts associated with solving the discretized equations to a sufficient accuracy. The opportunities are addressed only at a general level here, but an extensive list of references containing further details is included. The opportunities discussed are being addressed through the Fast Adaptive Aerospace Tools (FAAST) element of the Advanced Systems Concept to Test (ASCoT) and the 3rd Generation Reusable Launch Vehicles (RLV) projects at NASA Langley Research Center. The overall goal is to enable greater inroads into the design process with large-scale simulations.

14. SUBJECT TERMS
CFD, numerical methods, multigrid

15. NUMBER OF PAGES
27

16. PRICE CODE
A03

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	