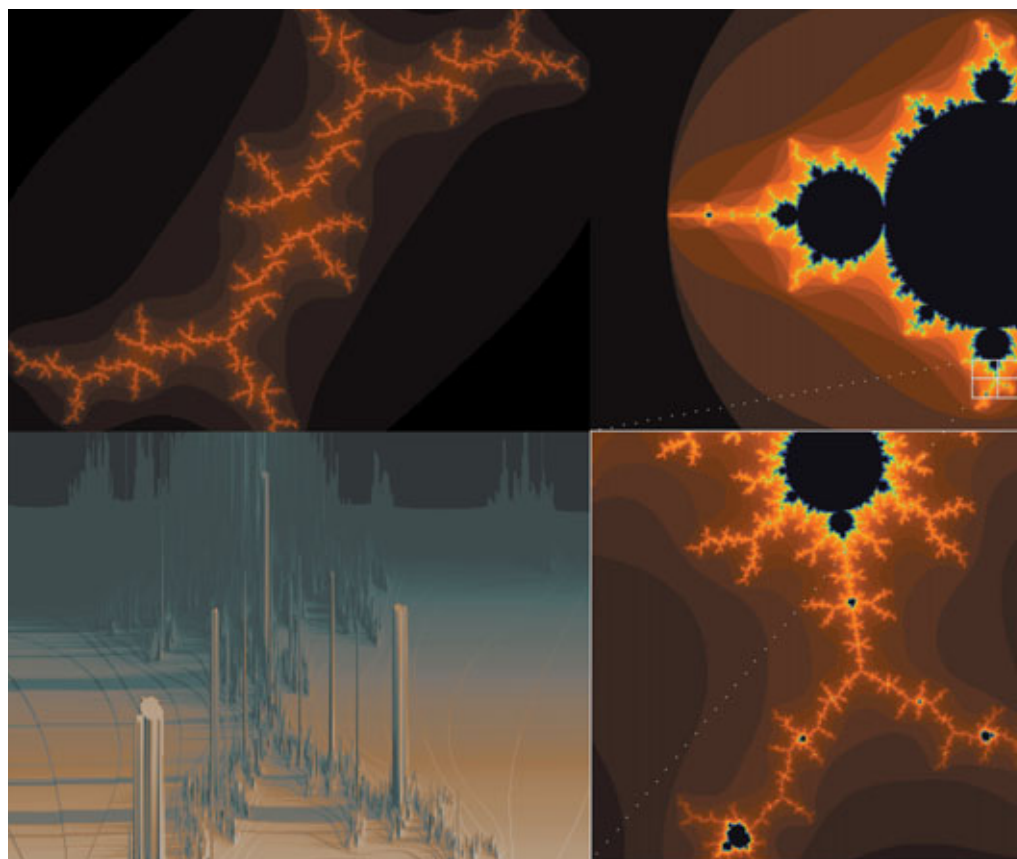


THE CHANGING GUISE OF THEORY FOR PREDICTIVE PHYSICS

Whether theory precedes or follows experiment, in the realm of physics it needs must make, in more or less provisional fashion, certain simplifications. That is the price to be paid, if it is to be predictive in practice at a given time. In particular, theory has to identify all significant parameters allowing a description of the phenomena at a given scale. In its newest guise, this idea may be seen, for instance, in the quantum field theory in particle physics, leading to the concept of effective theory.



Jean-Francois Colonna (CMAP/Ecole polytechnique, FT R&D).

Illustration of scale decoupling effects. Whereas iterated magnification of a printed straight line segment only shows additional details if printing defects become visible, a fractal geometry image (here, one titled Along the boundary of the Mandelbrot set) with no scale decoupling, built up iteratively, appears different for each magnification, showing ever more detail (actually repeating the same pattern, as shown at right). This is the equivalent of the renormalization group. The algorithm stipulates how to pass over from one scale to the next: at a given magnification, one may speak of an effective image. The image at lower left is a 3D view of the same pattern.



In the twentieth century, in two separate areas of physics, the theory of fundamental interactions⁽¹⁾ and the statistical mechanics of phase transitions, one of the fundamental notions in physics came to be questioned: *the decoupling of physics at very different scales*. Let us consider two examples to illustrate this notion. Straightforward dimensional analysis shows that the period of a pendulum varies as the square root of its length. This finding actually rests on the assumption that other lengths, such as the size of the constituent atoms of the pendulum, or the radius of the Earth, do not come into the picture, being respectively far too small or far too large. Likewise, we can calculate the trajectories of planets to a very good approximation by assuming that the planets and the Sun are material points, since their radii are much smaller than the sizes of the orbits.

Clearly, if this decoupling property were not generally true, predictive physics would be practically impossible.

The theory of fundamental interactions

In the 1930s, the first quantum electrodynamics calculations⁽¹⁾ were hampered by apparently insurmountable problems: many results were infinite, and these infinities were directly linked to the point nature of the electron. It quickly became apparent that these infinities were of fundamental character, appearing to follow unavoidably from the point nature of particles and the conservation of probabilities.

(1) In this respect, see *Clefs CEA* No. 36, p. 24.

However, an empirical solution to this problem, known as *renormalization*, was found. This consisted in expressing measurable quantities not in terms of the initial parameters in the theory, such as the electron's charge in the absence of interactions, but in terms of renormalized quantities such as the observed charge of the electron. The findings arrived at by this method were then confirmed, in spectacular fashion, by experiment. At first sight, this method appears to be an application of the usual principle: use parameters suited to the observation scale. However, in this context, three bizarre features stood out: the infinite ratio between initial and observed parameters, the requirement, if the renormalization method was to be made to work, to introduce at times unforeseen additional interactions, and finally the feature that the intensity of the interactions

Foucault's pendulum in the Pantheon (Paris). To arrive at the finding that the period of a pendulum varies as the square root of its length, it was necessary to assume that other lengths, such as the size of the pendulum's constituent atoms and the radius of the Earth, are not involved, being respectively far too small or far too large.



Palais de la Découverte

depended on observation scale. For instance, at ranges smaller than one electron wavelength, the theory predicts an “anti-screening” phenomenon. This phenomenon was later verified in very direct manner at high energies, in other words at short range: the observed charge of the electron at an energy of 100 GeV, which corresponds to the mass of the Z particle, is 4% higher than its “usual” value, i.e. the Coulomb charge.

Critical phenomena and Gaussian theory

Critical phenomenon theory sets out to describe continuous or second-order phase transitions in macroscopic systems, such as liquid–vapor transition, transitions in binary mixtures, superfluid helium, magnetic systems. These transitions are characterized by large-scale collective behaviors around the

transition temperature (the critical temperature T_c). It might then be expected that such behavior would be amenable to a macroscopic description, only involving a small number of parameters suited to this scale, making no explicit reference to the initial microscopic parameters or initial microscopic scale such as the size of atoms, crystal-lattice spacing or the range of forces. This concept leads to a Gaussian theory (for reasons analogous to the central limit theorem in probability theory).

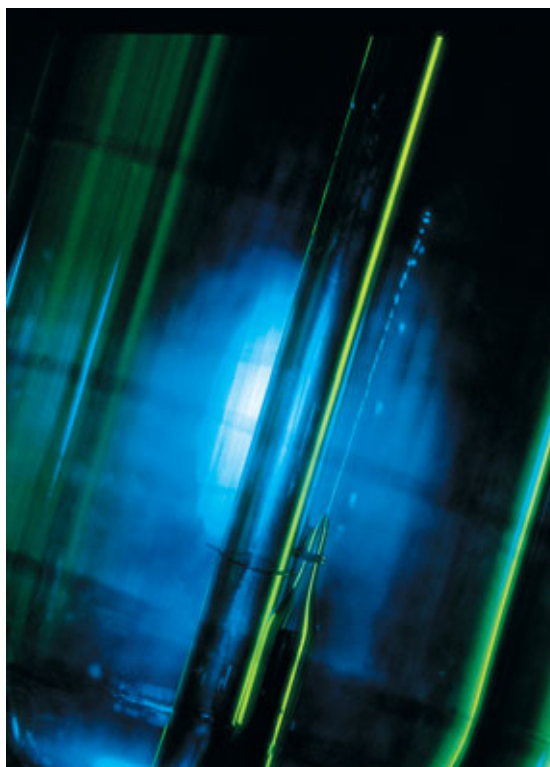
Physicists were very surprised at this theory's lack of agreement with either experiment or calculations made with statistical mechanical lattice models. The failure of the Gaussian theory showed that macroscopic properties could not be calculated if the microscopic scale were ignored altogether. Indeed, calculation of corrections to the Gaussian theory showed up infinities at the critical temperature. The divergences encountered in the field theory of particle physics and critical phenomenon theory in fact have a common source: to wit, non-decoupling of the various, different scales of physics. Infinities appear when an attempt is made to ignore as per usual – and in a manner that is usually justified – the existence of an underlying microscopic physical scale.

Universality and the renormalization group

In such a context, it might have been feared that macroscopic physics would be sensitive to all of short-range structure, and that long-range phenomena depend on the detailed microscopic dynamics, thus being essentially unpredictable. In fact, it was found empirically that only the distance scale associated to microscopic interactions and some of their general characteristics were significant, a property known as *universality*. The survival of a degree of universality, albeit reduced compared with Gaussian theory, was even more surprising.

To understand all these observations, a new conceptual framework was elaborated: the *renormalization group*. This is difficult to describe precisely in this article, however, the essential concept is the following one: an evolution equation is constructed for the variation of interactions as a function of the observation scale. As given initial condition one takes the microscopic interactions, and the *effective* interactions at longer range are looked for. If the evolution equation has fixed-point properties, in other words if for an entire class of initial interactions the same effective long-range interactions are found, the property of universality can be explained.

Fountain effect in superfluid helium. This is an instance of a phenomenon whose properties at macroscopic scale cannot be calculated when ignoring the microscopic scale altogether.



CEA/Oasis/D. Chaslerie



The planets of the solar system grouped together. Their trajectories were calculated assuming they were material points, since their radii are tiny compared to the size of their orbits.



PhotoDisc/StockTrek

This strategy was then used to explain all the properties of simple phase transitions. Remarkably, however, this established a relationship between fundamental interaction theory and phase transitions: the effective theory that describes long-range critical phenomena is a *quantum field theory*, of the same kind as that describing particle physics.

It then becomes hard to resist the urge to apply the same concepts to the theory of fundamental interactions.

Effective quantum field theories

One may now imagine that fundamental interactions are described at microscopic scale (microscopic, of course, as Planck length,⁽²⁾ compared with distance scales presently accessible to experiment), i.e. at very high energy, through a finite theory of unknown nature (*string theory*?). For reasons that have yet to be understood, this generates, through the cooperative effect of

a large number of degrees of freedom, a long-range physics with particles of very low mass (compared with the Planck mass,⁽²⁾ 10^{19} GeV).

At ranges slightly greater than microscopic scale, physics is described by a highly complicated field theory with an infinite number of interactions whose intensities vary with distance. As a consequence of the existence of a long-range fixed point, most interactions then decay rapidly and become negligible at long range (only one nevertheless brings about significant effects, gravity, due to its cumulative and attractive character). A finite number of interactions vary gradually (logarithmically) and remain extant at long range. These lead to the *standard model*, which describes nearly all of accessible particle physics.⁽¹⁾ Finally, some parameters can increase with range. For instance, in the standard model it would be

(2) Basic units considered by quantum physics, derived from the fundamental constants.

expected that the mass of the Higgs particle⁽¹⁾ would be close to the Planck mass, a mass incompatible with observed physics. This is a problem as yet unresolved, known as the *fine-tuning* problem.⁽¹⁾ One of the suggested solutions is a hypothetical supersymmetry.⁽¹⁾

This modern point of view, fundamentally based on the renormalization group and the concept of observation-scale-dependent interactions, not only yields a more consistent image of quantum field theory, but also provides a framework within which new phenomena may be discussed.

It also implies that quantum field theory is a provisional description that is not necessarily consistent at all scales, and ultimately destined to be supplanted by a more fundamental theory of radically different nature. Nevertheless, it should be emphasized that, for the time being, quantum field theory is still the most fertile framework for the investigation of many problems in physics in which a very large number of degrees of freedom interact strongly.

Jean Zinn-Justin
Physical Sciences Division
CEA Saclay Center

SIMULATIONS IN PARTICLE PHYSICS

In simulations accompanying particle physics experiments, the description, as detailed as possible, of physical processes, along with the development of increasingly sensitive detectors, plays an essential role. Chance, however, is also involved – as this is the best way of not introducing bias into the selection of data to be injected into the modeling of complex phenomena, and because quantum physics is intrinsically probabilistic. These simulations yield virtual data very close to what reality would be.

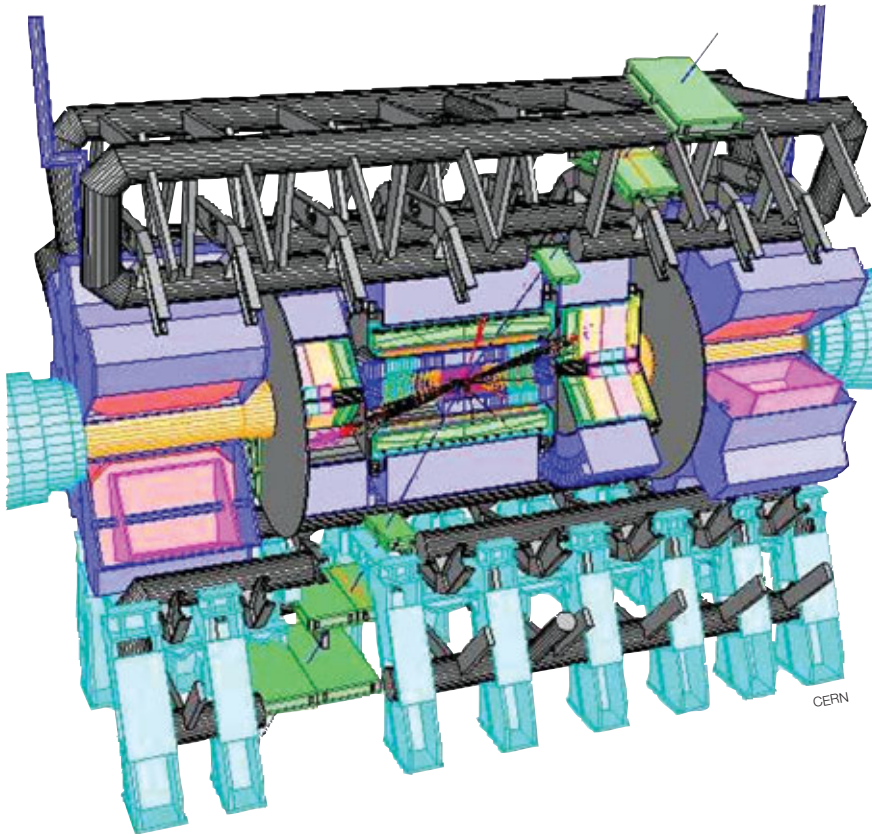


CERN

Listening to shop talk among particle physicists, you would no doubt be surprised to hear the word “Monte Carlo” so frequently. This is no reference to the site for a coming accelerator installation, nor to some conference center. “The” Monte Carlo is the simulation program implemented in every experiment, the essential

tool for the design of the detector, the development of data-analysis [algorithms](#), the calculation of detection efficiencies and elimination of background noise. The term does indeed derive from the name of the European gambling capital (Americans also speak of Monte Carlo, not “Las Vegas”!).

The name originates from the “**Monte Carlo integration method**” (see also Box A, ***What is a numerical simulation?***). What it is all about? Suppose you wish to redesign your dining room: in particular, you want a given intensity of illumination at the center of the table. To keep things simple to start with, you think of a single light, a ceiling fitting



Simulation of an event in the Atlas experiment using the Large Hadron Collider (LHC), currently under construction at CERN. To facilitate the view of the inside, only some of the detectors are shown.



hung over the table. If the light flux from the lamp is known, the calculation is easy. But you also have a standard lamp in the corner of the room, from which some light reaches the table after diffusion from the ceiling and the walls. This is getting more complicated. Just for the geometry, you now need to carry out several integrals of light flux, taking care to include all possible trajectories (ceiling light → table, standard lamp → table, standard lamp → ceiling → table, standard lamp → ceiling → wall → table...). After lengthy calculations, you realize that the ceiling and the walls neither reflect like mirrors nor act as perfect diffusers. You need wall “models”, and for each of these **models** all the integrals need to be done anew! (See Box A, **What is a numerical simulation?**)

A numerical solution is called for. You write a program that **simulates** light-ray generation by the lamps, and then plots the path of each one and calculates what happens to it, depending on the obstacles it meets. All that is needed now is to count the number of rays arriving at the table.

In this example, there is no need to involve chance in the generation of light rays. It is easy to generate rays uniformly distributed on an angle grid, provided there are enough of them to be sure no “interesting” path falls through a gap in the grid. However, casting at random the position of the rays initially emitted has the advantage of not inducing (in principle!) any regularity with ill-controlled consequences. Hence the use of the Monte Carlo integration method. The phy-

sicist asks the computer to generate “random” numbers to fix the direction of the initial ray.

However, chance may also be involved in the wall model. Suppose the wall is half reflector, half diffuser. For each incident ray, it produces several emerging rays: one reflected, the others diffused (each being assigned a fraction of the incoming ray’s intensity). Again, the problem of keeping count will arise. To keep the program simple, keeping one single ray is enough, its lot (reflected or diffused) being drawn at random (and, in the latter case, casting for the direction of diffusion).

In particle physics simulations, chance is also specifically involved for another reason. Since particles are **quantum** objects, their reactions can only be described by probabilities. This would also be the case if a quantum model of the wall were adopted. The light ray would then be a single photon and its reaction with the wall would be described in terms of probabilities (50% reflection, 50% diffusion), in this case, however, in fundamental fashion.

Indispensable for analysis...

From the 1970s on, with experiments becoming increasingly complex, the Monte Carlo simulation program has become indispensable for data analysis. Initially, this involved fairly simple calculations, such as for the detector’s efficiency for a particle from a given direction, with a given momentum.

However, to take into account the correlations between particles in the same event, the physicist was soon led to simulating a complete event, including the behavior of each particle in the detector, in increasingly detailed fashion.

The 1980s saw the development of software programs specialized in a particular area of simulation, which could be used for all experiments as “event generators.” The user feeds into the program the conditions of the experiment, for example a collision between an electron and a positron (positive electron) with an energy of 200 gigaelectronvolts, and the program generates simulated events, with all the particles for the final state, their type, direction and momentum. The physicist can, of course, control all parameters, activate or deactivate each possible reaction, allow or exclude disintegrations, etc.

Over time, theoreticians and experimentalists built into these programs the entirety of known physics, and most theoretical possibilities. There are a few such generators on the “market,”⁽¹⁾ and their authors strive to make subtle refinements or translate very swiftly into code the predictions from the latest vogue theory.

Another program then takes over from these generators. This takes a “turnkey” event and simulates the behavior of the particles in the

(1) It should be noted, however, that these programs are free of charge, like all scientific publications.

Installation of the CMS detector at the incipient LHC. The outer magnet ring has been inserted down the axis of the detector.



CERN

detector. The **software**⁽²⁾ provides the framework and the particle propagation system, as well as the essentials of the physics of particle/detector interaction: curvature in the magnetic field, ionization, bremsstrahlung (deceleration radiation), pair creation, interaction with nuclei, etc. The description of the experiment is written by the user, along with many special cases of detector response for which the model available in the software is inadequate. This part of the effort involves a great deal of work, of the order of one million lines of code for a large experiment, which is as much as for reconstruction software programs.

... and for the quest for a new physics

With such tools, it is possible to generate simulated events in the format of genuine actual data and then process them in the same reconstruction chain. The physicist is thus able to assess precisely the performance of the equipment and analysis algorithms. For example, events that would be produced by an as-yet-unknown physics may be combined with similar events generated by known processes, which would constitute background noise. The task then is to check

that the analysis correctly identifies the interesting events.

At the experiment design stage, the physicist may modify the type or design of the detectors to enhance sensitivity to a specific theoretical possibility. Simulations then make it possible to refine analytical algorithms and control their effectiveness. Finally, they provide a means of proving that a signal seen in an experiment does come from a new physics, or conversely that this hypothetical new physics does not exist, if no signal has been seen.

In some collaborations, one team is responsible for generating a set of simulated events, containing one or several signals from a new physics. This set is sent on to the teams charged with analysis, without the nature of the added signals being revealed to them. As a rule, everything works out well, but surprises do occur: an injected signal is ignored, however, knowing they have to find something, the physicists do discover one... where there is none!

In conclusion, all particle physics experiments use simulations, frequently involving millions of events. Indeed, the increased performances and lower cost of computers encourage this trend. This represents a considerable investment in human resources and equipment. Decisions about the type of program used, software architecture, databases, etc., all have consequences for the entire duration of the experiments, ten years or more. The community must thus redouble its efforts to standardize and share software and methods.

(2) There is only one professional software package (complete suite of programs) of this type open to all: Geant, used by most collaborative projects (others have written their own program entirely). Originally developed by the European Particle Physics Laboratory (CERN) as a utility, Geant, in its latest version (IV), is an "open" project, with many contributors. This original writing method still needs to prove itself...

Bruno Mansoulié

Physical Sciences Division
CEA Saclay Center

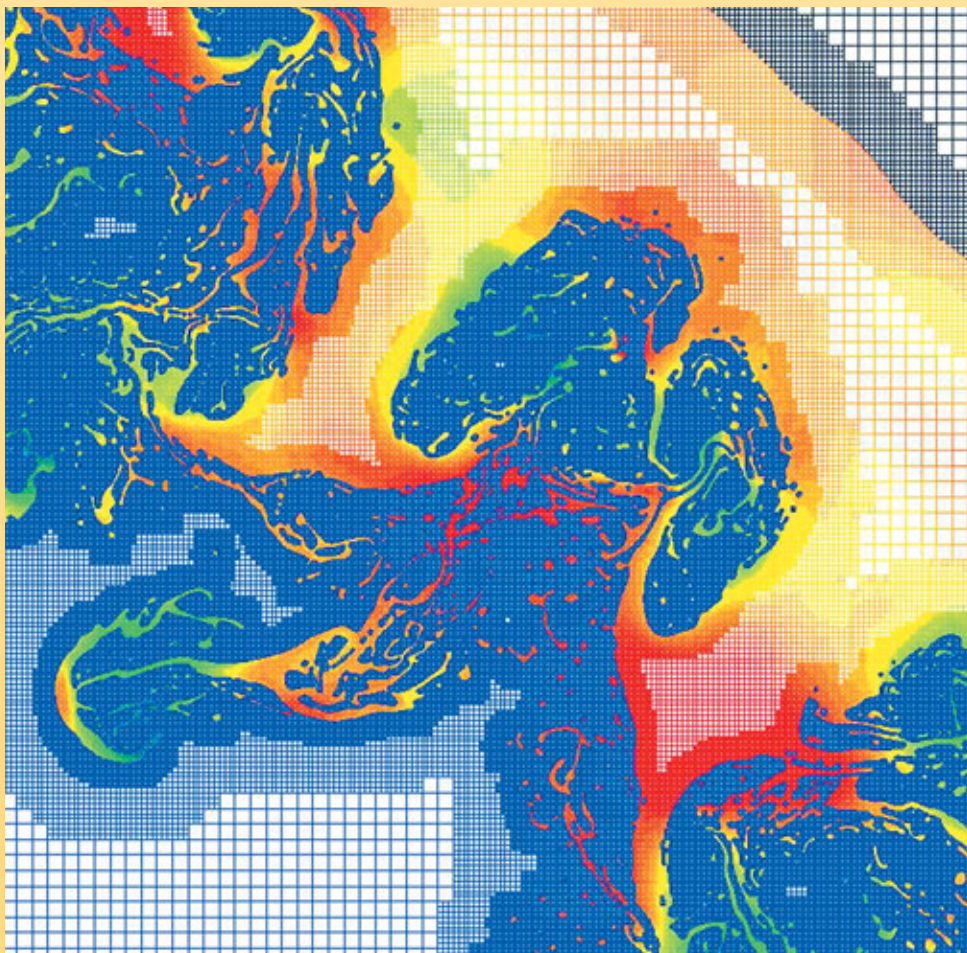
Numerical simulation consists in reproducing, through computation, a system's operation, described at a prior stage by an ensemble of **models**. It relies on specific mathematical and computational methods. The main stages involved in carrying out an investigation by means of numerical simulation are 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.



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 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.

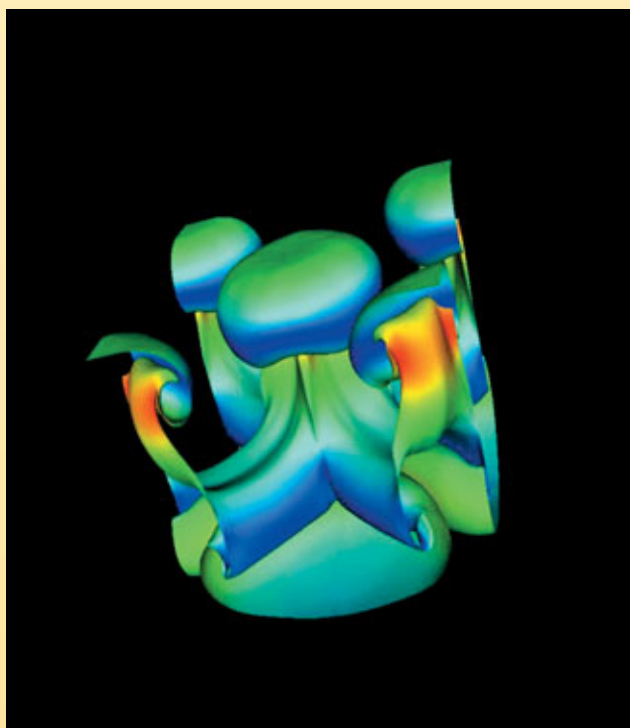
How does a numerical simulation proceed?

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) –, fol-



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.

lowed 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).

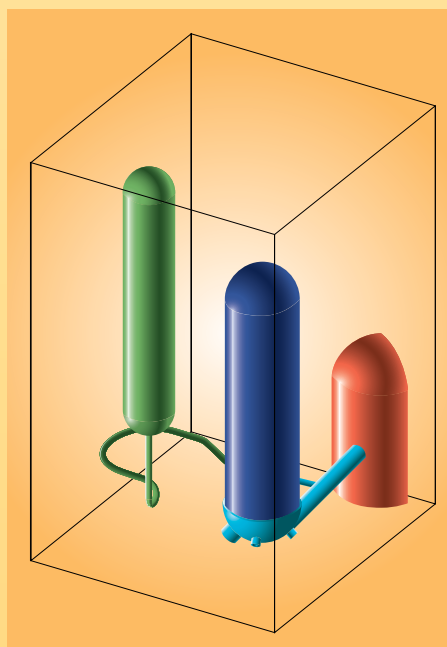


Figure A. Overall computation domain, including part of the reactor vessel (shown in red), the outlet pipe (hot leg, in light blue), steam generator (dark blue), and pressurizer (green).

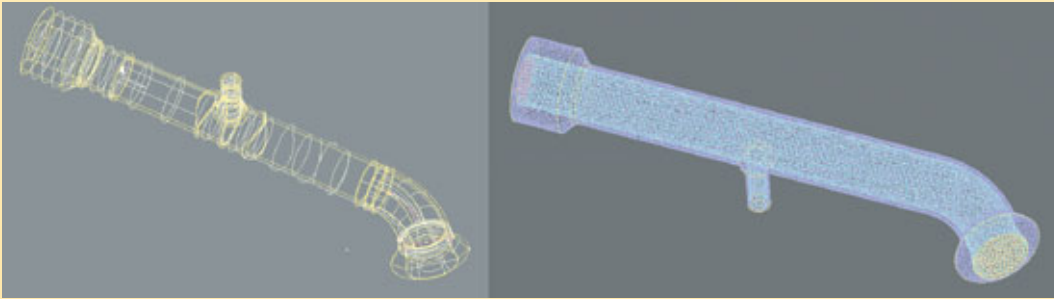
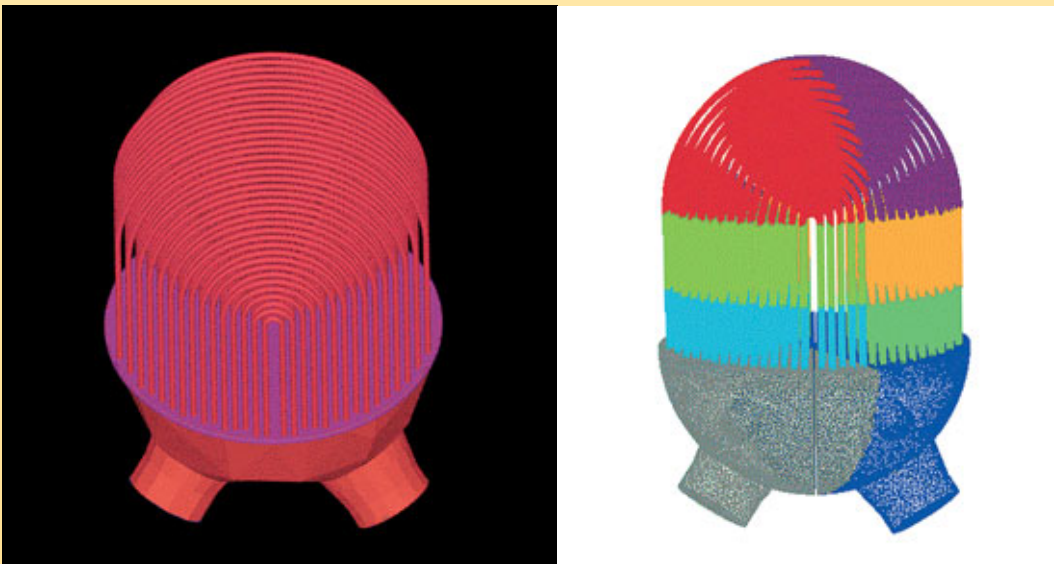


Figure B. CAD model of the hot leg of the reactor vessel outlet (left) and unstructured mesh for it (right).



Figures C and D.

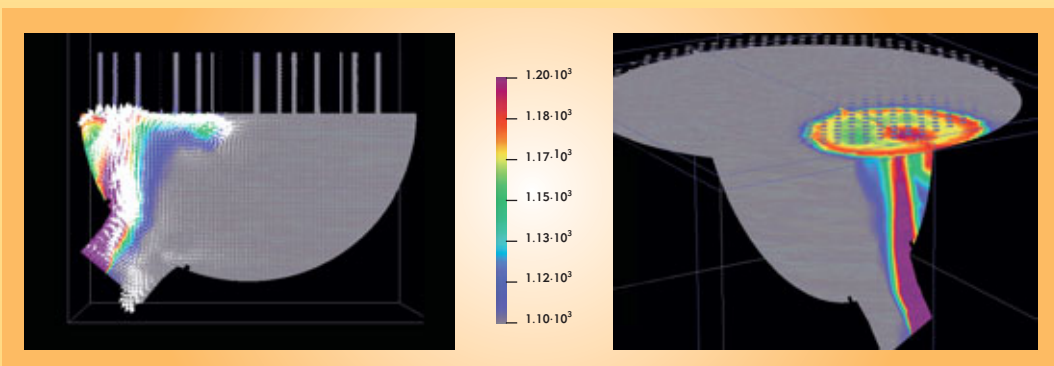


Figure E.

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 develop-

ments 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).

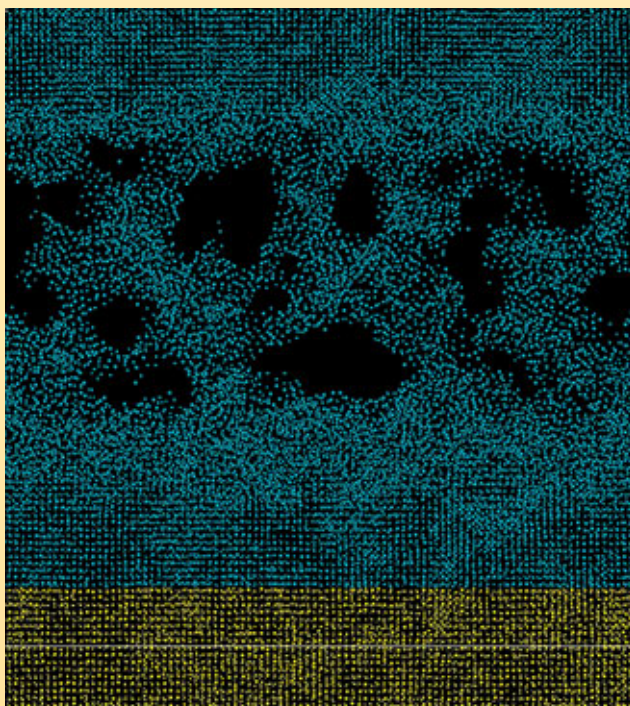
Currently, there are three main types of supercomputers: vector supercomputers, clusters of mini-computers with shared memory, and clusters of PCs (standard home computers). The choice between these architectures largely depends on the intended applications and uses. Vector supercomputers have very-high-performance 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 chosen 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 ($0.02 \mu\text{m}$ 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

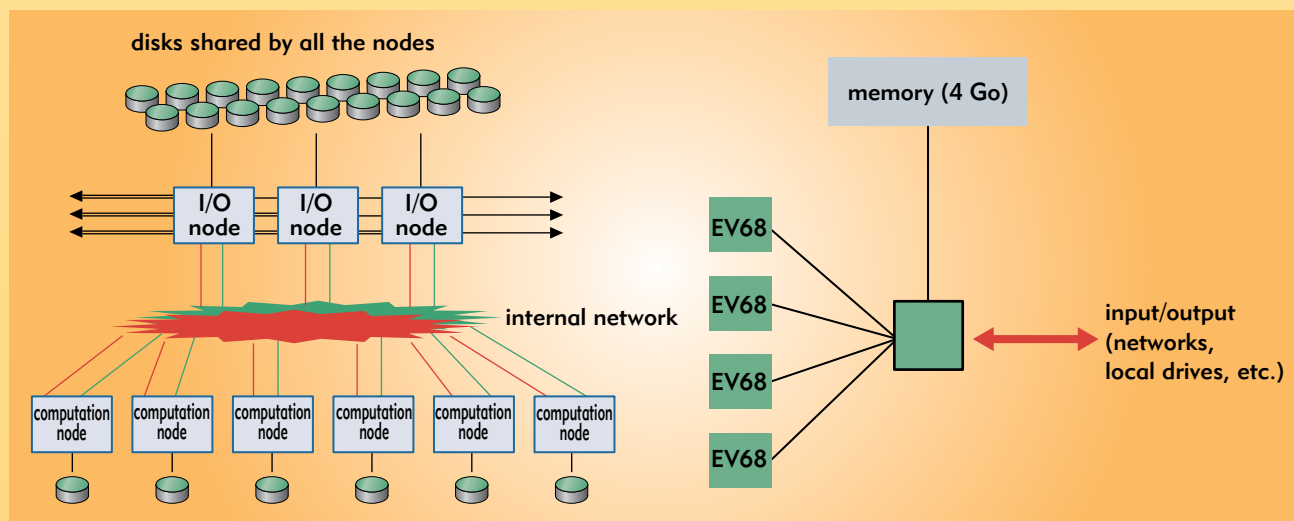


Figure. Architecture of an “SMP-cluster” type machine. At left, the general architecture (I/O = input/output), on the right, that of a node with four Alpha EV68 processors, clocked at 1 GHz.

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, paradoxically, 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 consi-

derable 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.

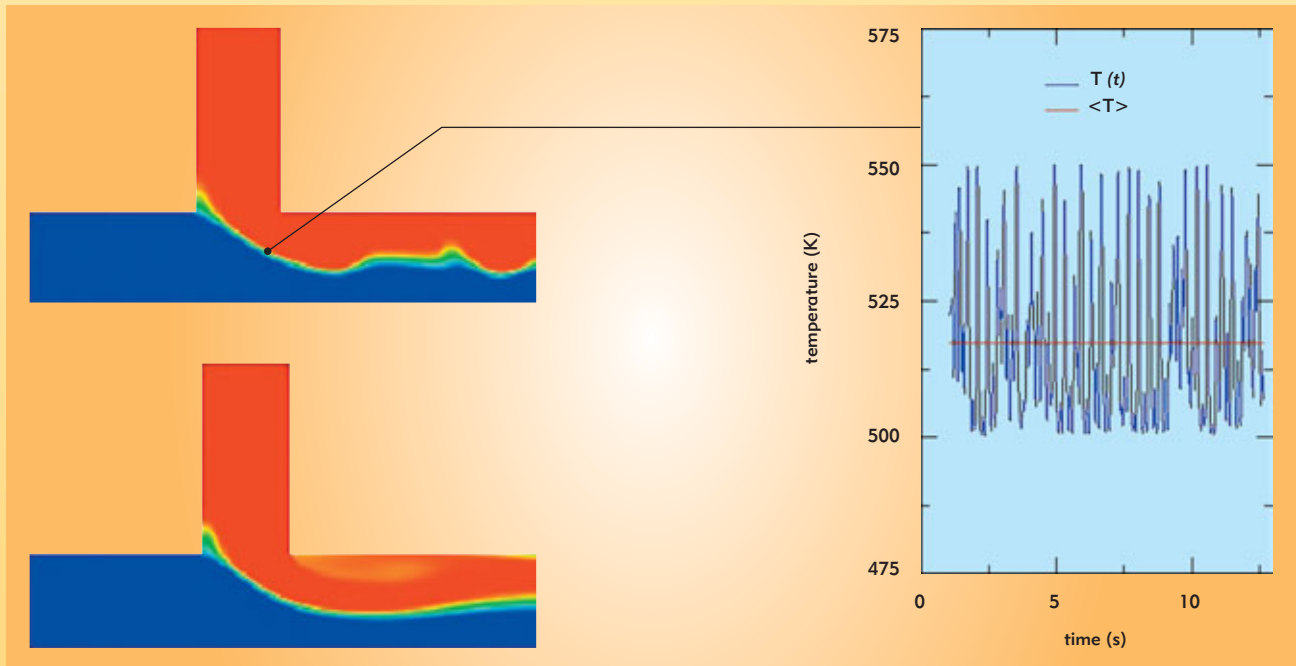


Figure. Instantaneous (top) and averaged (bottom) temperature field in a mixing situation. The curve shows the history of temperature at one point: fluctuating instantaneous value in blue and mean in red (according to Alexandre Chatelain, doctoral dissertation) (DEN/DTP/SMTH/LDTA).

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.

Frédéric Ducros

Nuclear Energy Division

CEA Grenoble Center

(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.