MODELING BIOLOGICAL MACROMOLECULES

Simulation is a complementary approach to experiment for the analysis of the huge quantity of data being produced by the many projects that are sequencing the genomes of various organisms. As part of its activity in this area, CEA has laboratories that model the structure and function of biological systems, such as proteins. In addition to increasing basic understanding of these systems, the results of this research will have industrial, medical and pharmaceutical implications, such as assisting in the development of new antibiotics, medicines and biomimetic compounds.

P. Dumas/CEA

Details of the target of a mass spectrometer showing protein samples (digested by trypsine) ready for proteomic analysis at CEA's Grenoble Center.

Simulation in the service of biology

The 20th century saw a transformation in our understanding of biology. This evolution occurred in all fields of the discipline and was based, for the most part, on the considerable advances obtained from the investigation of living organisms at the molecular level. Examples of advances in the last fifty years include the unraveling of the atomic structure of the principal biological **macromolecules** (Box 1), the detailed understanding of how these molecules operate inside the cell, and the impressive ability to manipulate biological systems at a microscopic scale. These discoveries culminated in the breakthroughs, in the last decade, of the **genome** programs (Box 2) and, in particular, with the publication in 2000 of the entire **sequence** of the 3 billion **base pairs** that make up the human **genome**.

This, however, is only the beginning. The deluge of data generated by these projects is set to increase. What is to be done with all this information and how can it best be used? It is clear that IT will play a major role in the answer to both of these questions, either through the storage and statistical analysis of the data – an approach encapsulated in the discipline of bioinfor-

1 Biological macromolecules

Cells are the fundamental entities in all living organisms. Apart from water, biological **macromolecules** are the major constituents of the cell, where they perform a multiplicity of functions. A biological macromolecule comprises sub-units of low molecular weight, added one to the other to form a long, chain-shaped **polymer**. Usually, each chain is formed of only one family of sub-units and the precise sequence of sub-units is essential to the function of the macromolecule. There are four major categories of macromolecules.

Proteins are probably the most important macromolecules, since they play a predominant role in most biological processes. For instance, **enzymes** are proteins that **catalyze** the majority of chemical reactions in the cell. Other classes of protein have more of a structural role or are involved in signaling,(1) regulation of **metabolism** or the immune system. Proteins are **amino acid** polymers – about twenty different types of amino acids are commonly found – and a protein may comprise several chains, each containing a few hundred amino acids. Proteins are often associated with other molecules, which assist in their biological tasks. The three-dimensional structure of proteins is very complex but critical to their function.

Nucleic acids – deoxyribonucleic acid (DNA) and ribonucleic acid (RNA) – are **nucleotide** polymers. **DNA**, in its double-strand form (two nucleotide chains arranged in a double helix), is the genetic material which, amongst other things, codes instructions for the amino-acid sequences of all the proteins synthesized by the cell. **RNA**, usually in a single-strand form (one nucleotide chain), is essential for protein synthesis.

Lipids, fundamental constituents of cell membranes, also play an important part in metabolism and as energy reserves. The lipids include a number of classes, such as phospholipids, triglycerides and steroids.

Polysaccharides are polymers of simple sugars, such as fructose and glucose. They play a structural role, particularly in plants (cellulose is a polysaccharide), are involved in molecular recognition and can serve as energy reserves.

(1) Transmission of signals allowing cells to communicate amongst themselves.

matics – or by employing the data to **model** the structure and function of biological systems using the laws of physics and chemistry. This paper discusses the latter approach and, in particular, the modeling of biological macromolecules at an atomic scale.

Modeling a system

Modeling requires a theoretical framework that defines how **models** of a system are built and what rules they obey. In the case of **molecular modeling** (see Box C, ***Molecular modeling***), this framework is specified by atomic theory and the laws of classical, **quantum** and statistical mechanics. Precise application of these theories depends, of course, on the nature of the system. For a biological macromolecule, a typical procedure involves five steps:

• **defining the composition of the system**. This step identifies the composition, number and type of macromolecules to be investigated, the nature of the environment – e.g. water or some other **solvent** – and the physical properties of the system – e.g. its temperature and pressure;

• **defining the laws that the model of the system obeys**. For molecular modeling, two elements are required. First, a method must be available to calculate the system's potential energy which is the sum of the interaction energies between all of the atoms in the system. This quantity is critical because it determines the most probable, or the most stable, configurations of the system. Second, an **algorithm** must be chosen that uses potential energy to explore the various configurations that are accessible to the system. This algorithm is often a classical dynamical one, which allows the motion of atoms to be followed as a function of time by solving the equations of **Newtonian mechanics**;

• **selecting initial conditions**, such as the position and velocity of each atom. Given the structural complexity of biological macromolecules, it is usual to start from structures determined experimentally by **X-ray crystallography** or **nuclear magnetic resonance (NMR)**;

• **simulating the system**, which involves solving the equations describing the atoms' motion. **Numerical simulation** (Box A, ***What is a numerical simulation?***) is indispensable in the case of biological macromolecules, since an analytical solution for such a large, complicated system of equations is not possible. The computation time required for a simulation varies depending on system size, the precision of the methods employed and the problem investigated. A realistic simulation of a system of about fifty thousand atoms would easily take up for some weeks the most powerful of today's computers;

• **analyzing the results.** This analysis, which is principally statistical, relates the

2 The genome projects

The human **genome** project and similar projects for other organisms, such as **yeast** and the mouse, consist in the **sequencing** and analysis of all the **DNA** that forms the inherited **genetic** makeup of an organism. The genome projects have spawned related approaches which also aim to investigate an aspect of the functioning of cells and organisms at a global level. Thus, for example: **structural genomics** tries to determine the three-dimensional structures of all the **proteins** (Box 1) in an organism, directly by resolving their structure, or indirectly by comparing them to other proteins with homologous sequences and known structures; **proteomics** is dedicated to characterizing all interactions between proteins within a cell; and **metabolomics** seeks to identify the composition and distribution of **metabolites** in the cell and how these change throughout the cell's life and under various external conditions.

results yielded by simulation to quantities measured experimentally. Ideally, this step raises new questions that may be investigated by new simulations or new experiments.

Examples of research

The Molecular Dynamics Laboratory at the Jean-Pierre-Ebel Institute of Structural Biology (CEA–CNRS–UJF/IBS) in Grenoble is mainly concerned with two broad areas of research: the detailed investigation of **enzyme** reaction mechanisms using simulation techniques based on **quantum chemistry**, and the simulation of larger-scale processes such as changes in the shape of **protein** structures (see *Predicting the 3D structure of proteins*). Apart from the basic knowledge gained from such research, the results of these projects have implications for the rational design of enzyme inhibitors(1) (medicines, herbicides, etc.) and in protein engineering.
Four current projects of the Molecular Dynamics Laboratory are presented here. It should be stressed that all of these studies, and all theoretical studies in general, call for close cooperation with experimentalists.

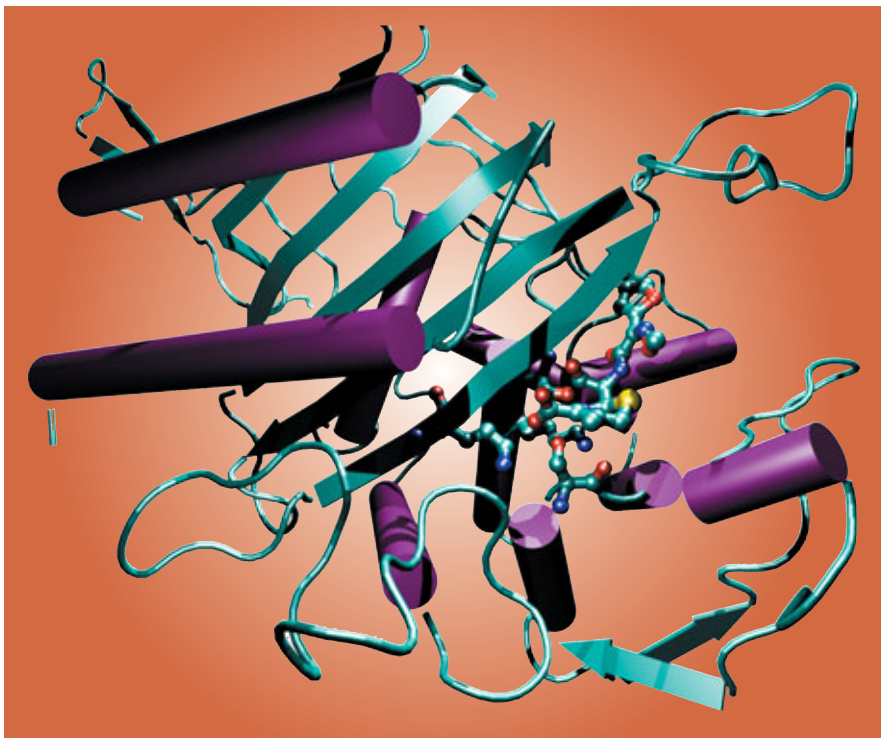
Bacterial PBPs

PBPs (penicillin-binding proteins) are essential for the synthesis of the cell wall that protects certain bacteria.(2) Their activity is blocked by β-lactam **antibiotics**, since these have a structure resembling the PBPs' natural substrate. Unfortunately, many pathogenic bacteria become resistant to these antibiotics through minor **mutations** in the structure of the PBPs. An investigation of the reaction mechanism between *Streptococcus pneumoniae* PBP2x and the antibiotic was undertaken (Figure 1) to identify characteristics of the reaction likely to help in the development of new antibiotics.

The HGXPRTase enzyme

The enzyme hypoxanthine-guanine-xanthine phosphoribosyltransferase (HGXPRTase) is a target for anti-malarial agents. Like many parasites, the organism responsible for malaria is incapable of synthesizing **purines**, which are essential components of **nucleic acids**. The HGXPRTase enzyme enables the parasite to scavenge purines from the contaminated human or animal host and convert them into purine **nucleotides**, which are indispensable for its survival. HGXPRTase acts by converting the purine hypoxanthine, and a sugar, ribosyl-1-pyrophosphate, into inosine monophosphate and pyrophosphate. This reaction was simulated using a classical **molecular dynamics** method and structures were suggested for reaction intermediates (Figure 2) which could serve as bases for inhibitor design.

(1) Enzyme inhibitors are specific compounds which inhibit the catalytic activity of an enzyme.
(2) Bacteria are generally unicellular microorganisms that have no nucleus and multiply rapidly.



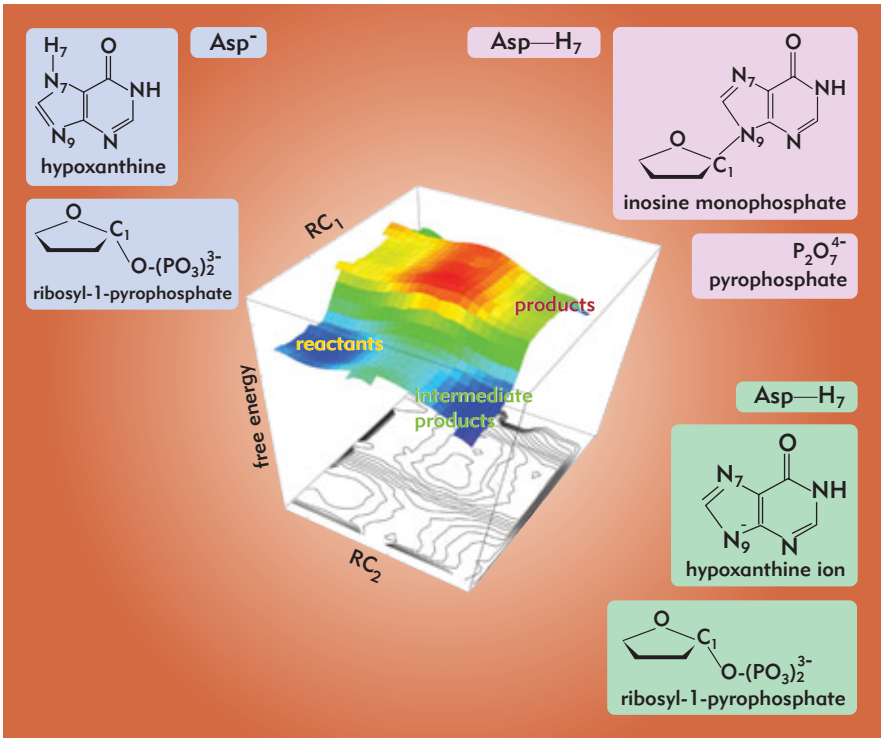
Monica Oliva (CEA–CNRS–UJF/IBS/Molecular Dynamics Laboratory) in collaboration with Otto Dideberg (CEA–CNRS–UJF/IBS/Macromolecular Crystallography Laboratory)

Figure 1. Active site of protein PBP2x, which binds the antibiotic cefuroxime. The antibiotic and important residues of the active site are shown explicitly (rods). The rest of the protein is illustrated with a ribbon representation, where only the path of the main chain is drawn (fine tubes). Some parts of the chain are shown as cylinders (α helices) and arrows (β sheets) and correspond to the classic structural elements found in most proteins.

The catalase enzyme

The catalase family transforms hydrogen peroxide into molecular oxygen and water. This reaction is essential at a cellular level because it is part of the defense system that protects organisms that can only live in the presence of oxygen (aerobes) against oxidative stress.(3) The very high catalytic potential of this enzyme is remarkable, given that its **active site** is buried 30 Å from the surface. Substrate (hydrogen peroxide) transport and elimination of the reaction products were investigated with molecular dynamics to determine whether the network of channels observed in the crystallographic structure plays a functional role. Thus, for example, the main channel linking the active site to the surface of the protein seems to be essential for the elimination of oxygen (Figure 3). Similar research, carried out in collaboration with the IBS Protein Crystallogenesis and Crystallography Laboratory, examined the channels that permit access of molecular hydrogen to the active site of another enzyme, hydrogenase.

(3) Oxidative stress is oxidation of cell components inducing damage to cells and tissues.

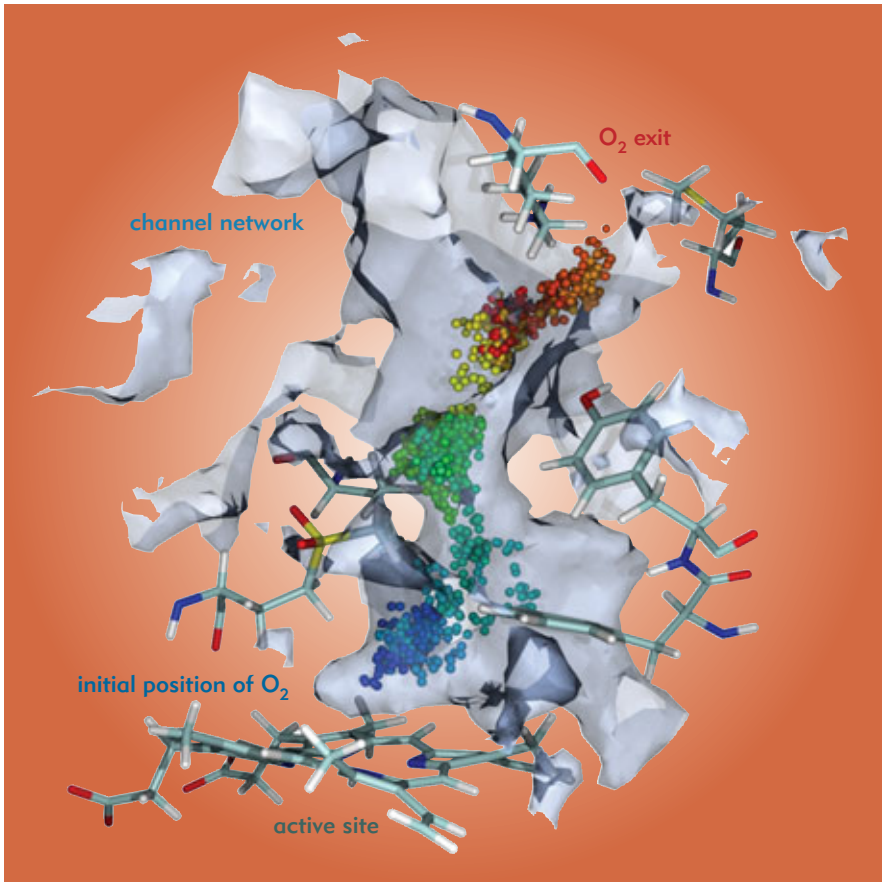


Aline Thomas (CEA–CNRS–UJF/IBS/Molecular Dynamics Laboratory)

Figure 2. Free-energy surface (thermodynamic quantity providing a criterion for equilibrium and the spontaneity of a process) for the conversion reaction of hypoxanthine and α-D-5-phosphoribosyl-1-pyrophosphate (reactants) into inosine monophosphate and pyrophosphate (products) by the HGXPRTase enzyme of the parasite Plasmodium falciparum. *The preferential path for the reaction involves an intermediate product, hypoxanthine, in the form of a negative ion. Asp indicates a protein group that accepts the hypoxanthine proton. The energy surface was calculated as a function of two reaction coordinates (RC_1 and RC_2) which can describe geometric changes during the reaction. The most probable structures correspond to the surface regions with lowest energy (in blue or purple).*

Transferrin

The normal task of this blood protein is the transport of iron to cells. However, it can also bind many other metals, including heavy metals(4) which are often highly toxic. Simulations of the dynamics of transferrin



Patricia Amara (CEA–CNRS–UJF/IBS/Molecular Dynamics Laboratory) in collaboration with Hélène Jouve (CEA–CNRS–UJF/IBS/Molecular Enzymology Laboratory)

Figure 3. Example of a diffusion trajectory for molecular oxygen O_2 (spheres), from the active site of the catalase to the surface of the protein via the enzyme channel network (represented here as a gray area). Only a few atoms of the active site and channels are shown (rods).

Molecular modeling

C

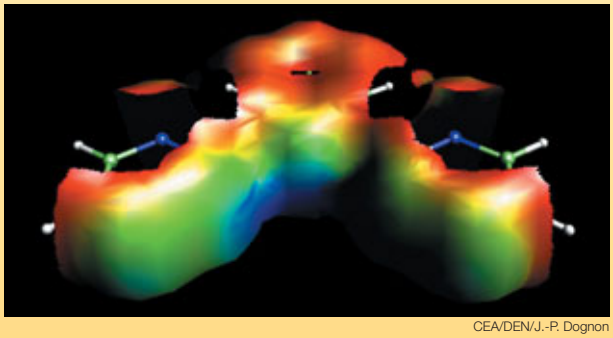
Researchers in biology, chemistry and the physics of materials increasingly use computational tools that enable them to model the behavior of molecules as a function of their structure. The accuracy of these approaches is now such that they are employed for the design of molecules and materials with specific properties.

A broad range of **theoretical** tools is available, including those based on the methods of quantum chemistry, grounded molecular mechanics and molecular dynamics.

Quantum chemistry is grounded on the laws of quantum mechanics and serves, above all, to describe the electronic structure of molecules. This is important for the understanding of processes such as chemical reactions.

Classical **molecular dynamics** simulates the motions of atoms in molecular systems, and the evolution of their spatial configuration, using the equations of classical mechanics. It gives access to structural, dynamic and thermodynamic properties.

Like quantum chemistry, **molecular mechanics** is a method that enables the investigation of the structure and behavior of molecules but it is less costly, faster and can be used to describe systems consisting of thousands of atoms, such as **biological macromolecules**.



CEA/DEN/J.-P. Dognon

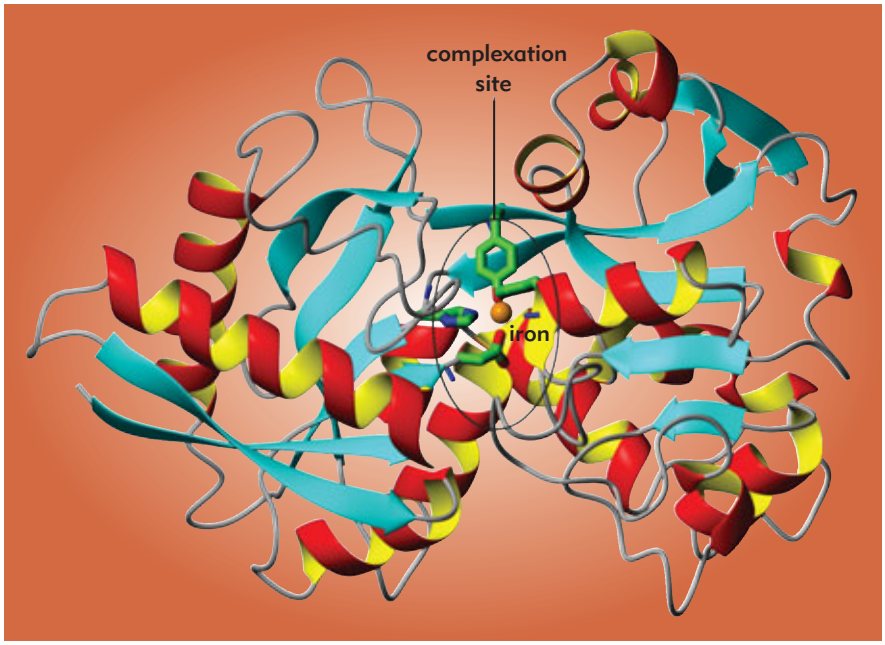
Representation of the electrostatic potential around the molecule, bis-triazinyl-pyridine (BTP) calculated by a quantum-chemical method. This molecule was developed for the Sanex process that separates actinides and lanthanides.

in an acidic environment were performed to analyse the iron-uptake and release mechanisms of the protein and to gain a better understanding of how it binds with heavy elements, such as uranium. The iron-**complexation site** comprises four **amino acids** and one carbonate molecule (Figure 4). This type of study may assist, in the long term, in the design of **biomimetic** molecules for the decorporation of **radionuclides**.

Simulation has a promising future in biology. Substantial advances have been made in the past few years due, to a large extent, to the exponential growth in the power of computers and improvements in simulation techniques. Today, simulation is a significant complement to experimental approaches, since real problems of fundamental interest can be tackled routinely. Undoubtedly, this trend will become more marked, leading simulation to contribute fully to the current revolution in biology. ●

Martin J. Field
Jean-Pierre-Ebel Institute
of Structural Biology
CEA–CNRS–UJF
Life Sciences Division
Grenoble

(4) Heavy metals are metals with a density greater than 4.5 g·cm^{-3}. They include zinc (7.14), cadmium (8.6) and lead (11.35).



David Rinaldo (CEA-CNRS-UJF/IBS/Molecular Dynamics Laboratory)

● ● ● ● ●

Figure 4. Three-dimensional structure of transferrin with, at center, shown as rods, the amino acids (carbon atoms in green, nitrogen in blue and oxygen in red) involved in complexation of iron (orange). The remainder of the protein is shown in a ribbon representation, with α helices (red and yellow) and β sheets (blue arrows). The complexation site is in the hinge region of the protein that permits the site to open and close thus explaining the mechanism of iron complexation and release.

What is a numerical simulation? A

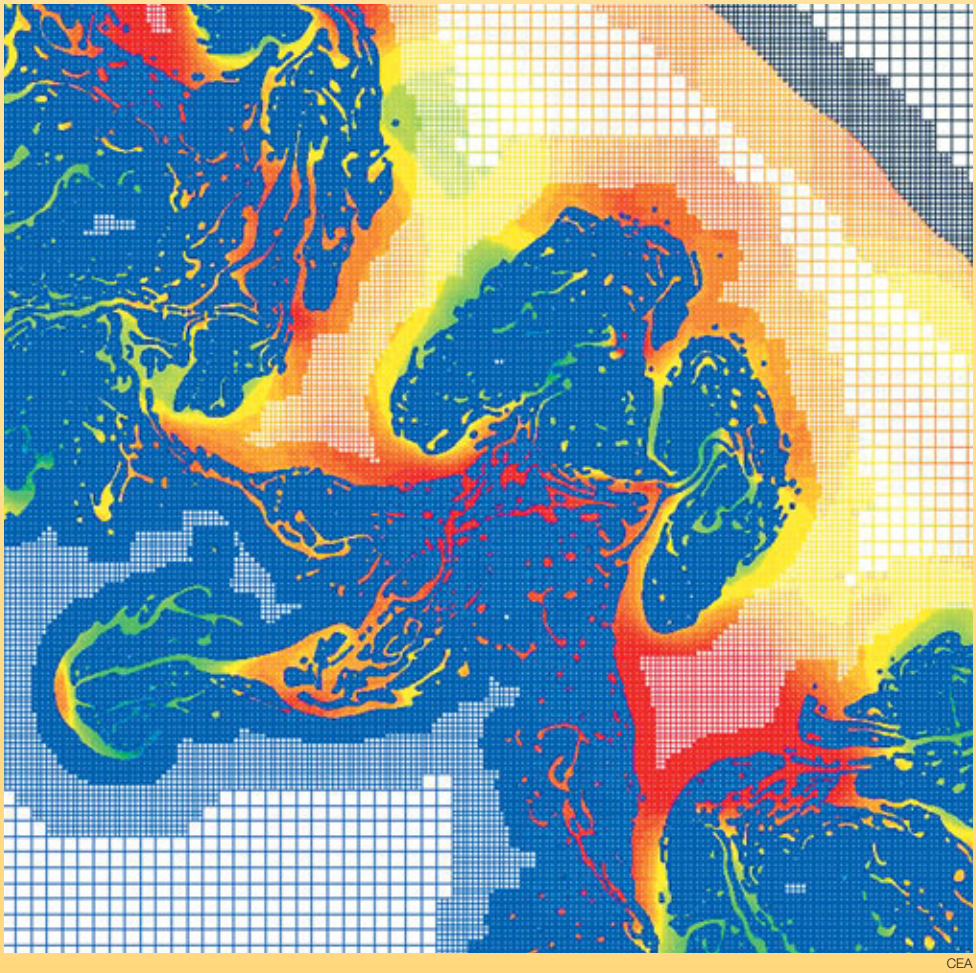
Numerical simulation consists in reproducing, through computation, a system's operation, described at a prior stage by an ensemble of **models**. It relies on specific mathematical and computational methods. The main stages involved in carrying out an investigation by means of numerical simulation are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

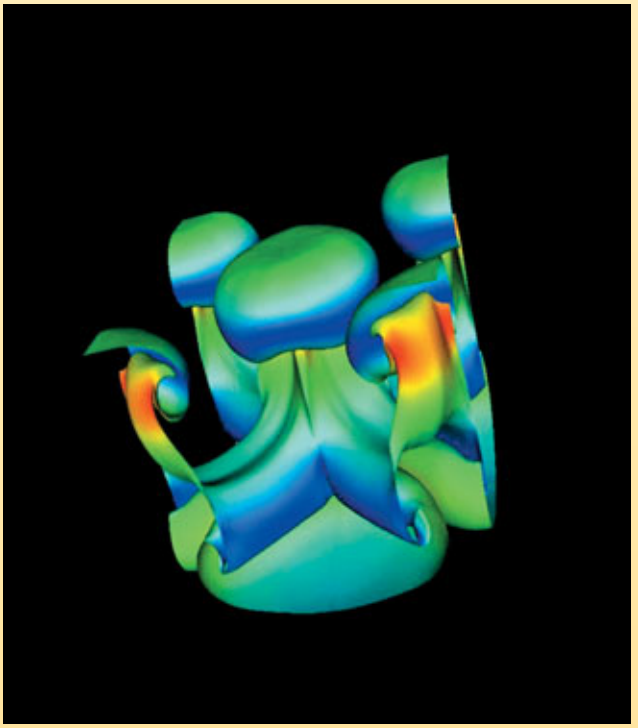
A refinement of meshing, **adaptive remeshing**, consists in adjusting cell size according to conditions, for example by making them smaller and more densely packed at the interfaces between two environments, where physical processes are most complex, or where variations are greatest.

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).

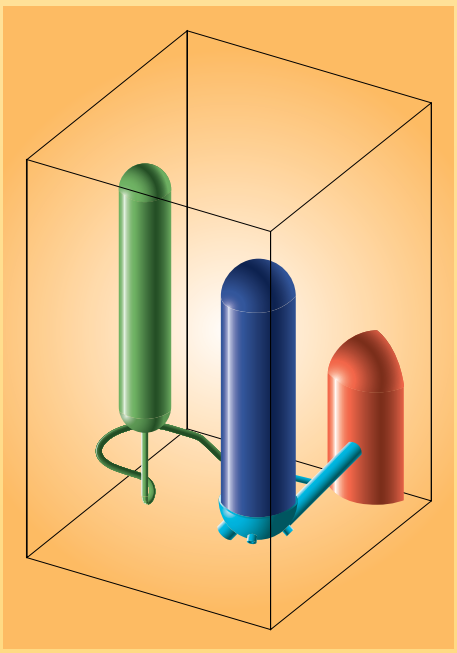


Figure A. Overall computation domain, including part of the reactor vessel (shown in red), the outlet pipe (hot leg, in light blue), steam generator (dark blue), and pressurizer (green).

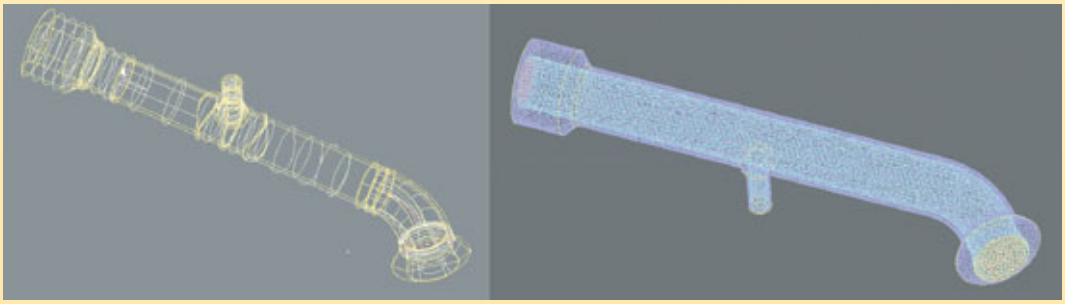
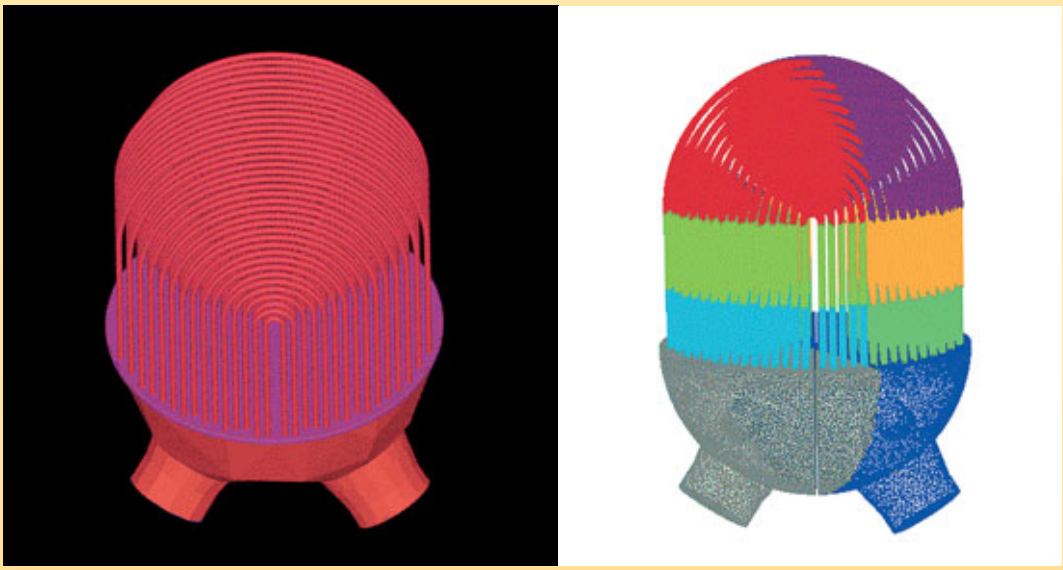


Figure B. CAD model of the hot leg of the reactor vessel outlet (left) and unstructured mesh for it (right).



Figures C and D.

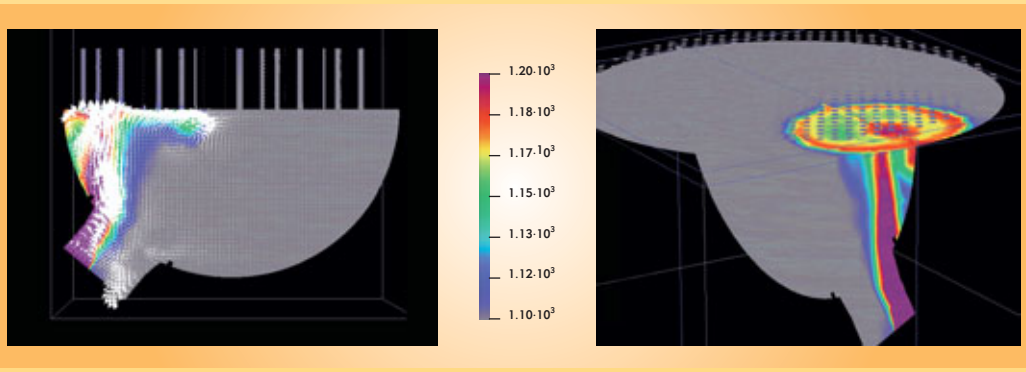


Figure E.

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.

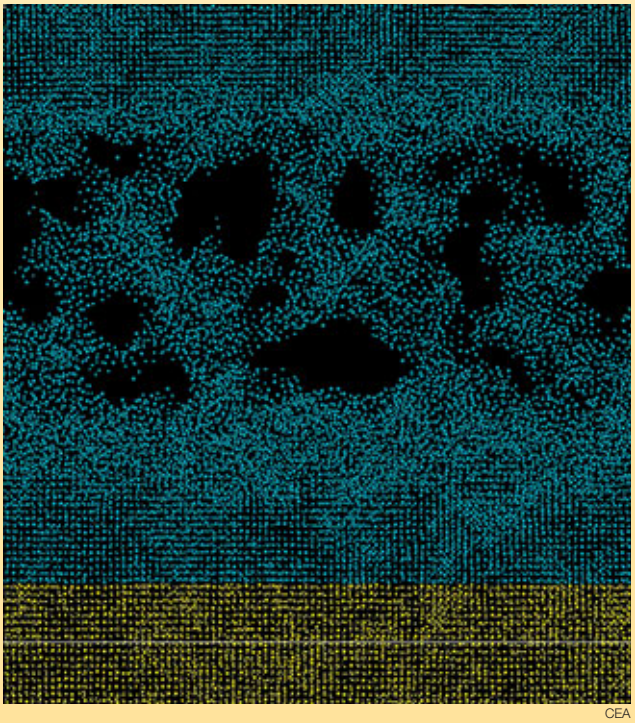
The most powerful computers are generally referred to as **supercomputers**. They currently attain capabilities counted in **teraflops** (1 Tflops = 10^{12} floating-point operations per second).

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

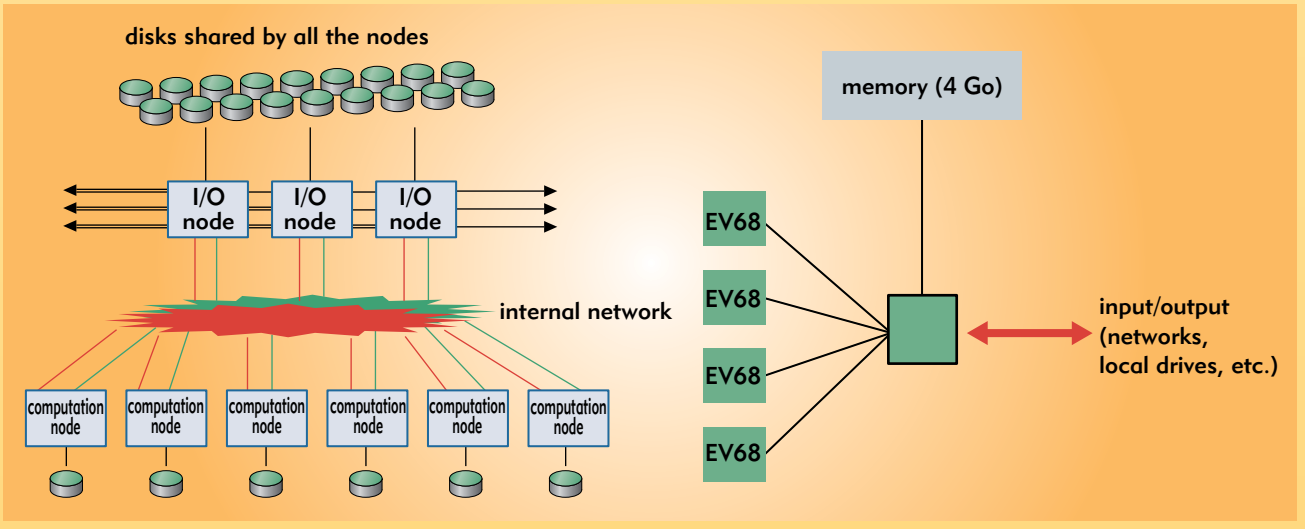


Figure. Architecture of an "SMP-cluster" type machine. At left, the general architecture (I/O = input/output), on the right, that of a node with four Alpha EV68 processors, clocked at 1 GHz.

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.

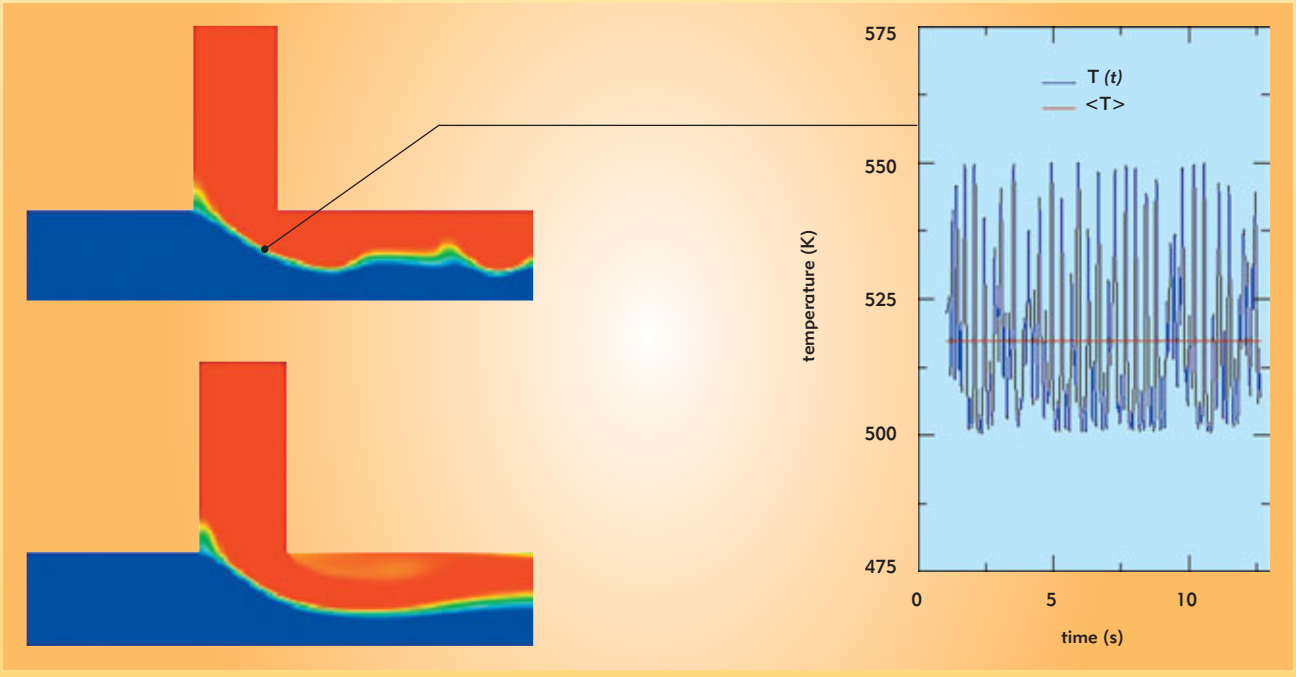


Figure. Instantaneous (top) and averaged (bottom) temperature field in a mixing situation. The curve shows the history of temperature at one point: fluctuating instantaneous value in blue and mean in red (according to Alexandre Chatelain, doctoral dissertation) (DEN/DTP/SMTH/LDTA).

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