1. Introduction

"Users should approach all software with prudent caution and healthy skepticism, for the history of science and engineering, including the still-young history of software engineering, is littered with failed promises." Henry Petroski[1]

1.1 An Overview: The Role of Computational Aerodynamics

What is computational aerodynamics (*CA*)? Theoretical aerodynamics has always provided insights to aerodynamicists through solutions of the governing equations of fluid mechanics. However, before computers became widely available the application of theoretical aerodynamics to specific problems was frequently impractical. Nevertheless, theoretical results from simplified model problems provided important insights which aerodynamicists used as a basis for developing aerodynamic concepts and understanding experimental results. However, aerodynamic design was carried out experimentally; primarily in wind tunnels. Starting nearly thirty years ago, and becoming increasingly important in the last decade, computational aerodynamics has become an important precursor and supplement to the use of the wind tunnel. Computational aerodynamics applies specific solutions of the governing equations of fluid mechanics to the design and analysis of vehicle systems. Usually this means the numerical solution of governing equations rather than numerical evaluation of analytically derived solutions. As soon as computers became available aerodynamicists started using them. The first computational aerodynamics computer programs that were reasonably general and easy to use became widely available in the late '60s, and started providing valuable design information for aerodynamics. Typically, they provided three-dimensional solutions for linear aerodynamics problems, and two-dimensional solutions of the nonlinear boundary layer equations. As with any new technology, this capability arose before engineers understood how to integrate it into the existing design process. Initially proponents claimed that computational aerodynamics would replace wind tunnels. It was well into the '70s before the early promise matured into a realization of the difficulties that would have to be overcome for computed solutions to replace wind tunnels. The wind tunnel is still in use, and, NASA has recently announced its intention to build two new wind tunnels. In the ensuing years computational aerodynamics has become an identifiable new technology, making important contributions to flight vehicle design. Now, there is a distinct body of knowledge that provides a foundation for work in the field.

Computational aerodynamics is one of the most important technologies in the development of advanced vehicles. Many engineers are actively involved in design and analysis using computational aerodynamics. Although numerous books have appeared describing the basic theory of computational fluid dynamics (CFD), guidance on the *application* of these methods is scarce. However, most engineers working in computational aerodynamics are applying these methods, and not developing new algorithms. There is a difference between CFD algorithm development and application skills. CFD algorithm developers have their specific interests and organizations. They are trying to solve fundamental algorithm problems and usually do not use their codes to do aerodynamic design and analysis. As a result, they generally have a poor understanding of the needs of and demands on the user[*]. Users must understand the algorithms and assumptions employed in the methods, and an education in the *effective use of the computational aerodynamics methods in engineering design and analysis*. The ability to approach aerodynamics problems using computational methods, assess the results, and make engineering decisions requires very different skills and attitudes than those associated with fundamental algorithm development.

Although you cannot use a computational aerodynamics code blindly and expect to obtain valid results, skilled engineers can obtain valuable results when computational aerodynamics is used with some skill, knowledge, ingenuity and judgment. The computer power available to every engineer today is greater than the total computing power available to the engineers who put men on the moon in the Apollo program, and even to those who designed the space shuttle. Unfortunately, it is possible for an engineer using this large computational power to make an error and not catch it. Several structural failures arising from faulty use of computational structures methodology have been documented recently.[1] Thus, significant responsibility accompanies the use of these immense computational capabilities.

It is impossible to anticipate the variety of requests that arise for computational aerodynamics analysis. Although we emphasize aircraft here, computational aerodynamics is also used in the analysis and design of missiles, cars, rotorcraft, submarines and ships. In addition to external flows, *CA* is used for internal flow problems, including inlets, turbomachinery, and nozzles. Although in a global, long-term sense, computational aerodynamics should replace the wind tunnel, for now this is not the case. Indeed, experimental and computational methods form a good complement to allow aerodynamicists to investigate problems and assess designs.

Typical major goals of computational aerodynamics include:

- vehicle design, *i.e*. development of optimum airfoils and wings for external performance, and inlets, diffusers, and nozzles for internal performance and aero-propulsion integration
- performance: estimation of the drag, lift, and moment characteristics of the vehicle
- definition of loads for structural design (including structural deformation under load)

---

[*] Although many developers lack interest in computing drag accurately, a few notable exceptions exist.[2]

- aeroelastic analysis, including flutter and divergence—requiring coupling with structural analysis and control system design analysis methodology)
- definition of aerodynamic characteristics for evaluation of stability, control, and handling characteristics (*i.e.*, provide the math model for flight simulation).

The current capability doesn't allow computational aerodynamics to accurately satisfy all these goals. Several difficulties prevent the use of computational aerodynamics in the most general situations, and engineering judgment must be exercised to obtain useful results. Difficulties preventing complete numerical simulation include both geometric and fluid mechanics complexity (one simple definition of aerodynamics is 50% flowfield, 50% geometry). The simplest fluid mechanics idealizations are available to provide information at the conceptual and preliminary design stages. Advanced computational methods, which are typically difficult to use and don't yet predict drag well, are used in a different role. The advanced methods are perhaps best used to investigate the detailed physics of the flow. The availability of detailed results over the entire surface, and also everywhere in the flowfield, provides a crucial supplement to wind tunnel testing. Used together, with wind tunnel data providing key anchor points to access and understand the accuracy of the computational method, significant advances in aerodynamic design have been demonstrated. Thus, advanced computational aerodynamics is truly an area where Hamming's adage,[†] "the purpose of computing is insight, not numbers" is true.

1.2 Current Status of Computational Aerodynamics

The capability of computational aerodynamics is continually improving. But, the claims of methodology developers not intimately acquainted with the problems of applying advanced methods should be viewed with caution. Algorithm developers frequently make overly optimistic claims. However, significant technology development resources are being directed toward improving the capability of CFD, and we can expect that in the future we will be relying much more heavily on CFD results alone to make engineering decisions. For example, the recent three-stage, air-launched, winged space booster Pegasus™[3] was designed using computational methods alone. No wind tunnel tests were done. The initial launches were successful, and it appeared that the accuracy of the analysis was adequate for this unmanned vehicle. However, after a subsequent launch failure, a dispute arose over whether the aerodynamics had been accurately predicted, or whether the control system was to sensitive to imperfections in the aerodynamic model. The problem was in the lateral-directional characterisitcs, an area often neglected by code developers.

A recent AIAA Progress Series volume edited by Henne[4] describes the state of the art in 1990 through many examples of applications (especially note the comments of Ray Hicks, a veteran CFD user and early advocate of the use of CFD in aerodynamics). For example, "normal" 2D airfoil analysis and design can now be done reliably using computational methods.

[†] Hamming authored a numerical methods book many years ago. The quotation cited is the frontispiece of the book.

A prospective computational aerodynamics user should understand the limitations. Bradley and Bhateley[5] have reviewed the situation and in 1983 proposed a classification scheme in terms of the types of flowfield. They divided the flowfield into seven categories, and categorized the capability to compute each type of flow over a variety of geometries of increasing complexity. Their capability chart is given in Table 1. The only capability they rated as good was attached flow over simple shapes. The capability today is better, but the classification idea is still valid and the capability is still the same in relation to each category.

Table 1. One point of view regarding computational aerodynamics capability.

Status of Computational Capability

	Attached Flow	Separated Flow	Vortex Flow	Mixed Vortex Attached	Mixed Vortex Separated	Dynamic	Complex Geometry Coupling
Axisymmetric and 2-D	good	fair	poor	poor	poor	fair	N/A[1]
Research Wing/Body	good	fair	fair	fair	poor	fair	poor
Transport Aircraft	good	fair	fair	fair	poor	fair	fair
Fighter Aircraft	fair	poor	fair	fair	poor	poor	poor
Special Purpose Aircraft	fair	poor	poor	poor	poor	poor	poor

from Bradley and Bhateley, reference 4

1 not applicable

Case studies provide another way to assess computational aerodynamics impact. Shevell[6] identified several aerodynamic design problems that arose in flight on various transport aircraft. He examined these problems to determine if the use of computational aerodynamics would have avoided these problems. His conclusion was that *uninformed* computational aerodynamics would likely not have prevented these problems. They included subtle aspects of attached flow airfoil and wing aerodynamics, the ability to compute deep stall characteristics of T-tail aircraft, the use of nacelle strakes to improve high lift and fuselage strakes to improve high alpha directional stability. The impact of CFD is being felt however. Rubbert and Goldhammer[7] have reviewed the situation at Boeing, and Busch[8] has reviewed the use of CFD in the design of the YF-23. In the case of the YF-23, an Euler analysis was used quite successfully. Thus, inviscid codes are proving to be of significant value at the project level in aerodynamic design.

However, the conceptual design community has voiced frustration with CFD.[9] At the conceptual design level decisions are made based on rapid evaluation of the performance potential of a variety of configurations, rather than the detailed study and development of a particular design over a number of years. At this level advanced CFD methods have not yet proven useful. The problem can be traced to the inability of the codes to predict drag directly in a conceptual design sense. Part of the problem here is a miscommunication between the conceptual design aerodynamicists and code developers. Conceptual designers want to know what level of performance can be expected from a configuration *after the aerodynamic design is done*. The aerodynamic design of a single configuration may take months (or years). Although work is in progress to improve the design situation, current advanced CFD methodology is essentially an analysis tool. To discriminate between a series of different candidate configurations using CFD a rapid design capability with accurate estimation of the eventual drag level achievable must be available. Linear theory methods provide some of this capability, but nonlinear methodology for complete configurations on the time-scale of a day is not yet available.

Although absolute values of drag are currently considered too difficult to compute using CFD, the AGARD Panel on CFD and Drag[10] suggested that CFD-based drag prediction was very effective when "embedded in an increment/decrement procedure involving experimental results for complete configurations, and CFD results for simplified configurations." Another useful application of CFD in this context is the assessment of wind tunnel model support effects and wall interference, which was done for the YF-23.[8]

1.3 Objectives and Guiding Principles in Using Computational Aerodynamics

The objective of this text is to provide an overview of computational aerodynamics as currently practiced, and an understanding of the basis for this technology and the terminology. We will emphasize the assumptions used in the various methods. We provide both the foundations and motivation for further study in computational aerodynamics. We will also use the available computational aerodynamics methods to develop an understanding of applied aerodynamics using computational methods. Although the objective is an emphasis on applications, the underlying theory is provided in some detail. Code implementation details are continually changing. However, much of the fundamental theory is now becoming well defined, and an understanding of the foundations of the methods is essential.

What is more important, we include many examples showing what steps users must take to determine if the answers they are obtaining in their applications are reasonable. How will you know if the answer is right when an engineering decision must be made based on computational aerodynamics? As discussed above, blind acceptance of computed results will lead to problems. Similarly, as described by Hancock,[11] advances in computational capability have led to increased demands on experimental aerodynamics. More experimental data must be taken and the conditions must be much more exacting than the level of aerodynamic testing frequently conducted in the past. Examples of the resulting interplay between computational and experimental work were given recently by Neumann.[12] In addition, code validation has become a field in its own right. Code assessment for the range of validity and accuracy is difficult and time consuming. However, the importance of this step cannot be overemphasized. The issues are described in detail in the paper by Bobbitt,[13] and the importance of code validation was reinforced in the 1993 Dryden Lecture,[14] which addressed code validation and defined the NASA Ames approach to the problem. The sidebar below is from a recent article by Petroski.[1] Each engineer must test a code before using it to make a decision.

> “ Perhaps the most damaging limitation is that software can be misused or used inappropriately by an inexperienced or overconfident engineer.”
>
> “ No software can be proven with absolute certainty to be totally error-free, and thus its design, construction and use should be approached as cautiously as that of any major structure, machine or system on which human lives depend. Although the reputation and track record of software producers and their packages can be relied on to a reasonable extent, good engineering always involves checking them out. If the black box cannot be opened, a good deal of confidence in it and understanding of its operation can be gained by testing.
>
> The proof tests to which software is subjected should involve the simple and ordinary as well as the complex and bizarre. A lot more might be learned about a finite element package, for example, by solving a problem whose solution is known rather than one whose answer is unknown. In the former case, something might be inferred about the limitations of the black box; in the latter, the output from the black box might bedazzle rather than enlighten. In the final analysis it is the proper attention to detail—in the human designer’s mind as well as in the computer software—that makes the most complex and powerful applications work properly.”[1]

Thus the objective of any computational aerodynamics work must be:

- Is the answer right?
- Assuming the answer is correct, what is computational aerodynamics revealing about the physics of the flowfield?

In this text current codes are described for each class of methods. This provides the reader with a basis for understanding what capability to expect, and a starting point in searching for an appropriate method. Readers should understand that these surveys are subject to rapid change when describing methods currently considered advanced.

1.4 Typical Steps to Using Computational Aerodynamics, the Art of the Analyst

Given a flowfield or aircraft to examine, we start with a physical problem, and then represent the physical situation with a mathematical model. We then obtain a solution for the mathematical problem and use that solution to deduce something about the physical problem. As noted above, skill and experience are required to carry out this sequence of steps. In particular, judgment has to be used to select the method to be used. Sometimes, within the allotted budget and time, a CFD solution cannot be used to obtain the desired results. That's why you have engineers and not engineering aides performing the analysis.

The process is given by Rubbert and Tinoco,[15] and is illustrated in Fig. 1, as requiring the following steps:

- Start with the real flow around the aircraft.
- Create a physical model of the flowfield, perhaps (and traditionally) considering it as an inviscid transonic flow, a boundary layer flow and a wake.
- Create the simplified mathematical model(s) to be solved.
- Carry out the numerical solution.
- Examine the results.
- Interpret the sequence of physical model, mathematical model, and numerical solution, together with the computed results to provide the final aerodynamic solution.

Notice here that the numerical solution of a computational problem is a small part of the total engineering process. Successful aerodynamicists must master the entire sequence of steps.

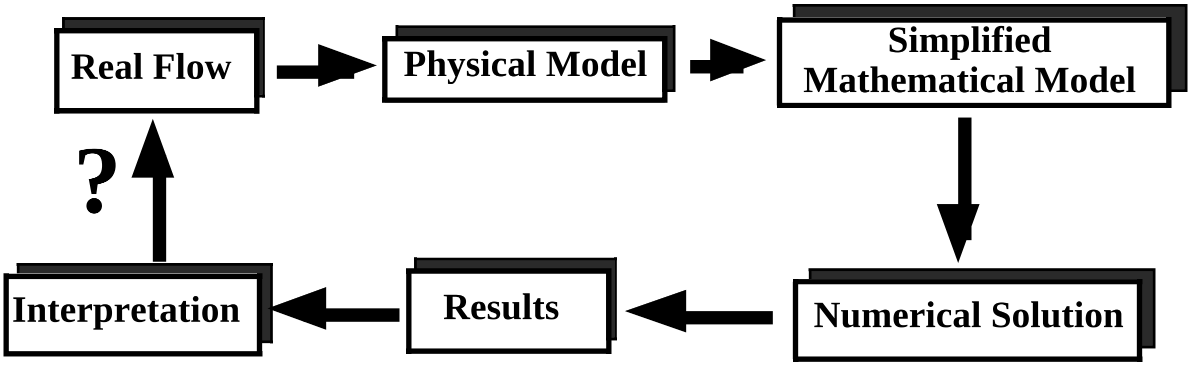

Figure 1. Steps in applying computational analysis to aerodynamics (Ref. 14).

1.5 Design vs Analysis: Computational Aerodynamics in Vehicle Design

Classical: repetitive analysis to design

Although computational fluid dynamics has become a major area of research, its use in the early stages of aircraft configuration development is not generally understood. An incredible variety of problems arise in advanced design, and this precludes the standard use of any simple, uniform procedure. Since the conceptual and preliminary design phases determine the basic configuration architecture, this is the area where improved design methods can make the biggest impact.

New configurations must exploit advanced technology to achieve improved performance over existing designs. In an ideal situation the new aircraft will incorporate new component concepts that have been developed extensively in basic R&D programs. One example is the use of advanced transonic airfoils that in the ’60s and early ’70s were developed in the wind tunnel. Today they are designed reliably using 2D computational codes such as GRUMFOIL.[16] Another example is the incorporation of the SC3[17] concept in a highly swept fighter design requiring efficient supersonic maneuvering.

A key to the successful development of a configuration is the participation of an experienced team that can project the possibilities for advanced performance without performing the work in detail. This experience base must be the result of having worked extensively with advanced vehicle design and computational methods. Hopefully, with this experience, reliable projections of the performance that can be obtained using computational design methods can be made with confidence.

Linear theory methods are used in conceptual design on a daily and even hourly basis. The aerodynamicist works with the configuration designer to develop a properly balanced design with optimum trimmed performance. This part of the problem can usually be treated well enough for conceptual design using linear methods. However, the high angle of attack characteristics and determination of acceptable control power is still a challenge for all levels of computational aerodynamics codes.[18] Currently, the “timescale” for using advanced codes is too long for conceptual design, approaching the time required for a wind tunnel test. It may require weeks or even months to obtain reliable solutions over a complete new geometry. Figure 2 presents an analogy between wind tunnel testing and computational methods.[19] The geometry definition required for advanced analysis is equivalent to the requirements for fabricating a wind tunnel model. This definition can be time consuming. Thus advanced codes are normally used to assess only a few specific features of a new vehicle concept. That feature might be a unique configuration idea, where you need to evaluate the viability of a new concept to reduce risk, and assure the program managers that the vehicle concept is realistic. Usually there is only time to examine one aspect of the design using advanced codes during the conceptual design phase.

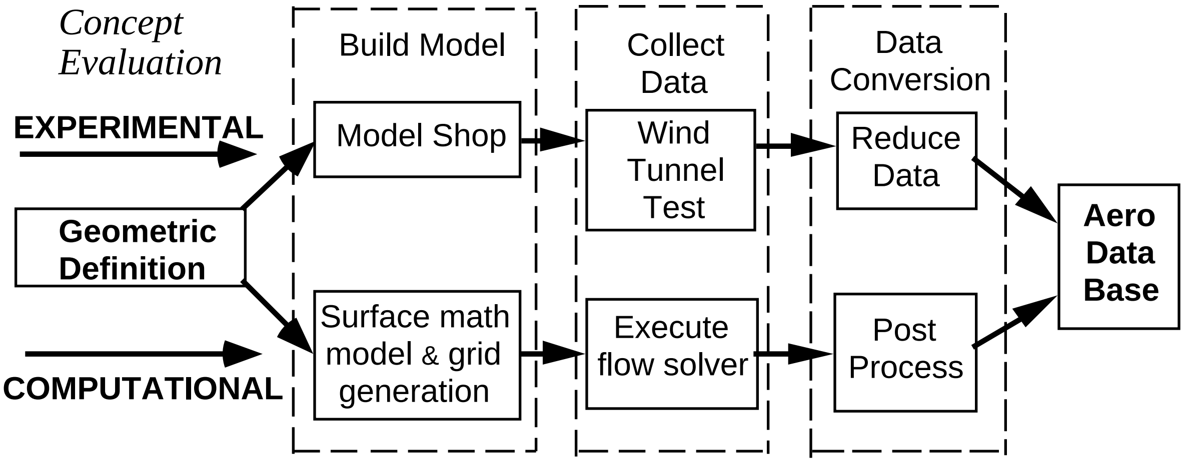

Figure 2. Analogy between computational and experimental aerodynamics, based on a figure by Bengelink.[19]

The Grumman X-29 forward swept wing airplane provides an illustrative example. The key idea in the Grumman proposal for a forward swept wing arose due to work on an aft swept wing design using an extremely high performance airfoil, the Grumman “K” foil. That airfoil was developed in 1974, initially by Don MacKenzie and then by MacKenzie working with Paul Bavitz. They used the hodograph method (see Chap. 11), and an early transonic viscous code developed by Bavitz[20] when he was on assignment from Grumman to NASA Langley working for Whitcomb. Wind tunnel work by Glenn Spacht (1977) led to the realization that the full performance of this airfoil could be realized only if the wing had a highly swept trailing edge, combined with reduced sweep on the leading edge. An inverse-taper, aft-swept wing or a conventionally tapered forward-swept wing planform provided the only means of meeting these requirements. Today, that conclusion could have been reached using 3D computational methods. The airfoil technology development was done using advanced methods. The rest of the design was done using more conventional (at that time) linear aerodynamic methods, verified and refined during wind tunnel tests.

Thus, computational aerodynamics was directly responsible for the airfoil design which was the core technology that led to the development of the X-29. Nevertheless, the wing design was done with the aid of at least some wind tunnel work. Two other aspects of the design besides the point performance development required testing. To define the aerodynamic model required to design the control system, the aerodynamic characteristics were documented over the complete range of angle of attack and sideslip for every combination of control surface deflections in an extensive wind tunnel test program before the first flight[*] was possible (note that the control system design is now frequently the reason for programs delays of advanced aircraft—detailed definition of the aerodynamic characteristics is required). Wind tunnel testing was also required to define the load distributions for structural design. The critical design loads usually occur at conditions well away from the design point and involve many loading conditions under separated flow conditions.

---

[*] The first flight took place on December 14, 1984, ten years after the airfoil was designed.

Examples of the interplay between configuration design and computational aerodynamics in the near future are the use of tailored forebodies developed using computational aerodynamics to provide specific levels of directional stability at high angle of attack, or components designed to achieve significant regions of laminar flow. The actual preliminary design phase would then integrate these components into the configuration without extremely intense computational aerodynamics work.

The use of computational aerodynamics becomes routine once the configuration geometry is well defined. Computational aerodynamics is heavily used in conjunction with wind tunnel tests once the program moves into a demonstration/validation stage. Often the codes are used to design incremental modifications to the wind tunnel-tested configuration. Many examples of configuration refinements and modifications using computational aerodynamics have been documented at this stage. Perhaps the best example is the design of the nacelle-wing installation on the recent Boeing transports, and especially the new big-engine 737.[15]

Advanced: Direct design and optimization

More important, but lagging behind the development of computational aerodynamics as an analysis tool, is the use of the computer to design and optimize the configuration directly. In this role the computer is used in a fundamentally different way than as the computational equivalent of a wind tunnel. This is the most efficient use of computational aerodynamics in vehicle design. Many ideas have been proposed and it is currently an area of active research, but relatively few methods have actually been completely developed. We will discuss and compare these methods in the following chapters. Here, again, the most successful applications have been for airfoils. Further discussion and references can be found in the paper describing “Smart Aerodynamic Optimization,”[21] and in the recent work in the area by a pioneer of analysis methods, Anthony Jameson.[22,23]

1.6 A Brief History of Computing Systems and Computational Aerodynamics

The development of computational aerodynamics is closely linked to the development of computers, and more recently computing systems and software. Recall that computers were, at least in part, originally developed for aerodynamics work: the creation of accurate ballistic shell trajectory tables. The other original reasons to develop computers were for cryptography, and subsequently for the nuclear weapons program.

Perhaps the most important early computational aerodynamics work conducted in the ’50s and ’60s was the work on “the blunt body problem.” At that time the prediction of the heat transfer and flight characteristics of ballistic missiles and manned space capsules entering the atmosphere were the “hot” items in fluid mechanics. Much of this work was done in machine and assembly language. Compilers were not advanced, frequently containing bugs themselves. Programmers thought that compilers produced code that executed much more slowly than code written by professional programmers in machine or assembly language (the original developers of programmable computers, Von Neuman and Turing, assumed that machine level instruction was sufficient—they never considered high level languages to be necessary). Patching bugs in the executable code was considered possible! These computers were slower and had much less storage than an Apple][Computer. Trying to solve problems that exceed the capability of the computer is a standard feature of computational aerodynamics. Despite rapid advances in computing technology, aerodynamicists always demand more speed and storage; even with the current Cray C-90.

Vortex lattice methods for aircraft applications were reported in 1963 (in Sweden, and at Boeing and Grumman in the US). At that time the vast majority of advanced fluid mechanics work was associated with the space program. Other methods were also under development, with Douglas being a leader.

I first used a computer in 1966 at McDonnell Aircraft in Saint Louis, Mo. We were doing trajectory analysis for hypersonic vehicles. The aerodynamic characteristics were calculated by hand using Newtonian flow theory. Cards were used for input, and some cases were run locally in Saint Louis (we had a "priority" of 5 minutes a night of CPU time, and no one talked about what computer was used). Long jobs were run on an underutilized company computer in Houston and output was flown back on the company plane, which made frequent trips to Houston (these were the days of Project Gemini).

The introduction of the IBM System 360 in the mid ’60s revolutionized access to computers for non-specialists. This was the first widely available, easily used computing system. VPI acquired a System 360 at that time, and an addition was made to Burruss Hall to house the computer center. Until then the university computer had been housed in a temporary wood building near the present site of Derring Hall. Initially, the only access to the computer was through submission of a box of cards. Jobs were run in a batch mode, and FORTRAN II was being supplanted by FORTRAN IV. It took hours or even days to get a job back. Students should understand that the computing power available through this process was much less than they have on today’s PCs. With the introduction of FORTRAN IV the scientific computing community started using a language that would be stable for many years. CDC introduced the CDC 6000 series computers at about this time, and the CDC 6600 became the computer of choice for scientific computing. Seymour Cray was one of the key designers of that computer. Later the CDC 7600 was introduced, and the CDC 7600 at NASA Ames, using the SCOPE operating system, was the best system I ever used. Our access to the CDC 7600 was still by submission of card decks. We used a CDC development environment known as UPDATE, that was an approach to what is known today as version control. It worked very well.

By 1970 aerodynamicists were solving linearized inviscid three dimensional incompressible flow problems routinely, and two dimensional boundary layer methods were available. The most important problem being tackled in 1970 was the computation of transonic flow. The first solutions began to be reported around 1970, and the first practical solution procedure was reported in 1971. The first half of the ’70s was dominated by the development of solutions for two-dimensional transonic flow. Three-dimensional transonic small disturbance theory solutions also began to appear.

At a major conference at NASA Langley* in 1975 one speaker drew on the rapid advances in capability to present a chart which could be used to project that computational aerodynamics would be fully developed by 1984.

A joke by the CFD researchers at NASA Ames in the mid-70s reflects the attitude of the time:

Question: “What do you use wind tunnels for?”

Answer: “They are places with lots of space, where you store your computer output”

Well, this shows that prediction is tricky, especially when it involves the future. (YB)

The early explosive advance in capability did not continue, and progress slowed. Advances became much more difficult. Why? Computational fluid dynamics (CFD) development became more rigorous, and:

i) Complexity of three-dimensional flowfields is not just “one more dimension.”
  - algorithms for the Euler equations were difficult to develop
  - computer storage requirements made remarkable demands on computers
  - handling this much information, pre- and post- computation is a job in itself

ii) Separated flow solutions were required.
  - numerical algorithms required further development
  - after the storage requirements began to be overcome, the inevitable limits of turbulence models became apparent

iii) Real life arbitrary geometry presented surprising challenges.
  - grid generation became a discipline in its own right

iv) Software development is the “tar pits” of engineering (see Chap. 3)
  - as more people work on a code the productivity decreases dramatically, individuals can no longer single-handedly create a complete new code

The situation today is again fluid. Key new developments in computing revolve around the dramatic advances in workstation technology, graphical interaction with results, and the computation of solutions using massively parallel processing technology. After working in an essentially stable

---

* “Aerodynamic Analysis Requiring Advanced Computers,” March 4-6, 1975. (see NASA SP-347)

computing environment for twenty years, computational aerodynamicists will be using exciting new computer hardware and new software products in the near future.

One advance in productivity is the adoption of UNIX as a common operating system on most current scientific computing systems. Although this can be an emotional subject, computational aerodynamics users benefit greatly. Code developers and computer systems people typically work with a single operating system. But, as a user, I have been in the position of having to compute on many different systems, frequently all in the same day. As an example, consider an actual case where proficiency in IBM CMS and TSO, VAX VMS, and Cray COS were required during the course of a single day's work effort. Each system had a different text editor! UNIX and *vi*, the common text editor, eliminates this problem.

1.7 Typical Method/Code Development Cycle

Code development is a long process. Initial efforts to develop the algorithm are only a portion of the effort. Figure 3 shows the typical process. Initial demonstration of a new algorithm is often done by a single CFD researcher and may take a year or two. Technology development is typically carried out by several engineers, applying the method using pilot codes tailored to solve a specific problem. The code itself is continually adjusted to obtain improved results and handle unforeseen situations. This activity may also last one or two years.

If the method provides a significant improvement compared to existing codes, it may be developed into a production code. Frequently computer science majors direct this activity. An attempt is made to anticipate future requirements, and make a very general code. This may also take years.

Once the production code is finished, the user has a code which typically cannot be changed without major coordination with support groups. Since five years may have elapsed since the method was originated, more advanced methods may already be available in the technology development stage, and the production code *may not be used as often as expected*. Instead, the latest capability may be available in a new pilot code. Thus pilot codes are frequently used in advanced vehicle analysis and design to get the best possible answers. How this problem is best handled requires the use of engineering judgment. The continually changing software problem adds another complication to the practice of computational aerodynamics.

One significant philosophical change emerged in code development over the last decades. Initially, each code had its own input geometry definition and graphics package to examine the results. Because this information is common to most codes, these tasks were separated, and the codes began to be designed so that the geometry/mesh generation became distinct from the flow solver. In addition, the graphical analysis was separated. Essentially, the results were stored as databases to be examined with another program. This allowed for more efficient code development. For the flow solver part of the problem this meant that the code required pre- and post-processing software. This work could be done independently of the flow solver. This approach, illustrated in Fig. 4, allows much more flexibility. Generally, it's much better for the user, although for some simple calculations it's much simpler if the flow solver handles grid generation and graphical output. This is sometimes possible in two-dimensional analysis, but rare for three-dimensional analysis, where the amount of data generated can be overwhelming.

An important consideration is the teamwork approach. Figure 4 showed the split of the code tasks. The use of a team to develop codes is just as important as the codes themselves. Whitfield has recently described his own experiences.[24] He found that using teams with individuals responsible for specific parts of the work is the key to good productivity. In particular, geometry definition and grid generation are areas where considerable skill and dedication are required. In his lab one individual is responsible for all the grid generation work. This is a graduate student or research engineer level activity, and productivity has improved using this approach. Similarly, post processing using computational flow visualization has been done by dedicating people to do this work exclusively. In this case undergraduate students are capable of handling the job. Consistent with my own experience, Whitfield has found that the *codes* do not produce results, but *people* produce results using the codes. The human element is at least important as the software and hardware. Finally, to simplify problems comparing results from different flow models, Whitfield is now using only Euler or Navier-Stokes solutions, and has stopped using the simpler flow models shown in Fig. 4.

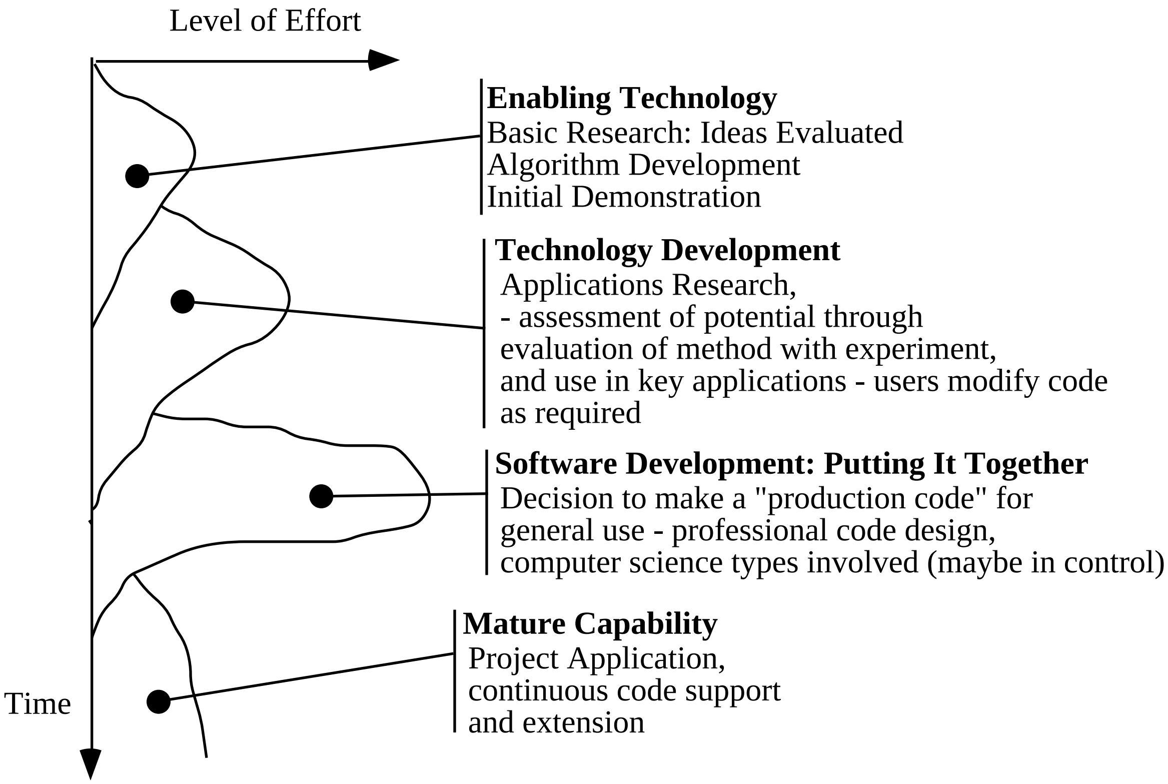

Figure 3. Code Development Process

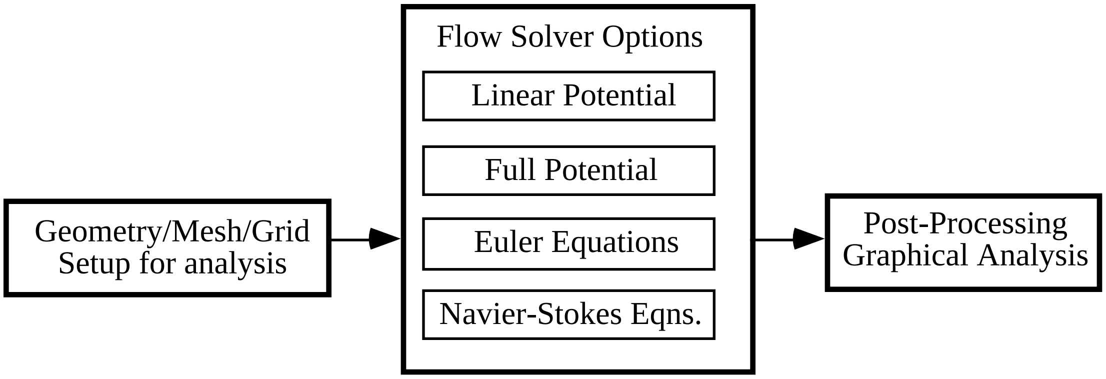

Figure 4. Typical split of functions in a CFD software system.

1.8 Overview of the following chapters

This text provides a systematic development of computational methods starting with early “pre-CFD” methods of computational aerodynamics that are still useful in aerodynamic design and analysis. These methods are used to introduce essential aspects of applied aerodynamics for airfoils and wings and an introduction to drag calculation methods. The basic ideas of CFD and grid generation are then discussed, followed by a presentation of viscous effects and transonic flows in aerodynamics. Finally, a discussion of extensions required to treat high speed aerodynamics problems is presented. In each case the theory is described, fundamental assumptions identified, a numerical implementation is presented, and examples illustrating the use of the method to understand aerodynamic design and analysis are given. Having acquired insight into basic applied aerodynamics, a brief tutorial covering the current advanced methods is presented. We conclude with a review of good computational aerodynamics procedures required to use computational aerodynamics in practice.

Upon the completion of the text you should be able to assess a problem for analysis using computational aerodynamics, formulate the problem, select a method, and obtain a solution. Then you should be able to use engineering judgment to decide if you have a valid engineering answer.

1.9 Exercise

Pick an airfoil. Select any airfoil of your choice for which you can find geometry, and the *experimental* pressure distributions, and force & moment data.

1. Plot the airfoil.
2. Plot the pressure distribution at one angle of attack.
3. Plot the force and moment data over a range of angles of attack. Make sure to include the drag polar.
4. Turn in a cover sheet describing airfoil and data source (make sure to include the test conditions: Reynolds number, Mach number and transition details, i.e., fixed or free transition. If fixed, where is it fixed?)).

Caution: You will use these data to compare with results from a computer program, and this data set will play a role in several assignments, so pick an airfoil you can use all semester. See Appendix A for additional information on airfoils, Appendix B for sources of data, and Appendix C for directions on the presentation of results (all of which aren't repeated here).

1.10 References

1. Petroski, H., "Failed Promises," *American Scientist*, Vol. 82, Jan-Feb. 1994, pp.6-9.
2. Jameson, A., "Re-Engineering the Design Process through Computation," AIAA Paper 97-0641, Jan. 1997.
3. Mendenhall, M.R., Lesieutre, D.J., Whittaker, C.H., and Curry, R.E., "Aerodynamic Analysis of Pegasus™ — Computations vs Reality," AIAA Paper 93-0520, Jan. 1993.
4. Henne, P.A., ed., *Applied Computational Aerodynamics,* AIAA, Washington, 1990.
5. Bradley, R.G., and Bhateley, I.C., "Computational Aerodynamic Design of Fighter Aircraft - Progress and Pitfalls," AIAA Paper 83-2063, August 1983.
6. Shevell, R.S., "Aerodynamic Bugs: Can CFD Spray Them Away?" AIAA Paper 85-4067, October, 1985.
7. Rubbert, P., and Goldhammer, M., "CFD in Design: An Airframe Perspective," AIAA Paper 89-0092, Jan. 1989.
8. Busch, R.J., Jr., "Computational Fluid Dynamics in the Design of the Northrop/McDonnell Douglas YF-23 ATF Prototype," AIAA 91-1627, June 1991.
9. Snyder, J.R., "CFD Needs in Conceptual Design," AIAA Paper 90-3209, September, 1990.
10. Sloof, J.W., ed., "Technical Status Review on Drag Prediction and Analysis from Computational Fluid Dynamics: State of the Art," AGARD-AR-256, May 1989.
11. Hancock, G.J., "Aerodynamics - the role of the computer," *Aeronautical Journal,* August/September, 1985, pp. 269-279.
12. Neumann, R.D., "CFD Validation - The Interaction of Experimental Capabilities and Numerical Computations," AIAA Paper 90-3030, August, 1990.
13. Bobbitt, Percy J., "The Pros and Cons of Code Validation," AIAA Paper 88-2535, June, 1988.
14. Marvin, J.G., "Dryden Lectureship in Research, A Perspective on CFD Validation," AIAA Paper 93-0002, Jan. 1993.
15. Rubbert, P.E., and Tinoco, E.N., "Impact of Computational Methods on Aircraft Design," AIAA Paper 83-2060, August, 1983.
16. Melnik, R.E., Chow, R.R., Mead, H.R., and Jameson, A., "A Multigrid Method for the Computation of Viscous/Inviscid Interaction on Airfoils," AIAA Paper 83-0234, 1983.
17. Mason, W.H., "A Wing Concept for Supersonic Maneuvering," NASA CR 3763, 1983.
18. Morrocco, J.D., "Lockheed ATF Team Cities Lessons Learned in Prototyping Effort," *Av. Wk.*, Nov. 5, 1990, pg. 87.

19. Bengelink, Ronald L., “The Integration of CFD and Experiment: An Industry Viewpoint,” AIAA Paper 88-2043, 1988.
20. Bavitz, P.C., “An Analysis Method for Two-Dimensional Transonic Viscous Flow,” NASA TN D-7718, 1974.
21. Aidala, P.V., Davis, W.H., Jr., and Mason, W.H., “Smart Aerodynamic Optimization,” AIAA Paper 83-1863, July 1983.
22. Jameson, Anthony, “Successes and Challenges in Computational Aerodynamics,” AIAA Paper 87-1184, 1987.
23. Jameson, A., “Aerodynamic design via control theory “ ICASE Report No. 88-64, Nov. 1988.
24. Whitfield, David L., “Perspective on Applied CFD,” AIAA Paper 95-0349, January, 1995.