1 INTRODUCTION AND GENERAL CONSIDERATIONS

Before going into a detailed description of applied Computational Fluid Dynamics (CFD) techniques, it seems proper to define its place among related disciplines. CFD is part of computational mechanics, which in turn is part of simulation techniques. Simulation is used by engineers and physicists to forecast or reconstruct the behaviour of an engineering product or physical situation under assumed or measured boundary conditions (geometry, initial states, loads, etc.). A variety of reasons can be cited for the increased importance that simulation techniques have achieved in recent years.

(a) *The need to forecast performance.* The inability to forecast accurately the performance of a new product can have a devastating effect on companies. The worst nightmare of an aircraft or car manufacturer is to build a prototype which has some hidden flaw that renders it inoperable or seriously degrades market appeal. Of the many examples that could be cited here, we just mention flutter or buzz for aircraft and unforeseen noise or vibrations for cars. With the development costs for new products being so large (about $\$4 \times 10^9$ for a new aircraft, $\$10^9$ for a new car; these and all subsequent quotations are in US$ and are accurate in the year 2000), a non-performing product can quickly lead to bankruptcy. The only way to minimize the risk of unexpected performance is through insight, i.e. information. Simulation techniques such as CFD can provide this information.

(b) *Cost of experiments.* Experiments, the only other alternative to simulations, are costly. A day in a large transonic windtunnel costs about $\$10^5$, not counting the personnel costs of planning, preparing the model, analysing the results, etc., as well as the hidden costs of waiting for availability and lost design time. An underground test for a nuclear device costs about $\$10^8$, and for a conventional weapon $\$10^7$. Other large experiments in physics can also command very high prices.

(c) *Impossibility of experiments.* In some instances, experiments are impossible to conduct. Examples are solar and galactic events, atmospheric nuclear explosions (banned after the Test Ban Treaty of 1963), or biomedical situations that would endanger the patient's life.

(d) *Insight.* Most large-scale simulations offer more insight than experiments. A mesh of 2×10^7 gridpoints is equivalent to an experiment with 2×10^7 probes or measuring devices. No experiment that the author is aware of has even nearly this many measuring locations. Moreover, many derived diagnostics (e.g. vorticity, shear, residence time, etc.) can easily be obtained in a simulation, but may be unobtainable in experiments.

(e) *Computer speed and memory.* Computer speed and memory capacity continue to double every 18 months (Moore's law). At the same time, algorithm development continues to

Applied Computational Fluid Dynamics Techniques Second Edition Rainald Löhner

Table 1.1. Increase of problem size

Size	Dimension	Code	Year	Problem	Machine
$>10^2$	2-D	FEFLO20	1983	Airfoil	ICL
$>10^3$	3-D	FEFLO30	1985	Forebody	Cyber-205
$>10^4$	2-D	FEFLO27	1986	Train	Cray-XMP
$>10^5$	3-D	FEFLO72	1989	Train	Cray-2
$>10^6$	3-D	FEFLO74	1991	T-62 Tank	Cray-2
$>10^7$	3-D	FEFLO96	1994	Garage	Cray-M90
$>10^8$	3-D	FEFLO98	1998	Village	SGI Origin 2000

improve accuracy and performance. This implies that ever more realistic simulations can be performed. Table 1.1 summarizes the size of a problem as a function of time from the author's own perspective. Note that in 1983 a problem with more that 1000 finite elements, being run at a university, was considered excessively large!

Although simulations would seem to be more advantageous, the reader should not discount experiments. They provide the only 'reality-check' during the development of new products. However, given the steep decline in computing costs, simulations will certainly reduce the number of required experiments. Boeing estimates indicate that the number of wind-tunnel hours required for the development of the Boeing-747 (1963) was reduced by a factor of 10 for the Boeing-767 (1982) (Rubbert (1988)) and by yet another factor of 10 for the Boeing-777 (1998). Since aerospace is one of the leading fields for simulations, these figures may be indicative of trends to be expected in other manufacturing sectors.

In CFD, the simulation of flows is accomplished by:

(a) solving numerically partial differential equations (PDEs);

(b) following the interaction of a large numbers of particles; or

(c) a combination of both.

The first model is used whenever a continuum assumption for the flow can be made. The second model is used for rarefied flows, where the continuum model is no longer valid. Combinations of fields and particles are used whenever some aspects of a complex problem are best modelled as a continuum, and others by discrete entities, or when the motion of passive marker particles is useful for visualizing flows. Examples where such combinations are commonly employed are plume flows with burning particles and ionized magneto-hydrodynamic flows.

Due to its relevance to the aerospace and defense industries, as well as to most manufacturing processes, CFD has been pursued actively ever since the first digital computers were developed. The Manhattan project was a major testbed and beneficiary of early CFD technology. Concepts such as artificial dissipation date from this time.

CFD, by its very nature, encompasses a variety of disciplines, which have been summarized in Figure 1.1 and may be enumerated in the following order of importance.

(a) *Engineering.* We live in a technology-driven world. Insight for practical engineering purposes is the reason why we pursue CFD. Forget the romantic vision of art for art's sake.

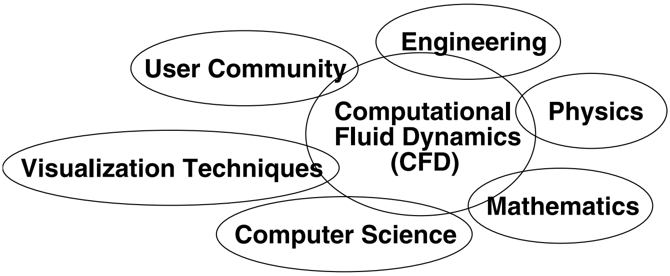

Figure 1.1. The multi-disciplinary nature of CFD

This is engineering, physics, medicine, or any such discipline, and if a CFD code cannot guide the analyst to better products or more understanding, it is simply useless.

(b) *Physics.* Physics explains the phenomena to be simulated for engineering purposes, and provides possible approximations and simplifications to the equations describing the flowfields. For example, the potential approximation, where applicable, represents CPU savings of several orders of magnitude as compared to full Reynolds-averaged Navier–Stokes (RANS) simulations. It is the task of this discipline to outline the domains of validity of the different assumptions and approximations that are possible.

(c) *Mathematics.* Mathematics has three different types of input for CFD applications. These are:

- *classical analysis*, which discusses the nature, boundary conditions, Green kernels, underlying variational principles, adjoint operators, etc., of the PDEs;

- *numerical analysis*, which describes the stability, convergence rates, uniqueness of solutions, well-posedness of numerical schemes, etc.; and

- *discrete mathematics*, which enables the rapid execution of arithmetic operations.

(d) *Computer science.* Computer science has mushroomed into many subdisciplines. The most important ones for CFD are:

- *algorithms*, which describe how to perform certain operations in an optimal way (e.g. the search of items in a list or in space);

- *coding*, so that the final code is portable, easy to modify and/or expand, easy to understand, user-friendly, etc.;

- *software*, which not only encompasses compilers, debuggers and operating systems, but also advanced graphics libraries (e.g. OpenGL); and

- *hardware*, which drives not only the realm of ever-expanding applications that would have been unthinkable a decade ago, but also influences to a large extent the algorithms employed and the way codes are written.

(e) *Visualization techniques.* The vast amounts of data produced by modern simulations need to be displayed in a sensible way. This not only refers to optimal algorithms to filter and traverse the data at hand, but also to ways of seeing this data (plane-cuts, iso-surfaces, X-rays, stereo-vision, etc.).

(f) *User community.* The final product of any CFD effort is a code that is to be used for engineering applications. Successful codes tend to have a user community. This introduces human factors which have to be accounted for: confidence and benchmarking, documentation and education, the individual motivation of the end-users, ego-factors, the not-invented-here syndrome, etc.

1.1. The CFD code

The end-product of any CFD effort is a code that is to be used for engineering applications, or the understanding of physical phenomena that were previously inaccessible. The quality of this tool will depend on the quality of ingredients listed above. Just as a chain is only as strong as its weakest link, a code is only as good as the worst of its ingredients. Given the breadth and variety of disciplines required for a good code, it is not surprising that only a few codes make it to a production environment, although many are written worldwide. Once a CFD code leaves the confines of research, it becomes a *tool*, i.e. a part of the *service industry*. CFD codes, like other tools, can be characterized and compared according to properties considered important by the user community. Some of these are:

- EU: ease of use (problem set-up, user interface, etc.);
- DO: documentation (manuals, help, etc.);
- GF: geometric flexibility;
- TT: turnaround time (set-up to end result);
- BM: benchmarking;
- AC: accuracy;
- SP: speed;
- EX: expandability to new areas/problems.

Like any other product, CFD codes have a customer base. This customer base can be categorized by the number of times a certain application has to be performed. Three main types of end-users may be identified:

(a) those that require a few occasional runs on new configurations to guide them in their designs (e.g. flow simulations in the manufacturing industries and process control);

(b) those that require a large number of runs to optimize highly sophisticated products (e.g. airfoil or wing optimization); and

Table 1.2. Priorities for different user environments

Type of run	No. of runs	Runtime	Desired properties
General purpose/ analysis	$O(1)$	Hours	EU, DO, GF, EX, TT, BM, AC, SP
Design/ optimization	$O(1000)$	Seconds	SP, TT, GF, AC, BM, EU, EX, DO
New physics	$O(10)$	Months	AC, BM, SP, TT, EU, GF, DO, EX

(c) those that require a few very detailed runs on extremely simple geometries in order to understand or discover new physics. These end-users are typically associated with government laboratories. Runs of this kind typically push the limits of tolerance for other users, and their lengths are often the subject of 'war stories' (e.g. more than two weeks of continuous CPU time on the fastest machine available).

According to the frequency of runs, the priorities change, as can be seen from Table 1.2.

The message is clear: before designing or comparing codes, one should ask how often the code is to be used on a particular application, how qualified the personnel are, what the maximum allowed turnaround time is, the expected accuracy, and the resources available. Only then can a proper design or choice of codes be made.

1.2. Porting research codes to an industrial context

Going from a research code to an industrial code requires a major change of focus. Industrial codes are characterized by:

- extensive manuals and other documentation;
- a 24-hour hotline answering service;
- a customer support team for special requests/applications;
- incorporation of changes through releases and training.

In short, they require an *organization* to support them. The CFD software and consulting market already exceeds \$300 million/year, and is expected to grow rapidly in the coming decade.

At present, CFD is being used extensively in many sectors of the manufacturing industry, and is advancing rapidly into new fields as the more complex physical models become available. In fact, the cost advantages of using CFD have become so apparent to industry that in many areas industry has become the driver, demanding usability, extensions and innovation at a rapid pace. Moreover, large integrators are demanding software standards so that the digital product line extends to their tier 1, 2, 3 suppliers.

1.3. Scope of the book

This book treats the different topics and disciplines required to carry out a CFD run in the order they appear or are required during a run:

(a) data structures (to represent, manage, generate and refine a mesh);

(b) grid generation (to create a mesh);

(c) approximation theory and flow solvers (to solve the PDEs, push particles on the mesh);

(d) interpolation (for particle–mesh solvers, and applications requiring remeshing or externally provided boundary conditions);

(e) adaptive mesh refinement (to minimize CPU and memory requirements); and

(f) efficient use of hardware (to minimize CPU requirements).

This order is different from the historical order in which these topics first appeared in CFD, and the order in which most CFD books are written.

Heavy emphasis is placed on CFD using unstructured (i.e. unordered) grids of triangles and tetrahedra. A number of reasons can be given for this emphasis.

- The only successfully industrialized CFD codes that provide user support, updates and an evolving technology to a large user base are based on unstructured grids. This development parallels the development of finite element codes for computational structural dynamics (CSD) in the 1960s.

- Once the problem has been defined for this more general class of grids, reverting to structured grids is a simple matter.

- A large number of very good books on CFD based on structured (and, to a lesser extent, unstructured) grids exist (e.g. Patankar (1980), Book (1981), Roache (1982), Anderson *et al*. (1984), Oran and Boris (1987), Hirsch (1991), Versteeg and Malalasekera (1996), Hoffmann and Chiang (1998), Ferziger and Peric (1999), Toro (1999), Turek (1999), Gresho and Sani (2000), Wesseling (2001), Blazek (2001), Lomax *et al*. (2001), Donea and Huerta (2002)), and there is no point writing yet another one that repeats most of the material.

As with any technological product, the final result is obtained after seemingly traversing a maze of detours. After all, why use a car (which has to be painted after assembly after mining/producing the iron and all other raw materials . . .) to go to the grocery shop when one can walk the half mile? The answer is that we want to do more with a car than drive half a mile. The same is true for CFD. If the requirement consists of a few simulations of flows past simple geometries, then all this development is not needed. To go the distance to realistic 3-D simulations of flows in or past complex geometries, no other way will do. The reader is therefore asked to be patient. The relevance of some parts will only become apparent in subsequent chapters.