COVER SHEET

NOTE: This conversheet is intended for you to list your article title and author(s) name only—this page will not appear on the CD-ROM.

Title: *High Performance Computational Engineering: Putting the 'E' Back in 'CSE'* for Proceedings of the 21st International Conference on Parallel Computational Fluid Dynamics (ParCFD 2009)

Authors: Dimitri J. Mavriplis

PAPER DEADLINE: ******SEPTEMBER 30, 2009*****

PAPER LENGTH: ****16 PAGES (Maximum)****

ABSTRACT

Over the last decade, 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. On the other hand, the capabilities of computational engineering problems have become increasingly stagnant as efforts have tended to concentrate on reducing simulation cost rather than increasing simulation capability. Computational engineering problems differ in many aspects from discovery-based computational science problems, and specific techniques for addressing these differences must be developed in order for engineering problems to benefit from the upcoming paradigm shift to massively parallel multicore architectures. Arguing that computational fluid dynamics in general, and computational aerodynamics in particular, represent some of the most advanced computational engineering applications, we identify specific barriers impeding the deployment of these problems to current and future HPC architectures and suggest several strategies for overcoming these impediments. The long term objective is to enable the development of significantly more capable production level computational fluid dynamic engineering tools by harnessing the power of the upcoming generation of massively parallel multicore architectures.

INTRODUCTION

Today, computational science and engineering (CSE) stands at a cross-roads, similar in some ways to the situation present twenty years ago at the initiation of the ParCFD conference series [1]. In 1989, CSE was on the verge of a paradigm shift in the transition from small numbers of high performance vector processors, to "massively" parallel configurations of microprocessors using on the order of hundreds of cpus [2]. Today, although Moore's law still holds, advanced computational capabilities are increasingly being sought through rapid increases in the number of processor cores, rather than clock speeds, leading to an explosion in the degree of parallelism which must be harnessed in order to make effective use of novel hardware. However, in many ways the CSE landscape is also substantially different than it was twenty years ago, when the leading high-performance computing drivers covered a

Dimitri Mavriplis, Professor, Department of Mechanical Engineering
University of Wyoming, Laramie, WY 82071, U.S.A.

broad area of science and engineering applications. Today, the advocacy for high performance computing (HPC) is largely driven 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 [3, 4, 5]. Unfortunately, this argument does not hold for engineering applications, since engineering is not discovery driven. Engineering problems can differ substantially from science applications, and generally focus on their own set of design-driven objectives, which incur a different set of barriers to effective use of HPC architectures. Unfortunately, the last ten years has seen a drop off in the development and deployment of new computational engineering techniques. During this period, industrial efforts have turned largely to reducing the cost of fixed simulation capabilities through migration to lower-cost hardware, with a general lack of interest in participating in the development of new computational techniques which will be needed to harness the full capabilities of the next generation of HPC hardware. The case for increased investments in engineering-based simulation has only been addressed in a recent NSF report [6]. Focusing on the need to invest in advanced HPC development specific to engineering problems is important, since the current situation may lead to future engineering-based simulation software which is incapable of harnessing the full potential of new HPC hardware, and which will therefore be limited to use only with low-end commodity hardware.

Although computational fluid dynamics (CFD) only overlaps partially with the more general field of computational engineering, a strong argument can be made why CFD in general, and computational aeronautics in particular, should play a leading role in driving research and investment in new computational engineering developments. Aeronautics represents one of the most technologically advanced engineering fields, and international advances in aeronautics have benefited from a particularly effective collaboration and combination between government, industrial, and academic research which is seldom duplicated in other engineering fields. Aeronautics has traditionally been at the forefront of HPC developments, and is thus naturally well positioned to resume its role as a principal driver of advances in high performance computing.

Ten years ago, this author delivered a plenary talk at the 1999 ParCFD conference [7], describing what at the time was a leading-edge unstructured mesh aerodynamic calculation using 35 million grid points running on 1500 processors on a CRAY T3E. This achievement could be seen as having successfully addressed the challenge posed in 1989 of migrating CFD simulations from vector processors to "massively" parallel computer architectures. The question before us today is whether the engineering community will be able to continue these advances by demonstrating radical new simulation capabilities running on the massively parallel multicore architectures involving 100,000 to 1 million cores projected to come on-line over the next decade.

One of the impediments to continual advances in computational engineering capabilities is a persisting notion that computational engineering problems are not complex enough to warrant the use of computational resources on a vast scale comparable to discovery-driven science problems. This must be countered by identifying the potential new engineering frontiers to be opened by the availability of substantially more capable computational hardware, and by describing specific enabling objectives which new engineering simulations must target. In the next section, we describe various computational aerodynamic engineering problems which are currently considered intractable, but which could revolutionize the practice of aeronautical engineering if brought to fruition. In the subsequent sections, we identify some of the barriers to achieving these capabilities and propose various strategies for addressing these difficulties.

TARGET ENGINEERING CFD CAPABILITIES

Higher Fidelity

The field of aeronautics has traditionally been a strong driver in the continual development and adoption of higher fidelity computational engineering tools 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. However, over the last decade, the high-fidelity method of choice seems to have stalled out with RANS methods, and efforts have often turned to reducing the cost of this fixed capability rather than pursuing even higher-fidelity tools. The obvious path towards higher fidelity involves the development and adoption of Detached Eddy (DES) or Large Eddy (LES) simulation methods in a production environment [8, 9]. The increased costs of these methods stem not only from the need to employ much higher spatial resolution for capturing all relevant turbulence structures down to the inertial range scales, but also from the necessary inclusion of the temporal dimension in these inherently unsteady problems. Therefore, successful deployment of these methods on future hardware will require much higher scalability on much larger processor or core counts in order to provide practical turnaround time in a production setting.

Flight Envelope Simulation

Characterization of the complete flight envelope of a new aerospace vehicle is an important aerospace engineering design objective which can be met increasingly with more sophisticated computational tools. 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. An example of work in this direction is provided in References [10, 11, 12, 13]. Figure 1 provides an illustration of a flight-envelope data-base generation using the CART3D simulation code for a hypothetical liquid glide-back booster. The mesh used for the inviscid cartesian approach 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. Figure 1 depicts the computed aerodynamic data-base, as a carpet plot of the pitching moment coefficient 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 [12]. The long term objective must be to incorporate a wider range of parameters and variations of each parameter in these simulations, while at the same time raising the fidelity of the simulation to incorporate RANS and eventually LES techniques. Given an optimistic estimated turnaround time of one hour for a single point RANS solution on current-day computational hardware, and an estimate of O(10^6) individual simulations to be performed (covering 10 instances of 6 parameters each), 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.

Digital Flight

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. Figure 2 provides an example of a comprehensive digital flight simulation capability under development in the HIARMS project sponsored by the US Army Aeroflightdynamics Directorate [14, 15]. In this project, an unstructured mesh approach is used near the body to capture the geometric complexity, while a higher-order adaptive cartesian solver is used away from the body to adequately capture the vortices and wake effects. These codes are further linked through a Python interface to other modules which simulate structural analysis and flight control. The goal is to enable the multidisciplinary simulation of maneuvering rotorcraft configurations on massively parallel machines with higher accuracy and throughput compared to previous efforts. The HIARMS project falls under the larger CREATE-AV effort established in the US by the Department of Defense for building engineering software tools based on high fidelity computational simulations which can make effective use of massively parallel hardware for reducing risk in the weapons acquisition process [16, 17]. However, these efforts are geared towards using existing solver and component technologies on current-day high-end architectures involving O(1000) processors and are not directly addressing the advent of even more radically large scale parallel multicore architectures.

Uncertainty Quantification

As engineering simulation tools mature and more complex simulation problems are attempted, the quality and usefulness of the simulation results depends more and more on our ability to quantify the uncertainties in the computed simulation outputs. Thus, uncertainty quantification has begun to emerge as an important area in modern computational engineering. Today, it is no longer sufficient to predict objectives related to vehicle performance using a specific physical model with deterministic inputs. Rather, a probability distribution function of the simulation objectives is required as a function of the uncertainties inherent in the simulation input parameters, in order to establish confidence levels over a range of performance predictions.

Consider, for example, the uncertainties caused by geometric variances such as manufacturing tolerances. Sidwell and Darmofal have analyzed cooled turbine blades and shown that a leading cause of decreased blade life is cooling flow variability due to manufacturing tolerances [18]. 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% [18].

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 to this prediction. A Monte-Carlo approach to modeling the variability consists of performing 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 are necessary. Parallel implementation of probabilistic methods (such as Monte Carlo) is often trivial and in this sense represents a natural target of opportunity for future HPC applications. However, at the same time the growth in the number of simulations coupled with the drive to higher-fidelity simulations calls for the development of more sophisticated uncertainty quantification methods.

Another source of uncertainty may be caused by empirical model inputs to the simulation. As an example, consider the uncertainty inherent in numerical simulations of aero-heating effects for the design of thermal protection systems for hypersonic vehicles. Because many of the parameters used in the real gas models for high speed flows have been determined empirically, any aero-heating uncertainty quantification must include the effect of these model parameter uncertainties. This type of uncertainty analysis has been performed in reference [19], using a Monte Carlo technique where the real gas model parameters were varied based on assigned or estimated probability distribution functions, resulting in the requirement of performing several thousand high-fidelity simulations in order to obtain a suitable description of the aero-heating uncertainty distribution.

In practice, uncertainties in numerical simulations arise from multiple additional sources including numerical error sources due to spatial and temporal discretization error, algebraic errors due to incomplete numerical convergence, and inter-disciplinary coupling errors for multidisciplinary simulations. Developing techniques which take advantage of increased hardware capabilities for assessing and controlling all these types of uncertainties will become increasingly important in future applications.

Other Areas

In a previous survey publication [20] we have outlined various other applications and objectives which could be made feasible with increased computational capabilities. For example, the simulation of a complete turbofan engine using three-dimensional unsteady CFD would allow a virtual engine testing capability, through which a better understanding of engine component interactions could be obtained, and which could be made to rival the accuracy of physical testing, through the development of better algorithms and models which make effective use of advanced hardware capabilities.

The broad field of design optimization remains one of the principal objectives of many computational engineering problems. While low-fidelity methods are often used for non-local optimization problems, adjoint techniques have made the use of high-fidelity simulation tools practical for local optimization problems, particularly in the case of steady-state and single discipline problems. However, these capabilities must be extended to more complex problems such as time-dependent problems (for example involving the optimization of time-dependent rotorcraft blades or maneuvering vehicles), multidisciplinary optimization (aeroelastic, aero-servo-elastic, and aerothermo-servo-elastic optimization) [21, 22, 23, 24, 25], as well as optimization under uncertainty, often referred to as robust optimization [26, 27]. Additionally, improved non-local design optimization techniques [28, 29] which enable the use of high-fidelity simulations for finding truly global optima must be developed and made practical on future HPC hardware.

COMPUTATIONAL ENGINEERING ROADBLOCKS

High resolution requirements

One of the principal reasons that computational science and engineering problems are so demanding is that the governing physical phenomena involve such a wide range of spatial and temporal scales, which interact in complex nonlinear manners. In aerospace engineering problems, these scales occur both due to geometrical considerations, as well as due to the governing physical phenomena such as turbulence and transition. An illustrative example is provided in Figure 3(a), which depicts the wide range of scales present in unsteady turbomachinery flows. 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 (DPW) [30, 31, 32] have shown, lack of sufficient resolution remains the principal limiting factor in achieving accurate and predictive simulation outcomes, even for relatively well understood problems such as steady-state cruise condition aerodynamics. This is illustrated in Figure 3(b) where the collective results from the third DPW workshop plotted as a function of grid resolution are shown, illustrating a lack of convincing grid convergence even for grid sizes ranging up to 100 million points. 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.

For science-based simulations, which often can be performed in the presence of simple geometries that can be represented analytically or through customized software, increased spatial resolution can be accommodated in a relatively straightforward manner with expanding parallelism through multicore architectures. By contrast, the complex geometries most often associated with engineering problems have made the generation of highly resolved meshes (billion plus cells) impractical, since specialized (mostly commercial) grid generation and geometry definition (CAD) software is required, which most often runs in serial. Additionally, these large mesh datasets must be loaded onto the parallel machine at startup, and retrieved along with the computed solution at each time step (or selected time steps) of the simulation, for time-dependent dynamic mesh problems, constituting an I/O bottleneck on current HPC hardware. For these reasons, practical engineering applications have generally been confined to commodity-type hardware, making use of the order of hundreds of cpus or cores, and prospects for scaling up to extremely large parallel architectures remain uncertain.

Low resolution requirements

In spite of the need for high resolution described above, many engineering simulations provide suitable information without the need for overall high accuracy and resolution. This is because engineering problems are often concerned with a small number of specific objectives, which are needed to evaluate product performance or guide design decisions. Furthermore, the accuracy required for useful simulations in the design process is highly variable, where sometimes crude ball-park accuracy is sufficient, while at other times high accuracy is required. In most cases however, good accuracy in the relevant objective does not require fully resolving all physical phenomena present in the governing physics, but only those that have a strong impact on the important objectives. These characteristics of engineering simulations often mean that extensive spatial resolution (for example billion plus grid cell calculations) is not necessary for many problems. However, these problems remain extremely computationally intensive due to the need to take large numbers of small time steps, the cost of solving stiff non-linear systems at each time step, and in the case of design optimization problems, the cost of performing large numbers of optimization steps. Since most current-day applications rely on spatial domain decomposition for achieving parallelism, most engineering simulations have lagged at exploiting the levels of parallelism needed for achieving petaflops simulations on the most recent hardware. In such cases, parallelism nust be sought in directions other than the spatial dimension. For example, space-time parallelism may prove to be useful for long-time integration problems of modest spatial resolution. Enhanced parallelism may also be sought in the design space for optimization problems, or through the formulation of more scalable sensitivity analysis techniques. For uncertainty quantification problems, natural parallelism opportunities exist in the use of Monte Carlo or other ensemble averaging techniques.

Multidisciplinary Complexity

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 multi-physics 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. 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.

ENABLING TECHNIQUES

Specialized algorithms and techniques must be developed to overcome the roadblocks discussed in the previous section, and to enable the solution of specific engineering problems to benefit from current and future emerging HPC hardware.

Parallel Mesh Management

For simulations which require high levels of spatial resolution, parallelism will be obtained naturally through domain decomposition. However, the barriers to generating and maintaining highly resolved boundary fitted computational meshes, as mentioned previously, must be overcome. One approach to this problem is to invest in the development of parallel mesh generation strategies. Another approach is to place more reliance on in-situ parallel adaptive mesh refinement. Under this strategy, an initial coarse mesh is first generated sequentially using existing mesh generation software and loaded onto the parallel machine. As the simulation progresses, the mesh is adaptively refined and de-refined on the parallel machine. In fact, a comprehensive approach to enabling large-scale simulations should bypass all sequential and I/O bottlenecks by only sending coarse initial meshes to the machine, and only retrieving visualization information and a small number of engineering objectives from the simulation. The realization of this approach to large scale engineering simulation requires that a full description of the geometry be directly accessible from the parallel machine for tasks including adaptive mesh refinement and design optimization. In general this is not the case today, since most geometry definitions reside inside CAD systems which have only been ported to traditional workstation-class computers and are not available in source form. Therefore, flexible access to geometric information from current and future massively parallel architectures must be developed, using paradigms such as client-server models and/or surrogate geometry representations.

Alternate avenues for parallelism

As mentioned in the previous section, there exist many engineering problems which do not require ever increasing levels of spatial resolution, yet which remain very computationally intensive. In such cases, parallelism must be sought in directions other than the spatial dimension. For unsteady problems, various techniques have been proposed for extracting additional parallelism from the time dimension. One example can be found in time spectral methods, which simulate a nearly periodic flow using multiple instances in time which must all be solved simultaneously [33, 34, 35]. Another possibility is the use of space-time slabs, particularly when using different time step sizes in different regions of the domain within a time-implicit approach. Such an approach is motivated by the observation (using local error estimates) that the majority of the temporal error contributions is confined to relatively small spatial domains. This is illustrated by the temporal error plot in Figure 4(a), taken from a two-dimensional oscillating airfoil calculation using the adjoint approach to estimate temporal error with the airfoil lift as the objective function [36], showing the temporal error near the airfoil surface to be several orders of magnitude larger than the average domain error values. A variable time-step space-time slab formulation is illustrated in Figure 4(b) for a simple first-order backwards-difference time discretization. This approach results in a larger implicit system which can take advantage of increased parallelism in the time domain. Note that the entire time domain is not treated in parallel in this approach, but only each specific time slab in sequence. Additionally, computational savings are expected through the use of larger time steps in regions of the domain where high temporal accuracy is not required. In this approach, each space-time slab must be redistributed across the parallel machine in order to ensure load balancing for the ensuing fully-coupled implicit solution procedure, and this can be achieved using the same dynamic load-balancing approaches developed for parallel adaptive mesh refinement [37, 38].

For design optimization problems, and for uncertainty quantification problems, natural opportunities for increased parallelism exist by performing simultaneous simulations in parameter space. The principal development tasks in these cases involve the development of parallel management software for spawning cases and retrieving and combining results in an automated fashion, making the most effective use of available computational hardware. However, the computational requirements of these tasks can grow exponentially with increasing parameter space size, and the development of effective reduced-order models which can be computed efficiently in parallel will remain an important area of future development [39].

Increased simulation reliability

Adjoint methods have become popular for aerodynamic design optimization problems [40, 41, 42], and more recently for error estimation and control [43, 44, 36]. The power of adjoint methods is that they enable the calculation of the sensitivity of one simulation objective with respect to any number of simulation inputs or design variables at the cost of one adjoint solution, which is approximately equivalent to the cost of one additional simulation. In the discrete adjoint approach, the discretized form of the original governing equations is linearized and transposed, and then solved to obtain the sensitivities. This process can be modularized and extended to relatively complex simulation frameworks, including unsteady and multidisciplinary simulation problems [23, 45]. In addition to enabling efficient implementation of gradient-based optimization for such problems, this formulation can also be used to provide estimates of error sources arising in complex simulations, including spatial discretization errors, temporal errors, algebraic errors and disciplinary coupling errors. These error estimates can be used either to provide error bounds on the simulation outputs, and/or to drive adaptive processes which seek to reduce these different error sources through spatial or temporal refinement, or increases in convergence tolerances. Figure 5 illustrates the use of adjoint-driven temporal and algebraic error control for a pitching airfoil/vortex problem. In this case, the vortex is convected past a pitching airfoil, altering the unsteady lift distribution on the airfoil as the vortex approaches. The error in the time-integrated airfoil lift is estimated using the adjoint procedure, and used to refine the time step sizes in regions of large temporal error and to tighten up the implicit system convergence tolerances in regions of large algebraic error. The resulting automatically determined distribution of time step sizes and convergence tolerances after three adaptive passes are depicted in Figure 5(c) and (d) respectively. This adaptive procedure was shown to result in enhanced accuracy at reduced cost for the time-dependent lift objective compared to global time step and convergence tolerance settings in reference [36].

Adjoint-based error estimation problems are relatively expensive, particularly in the unsteady case, since the entire time-history of the solution must be saved, and the adjoint equations must be integrated backwards in time [23, 21, 24]. Additionally, if there exists more than one important objective, the process must be repeated for each individual objective. However, the benefit is a solution which is not only more accurate, but for which the accuracy level can be quantified with some certainty, and for which the relative contribution of various error sources can be identified. Therefore, as computational capabilities increase, it will prove useful to progressively incorporate more techniques of this sort, which enhance the quality of the solution at additional computational cost, and to seek out strategies for mapping multiple adjoint problems or sub-problems to parallel hardware for complex real world engineering problems.

Alternate discretizations

The use of alternate discretizations must also be considered for enhancing the scalability of engineering simulation problems. In particular, the use of high-order discretizations such as Discontinuous Galerkin methods offers promising benefits for use with future HPC architectures. Employing a higher-order accurate discretization entails the use of lower resolution meshes for equivalent levels of accuracy, as compared to a traditional second-order accurate formulation. This reduces not only the sequential grid generation and I/O bottleneck, but also de-emphasizes the cost and complexity of all grid related operations, such as pre- and post-processing, adaptive refinement, and dynamic load balancing. Additionally, DG discretizations rely on a compact (nearest-neighbor) stencil, with dense block matrices at the element level and demonstrate superior scalability on highly parallel multi-core architectures even for relatively coarse meshes. This is illustrated in Figure 6, where the scalability of a Discontinuous Galerkin solver running on a relatively coarse mesh of 2.5 million cells is plotted, using up to 2008 processors on the NASA Columbia Supercomputer. Although poor scalability is observed for low order discretizations (corresponding to the situation in traditional finite-volume schemes), scalability improves dramatically for higher order discretizations on the same mesh due to the increasingly large and dense matrix block sizes as the order of accuracy is raised.

CONCLUSIONS

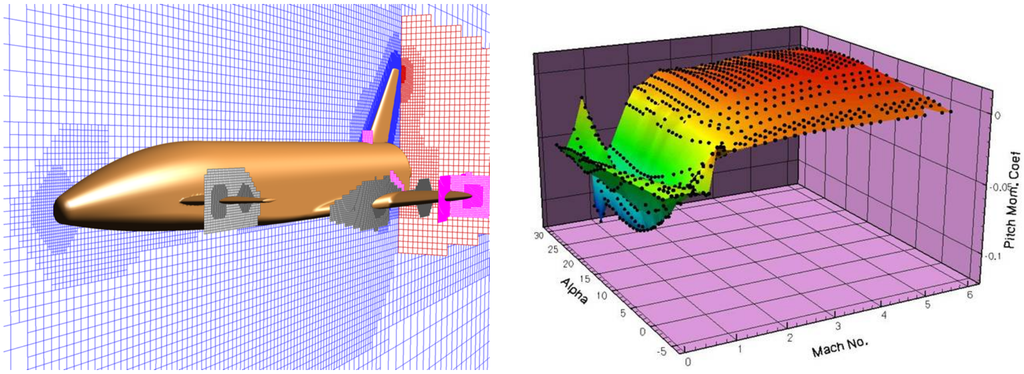
In this paper, we have outlined some of the characteristics which make engineering problems in general, and computational fluid dynamic problems in aerodynamics in particular, different than the class of discovery-based science problems which are increasingly dominating the field of high performance computing. Because we are on the verge of another paradigm shift in future HPC hardware trends, it is important to proactively consider how this will affect the field of computational fluid dynamics and to develop new strategies early on for dealing with these changes. We have identified some of the important barriers for porting current-day CFD methods to massively parallel multicore architectures, and have offered some insights into techniques for addressing these issues. The long term goal is to enable the development of large-scale engineering simulation tools which can take full advantage of future HPC architecture capabilities, thus opening new frontiers in engineering simulation capabilities. Failure to do so will result in the stagnation of our simulation capabilities as current tools are forced to run exclusively on low-end commodity hardware.

REFERENCES

1. In *Parallel Computational Fluid Dynamics - 1989. Manhattan Beach, CA, USA.*, May 1989. no proceedings.

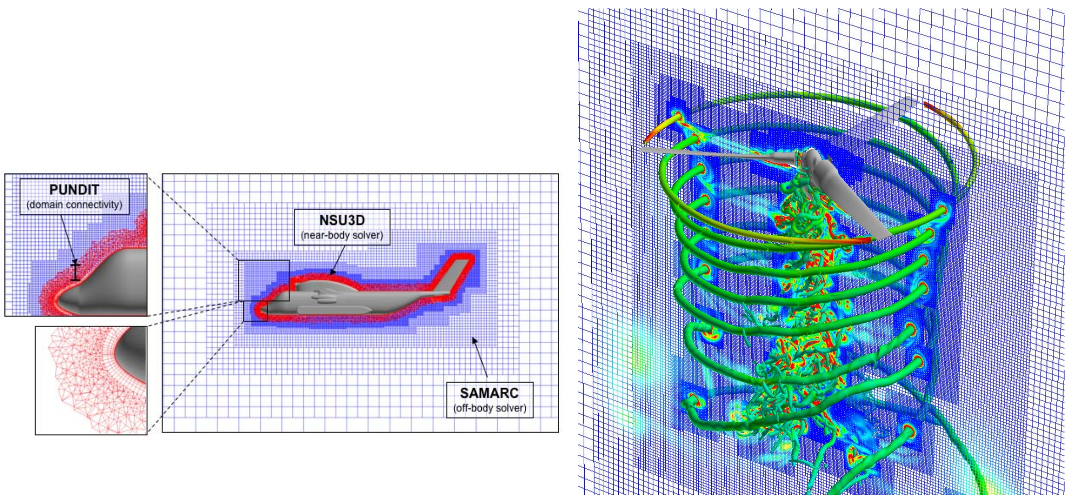
2. C. Bishof and H. D. Simon. The implementation of the US High Performace Computing and Communication Program. 1992.
3. SciDAC: Science discovery through advanced computing. US Department of Energy: see http://www.scidac.org/, 2006.
4. A science-based case for large-scale simulation, Volume I. *Office of Science, US Department of Energy*, July 2003. 76pp.
5. A science-based case for large-scale simulation, Volume 2. *Office of Science, US Department of Energy*, September 2004. 254pp.
6. 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.
7. D. J. Mavriplis. Large-scale parallel viscous flow computations using an unstructured multigrid algorithm. In *Parallel Computational Fluid Dynamics - Towards Teraflops, Optimization and Novel Formulations, Proceedings of the Parallel CFD 1999 conference, Williamsburgh, VA, USA; eds. D. Keyes, A. Ecer, J. Periaux, N. Satofuka and P. Fox*. Elsevier, May 1999.
8. 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. Paper presented at the First AFOSR International Conference on DNS and LES, Louisiana Tech University, Ruston, Louisiana, August 1997.
9. J. Forsythe, K. Squires, K. Wurtzler, and P. Spalart. Detached eddy simulation of fighter aircraft at high alpha. AIAA Paper 2002-0591, January 2002.
10. 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.
11. 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. Tied for Best paper award at SC05., November 2005.
12. 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.
13. 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.
14. J. Sitaraman, A. Katz, B. Jayaraman, A. Wissink, and V. Sankaran. Evaluation of a multi-solver paradigm for CFD using unstructured and structured adaptive cartesian grids. AIAA Paper 2008-0660, January 2008.
15. V. Sankaran. Computationally based engineering for air vehicle acquisition: Rotary wing simulation. In *Proceedings for the Parallel CFD 2009 Conference, Moffet Field, CA, USA;*, 2009.
16. C. Atwood. CREATE Air Vehicles overview. AIAA-Paper 2008-0711, January 2008.
17. R. Meakin. Computationally based engineering for air vehicle acquisition: The CREATE-AV program. In *Proceedings for the Parallel CFD 2009 Conference, Moffet Field, CA, USA;*, 2009.
18. 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.
19. M. J. Wright nd D. Bose and Y.-K. Chen. Probabilistic modeling of aerothermal and thermal protection material response uncertainties. *AIAA Journal*, 45(2):399–425, February 2007.
20. D. J. Mavriplis, D. Darmofal, D. Keyes, and M. Turner. Petaflops opportunities for the NASA fundamental aeronautics program. AIAA-Paper 2007-4048, Invited paper at the AIAA Computational Fluid Dynamics Meeting, Miami FL, July 2007.
21. K. Mani and D. J. Mavriplis. An unsteady discrete adjoint formulation for two-dimensional flow problems with deforming meshes. *AIAA Journal*, 46(6):1351–1364, 2008.
22. K. Mani and D. J. Mavriplis. Adjoint based sensitivity formulation for fully coupled unsteady aeroelasticity problems. *AIAA Journal*, 47(8):1902–1915, 2009.
23. M. P. Rumpfkeil and D. W. Zingg. The optimal control of unsteady flows with a discrete adjoint method. *Optimization and Engineering*, 2008. doi:10.1007/s11081-008-9035-5.
24. D. J. Mavriplis. Solution of the unsteady discrete adjoint for three- dimensional problems on dynamically deforming unstructured meshes. AIAA Paper 2008-0727, January 2008.
25. H. S. Chung and J. Alonso. py-MDO:A framework for high fidelity multidsciplinary optimization. AIAA Paper 2004-4480, September 2004.
26. M. M. Putko, A.C. Taylor, P. A. Newman, and L. L. Green. Approach for uncertainty propagation and robust design in CFD using sensitivity derivatives. AIAA-Paper 2001-2528, 15th AIAA

Computational Fluid Dynamics Conference, Anaheim CA, June 2001.
27. L. Huyse, S. Padula, R. M. Lewis, and W. Li. Probabilistic approach to free-form airfoil shape optimization under uncertainty. *AIAA Journal*, 40(9):1769–1772, 2002.
28. H. S. Chung and J. Alonso. Using gradients to construct co-Kriging approximation models for high dimensional design optimization problems. AIAA Paper 2002-0317, January 2002.
29. J. Laurenceau and P. Sagaut. Building efficient response surfaces for aerodynamic functions with Kriging and coKriging. *AIAA Journal*, 46(2):498–507, February 2008.
30. 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.
31. 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.
32. 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.
33. A. Gopinath and A. Jameson. Time spectral method for periodic unsteady computations over two- and three- dimensional bodies. AIAA-Paper 2005-1220, 2005.
34. S. Choi and A. Datta. CFD prediction of rotor loads using time-spectral method and exact fluid-structure interface. AIAA-Paper 2008-7325, 2008.
35. K. Ekici and K. Hall. Nonlinear analysis of unsteady flows in multistage turbomachines using harmonic balance. *AIAA Journal*, 45(5):1047–1057, 2007.
36. K. Mani and D. J. Mavriplis. Error estimation and adaptation for functional outputs in time-dependent flow problems. AIAA Paper 2009-1495, January 2009.
37. U.V. Catalyurek, E.G. Boman, K.D. Devine, D. Bozdag, R.T. Heaphy, and L.A. Riesen. Hypergraph-based dynamic load balancing for adaptive scientific computations. In *Proc. of 21st International Parallel and Distributed Processing Symposium (IPDPS'07)*. IEEE, 2007. Also available as Sandia National Labs Tech Report SAND2006-6450C.
38. A. Wissink, J. Sitaraman, D. J. Mavriplis, T. Pulliam, and V. Sankaran. A python- based infrastructure for overset CFD with adaptive cartesian grids. AIAA Paper 2008-0927, January 2008.
39. 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.
40. O. Pironneau. On optimum design in fluid mechanics. *Journal of Fluid Mechanics*, 64:97–110, 1974.
41. A. Jameson. Re-engineering the design process through computation. *J. of Aircraft*, 36:36–50, 1999.
42. H.-J. Kim, D. Sasaki, S. Obayashi, and K. Nakahashi. Aerodynamic optimization of supersonic transport wing using unstructured adjoint method. *AIAA Journal*, 39(6):1011–1020, June 2001.
43. D.A. Venditti and D.L. Darmofal. Grid adaptation for functional outputs: Application to 2-D inviscid compressible flow. *Journal of Computational Physics*, 176:40–69, 2002.
44. M. Wintzer, M. Nemec, and M. Aftosmis. Adjoint-based adaptive mesh refinement for sonic boom prediction. AIAA Paper 2008-6593, 26th AIAA Applied Aerodynamics Conference, Honolulu, HI, August 2008.
45. K. Mani. Application of the discrete adjoint method to coupled multidisciplinary unsteady flow problems for error estimation and optimization. PhD Thesis, Department of Mechanical Engineering, University of Wyoming, May 2009.
46. 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 ¡96¿ 2006 Workshop on Unsteady Flows in Turbomachinery*, volume NASA/CP ¡96¿ 2006-214484, December 2006.

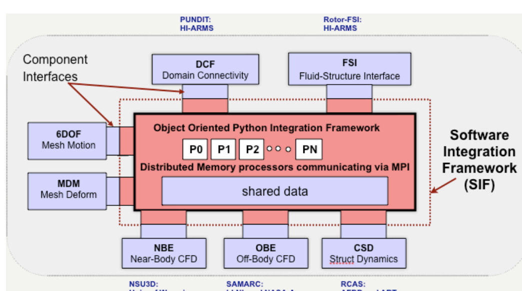


(a) (b)

Figure 1. Cut-cell Cartesian mesh and computed carpet plot showing the variation of pitching moment with Mach number and angle of attack at 0 degrees sideslip for an aerospace vehicle configuration. Reproduced from [10] with permission.

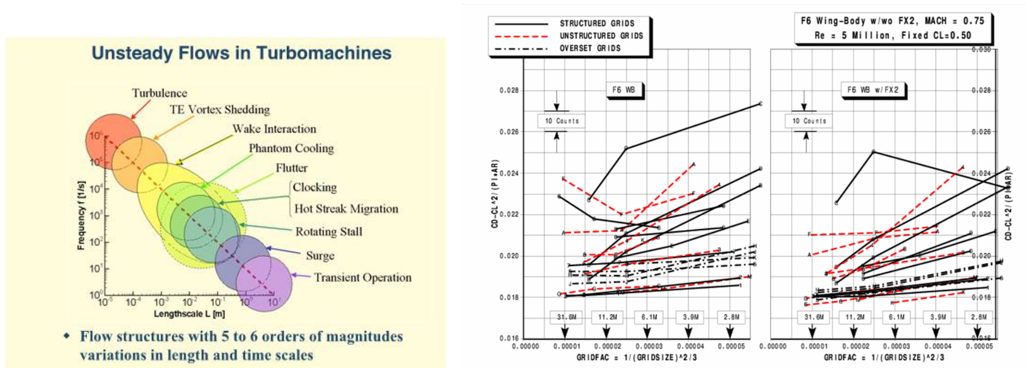


(a) (b)



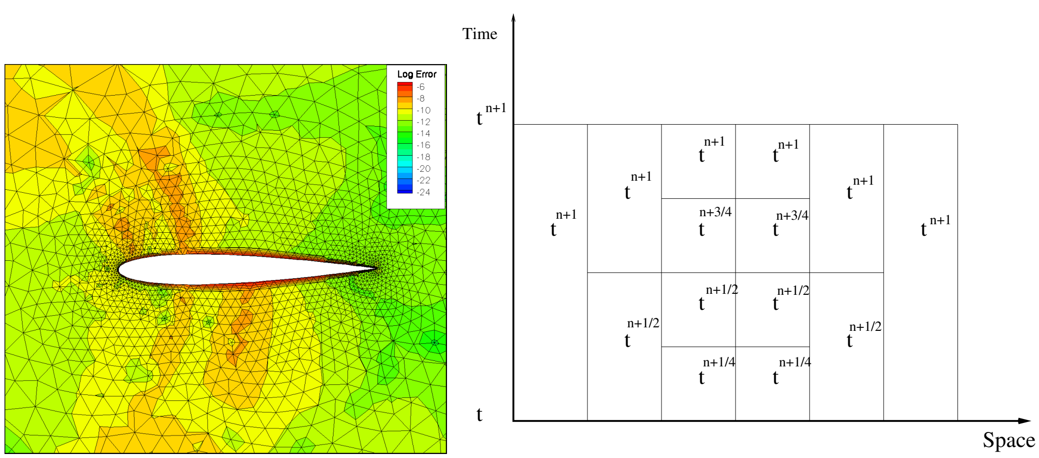
(c)

Figure 2. Illustration of multidisciplinary digital flight simulation for rotorcraft configuration under the US Army HIARMS project [14]; (a) Dual mesh paradigm using unstructured mesh near body and rotor, and structured mesh off-body; (b) Adaptive mesh refinement for capturing wakes and vortices; (c) Multidisciplinary software structure for linking disciplinary codes including CFD, structural dynamics, and flight control (6DOF).



(a) (b)

Figure 3. (a) Illustration of wide variation of scales in turbomachinery flow (reproduced from [46]); (b) Collective results from 3rd Drag Prediction Workshop showing questionable grid convergence with increasing grid resolution for wing-body test case with side-body separation (left), and wing-body test case with no separation (right).



(a) (b)

Figure 4. (a) Contours of adjoint-estimated temporal error for pitching airfoil problem illustrating large spatial variation of temporal error. (b) Illustration of implicit space-time formulation in one spatial dimension making use of different size time steps in different spatial locations in mesh.

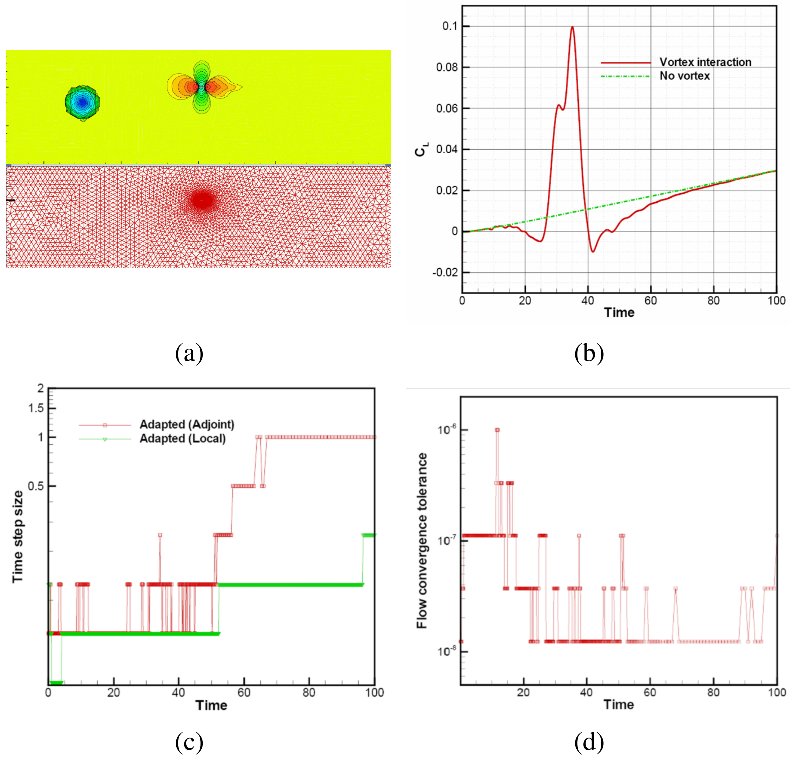


Figure 5. Adaptive solution of pitching airfoil/vortex problem reproduced from reference [36]. (a) Computational mesh and solution at initial time illustrating convecting vortex and airfoil position. (b) Time history of lift coefficient on airfoil. (c) Adaptively selected time steps using adjoint temporal error estimation scheme (compared with local temporal error estimation scheme based on ordinary differential equation integration techniques). (d) Adaptively selected implicit system convergence tolerance at each time step.

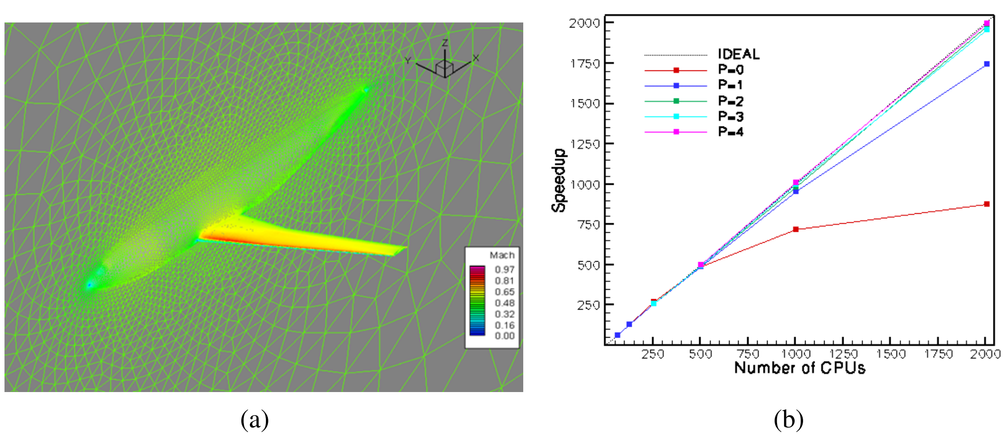


Figure 6. (a) Illustration of inviscid compressible flow computed over aircraft configuration using high-order Discontinuous Galerkin discretization using mesh of 2.5 million tetrahedra. (b) Scalability of DG aircraft simulation on NASA Columbia machine for various orders of discretization accuracy on same mesh.