18th AIAA Computational Fluid Dynamics Conference
25 - 28 June 2007, Miami, FL

AIAA 2007-4084

Petaflops Opportunities for the NASA Fundamental Aeronautics Program

Dimitri J. Mavriplis [*]
David Darmofal [†]
David Keyes [‡]
Mark Turner [§]

The premise of this paper is the observation that the engineering community in general, and the NASA aeronautics program in particular, have not been active participants in the renewed interest in high performance computing at the national level. Advocacy for high performance computing has increasingly been taken up by the science community with the argument that computational methods are becoming a third pillar of scientific discovery alongside theory and experiment. Computational engineering, on the other hand, has continually been relegated to a set of mature software tools which run on commodity hardware, with the notion that engineering problems are not complex enough to warrant the deployment of state-of-the-art hardware on such a vast scale. We argue that engineering practices can benefit equally from an aggressive program in high performance computational methods, and that these problems are at least as important as science problems, particularly with regards to any national competitiveness agenda. Because NASA aeronautics has historically been a principal driver of computational engineering research and development, the current situation represents an opportunity for the NASA aeronautics program to resume its role as a leading advocate for high performance computational engineering at the national level. We outline a sample set of Grand Challenge problems which are used to illustrate the potential benefits a reinvigorated program could produce, and use these examples to identify critical barriers to progress and required areas of investment. We conclude by noting that other communities have spent significant efforts in formulating the case for increased investment in high performance computing activities, and that a similar roadmap will be required for the engineering community.

I. Introduction

In 1976, the ILLIAC IV supercomputer went into production use at the NASA Ames Research Center. Although the performance of this machine was below original design expectations, the ILLIAC IV nevertheless constituted the most powerful supercomputer in the world at the time,[1] and gave NASA researchers an order of magnitude more computational power than had previously been available. The driving applications in the agency at that time were none other than computational fluid dynamics (CFD), and NASA quickly became the high-performance computing (HPC) leader in this field, thanks in part to visionary leadership, state-of-the-art facilities, and forward thinking education and hiring practices.[2] The rapid pace of development and early success of CFD within the NASA aeronautics program led to the creation of the Numerical Aerodynamic Simulator (NAS), which hosted a variety of leading edge supercomputers over the 80's and 90's. When the US Government developed a comprehensive multi-agency program for high-performance computing under the High-Performance Computing and Communication Program in the 1990's (HPCCP), NASA was naturally a strong participant, with a FY 1992 budget of \$71.2M (relative to a \$92.3M FY 1992

---

[*] Professor, Department of Mechanical Engineering, University of Wyoming, AIAA Associate Fellow.
[†] Associate Professor, Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, Member AIAA.
[‡] Professor, Department of Applied Physics and Applied Mathematics, Columbia University.
[§] Research Professor, Department of Aerospace Engineering & Engineering Mechanics, University of Cincinnati, AIAA Associate Fellow.

budget for the Department of Energy (DOE)), within an overall HPCCP budget of $596.6M for FY 1992.[3] NASA participation consisted mainly of two programs, Earth and Space Sciences (ESS), and Computational Aerosciences (CAS), the latter of which fell under the NASA aeronautics program. In 1992, while the first massively parallel computing architectures with O(100) processors were coming online, the stated NASA HPCCP goals were to "develop algorithm and architectural testbeds capable of fully utilizing massively parallel concepts and increasing end-to-end performance", and to "develop massively parallel architectures scalable to sustained teraflops performance".[3] The CAS component of HPCCP focused on the Grand Challenge of "integrated, multi-disciplinary simulations and design optimization of aerospace vehicles throughout their mission profiles".

In 2004, the 10,240-cpu Columbia machine was powered up in the Advanced Supercomputing Division at NASA Ames. At the time, Columbia was the second most powerful machine in the world[4] and remains one of the top 10 most powerful installed sites today.[5] While NASA's HPC hardware remains state-of-the-art, the focus of the programs which employ advanced computing within NASA are substantially different from 10 or 20 years ago. On the one hand, the NASA Columbia machine is used primarily as a capacity computing facility, processing a large number of mostly modest size jobs of O(100) cpus. At the same time, the portion of these resources devoted to the NASA aeronautics program in general, and to CFD in particular, is much smaller than in the past. Certainly this is to be expected to some degree, with the rise of other computationally intensive disciplines (e.g., computational materials), and with the drastic reduction in the NASA aeronautics program over the last decade, coupled with multiple program reorganizations and refocusing. However, NASA HPC programs have been notoriously absent in recent government inter-agency HPC planning activities. For example, there is no mention of NASA aeronautics HPC activities in the FY 2006 budget supplement of the NITRD,[6] aerospace CFD activities are only briefly alluded to in the 2005 PITAC report,[7] while there is no mention of such activities in the recent National Science Foundation (NSF) report on Simulation-Based Engineering Science.[8]

Over the last decade, leadership in HPC endeavors has been increasingly taken up by other agencies such as the DOE with programs such as SciDAC,[9] and the NSF, through the formation of the Office of Cyberinfrastructure (OCI) and the funding of several large computational facilities, as well as within the Geosciences community.[10,11] The advocacy for increasing investments in HPC hardware, software, and associated research at these agencies has been driven in large part by the science community, with the argument that increased investments in HPC will lead to greater scientific discoveries, using simulation as a third pillar in the scientific process, along with theory and experiment.[9,12,13] While the science community has formulated a strong case for increased investments in HPC, the engineering community has practically abandoned the idea of state-of-the-art simulation-based engineering using HPC. Too often, engineering simulation software is viewed as a commodity, available through commercial software vendors, which is not central to the success of the engineering project, or as part of the core competency an engineering focused company or industry. The case for increased investments in engineering-based simulation has only recently been addressed in a recent NSF report,[8] and has yet to result in increased public investments in this area. NASA aeronautics HPC activities such as the previous CAS program fall squarely into this category. Although some portion of CFD research can be thought of as science-driven, such as turbulence research,[14] the CAS focus of "integrated, multi-disciplinary simulations and design optimization of aerospace vehicles throughout their mission profiles", is still appropriate today, and is principally engineering driven.

The case for HPC simulation-based engineering science has been detailed in the recent NSF report,[8] using various examples such as structural analysis and crash simulations, but with no mention of CFD-type activities. While structural analysis is perhaps the most pervasive simulation-based engineering application, CFD has traditionally been more at the forefront of HPC, both in terms of computational requirements, and in effective government-industry -academic collaborations, particularly in the aerospace fields.

Recently, there has been renewed interest in accelerating the pace of HPC activities in the US, with the establishment of the Competitiveness Initiative by the US administration,[15] and the proposed increases in budgets for HPC activities in agencies such as DOE, NSF and NIST. While many of these activities are science driven, a strong case can be made that engineering driven HPC activities should be given a high priority as well in any initiative dealing with competitiveness, since engineering lies at the interface of science/technology and economic competitiveness.

The recent reformulation of the NASA Aeronautics program represents an opportunity for NASA to capitalize on the increasing national attention to HPC activities and competitiveness. Some of the stated objectives of the reformulated program include focusing on fundamental and pre-competitive "long-term cutting edge research in the core aeronautics disciplines across all flight regimes" and providing "aerospace research that benefits the community broadly".[16] Historically, the "common themes of physics-based analysis tools and multidisciplinary design tools" have been pervasive within NASA aeronautics, from the days of the ILLIAC IV to the current reformulated program.[16] Successive external reviews of NASA aeronautics programs over the last two decades by organizations such as the NAE and others have repeatedly praised the world-class status and leading-edge technical contributions of the simulation-based engineering tools developed under these programs. The connection between the development of such tools and the broader goal of NASA Aeronautics as well as engineering contributions to a National Competitiveness Initiative has been stated succinctly in the Decadal Survey of Civil Aeronautics as: "an important benefit of advances in physics-based analysis tools is the new technology and systems frontiers they open".[17]

This renewed focus on foundational research, coupled with an historical record of leadership and excellence in simulation-based engineering tools, constitute an opportunity for NASA to take the lead in advocating for and participating in engineering-driven HPC activities at the national level. In some sense, NASA would be resuming the leadership role it once enjoyed in this area 30 years ago, using the high-technology field of aerospace engineering as a driver for other engineering disciplines. Of course, in an environment of limited resources, such an undertaking must be done in collaboration with other relevant national agencies. Duplication of research more appropriately performed at other government agencies and in industry should be avoided, but other on-going HPC activities should be leveraged as much as possible while concentrating on the areas most appropriate to the NASA mission.

Various other communities have devoted large efforts to formulating the case for increased investment in HPC, such as the original Accelerated Strategic Computing Initiative (ASCI) of the DOE, and more recently endeavors such as SciDAC,[9] and the ScaLES workshops.[12, 13] Similarly, the Geosciences community has produced a two volume report on the technical and societal needs and financial requirements for a broad collaboration among geoscientists centered around petascale computing, including the advocacy of a new petascale computational facility.[10, 11]

The intent of this paper is to argue that the NASA Aeronautics Research Mission Directorate (ARMD) should take an important if not leading role in the advocacy for HPC activities in Simulation-Based Engineering Science, as discussed in the NSF report.[8] Furthermore, NASA should more fully engage the various on-going and developing national HPC efforts. In the following, we propose a long term vision for HPC within the NASA Aeronautics program, based on the formulation of several Grand Challenge problems, and assessment of the advances in foundational capabilities these would enable, as well as an evaluation of the computational requirements, barriers to progress, and required areas of investment of these challenges.

II. Grand Challenges

One of the obstacles to increased investments in HPC for simulation-based engineering projects is a notion that engineering problems do not require ever increasing amounts of computational power. While it is true that capabilities exist that are used successfully in every-day engineering calculations, radical advances in simulation capability are possible through the coupling of increased computational power with more capable algorithms.

Historically, simulation-based engineering capabilities have been developed and demonstrated using public funding. As the potential benefits of these technologies becomes apparent, industry has generally been very eager to adopt these technologies and integrate them into their product development cycle, often making the large investments required in term of hardware, software and personnel. However, once the capabilities become integrated, industrial emphasis more often turns to the goal of reducing the cost of this fixed simulation capability (generally through migration to newly available cheaper hardware) rather than seeking increases in simulation capabilities at fixed cost (i.e., by continually acquiring the latest available high end hardware). For example, the development of computational aerodynamics in the aerospace community was characterized by a continual drive to higher fidelity (and more accurate) methods from the 1970's to the 1990's, beginning with panel methods, proceeding to linearized and nonlinear potential flow methods, inviscid flow (Euler) methods, and culminating with Reynolds-averaged Navier-Stokes (RANS) methods in the 1990's. Throughout this development period, effective use of these methods required investment in the most capable high-end computing hardware available at the time, an investment industry was more than willing to shoulder in return for the improved simulation capability. In the early 1990's, it would not be uncommon for larger corporations to house HPC hardware on a par with most of the leading national labs, for either aerodynamics in the aerospace industry, or crash simulations in the automobile industry. However, the last decade has seen a stagnation of the capabilities used in aerodynamic simulation in the aerospace industry, with RANS simulations having become the high-fidelity simulation method of choice, and emphasis being directed towards reduction in the cost of RANS simulations through migration towards more cost-effective but no longer leading-edge computational hardware.

As long as the potential advances enabled through advanced algorithms in combination with leading edge computational hardware are not made evident to industrial users, simulation-based engineering capabilities will remain stagnant in these areas. Arguably, the most appropriate place for the development and demonstration of these capabilities resides in the public sector.

The science community has been very effective in demonstrating scientific discoveries that can be made possible through advances in HPC. Similarly, the engineering community should strive to demonstrate the benefit of HPC advances in this field. The formulation of Grand Challenge problems is used for this purpose in this section. The intent in defining these Grand Challenge problems is not necessarily to lead to a focused program that seeks at all cost to demonstrate the described Grand Challenge capabilities, but rather to set long-term goals for driving the essential developments that will be required to achieve these and other types of advances in simulation capabilities, and to illustrate the new frontiers that such capabilities would enable.

A. Complete Flight-Envelope Characterization (Digital Flight)

Currently, high-fidelity computational fluid dynamics simulations, most notably RANS simulations, are generally used to study only a small number of critical points (such as cruise conditions) within the flight envelope. In addition to cruise condition design, a characterization of the complete flight envelope of the vehicle is required early on in the design process, in order to provide the basis for the range of structural loads or other effects, such as heating, which may be encountered by the operational vehicle, as well as for other tasks such as flight-control system design and development. The complete flight envelope of a vehicle is most often obtained through a combination of wind-tunnel experiments and low-fidelity engineering-based simulation tools. A long-standing grand-challenge within the computational aerodynamics community is to use high-fidelity computational fluid dynamics tools to provide a characterization of the complete flight envelope of an aerospace vehicle.[18] Typically, two types of computational approaches may be considered to this problem. In the first, an aerodynamic data base is constructed by computing static force and moment (and possibly other quantities of interest such as loads/heating) at a discrete (but large) number of points covering the entire flight envelope. These may be augmented by static and dynamic stability derivatives in the vicinity of each point.[19,20,21,22] In the second approach, a full time-dependent simulation of a maneuvering aircraft is to be simulated using computational methods, which most often requires the inclusion of multidisciplinary effects such as aerodynamics, structures, and the flight-control system, in order to provide a realistic description of the physical process.[23,24,25]

While both approaches offer complementary benefits and drawbacks, the computational challenges associated with either approach are enormous. For example, even in the simplest static data-base analysis case, the large parameter space constituting the flight envelope (involving multiple instances of flow parameters combined with configurational parameters such as control surface deflections) can easily lead to the requirement of computing $O(10^6)$ individual steady-state cases, which may correspond to $O(10^2)$ *years* of computational time, given the computational rates of current-day RANS solvers on commodity computer clusters.[18] For dynamic (digital flight) simulations, the computational challenge can be even larger, given that a few seconds of real-time flight simulation may correspond to hundreds of hours of computational time when adequate spatial and temporal resolution are considered.

Clearly, complete digital flight-envelope characterization is not feasible today in a timely or cost-effective manner, due to the projected computation expense with current-day hardware. Additionally, current-day computational capabilities are neither accurate, efficient nor robust enough to be used on such a grand scale.[26] However, advances in hardware coupled with the development of more effective algorithms, including techniques such as reduced-order methods, which enable the systematic generation of cost-efficient low-dimensional representations of large-scale systems,[27] can be expected to result in tractable flight-envelope characterizations in the not too distant future.

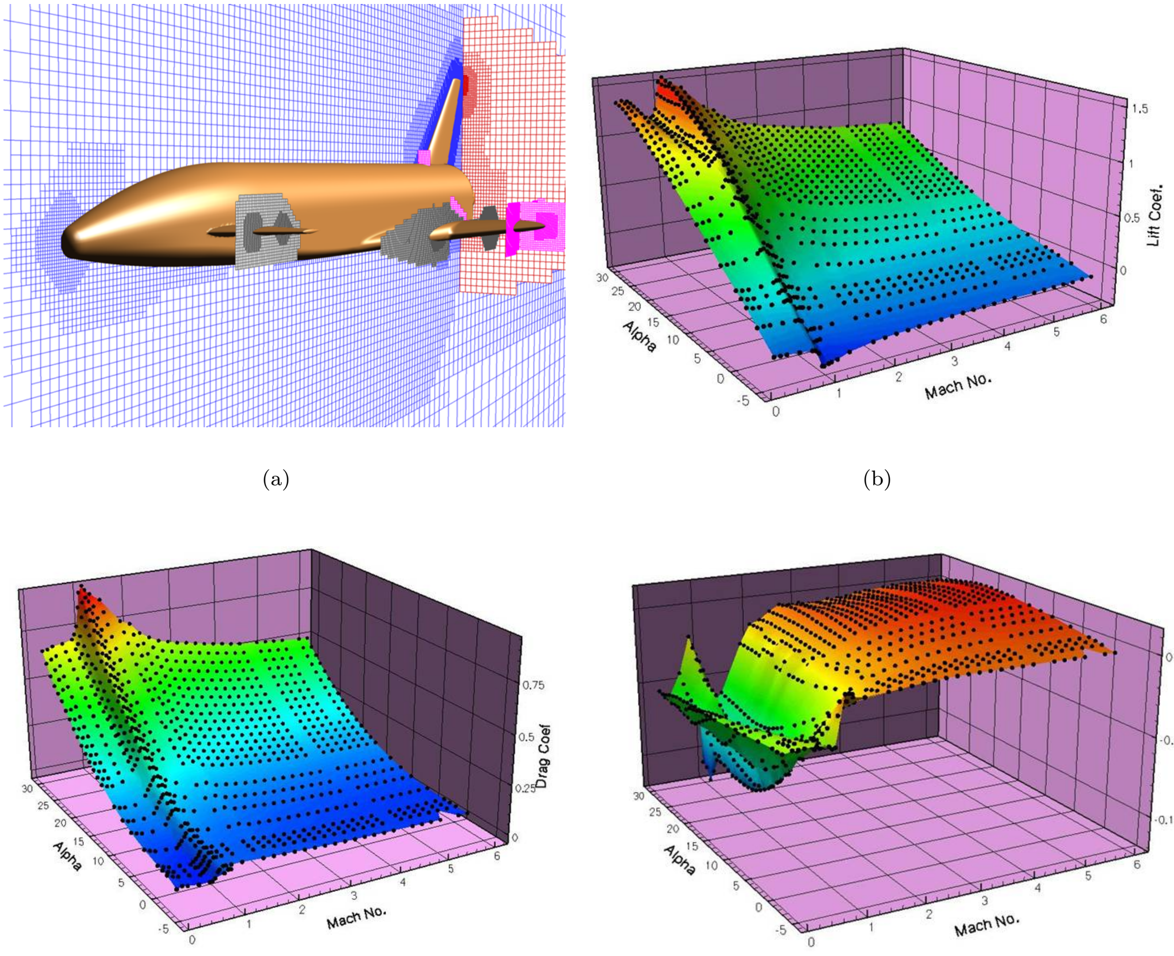
The payoff of achieving a reliable computational capability for complete flight-envelope characterization would be enormous. Aerodynamic data-base acquisition remains a costly and risk-prone endeavor in almost all aerospace vehicle design programs. Wind-tunnel based approaches have known deficiencies in various areas of the flight envelope, and wind-tunnel to flight scaling issues may involve unacceptable or (even worse) unexpected uncertainties. The reliance of current computational methods on non-physics based empirical or low-fidelity engineering methods (largely dictated by computational costs) involves large uncertainties and results in these methods being confined to conceptual design phases in many cases. Numerous recent aircraft programs have incurred delays and/or cost overruns as a direct result of unforeseen flight-envelope characteristics. The need for an accurate aerodynamic data-base characterization early on in such programs is of increasing importance as additional performance and lower cost are sought and design metrics become more interdisciplinary. Computational methods offer the most promising long-term approach for achieving such a capability, driven in large part by increasing computational capabilities and diminishing costs. In addition, computational methods allow for rapid assessment of configuration changes or other design decisions, while at the same time enabling a physical understanding of potential problem areas. The F-18-E wing drop problem provides a good example of this aspect, where computational methods were used to identify the mechanisms associated with the uncommanded transonic wing roll problem of this new configuration.[28]

Thus, while complete flight-envelope characterization is not cost or time effective in the current environment, the constant exponential increase in available computational power ensures that at some point simulation-based flight-envelope characterization will become both timely and cost competitive. However, in order to provide a useful simulation tool, the accuracy, reliability, and capabilities of current computational tools must be substantially increased over what is currently available.[26] In other words, investment must be made today in these technologies in order to precipitate the advent of a complete flight-envelope simulation capability in the future, as cost effective hardware becomes available for such tasks.

1. *High-Fidelity Aerodynamic Data-Base Generation*

The first approach to flight-envelope characterization entails the generation of an aerodynamic data base that can be used subsequently in the design process. In many programs, this is currently obtained through the use of low-fidelity engineering simulation tools, used mostly in the conceptual design phase, and with wind-tunnel experiments used for obtaining a more accurate data-base characterization later in the program. The goal of a digital flight-envelope characterization is to replace the low-fidelity engineering simulation tools with high fidelity physics-based tools for obtaining a more accurate data-base characterization early on in the program, thus reducing risk, and for supplanting portions of the expensive wind tunnel campaign required later in the program, thus reducing cost. Although costly, once the initial data-base has been constructed and archived, it may be used repeatedly for various applications by “flying” the vehicle through the data-base for such tasks as flight-control system design, or trajectory analysis.

In the simplest application, a purely static data-base may be obtained, by computing and archiving force and moment coefficients for various instances of the relevant parameters defining the flight envelope. The flight-envelope parameter space can be divided into flow parameters, and configuration parameters. For example the flow parameters will generally consist of the Mach number, the incidence and the sideslip (and possibly Reynolds number). The configuration parameters may consist of various control surface deflections such as ailerons, elevator, rudder etc. If we assume that computational solutions are required for five different instances of each parameter (for example Mach numbers of 0.7 0.75, 0.8 0.85, 0.9, five different incidences, sideslips, aileron deflection settings, etc.) the size of the complete matrix of required solutions becomes 5^p, which for the above example ($p = \textit{number of parameters} = 6$) corresponds to 15,625 cases. Adding to this estimate the desire to have more parameter instances, or additional effects, such as engine thrust settings and Reynolds number effects, and the total number of required simulations can easily increase up to order $O(10^6)$. Given an optimistic estimated turnaround time of one hour for a single point RANS solution on current-day computational hardware, the total time required for such a data-base generation extends into the end of this century. However, an $O(10^3)$ increase in computational power, obtained either by waiting out Moore’s law for 15 years, or by securing access to a machine of $O(100,000)$ cpus, results in the possibility of generating such a data-base in approximately one month of elapsed time. These estimates are on a par with various large-scale science applications being planned for the next generation of petaflops machines. However, such large-scale calculations for aeronautics applications have never been attempted or even considered. The impediments to such large-scale aerodynamic calculations are thus not related to the capabilities of current hardware technology, but rather, on the one hand, to access to such hardware, and on the other hand (perhaps more importantly), to the maturity of current solver and software technologies in successfully and automatically filling in the required data-base. While such a large-scale demonstration calculation may indeed be a one-of-a-kind demonstration with little industrial value, the intrinsic value of such a task is the identification of the solver and software technology areas that require further improvement



(a) (b)

(c) (d)

Figure 1. Cut-cell Cartesian mesh and computed carpet plots showing the variation of lift, drag, and pitching moment with Mach number and angle of attack at 0 degrees sideslip for an aerospace vehicle configuration. Reproduced from[29] with permission.

in order to result in a reliable capability, which will naturally result in a cost-effective design tool in the future as hardware costs decrease following Moore's law.

An example of work in this direction is provided in References,[29, 30, 31, 32] where an automatic aerodynamic data-base generation tool running concurrently on large numbers of processors has been demonstrated. Aerodynamic data-base generation has been performed using both a high-fidelity RANS code (OVERFLOW) and a lower-fidelity inviscid flow cut-cell Cartesian mesh solver (CART3D). The subject geometry is a liquid glide-back booster, shown in Figure 1(a), along with the mesh used for the inviscid cartesian approach, which contains a total of 1.4 million cells. Variations in the Mach number, incidence, and side-slip angles were considered for the aerodynamic data-base generation, which required a total of 2863 inviscid flow steady-state solutions. Figures 1(b) through 1(d) depict the computed aerodynamic data-base, as a carpet plot of lift, drag and moment coefficients with respect to the Mach number and incidence variations, at a fixed sideslip angle of 0 degrees, representing just one slice of the entire data-base. The efficiency of the inviscid cartesian approach enabled all 2863 simulations to be performed in a period of approximately one week. The more computationally intensive RANS simulations performed with OVERFLOW, using an overset mesh system of 8.5 million points, resulted in the production of 211 steady-state simulations in the same one week time frame. This difference between RANS and inviscid flow solution throughput is indicative of the additional computational expense of the higher fidelity RANS simulations, and raising the throughput for

RANS simulations to the levels demonstrated by CART3D will require an order of magnitude increase in computational power. However, a doubling of the mesh resolution in each direction, and the extension of the parameter space to include configurational variations, such as control surface deflections, can easily result in more than two orders of magnitude larger computational requirements, illustrating how quickly the desired long-term objectives of such an exercise can exceed available computational capabilities. Nevertheless, the development and demonstration of this capability is crucial for identifying areas that require special attention in the maturation of this technology, such as fully automated and fail-safe geometry modeling and mesh generation, robust flow solver capabilities, and automated job scheduling and data archiving.

2. Dynamic Flight Simulation

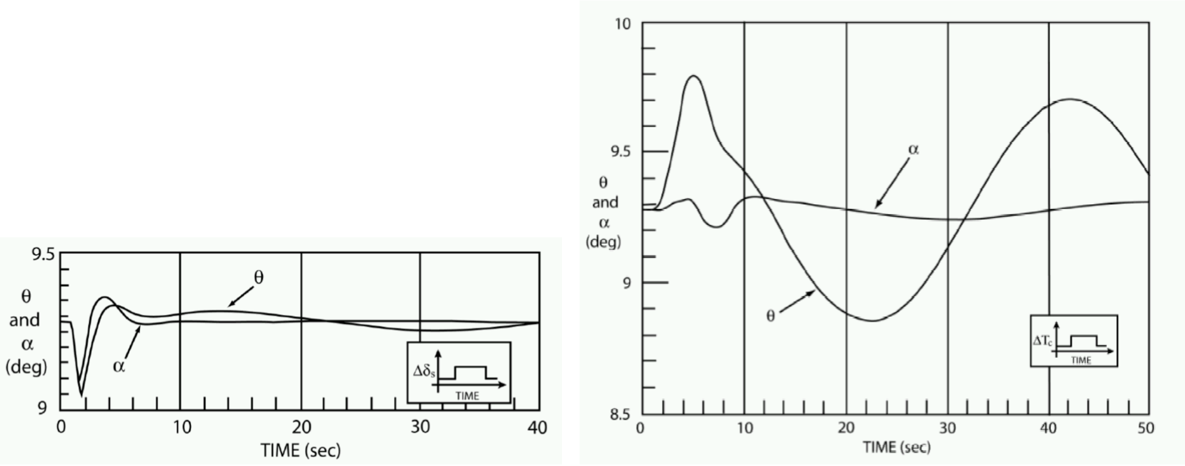
A more ambitious computational challenge involves the dynamic simulation of a time-dependent maneuvering aircraft, including all the relevant physical effects required for a realistic and useful simulation. At a minimum, these will most often include aerodynamics, structural analysis, and the flight-control system, although other important effects may need to be considered such as surface heating, or acoustics, depending on the application.

In this approach, each simulation consists of integrating the various disciplinary packages in a time-accurate coupled fashion for the full duration of the simulated vehicle maneuver or trajectory. As opposed to the previously discussed data-base approach, a complete flight-envelope characterization cannot be pre-computed and archived. Rather, each individual maneuver or trajectory will require a new time-dependent calculation. However, specific maneuvers can be devised in order to extract dynamic stability derivatives from the simulation.[24, 25, 19] The advantage of this approach, whether used to compute specific stability derivatives or to simulate complete maneuvers or mission profiles, is that it accurately takes into account the full dynamic, nonlinear, and inter-disciplinary coupling effects in the targeted regions of the flight envelope. While multi-disciplinary (i.e., aero-structural, aero-thermal) effects may be considered in the data-base approach, these will most likely need to be more tightly coupled in the dynamic simulation approach in order to capture the important nonlinear effects in the time-domain. This in turn places additional requirements on the individual solver efficiencies and the inter-disciplinary coupling strategies.

While the dynamic flight simulation approach can ultimately provide higher fidelity simulation for important effects in the flight envelope, it is also substantially more computationally intensive than the data-base approach, and these two approaches should be seen as complementary capabilities in any vehicle design exercise. For example, adequate characterization of the flight envelope can be obtained using the data-base approach in regions where the dynamical effects are known to be of secondary importance, while full dynamic flight simulation can be used to investigate more critical regions of the envelope. Similarly, most initial design tasks are best carried out using the aerodynamic data-base approach, such as the design of the flight-control system, since it would be virtually impossible to perform a dynamic flight simulation with no prior knowledge of the required flight control laws, although these may refined in latter phases using dynamic simulations.

The computational requirements for performing a single discipline (aerodynamics only) dynamic flight simulation on a realistic configuration using a RANS solver have been estimated in Reference.[18] The limited problem of studying the vehicle response to a step input to one of the control surfaces (elevator) or throttle is considered. Using data from reference,[33] reproduced in Figure 2(a), it can be seen that a physical time interval of approximately 60 seconds is appropriate (considering the phugoid mode) for this limited problem. Using published timings from current-day production CFD solvers on state-of-the-art hardware such as the NASA Columbia Supercomputer, the simulation of 60 seconds of flight time is estimated to require 1.5 days of computational time on 512 cpus of the NASA Columbia system.[18] Note that this estimate could easily be increased by one or more orders of magnitude simply by considering increases in spatial resolution (50 million grid points were considered) or temporal resolution (using a physical time step frequency of 50 Hz) or by including multi-physics effects such as structural analysis, heating, or other effects. Evidently, the task of computing a complicated flight maneuver with multi-disciplinary effects at high temporal and spatial resolution is well beyond the capabilities of currently available hardware.

However, several efforts have demonstrated the simulation of maneuvering flight-vehicles on current-day computational hardware using reduced fidelity and/or single disciplines. In references,[24, 25] an unstructured RANS solver was used to simulate rigid (non-elastic) fighter configurations undergoing forced motion maneuvers, using an unstructured mesh of approximately 3 million cells (for a half span aircraft model). In reference,[23] free-flight aircraft maneuvers are simulated using full aerodynamics, structural mechanics and



(a) Response to a step function elevator change of 2 degrees, from reference.[33]

(b) Response to a step function throttle change of 0.1, from reference.[33]

Figure 2. Response to step control inputs. Reproduced from.[33]

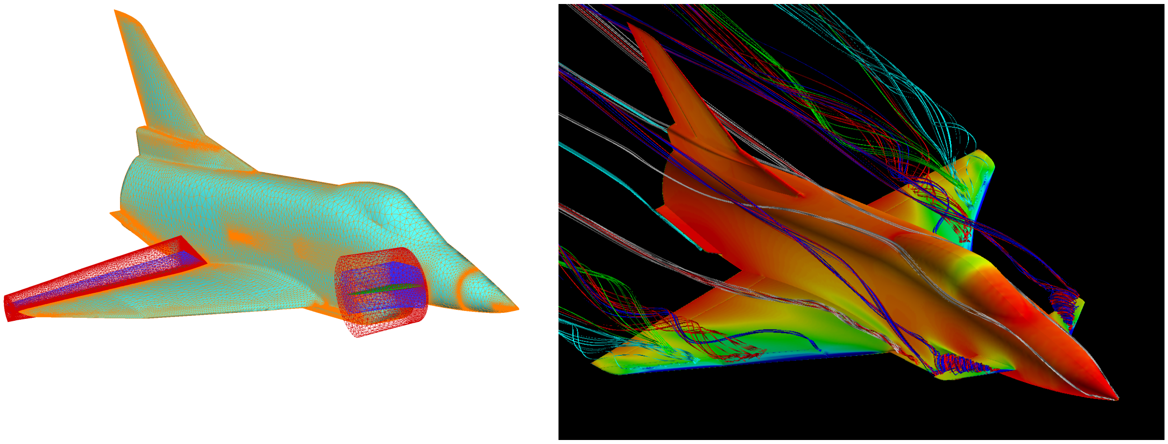
flight mechanics coupling, although the computationally aerodynamics simulation component is limited to inviscid flows. Figure 3(a) illustrates the configuration used for this study, which corresponds to an X-31 fighter aircraft with configurable control surfaces, which are modeled using an overset unstructured mesh approach. One of the defining features of these types of simulations is the complex flow physics that must be captured appropriately in order to obtain realistic loading profiles, as shown in Figure 3(b). Figure 3(c) illustrates a trimming scenario simulated for this configuration, involving trailing-edge flap deflections in order to zero out the pitching moment. A free-to-roll maneuver around the longitudinal axis was also simulated, using a simplified one-degree-of-freedom flight mechanics model, where the acceleration of the model is only dependent on the rolling moment and the moment of inertia about the x-axis. The unstructured mesh used for inviscid aerodynamics simulation contained 3.44 million points, and the finite-element structural model contained 273 nodes with 819 discrete translational degrees of freedom. The predicted position of the model during the maneuver is depicted in Figure 3(d), where the effect of the inclusion of the fully coupled elastic structure is shown to be significant. The total simulated time in this example, as well as in the simulations performed in reference[24] are of the order of one second, suggesting that the estimates from reference[18] described above may be overly pessimistic, although the total number of time steps for the trimming scenario shown in Figure 3(c) are similar to those discussed in Reference.[18]

The above examples illustrate the possibility of undertaking multi-disciplinary dynamical simulations of maneuvering vehicles on current high-end computer hardware, although the fidelity of current efforts in terms of grid resolution, model complexity, and inter-disciplinary coupling is still only a fraction of what is desired in the long term. Nevertheless, the importance of such demonstrations is the insight they provide into the areas of current technologies in need of further investment in order to result in reliable simulation capabilities on future cost-effective hardware.

B. Grand Challenges in Propulsion

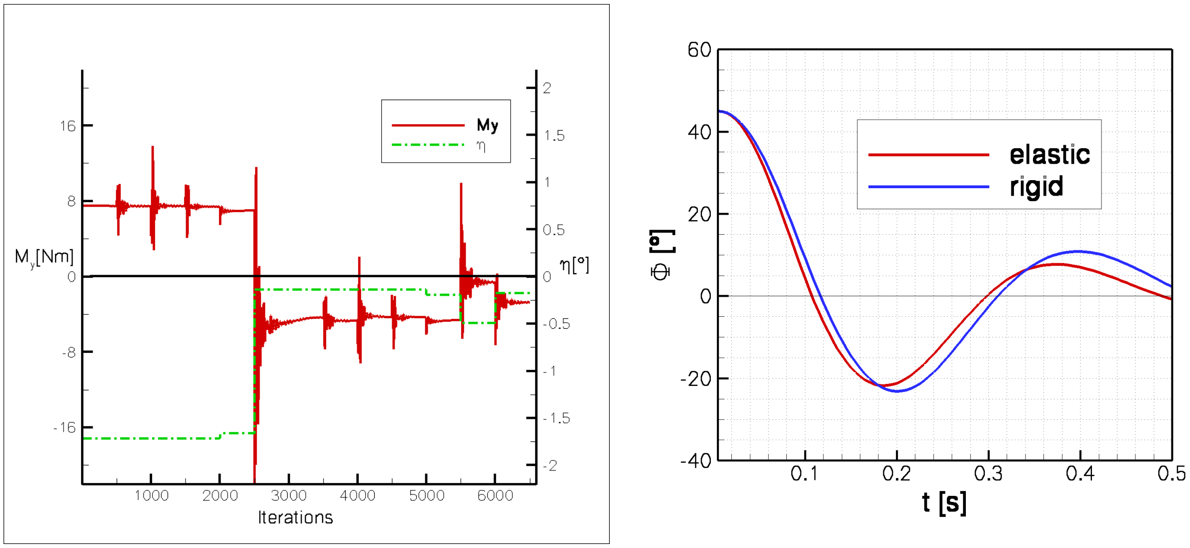
Propulsion technology, both revolutionary and evolutionary, has been a key contributor to the advances made in aeronautics over the years. For example, the new Boeing 787 Dreamliner is claimed to be 20% more fuel efficient than the products it is designed to replace, with advances in propulsion technology accounting for close to half of this benefit. Similarly, the new generation of high-bypass turbofan engines such as the GEnx are claimed to offer 15% better specific fuel consumption than the engines they will be replacing.

Engine design has traditionally been built on zero dimensional cycle models with maps that represent the different components such as a compressor or combustor. The components themselves have traditionally been designed with empiricism, one-dimensional and two-dimensional models, and occasionally with



(a) Illustration of unstructured mesh about X-31 aircraft configuration with overset mesh system for modeling of movable control surfaces.

(b) Computed flow topology during pitching maneuver at $\alpha = 14^o$ visualized as surface pressure distribution and off-body streamlines.



(c) Convergence of the pitching moment during trimming simulation and the corresponding trailing-edge flap deflection.

(d) Corresponding rigid and elastic position of the X-31 configuration determined by the roll angle during a free-to-roll maneuver around the longitudinal axis.

Figure 3. Fully coupled aerodynamic, structural mechanics and flight mechanics maneuvering aircraft simulations. Reproduced from[23] with permission.

three-dimensional steady-state simulations. Analysis of a component using three-dimensional time-accurate computational fluid dynamics (CFD) to predict performance and operability is being demonstrated as research, but has not yet been incorporated as standard design practice. The maps are generated by testing the individual components or scaling previous designs. However, this design approach does not allow reliable excursions from the previously known conventional design space.

To simulate a complete engine using three-dimensional unsteady CFD would allow a virtual engine testing capability, which would in turn provide a better understanding of component interactions, and through which real operations could be addressed. A Full Engine Simulation is an ideal Grand Challenge. It is complex, multi-disciplinary, and requires large scale parallel computing. Two efforts toward this goal have already been undertaken and are discussed in the next section. In the future, with improved computational hardware, better algorithms and models, and attention to validation, it should be possible to create a Full Engine Simulation capability that rivals the accuracy of physical testing.

1. Current status in 2007

There have been two primary efforts in Full Engine Simulation in the US. A NASA funded effort which culminated in a simulation of an entire GE90 engine using steady RANS at takeoff conditions was completed in 2004. The other principal effort is part of the Department of Energy (DOE) Advanced Strategic Computing (ASC) Initiative program led by Stanford University. This effort has demonstrated an unsteady RANS capability for turbomachinery coupled to an LES simulation of the combustor for modeling a 20-degree sector of the annulus of an engine. It is useful to examine the achievements and challenges of these existing programs in order to assess the requirements for future Grand Challenge full engine simulations.

Figure 4 shows a time line for producing a full engine simulation of the GE90 in the NASA simulation program. In essence, this effort really began in 1985 by being able to analyze a single blade row passage and built up to components such as the compressor, combustor, and turbine. The core was completed in 2001, and the entire engine was simulated at a takeoff condition based entirely on steady-state numerical simulations in 2004.[34,35,36] Computations were performed on NASA supercomputers and achieved near perfect scalability on 256 processors, and dropping to 75% parallel efficiency on 512 processors. A very useful feature of this work was the ability to couple the simulation with a cycle code or zero-dimensional code that efficiently balances the operating points of all the components. A zooming capability has been demonstrated[36] and is shown in Figure 5. The components have been run starting at the inlet and going downstream passing circumferentially averaged radial profiles from one component to the next. The back pressure or flow rate of a component has been determined from the cycle point for the actual engine, although this may not always be consistent with the numerical simulation. To create a simulation that balances the torque on the shaft, the cycle is rebalanced using maps created from the high fidelity simulation.

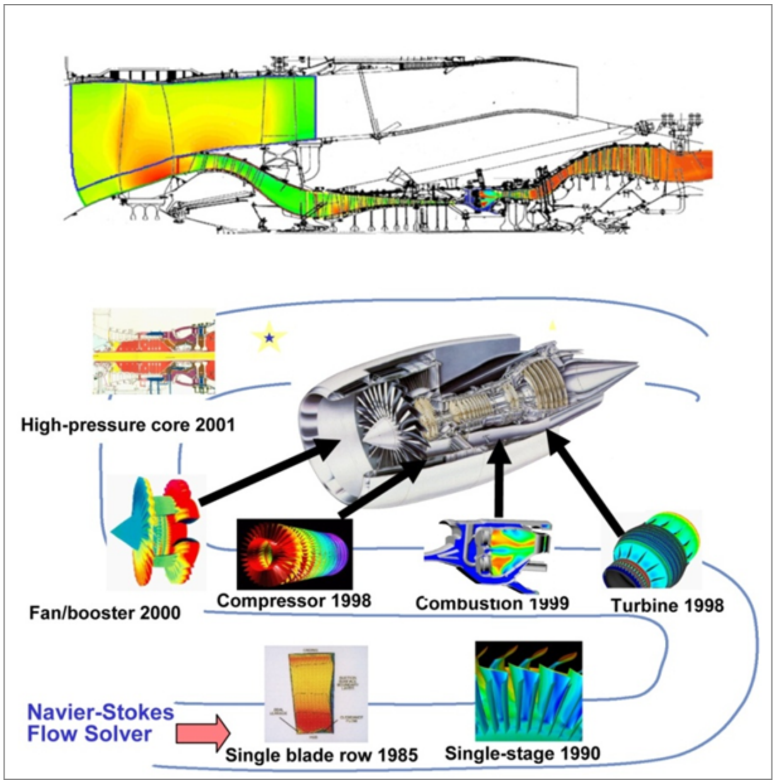


Figure 4. Illustration of GE90 Turbofan Engine used as test bed for high fidelity numerical engine simulation. Full Numerical Simulation with steady codes was demonstrated in 2004.

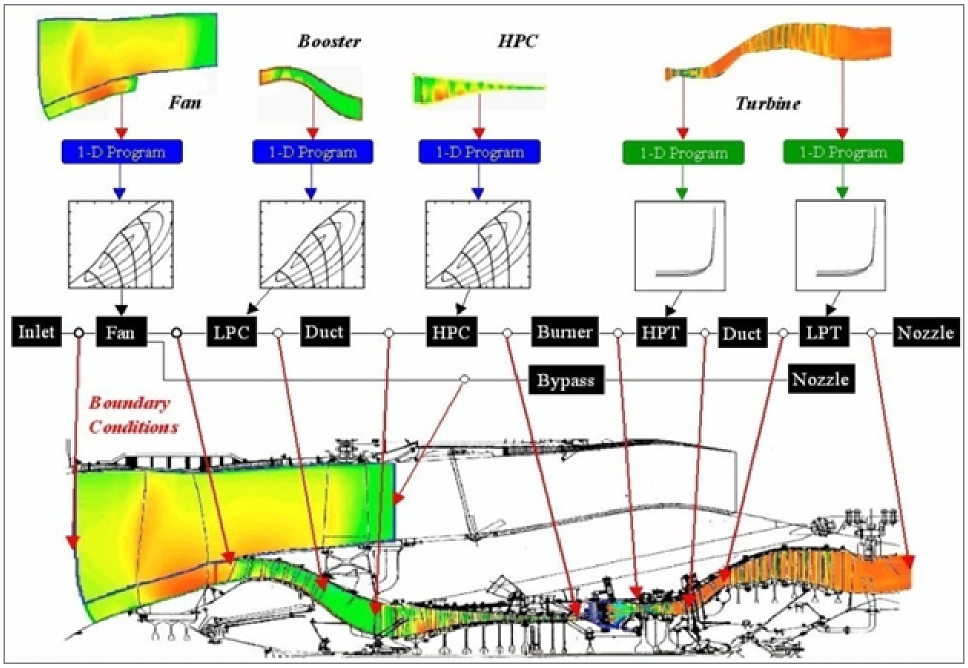


Figure 5. Process used by Full GE90 Turbofan Engine simulation. High fidelity component simulations are reduced to maps so engine can be balanced by cycle code.

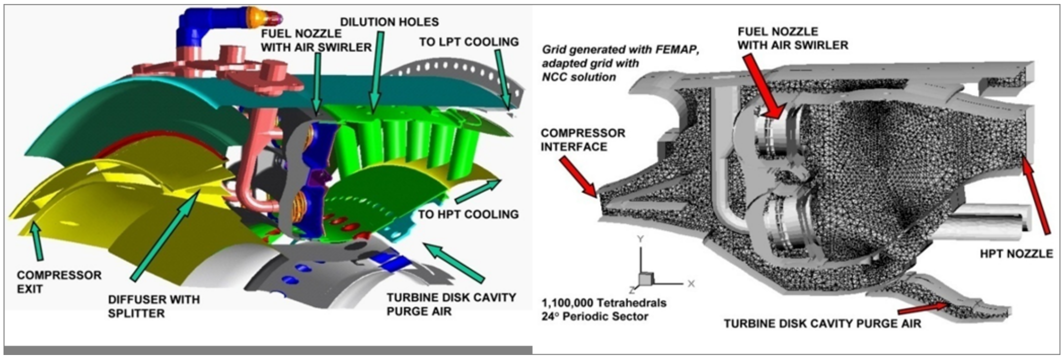


Figure 6. Geometry of the GE90 Combustor, and the unstructured grid used by the National Combustor Code (NCC).[34]

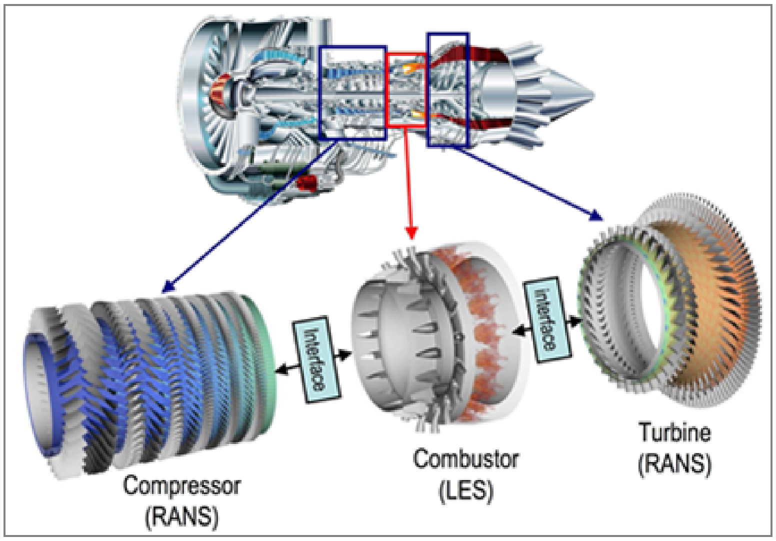


Figure 7. Decomposition of the engine for flow simulations for High Fidelity simulations performed at Stanford University under the DOE ASC program. Compressor and turbine simulations are performed with RANS models; Combustor simulation is performed with LES.[37]

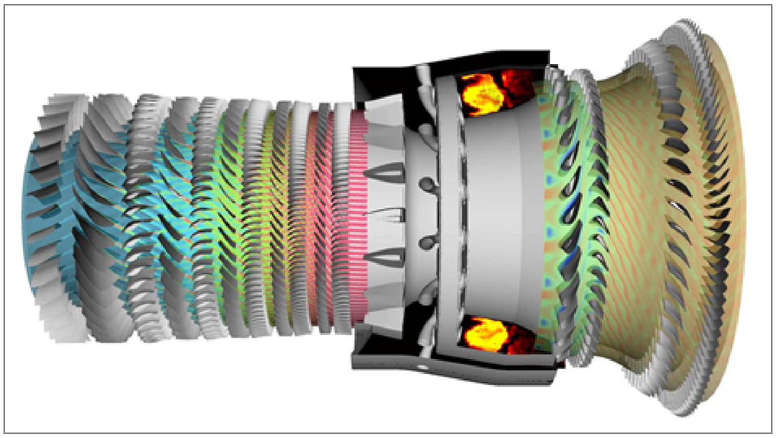


Figure 8. Simulation of 20-degree sector of the full high-spool of a Pratt & Whitney engine. Compressor and turbine shows axial momentum; combustor shows temperature.[37]

The combustor geometry for the GE90 is shown in Figure 6. An unstructured grid was used to grid the complex internal geometry of the combustor as shown in the figure. The combustor was solved with the National Combustor Code (NCC) which is a RANS code that also includes advanced spray combustion models. Good parallel efficiency of the solver with gaseous fuel was demonstrated up to 512 processors. The simulation of the GE90 produced a value for the Specific Fuel Consumption (SFC) that was within 1% of fleet data, although SFC predictions to within 0.25%, which is the typical engine-to-engine variation for production engines, remains the longer term simulation goal.

The Stanford effort supported by the DOE ASC program has focused on unsteady simulations and interface coupling of different codes. Figure 7 shows the methodology used to decompose the engine into its components. The compressor and turbine are simulated with a RANS solver and the combustor is solved with an LES code. The interfaces between the components are especially difficult to model accurately and consistently. Not only are different codes used for the various components, but the different components also have different physical modeling of turbulence. Simulation of a 20-degree sector of the core has been demonstrated as shown in Figure 8. To achieve suitable parallel scalability, good load balancing, and to address I/O and interface bottlenecks, several extensive re-writes of the code have been undertaken. The combustor code also includes a fuel breakup model that needs to dynamically reshuffle the processors to reflect the fuel motion.[37]

Although these two demonstrations have been very effective at advancing the state-of-the-art in engine simulation technology, severe limitations remain. On the one hand, the NASA GE90 effort involved only steady-state simulation tools. Some of the component simulations such as the high pressure compressor were also found to be difficult to converge, and the interface matching remains a weak point of this approach. Some of these issues are currently being addressed in NASA sponsored efforts as mentioned in Claus et al.[38] The Stanford effort has addressed improved modeling capability with unsteady RANS techniques and an LES approach in the combustor. However, compressor and turbine blade counts have been modified to limit the problem to a 20-degree sector of the annulus. In addition, cooling flows in the turbine have only been added as a bulk model which is even simpler than the source term model used in the NASA GE90 approach. Most of the GE90 effort used rig data to validate the components. This was possible since most of this validation took place at GE, although the details of this validation could not be published. The Stanford effort had access to only a limited amount of component test data. Therefore most of the Stanford effort is only partially validated.

Another example of high fidelity engine simulation is shown in Figure 9. The entire annulus of the first three blade rows of a military fan has been simulated[39] by the unsteady code, TURBO.[40] In this figure the annulus is unwrapped; the inlet is at the bottom and the exit is at the top. In this simulation, a total pressure distortion has been prescribed at the inlet to the fan, and the inlet total temperature is uniform, in order to replicate a fan test. The pressure distortion carries through the first stage, and it also induces a total temperature distortion on the exit as seen in the figure. This effect has been shown to match experiments reasonably well. Future efforts of this work are described in reference,[39] where the simulation of the entire annulus of the entire 7 blade rows of the fan will be undertaken on a grid of 500 million mesh points, using up to 800 processors on a DoD computer.

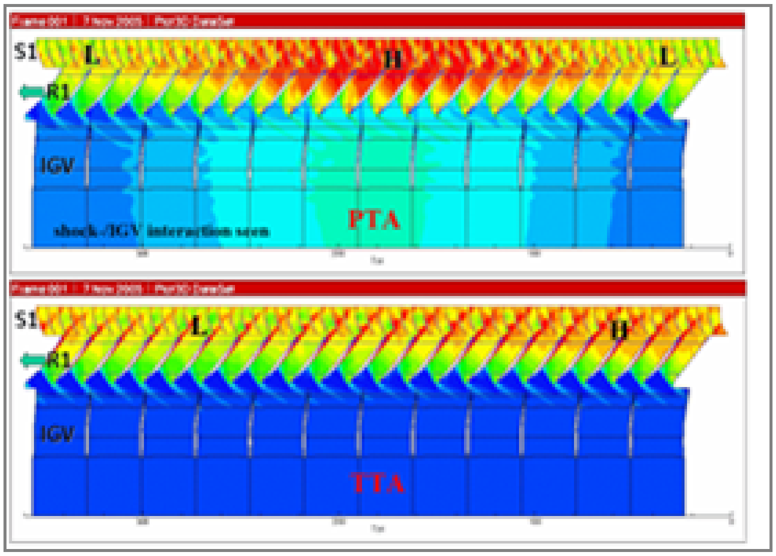


Figure 9. Full Annulus Inlet Distortion simulation through first three blade rows of a military fan. Snapshot of the absolute total pressure and induced total temperature distortion at 50% span (geometry not to scale).[39]

2. *Problem Complexity*

A defining feature of full engine simulations is the complexity of the problem both in terms of geometrical details which must be modeled, as well as in terms of the overall range of scales which must be adequately captured. The time and length scales of the full engine are captured at the large scale by a distortion or stall phenomenon as observed in full annulus fan simulations. At the other extreme, the cooling hole geometry and other small features such as turbulators place demands on the required grid sizes for resolving these features. Figure 10 is a schematic of a cooled high pressure turbine rotor. It shows the complexity of the cooling passages, pressure side bleed, and cooling holes. This schematic comes from a patent issued to GE for the idea of adding turbulator strips in the trailing edge plenum.[41] This small feature has been added to improve the durability of the turbine. At the elevated temperatures in a gas turbine engine, a 20-degree Celsius increase of metal temperature can decrease blade life by 50%.[42] On each blade there can be 400 cooling holes, and the GE90 has 68 first stage rotor blades. The number of holes in the first stage nozzle is similar.

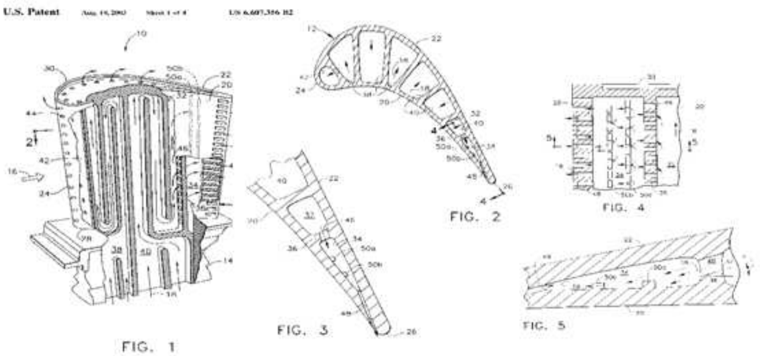
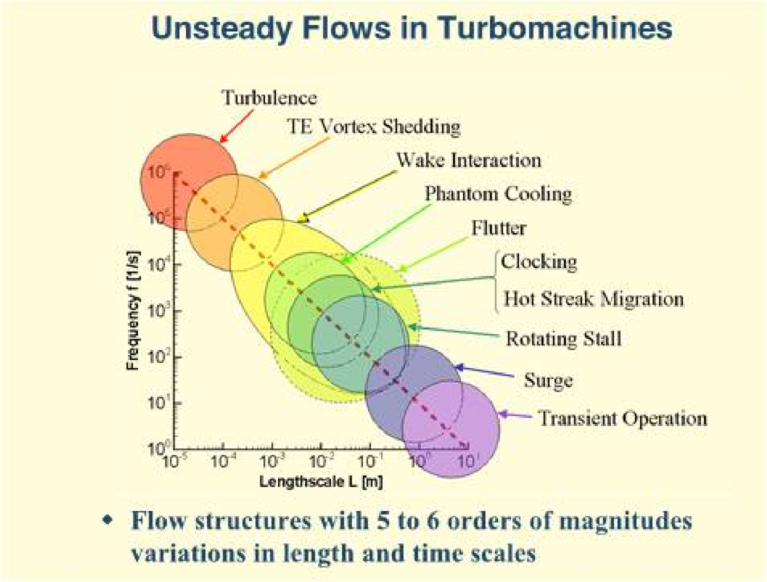


Figure 10. Schematic of a cooled high pressure turbine rotor blade from US Patent no. 6,607,356.[41] This patent added radial turbulator strips (shown as 50a and 50b) at the trailing edge slots.

Figure 11 is a schematic of the time and length scales that would be seen in a full engine simulation for the GE90. There are five orders of magnitude is physical size, and five orders of magnitude in time or frequency scales. As mentioned previously, the NASA effort addressed cooling flows in the high pressure turbine using source terms, and the Stanford effort added bulk amounts in an even simpler one-dimensional manner. Ultimately, high-fidelity full engine simulations must be capable of resolving all these important scales, down to the smallest features illustrated in Figure 10. It will take a combination of accurate solvers, validated physical models and huge computational power to address this disparity in time and length scales.



Unsteady Flows in Turbomachines

- Flow structures with 5 to 6 orders of magnitudes variations in length and time scales

Figure 11. Time and length scales for engine simulations.[42]

3. The Transient Full Turbofan Simulation: The Ultimate Propulsion Grand Challenge

A transient full turbofan multidisciplinary simulation represents the ultimate grand challenge for aerospace propulsion. Knowledge gained from the NASA GE90 effort and the Stanford/DOE effort should be leveraged as much as possible for such a project. A flowchart illustrating a potential path for accomplishing this goal is shown in Figure 12. Each blue box represents an intermediate challenge of projects that can be undertaken concurrently. The yellow boxes show modeling tools and capabilities that must be developed in order to make all the pieces work in unison. This flowchart describes an analysis capability, which could be followed through with a design capability, which would allow for multidisciplinary optimization of the individual and coupled components. However, the analysis capability must be developed and validated first, and the design capability can be expected to require substantially more computational resources.

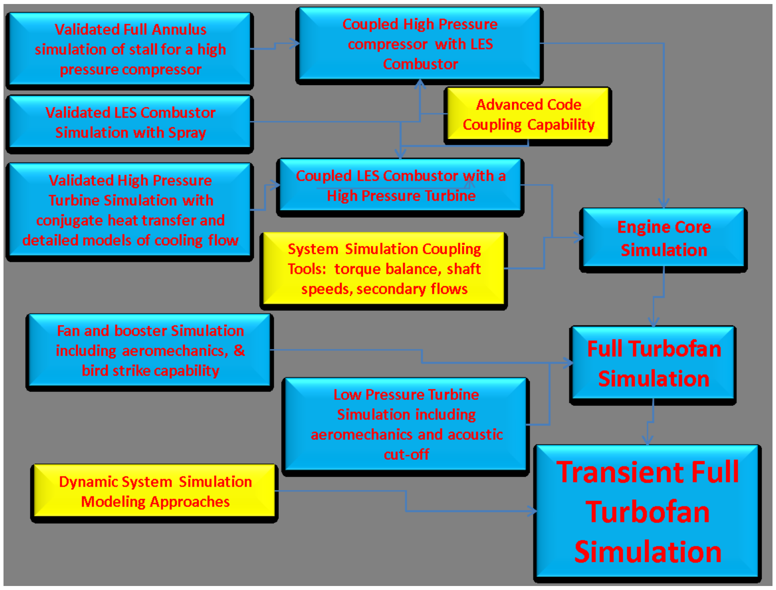


Figure 12. The ultimate Propulsion Grand Challenge is the Transient Full Turbofan Simulation. This flowchart shows one approach for achieving the fully coupled simulation capability with important intermediate Challenges which will be required.

A step by step program for achieving the Full Turbofan Simulation Grand Challenge must first overcome the following component simulation challenges:

1. High Pressure Compressor Simulation: The high pressure compressor is still a huge challenge. State of the art compressors typically include 7 to 16 stages. Stall prediction will require a full annulus simulation. Low speed operation is possible through variable stator vanes, which are not typically modeled correctly. Tip clearance varies with the thermal characteristics of the engine, and must be modeled by coupling with a structural solver. Cavities and bleeds must be prescribed and modeled. Another limiting factor is the large amount of time required for acoustic waves to travel upstream and downstream in a compressor. This dictates the total number of solver iterations required before a periodic state is reached.

2. LES Combustor Simulation with Spray: The Stanford effort has made tremendous progress in this area. However, one critical concern is the sharp delimitation of the combustion simulation at the combustor interface, which is inconsistent with the conventional wisdom that chemical reactions are still occurring in the first stage turbine nozzle. In these regions, cooling flows of pure air provide fresh oxygen to the combustion process, and the turbine nozzle accelerates the flow to near sonic or supersonic speeds depending on the turbine. This will challenge the use of incompressible LES solvers for the combustion regions. Another extremely difficult issue is validation. Combustor rigs are limited in pressure, sector size, or detailed instrumentation, and most available data is considered extremely proprietary.

3. High Pressure Turbine Simulation: The heat transfer and life predictions of a high pressure turbine are critical to maintaining durability in the engine. As mentioned previously, an increase of 20 degrees Celsius can decrease life by 50%. As shown in Figure 10, the turbine blades and vanes are hollow. To get metal temperatures requires a conjugate heat transfer model where the flow field is solved with CFD and the metal region solves the heat conduction equation usually with a finite element solver. The finite element solver grid can then be used along with the temperatures and pressures to

calculate stress and potentially life. Current practice has found that roughly one million grid points are required to accurately capture the physics of one cooling hole and the plenum and there is debate as to whether these effects can be simulated accurately with RANS models or whether more expensive LES approaches are required.

Assuming all these issues can be overcome and the individual components can be modeled to high accuracy, the next step would be to couple selected components such as the compressor and combustor and the combustor and turbine. The coupled compressor and combustor will add an ability to look at combustor screech. The coupled combustor-turbine would be a very valuable tool by itself. The issue of where combustion stops and the placement of the interface plane is very critical. It is also possible to add rotation to a combustor code that works with high speed flow to address this issue. An ability to simulate the entire hot section would be of extreme value to industry, but this challenge problem will require hundreds of thousands of processors with todays algorithms to get to the fidelity to predict emissions, metal temperatures, turbine efficiency, and life.

A system simulation tool is required to balance the torque and shaft speeds between the compressor and turbine as well as the secondary flows. When this is coupled through zooming and reduced-order modeling, then the Engine Core Simulation can be accomplished.

A fan and booster simulation challenge can be set up and run concurrently. The fan design is challenging because it must be able to get past flutter, and other aeromechanical issues such as separated flow vibration. One of the issues that is often ignored is that the rotor blades, especially the fan blades, change their shape based mostly on their physical rotation speed, and not a corrected speed. This means that the speed, and therefore the geometry and tip clearance will be different at cruise than at altitude.

The Low Pressure Turbine is a difficult component mainly due to aeromechanics issues, especially flutter. The geometry generally includes high-aspect ratio blades and high blade counts. For example, the GE90 has a 6 stage or 12 blade row low pressure turbine and the average number of blades per blade row is 147. The cavities dominate the loss, although transition modeling is critical for obtaining the correct loss. Transition is especially important at lower Reynolds numbers for smaller machines or at high altitude.

The complete Full Turbofan Simulation will require the development of a System Simulation Coupling Tool to couple the engine core, the fan and booster, and the low pressure turbine. With variable cycle engines comes the ability to transition control devices to optimize an engine for multiple operating points. A Dynamic System Simulation Modeling Approach must be developed for this application. This can then be coupled with the Full Turbofan simulation to create the Transient Turbofan Simulation. This will be a radical design tool for a variable cycle engine, and could also be integrated with the aircraft system simulation tools.

C. Computational Design and Optimization

A key difference between engineering-based computation and science-based computation is the engineering emphasis on design. In engineering, the motivation for essentially all computation is design; while in science, the motivation is more often to gain understanding of some phenomenon. In the Grand Challenges described in the previous sections, the incentive to use high-performance computation is to improve the design process by increasing model fidelity. However, the design process can be improved through other avenues beyond increased model fidelity.

For example, over the past twenty years, significant advances have been made in the application of optimization to computational models. Within aerodynamics, optimization algorithms have grown in complexity from inverse design methods to design optimization (in which an objective function is minimized). At the same time, the underlying aerodynamic models to which optimization has been successfully applied have progressed from inviscid flows to interacting boundary-layer methods to two-dimensional RANS and even three-dimensional RANS. Over the past ten years in particular, adjoint methods have been developed that significantly decrease the computational requirements in the (typical) situation in which the number of objective functions and constraints is much smaller than the number of design parameters.

Beyond aerodynamics, multi-disciplinary analysis and design has been another major research activity. In many instances, this research involves the coupling of existing lower fidelity models to create a multi-disciplinary model which does not overwhelm available computational power. Another factor in the use of lower fidelity models is that they are often easier to couple and more robust (though not necessarily more accurate) over a wide range of conditions than higher-fidelity models. However, coupling of higher-fidelity models has occurred in particular for aeroelastic problems.

Another approach to improve the design process is through the incorporation of variability. While most computational design is performed in a deterministic manner, in fact aerospace vehicles are subject to variability. Manufacturing variability and operational wear is unavoidable and can lead to geometries which are substantially different from the design intent. The environment that aerospace vehicles operate has variability. The impact of this variability can lead to variations in performance, or in the worse case, result in failure. As a result, analysis and design methods which account for this variability can and have had significant impact on the design of components of aerospace vehicles.

Probabilistic techniques applied to structural analysis and design have been in use in the aerospace industry for more than two decades.[43] For instance, NASA has employed probabilistic design methods to assess the reliability of advanced launch systems.[44,45] Furthermore, NASA has developed a probabilistic structural analysis code, NESSUS (numerical evaluation of stochastic structures under stress), which has been used in static and dynamic analyses of critical structural components of the space shuttle's main engine.[46,47] The Air Force has likewise been involved in implementing probabilistic design tools in an attempt to decrease the typically large safety factors involved in designing turbine disks.[43] The Air Force has also used probabilistic methods in predicting the life of other critical engine components[48] and airframes.[49]

In contrast to structural analysis and durability, there have been considerably fewer investigations involving aerothermal probabilistic analysis and design. Probabilistic aerothermal analysis is particularly challenging due to the complexity of the physical phenomena encountered and the resultant increase in computational requirements. Until recently, probabilistic aerothermal analysis and design have been deemed prohibitively expensive. However, the continual increase in computational power is making probabilistic analysis possible in a number of practical applications.

In the remainder of this section, we briefly describe two specific grand challenges in computational design and optimization. In the first, we propose a time-dependent high-fidelity optimization problem. In the second, probabilistic design of cooled turbine blades is considered.

1. *Time-dependent Optimization of a Rotorcraft Blade*

Rotorcraft flows provide a significant challenge to computational fluid dynamics algorithms. Transonic effects occur on advancing blades in forward flight, while boundary layer separation occurs on retreating blades. In other flight conditions, Blade Vortex Interaction (BVI) can occur in which a blade passes near or even intercepts the vortical wake from the preceding blade. BVI is a major source of rotorcraft noise. In particular, the severity of BVI-induced noise is exasperated because BVI commonly occurs when helicopters are near the ground.

Further complicating the analysis of rotorcraft is that many situations require multi-disciplinary analysis due to the strong coupling of rotor aerodynamics and structural dynamics. For example, the unsteady forces resulting from boundary layer separation reduce performance and increase blade fatigue. In extreme situations, the resulting blade vibrations actually limit the operating envelope of the helicopter. As a result of the close coupling of aerodynamics and structures, rotorcraft design has employed multi-disciplinary analysis for over thirty years; however, until recently, the disciplinary models have been lower fidelity.[50]

In a recent review, Strawn, Caradonna, and Duque state that remaining challenges for the application of CFD to rotorcraft center on modeling of (1) retreating blade stall and (2) rotor wake evolution.[51] In the case of blade stall, the difficulty is on adequate modeling of transition and turbulence. The rotor wake evolution challenge lies in the accurate convection of the vortical structures for many core lengths. They estimate that accurate convection of the wake may require meshes with greater than one billion grid points, unless dynamic adaptive methods or other vortex preservation algorithms are employed.

Clearly, high-fidelity rotorcraft analysis represents a major hurdle. In the context of computational optimization, rotor blade design represents a challenge that will require significant advances to harness the potential of high-performance computing. In particular, due to the presence of unsteady flows and blade motion, optimization algorithms are required for time-dependent problems. One natural possibility to develop an efficient optimization algorithm is an adjoint method. As mentioned previously, adjoint methods enable the computation of sensitivities for a given design objective at a cost which is essentially independent of the number of design variables. For unsteady problems, the solution of a time-dependent adjoint is required in which the adjoint is marched backwards in time about the nonlinear primal problem, from the final simulation time to the initial condition. To do this, the primal solution can be written to disk during the analysis run and read back in during the adjoint computation, or a combination of read from disk with recalculation using a checkpointing algorithm may be pursued.[52,53]

While the cost of a time-dependent adjoint problem is daunting, adjoint-based optimization has a history of being applied to large-scale, time-dependent problems in climate, ocean, and earthquake modeling.[54] Some adjoint-based optimization for time-dependent aerodynamic applications have appeared recently.[55,56,57] Furthermore, adjoints can be used to construct output-based error estimates to drive an adaptive method.[58,59,60,61] Although output-based error estimation and adaptation has been mostly applied to steady problems, recent work has shown the potential for these methods in the context of unsteady simulations.[62] More generally, an output-based adaptation approach coupled with adjoint-based optimization would be extremely attractive for rotorcraft applications. Furthermore, simultaneous adaptation and design for unsteady, high-fidelity analysis has a variety of applications throughout aerospace engineering beyond rotorcraft.

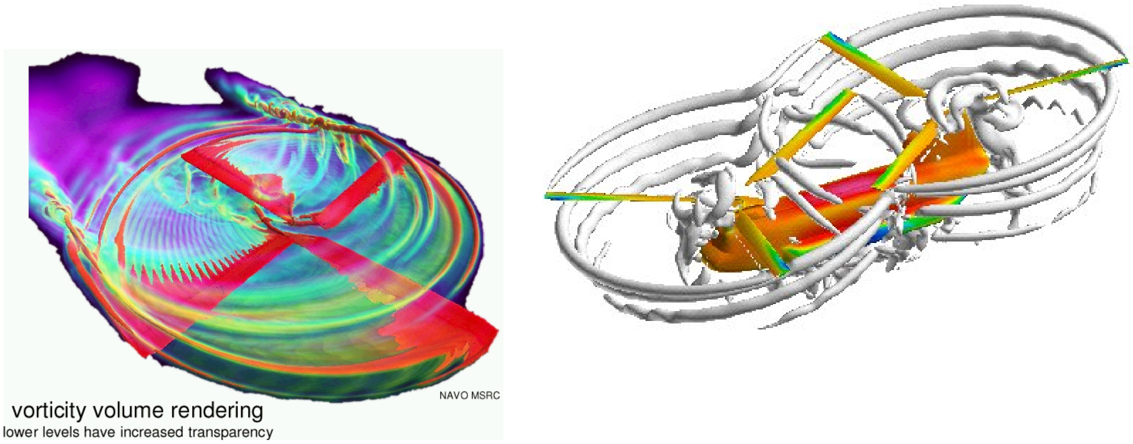
The computational requirements for performing an unsteady rotorcraft design optimization problem can be estimated based on state-of-the-art rotorcraft analysis capabilities,[63,64,65] and current two dimensional unsteady design optimization results.[57] Figure 13(a) illustrates an unsteady RANS simulation of an isolated rotor computed using the overset mesh code OVERFLOW, while Figure 13(b) illustrates a full configuration twin rotor calculation using the same methodology.[63,64] Typical OVERFLOW computations using a baseline grid of 24 million points are reported to require 4.7 hours of run time on 80 cpus to simulate 90 degrees of rotation, and computations using in excess of 100 million grid points have been demonstrated. In Figure 14, reproduced from reference,[57] the shape optimization of a 2D pitching airfoil with prescribed target time-dependent lift and drag profiles is seen to require of the order of 50 to 90 design cycles, where each design cycle corresponds to a time-dependent simulation followed by a time-dependent adjoint solution (integrating backwards over the same simulation time interval). Assuming the flow and adjoint solution costs to be roughly equivalent, this translates into a factor of 100 to 200 more computational cost over the analysis problem alone. Thus, an unsteady design optimization rotorcraft problem using 100 million grid points with a simulation time corresponding to a full 360 degree rotation of the blades could be achieved in 60 wall clock hours on 10,000 cpus, which is of the same order some recent state-of-the-art science applications, such as those described in the following section (see Figure 15.) Recently, NASA Langley's FUN3D unstructured mesh solver has been extended to include overset and actuator disk models, and has applied them to full rotorcraft simulations.[66,65] Since FUN3D operates on unstructured meshes and includes a discrete adjoint capability, the potential for adjoint-based adaptation and design is conceivable in the near future, although current calculations were performed on fixed meshes. Assuming some gain in efficiency due to adaptation, the computational estimates for an unstructured rotorcraft design optimization would be similar to the above overset-based estimates.

2. Probabilistic Design of Cooled Turbine Blades

Temperature-related damage to turbine blades is a leading cause of unscheduled engine removals for gas turbine engines and is one of the largest contributors to engine maintenance costs.[67,68,69] As noted earlier, a 20 degree Celsius increase of metal temperature can decrease blade life by 50%.[42] When turbine blades fail unexpectedly, an in-service engine must be removed from the aircraft for repair, often resulting in flight delays, cancellations, and thus lost airline revenue.

Predicting the life of a turbine blade is a multi-disciplinary problem requiring modeling of external aerodynamics (around the blade in the main gaspath), cooling passage flow, heat transfer, structural dynamics, and lifing. While these disciplinary simulations can be performed using high-fidelity models, in practice, turbine lifing analysis judiciously combines low and high-fidelity models to make the problem computationally tractable on commodity hardware. However, as argued earlier, the HPC hardware of today is the commodity hardware of tomorrow; thus, to take advantage of cutting-edge computational power in turbine lifing, it is critical to devote resources to this problem now. Furthermore, blade failure is often dominated by localized phenomenon such that higher-fidelity modeling would be an essential ingredient for improving the reliability of lifing analysis.

The need for HPC in turbine lifing is further motivated by the importance of accounting for variability in lifing predictions. For example, due to the complexity and expense of producing cooled turbine blades, manufacturing variability results in blades entering production turbines that have different flow capacities (i.e. different flow rates are observed in the cooling passages of different blades when subjected to the same pressure ratio). The allowed variability is often on the order of ±10% of the nominal flow rate. Sidwell



(a) Volume rendering of time dependent coupled aero-structural rotor simulation reproduced from[63]

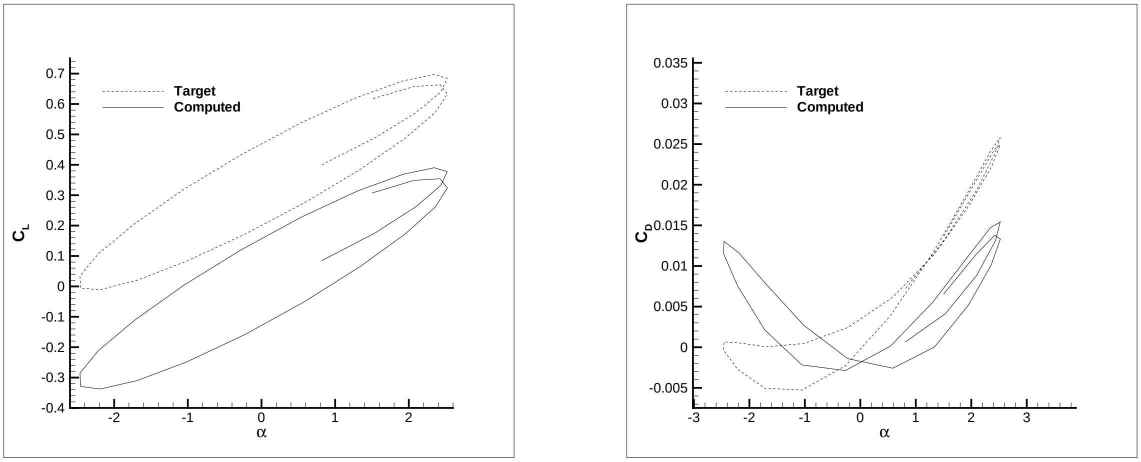
(b) Large scale full configuration rotorcraft simulation reproduced from.[64] Calculation required 12.5 hours per revolution on 256 processors of an IBM Power5 system using 78 million grid points.

Figure 13. Example of large-scale rotorcraft analysis simulations. Reproduced from[63, 64] with permission

and Darmofal have analyzed cooled turbine blades and shown that a leading cause of decreased blade life is cooling flow variability due to manufacturing.[70] Beyond the significant sensitivity of blade life to temperature (and hence cooling flow), the impact of low-life blades is further amplified by the large number of blades contained in a turbine row. Specifically, if the probability of manufacturing a low-life blade is p, then for a row of n turbine blades, the probability of having at least one low-life blade is: $1-(1-p)^n$. For example, if a low-life blade is manufactured at a rate of 1%, then the probability of observing at least one low-life blade in a row of 80 blades is approximately 55%.[70]

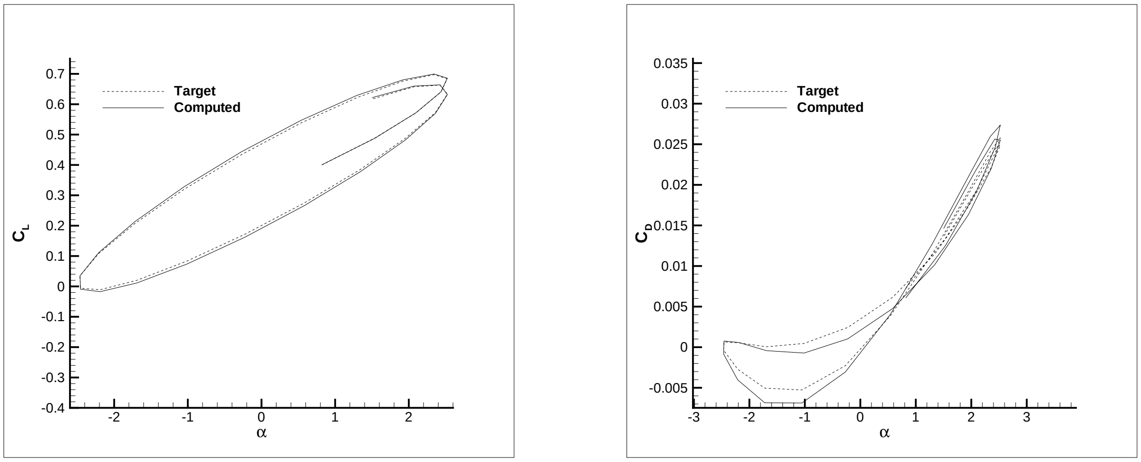
A common, low-fidelity approach for modeling variability is to predict the lifing of the nominal (design-intent) blade and then apply a historical variability (often based on Weibull distributions) to this prediction. A "first principles" approach to modeling the variability would be to perform probabilistic simulations in which variability is propagated through the lifing analysis. Thus, instead of requiring a single lifing analysis, multiple (i.e. thousands or more) lifing analyses will be necessary. Parallel implementation of probabilistic methods (such as Monte Carlo) is often trivially parallelized since the different solutions can be performed completely independently of each other; thus, beyond HPC issues for the underlying analyses, no additional work is required to leverage HPC for the probabilistic simulations. In this sense, probabilistic design is one natural target of opportunity for future HPC activities.

To estimate the computational requirements for a probabilistic lifing simulation for cooled turbine blades, we utilize estimates and timings from recent CFD calculations of a filmed-cooled turbine by Hildebrandt et al[71] and Burdet & Abhari.[72] In the former work, three-dimensional unsteady Navier-Stokes simulations were performed over a turbine stage including the individual cooling holes and simplified internal flow passages (e.g. no turbulator strips). The simulations used an overset grid with a total of 2.1 million points and required approximately 500 cpu-hours. While the mesh was concentrated in the boundary layers and around the holes, with $y^+ = 1-2$ for the first layer, this mesh is still considered relatively coarse. For example, Burdet and Abhari estimate that between 50-to-100 million grid points would be required to accurately simulate the aerothermodynamics of a film-cooled turbine blade.[72] If we assume that 1,000 simulations are required in the probabilistic lifing, and that the computational costs would be dominated by the CFD analysis, then a conservative estimate for a probabilistic lifing simulation is 2,500 wall clock hours on a cluster of 10,000 cpu's.



(a) Computed time-dependent lift on initial airfoil shape and target design time-dependent lift profile.

(b) Computed time-dependent drag on initial airfoil shape and target design time-dependent drag profile.



(c) Computed time-dependent lift after 90 design cycles compared with target design time-dependent lift profile.

(d) Computed time-dependent drag after 90 design cycles compared with target design time-dependent drag profile.

Figure 14. Example of time-dependent design optimization for two-dimensional inviscid pitching airfoil problem from.[57]

D. Other Challenges

The above examples represent but several instances of a wide range of computational problems which are of great importance in aerospace engineering. We note that these methods and the technologies required to drive them are truly foundational, in that they transcend all speed regimes, from subsonic, to supersonic, and hypersonic. Thus the development of these computational capabilities would revolutionize not only the traditional aeronautics disciplines such as fixed and rotary wing aircraft design practices, but also other important space operations missions such as access to space and hypersonic re-entry design problems.

There exist many other important disciplines and problems which have not been addressed in this paper. For example, computational aeroacoustics represents an important and demanding discipline area, particularly for civil aeronautics, which has not been discussed in our examples. Aeroacoustics alone can be expected to be a strong contender for future computational resources, and coupling aeroacoustics with any of the disciplines and challenges described in this section will lead to even larger computational requirements and algorithmic issues. Similarly, other disciplines will take on added importance depending on the driving applications, such as conjugate and radiative heat transfer for hypersonic problems. Not only can additional disciplines be considered, but many of the problems discussed above can be combined to form even more demanding problems. Thus, in the future, multi-disciplinary optimizations of virtual flight simulations may be considered, combining high resolution, advanced physics, and uncertainty quantification. Other methods currently considered unfeasible or not useful may also appear with increased computational power and algorithmic development. For example, the ability to combine experimental (wind-tunnel or flight-test) data with computational simulations through data-assimilation techniques in order to obtain more accurate and reliable simulation outcomes has not been considered in the aerospace community, although such methods are prevalent in other disciplines, such as atmospheric sciences. From the above discussion, it should be evident that these engineering simulation problems have essentially limitless appetite for increased computational power for the foreseeable future. However, in spite of the advances in computational hardware, significant barriers exist in order to enable these types of problems to be tackled successfully. In fact, there is a growing perception that advances in computational hardware are outstripping our ability to model these important engineering problems.

III. Barriers to Progress

A. Massive Parallelism

Almost all assessments of future high-performance computing architectures involve predictions of substantially increased parallelism, using upwards of 100,000 cpus or cores. These architectural trends are being driven by various technological factors, including the rapidly growing cost of further increases in processor clock speed, and the emergence of power density and cooling requirements as dominant considerations in supercomputer installations (producing a drive to larger numbers of slower but less power demanding cpus). While predictions of future architectural trends are always tentative, the overwhelming evidence, from the appearance of dual-core laptop computers, to the 131,072-processor IBM Blue Gene machine at Lawrence Livermore National Laboratory, point to large increases in parallelism in future supercomputer installations. Furthermore, the ratio between memory latency and cpu floating point speed has been continually increasing, and this trend is expected to continue at least over the next decade.

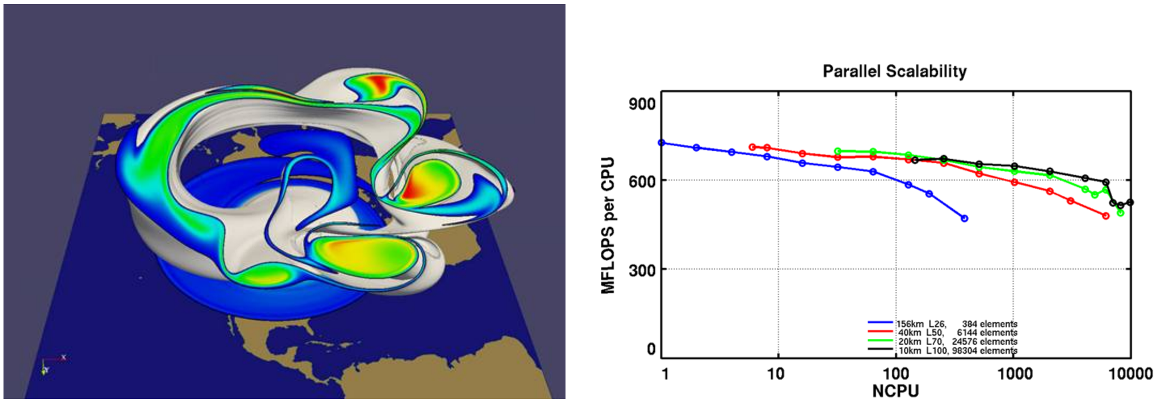
The NASA Columbia machine contains a total of 10,240 cpus representing one of the most capable supercomputer installations in the world. However, to our knowledge, only two real-world applications have ever been run on more than 4,000 cpus of this machine. This is partly due to the configuration of the machine, which requires hybrid communication strategies for such large runs, and has been optimized for larger numbers of smaller applications. However, Columbia contains a tightly clustered group of 2,048 cpus, with a single memory image, fast interconnect, and high reliability. While some Earth Science applications have made good use of this Columbia subsystem, virtually no NASA CFD codes are run regularly on this system.

One of the reasons for the lack of CFD applications on the 2,048-cpu subsystem is the inadequate scalability of these codes on massively parallel machines. One can only conclude from the benchmarks published so far,[73] that the scalability of most of these codes tops out around 512 cpus, although a few benchmarks above 1000 cpus have recently appeared.[74] This can be partly attributed to under recent underinvestment in this area within NASA, as well as to a policy of supporting capacity at the expense of capability computing, which further hinders any incentive for the development of large scale applications. Additionally, there is the persisting notion that computational engineering problems are not complex enough to warrant the use of computational resources on such a vast scale.

This situation is to be contrasted with that in the computational science community, where continual advocacy for higher-end simulations has been pursued based on the premise of‘ “discovering new science” and enabling new capabilities. For example, in the context of magnetically-confined fusion energy simulations, kinetic models containing billions of particles have been demonstrated on massively parallel architectures. For instance, the GTC code for the simulation of core turbulence in tokamaks sustains 3.3 Tflop/s over 70-hour production runs on 6,400 processors of the Cray XT3 Jaguar at ORNL, with similar scale runs routinely used for the tokamak edge turbulent transport code XGC1.[75,76] In the context of classical molecular dynamics, over a half-trillion atom simulations of the solidification of metals have sustained over 100 Tflop/s in runs lasting 7 hours on 131,072 processors of the BlueGene/L at Lawrence Livermore National Laboratory.[77] A recent large scale atmospheric simulation performed on the Red Storm supercomputer at Sandia National

Lab is illustrated in Figure 15, reproduced from.[78,79] This simulation was run on 7,000 cpus of Red Storm for 36 wall clock hours, using 300,000 time steps, and a spatial resolution of 1 billion grid points with ninth-order accuracy, and generated 1 Tbyte of data. While the DOE HPC roadmap calls for ever larger simulations in the years to come, simulations on this scale are considered intractable in the engineering community in general, and within the NASA aeronautics program in particular. Yet many of the issues discussed in the preceding section on Grand Challenges could begin to be addressed today if this scale of computational simulation were currently made available on a regular basis.

In the 1990's, the CAS program was successful in transitioning NASA CFD codes from small numbers (e.g., 8) of vector processors to hundreds of cache-based micro-processors, which represented the dominant configuration of HPC machines at the end of the program. However, as parallelism has increased to $O(10^3)$ cpus and beyond, little investment has been made to help NASA codes keep up with this trend. As a result, an order of magnitude of computational capability housed within the Columbia machine room, which was procured at considerable cost, is unaccessible to NASA CFD codes. In the future, as the level of parallelism in HPC systems continues to grow, without a concerted effort to address scalability and memory latency issues, the disparity between available and accessible computational capability for NASA CFD codes will grow to several orders of magnitude. While very large-scale CFD calculations are often one-of-a-kind simulations and may not be deemed relevant for industrial applications, there are several reasons for pursuing such goals. One the one hand, it is important to demonstrate the enabling simulation capabilities which are possible given state-of-the-art hardware and software, as these so-called hero calculations of today will quickly become production calculations of tomorrow. Furthermore, because parallelism is increasing across the board for all computer architectures, at the low and mid-range as well as at the supercomputer level, HPC installations of 100,000 or more cpus will be accompanied by department level machines of 10,000 or more cpus. This is already in evidence to some degree with machines such as the IBM Blue Gene, where a single rack machine with 2048 cpus can be procured for under $1M. Thus, production level engineering simulations will be expected that can make effective use of much more parallelism than is currently used today. To ignore these architectural trends will result in the self-fulfilling prophecy of simulation-based engineering products being confined to commodity capabilities which are only usable on low end commodity architectures.



(a) Isosurface of potential vorticity computed by large scale atmospheric simulation (HOMME) on Red Storm Supercomputer

(b) Scalability of parallel atmospheric simulation (HOMME) on Red Storm Supercomputer

Figure 15. Illustration of large scale massively parallel atmospheric simulation run on Red Storm Supercomputer at Sandia National Laboratory. Simulation required 36 wall clock hours running on 7000 cpus using 300,000 time steps and using 1 billion grid points with ninth-order spatial accuracy, and generating 1 Tbyte of data. Reproduced from[78] with permission.

B. The Problem of Scales and Resolution

One of the principal reasons that computational science and engineering problems are so demanding, is that governing physical phenomena involve such a wide range of spatial and temporal scales that interact in complex nonlinear manners. In aerospace engineering problems, these scales occur both due to geometrical considerations, as detailed in the previous description of computational propulsion problems, as well as due to the governing physical phenomena such as turbulence and transition. Estimates showing the computational intractability of fully resolved turbulent flow over aircraft at flight Reynolds numbers have appeared periodically over the last 30 years.[80, 81] The scale issue has been the principal driver towards consistently higher resolution since the dawn of computational aerodynamics. However, as community studies such as the AIAA Drag Prediction Workshops[82, 83, 84] have shown, lack of sufficient resolution remains the principal limiting factor in achieving accurate and predictive simulation outcomes. The importance of increased resolution for computational engineering simulations is hard to overstate. When the simulation resolution is increased, not only are the same physical phenomena simulated more accurately, but new physics can be resolved and simulated, rather than modeled, offering the possibility of greatly increased simulation fidelity. This is the principal idea behind large eddy simulation (LES), which is predicated on resolving all scales down to the universal turbulence range, where modeling is much more reliably achieved due to the universality of these sub-grid turbulence scales.

On the other hand, increasing simulation resolution can rapidly consume available computational resources, since for simple explicit schemes, the computational work is inversely proportional to the fourth power of the grid cell size. Thus, for a second-order accurate explicit scheme, dividing the cell size by a factor of two results in a four-fold increase in accuracy, but a sixteen-fold increase in computational work. However, one important aspect of most *engineering* simulations is that not all details of the physics are necessarily important in determining the accuracy of the simulation objective. This is to be contrasted with many science-based simulations, where a detailed description of all the physical phenomena may be sought, as for example in fundamental studies in transition mechanisms for turbulence. In engineering problems, the simulation objective may be a relatively small number of parameters such as force and moment coefficient, or a lower dimensional field such as surface pressure or temperature. The key to a successful and efficient simulation outcome lies in identifying the physical phenomena and scales involved which have the most important impact on the important engineering objectives, and concentrating the computational resources in these areas at the expense of the other less critical areas. Examples of this strategy for prioritizing resource allocation can be found in the use of time-implicit methods for aeroelasticity, where the smallest (turbulence) time scales are not deemed important for the simulation outcome compared to the longer time scale of the structural response. Other examples include the use of adaptive meshing refinement techniques in most types of steady and unsteady simulations, and in the future, increased use of sensitivity analysis and uncertainty quantification methods.

C. Multiple Physics

It is widely appreciated that the important engineering problems in today's environment are becoming increasingly multi-disciplinary. This implies, from a simulation standpoint, that additional physics from different disciplines must be incorporated into leading-edge simulations, and the appropriate or important coupling effects of these disciplines must be taken into account. There are various challenges associated with multiphysics or multi-disciplinary simulations. The most prevalent approach for building up multi-disciplinary simulation capabilities is to couple together existing mature single-discipline simulations. However, wide disparities in algorithmic techniques, simulation accuracy, resolution, and computational scalability, tend to complicate such exercises. While loose coupling is often chosen as the implementation path of least resistance, many of the important problems are highly coupled in nature and will require stronger coupling approaches, both for solution of the analysis problem, and for other important and emerging tasks, such as sensitivity analysis and uncertainty propagation and quantification. The coupling of multiple disparate disciplines also leads to the requirement to form inter-disciplinary personnel teams, since the complexity of these problems is such that mastering all disciplines simultaneously has become well beyond the reach of a single individual. Software complexity is also greatly increased, due to the wider range of physical phenomena to be simulated, the more complex coupling between these areas, and the different legacies associated with each component code.

D. Simulation Fidelity, Robustness, and Automation

The principal purpose of using ever more powerful computational hardware, higher resolution, and coupling additional disciplines is to provide higher fidelity simulation outcomes. However, in spite of decades of advances in all of these factors, there remain large areas where useful engineering simulation fidelity has progressed very slowly. For example, whereas RANS CFD methods have enjoyed considerable success at predicting on-design cases with relatively small areas of separated flow, the use of CFD methods for the prediction of off-design conditions, such as high-lift flows, cases with deflected control surfaces, or other extreme conditions for both external and internal flows, has long been a weak point of these methods. The recent Drag Prediction Workshop series has shown the difficulties in predicting even simple aerodynamic configurations with non-trivial flow separation issues in the transonic regime[82, 83, 84] in spite of an increase of over an order of magnitude in the resolution of the meshes used over the 6 year period of this workshop series. Similarly, a recent workshop at NASA Langley on the use of computational methods for stability and control (COMSAC),[26] which can be thought of as a subset of the broader challenge of digital flight discussed previously, produced general agreement that current CFD technology was not mature enough to be used in its present form for such tasks. In addition to the large computational resource requirements, three areas in need of substantial improvement in current CFD technology were identified in the workshop: accuracy and fidelity of predictions for separated flows, robustness of methods, and ease of use or automation.

Because computational engineering is generally concerned with the accurate prediction of a small set of design metrics, the added cost of increased fidelity is only warranted if it positively impacts these specific engineering objectives. While high-fidelity simulations such as RANS methods can provide intricate flow physics details throughout the domain, if the important design objectives are highly sensitive to particular phenomena which are not well captured or modeled by the simulation, such as turbulence or transition effects, then the entire simulation may be of little engineering value. In many cases, the uncertainty in the specific design objectives may be so large due to these effects, that a high-fidelity RANS type approach may offer no advantage over simpler non-physics based engineering approaches. As an example, a quote from reference[85] illustrates an example where a RANS method required 10^4 more computational effort than the prevailing non-physics based engineering method, but delivered no better accuracy for the design objective at hand.

Clearly, the usefulness of CFD and other associated physics simulation packages must be based on their ability to deliver accurate high-fidelity specific objectives, in order to justify their increased cost. This will require the development of better physical models, such as transition, turbulence, or LES and DES models, in addition to increased accuracy and resolution capabilities of discretization schemes. On the other hand, the requirements of CFD and other disciplinary physics packages can be expected to vary depending on the simulation objective. For example, the traditional focus on accurate drag prediction for transonic cruise configurations[84] is of less importance than the prediction of maximum lift for high-lift configurations, and similarly, for maneuvering aircraft, accuracy in absolute lift or drag may be traded for better moment, stability derivative, or even heating predictions for hypersonic applications. The precise manners in which such tradeoffs are made algorithmically, remain to be investigated.

In the context of production engineering simulations, the computational methods employed must also be exceptionally robust, since it will be impractical to manually intervene for isolated cases which fail in large computation projects. Due to the nonlinear nature of the flow problems, and the wide range of conditions which must be considered, even the most robust algorithms may encounter problematic areas, and a hierarchical recovery procedure may be considered, using lower fidelity but more realizable techniques as substitutes for the failed high-fidelity method. In addition, some measure of error estimation must be supplied to the user, or incorporated into the data-base, or propagated throughout the dynamic simulation, in order to ensure that the final result is not overly contaminated by the lower fidelity solution in these regions. These are examples technologies which would not necessarily be considered critical in today's computational environment.

Finally, because the end goal is the development of a multi-disciplinary tool to be used by a design engineer, the methods must be sufficiently robust and automated to be used by non-experts. Thus, large investments in automation frameworks, data-archiving, retrieving and analysis software must be considered concurrently with the more traditional algorithmic computational efforts.

E. Software Complexity

The rise in simulation software complexity is rapidly becoming one of the most important barriers to continued advances in computational simulation problems, a problem to which aerospace engineering is no exception. Whereas twenty years ago, a proficient researcher could contribute to the field through a self-developed simulation code of several thousand lines, today most leading-edge codes are team efforts which involve hundreds of thousands of lines of software. This is problematic, since it is often observed that the per-person productivity of a software project decreases rapidly with increasing numbers of contributing developers, as more effort must be spent on coordination, software verification, and organizational issues.[86,87] This is also particularly problematic in the academic setting, where a core group of experienced developers is difficult to maintain, and where graduate student residence time is generally insufficient for either building such a capability, or even thoroughly learning an existing capability.

Increasing software complexity is driven by many of the same issues discussed above, including the drive to massively parallel computing architectures, the increasing importance of multi-disciplinary simulation capabilities, and the use of better and more sophisticated algorithms, which now constitute the entry-level playing field for competitive simulation capabilities.

Strategies for mitigating software complexity issues must be considered carefully, since these approaches ultimately trade flexibility for standardization. Poor assumptions made up-front can have paralyzing consequences downstream in a large software development project, and strategies for maintaining or re-incorporating added flexibility must be developed. The expected continual increase in software complexity means that these issues will only increase in importance over time.

F. Technology Transfer

One of the defining features of computational aerodynamics has been the close interaction which has prevailed between academia, government laboratories such as NASA and the DoD, and industry. Computational fluid dynamics has progressed rapidly through constant exchange of ideas, personnel, and software among these groups, much more so than other disciplines such as structural analysis, which quickly became a mature engineering commodity tool supplied by third-party vendors, and much more so than other industries, such as the automotive industry, which views itself as a user of mature computational tools, rather than a driver in developing new state-of-the-art techniques. This is precisely why NASA aeronautics has been and will continue to be so well positioned to serve as a driver in computational engineering research and development at a national level.

The aerospace industry has traditionally been a strong driver in the continual development and adoption of higher fidelity computational methods over the last 30 years, from the appearance of panel methods in the 1970's, to the development of full potential methods in the 1980's, and the adoption of Reynolds-averaged Navier-Stokes methods (RANS) in the 1990's. The interest and participation of industry is documented by the many theoretical contributions made by industry research departments and the flow of top researchers between industry, academia, and government research laboratories. However, the last 10 years has seen a drop off in the level of industrial interest and expertise in participating in the development and deployment of new computational techniques, as the high-fidelity method of choice seems to have stalled out with RANS methods, and industrial efforts have turned to reducing the cost of this fixed capability through migration to lower-cost hardware, and a general lack of interest in participating in the development of new computational techniques. In many areas, the "next logical step" in computational tools, such as time-dependent detached-eddy simulations, CFD for stability and control, and high-fidelity nonlinear aeroelasticity, have made little progress in industry, although there has been considerable activities in these areas in academia and government.

There are several possible explanations for this development. On the one hand, the increase in software complexity makes the development and use of the latest computational tools more expensive, not just in terms of hardware requirements, but especially in terms of in-house expertise required to understand and manage such tools. At the same time, industrial cost-cutting has brought on the demise of most industry research departments, and a dramatic reduction in the level of in-house expertise available for such tasks. This in turn has led to the outsourcing of many computational tools to third party software vendors, many of whom are reluctant to engage in the development of specialized high-end software products, due to the need to server the broader software market. While this approach effectively reduces the cost of maintaining in-house industrial expertise, it results on a reliance on commodity engineering tools, and limits industry's ability to collaborate with and participate in the development of leading-edge computational tools with others in government and academia. Finally, newly developed computational tools must deliver sufficiently superior accuracy and reliability in the relevant engineering metrics in order to justify the added cost of these methods. As mentioned previously, this has often been lacking due to various physical modeling or other technology bottlenecks.

The need to engage industrial partners marks a fundamental difference between computational engineering and computational science. In the absence of industrial interest, the rest of the engineering research community runs the risk of finding themselves irrelevant in the longer term. Industrial input must be continually sought on the high-impact areas for computational engineering, and industry can contribute at the very least simply by advocating the utility and relevance of investment in these areas. As the research arm of an agency that participates in the design and operation of flight hardware, NASA Aeronautics Mission Directorate is perhaps uniquely positioned to fill this role.

IV. Areas of Investment

A. Hardware Availability and Scalable Solvers

One of the most effective ways for spurring advances in high-end computational methods is by increasing the availability of high-performance computational hardware. This has been recognized by the NSF in their Office of Cyber Infrastructure (OCI) objectives through the investment in Tier 1 through Tier 3 systems, by the DOE through the Advanced Simulation and Computing (ASC) program, and by the Department of Defense (DoD) through the High Performance Computing Modernization Program (HPCMP). NASA has historically been successful at providing high-end computational facilities to agency users, as well as academic and industry collaborators. However, as stated in section III, these facilities have progressively become more capacity rather than capability oriented, as the average job size currently run on the Columbia supercomputer is of the order of several hundred cpus. Even on the unique and relatively restricted-access shared-memory image 2048 cpu subsystem of Columbia, the most prevalent job size is 512 cpus. In many instances, it would be more cost effective to run the 512 jobs on regular standalone Columbia 512 SGI Altix nodes, while smaller O(100) cpu jobs are most cost effectively run on on commodity cluster computers. One strategy would be to isolate one or more Columbia nodes and limit these to run jobs of 256 or more cpus, while the 2048 cpu system could be limited to run jobs no smaller than 1000 cpus, since these cannot be run anywhere else within the agency. Future hardware acquisitions should also take into account the special requirements for advancing the development of high-end computational methods.

After a few decades in which individual processors have followed the observation known as “Moores Law from a 1965 paper by Gordon Moore,[88] clock rates have begun to stagnate, and further increases in aggregate computational power available to aerodynamic and other applications will apparently come primarily through increased concurrency. Core counts in the hundreds of thousands should characterize the first of the petaflop/s machines, scheduled to be available from Cray and IBM in 2009, and applications that wish to exploit the leading edge will shortly need to be decomposable with load imbalance in the parts per million on upwards of a million processors. The Cray XT system is favored by three Department of Energy laboratories for their first petascale computers, and the IBM Blue Gene system by two others. NSFs Tier One competition will inject another petascale platform into the national research environment within one or two years of the DOE machines.

The vast majority of the simulation community is similar to the NASA aerodynamics community in currently running most production jobs on approximately 1000 processors or less. In fact, approximately 60% of the processor-hours burned at the National Energy Research Supercomputer Center (NERSC) facility during calendar year 2005 were for jobs with concurrency smaller than 1,024 processors. Given this track record, it may not be wise for the agency with the most pronounced mandate for maintaining the competitiveness of the nations aerodynamic simulation to wait for related disciplines to learn how to colonize petascale hardware. Some investment in a prototype system and, more importantly, investment in the human capital to effectively harness the power of petascale systems for aerodynamics is required.

Of course, any modifications to the resource allocation strategy will need to take into account implications such as the reduction in agency capacity computing this would entail, as well as the possibility that the large jobs queues may be undersubscribed because of a lack of interest and capability for running such large jobs. Nevertheless, sufficient access to capable hardware is a pre-requisite for the development of tomorrow’s massively parallel applications. Additionally, an accompanying investment in the development of scalable solvers will be required, in order to develop and demonstrate the potential of running such large jobs. This two-pronged approach will be necessary today in order to meet tomorrow’s rapidly expanding level parallelism head on.

B. Algorithmic Development

It has often been pointed out that advances in algorithmic techniques have had an equally, if not much greater, impact on our ability to compute complex problems than advances in hardware capabilities.[89] Developing new enabling algorithms and incorporating these into usable engineering tools requires a broad investment strategy covering important areas of applied mathematics, computer science, as well as various aerospace engineering disciplines.

However, NASA's fundamental aeronautics research portfolio today has very little in the way of foundational algorithm development. Instead it is geared to programmatic interests, and is divided into categories such as subsonics, supersonics and hypersonics. Algorithm development is needed across all these categories (one example is better nonlinear equation solvers), and if it is truly foundational, is not likely to have a fully realized deliverable or be ported to a production NASA code in the expected two- or three-year time frame.

Many of the current enabling techniques in CFD are the direct result of past NASA investment in foundational algorithmic areas. For example, in the 1960's and 1970's, various in-house NASA Ames research groups were considered international leaders in algorithmic developments for computational fluid dynamics. Much of the high-order methodology used in finite-volume schemes today, both at NASA and through the community including many science-based applications, were developed in the 1980's by visiting scientists at ICASE, in close collaboration with NASA Langley scientists, who immediately put the technology to use in NASA codes.

In-house institutes such as ICASE, RIACS, and ICOMP have had a significant impact on NASA's computational practice, as did other ties to academia funded by past NASA aeronautics programs. In many of these instances, NASA aeronautics investment produced returns far beyond the immediate aeronautics mission drivers, benefiting the entire computational science and engineering community.

In the long term, the current mission driven application approach risks leading to the situation where hardware advances outstrip our capability to model complex problems. At some level, further advances using existing techniques may become difficult or impossible and new algorithms, new techniques and new software will be required. For example, it is possible that the advent of massively parallel machines with deep memory hierarchies will require a re-engineering (and complete rewrite) of existing algorithms and solvers in many NASA codes. The most evident example is the performance of tridiagonal or pentadiagonal implicit solvers in structured mesh codes such as OVERFLOW, which must be broken into ever smaller pieces, losing their effectiveness for stiff problems on increasingly parallel systems, which will shortly be approaching hundreds of thousands of parallel cores.

Additionally, it is likely that many of the barriers to progress listed in section III may be most effectively addressed through the development of novel techniques. For example:

- A growing consensus is emerging that the use of higher-order discretizations, such as Discontinuous Galerkin methods, may offer the best long term approach for capturing the widest range of scales at acceptable cost, particularly when combined with *hp* adaptive techniques, due to the superior asymptotic properties of these methods.

- In design optimization problems, the development of adjoint methods has proven to be an enabling factor in the solution of optimization problems with large numbers of design variables. This methodology has opened the door to new classes of problems previously out of reach, but many opportunities remain. Improvements in optimization algorithms customized for aerospace problems could provide orders of magnitude improvement in computing resources.

- Model reduction allows the systematic generation of cost-efficient, low-dimensional representations of large-scale systems resulting from discretization of aerodynamic systems. For example, Willcox applied model reduction to active flow control of a supersonic diffuser and achieved a three order of magnitude reduction in the size of the state space.[27] Reduction methodology has been developed and applied in many disciplines and is embedded in many commercial control technologies. In the context of turbulent flows, it has been applied for four decades under the rubric of principal component analysis.[90] Challenges in the application of model reduction techniques in aerodynamics include the robustness of reduction techniques, bounds on errors over ranges of operating conditions, and the cost of updating reduced order models if conditions evolve outside of ranges of validity.

- Another important area which has only begun to be investigated is uncertainty quantification, including comprehensive sensitivity analysis techniques and stochastic methods for multidisciplinary problems.

Advances here could revolutionize our capability for reliable, accurate simulations and robust design practices.

A modest investment in foundational algorithmic research would complement the necessary and more immediate programmatic needs of the agency. This would also provide the required internal expertise for guiding major future investment decisions concerning adoption of new technology for production software.

NASA could choose among a variety of ways to invest in foundational algorithm development, but this would most likely involve leveraging some combination of in-house and academic expertise. Tight academic collaboration through in-house or affiliated quasi-academic institutes is a potential avenue which has been used at NASA with great success in the past, and which has been replicated by others such as the ISCR at Lawrence Livermore National Laboratory, and internationally such as the Center for Computer Applications in Aerospace Science and Engineering (CASE) at the DLR in Germany, and at CERFACS in France.

C. Physical Modeling

Because all physical phenomena will not be able to be resolved for the future, physical modeling will continue to play an important role in many computational physics problems. As mentioned previously, this can be particularly important for computational engineering problems, where the objectives of interest may depend critically on the reliability of one or more physical models, even in the presence of overall high simulation fidelity. For example, the accurate determination of heating rates in hypersonic flows hinges on the ability to predict boundary layer transition, which cannot be simulated from first principles, even for the highest resolution simulations performed today, and must therefore include some degree of modeling.

As computational capabilities advance, new physical models will be required both to improve the accuracy and reliability of current approaches, as well as due to the changing nature of required physical models. Thus, for example, in the area of turbulent flows, future modeling investments may shift more towards LES sub-grid scale models in the place of past emphasis on RANS field-equation turbulence models.

Propulsion has some special modeling issues, which include liquid fuel breakup models for describing complex two phase physics, as well as chemical kinetics modeling. While combustion chemistry modeling has been pursued by other government research agencies such as the DOE, the particular aspects of chemical kinetics related to high speed flows are of principal interest to the aeronautics community and NASA in particular.

Transition modeling will continue to play a pivotal role in the success of simulation efforts for such diverse areas as hypersonic heating rates, low pressure turbine performance at high altitude cruise conditions for propulsion, and commercial aircraft high-lift design. Other areas which may be significantly dependent on advances in modeling include aircraft stall, control surface effectiveness, as well as tip clearance flows, blade cooling and endwall loss for propulsion problems.

The advent of more capable hardware coupled with algorithmic advances opens up new possibilities for improved physical model development and implementation. For example, multiscale methods may be used through upscaling techniques to develop computational models of phenomena which cannot be effectively resolved. Sensitivity analysis techniques can be used to determine important model parameters, and data-assimilation techniques using carefully designed experiments may enable the development of optimally calibrated models over specific domains of validity. Other techniques such as reduced-order modeling and stochastic methods can lead to more tractable and reliable physical models with quantifiable uncertainty bounds.

D. Supporting Experiments

NASA aeronautics has a long and rich history of investing in experimental programs in support of both fundamental flow physics research as well as full configuration aerodynamics with industrial interest. Over the years, a third objective of experimental programs has developed along with the rise in importance of computational methods, namely the need for experiments designed specifically for computational method validation.

Early examples include the NASA Ames transonic bump experiments[91] which served as validation benchmarks for new turbulence model development in the 1980's. In the 1990's under the Integrated Wing Design component of the Advanced Subsonic Transport focused program, a series of high-lift wind-tunnel experiments were performed, including extensive off-body flow field measurements, which were used for CFD validation purposes, with strong industrial participation.[92,93,94] Reynolds number scaling effects have been the subject of various wind-tunnel campaigns in the National Transonic Facility at NASA Langley, with supporting CFD validation studies. On several occasions, NASA experimental flight test data also has been used for CFD validation purposes, such as the NASA Langley Boeing 737 high-lift research flight tests in the early 1990's,[95] and more recently, the CAWAPI program based on the F16-XL flight test data.[96,97]

Experimental programs designed for computational validation purposes often differ significantly from experiments intended for flow physics research or aerospace product development. Such experiments should be easily reproducible in the virtual world, and open data-sets and shared (numerically represented) geometries are most suitable for establishing long lasting benchmarks which may be used for years to come for validation purposes. Current NASA investment in an NTF test in support of the AIAA Drag Prediction Workshop series provides a good example of a strongly needed experimental data-base on a generic configuration with broad community interest. On the other hand, in the propulsion area, although NASA has funded extensive experimental test programs over the years, there remains a complete lack of engine data sets including geometry, test conditions and test data which are not proprietary.

The ideal scenario is one in which any new computational methods development program incorporates an equivalent experimental program for validation of the developed computational methods. As an example, the digital flight program underway at the DLR in Germany contains a substantial experimental component, using guided and free motion wind tunnel models for developing a maneuvering aircraft CFD validation data-base.[23] In propulsion, the development of component coupling experiments designed for CFD code validation and to better understand component interaction would provide much needed valuable data-sets.

In the future, innovative experimental approaches may be envisioned in order to enable novel techniques such as data-assimilation for computational methods, where experimental data is used to increase the fidelity of the numerical simulations, and other techniques for incorporating effects such as sensitivity to geometric uncertainties or manufacturing variabilities, in order to assess the viability of numerical uncertainty quantification techniques, and robust design methodologies.

E. Software Issues

A massively parallel multidisciplinary optimization simulation facility is an ecology of hardware, software, and data, the complexity and organization of which are well beyond what any single human expert can comprehend. For similarly complex experimental and observational facilities, such as accelerators and telescopes, specialized "end stations" are built and staffed by experts to maximize the productivity of the physicist user community. This has inspired attempts to create application frameworks for scientific and engineering users of supercomputers that promise to: (1) allow interaction in specialized high-level languages or graphical user interfaces that hide the details of default parallel data distribution, geometric specifications, discretizations, and algorithms, (2) provide plug-and-play functionality for alternative formulations of these specifications, (3) link symbolically and associatively to extensive data sources, possibly remote, and (4) leverage development of simulation infrastructure in common between many different applications that may run in the same facility. A framework is, in effect, a contract in which users agree to meet the requirements of an interface, give up source control over the majority of the software required to execute their applications, and rely on resources beyond their immediate control in return for being able to function at a level of expertise in computer science and applied mathematics beyond their own, access a suite of modules that expands with time and experience and adapts to newly available hardware under a fixed or slowly evolving interface, and enter a large peer user community over which costs are amortized.

MATLAB,[98] with its various specialized toolkits, is often cited as a successful serial example of a computational framework. In it, users may request, for instance, the inverse action of a matrix on a right-hand side without concern for how the matrix is laid out in memory or any the properties such as sparsity, symmetry, definiteness, and the like that are relevant to selecting the best algorithm for it. The Common Component Architecture Forum[99] is dedicated to defining a minimal set of standard interfaces that a high-performance component framework has to provide to components, such as meshers or solvers, and can expect from them, in order to allow disparate components to be composed to build an application that runs on massively parallel hardware.

In practice, the high performance implementation of this vision is so far flawed and incomplete. The price of abstract interfaces with rigorously enforced contracts, if propagated all the way to interior kernels, is often poor performance. There is also the risk of the marketplace. Parallel languages, APIs, and frameworks emerge and disappear at a rate that makes it hard to convince a user to embrace a new paradigm. Software engineering issues go beyond the difficulties of transparent use of high performance machines. New user code must still be debugged and maintained, and keeping up with verification and validation through a codes lifetime is a difficult task. The task for an individual user is reduced through amortization of common components, but not eliminated. Most seriously lacking in existing frameworks are tools to measure performance. Currently available tools are mostly still in research stages, and possess unintuitive user interfaces with steep learning curves. A useful performance tool must be tunable to different degrees of monitoring so that production runs can offer basic usage statistics without severe compromise in performance, while developmental runs can access and pay for the full range of statistics. Parallel debuggers have not kept up with the massive expansion of parallel granularity in the last decade as the computer vendors with the best such software have either gone out of business or dropped support. Performance monitors and debuggers should be deliverable in principle, but the cost model has been to get hardware onto the floor as quickly as possible and to require users to span the software gap. In previous years, NASA's Advanced Supercomputing Division (NAS) took the lead in spearheading some development along these lines. Presently, much of their work is outsourced and their mission no longer includes such development, nor has it been picked up. The path to full-featured frameworks will be incremental. Though there is a long way to go, and a sustained commitment is required to produce confidence that leads to adoption, there have been encouraging steps in the direction of reusable components in the form of parallel toolkits of limited scope.[100, 101, 102, 103, 104, 105] The potential of software frameworks in computational aerodynamics has both evolutionary and revolutionary aspects. Evolving legacy codes into such a framework would preserve the valuable experience base that users have with a given code and interface and eases verified portability to future platforms. Libraries of some carefully chosen set of tools could also be provided, for example flux functions of various types, or for higher order methods, basis function quadrature kernels. This low-level approach follows the model of Eispack and Linpack, and application programmer interfaces (API) such as the message passing interface (MPI), which represent some of the most successful such library efforts. Other community-wide needs and concerns could be addressed through the development, standardization, and availability of generic but easily adopted and minimally intrusive capabilities for performing tasks such as unit testing, regression testing, and other verification and validation needs.

A longer term and revolutionary possibility is to build a multidisciplinary design optimization (MDO) code base of interchangeable modules, offering many options for each computational task, electable at job launch or even dynamically and adaptively during execution. The Community Climate Model community has pursued integration of this sort incrementally, first by linking disparate codes together by files in a workflow-like script, then by connecting them running simultaneously through a software coupler. Though its Fusion Simulation Program, the Department of Energy has funded several pairwise combinations of simulation codes that are experimenting with different frameworks. NASA should track the development of frameworks in such related, PDE-based disciplines to learn from mistakes and leverage successes, focusing its investment and ultimately adding modules specialized to its unique modeling requirements. The current MDO code base could be drawn upon to create modules that interface with a common infrastructure of meshers, solvers, visualizers, data miners and archivers, developed in concert with the balance of the computational science and engineering community.

F. Educational Issues

A critical need for a successful long term effort in computational engineering and high performance computing must include education. NASA can play a leading role in advocating and supporting education in computational engineering. Opportunities exist to improve education in computational engineering at both the undergraduate and graduate levels.

At the undergraduate level, standards do not exist across the United States for the desired knowledge of computational engineering. Many engineering programs do not require any classes in numerical methods, and it is often left as a subject to be “picked up” in an *ad hoc* way, rather than a subject worthy of systematic study. Training in computational engineering at the undergraduate level is critical for at least two reasons. First, due to its pervasiveness and continued growth, undergraduates that enter the workforce will encounter computational engineering. Thus, some background in the foundations of computational engineering is essential for the engineer of the future. Second, this exposure is critical to attract new graduate students to the area.

At the graduate level, NASA can play a variety of roles. Beyond research grants aimed at computational engineering, NASA could substantially improve its fellowship program, the Graduate Student Researchers Program (GSRP). A model which is highly effective at attracting the best graduate students is the DOE

Computational Science Graduate Fellowship (CSGF). The DOE CSGF provides four years of support including full tuition and a stipend of $31,200. Currently, the DOE program funds about 15 new students each year (about 60 students total in the pipeline at any time). By comparison, the GSRP program provides a $21,000 stipend but no tuition coverage. Both the DOE and NASA programs provide an opportunity for the fellow to spend 2-3 months at a lab or center. This year, NASA ARMD offered a new Aeronautics Fellowship to supplement the GSRP that provides an additional $10,000 per year for two years. Even accounting for this improvement, the DOE program is substantially more attractive.

In addition to financial support, tighter collaboration with academic faculty members and their students through visitor and summer programs has proven to be an effective means of generating student interest while accelerating research outcomes. For example, this year the Institute for Scientific Computing Research (ISCR) at Lawrence Livermore National Laboratory (LLNL) is hosting 67 summer students and has 8 ongoing collaborations with visiting faculty members, each of which is identified with an in-house LLNL principal investigator. NASA aeronautics actually led the way in this type of collaboration model through its quasi-academic institutes such as ICASE, RIACS and ICOMP, but these programs have been radically scaled back over the last decade as part of the general retreat from foundational algorithmic work within NASA aeronautics.

V. Concluding Remarks

Over the last decade, the fate of NASA Aeronautics has been one of declining budgets and questionable executive support despite repeated surveys and reports commissioned by organizations such as the NAE and others which have praised the contributions of this program to the field of aerospace engineering and its importance to the national well being. As NASA embarks on its technically challenging exploration program, it may have been forgotten that, as the only true engineering research component of the agency, NASA aeronautics has naturally been responsible for the development of the vast array of engineering tools, techniques and knowledge which have served all of NASA's other programs, and the entire engineering community, so well over the years.

Many of the studies and recommendations in support of NASA Aeronautics, including the recently Congressionally mandated thousand-page report produced by the National Institute of Aerospace (NIA) in 2005[106] focused almost exclusively on the impact of NASA aeronautics on the aerospace industry and the national air transportation system.

In this paper, we have implicitly taken a broader view of the importance of the NASA aeronautics program, as a national driver of engineering research, much in the same vein as the DOE Office of Science functions as a driver at the national level for issues in basic scientific research. This position has enabled the DOE and other agencies such as the NSF to capitalize on the recent renewed interest at the national level in high performance computing.[15] Furthermore, mission agencies such as the DOE and NASA bring a level of focus, relevance, and internal capabilities that complement the more academic research approach of other agencies such as the NSF, and their presence is essential in constituting a broad-based science and engineering research portfolio.

Programs such as the relatively small and science-focused SciDAC initiative of the Office of Science of the Department of Energy or the relatively large and technologically and educationally focused Cyberinfrastructure program of the National Science Foundation do not spring fully grown from the heads of Congressional budget officers, like Athena from Zeus. Their formal origins can be traced often to reports that originate from communities of experts, from the grassroots upwards. Such reports originate, in turn, from workshops that meet intensively for a few days or from commissions or panels that hold a series of public and private meetings for a period of a year, and invite experts to provide testimonies and whitepapers. They are not always successful on the first attempt.

The SciDAC program (www.scidac.gov), born in 2001, can be traced in its revolutionary philosophy of proposals in different disciplines (applications and enabling technologies in mathematics and computer science) with cross-linked deliverables, to a short 2000 report "Scientific Discovery through Advanced Computing" (www.pnl.gov/scales/docs/scidac_2000.pdf) by a single author. However, this report was preceded by a large national workshop on "Advanced Scientific Computing" in 1998 that recommended an earlier initiative of similar proportions pursuing the same opportunity, known as the "Scientific Simulation Initiative" (SSI). Furthermore, the early successes of the SciDAC program in fostering interdisciplinary collaborations motivated by diverse applications and powered by common enabling technologies, were captured in a 2003 community report from 315 individuals entitled "A Science-based Case for Large-scale Simulation" (SCaLeS, www.pnl.gov/scales). The first volume of SCaLeS was followed in 2004 with a companion volume that proposed out 26 related research areas for increased support in more detail. The SCaLeS report was not immediately successful in increasing support for SciDAC or large-scale simulation generally, but the interdisciplinary agendas begun while generating the SCaLeS report have resulted in numerous spinoff workshops and reports in applications domains, such as fission, fusion, materials science, etc., that have led to increased visibility and support for simulation in particular domains, and SciDAC has been renewed for a second five years, as of 2006.

Similarly, the Geosciences community, largely sponsored by the NSF, has formed the "Ad Hoc Committee and Technical Working Group for a Petascale Collaboratoty for the Geosciences". Through a series of workshops, this group has produced a two volume report on the technical and societal needs and financial requirements for a broad collaboration among geoscientists centered around petascale computing, including the advocacy of a new petascale computational facility.[10,11]

The NSF Cyberinfrastructure report of 2003 (http://www.nsf.gov/od/oci/reports/toc.jsp) was a lengthy project that included 62 testimonials, 700 responses to a community survey, and many layers of review and criticism. It was carried to completion by a panel of nine. Though long in gestation, it led to the creation of an entirely new cross-cutting administrative entity at the National Science Foundation, the Office of Cyberinfrastructure (OCI), headed initially by the chair of the panel that wrote the report. The research and training opportunities available under OCI are just now beginning to become apparent, and are facing considerable excitement.

Persistence is necessary to create community consensus and to employ that consensus to pressure executive branch agencies to propose (and ultimately Congress to appropriate on behalf of) new research programs with a central role for simulation. NASA has a set of science and engineering missions as exciting and diverse as those in DOE's Office of Science or NSF's Office of Cyberinfrastructure. It also has a set of university collaborators as diverse geographically and topically as those in the SciDAC program. Unlike some of the hidden or obscure features of the SciDAC scientific mission that are difficult to explain, NASA's aerodynamics mission is front and center to many Americans, and enthusiasm for this mission runs broad and deep in the American public. It should be possible to create NASA-oriented simulation programs of similar vigor.

VI. Acknowledgements

The authors would like to acknowledge Marsha Berger, from the Courant Institute, NYU, New York NY, for her contributions to this paper through suggestions and editing of the manuscript. Special thanks are also due to the various referenced authors who supplied supporting figures for this paper.

VII. Appendix I: Lessons from the Gordon Bell Prize History

The Gordon Bell Prizes offer an interesting perspective on the history of computational simulation at the high end, a history in which NASA aeronautics is a notable pioneer in one key respect, but has otherwise been shut out. Awarded most years since 1988 in multiple categories for notable progress on real applications, the prizes have attracted the attention of leading computational researchers through nearly two decades of evolving architectures. Winners must document performance issues well beyond the details found in other computational science and engineering papers, which emphasize instead the results of the simulation. Many have argued that the orientation of the Gordon Bell prizes to computational rate distorts appreciation of progress in hierarchical and adaptive algorithms on geometrically fitted grids, since such applications are capable of much more accurate solutions per flop, but appear weak in flop/s rate compared to static structured or cartesian-grid methods with simple explicit time-stepping schemes. This deficiency has been partially addressed since 1999 with a version of the Prize that NASA had a role in inventing.

The peak performance prize has been awarded 18 times from 1988 through 2006, during which time the peak performance of the winning entry has improved by more than five orders of magnitude from 1 Gflop/s to over 200 Tflop/s. PDE-based formulations have claimed the prize 8 times, N-body formulations 4 times, molecular dynamics 3 times, with the remaining prizes going to Monte Carlo and integral equation formulations. Among applications described by PDEs, computational fluid dynamics applications took the prize in 1996 on the Japanese Numerical Wind Tunnel using 160 vector processors, in 1999 on ASCI Blue Pacific using 5,832 Power microprocessors, and in 2004 on the Japanese Earth Simulation using 4,096 vector processors. The vector-based winners employed Cartesian index-space grids. Over the 18-year window of the peak performance prize, Moore's Law accounts for at most an improvement factor of at most about 4,096 (12 doubling periods); in fact, computational rate improves only about half as fast as the density of transistors on a chip, which would therefore account only for an improvement factor of about 64 over 18 years (6 doubling periods for per-chip flop/s performance). The flop/s improvement ratio of 200,000 obtained on real applications comes dominantly from concurrency, with the latest two molecular dynamics winners successfully employing 131,072 processors of the IBM Blue Gene/L, paying assiduous attention to load-balancing at each phase of the computation.

Price-per-performance on real applications is measured in another Gordon Bell Prize, which saw improvements of four orders of magnitude between 1989 and 2001, but has stagnated since there, for general-purpose computer systems, while FPGAs, GPUs, and other special-purpose hardware has continued down this price-per-performance curve. Non-aerodynamics computational fluid dynamics applications on custom clusters claimed the price-per-performance prize in 1995 and 2000.

The Bell special achievement prize has been awarded most years since 1999, when it was created to recognize an unstructured-grid NASA external aerodynamics application that was ported to the 1 Tflop/s capable ASCI Red machine, and executed at 0.23 Tflop/s on 3,072 Intel dual-processor nodes.[107,108] Scalability is only part of the challenge for codes based on finite discretizations (differences, elements, or volumes), as memory bandwidth has lagged processor clock rates and is now approximately two orders of magnitude behind. This implies that bandwidth-limited codes such as most PDE applications will execute at approximately 1% of achievable performance unless structured for efficient reuse of cached operands. This can be done by optimizing compilers for data accessed by Cartesian indices, but generally needs to be done by and in unstructured problems. All of the recent Gordon Bell prizes have been claimed by teams in which computer scientists familiar with the hardware have played a prominent role.

NASA need not reinvent the developments in enabling technologies and infrastructure on display in the singular achievements of projects such as those that have claimed Gordon Bell Prizes. It can quickly absorb the best research program models, software, and infrastructure, and further focus them on its unique missions. High-end computation in support of the multidisciplinary aeronautics mission has driven and can again drive the national simulation agenda.

References

[1] D. L. Slotnick. The fastest computer. *Scientific American*, February 1971.

[2] T. H. Pulliam, K. Kutler, and V. Rossow. Harvard Lomax: His quiet legacy to CFD. June 1999. Paper presented at the 14 AIAA Fluid Dynamics Conference, Norfolk VA.

[3] C. Bishof and H. D. Simon. The implementation of the US High Performace Computing and Communication Program. 1992.

[4] F. R. Bailey. High end computing challenges in aerospace design and engineering. July 2004. Paper presented at the 3rd Int. Conf. on CFD, Toronto, Canada.

[5] Top 500 supercomputer sites. http://www.top500.org.

[6] NITRD supplement to the President's budget. *Report by the Subcommittee on Networking and Information Technology Research and Development, Committee on Technology, National Science and Technology Council, Office of Science and Technology Policy*, February 2005.

[7] Computational Science: Ensuring America's Competitiveness. *Report to the President, President's Information Technology Advisory Committee*, June 2005.

[8] Simulation Based Engineering Science: Revolutionizing engineering science through simulation. *NSF Report*, May 2006. Report of the NSF Blue Ribbonn Panel on Simulation Based Engineering Science.

[9] Scientific Discovery through Advanced Computing. http://www.scidac.gov.

[10] Ad Hoc Committee and Technical Working Group for a Petascale Collaboratory for the Geosciences. Establishing a petascale collaboratory for the geosciences:scientific frontiers. *Report to the Geosciences Community, UCAR/JOS. 80pp*, June 2005.

[11] Ad Hoc Committee and Technical Working Group for a Petascale Collaboratory for th e Geosciences. Establishing a petascale collaboratory for the geosciences:technical and budgetary prospetus. *Report to the Geosciences Community, UCAR/JOS. 56pp*, October 2005.

[12] A Science-Based Case for Large-Scale simulation, Volume I. *Office of Science, US Department of Energy*, July 2003. 76pp.

[13] A Science-Based Case for Large-Scale simulation, Volume II. *Office of Science, US Department of Energy*, September 2004. 254pp.

[14] P. Moin and J. Kim. Tackling turbulence wth supercomputers. *Scientific American*, January 1997.

[15] American competitiveness initiative. *Domestic Policy Council, Office of Science and Technology Policy*, February 2006.

[16] L. Porter. Congressional Testimony to the House Science Committee. September 2006.

[17] Decadal Survey of Civil Aeronautics. 2006. National Academies Press.

[18] M. D. Salas. Digital flight: The last CFD aeronautical grand challenge. *Journal of Scientific Computing*, 28(2-3):479–505, September 2006.

[19] S. M. Murman and M. J. Aftosmis. Dynamic analysis of atmospheric-entry probes and capsules. AIAA Paper 2007-0074, presented at the 45th AIAA ASM Meeting, Reno NV., January 2007.

[20] L. Green and A. Spence. Applications of computational methods for dynamic stability and control derivatives. AIAA Paper 2004-0377, presented at the 42nd AIAA ASM Meeting, Reno, NV, January 2004.

[21] A. G. Godfrey and E. M. Cliff. Direct calculation of aerodynamic force derivatives - A sensitivity-equation approach. AIAA Paper 1998-0393, January 1998.

[22] P. Crisafulli, M. Kaufman, A. A. Giunta, W. H. Mason, B. Grossman, L. T. Watson, and R. T. Haftka. Response surface approximations for pitching moment including pitch-up in the MDO design of an HSCT. AIAA Paper 1996-4136, January 1996.

[23] A. Schütte, G. Einarsson, A. Raichle, B. Schöning, M. Orlt, J. Neumann, J. Arnold, W. Mönnich, and T. Forkert. Numerical simulation of maneuvering aircraft by aerodynamic, flight mechanics and structural mechanics coupling. AIAA Paper 2007-1070, presented at the 45th AIAA ASM Meeting, Reno NV., January 2007.

[24] S. Görtz, D. McDaniel, and S. Morton. Towards an efficient aircraft stability and control analysis capability using high-fidelity CFD. AIAA Paper 2007-1053, presented at the 45th AIAA ASM Meeting, Reno NV., January 2007.

[25] J. P. Dean, S. A. Morton, D. R. McDaniel, and S. Görtz. Efficient high resolution modeling of fighter aircraft with stores for stability and control clearance. AIAA Paper 2007-1652, presented at USAF T& E Days, Destin FL, February 2007.

[26] COMSAC: Computational methods for stability and control. NASA/CP-2004-213028, Parts 1 and 2., eds. R. Hall, C. M. Fremaux, and J. R. Chambers., April 2004.

[27] K. Willcox. Model reduction for large-scale applications in computational fluid dynamics. Real-Time PDE-Constrained Optimization, L. Biegler, O. Ghattas, M. Heinkenschloss, D. Keyes, and B. van Bloemen Waanders, eds. SIAM Philadelphia, February 2007.

[28] S. H. Woodson, B. E. Green, J. J. Chung, D. V. Grove, P. C. Parikh, and J. R. Forsythe. Understanding abrupt wing stall with computational fluid dynamics. *AIAA Journal*, 42(3):578–585, 2005.

[29] S. M. Murman, M. J. Aftosmis, and M. Nemec. Automated parameter studies using a Cartesian method. AIAA Paper 2004-5076, presented at the 22nd AIAA Applied Aerodynamics Conference, Providence, RI, August 2004.

[30] D. J. Mavriplis, M. Aftosmis, and M. Berger. High-resolution aerospace applications using the NASA Columbia supercomputer. Paper presented at the 2005 Supercomputing Conference, Seattle, WA., November 2005.

[31] N. M. Chaderjian, S. E. Rogers, M. J. Aftosmis, S A Pandya, , J. Ahmad, and E. Tejnil. Automated Euler and Navier-Stokes database generation for a glide-back booster. Proceedings of the 3rd International Conference on Computational Fluid Dynamics, Toronto, Canada, Springer-Verlag, July 2004.

[32] S. E. Rogers, M. J. Aftosmis, S A Pandya, N. M. Chaderjian, E. Tejnil, and J. Ahmad. Automated CFD parameter studies on distributed parallel computers. AIAA Paper 2003-4229 , presented at the 16th AIAA Computational Fluid Dynamics Conference, June 2003.

[33] B. L. Stevens and F. L. Lewis. *Aircraft Control and Simulation, 2nd edition.* John Wiley and Sons, New Jersey., 2003.

[34] M. Turner, R. Ryder, A. Norris, M. Celestina, J. Moder, N. Lui, J. Adamczyk, and J. Veres. High Fidelity 3D Turbofan Engine Simulation with Emphasis on Turbomachinery-Combustor Coupling. July 2002. AIAA 2002-3769.

[35] M. Turner and A. Norris J. Veres. High Fidelity 3D Simulation of the GE90. Orlando, FL, June 2003. AIAA 2003-3996.

[36] M. G. Turner, J. A. Reed, R. Ryder, and J. P. Veres. Multi-Fidelity Simulation of a Turbofan Engine with Results Zoomed into Mini-Maps for a Zero-D Cycle Simulation. Vienna, Austria, June 2004. ASME Paper No. GT2004-53956.

[37] J. U. Schlüter, X. Wuy, E. Weide, S. Hahn, J. J. Alonso, and H. Pitsch. Integrated LES- RANS of an Entire High-Spool of a Gas Turbine. Reno NV, Jan. 2006. AIAA Paper 2006-897.

[38] R. Claus, S. Townsend, D. Carney, J. Horowitz, and M. Turner. A Case Study of High Fidelity Engine System Simulation. In *42nd AIAA/ASME/SAE/ASEE Joint Propulsion Conference*, Sacramento, CA, July 2006. AIAA 2006-4971.

[39] S. Gorrell, T. F. Yao, and J. Vickery. Computational Science and Engineering Advances Understanding of Complex Unsteady Flows in High Performance Fans and Compressors. In *Proceedings of the 2006 Users Group Conference*, Denver, CO, June 26-29 2006. DoD High Performance Computing Modernization Program.

[40] J. P. Chen and W. R. Briley. A Parallel Flow Solver for Unsteady Multiple Blade Row Turbomachinery Simulations. In *ASME Turbo Expo 2001*, 2001. ASME Paper no. 2001-GT-0348.

[41] R. F. Manning and M. E. Taslim. Crossover Cooled Airfoil Trailing Edge. US Patent No. 6,607,356 B2, Aug 19, 2003.

[42] R. S. Abhari. Unsteady Fluid Dynamics of Turbines: Perspective on Possible Directions to Improve Future Engine Designs. In J. E. LaGraff, D. E. Ashpis, L. G. Martin, Oldfield, and J. P. Gostelow, editors, *Proceedings of Minnowbrook V 2006 Workshop on Unsteady Flows in Turbomachinery*, volume NASA/CP 2006-214484, December 2006.

[43] C. Lykins, D. Thomson, and C. Pomfret. The Air Force's application of probabilistics to gas turbine engines. AIAA-94-1440-CP, AIAA, 1994.

[44] F. M. Safie. Use of probabilistic design methods for NASA applications. *Reliability Technology*, 28:17–24, 1992.

[45] C. C. Chamis. Probabilistic structural analysis methods for space propulsion system components. *Probabilistic Engineering Mechanics*, 2(2):100–10, 1987.

[46] H. R. Millwater and Y.-T. Wu. Global/local methods for probabilistic structural analysis. AIAA Paper 93-1378-CP, 1993.

[47] R. S. Ryan and J. S. Townsend. Application of probabilistic analysis/design methods in space programs: The approaches, the status, and the needs. AIAA Paper 93-1381, 1993.

[48] D. E. Thomson and T. G. Fecke. Air force perspective on probabilistic design. *Reliability Technology*, 28:25–32, 1994.

[49] J. N. Yang, S. D. Manning, J. L. Rudd, and M. E. Artley. Probabilistic durability analysis methods for metallic airframes. *Probabilistic Engineering Mechanics*, 2(1):9–15, 1987.

[50] Donald L. Kunz. Comprehensive rotorcraft analysis: Past, present, and future. AIAA Paper 2005-2244, 2005.

[51] R.C. Strawn, F.X. Caradonna, and E.P.N. Duque. Thirty years of rotorcraft computational fluid dynamics research and development. *Journal of the American Helicopter Society*, 51(1), 2006.

[52] Andreas Griewank. Achieving logarithmic growth of temporal and spatial complexity in reverse automatic differentiation. *Optimization Methods and Software*, 1:35–54, 1992.

[53] Andreas Griewank and Andrea Walther. Algorithm 799: Revolve: an implementation of checkpointing for the reverse or adjoint mode of computational differentiation. *ACM Transactions on Mathematical Software*, 26(1):19–45, March 2000.

[54] R. M. Errico. What is an adjoint model ? *Bulletin of the American Meteorological Society*, 8(11):2577–2591, 1997.

[55] K. Ghayour and O. Baysal. Limit-cycle shape optimization using time-dependent transonic equation. AIAA Paper 1999-3375, 1999.

[56] S. Nadarajah and A. Jameson. Optimum shape design for unsteady three-dimensional viscous flows using a non-linear frequency domain method. AIAA Paper 2006-3455, 2006.

[57] K. Mani and D. J. Mavriplis. An unsteady discrete adjoint formulation for two-dimensional flow problems with deforming meshes. AIAA Paper 2007-0060, presented at the 45th AIAA Aerospace Sciences Meeting and Exhibit, Reno NV., January 2007.

[58] R. Becker and R. Rannacher. An optimal control approach to a posteriori error estimation in finite element methods. In A. Iserles, editor, *Acta Numerica*. Cambridge University Press, 2001.

[59] Ralf Hartmann and Paul Houston. Adaptive discontinuous Galerkin finite element methods for the compressible Euler equations. *Journal of Computational Physics*, 183(2):508–532, 2002.

[60] D. A. Venditti and D. L. Darmofal. Anisotropic grid adaptation for functional outputs: application to two-dimensional viscous flows. *Journal of Computational Physics*, 187(1):22–46, 2003.

[61] K. Fidkowski and D. Darmofal. An adaptive simplex cut-cell method for discontinuous Galerkin discretizations of the Navier-Stokes equations. AIAA Paper 2007-3941, 2007.

[62] K. Mani and D. J. Mavriplis. Discrete adjoint based time- step adaptation and error reduction in unsteady flow problems. AIAA Paper 2007-3944, presented at the 18th AIAA Computational Fluid Dynamics Conference, Miami FL., June 2007.

[63] M. Potsdam, H. Yeo, and W. Johnson. Rotor airloads prediction using loose aerodynamic/structural coupling. *AIAA Journal of Aircraft*, 43(3):732–742, 2006.

[64] A. Dimanlig, E. Meadowcroft, R. Strawn, and M. Potsdam. Computational modeling of the CH-47 helicopter in hover. American Helicopter Society 63nd Annual Forum, Virginia Beach, VA, May 2007.

[65] T. Renaud, D. O'Brien, M. Smith, and M. Potsdam. Evaluation of isolated fuselage and rotor-fuselage interaction using CFD. *Journal of the American Helicopter Society*, 2007 (to appear).

[66] D. O'Brien. *Analysis of Computational Modeling Techniques for Complete Rotorcraft Configurations*. PhD thesis, Georgia Institute of Technology, 2006.

[67] J. D. Cyrus. Engine component life prediction methodology for conceptual design investigations. ASME Paper 86-GT-24.

[68] I.Y. Tumer and A. Bajwa. Learning about how aircraft engines work and fail. AIAA Paper AIAA-99-2850.

[69] M.I. Wood. Gas turbine hot section components: The challenge of 'residual life' assessment. In *Proceedings of the I MECH E: Journal of Power and Energy, Volume 214 Part A, pp.193-201*.

[70] V. Sidwell and D. Darmofal. The impact of blade-to-blade variability and assembly on turbine cooling performance. *ASME Journal of Turbomachinery*, 127(4):763–770, 2005.

[71] T. Hildebrandt, J. Ettrich, M. Kluge, M. Swoboda, A. Keskin, F. Haselbach, and H. Schiffer. *Unsteady Aerodynamics, Aeroacoustics and Aeroelasticity of Turbomachines*, chapter Unsteady 3D Navier-Stokes Calculation of a Film-cooled Turbine Stage with Discrete Cooling Hole. Springer, Netherlands, 2006.

[72] A. Burdet and R. Abhari. Three-dimensional flow prediction and improvement of holes arrangement of a film-cooled turbine blade using a feature-based jet model. *ASME Journal of Turbomachinery*, 129(2):258–268, 2007.

[73] Rupak Biswas, M. Jahed Djomehri, Robert Hood, Haoqiang Jin, Cetin Kiris, and Subhash Saini. An application-based performance characterization of the Columbia Supercluster. Paper presented at the 2005 Supercomputing Conference, Seattle, WA., November 2005.

[74] R. Biswas, M. J. Aftosmis, C. Kiris, and B.-W. Shen. Petascale computing: Impact on future NASA missions. In Petascale Computing: Architectures and Algorithms (D. Bader, ed.), Chapman and Hall / CRC Press, (to appear)., 2007.

[75] Z. Lin, L. Chen, I. Holod, Y. Nishimura, H. Qu, S. Ethier, G. Rewoldt, W. X. Wang, Y. Chen, J. Kohut, S. Parker, and S. Klasky. Electron transport driven by short wavelength trapped electron mode turbulence. Paper IAEA-CN-138/TH/P2-8 Presented at Fusion Energy 2006, Proceedings of the 21st IAEA Conference, Chengdu China, 2006.

[76] W. W. Lee, S. Ethier, T. G. Jenkins, W. X. Wang, J. L. V. Lewandowski, G. Rewoldt, W. M. Tang, S. E. Parker, Y. Chen, and Z. Lin. Long time simulations of microturbulence in fusion plasmas. Paper IAEA-CN-138/TH/2-6Rb, Presented at Fusion Energy 2006, Proceedings of the 21st IAEA Conference, Chengdu China, 2006.

[77] F. H. Streitz, J. N. Glosli, M. V. Patel, B. Chan, R. K. Yates, B. R. de Supinski, J. Sexton, and J. A. Gunnels. 100+ TFlop Solidification Simulations on BlueGene/L. Proceedings of Supercomputing 2005 Conference, Seattle WA, ACM, Los Alamitos, November 2005.

[78] W. F. Spotz and M. A. Taylor. A massively parallel scalable atmospheric model. Sandia National Laboratory Report SAND2006-7626, Albuquerque, NM, December 2006.

[79] J. Dennis, A. Fournier, W. F. Spotz, A. St.-Cyr, M. A. Taylor, S. Thomas, and H. Tufo. High resolution mesh convergence properties and parallel efficiency of a spectral element atmospheric dynamical core. *Int. J. High Perf. Computing Appl.*, 19:225–235, 2005.

[80] D. R. Chapman. Computational aerodynamics development and outlook. *AIAA Journal*, 17(12):1293–1313, 1979.

[81] P. R. Spalart, W.-H. Jou, M. Strelets, and S. R. Allmaras. Comments on the feasibility of LES for wings and on a hybrid RANS/LES approach. Proceedings of the First AFOSR International Conference on DNS and LES, Louisiana Tech University Ruston LA., August 1997.

[82] D. W. Levy, T. Zickuhr, J. Vassberg, S. Agrawal, R. A. Wahls, S. Pirzadeh, and M. J. Hemsch. Data summary from the first AIAA computational fluid dynamics drag prediction workshop. *Journal of Aircraft*, 40(5):875–882, 2003.

[83] K. Laflin, S. M. Klausmeyer, T. Zickuhr, J. C. Vassberg, R. A. Wahls, J. H. Morrison, O. P. Brodersen, M. E. Rakowitz, E. N. Tinoco, and J. Godard. Data summary from second AIAA computational fluid dynamics drag prediction workshop. *Journal of Aircraft*, 42(5):1165–1178, 2005.

[84] J. C. Vassberg, E. N. Tinoco, M. Mani, O. P. Brodersen, B. Eisfeld, R. A. Wahls, J. H. Morrison, T. Zickuhr, K. R. Laflin, and D. J. Mavriplis. Summary of the third AIAA CFD drag prediction workshop. AIAA Paper 2007-0260, January 2007.

[85] R. Cunnington, T. Takahashi, and L. Bays. Incorporation of aerothermodynamic analysis tools into the multi-disciplinary conceptual design process. AIAA Paper 2007-0406, January 2007.

[86] W. S. Humphrey. Why big software projects fail: The 12 key questions. 2004.

[87] See faler.wordpress.com/2007/04/24/the-diminishing-return-of-adding-personnel-to-a-task/.

[88] G. Moore. Cramming more components onto integrated circuits. *Electronics Magazine*, 38(8):114–117, 1965.

[89] Grand challenges: High performance computing and communications. Supplement to the President's Fiscal Year 1992 Budget, OSTP, Committee on Physical, Mathematics, and Engineering Sciences, Federal Coordinating Council for Science, Engineering, and Technology, 1992.

[90] J. L. Lumley. The structures of inhomogeneous turbulent flow. Atmospheric Turbulence and Radio Wave Propagation, A. Yaglom and V. Tatarski, eds., Nauka, Moscow, 1967.

[91] W. D. Bachalo and A. D. Johnson. An investigation of transonic turbulent boundary layer separation generated on an axisymmetric flow model. AIAA Paper 1979-1479, 1979.

[92] W. O. Valarezo, C. J. Dominick, and R. J. McGhee. Multielement airfoil performance due to Reynolds and Mach number variations. *AIAA Journal of Aircraft*, 30(5):689–694, 1993.

[93] F. W. Spaid and F. T. Lynch. High-Reynolds number, multi-element airfoil flowfield measurements. AIAA Paper 1996-0682, January 1996.

[94] M. Chaffin and S. Pirzadeh. Unstructured navier-stokes high-lift computations on a trapezoidal wing. AIAA Paper 2005-5084, June 2005.

[95] L. P. Yip, P. M. Vijgen, J. D. Hardin, and C. P. van Dam. In-flight pressure measurements on a subsonic transport high-lift wing section. *AIAA Journal of Aircraft*, 32(3):529–538, 1995.

[96] J. Lamar and C. Obara. Review of cranked-arrow wing aerodynamics project: Its international aeronautical community role. AIAA Paper 2007-0487, January 2007.

[97] A. Rizzi, O. Boelens, A. Jirasek, and K. Badcock. What was learned from numerical simulations of F-16XL (CAWAPI) at flight conditions. AIAA Paper 2007-0683, January 2007.

[98] The mathworks – MATLAB and Simulink for technical computing. www.mathworks.com.

[99] Common Component Architecture Forum. See www.cca-forum.org/.

[100] PETSc. Portable Extensible Toolkit for Scientific Computation. See www.mcs.anl.gov/petsc.

[101] *hypre*. High Performance Preconditioners. See www.llnl.gov/CASC/hypre.

[102] Prometheus. Prometheus. See www.columbia.edu/~ma2325/ug.html.

[103] SUNDIALS. Suite of Nonlinear Differential and Algebraic Equation Solvers. See www.llnl.gov/casc/sundials.

[104] SuperLU. Sparse Direct Solver. See crd.lbl.gov/~xiaoye/SuperLU.

[105] Trilinos. See software.sandia.gov/trilinos.

[106] Responding to the Call: Aviation Plan for American Leadership. Congressionally requested report produced unde the auspices of the National Institute of Aerospace, Hampton VA. Available at http://www.nianet.org/pubs/AviationPlan.php, April 2005.

[107] W. K. Anderson, W. D. Gropp, D. K. Kaushik, D. E. Keyes, and B. F. Smith. Achieving high sustained performance in an unstructured mesh CFD application. In *Bell Prize award paper, Special Category, in the Proceedings of SC'99, IEEE, Los Alamitos, CA.*, 1999.

[108] W. D. Gropp, D. K. Kaushik, D. E. Keyes, and B. F. Smith. High performance parallel implicit CFD. volume 27, pages 337–362, 2001.