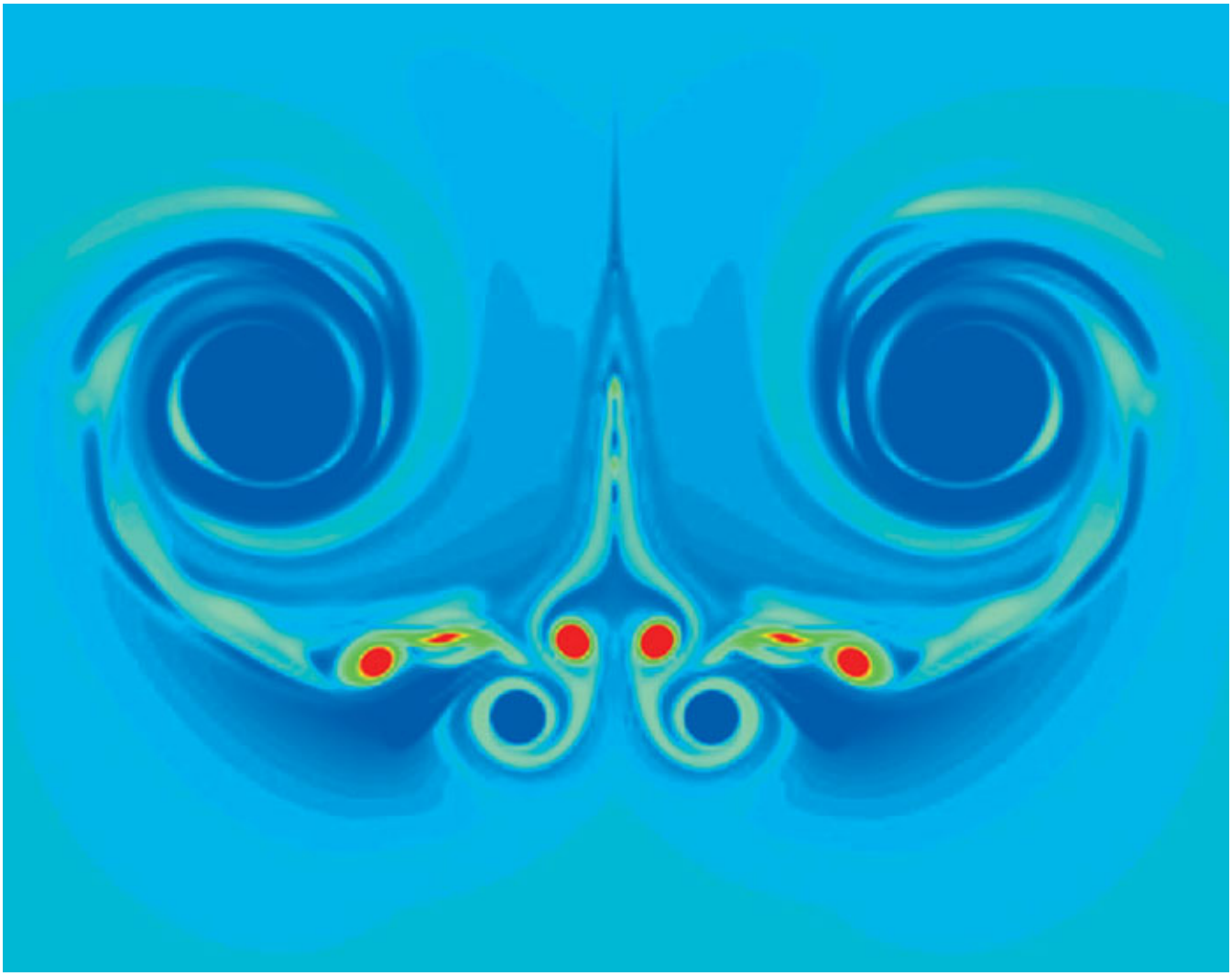
UNDERSTANDING, DESIGN, ACTION: THE THREEFOLD PURPOSE OF SIMULATION

As a scientific activity, simulation may seem, at first blush, to afford equal amounts of promise and of risk. While they should remain aware of its limitations, research workers and engineers do find in it an irreplaceable instrument to overcome the shortcomings or impossibilities inherent in experimentation.



Onera

Numerical simulation of vortices in the wake of an airliner. In aerodynamics, planemakers have been investigating for a long time, in wind tunnels, the airflows around scale models.

To simulate, according to the dictionary, is *"to proffer as real what is not so, by imitating the appearance of the real thing in which one seeks to make believe."* Simulation, in common parlance, is indeed cognate to sham, disguise, trickery, play-acting. It is set apart from mimicry in that it is neither pure copy, nor mere reproduction. As a rule, it consists in imitation of an original, with regard to which it always introduces variations. Hence, simulation is ever twofold, being both play and exertion, deceit and feat of virtuosity, artifice and art, dissimulation and training. On the one hand, it may be seen as a peddler of illusions. And on the other, as warranty of attainment and further conquest.

Incremental reality, or less than real?

A number of these aspects may be seen in the way the sciences make use of simulation. It harks back, in this context also, to the notion of imitation. It consists, indeed, in an imitation "to the square" of nature, in that it is not nature itself it reproduces, but the

PhotoDisc

Simulation, among other applications, allows the optimum configuration for a road network to be determined.

model by which we have first elected to represent the latter. It thus displays an "incremental reality," inasmuch as it makes explicit what, in a model or theory, is only implicit or submerged. Good simulation is akin to revelation.

However, it does also happen that, poorly carried out or wrongly interpreted, it may only disclose, on the contrary, some illusory sub- or infra-reality, or outrageously caricature what it is supposed to portray. It may, for instance, exceed the boundaries of a model, go outside its domain of validity, miss an essential process, without any warning sign being given out instantly. Bad simulation is akin to *trompe-l'oeil* and counterfeit.

Since the model it is based on is always incomplete, modeling can never of itself claim to be complete. What it shows is, at best, faithful fragments, albeit purified and simplified, of reality. Other aspects of that same reality will be left in the shadows. Despite what the words stand for, simulation thus always goes hand in hand with dissimulation. Is this a serious issue? No, since nothing is lost, so long as what it conceals or passes over is not relevant in the context of the problem at hand. Indeed, what is a good model, if not that which encompasses the minimum number of ingredients required to exhibit universality, with regard to what one seeks to describe, or to understand?

Generally speaking, for a scientist, to simulate is to carry out "experiments on a model": he sets up an artificial reproduction of the phenomenon he wishes to investigate, and then observes its behavior as he varies the actions that may be put to work on it, deriving from this, inductively, what would actually happen, under the influence of similar actions. In practice, this methodology comes in a number of variants, each presenting varying advantages. Let us survey some of them.

Entering the "experimental no man's lands"

Simulation has, first of all, a heuristic dimension. It shows what abstract formalisms do not make directly apparent, fleshes out equations, clothes theories, makes concepts speak out, yields variations on their concrete applications (often in highly esthetic fashion). Such ability makes it, at times, indispensable. This is the case when the equations are too complex. And equally so when predictions from a theory cannot be confronted with direct experiment, owing to ethical considerations, time constraints, budgetary restrictions or natural obstacles. Thus, simulation makes it possible to describe what goes on in the core of a star, or in a collision between galaxies, to reconstruct climate prevailing a hundred thousand years ago, extrapolate certain physical processes over the very long term, determine the optimum configuration for a road network, predict the impact of various economic measures on consumption or saving. When confronting an "experimental no man's land," simulation becomes an instrument that is irreplaceable. Doing good simulation is, ultimately, to avail oneself of the means for improved understanding, better design, and, most crucially, enhanced action.

Simulation is also decisive in situations where no sound theoretical foundation is available, and what is sought is precisely to work out a theory to account for the observational data. One is thus able to specify precisely the concrete consequences of the various possible theoretical models, determine which one yields the most reliable approximation, and then be in a position to understand, or even optimize, certain processes.

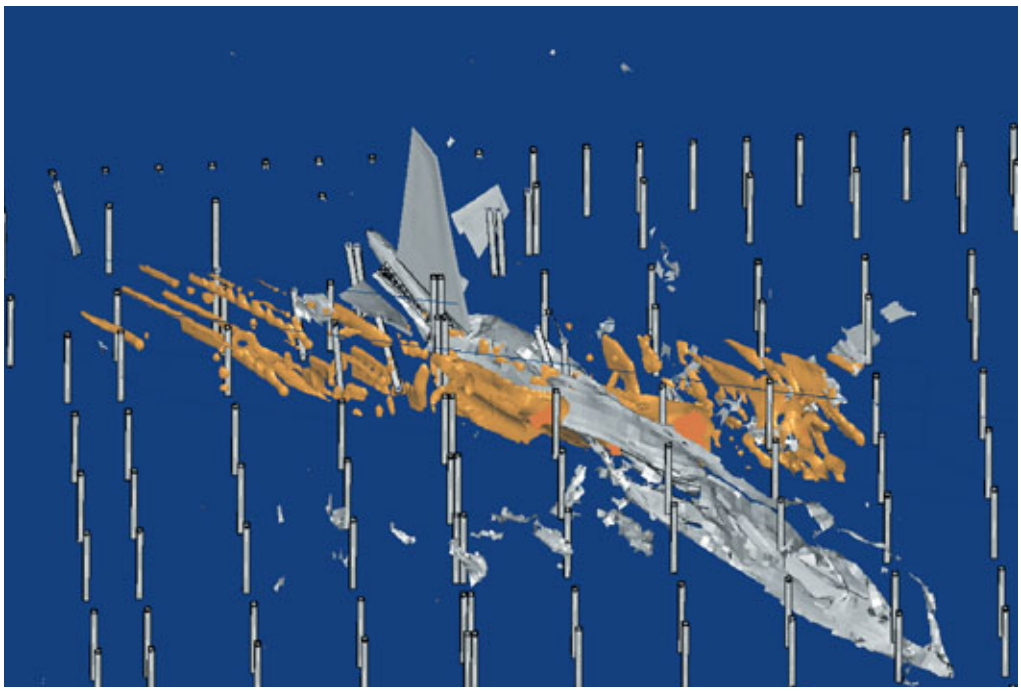
Simulation is not solely numerical. It may be "analog" One may, for example, use scaling laws involving dimensionless numbers to carry out small-scale experiments, in order to account for phenomena which, in nature, occur on large scales or over great time spans. This is how convection in the Earth's mantle is simulated, by investigating fluid motions within containers filled with a heated viscous liquid. In like manner, one may investigate fluid circulation in a scale model for climate prediction. In the field of aerodynamics, it is possible to sketch out, in a wind tunnel, airflows around an aircraft model. Such analog simulations must be complemented with numerical simulations, which serve to lower their cost, and the numbers required.

Finally, simulation makes it possible to get a system (a robot, an airplane, a nuclear power station) to operate virtually, by way of a computer program. The principle is simple: if the program has properly integrated all behavior modes of the various subsystems, then the operator can act "as if" he was working on the actual system. That is the concept on which the most conventional simulators (driving, flight, traffic, piloting…) are based. Their assigned function is the training of personnel, assisting decision making or action in a real environment, or even predicting a system's behavior in an environment not accessible to humans.

Between theory and experiment

As may be seen, simulation presents a multiplicity of aspects. But what of its epistemological status? Neither that of theory, nor yet of experiment. Indeed, simulation has unquestionably brought in a new relationship, as regards our representation of the

A good simulation may yield more information than an overall experiment that might be inadequately prepared or hard to reproduce, if only for reasons of costs. Top, simulation of the destruction of the Boeing 757 that crashed into the Pentagon buildings during the attacks of 11 September 2001. Carried out on the basis of physical laws by research workers at Purdue University and civil engineers, this reconstruction shows how the steel-reinforced concrete pillars shredded the plane to fragments, in particular its wings, and how burning kerosene spread (shown in orange). Bottom, this deliberate-crash test of a Boeing 720, carried out on 1 December 1984 by NASA at Edwards Air Force Base (California), had the purpose, in particular, of testing an additive intended to restrict kerosene flammability. This fell far short of yielding all the hoped-for results.



School of Civil Engineering and Department of Computer Sciences/Purdue University

Nasa

world. Right into the 1950s, theory and experiment were presented as dialectically correlated activities (Gaston Bachelard, Karl Popper). The idea was that science was born out of an unceasing to-and-fro between the one and the other. This context has changed with the arrival, en masse, of computers and numerical calculus, to such an extent that new questions are now being posed: could simulations have the same status as experiments? Will they achieve the capacity, in some cases, of falsifying a theory? Do they invariably enable a better understanding of the models they illustrate? As for the positive effect of simulation, which has to do with savings in terms of time and money, or indeed to the access it allows to hostile or dangerous realities through the contribution of virtual realities, should this not be weighed against the risk of shying away from reality and concrete experimentation? It does remain that simulation has gradually altered the function assigned to reality in the sciences. Previously, scientist would refer to an actuality which they deemed reliably secure, a reality that was unassailable, authentic, primordial, no fiction. Nowadays, it is the very nature of that reality which is undergoing a sea-change, through the generalized utilization of simulations, modeling and scenarios. Of old, reality stood as either the object, or the obstacle, the criterion, or the test of science. It now enfolds itself in the virtual, to such an extent that its outlines are no longer fixed.

Simulating science is concerned with the possibles, and not merely what is.

Étienne Klein
Physical Sciences Division
CEA Saclay Center

SIMULATION AS METHODOLOGY FOR R&D

Scientific research, which traditionally is organized according to a dialectic of theory and experiment, has gradually become structured into a triptych, comprising theory and modeling, numerical simulation, and experimental verification. This methodology of simulation, which has reached varying degrees of maturity according to the sectors involved, is now achieving general acceptance.

One of the ways of advancing scientific knowledge is to put forward rational explanations for natural phenomena, through the construction of **models**. Such mental schemas, simplified with regard to reality, are suggested by experimental observations, and in turn give rise to experiments. A model is thus an abstraction from reality, which, as ideas mature, may gradually be refined (from Newton's theory of falling bodies to Einstein's theory of general relativity, for example). It is expressed by means of mathematical equations. In the simpler cases, the equations may be solved by analytical methods. In general, that is not possible. To establish a model's degree of relevance with regard to experimental data, it is often necessary to carry out computer resolution of its equations

There is thus an inrush of **numerical simulation**, into the dialectic of theory and experiment at the heart of the scientific approach (see Box A, ***What is a numerical simulation?***). Numerical simulation is thus gradually pervading the realm of knowledge, and multiplying research workers' powers of investigation. They now have the ability, for example, to simulate the collision of two galaxies, to gain a better understanding of some of the structures observed through giant telescopes.

An unordered triptych

Scientific research then accords to the following cycle: theory and modeling, numerical simulation, and experimental verification. This is what we refer to, in the present paper, as the methodology of simulation. In concrete terms, knowledge of the physical process, or of the ensemble of processes, investigated is brought together into a software program enabling the resolution of the equations in the models established at a prior stage. How is that program validated? By simulating, precisely, the process thought to give rise to the observed phenomenon. If the numerical and experimental findings are in close agreement, the model thus translated into software is recognized as valid, to the chosen degree of approximation. Such validation can often be complex, since many processes are involved. It then is necessary to proceed step by step, by verifying that each and every elementary model in the overall description is satisfactory. This is *partial* validation. Validation is then required for the whole, which entails taking on board the interactions between elementary models. This is *overall* validation.

In climate modeling, for instance, research workers are involved in their separate ways in work on oceans, the atmosphere, and the biosphere, each of these major components comprising many models. They then verify the whole, by simulating a period of climate from the past, which simulation may be validated through analysis of ice core samples.

The great benefit of this methodology is the way it brings a capacity for detailed exploration of phenomena which simply may not be reproduced in the laboratory, either because that is not possible (a galaxy!), or because it would be utterly unthinkable (a chemical or nuclear hazard, for instance).

An approach transferred to industry

This rapid maturing of the simulation approach has led to its being used in technological research and industrial development. It is indeed very tempting to go in for simulation in these areas of activity. The reasons for this are simple. First, with simulation, there is a greater margin for concept optimization. While the number of actual experimental mockups that may be made is limited, one can put in as many variations as desired into numerical simulations. Recourse to experiment being reduced, development is less costly and faster, and products are brought more quickly to market. All this of course assumes that the simulation software program is available; indeed, it is in developing it that part of the cost is incurred. Design of an object thus involves simulation of its behavior, optimization of its details, and verification of its good functioning, through a limited number of experiments.

Is that the end there is to it? Strictly speaking, it is not possible to extract more infor-

What is a numerical simulation?

A

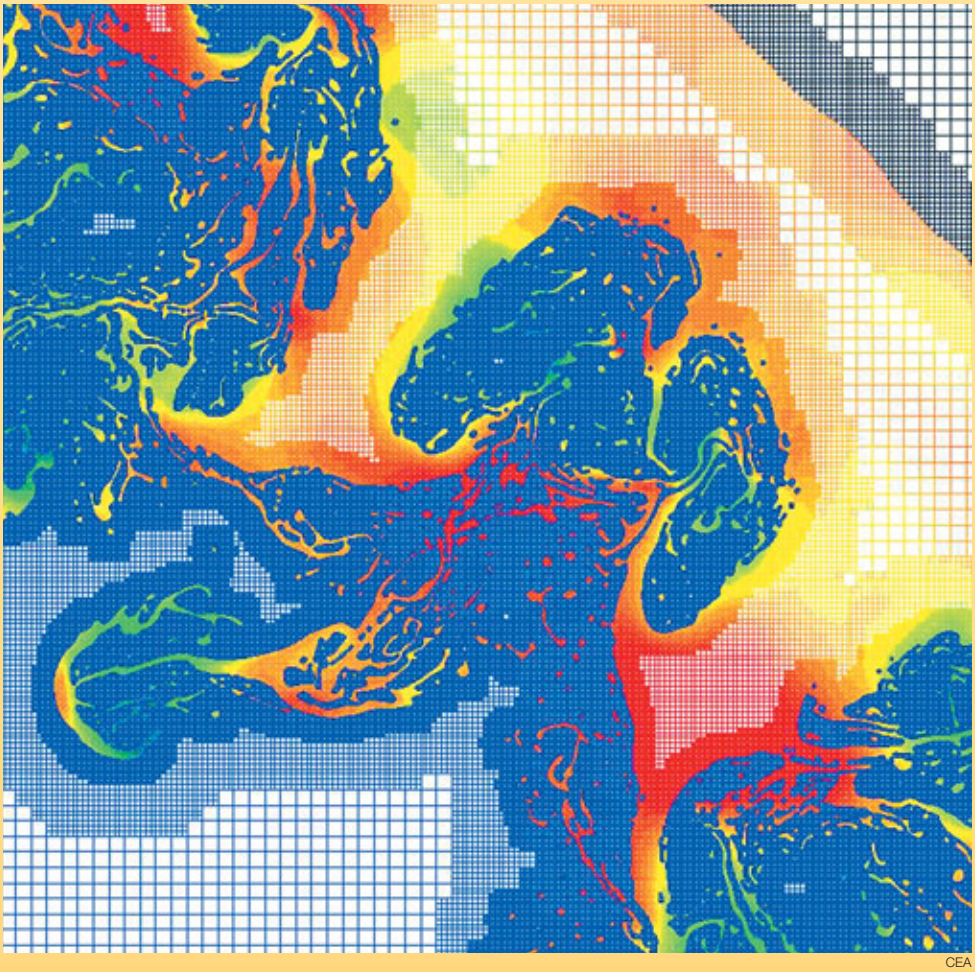
Numerical simulation consists in reproducing, through computation, a system's operation, described at a prior stage by an ensemble of **models**. It relies on specific mathematical and computational methods. The main stages involved in carrying out an investigation by means of numerical simulation are practices common to many sectors of research and industry, in particular nuclear engineering, aerospace or automotive.

At every point of the "object" considered, a number of physical quantities (velocity, temperature…) describe the state and evolution of the system being investigated. These are not independent, being linked and governed by **equations**, generally **partial differential** equations. These equations are the expression in mathematical terms of the physical laws modeling the object's behavior. Simulating the latter's state is to determine – at every point, ideally – the numerical values for its parameters. As there is an infinite number of points, and thus an infinite number of values to be calculated, this goal is unattainable (except in some very special cases, where the initial equations may be solved by analytical formulae). A natural approximation hence consists in considering only a finite number of points. The parameter values to be computed are thus finite in number, and the operations required become manageable, thanks to the computer. The actual number of points processed will depend, of course, on computational power: the greater the number, the better the object's description will ultimately be. The basis of parameter computation, as of numerical simulation, is thus the reduction of the infinite to the finite: **discretization**.

How exactly does one operate, starting from the model's mathematical equations? Two methods are very commonly used, being representative, respectively, of **deterministic computation** methods, resolving the equations governing the processes investigated after discretization of the variables, and methods of **statistical** or **probabilistic calculus**.

The principle of the former, known as the **finite-volume method**, dates from before the time of computer utilization. Each of the object's points is simply assimilated to a small elementary volume (a cube, for instance), hence the *finite-volume* tag. Plasma is thus considered as a set or lattice of contiguous volumes, which, by analogy to the makeup of netting, will be referred to as a **mesh**. The parameters for the object's state are now defined in each mesh cell. For each one of these, by reformulating the model's mathematical equations in terms of volume averages, it will then be possible to build up *algebraic relations* between the parameters for one cell and those of its neighbors. In total, there will be as many relations as there are unknown parameters, and it will be up to the computer to resolve the *system* of relations obtained. For that purpose, it will be necessary to turn to the techniques of **numerical analysis**, and to program specific **algorithms**.

The rising power of computers has allowed an increasing fineness of discretization, making it possible to go from a few tens of cells in the 1960s to several tens of thousands in the 1980s, through to millions in the 1990s, and up to some ten billion cells nowadays (Tera machine at CEA's Military Applications Division), a figure that should increase tenfold by the end of the decade.

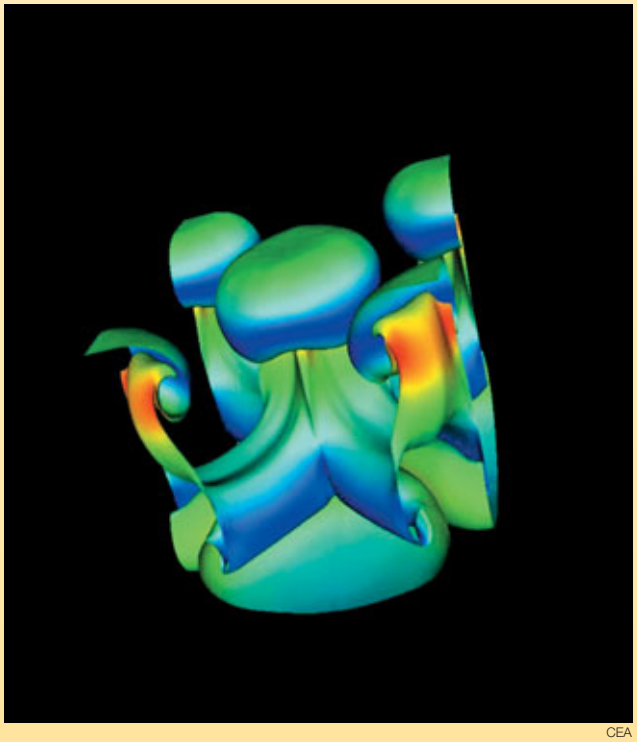


CEA

Example of an image from a 2D simulation of instabilities, carried out with CEA's Tera supercomputer. Computation involved adaptive meshing, featuring finer resolution in the areas where processes are at their most complex.

A

A refinement of meshing, **adaptive remeshing**, consists in adjusting cell size according to conditions, for example by making them smaller and more densely packed at the interfaces between two environments, where physical processes are most complex, or where variations are greatest.
The finite-volume method can be applied to highly diverse physical and mathematical situations. It allows any shape of mesh cell (cube, hexahedron, tetrahedron…), and the mesh may be altered in the course of computation, according to geometric or physical criteria. Finally, it is easy to implement in the context of **parallel computers** (see Box B, ***Computational resources for high-performance numerical computation***), as the mesh may be subjected to partitioning for the purposes of computation on this type of machine (example: Figure B).
Also included in this same group are the **finite-difference method**, a special case of the finite-volume method where cell walls are orthogonal, and the **finite-element method**, where a variety of cell types may be juxtaposed.
The second major method, the so-called **Monte Carlo** method, is particularly suited to the simulation of *particle transport*, for example of neutrons or photons in a **plasma** (see *Simulations in particle physics*). This kind of transport is in fact characterized by a succession of stages, where each particle may be subject to a variety of events (diffusion, absorption, emission…) that are possible *a priori*. Elementary probabilities for each of these events are known individually, for each particle.
It is then a natural move to assimilate a point in the plasma to a particle. A set of particles, finite in number, will form a representative sample of the infinity of particles in the plasma, as for a statistical survey. From one stage to the next, the sample's evolution will be determined by random draws (hence the method's name). The effectiveness of the method, implemented in Los Alamos as early as the 1940s, is of course dependent on the statistical quality of the random draws. There are, for just this purpose, *random-number* methods available, well suited to computer processing.



CEA

3D simulation carried out with the Tera supercomputer, set up at the end of 2001 at CEA's DAM-Île de France Center, at Bruyères-le-Châtel (Essonne* département*).

Finite-volume and Monte Carlo methods have been, and still are, the occasion for many mathematical investigations. These studies are devoted, in particular, to narrowing down these methods' convergence, i.e. the manner in which approximation precision varies with cell or particle number. This issue arises naturally, when confronting results from numerical simulation to experimental findings.

How does a numerical simulation proceed?

Reference is often made to *numerical experiments*, to emphasize the analogy between performing a numerical simulation and carrying out a physical experiment.
In short, the latter makes use of an experimental setup, configured in accordance with initial conditions (for temperature, pressure…) and control parameters (duration of the experiment, of measurements…). In the course of the experiment, the setup yields measurement points, which are recorded. These records are then analyzed and interpreted.
In a numerical simulation, the experimental setup consists in an ensemble of computational programs, run on computers. The **computation codes**, or **software** programs, are the expression, via numerical algorithms, of the mathematical formulations of the physical models being investigated. Prior to computation, and subsequent to it, *environment software* programs manage a number of complex operations for the preparation of computations and analysis of the results.
The initial data for the simulation will comprise, first of all, the delineation of the computation domain – on the basis of an approximate representation of the geometric shapes (produced by means of drafting and CAD [computer-assisted design] software) –, followed by discretization of this computation domain over a mesh, as well as the values for the physical parameters over that mesh, and the control parameters to ensure proper running of the programs… All these data (produced and managed by the environment software programs) will be taken up and verified by the codes. The actual results from the computations, i.e. the numerical values for the physical parameters, will be saved on the fly. In fact, a specific protocol will structure the computer-generated information, to form it into a numerical database.
A complete protocol organizes the electronic exchange of required information (dimensions, in particular) in accordance with predefined formats: modeler,(1) mesher,(2) mesh partitioner, com-

(1) The modeler is a tool enabling the generation and manipulation of points, curves and surfaces, for the purposes, for example, of mesh generation.
(2) The geometric shapes of a mesh are described by sets of points connected by curves and surfaces (Bézier curves and surfaces, for instance), representing its boundaries.

putation codes, visualization and analysis software programs. *Sensitivity* studies regarding the results (sensitivity to meshes and models) form part of the numerical "experiments."
On completion of computation (numerical resolution of the equations describing the physical processes occurring in each cell), analysis of the results by specialists will rely on use of the numerical database. This will involve a number of stages: selective extraction of data (according to the physical parameter of interest) and visualization, and data extraction and transfer for the purposes of computing and visualizing diagnostics.
This parallel between performing a computation case for a numerical experiment and carrying out a physical experiment does not end there: the numerical results will be compared to the experimental findings. This comparative analysis, carried out on the basis of standardized quantitative criteria, will make demands on both the experience and skill of engineers, physicists, and mathematicians. Its will result in further improvements to physical models and simulation software programs.

Bruno Scheurer
Military Applications Division
CEA DAM-Ile de France Center

Frederic Ducros and **Ulrich Bieder**
Nuclear Energy Division
CEA Grenoble Center

The example of a thermalhydraulics computation

Implementation of a numerical simulation protocol may be illustrated by the work carried out by the team developing the **thermalhydraulics** computation software Trio U. This work was carried out in the context of a study conducted in collaboration with the French Radiological Protection and Nuclear Safety Institute (IRSN: Institut de radioprotection et de sûreté nucléaire). The aim was to obtain very accurate data to provide engineers with wall heat-stress values for the components of a pressurized-water reactor in case of a major accident involving turbulent natural circulation of hot gases. This investigation requires simultaneous modeling of large-scale "system" effects and of small-scale **turbulent** processes (see Box F, ***Modeling and simulation of turbulent flows***).
This begins with specification of the overall computation model (Figure A), followed by production of the CAD model and corresponding mesh with commercial software programs (Figure B). Meshes of over five million cells require use of powerful graphics stations. In this example, the mesh for a steam generator (Figures C and D) has been partitioned to parcel out computation over eight processors on one of CEA's parallel computers: each color stands for a zone assigned to a specific processor. The computations, whose boundary conditions are provided by way of a "system" computation (Icare–Cathare), yield results which it is up to the specialists to interpret. In this case, visualization on graphics stations of the instantaneous values of the velocity field show the impact of a hot plume on the steam generator's tube-plate (section of the velocity field, at left on Figure E), and instantaneous temperature in the water box (at right).

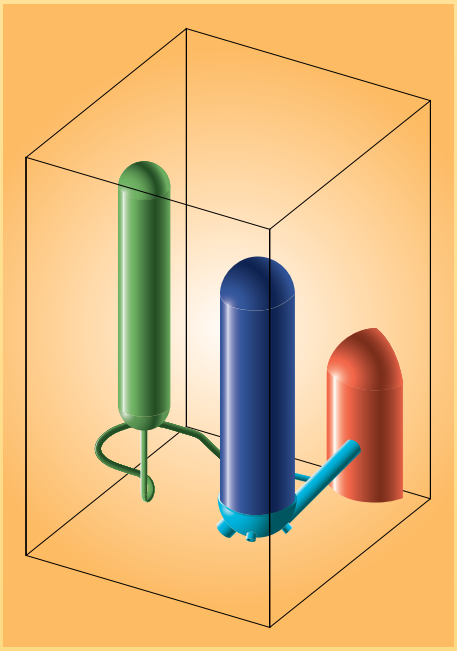


Figure A. Overall computation domain, including part of the reactor vessel (shown in red), the outlet pipe (hot leg, in light blue), steam generator (dark blue), and pressurizer (green).

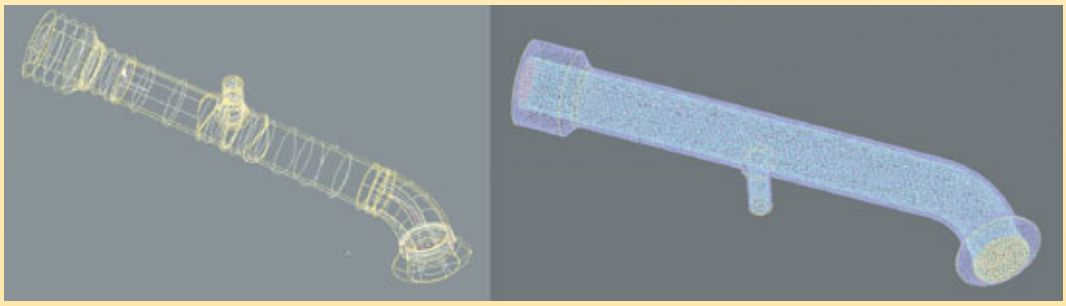
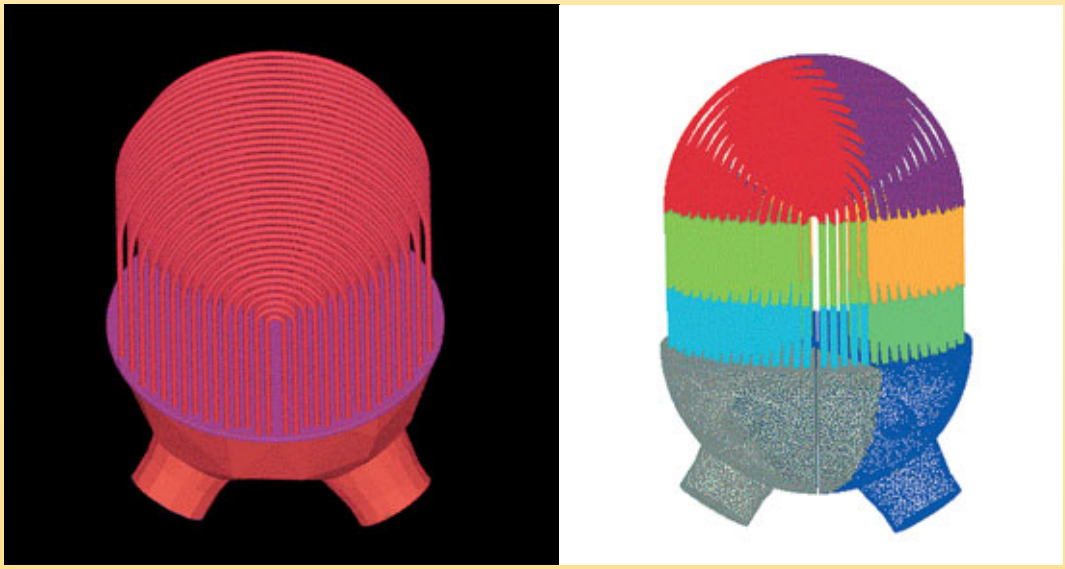


Figure B. CAD model of the hot leg of the reactor vessel outlet (left) and unstructured mesh for it (right).



Figures C and D.

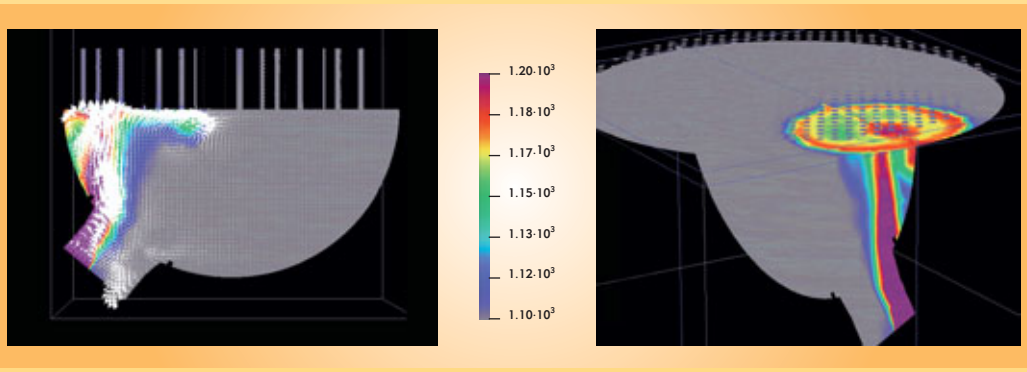
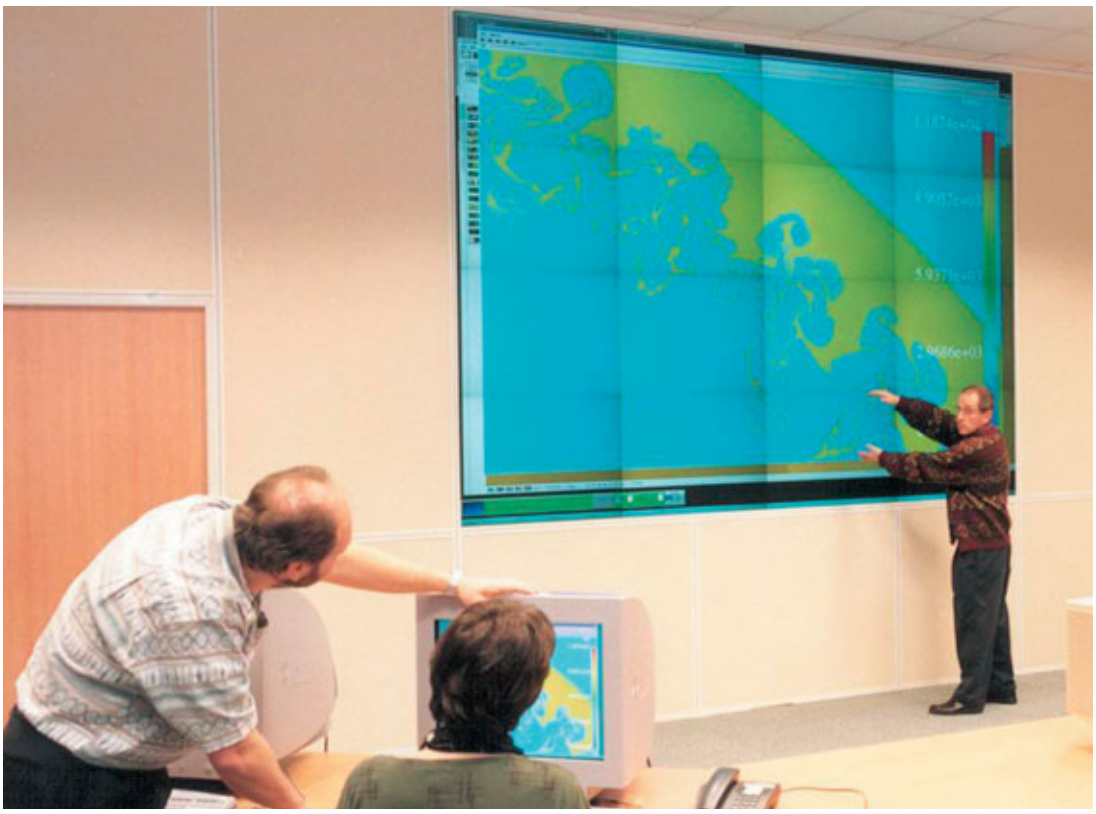


Figure E.



The image wall, recently installed at the Ile de France Center of CEA's Military Applications Division, sited at Bruyères-le Châtel. allows research workers to visualize, thanks to its 3,200 × 2,400-pixel definition, very fine details in overall simulations performed by a teraflops computer, such as the current Tera model.

CEA/DAM

mation from a model than was first put into it. What is novel with numerical simulation tools is that the consequences predicted by simulation of complex systems far exceed, in terms of sheer number, the scenarios that can be foreseen *a priori* by the research worker. In a way, simulation turns into a conceptual guide to identify new avenues of research, or to select the most relevant ones. This of course does not involve extrapolation, but rather an analysis of the possible, in a vast parameter space. Improvement of the models is thus fed by their exploration at the hands of the research worker. From this will follow the highlighting of further requirements, and more precise experiments.

The economic prime mover

The industrial returns from technological R&D are also a powerful prime mover of improvement. To secure gains of a few fractions of a (euro) cent on the price of a kilowatt-hour, for example, represents a major stake for the nuclear power sector. How is this to be achieved? For instance, in the design of nuclear power stations, margins are built in to ensure their safety; these in turn depend on our knowledge of the physical processes involved. Better knowledge of the underlying processes will enable optimization of the design, while preserving, or enhancing, safety levels.

There is thus a requirement to develop more accurate models, carry out detailed numerical simulations of them, and then validate them. This entails both more powerful computers, and validation experiments using more accurate and sensitive instrumentation than heretofore, but the potential benefit from this is a major one. This outlook is now prevalent in all sectors of industry, with varying levels of maturity.

Simulation software programs, once validated by an ensemble of experiments, are thus tremendous instruments: analytical tools for the research worker, enabling him to identify, in highly complex systems, the more important processes, and optimization tools for the engineer. The programs are not frozen: they are constantly being enhanced with more accurate models, made possible both by scientific advances and access to more powerful computers – and, of course, by rigorous validation, involving ever more sophisticated instrumentation.

Didier Besnard
Director, Simulation Program
Military Applications Division
CEA DAM-Ile de France Center

Computational resources for high-performance numerical simulation

B

Carrying out more accurate **numerical simulations** requires the use of more complex physical and numerical **models** applied to more detailed descriptions of the simulated objects (see Box A, ***What is a numerical simulation?***). All this requires advances in the area of simulation software but also a considerable increase in the capacity of the computer systems on which the software runs.

Scalar and vector processors

The key element of the computer is the processor, which is the basic unit that executes a program to carry out a computation. There are two main types of processors, **scalar processors** and **vector processors**. The former type carries out operations on elementary (scalar) numbers, for instance the addition of two numbers. The second type carries out operations on arrays of numbers (vectors), for example adding elementwise the numbers belonging to two sets of 500 elements. For this reason, they are particularly well suited to numerical simulation: when executing an operation of this type, a vector processor can operate at a rate close to its maximum (peak) performance. The same operation with a scalar processor requires many independent operations (operating one vector element at a time) executed at a rate well below its peak rate. The main advantage of scalar processors is their price: these are general-purpose microprocessors whose design and production costs can be written-down across broad markets.

Strengths and constraints of parallelism

Recent computers allow high performances partly by using a higher operating frequency, partly by trying to carry out several operations simultaneously: this is a first level of **parallelism**. The speeding up in frequency is bounded by developments in microelectronics technology, whereas interdependency between the instructions to be carried out by the processor limits the amount of parallelism that is possible. Simultaneous use of several processors is a second level of parallelism allowing better performance, provided programs able to take advantage of this are available. Whereas parallelism at processor level is automatic, parallelism *between processors* in a parallel computer must be taken into account by the programmer, who has to split his program into independent parts and make provisions for the necessary communication between them. Often, this is done by partitioning the domain on which the computation is done. Each processor simulates the behavior of one domain and regular communications between processors ensure consistency for the overall computation. To achieve an efficient parallel program, a balanced share of the workload must be ensured among the individual processors and efforts must be made to limit communications costs.

The various architectures

A variety of equipment types are used for numerical simulation. From their desktop computer where they prepare computations and analyze the results, users access shared computation, storage and visualization resources far more powerful than their own. All of these machines are connected by networks, enabling information to circulate between them at rates compatible with the volume of data produced, which can be as much as 1 **terabyte** (1 TB = 10^{12} bytes) of data for one single simulation.

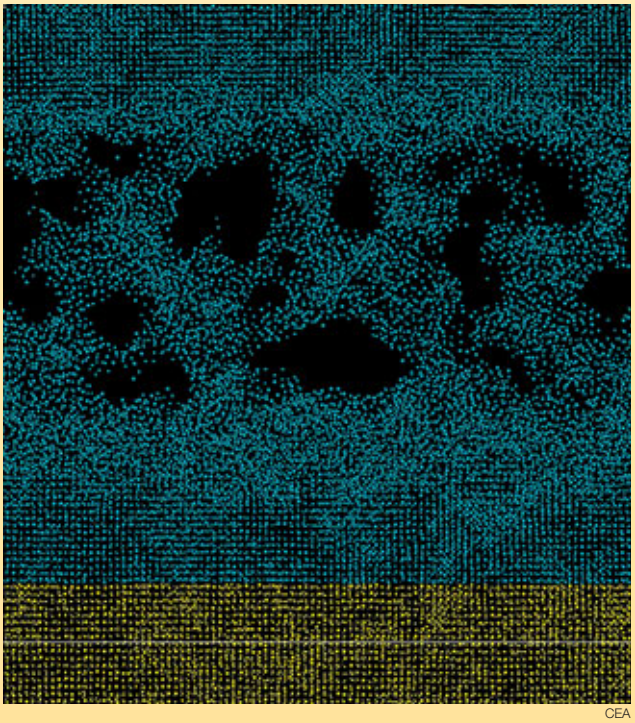
The most powerful computers are generally referred to as **supercomputers**. They currently attain capabilities counted in **teraflops** (1 Tflops = 10^{12} floating-point operations per second).

Currently, there are three main types of supercomputers: vector supercomputers, clusters of mini-computers with shared memory, and clusters of PCs (standard home computers). The choice between these architectures largely depends on the intended applications and uses. Vector supercomputers have very-high-performance processors but it is difficult to increase their computing performance by adding processors. PC clusters are inexpensive but poorly suited to environments where many users perform numerous large-scale computations (in terms of memory and input/output).

It is mainly for these reasons that CEA's Military Applications Division (DAM) has choosen for its Simulation Program (see *The Simulation Program: weapons assurance without nuclear testing*) architectures of the shared-memory mini-computer cluster type, also known as **clusters of SMPs** (symmetric multiprocessing). Such a system uses as a basic building block a mini-computer featuring several microprocessors sharing a common memory (see Figure). As these mini-computers are in widespread use in a variety of fields, ranging from banks to web servers through design offices, they offer an excellent performance/price ratio. These basic "blocks" (also known as *nodes*) are connected by a high-per-

Installed at CEA (DAM-Ile de France Center) in December 2001, the TERA machine designed by Compaq (now HP) has for its basic element a mini-computer with 4 x 1-GHz processors sharing 4 GB of memory and giving a total performance of 8 Gflops. These basic elements are interconnected through a fast network designed by Quadrics Ltd. A synchronization operation across all 2,560 processors is completed in under 25 microseconds. The overall file system offers 50 terabytes of storage space for input/output with an aggregate bandwidth of 7.5 GB/s.

Computational resources for high-performance numerical simulation (cont'd)



CEA

Parallel computers are well suited to numerical methods based on meshing (see Box A, What is a numerical simulation?) but equally to processing ab-initio calculations such as this molecular-dynamics simulation of impact damage to two copper plates moving at 1 km/s (see Simulation of materials*). The system under consideration includes 100,000 atoms of copper representing a square-section (0.02 µm square) parallelogram of normal density. The atoms interact in accordance with an embedded atom potential over approximately 4–6 picoseconds. The calculation, performed on 18 processors of the Tera supercomputer at Bruyères-le-Châtel using the CEA-developed Stamp software, accounted for some ten minutes of "user" time (calculation carried out by B. Magne). Tests involving up to 64 million atoms have been carried out, requiring 256 processors over some one hundred hours.*

formance network: the cumulated power of several hundreds of these "blocks" can reach several Tflops. One then speaks of a **massively parallel computer**.

Such power can be made available for one single parallel application using all the supercomputer's resources, but also for many independent applications, whether parallel or not, each using part of the resources.

While the characteristic emphasized to describe a supercomputer is usually its computational power, the input/output aspect should not be ignored. These machines, capable of running large-scale simulations, must have storage systems with suitable capacities and performance. In clusters of SMPs, each mini-computer has a local disk space. However, it is not advisable to use this space for the user files because it would require the user to move explicitly his data between each distinct stage of his calculation. For this reason, it is important to have disk space accessible by all of the mini-computers making up the supercomputer. This space generally consists in sets of disk drives connected to nodes whose main function is to manage them. Just as for computation, parallelism of input/output allows high performance to be obtained. For such purposes, parallel overall file systems must be implemented, enabling rapid and unrestricted access to the shared disk space.

While they offer considerable computational power, clusters of SMPs nevertheless pose a number of challenges. Among the most important, in addition to programming simulation software capable of using efficiently a large number of processors, is the development of operating systems and associated software tools compatible with such configurations, and fault-tolerant.

François Robin
Military Applications Division
CEA, DAM–Ile de France Center

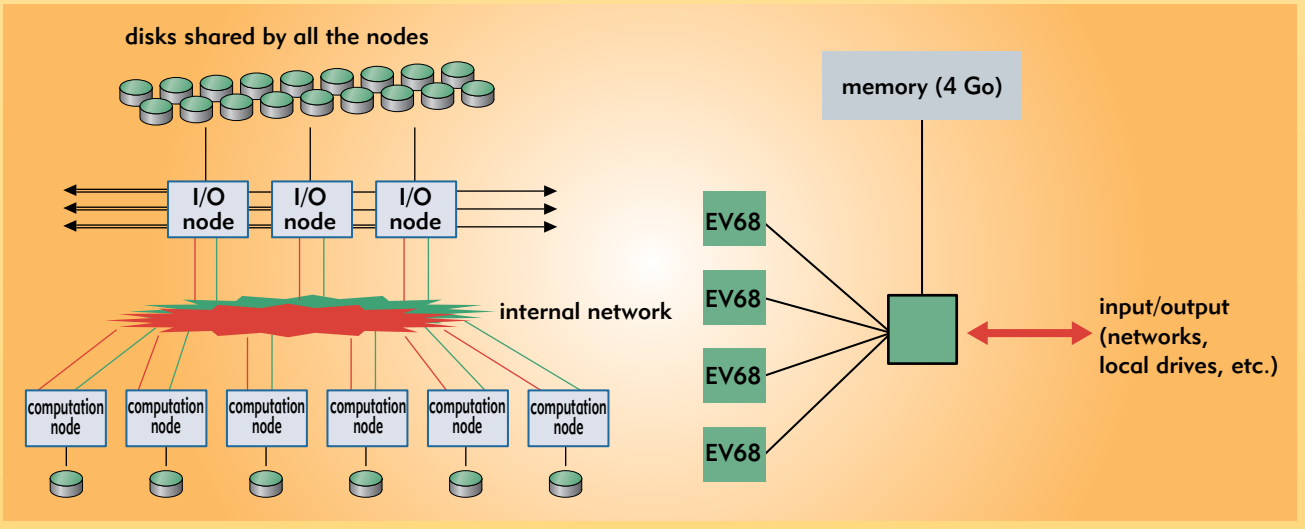


Figure. Architecture of an "SMP-cluster" type machine. At left, the general architecture (I/O = input/output), on the right, that of a node with four Alpha EV68 processors, clocked at 1 GHz.

Modeling and simulation of turbulent flows F

Turbulence, or disturbance in so-called turbulent flow, develops in most of the flows that condition our immediate environment (rivers, ocean, atmosphere). It also turns out to be one, if not the, dimensioning parameter in a large number of industrial flows (related to energy generation or conversion, aerodynamics, etc.). Thus, it is not surprising that a drive is being launched to achieve prediction for the process – albeit in approximate fashion as yet – especially when it combines with complicating processes (stratification, combustion, presence of several phases, etc.). This is because, paraxodically, even though it is possible to predict the turbulent nature of a flow and even, from a theoretical standpoint, to highlight certain common – and apparently universal – characteristics of turbulent flows,(1) their prediction, in specific cases, remains tricky. Indeed, it must take into account the considerable range of space and time scales(2) involved in any flow of this type.

Researchers, however, are not without resources, nowadays, when approaching this problem. First, the equations governing the evolution of turbulent flows over space and time (Navier–Stokes equations(3)) are known. Their complete solution, in highly favorable cases, has led to predictive descriptions. However, systematic use of this method of resolution comes up against two major difficulties: on the one hand, it would require complete, simultaneous knowledge of all variables attached to the flow, and of the forced-flow conditions imposed on it,(4) and, on the other hand, it would mobilize computational resources that will remain unrealistic for decades yet.

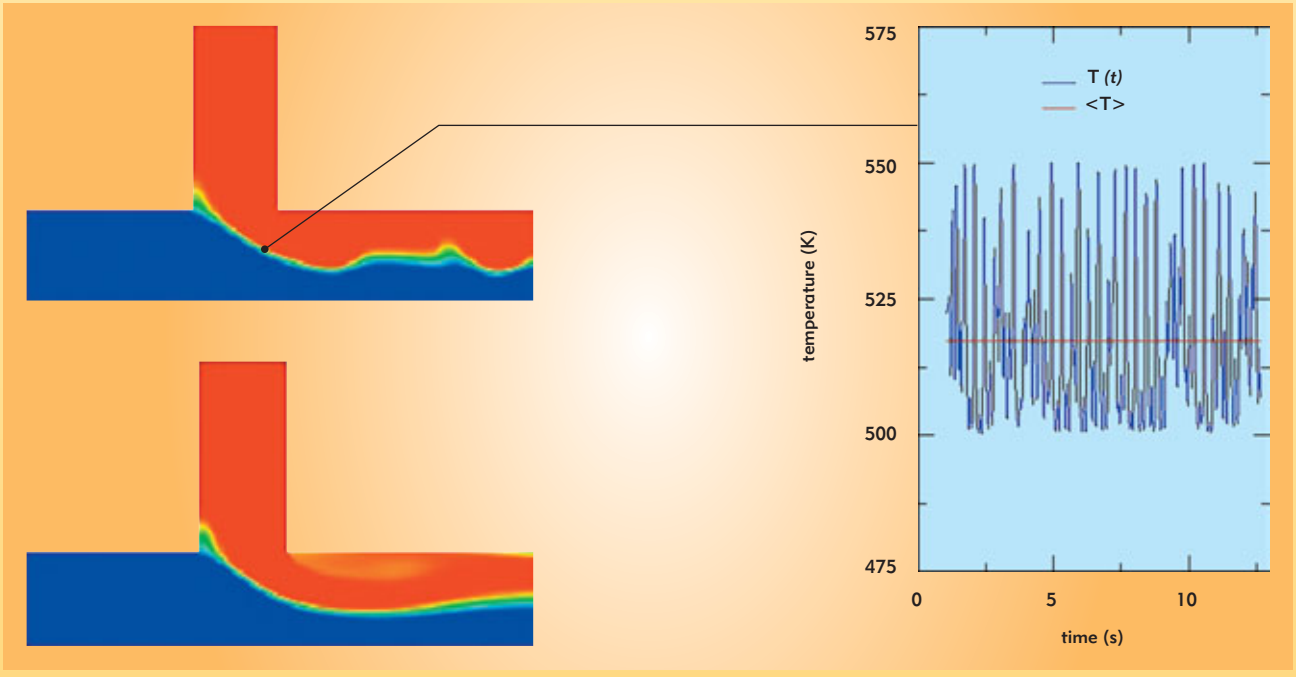


Figure. Instantaneous (top) and averaged (bottom) temperature field in a mixing situation. The curve shows the history of temperature at one point: fluctuating instantaneous value in blue and mean in red (according to Alexandre Chatelain, doctoral dissertation) (DEN/DTP/SMTH/LDTA).

The sole option, based on the fluctuating character of the flow due to turbulent agitation, must thus be to define and use average values. One of the most widely adopted approaches consists in looking at the problem from a statistical angle. The mean overall values for velocity, pressure, temperature… whose distribution characterizes the turbulent flow, are defined as the principal variables of the flow one then seeks to qualify relative to those mean values. This leads to a decomposition of the motion (the so-called Reynolds decomposition) into mean and fluctuating fields, the latter being the measure of the instantaneous local difference between each actual quantity and its mean (Figure). These fluctuations represent the turbulence and cover a major part of the Kolmogorov spectrum.(1)

This operation considerably lowers the number of degrees of liberty of the problem, making it amenable to computational treatment. It does also involve many difficulties: first, it should be noted that, precisely due to the non-linearity of the equations of motion, any average process leads to new, unknown terms that must be estimated. By closing the door on complete, deterministic description of the phenomenon, we open one to modeling, i.e. to the representation of the effects of turbulence on mean variables.

Many advances have been made since the early models (Prandtl, 1925). Modeling schemas have moved unabated towards greater complexity, grounded on the generally verified fact that any new extension allows the previously gained properties to be preserved. It should also be noted that, even if many new developments are emphasizing anew the need to treat flows by respecting their *non-stationary* character, the most popular modeling techniques were developed in the context of *stationary* flows, for which, consequently, only a representation of the flow's temporal mean can be achieved: in the final mathematical model, the effects of turbulence thus stem wholly from the modeling process.

It is equally remarkable that, despite extensive work, no modeling has yet been capable of accounting for all of the processes influencing turbulence or influenced by it (transition, non-stationarity, stratification, compression, etc.). Which, for the time being, would seem to preclude statistical modeling from entertaining any ambitions of universality.

Despite these limitations, most of the common statistical modeling techniques are now available in commercial codes and industrial tools. One cannot claim that they enable predictive computations in every situation. They are of varying accuracy, yielding useful results for the engineer in controlled, favorable situations (prediction of drag to an accuracy of 5–10%, sometimes better, for some profiles), but sometimes inaccurate in situations that subsequently turn out to lie outside the model's domain of validity. Any controlled use of modeling is based, therefore, on a qualification specific to the type of flow to be processed. Alternative modeling techniques, meeting the requirement for greater accuracy across broader ranges of space and time scales, and therefore based on a "mean" operator of a different nature, are currently being developed and represent new ways forward.

The landscape of turbulence modeling today is highly complex, and the unification of viewpoints and of the various modeling concepts remains a challenge. The tempting goal of modeling with universal validity thus remains out of order. Actual implementation proceeds, in most cases, from compromises, guided as a rule by the engineer's know-how.

(1) One may mention the spectral distribution of turbulent kinetic energy known as the "Kolmogorov spectrum," which illustrates very simply the hierarchy of scales, from large, energy-carrying scales to ever smaller, less energetic scales.
(2) This range results from the non-linearities of the equations of motion, giving rise to a broad range of spatial and temporal scales. This range is an increasing function of the Reynolds number, Re, which is a measure of the inertial force to viscous force ratio.
(3) The hypothesis that complete resolution of the Navier–Stokes equations allows simulation of turbulence is generally accepted to be true, at any rate for the range of shock-free flows.
(4) This is a problem governed by initial and boundary conditions.

Frédéric Ducros
Nuclear Energy Division
CEA Grenoble Center