PHILOSOPHICAL TRANSACTIONS OF THE ROYAL SOCIETY A MATHEMATICAL, PHYSICAL & ENGINEERING SCIENCES

Global Scientific and Engineering Simulations on Scalar, Vector and Parallel LCAP-Type Supercomputers [and Discussion]

E. Clementi and S. F. Reddaway

Phil. Trans. R. Soc. Lond. A 1988 **326**, 445-470
doi: 10.1098/rsta.1988.0097

Phil. Trans. R. Soc. Lond. A **326**, 445–470 (1988) [445]
Printed in Great Britain

Global scientific and engineering simulations on scalar, vector and parallel LCAP-type supercomputers

By E. Clementi

IBM Corporation, Scientific and Engineering Computation Department 48*B*/*MS* 428,
Neighborhood Road, Kingston, New York 12401, *U.S.A.*

We present here an example of the global simulation approach to complex systems, selecting, as example, the study of a liquid, specifically water. We start by building the molecules of the liquid from nuclei and electrons using quantum mechanics. Next we obtain the interaction potentials (two, three and four body), again by quantum mechanics. Then we use Monte Carlo and molecular dynamics to study the motions of a water molecule within its Onsager sphere, and the collective properties; subsequently we overlap fluid dynamics by considering a flow along a channel with or without obstacles; finally we extend further and report a preliminary simulation of a Bénard problem, using Newton's equations. These simulations are performed on a parallel supercomputer which we have recently assembled; the system is briefly described in the second part of this work. A number of applications in science and engineering are analysed with attention to the degree of parallelization achievable with and without special hardwares such as busses and bulk shared memory. To conclude the implication of the global simulation methodology and supercomputers evolution is discussed in terms of productivity of information.

It is not uncommon to find us, theoreticians, promoters of interdisciplinarity and of universality of knowledge, working in rather narrow (even if most exciting) scientific areas. This somewhat inconsistent behaviour carries over to computational chemistry; on one hand, we more or less explicitly claim that laboratory experiments can be complemented, or possibly duplicated or even rejected in favour of quantum chemical simulations. On the other hand, unfortunately, most often the computational 'test tubes' can only handle either molecular systems with two to ten electrons (at the wavenumber accuracy level) or with a few hundred (at the kilocalorie per mole accuracy level) or up to one thousand (but at the minimal basis set accuracy level). Notice that above we have listed three levels of approximations but each one requires supercomputers; indeed standard computations deal with simpler systems than those characterized above. In addition, very simple laboratory operations, such as dissolving molecules, cooling or warming up a solution, stirring, following a reaction path in a solution, etc. can hardly be simulated because most often the quantum chemical experiment is 'temperature independent', or '*in vacuo*', etc. The theoretical, numerical and computational difficulties in attempting to overcome these limitations are very real and well known but possibly not sufficient effort is being addressed to hit head-on this most unyielding problem. Below we summarize our tentative steps and attempts to untangle this situation.

In this context it is essential to focus on the supercomputers, namely, on the largest, fastest and most powerful available computer of our time. Let me note that obviously computers with one to a few gigaflop (10^9 floating point operations per second) peak performance can be used to do computations requiring megaflop (10^6 floating point operations per second) or even lower performance and, actually, a price–performance analysis might suggest a move exactly in this direction. Yet it would be a pity to use such beautiful machines for such low-complexity tasks. Indeed supercomputers should be used essentially for supercomputations and for the exploration of new avenues. Achieving the ability to simulate laboratory experiments realistically is one such goal.

For sometime we have been advocating the global-simulation approach to computer simulations in science and engineering (Clementi 1985*a*). The main idea is borrowed from the events leading the U.K. to the first industrial revolution, when, assembly lines were introduced, thus enormously increasing the productivity per worker unit. In an assembly line one starts with simple raw material and, step by step, without interruption, one transforms the raw materials into a more and more finished and complex product. In the global-simulation model we start with the most simple assumption, namely that molecules are built up of point charge nuclei and electrons. With quantum mechanics we can then assemble simple molecules, or larger ones; generally this is often done in a timeless and temperature-less environment. With statistical mechanics we can consider many molecules at a given pressure and temperature, pass from an enthalpic description to a free-energy description, and consider trajectories involving time; finally for even larger systems we can use fluid dynamics where viscosity, transport coefficient, convection, turbulence, etc. can be analysed. Of course all this is well known, at the theoretical level, but the novel aspect we have been stressing is how to move from one mechanics to the next one in such a way that the entire process is self-consistent with computer simulations at the operational level. Another novel proposal is to use molecules, Newton equations and boundary conditions to solve problems previously considered at the fluid dynamical level only.

Briefly stated in the global simulation model we decompose a given problem into n subproblems each one corresponding to a submodel $1, \ldots, i, \ldots n$. The operational rules are that the input needed in the submodel i is fully obtained as the output from submodel $(i-1)$ and that the input for submodel 1 must be very simple, in a computational sense. Thus the global simulation approach is an assembly line designed to increase our productivity in generating information.

Let us consider, as an example, the simulation of some chemical in the liquid phase and to be more specific let us consider water, a rather complicated but most interesting liquid. To start with (namely, the input for the first submodel) all we need to know is that one molecule of water has the stoichiometrical composition of two hydrogen atoms and one oxygen atom, which, to fit our submodel, we translate into the notion that our molecule is made up of ten electrons and has two one-charge point-like nuclei plus one more with eight positive charges. Standard quantum chemistry yields a stable molecule; within the Hartree–Fock approximation (Clementi & Popkie 1972), the stimulated water molecule has an HOH slightly too wide relative to experimental data (Benedict *et al.* 1956). The correct (within experimental accuracy) value is easily obtained by correlating the electron's motions (Bartlett *et al.* 1979). Thus now we know the geometry and the energetic of the water molecule from simulations; it is not difficult to obtain also the vibrational motions by mapping the intra-atomic potential surface in the vicinity of the equilibrium configuration (Bartlett *et al.* 1979).

Let us move from one molecule of water to a few, several (Kim *et al.* 1986) and finally many (Popkie *et al.* 1973). Computations of the interaction between two molecules even at the near to Hartree–Fock limit (Popkie *et al.* 1973), yield the well-known hydrogen-bond dimer (see figure 1). This picture can be improved by using correlated wave functions and energies (Matsuoka *et al.* 1976) (e.g. configuration interaction approach). The computed energy points, sampling the potential energy surface, can be fitted by some analytical expression, within the guidelines of some simple physical model. This way we obtained the two-rigid-waters intermolecular potential (two body), known as M.C.Y. (Matsuoka *et al.* 1976), or, by relaxing the water 'rigidity' and allowing vibrational flexibility, the M.C.Y.L. two-flexible body potential (Lie *et al.* 1986). In the same way one could proceed and consider trimers, tetramers, etc. A study of small water clusters (Kim *et al.* 1986) remind us that even for few water molecules, the cluster can have many stable geometries, all very near in energy, namely that we must face a difficult minimization problem. It should be noted that the cluster study provides us also with very interesting geometrical arrangements of hydrogen bonded structures, leading to 'clathrate-type' complexes. Obviously one should not build up indefinitely the cluster size, not only because it would become computationally impractical and eventually impossible, but mainly because it would be the logically incorrect approach to a problem whose answers are to be found in the understanding of statistical fluctuations. Incidentally, there is only limited agreement between our work on clusters and a recent paper by Belford & Campbell (1987).

It is time to move to the second submodel, namely equilibrium statistical mechanics, limiting ourself to two main techniques, namely Metropolis Monte Carlo (Metropolis *et al.* 1953) and molecular dynamics (Alder 1964). In the former, N molecules are placed in a volume V at a given temperature T and many (millions) of geometrical configurations are randomly

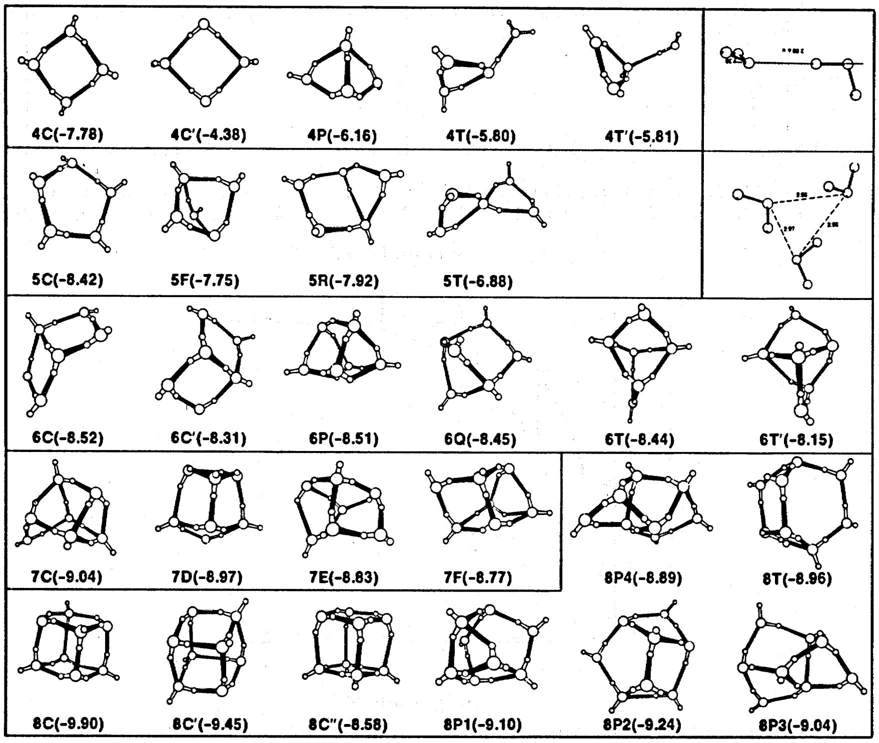


FIGURE 1. Clusters of water molecules $(H_2O)_n$. For $N \geqslant 4$ the most stable configuration is given at the left, the less stable at the right. For details see Kim *et al.* (1986).

generated, their energy computed with the interaction potentials obtained from quantum mechanics. Properties are statistically averaged in the Boltzmann sense. Periodic boundary conditions are imposed on the N particles. In molecular dynamics again we place N molecules in a given volume V and let it interact according the quantum-mechanically derived potential; for each particle the Newton's equations of motion are solved, namely we compute molecular trajectories, i.e. each molecule will modify its translational and rotational motion reacting to the field exerted by the $N-1$ surrounding molecules. Only motions corresponding to a very small temperature range (ideally one temperature value) are considered. The motions are discretized in time steps generally of 0.5×10^{-15} s for rigid molecules and 1.25×10^{-15} s for vibrating molecules.

Figure 2 reports some of the Monte Carlo results (Lie *et al.* 1976; Lie & Clementi 1986; Clementi & Corongiu 1983; Detrich *et al.* 1984). At the top we compare the computed and the

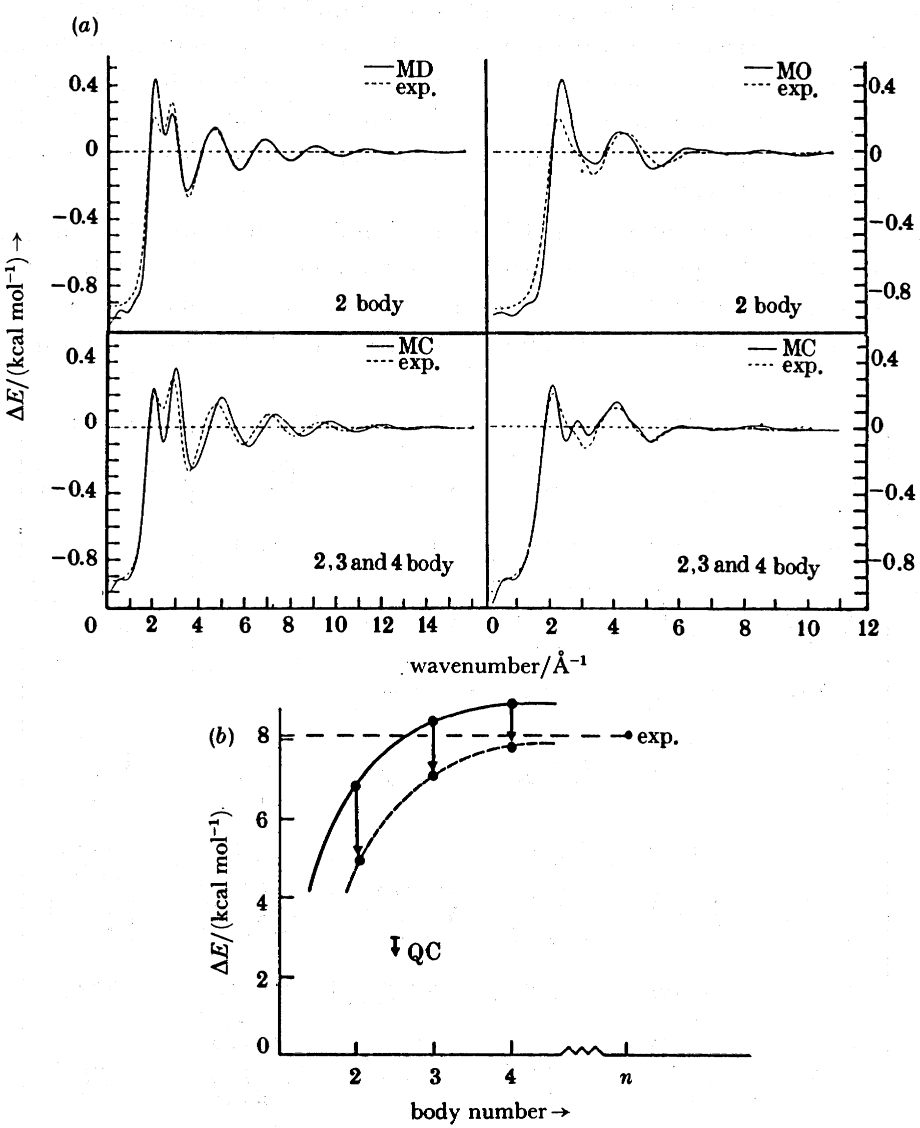


FIGURE 2. (*a*) X-ray (left) and neutron-beam (right) spectra of liquid water at room temperature either from experiment or from simulations with or without three- and four-body corrections. (1 Å = 10^{-1} nm = 10^{-10} m.) (*b*) Monte Carlo enthalpy of liquid water (T = 298 K; N = 512) from two-, three- and four-body potentials.

PHILOSOPHICAL TRANSACTIONS OF THE ROYAL SOCIETY A MATHEMATICAL, PHYSICAL & ENGINEERING SCIENCES

experimental scattering intensity for liquid water at room temperature using either X-ray (left) or neutron beam (right), comparing simulations at the two-body level with those which includes three- and four-body corrections. As we can see, the latter brings about a marginal improvement to an already good match.

However, for the enthalpy, the many-body corrections are very important. Thus static and structural properties are expected to be well described at the two-body approximation (and even better agreement can be obtained from well-calibrated effective potentials) but this might not be so for dynamical properties.

The latter type of properties can be subdivided into two groups, namely those relating the motions (translations and rotations) of one water molecule within its solvation cell (Wojcik *et al.* 1986*a*) and those describing the behaviour of many water molecules (Wojcik & Clementi 1986*b*) within a box of size L and with wavenumber $K = 2\pi L$.

From figure 3 we learn that rotational motion (angular correlation function, ACF) is nearly five times faster than the translational motion (velocity correlation function, VCF) for rigid molecules (Clementi & Corongiu 1983). These motions and the corresponding orientational correlation functions are at the origin of the infrared and Raman spectra for liquid water. The

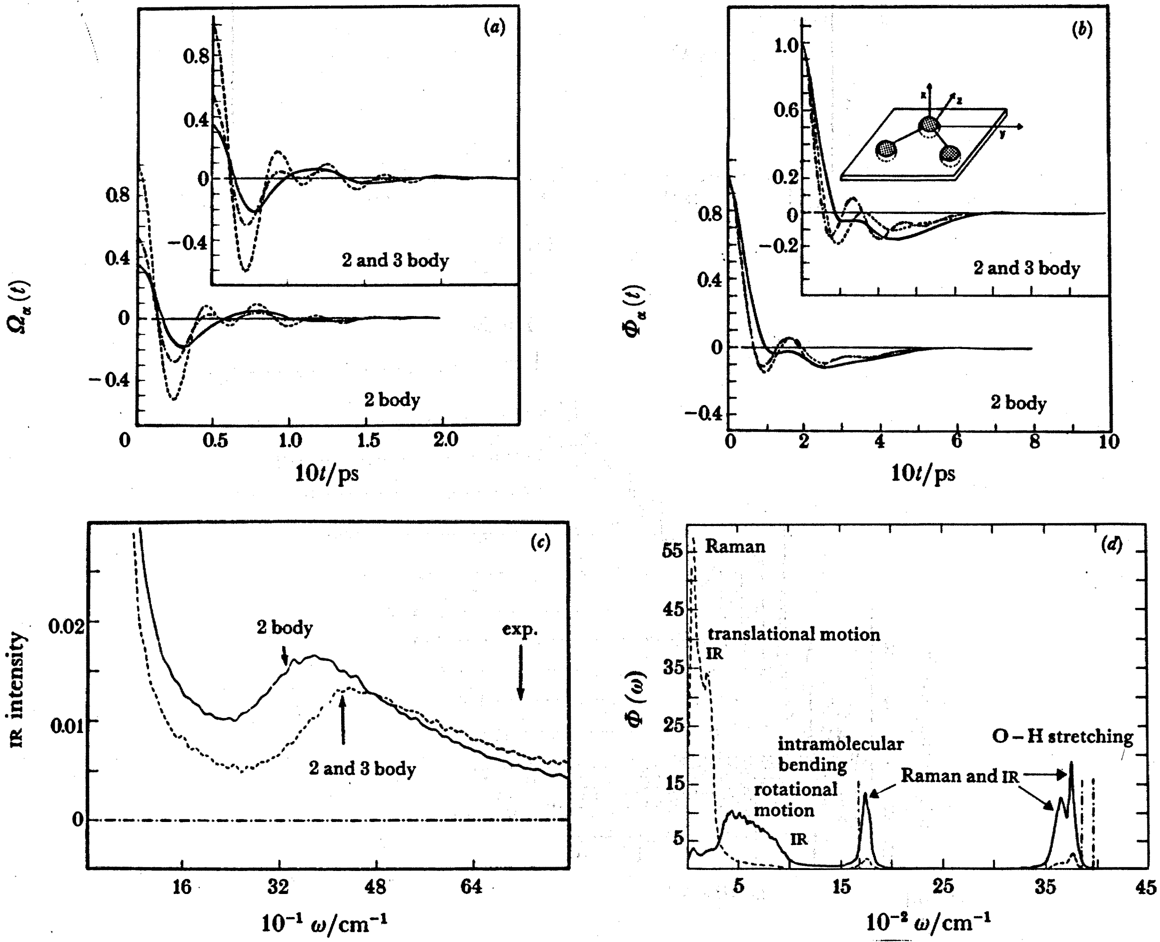


FIGURE 3. (*a*) Angular and (*b*) translational velocity correlation functions for liquid water at room temperature from molecular dynamics simulations with and without three-body corrections. Solid line, x-direction; broken line, y-direction; dashed–dotted line, z-direction. (*c*) Computed IR spectra. (*d*) Computed Raman spectra. Solid line, H; broken line, O; dashed–dotted line, H_2O.

computed spectra are reported at the bottom of figure 3; note that the librational IR spectra is incorrectly shifted to low values (insert at the bottom left) whereas the intramolecular motions (bottom right insert) are computed in remarkable agreement with experiments (Lie & Clementi 1986). For the intramolecular motions we have also reported equivalent spectra for the gas phase. The spectral shifts, comparing gas to the liquid is very accurately reproduced in our simulation. In figure 4 we report two 'collective' properties namely the sound velocity and the dynamical structure factor. Notice the lack of experimental data for sound velocity at small K values, namely when approaching the hydrodynamic limit; we recall that the shape

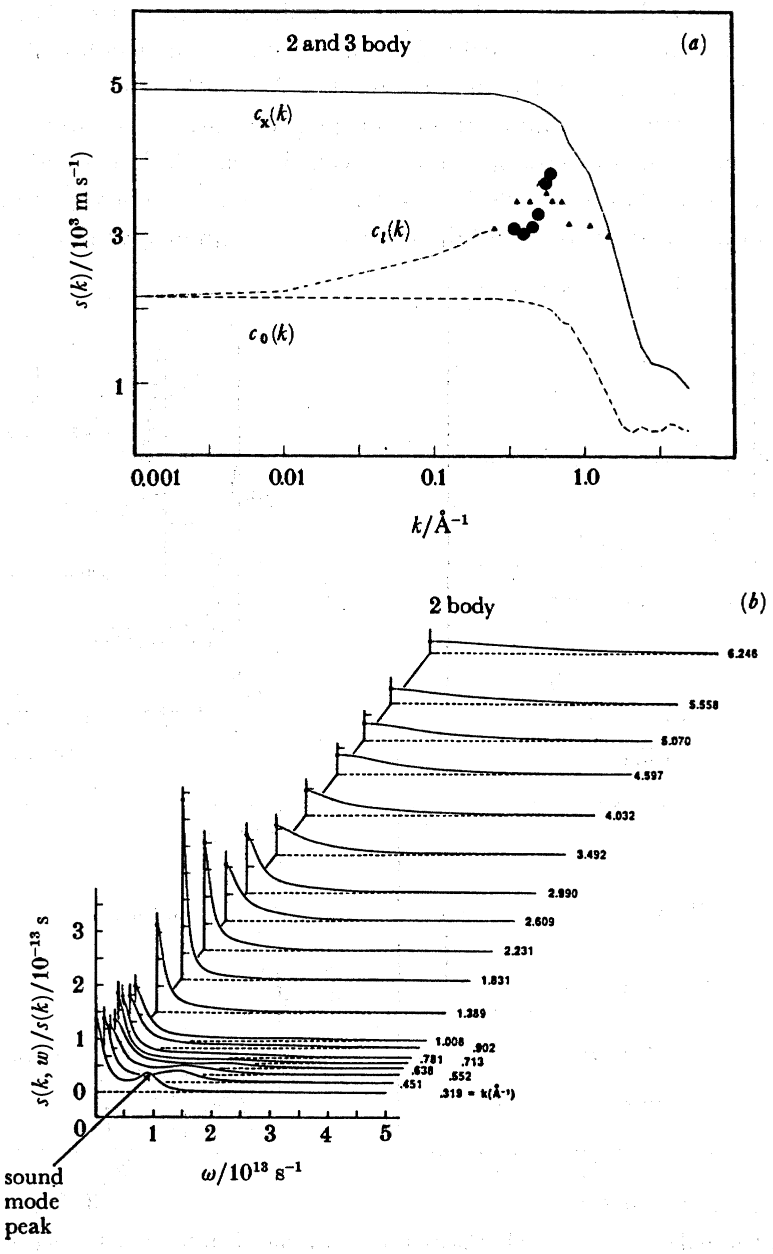


FIGURE 4. (*a*) Sound velocities from molecular simulations. ▲, molecular dynamic simulations; ●, experimental values. (*b*) Dynamical structure factor from molecular simulations. $S(k,\omega) = \frac{1}{2}\pi \int e^{i\omega t} F(k,t)\,\mathrm{d}t$, where $F(\bar{k},t) = \langle \rho_{\bar{k}}(0)\,\rho_{-\bar{k}}(t)\rangle$ is the density correlation function.

of the structure factor for $K = 0.319$ is not too dissimilar from the one predicted from hydrodynamic (Clementi *et al.* 1986*c*).

The simulated results presented in figures 2, 3 and 4 show that within the 'global simulation spirit' we can indeed progress from nuclei and electrons to a few water molecules, to the hydrodynamic limit. Let us now progress into the area hitherto exclusively reserved for fluid dynamics, i.e. for solutions of Navier–Stokes equations. We ask whether we can trust the Navier–Stokes equations for very small system of few hundred ångström size and for events having a lifetime in the picosecond or nanosecond range. Alternatively stated, the question concerns the description of systems generally larger than those analysed by molecular dynamics but much smaller relative to fluid dynamics. We did answer these basic questions by performing molecular dynamic-type computations with molecules or atoms interacting via intermolecular potentials, but in an open (non-equilibrium) system and with those boundary conditions which are typical in fluid dynamics. For example in figure 5 we consider a flow past a plate (Hannon *et al.* 1986*a*) (upper insert) or past a cylinder (Rapaport *et al.* 1986; Hannon *et al.* 1986*b*, 1987) (bottom insert). Our simulations were performed on *ca.* 10^5 particles for events in the picosecond to nanosecond range over a scale of a few hundred ångströms. More precisely, as an example, the parameters characterizing the flow past a cylinder simulation are reported as follows: (1) number of particles is 160000; (2) diameter of obstruct is 250 Å; (3) soft-sphere potential ($\sigma = 3.4$ Å, $\epsilon = 120\ \mathrm{K_B}$); (4) gravitational acceleration is $7.1 \times 10^{13}\ \mathrm{cm\ s^{-2}}$; (5) cell size is 25 Å × 25 Å; (5) number of steps is 120000; (7) system size is 1500 Å × 1500 Å; (8) periodicity in x-, z-directions; (9) density is $1.42\ \mathrm{g\ cm^{-3}}$; (10) time step is $0.93 \times 10^{-15}\ \mathrm{cm\ s^{-2}}$; and (11) properties by averaging over particles, in each cell every 5 steps.

The simulated flow past the cylinder can be typified by the following evolution (1) $T = 0$, the uniform flow changes rapidly into Stokes flow around the cylindrical obstacle, (2) $T = 100$ ps, pair of counter-rotating eddies develop adjacent to the down stream edge, (3) $T = 260$ ps, eddies develop to about 1.4 times the diameter of the obstacle, (4) $T = 310$ ps, right eddy disappears, (5) $T = 370$ ps, again two eddies, (6) $T = 470$ ps, left eddy disappears, (7) $T = 610$ ps, oscillatory wake develops and (8) $T = 1000$ ps, eddies vanish and sinuous wakes appear. We have reported that such systems scale extremely well and very accurately from results and laboratory experiments of system at the square centimetre size and with a lifetime of seconds. For example, the Reynolds number at which we observe wakes past the cylinder is about 25 ± 10, i.e. essentially equal to 35, the Reynolds number for which laboratory data observed the same behaviour (Rapaport & Clementi 1986).

Equally gratifying are the results of figure 6 where we report on Bénard type flow obtained by considering 160000 argon atoms (Hannon *et al.* 1988). This rather astonishing accomplishment (even if preliminary, because we have not obtained true Bénard patterns) can be appreciated by comparing figure 6 with Bénard patterns observed in laboratories at a trillion times larger scale shown in figure 7; another most gratifying feature is that the Bénard patterns have been obtained from physically meaningful values for the particle's mass (alternatively one could have varied the flow parameters, considered as numerical rather than physical variables). These results are exciting, but, in the context of this paper, the most relevant aspect is that indeed our 'global method' is verified, starting from systems with few nuclei and electrons up to vastly larger systems, previously described only by fluid dynamics.

The above study on a liquid is an example of the global-simulation approach; we did not

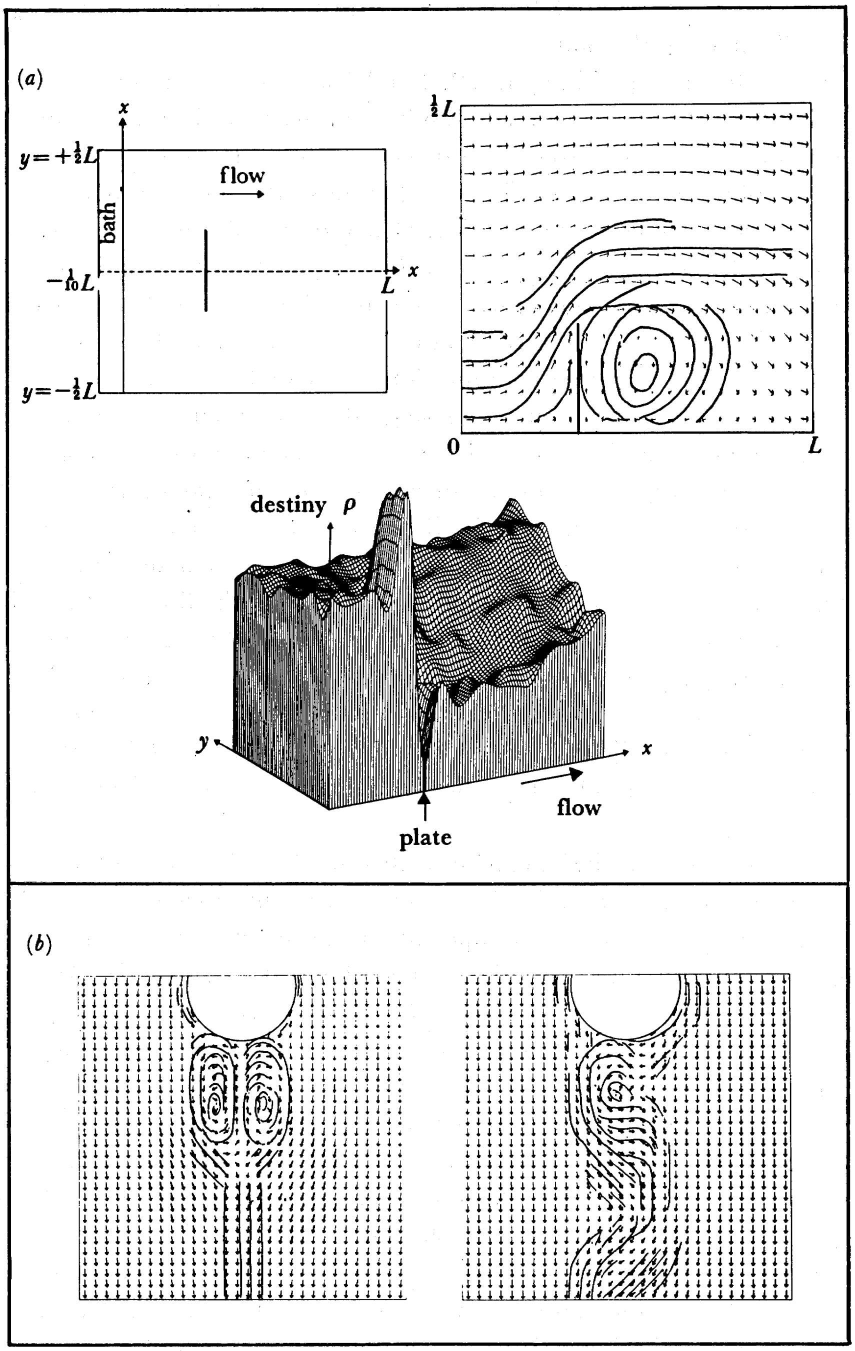


Figure 5. (*a*) Top: schematic diagram of a flow past a plate (left) and visualization of velocity field for steady state. Arrow lengths are scaled as the square root of the ratio of individual cell velocity to maximum cell velocity. Bottom: density contour for steady-state flow. (*b*) Velocity field for flow past a cylinder; eddy formation is shown.

work on water to have a show test case for the global-simulation methodology. Our interest in water relates to our work in biophysics and biochemistry: as is well known water is the most abundant solvent on earth and the medium to carry life. Let us briefly report on our simulations on water surrounding DNA either in the B- or Z-conformations with Li^+, Ne^+ and K^+ counterions. From these simulations, done both at the Monte Carlo and molecular

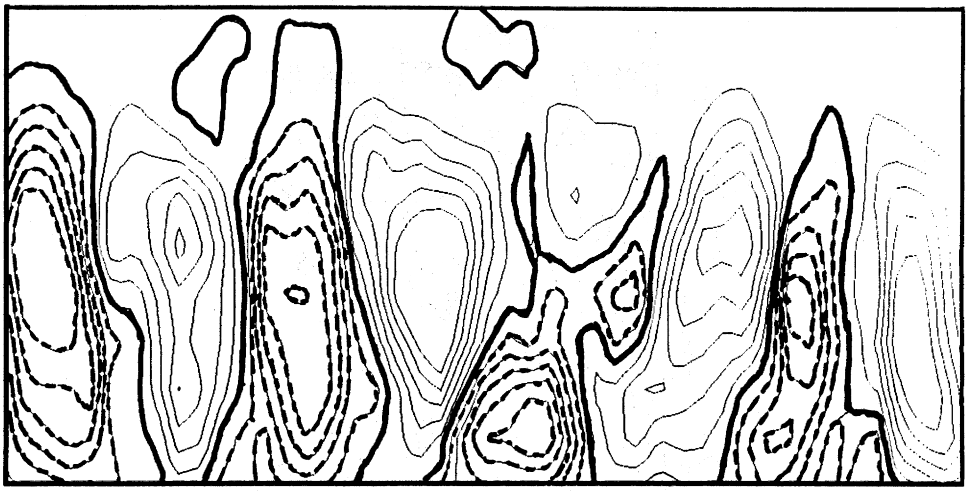


FIGURE 6. Simulated Bénard flow obtained from 200000 neon atoms. Temperature of hot wall, 486.5 K; temperature of cold wall, 86.5 K; system size, 836 Å × 3344 Å; periodic in x-direction; soft-sphere potential with $\sigma = 3.4$ Å and $\epsilon = 120\ k_B$; gravitational acceleration, 4×10^{12} cm s^2; solid line, clockwise rotation; broken line, anticlockwise rotation; bold line, zero contour.

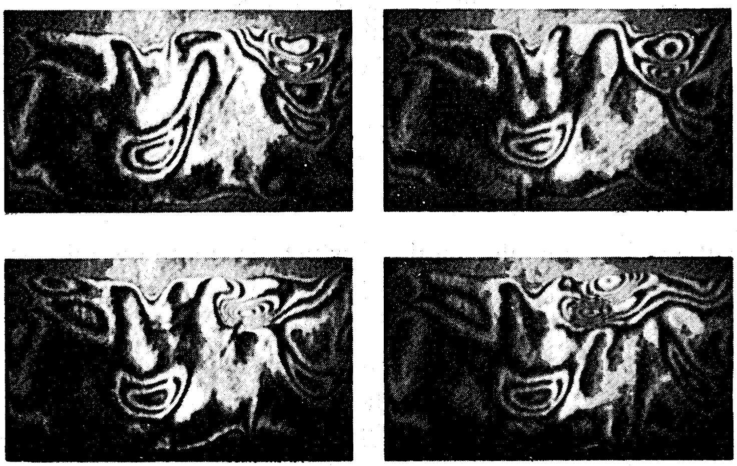


FIGURE 7. Laboratory Bénard-flow, taken from a film, with the permission of P. Bergé and M. Dubois, C.E.N. Saclay, Gif-sur-Yvette, France.

dynamic levels (Clementi & Corongiu 1979*a*, *b*; Corongiu & Clementi 1981; Kandadai & Clementi 1987) we have determined the number of water molecules hydrating the phosphate and the sugar groups as well as the base pairs, extending on the classical papers by Lord in the early 1960s (Falk *et al.* 1963). We have also discovered that a chain (or filaments) of water molecules (one hydrogen bonded to the other) runs within the minor groove in B-DNA, as shown in figure 8.

This finding was later confirmed in the X-ray experiments of Dickerson (Dickerson *et al.* 1981). Strictly speaking this finding is not unexpected; indeed water organizes itself into filaments whenever in contact with strongly attractive sites, as we have shown for solvated ions since 1975 (Fromm *et al.* 1975; Clementi & Barsotti 1978). The filament-like structure differs from the clathrate-type organization of the molecules of water characteristic for bulk water and for water near a hydrophobic site. We note that the length of the filament is proportional to the attraction forces of the hydrated site and the filaments have a measurable lifetime longer,

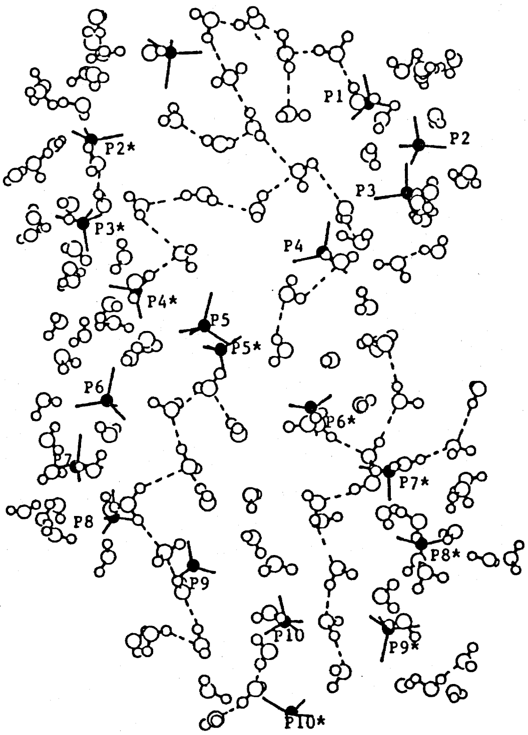


Figure 8. Pattern of water molecule chains in B-DNA. The hydrogen bonds are explicitly shown (dotted segments) and so are the phosphate group; the sugar and bases are omitted.

for example, than the water's angular motions, which originate the infrared libration bands. Thermal motions clearly tend to disrupt the filament's organization and cooling down the water or increasing the concentration with appropriate solutes will make these structures more permanent. To conclude, the water medium surrounding DNA for the first few layers is very different from bulk water. We also have predicted that the counterions select a well-defined pattern around the DNA helices as indicated in figures 9 and 10 where we also present the water radial distribution functions for B-DNA and Z-DNA respectively. From our simulations we learn that the ions have relatively low mobility (at least those ions which equilibrate the negative charges of the phosphate groups) at least for the few picoseconds of our simulation.

Thus the nucleic acid field experienced by chemicals in solutions (and *in vivo*) is much richer than the one we might envision using only X-ray data. The process of recognition of the genetic code clearly is affected by the water and counterions structure. In this context we turn to another simulation (Clementi & Corongiu 1981) and move a glycine alongside DNA keeping it parallel to the vertical axis of B-DNA; the simulated glycine interaction with water, counterions and DNA is reported in figure 1. Notice that the glycine is still rather far from DNA; by moving it closer, the glycine–water and the glycine–counterion interaction curves would become more and more structured.

Clearly, this is an idealized experiment, because we have constrained glycine's motions to be vertical, alongside B-DNA, rather than considering many trajectories. On one hand, the simulation is computationally very expensive even with supercomputers. On the other hand the pay-off could be most notable; indeed we would learn on 'the energetic cost' for a molecule

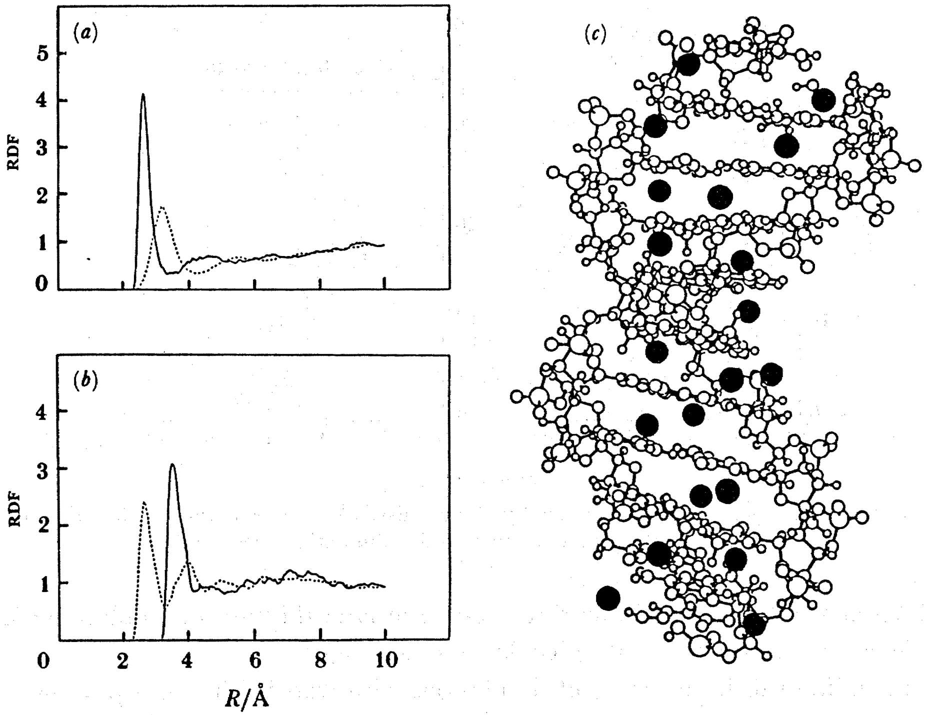


FIGURE 9. B-DNA. (*a*) Radial distribution functions for counterions–water. Solid line, ion–oxygen; dotted line, ion–hydrogen. (*b*) Radial distribution functions for phosphate–water. Solid line, PO_4–oxygen; dotted line, PO_4–hydrogen. (*c*) *xz* Projection of B-DNA with K^+ counterions.

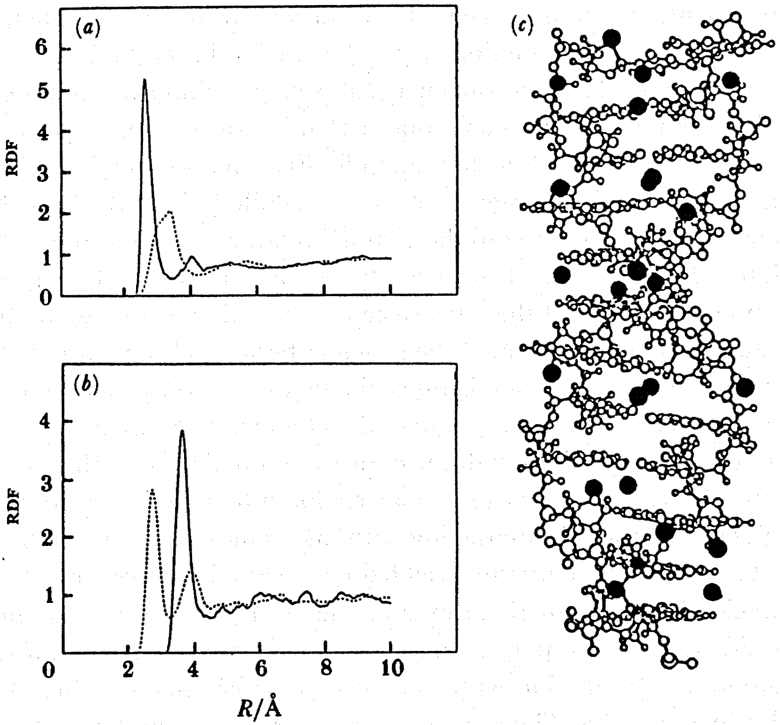


FIGURE 10. Z-DNA. (*a*) Radial distribution functions for counterions–water. Solid line, ion–oxygen; dotted line, ion–hydrogen. (*b*) Radial distribution functions for phosphate–water. Solid line, PO_4–oxygen; dotted line, PO_4–hydrogen. (*c*) *xz* Projection of Z-DNA with K^+ counterions.

PHILOSOPHICAL TRANSACTIONS OF THE ROYAL SOCIETY A MATHEMATICAL, PHYSICAL & ENGINEERING SCIENCES

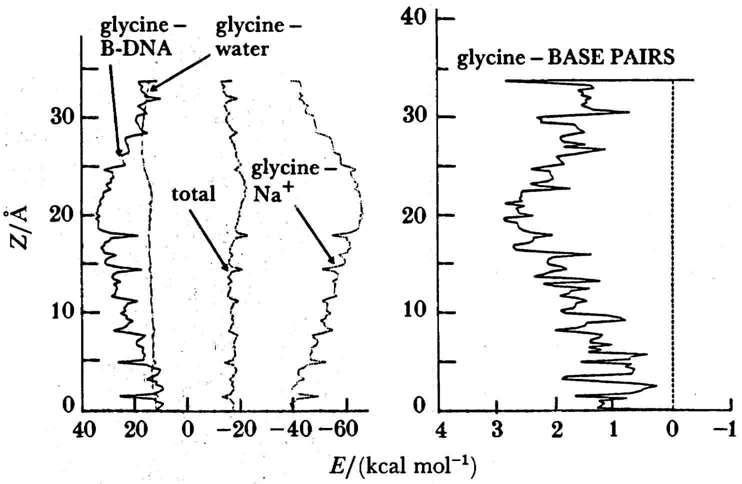


Figure 11. Interaction energy of glycine moving along a B-DNA turn in solution with Na$^+$ counterions. This is the energy needed 'to read few lines of a genetic code'.

when reading molecular genetics blueprints. Sooner or later this type of simulation will become a 'must', being one of the keys for molecular engineering.

Before concluding this brief survey of simulations with water either as a pure liquid or as a solvent, we shall touch on the very broad chapter of water interacting with ions. For alkali and halogens we refer to an early monographic work (Clementi 1976), here we shall comment on monovalent ions passing through gramicidin A, a transmembrane channel. Monte Carlo computations (Kim & Clementi 1985; Kim *et al.* 1985) did show that the ions coordination number remains essentially the same either within or outside the channel, thus making it reasonable to assume that the interaction energy for the ion in solution outside the membrane is not drastically different relative to the energy of the ion within the channel. However, our early computations did yield an interaction potential deeper than must be assumed to explain the transport mechanism (see, for example, Eisenman & Sandblom 1984). The same unsatisfactory feature, actually stronger, is present in Pullman's work (Etchebest *et al.* 1984). By using our potentials the attraction to the channel is large enough as to prevent the ion from passing through the channel, even when we apply a strong electric field. Thus we turned to our interaction potential and realized that the 'acceptable' fitting error (about 10% at the most attractive minimum) was at the origin of the problem because of the error's additivity for each one of the many amino-acids residues forming the channel. To bypass the problem, we have recently obtained a new interaction potential (Gomperts & Clementi 1987), this time computing the interaction of the ion with the entire macromolecule, rather than the individual residues. In figure 12 Kim's results are reported for interaction potentials fitted from the computed energies of Na$^+$ (or K$^+$) interacting with the amino-acids which build up gramicidin. The addition of these partial interactions yielded the energetic profiles of figure 12 for the ions with either gramicidin (in the dimeric or monomeric forms) or with gramicidin stripped of the residue amino-acid, thus in the polyglycine back-bone form. Massive and direct quantum-mechanical computations of the ion with the polyglycine back-bone bring about the two top energetic profiles in figure 12. These computations are less demanding than the quantum mechanical bench-mark computation of GA with K$^+$ but must be repeated many times for different positions of the ions (a few hundred points are needed). This work was done on

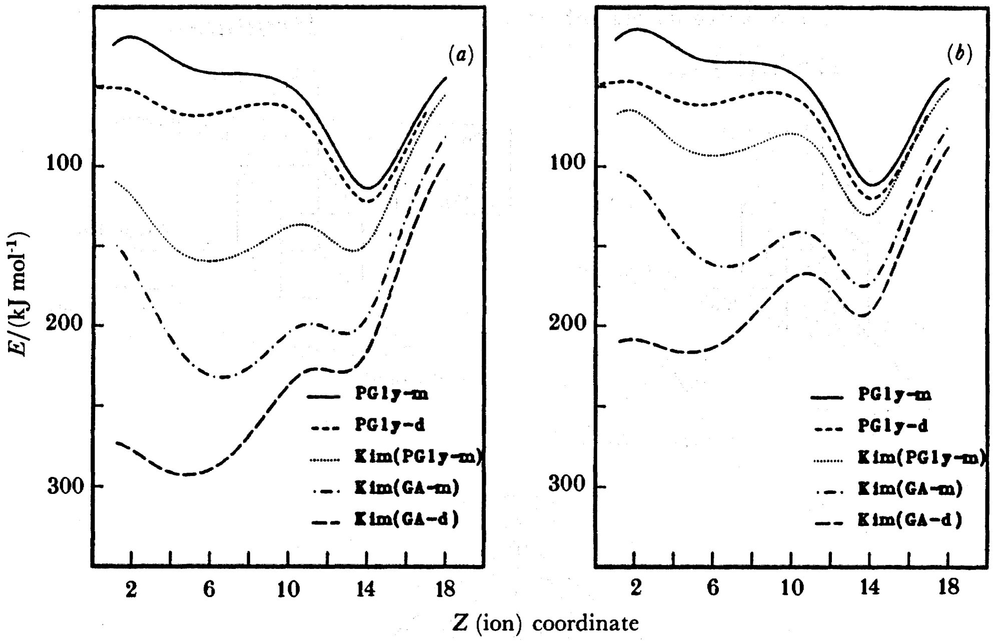


FIGURE 12. Interaction energy profiles for (*a*) Na^+ and (*b*) K^+ ions moving through a gramicidin channel (centre of channel at $Z = 0$ Å, beginning at $Z = 14$ Å). Top two profiles result from massive computations with the entire macromolecule; bottom three are obtained assuming additivity of the interactions with residues.

LCAP-2 one of our experimental parallel supercomputers. As explained below, in the section dealing with computer architecture, this simulation did require a massive computation, on a scale previously never reported in computational chemistry. A most interesting feature of the simulation is that when the ion is sandwiched in between two molecules of water, then the ion moves through the channel, whereas it only rattles about the initial position of the channel when the two water's molecules are absent. The waters undergo continuous rotations and librations concerted in such a way as to pull and push the ion through the GA channel. In closing this short note we cannot but wonder how much we are missing in modelling molecular recognition by neglecting, as is often done, solvents and dynamics. Let me stress, however, the non-negotiable need of supercomputer availability on a massive scale.

The tool to perform our simulation is an experimental supercomputer which has been assembled in our department. Let us recall then when we operate on numbers, we can select between either the scalar, S, or the vector, V, or the parallel, P, modes of operation. Let us consider S, V, P like vectors of a three-dimensional space. Any computation is then associated to a trajectory within the S, V, P space, as shown in figure 13. For complex algorithms, the corresponding trajectories often will occur within a preferential volume in the S, V, P, space, thus defining an envelope, characteristic of the given algorithm. The ideal computer for an algorithm's is the one which has high-performance for the algorithm's envelope in the S, V, P space. For global simulations the ideal computer must have super-performance along each one of the three dimensions, namely, S, V, P being most unlikely that any simpler choice will suffice. Alternatively stated, the envelope for global simulations extends over most of the S, V, P space. In the early 1980s, when we extended our simulations to molecular dynamics and fluid dynamics, there was no such supercomputer; henceforth the need for us to build one.

Our first attempt resulted in the Loosely Coupled Array of Processors mark 1 (LCAP-1)

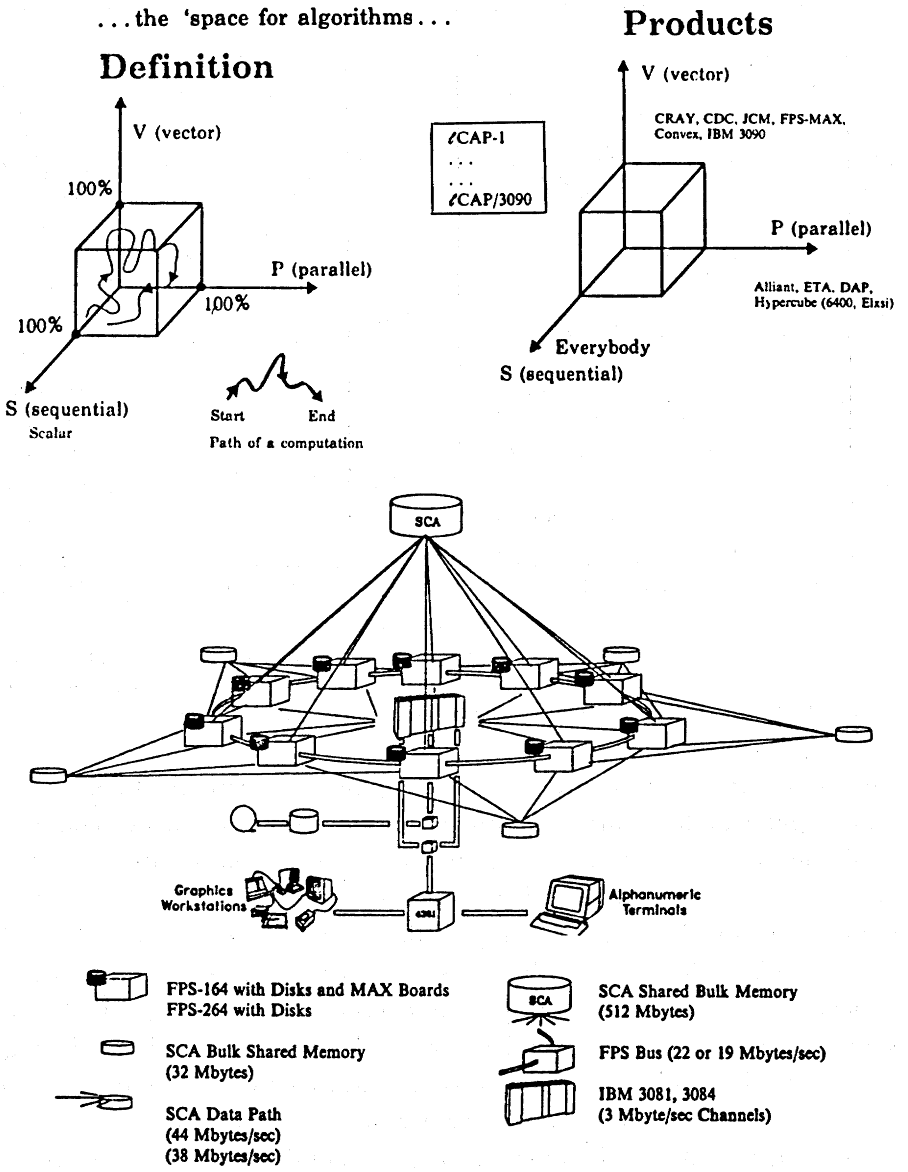


Figure 13. LCAP-1 and LCAP-2 supercomputers (see text) and the SVP space.

consisting of ten FPS-164 array processors each one with local disk packs and eight megabytes of random memory; two vector accelerators (Max-Board) bring the FPS-164 peak performance from 11 to 55 megaflops. These ten nodes are individually connected to an IBM mainframe via standard IBM channels. In addition five memories, each one of 32 megabytes, connect (with a high transmission rate of 44 megabytes s^{-1}) four successive nodes, forming a double ring connectivity; further a large memory of 512 megabytes connects each one of the ten nodes. Our computer has about 550 megaflops peak performance. A second system, LCAP-2, was constructed with FPS-264 replacing the FPS-164; this second system has about 380 megaflops peak performance. The two configurations were connected for experimentation in 'cluster to cluster' parallel computing and the resulting system – with about one gigaflop performance – is called LCAP-3 and is reported in figure 14.

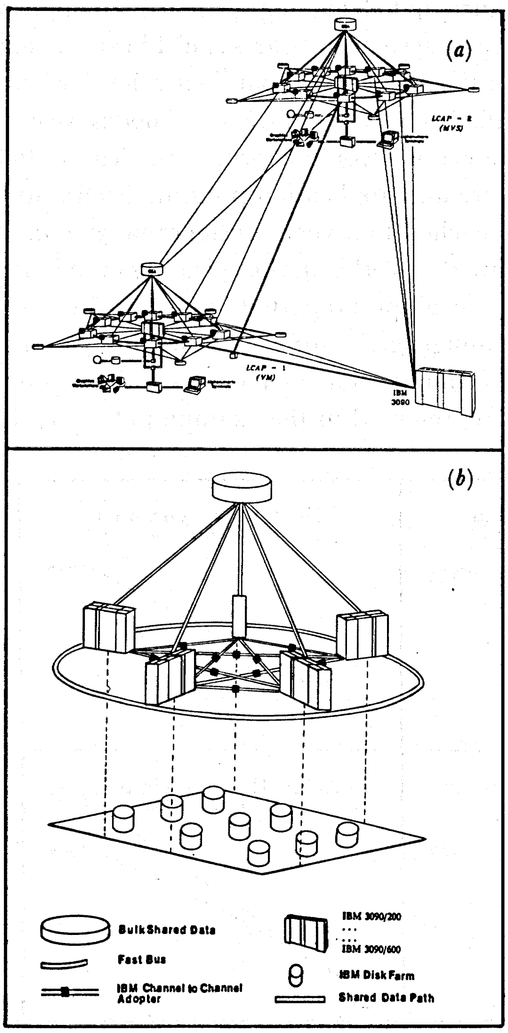


FIGURE 14. (*a*) LCAP-3 and (*b*) LCAP-3090: the ten nodes in each one of the two clusters of LCAP-3 are explicitly drawn; for LCAP-3090 the nodes of each cluster are compacted into one box.

The power of LCAP-3 can be exemplified by recalling that an *ab initio* SCF computation for gramicidin and a K^+ ion required only *ca.* 11 h of elapsed time; we note that (1) this system has 276 atoms with 1010 electrons; (2) we used 2736 primitive gaussian functions; (3) the SCF threshold was set at 10^{-5} a.u.; (4) only integrals smaller than 10^{-5} a.u. were neglected, all the others, for a total of 433646989, were computed in double precision and stored on disk-packs (working in parallel). For additional details on the LCAP system and performance relative to other commercial supercomputers, we refer elsewhere (Clementi *et al.* 1986*a*; Clementi & Logan 1987).

We are now assembling one more supercomputer, called LCAP-3090; in this system each cluster is an IBM-3090 system that can have up to six nodes (IBM-3090/600 with 6 vector facilities); we are considering up to five clusters. (At the bottom of figure 14 we have shown an LCAP-3090 built up by IBM-3090/200 clusters.) The advantages of LCAP-3090 over LCAP-3 are (1) one single vendor for all hardware and softwares, (2) very high hardware and system software reliability, (3) portability of application programs relative to LCAP-1 and

LCAP-2, (4) peak performance up to 3 gigaflops, (5) growth opportunity starting from a single node and (6) vector hardware, system softwares and library availability.

In figure 15 we report on the efficiency in parallelization for several (12) application areas. The optimum line (straightline) states that if we use N nodes we expect at best an N-fold speed up relative to $N = 1$. Of course one would expect some degradation due to overhead both at the system software, hardware and application program levels. Still for the many applications reported, the ideal goal is reached to a very considerable extent. We expect that the LCAP-3090 will cut down even more the overhead, thus yielding even better parallelization. For more detailed references on the 12 examples reported in figure 15 we refer to Nguyen *et al.* (1985), Clementi *et al.* (1985*b*), Domingo *et al.* (1985), Clementi & Corongiu (1985*c*), Clementi *et al.* (1986*b*). Notice that most of computational chemistry and very important areas in physics and engineering are essentially represented in the examples of figure 15.

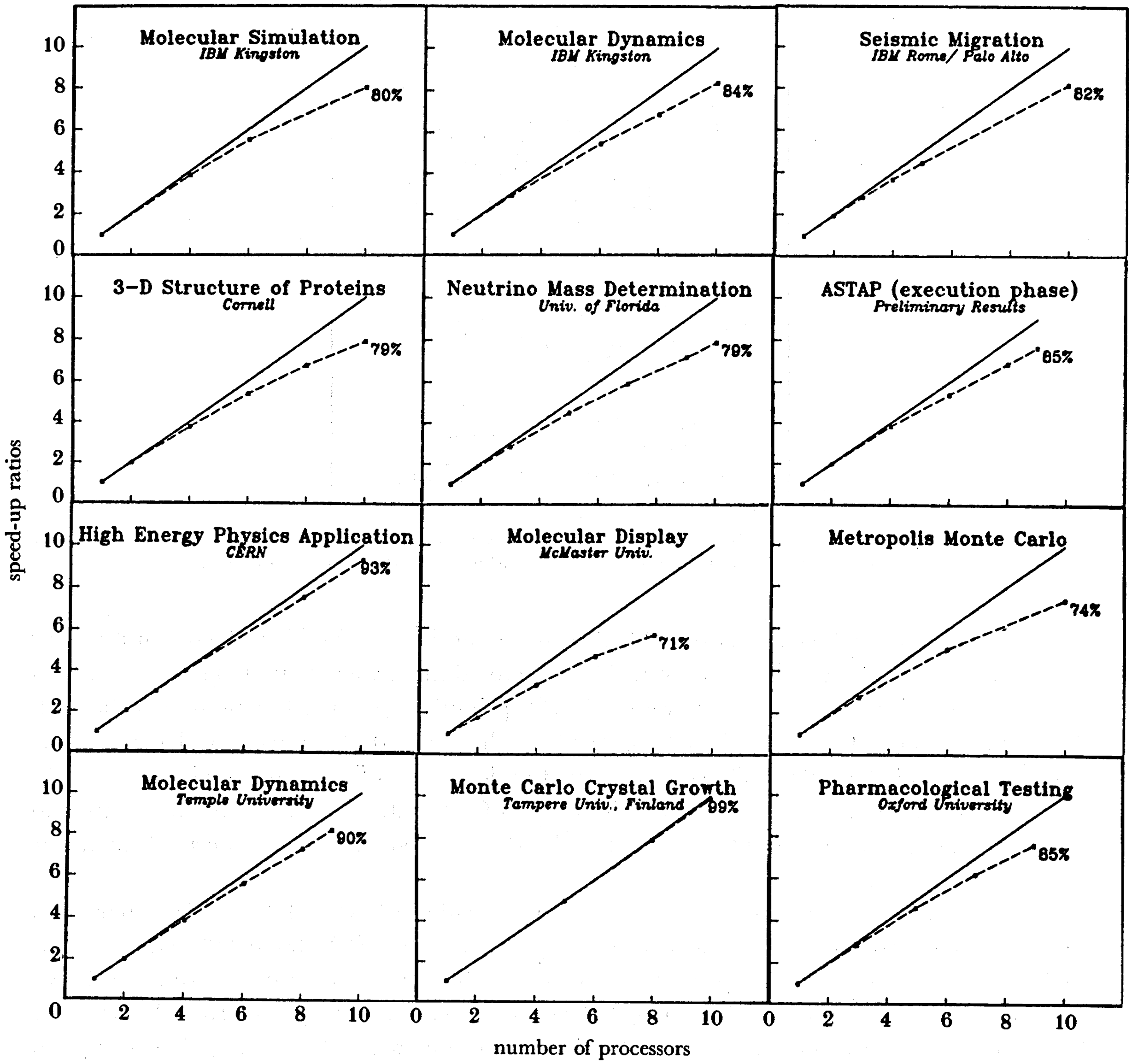


FIGURE 15. Percentage parallelization in simulations for science and engineering.

The 12 examples reported in figure 15 have few characteristics in common. To start with, the codes were written originally for sequential computers and the parallelization was obtained by our 'first-generation' precompilers (1983–1985 vintage). The reported 'speed-up' (namely the ratio T_n/T_1, where T_1 is the elapsed time for execution on 1 node (or AP, array processor) and T_n is the elapsed time for the same job executed in parallel on n nodes) are obtained considering the full job, not only kernels; thus input–output timing is included. Alternatively stated, the examples correspond to computations, executed to elicit interest in a scientist and/or an engineer rather than in a computer scientist or a hardware computer architect. In general, we have learned that *grosso modo* 8–12 FPS-164, arranged as in LCAP-1, exhibits the performance of one CPU of a CRAY XMP machine with a solid-state disk. Because in the reported examples (as in most of our work) we have made no use of the FPS matrix accelerators (Max Boards), the measured LCAP-2 performance is three to four times better than the one above measured with LCAP-1. We recall that the FPS-264 performance is three to four times better than the FPS-164, thus our previous statement is an expected one for simulations with high degrees of parallelization.

The above 12 examples did require little code modification for the translation from sequential to parallel and no algorithm modification. Indeed, the above examples correspond to what we refer to as 'natural parallelism'. Very roughly, a naturally parallel code mainly consists of either of a set of nested loops (the larger the set, the more naturally parallel) or of only one loop but with intensive computations at each loop step (i.e. large granularity). Notice that for the set of nested loops, the innermost loop often is vectorizable, and in this case we have optimal computational conditions, namely easy vectorization and easy parallelization. Quantum and classical many-body dynamics and also quantum or classical stochastic problems are most often naturally parallelizable. Other natural parallel codes are those that simulate computations easily 'partitionable' in subtasks. An interesting example of the latter is 'graphics', where clearly the rotation and hidden line computing of a complex image can just as well be obtained by operating with many parcels of the image all at the same time.

'Natural' parallelism is often found in complex solutions because it tailors well the scientific method. Indeed, for a complex problem most often we decompose complexity, by solving first for only some of the variable (the most relevant one) and then by coupling the obtained partial solutions either with perturbation or relaxation techniques or appropriate many-body expansions.

However, codes and algorithms are not always naturally parallel. In such cases it is basic that each node communicates efficiently with other (either few or many) nodes. This can be accomplished poorly with today's slow channels but rather efficiently with busses and bulk shared memories. For this reason, from 1985 to 1987 we have devoted much attention to these communication devices, which change our LCAP into TCAP, i.e. we go from loose to tight coupling. (For details see FPSBUS Software Manual, Release G, Publication no. 860-7313-004A, 1976, Floating Point System, Inc., Beavertown, Oregon, U.S.A. (1986) and *shared bulk memory system software manual*, Version 2.0, Scientific Computing Associates, Yale, New Haven, Connecticut, U.S.A. (1987).)

Before discussing these special devices, let us report on an application which makes use of the Max boards. The Max boards can be used either by calling a library of special subroutines or by direct coding in FPS (assembly language APAL); the latter represents a far from trivial undertaking, in general. Recently the Max boards have been used in the parallelization of a quantum chemical code which approximates the electronic correlation correction by using second-order Moller–Plesset perturbation (M.P.2). The code is one of the special features of a general molecular package called HONDO (Dupuis *et al.* 1980) and the work is reported in detail elsewhere (Watts & Dupuis 1987). Below we report the execution time on the LCAP-1 for a molecular problem with 120 contracted gaussian functions. The timing refers to the two-electron transformation process and is given for the following: (A) no Max boards, (B) one Max board per attached processor (FPS-164) and (C) two Max boards for each processor. We report for parallelization on 1, 2, 4 and 6 processors respectively. The parallelization efficiency is given followed by the obtained megaflop rate, in parentheses: (A) 100% (9.2), 95.5% (18.2), 98.8% (36.2) and 96.5% (53.2); (B) 100% (15.3), 99.9% (30.6), 98.8% (60.6) and 96.8% (89.0); (C) 100% (23.8), 99.3% (47.3), 98.2% (93.5) and 96.2% (137.5).

The FPSBUS$^{\text{tm}}$ software is based on a message-passing scheme; the user has to specify from which end to which processor data are sent and received. If the same data have to be sent from one processor to all the others, then the broadcast mode can be used.

The facilities provided by the precompiler for using the bulk memories offer a choice of two modes, the shared and the message passing mode. In the shared mode the data to be shared is stored in the memory at locations known to all the processors, and it is passed between processors by specifying a write of data from one processor by a read of data by another one. In the message passing mode, messages, in the form of contiguous strips of data, are passed between processors via a network of paths transparent to the user, who specifies in his code the 'send' and the 'receive' transactions along with the targeted processors.

Performance measurements have been done using as processors either the FPS-164 or the FPS-264, exploiting the three communication schemes above described and choosing the patterns of data transmission given in figure 16. Figure 16*a* has been called STAR: in this configuration one processor (considered the master) sends data to the others (the slaves); (*b*) has been called RING (RNG for short): in this case each processor sends and receives data from its two neighbouring processors; (*c*) has been called LINE (LN for short): this configuration can be achieved by cutting the RING configuration, so that two end processors have only one neighbour.

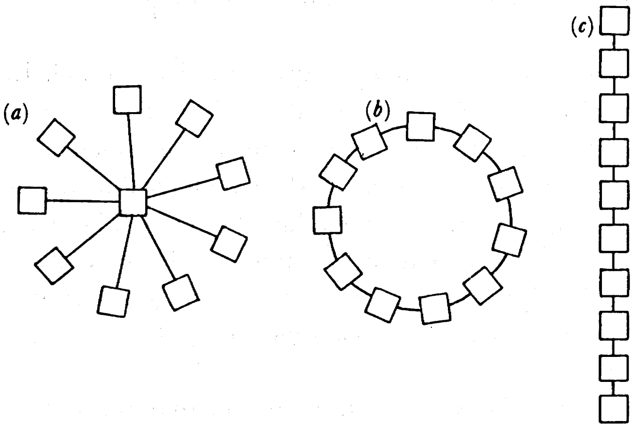


FIGURE 16. Star, ring and line configuration of nodes.

Measurements have been performed with different numbers of processors in the configurations and a variety of data transmission sizes. In the top two insets of figure 17 we report the results for the minimum number of processors used for each configuration (two for (*a*) and three for (*b*) and (*c*)), whereas in the bottom insets the curves are for configurations including eight processors. The figure presents the measured transmission rate against the number of bytes transmitted in a transaction; data size ranges from 8 bytes (one word) up to 800 K bytes (100 K words).

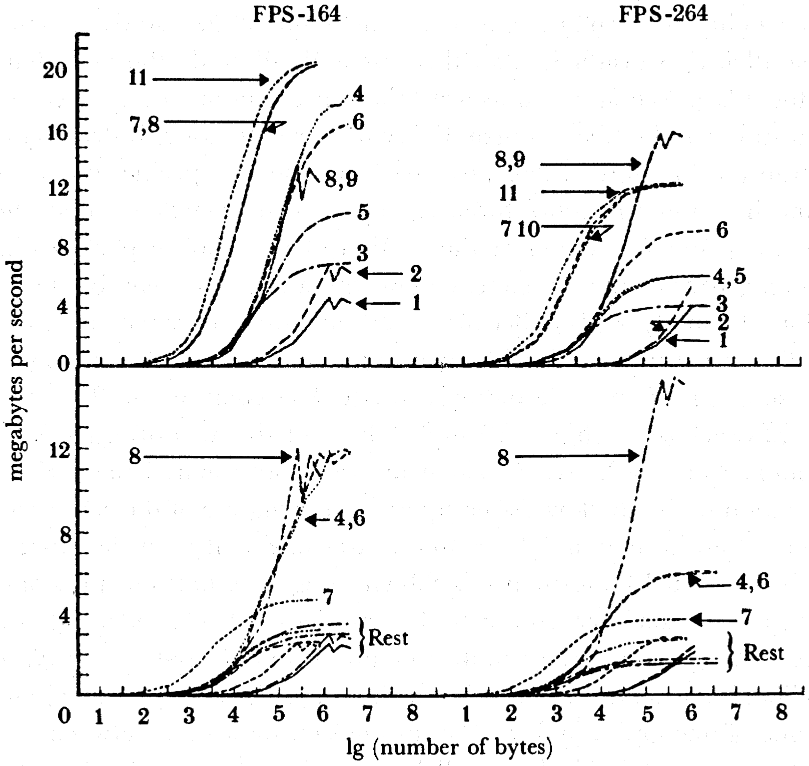


FIGURE 17. Data transfer rates from FPBUS and SCA$^{\text{tm}}$ bulk shared memories. The full bandwidth can be obtained by multiplying the above transfer rates by the number of nodes active in a transaction. 1, RNG BUS; 2, LN BUS; 3, RNG SM; 4, RNG MP; 5, LN SM; 6, LN MP; 7, BROAD SM; 8, BROAD BUS; 9, STAR BUS; 10, STAR SM; 11, STAR MP.

The notation for each curve in figure 17 has to be interpreted as follows: first the configuration is specified, either RNG or LN or STAR or BROAD (if the broadcast mode is used); then the device used for the communication is given, namely, BUS means that the transaction occurs through the FPS-BUS; SM is the short notation for the bulk memories in shared mode and MP for the bulk memories in message-passing mode (Corongiu *et al.* 1987).

Two features are common to the experiments of figure 17, namely the notable initialization delay (latency) time (note how slowly the curves rise) and the drastic degradation in passing from 2 to 8 nodes (compare top curves with bottom ones). Concerning the degradation one can see that often one reaches transmission rates of about 3 megabytes s^{-1} namely transmission rates typical of peak performance for channels. Let us exemplify for the star configuration using the bulk memory in shared mode on LCAP-1 (i.e. with FPS-164 nodes): 22 mbytes of data are transmitted in one second for one-to-one node transmission but only about 3 mbytes of data are transmitted in one second for a 1 to 8 node transmission. On LCAP-2, with the FPS-264 the corresponding values are about 13 mbytes and again about 3 mbytes (the memory characteristics are responsible for the degradation between 264 and 164). These data point out the need of optimal softwares to decrease latency and degradation.

An interesting example on these communication devices reports on the 'speed-up' for Metropolis–Monte Carlo study of water using 2- and 3-body potentials for a box containing 512 particles (Corongiu *et al.* 1987). The net conclusion of this study is summarized in figure 18; loose parallelism is insufficient for this case and, all in all, the bulk shared memory is preferable to the BUS. We stress, however, that these timing examples should not be extrapolated to machines and software much different to the one used. Indeed for example the latency time for transmission of data depends very much on the specific system software and hardware that one has. Let us report three examples of intermediate, fine and very fine granularity. The study and prediction of the motion of waterborne particles in bay areas subject to tidal forces, is critically dependent upon the ability to solve for the tide-induced residual circulation (Csanady 1982). Because these features are typically small scale, the number of grid cells needed to resolve the flow even in modest domains can be very high. We present here the solution of the tide-induced residual circulation in Buzzard's Bay and Vineyard Sound, Massachusetts (figure 19 top). The grid size was of order 200 × 200. The solutions were obtained on the LCAP-1 system by using bulk shared memories by a finite-difference discretization of the shallow-water equation making use of the conservation of mass and momentum of a fluid in motion. The solutions are consistent with the observed mean at several locations. The parallel performance is shown in figure 9 bottom; previous experience with the same program without using the shared memories resulted in a 'speed down', i.e. the computation time increased with the number of processors (Capotondi *et al.* 1987). This simulation exemplifies intermediate granularity with a ratio T_c/T_t at least 100 (where T_c is the computation at time of one grain, and T_t the transmission data time between two grains).

Another and related application is the parallelization (Hsu & Sonnad 1987) of a numerical model for ocean circulation (Haidvogel 1981). The model is based on the primitive equation in time and in the three space dimensions. The equations of motion are derived from conservation of momentum, mass and thermal energy. The fluid is assumed incompressible; bottom topology is included. The model is cast in the form of seven partial differential equations which form the complete set of governing equations. For the numerical benchmark the horizontal grid is 21 × 80 and the vertical structure is represented by the barotropic mode and the first four vertical baroclinic modes. A fraction (23%) of the computation is inherently sequential. The remaining is parallelizable by partitioning the computational domain in one of the horizontal directions, and storing the data in the bulk shared memory. With the 10 processors of LCAP-1 a speed up of 68% was experimented and 11% of the computational time was lost as data transfer overhead.

The pseudo-spectral method is a relatively recent method for the solution of partial differential equations, which is gaining in importance because of the very high accuracies achieved with this technique (Orzag 1972). It works by using a dual representation of the variables in the grid domain and wavenumber domain, with continuous transformations between the two representations. High efficiencies in the transformations between the two

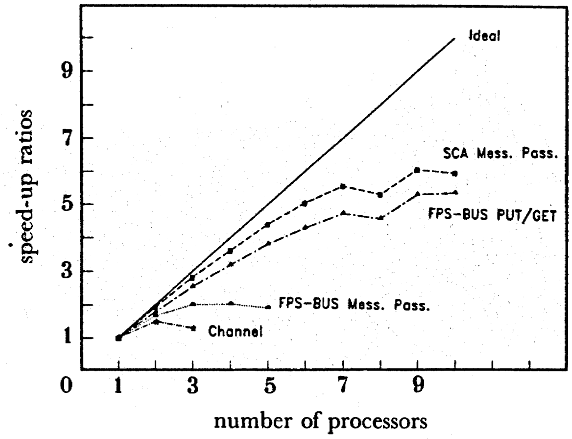


FIGURE 18. Parallelization with either channels of FPSBUS or SCA bulk shared memories. Metropolis Monte Carlo (liquid water, two body and three body).

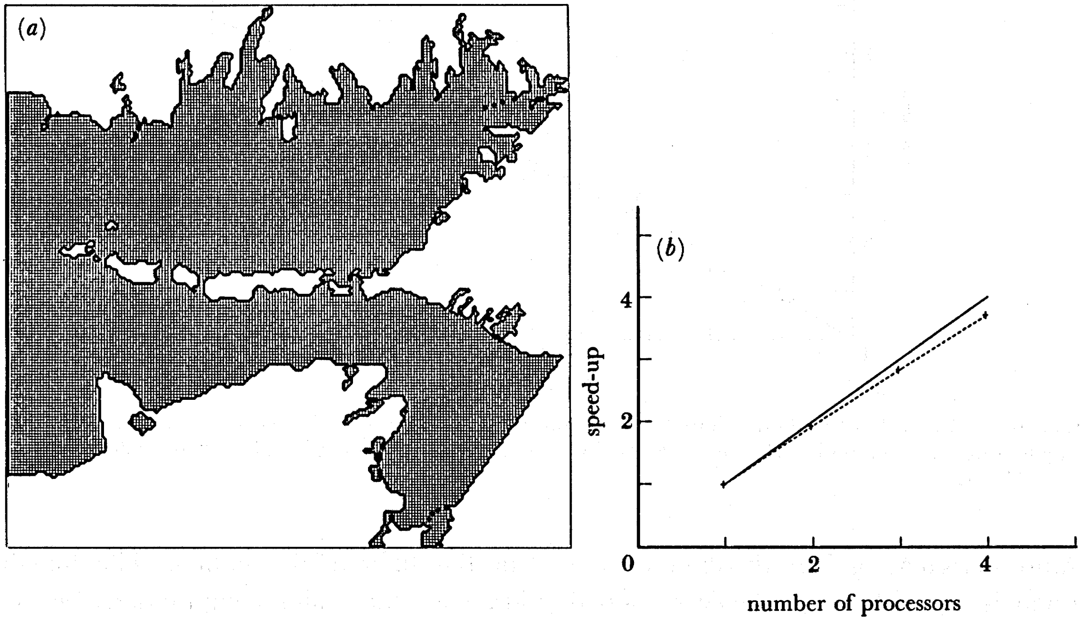


FIGURE 19. (*a*) Computational domain and grid for Buzzard Bay, Massachusetts. (*b*) Speed-up with SCA bulk shared memories: solid line, ideal results; broken line, observed results.

representations, are achieved through the use of fast Fourier transforms which forms the major computational component of the calculations. A 2D program for the solution of a hyperbolic equation by the pseudo-spectral method has been implemented in parallel with a matrix slice approach for performing a 2D FFT in parallel. The problem dealt with advection of an initial concentration of pollutants (shown in figure 20 (top)), in a region with a circular velocity distribution (Christidis *et al.* 1987). The parallel efficiency on the LCAP system with bulk shared memory for various size problems is shown in figure 20 (bottom). Again previous experience without the bulk shared memory showed an increase in running time with increasing processors. This simulation is an example of fine granularity with $T_c/T_t \approx 60$.

The multigrid method is one of the fastest methods for the solution of systems of algebraic

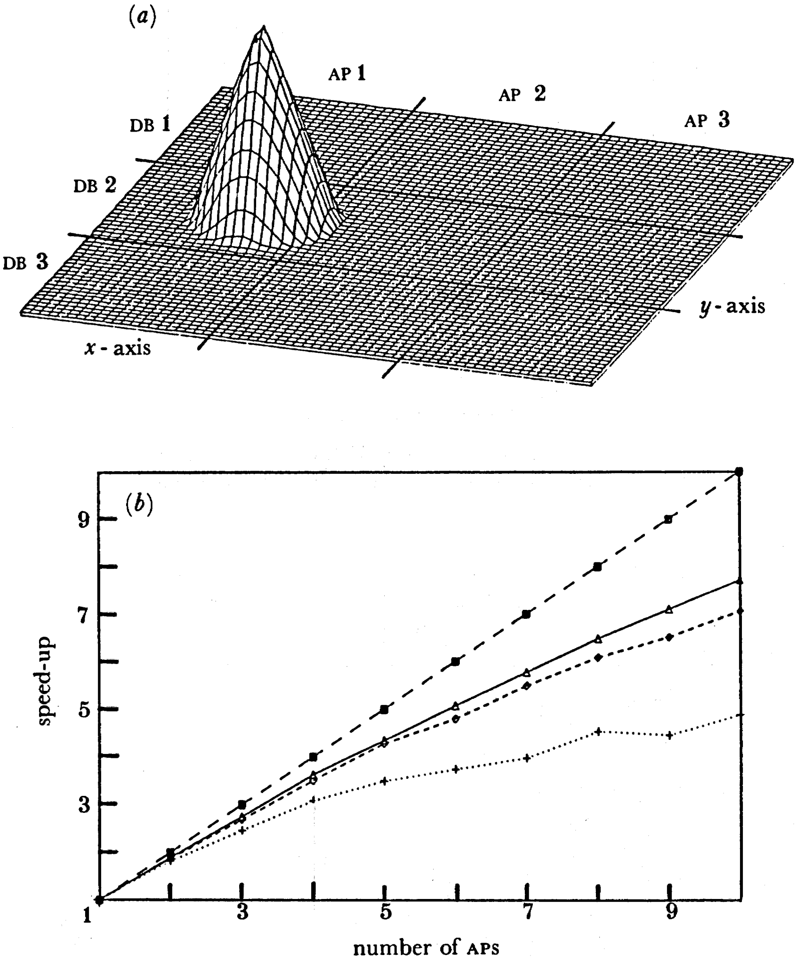


Figure 20. (*a*) Conical distribution at initial time and partitioning of data blocks, DB among nodes, AP; (*b*) speed-up for different grids on LCAP, obtained from SCA bulk shared memory. ■, ideal; △, 256 × 256; ○, 128 × 128; +, 64 × 64.

equations originating from the discretization of partial differential equations. The domain of solution is covered by a scheme of nested grids, with the major computations being the smoothing of the solution, restriction of the residual and prolongation of the solution at various grid levels. The smoothing operation is typically a sequential procedure, but the use of red–black ordering of the nodes, allows parallel implementation of this step and also improves the efficiency of the smoother (even as a sequential operation). Red–black ordering is a process of 'colouring' alternate nodes in the grid; the smoothing is done alternately for the red nodes and then for the black node. The parallel efficiency of the overall process, on the LCAP-1, when used for the solution of Poisson's equation on a square is shown in figure 21. Again it is to be noted that running this algorithm without a shared memory (using the host channels for communication), would result in a slowdown rather than a speedup with increasing processors (Herbin *et al.* 1987). This simulation is an example of very fine granularity with T_c/T_t about 30.

In the previous examples we have considered nuclei, electrons, atoms and molecules, liquids and solutions, simple fluids, and applications to meteorology and oceanography. It seems

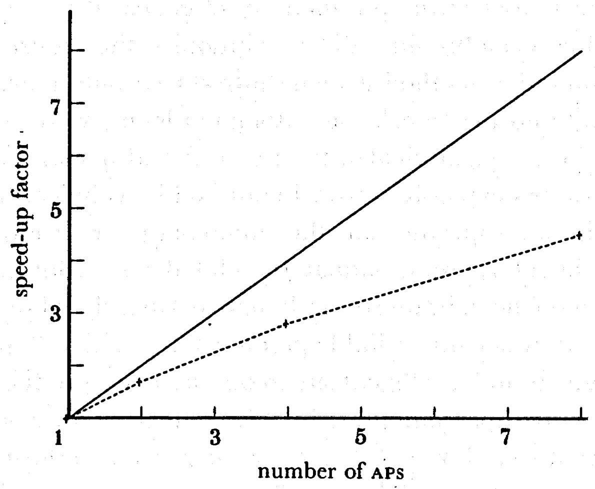


FIGURE 21. Speed-up obtained with SCA bulk shared memories: solid line, ideal results; broken line, actual results.

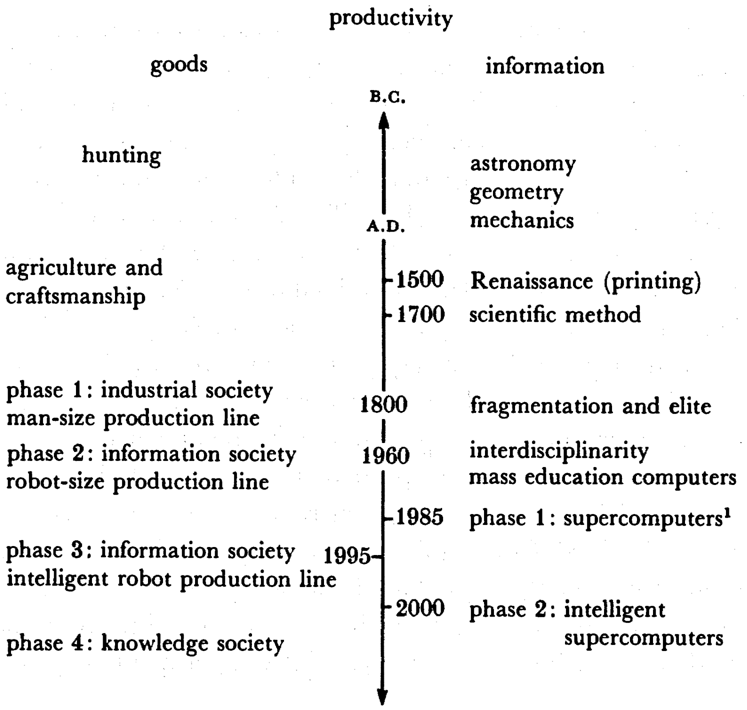
proper to conclude with an application in celestial mechanics. Project Spaceguard is a numerical experiment, carried out by Dr Milani and Dr Carpino (University of Pisa) dedicated to understanding the long-term dynamics of planet-crossing asteroids. It is meant to extend and improve current theory based on analytic and semi-analytic models (Opik 1976) that attempt to predict asteroid planet probabilities of collision. The calculations differ from traditional computations of hierarchical organized planetary trajectories (Milani & Nobili 1983) by recognizing the essential chaotic nature of the asteroids' orbits, i.e. they are associated with positive Lyapounov exponents. The experiment was thus a statistical measurement that involved integrating through classical mechanics the positions of all known, approximately 400, asteroids whose orbits intersect those of Earth, Venus and Mars in conjunction with the seven major planets of the Solar System over a time span of 200000 years.

The computation amounted to approximately 8.5 teraflops and generated approximately 3 gigabytes of output associated with nearly half a million close encounters. The statistical analysis of the output is in progress. These calculations were performed on the LCAP-2 system using 10 APS for approximately 40 h and at a sustained rate of 95 megaflops. The parallelization efficiency corresponded to approximately 90% of linear speedup.

The parallelization strategy involved assigning to each of 9 APS a different set of asteroids to track and to the remaining processor the task of tracking the planets and handling all output to the host. At the beginning of each group of time steps the planet-tracking AP calculated the positions of the planets for a successive group of timesteps and the asteroid tracking APS calculated the positions of the asteroids for the current group of timesteps. Computation was thus both parallel and pipelined. Communication between the planet tracking AP and asteroid tracking APS was implemented through shared memory. Given the parallel grain size this communication would not have been feasible with host channels but would have been possible but not optimal with the fast bus. More details of the experiment may be found in Milani *et al.* 1977).

Let us now return briefly on consideration relating the global simulation modelling to productivity; as it is known the latter, correlates both to goods and to information. The main evolution of productivity in the human history is sketched in scheme 1. The productivity of information trails the main phases of the productivity of goods; it is therefore not surprising that whereas the production assembly-lines did revolutionize the industrial output since the early 1800s, the global simulation method is being introduced only now. We recall that two conditions (market demand and the 'tool', the automated loom) were essential for initiating a revolution in the textile industry. Equivalently, the need and demand of interdisciplinarity of information, essential to avoid expensive errors in our highly technical (and thus risk-prone) society, had to wait for the appropriate tool, the supercomputer, before we could set up a production assembly line for information namely the global simulation approach.

The advances in material sciences, brain research, networking, signal processing, cybernetics and artificial intelligence are sufficiently reliable pointers for confidently predicting intelligent robots for productivity of goods and intelligent supercomputers for productivity of information (see scheme 1). An intelligent supercomputer, in this context, is an engine of one to a few gigaflops sustained performance, thus with much higher peak performance (up to about 100 gigaflops) and with the attributes to hear, talk and, within the limit of expert systems, understand some extremely rudimental language (such as FORTRAN) and to see, display and 'understand' few two dimensional patterns, like digits and logical-mathematical symbols. We recall that pattern and voice recognition and artificial intelligence are now most definitely capable of these tasks, which could notably increase the 'user friendliness' of computer systems and thus enhance the user's productivity. The raw speed requested for the intelligent supercomputer is conservative. Indeed we recall that an LCAP/3090 with 30 nodes (five clusters of IBM-3090/600) exhibits a peak performance of about 3 gigaflops; if we adopt the



[1] IBM starts mass production of IBM-3090 vectors

SCHEME 1. Sketch on same aspects of productivity's evolution. 'Landmark's dates' for productivity of both goods and information and are drawn onto the time axis.

same architecture and softwares. Thus the forecast of an intelligent supercomputer within this century is most reasonable.

These ideas are fascinating, but we feel a need to conclude with a hope and a wish, namely that information will evolve into knowledge.

References

Alder, B. J. 1964 *J. chem. Phys.* **40**, 2724.

Bartlett, R. J., Shavitt, I. & Purvis, G. D. 1979 *J. chem. Phys.* **71**, 281.

Belford, D. & Campbell, E. S. 1987 *J. chem. Phys.* **88**, 7013.

Benedict, W. S., Gailer, N. & Plyler, E. K. 1956 *J. phys. Chem.* **24**, 1139.

Capotondi, A., Signell, R., Beardsley, R. & Sonnad, V. 1987 IBM Tech. Rep., KGN-132.

Christidis, Z. D. & Sonnad, V. 1987 IBM Tech. Rep., KGN-143.

Clementi, E. & Popkie, H. 1972 *J. chem. Phys.* **57**, 1077.

Clementi, E. 1976 Determination of liquid water structure, coordination numbers for ions and solvation for biological molecules. In *Lectures notes in chemistry*, vol. 2, pp. 1–108. Heidelberg, New York: Springer-Verlag.

Clementi, E. & Barsotti, R. 1978 Study of the structure of molecular complexes. XVI. Coordination numbers for Li^+, Na^+, K^+, F^- and Cl^- in water. *Chem. phys. Lett.* **59**, 21–25.

Clementi, E. & Corongiu, G. 1979*a* *Biopolymers* **18**, 2431.

Clementi, E. & Corongiu, G. 1979*b* *Int. J. Quantum Chem.* **16**, 897.

Clementi, E. & Corongiu, G. 1981 In *Biomolecular sterodynamics* (ed. R. H. Sarma), vol. 1, pp. 209. New York: Adenine Press.

Clementi, E. & Corongiu, G. 1983 *Int. J. Quantum Chem. Symp.* **10**, 31.

Clementi, E. 1985*a* *J. phys. Chem.* **89**, 4426.

Clementi, E., Corongiu, G. & Detrich, J. 1985*b* *Comput. Phys. Comm.* **37**, 287.

Clementi, E. & Corongiu, G. 1985*c* Advances in biophysics (ed. A. Wada). Limerick: Elsevier.

Clementi, E., Chin, S. & Logan, D. 1986*a* *Israel J. Chem.* **27** (2), 127.

Clementi, E., Logan, D., Sonnad, V., Christidis, Z. & Capotondi, A. 1986*b* *Comput. mech. Engng* **5** (3), 42.

Clementi, E., Lie, G. C., Hannon, L., Rapaport, D. C. & Wojcik, M. 1986*c* *Structure and dynamics of nucleic acids, proteins and membranes* (ed. E. Clementi & S. Chin). New York: Plenum Publishing.

Clementi, E. & Logan, D. 1987 *Special purpose computers* (ed. B. Adler). New York: Academic Press.

Corongiu, G. & Clementi, E. 1981 *Biopolymers* **20**, 551.

Corongiu, G., Detrich, J. & Clementi, E. 1981 IBM Tech. Rep.

Corongiu, G., Detrich, J. & Clementi, E. 1987 IBM Tech. Rep.

Csanady, G. T. 1982 *Circulation in the coastal ocean.* Reidel Holland, Kluwer Academic.

Detrich, J., Corongiu, G. & Clementi, E. 1984 *Int. J. Quantum Chem.* **18**, 701.

Dickerson, R. E., Drew, H. R. & Connor, B. 1981 *Biomolec. stereodynam.* **1**, 1.

Domingo, L. & Clementi, E. 1985 IBM Tech. Rep, KGN-17.

Dupuis, M., Spangler, D. & Wendoloski, J. J. 1980 NRCC Software Catalog, vol. 1, program no. QG01, Lawrence Berkeley Laboratory, University of California.

Eisenman, G. & Sandblom, J. P. 1984 *Biophys. J.* **45**, 88.

Etchebest, X., Ranganathan, S. & Pullman, A. 1984 *Fed. Eur. Biochem. Soc.* **173**, 301.

Falk, M., Hartman, K. A. & Lord, R. C. 1963 *J. Am. chem. Soc.* **85**, 391.

Fromm, J., Clementi, E. & Watt, R. O. 1975 *J. chem. Phys.* **62**, 1388.

Gomperts, R. & Clementi, E. 1987 IBM Tech. Rep. KGN-107.

Gomperts, R. & Clementi, E. 1987 IBM Tech. Rep. KGN-159.

Haidvogel, B. D. 1981 Woods Hole Oceanographic Institute Tech. Rep. WHOI-80-90 (October).

Hannon, L., Lie, G. C. & Clementi, E. 1986*a* *J. Sci. Comput.* **1**, 145.

Hannon, L., Lie, G. C. & Clementi, E. 1986*b* *Phys. Lett.* A **119**, 174.

Hannon, L., Lie, G. C., Clementi, E. & Yakhot, V. 1987 IBM Tech. Rep. KGN-128.

Hannon, L., Lie, G. C. & Clementi, E. 1988 *J. statist. Phys.* **51**, 965.

Herbin, R., Gerbi, S. & Sonnad, V. 1987 IBM Tech. Rep. KGN-155.

Hsu, H. & Sonnad, V. 1987 IBM Tech. Rep. KGN-134.

Kandadai, S. N. & Clementi, E. 1987 *Biopolymers* **26**, 1901.

Kim, K. S. & Clementi, E. 1985 *J. Am. chem. Soc.* **107**, 227.

Kim, K. S., Nguyen, H. L., Swaminathan, P. K. & Clementi, E. 1985 *J. phys. Chem.* **89**, 2870.

Kim, K. S., Dupuis, M., Lie, G. C. & Clementi, E. 1986 *Chem. phys. Lett.* **131**, 451.

Lie, G. C. & Clementi, E. 1986 *Phys. Rev.* **33** (4), 2679.

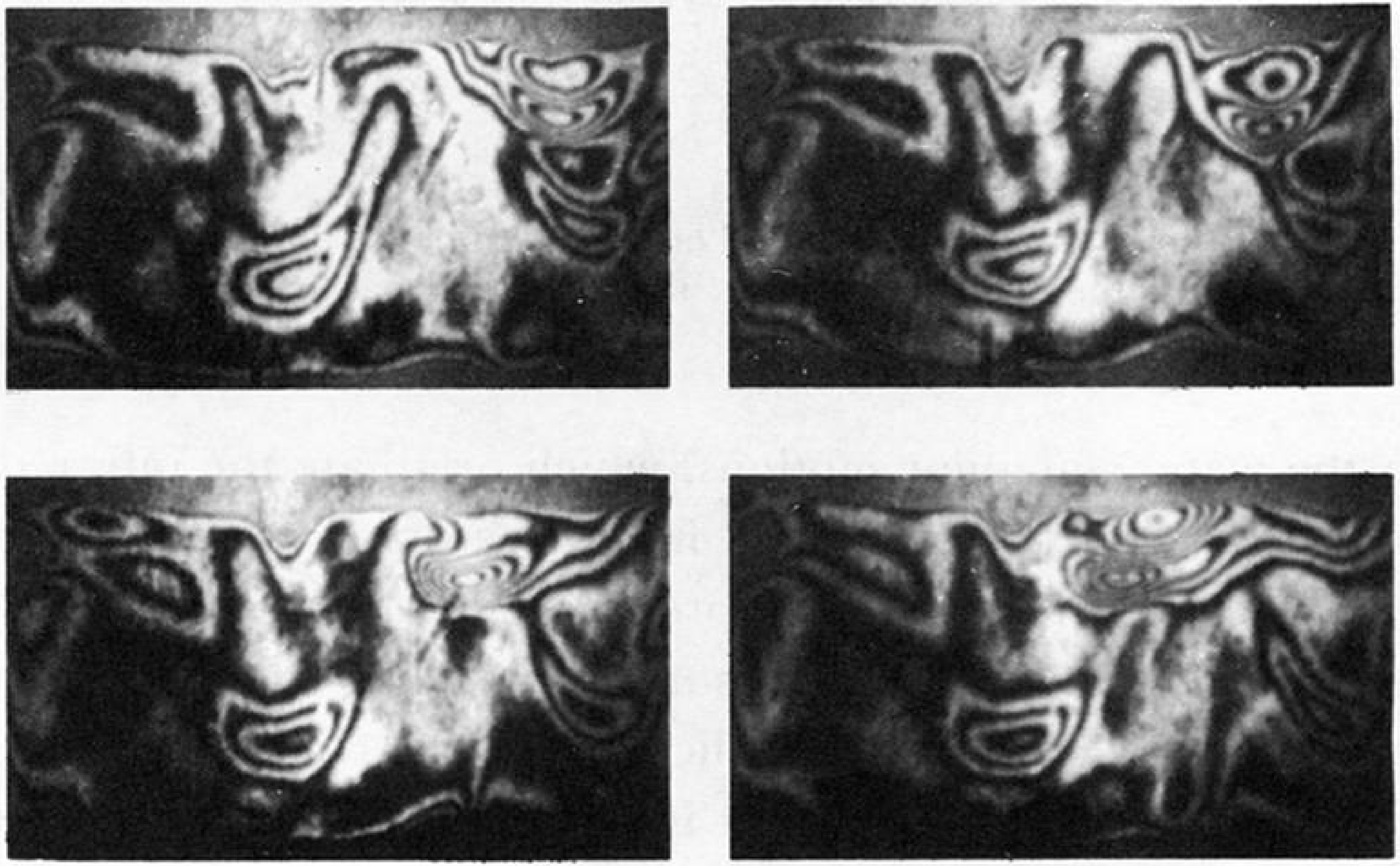
Lie, G. C., Clementi, E. & Yoshimine, M. 1976 *J. chem. Phys.* **64**, 2314.
Matsuoka, O., Clementi, E. & Yoshimine, M. 1976 *J. chem. Phys.* **64**, 1351.
Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H. & Teller, E. 1953 *J. chem. Phys.* **21**, 1087.
Milani, A. & Nobili, A. M. 1983 *Celest. Mech.* **31**, 241–291.
Milani, A., Carpino, M. & Logan, D. 1987 IBM Tech. Rep. KGN-161.
Nguyen, H. L., Khanmohammadbaigi, H. & Clementi, E. 1985 *J. Comput. Chem.* **6**, 634.
Opik, E. J. 1976 New York: Elsevier.
Orszag, S. A. 1972 *Stud. appl. Math.* **51**, 253.
Popkie, H., Kistenmacher, H. & Clementi, E. 1973 *J. chem. Phys.* **59**, 1325.
Rapaport, D. C. & Clementi, E. 1986 *Phys. rev. Lett.* **57**, 695.
Watts, J. D. & Dupuis, M. 1987 IBM Tech. Rep. KGN-106.
Wojcik, M. & Clementi, E. 1986*a* *J. chem. Phys.* **85**, 3544.
Wojcik, M. & Clementi, E. 1986*b* *J. chem. Phys.* **85**, 6085.

Discussion

S. F. Reddaway (3 *Woodford Close, Ashwell, Baldock, Hertfordshire*). What are Dr Clementi's views of doing fluid dynamics by lattice gas methods that are simple, highly parallel and use boolean processing rather than floating point?

E. Clementi. Cellular automata methods cannot deal, in general, with boundaries except in the trivial case when the function goes to zero.

S. F. Reddaway. Boundaries are now being incorporated into codes.



IGURE 7. Laboratory Bénard-flow, taken from a film, with the permission of P. Bergé and M. Dubois, C.E.N. Saclay, Gif-sur-Yvette, France.