Overcoming the Isolation

Berni Alder, Professor Emeritus at the University of California, Berkeley, who pioneered the use of particle methods in hydrodynamics, took the opportunity as a keynote speaker at PC'94 to reflect on why computational physics, in spite of its successes, remains somewhat isolated in academic circles.

Although there remains the problem of handling the interaction between particles in simulating, for example, Schrödinger's equation, Monte Carlo methods represent a great success for computational physics. However, other major challenges have not yet been tackled successfully. They include three-dimensional turbulence where there has been little progress for 60 years as distance and time-scales are very large, and pattern recognition where the hopes for massively parallel computation have not materialized. There will be little progress in pattern recognition until much more complexity is built into computing nodes, thus mimicking neural networks. The folding of DNA molecules and why it takes place so quickly are poorly understood; a new paradigm is needed as random folding clearly does not work. People are groping for ideas, although it seems that the presence of water is crucial in helping guide the DNA molecules through metastable intermediate states (see figure).

What is perhaps sad is the fact that computational physicists are often not accepted as equal partners in the academic world in spite of their considerable achievements which include, for example, calculating the mass of the proton from first principles (the challenge now is to calculate the mass of the hydrogen atom). They have tended to seek homes outside mainstream physics in areas such as mechanical engineering and applied materials research. The situation arises because an undergraduate curriculum in computational physics has not been defined, and because there is some confusion as to what a computational physicist represents.

Claudio Rebbi, in summarizing below how a curriculum can be developed, goes a long way to explaining what computational physics tries to achieve. However, the community itself is partly responsible as it has not focussed sufficiently on theoretical aspects and the development of algorithms, but tended instead to be absorbed by advances in computer technology. Hence, in spite of considerable discussion and development, few qualitatively new algorithms have emerged in the field of massively parallel computing.

However, one should not be too critical as there are efforts under way to identify new paradigms. Consider, for instance, the field of computational fluid dynamics. Lattice-gas methods have not been successful in simulations as the largest computers can only handle 10^6 particles for 10^{-8} seconds. Monte Carlo methods have been shown to be very effective, but only at low Reynolds numbers. Using a hybrid approach that combines the simulation of particle scattering at boundaries with continuum hydrodynamic methods away from the interface, it is becoming possible to tackle more relevant high-Reynolds number problems. The crucial issue is to match the Navier-Stoke's formalism with particle-like simulations by ensuring that conservation laws are satisfied. The first results for a magnetic read-write head floating just above a rapidly spinning magnetic recording disc show that one can realistically simulate the development of fluid-flow instabilities. If Rebbi's ideas are widely adopted it should be possible to develop other paradigms, and envisage further successes.

Predicting three-dimensional protein structures. Attempts to reliably predict the structures of proteins are in their infancy. One approach is based on the idea that native protein structures are in the state with the lowest free energy. The interaction between the protein and its water solvent is known to play a dominant role in the stability of the protein. This interaction can be introduced by assuming that it is equal to the sum over all atoms of the atomic solvation parameter times the accessible surface area. Braun and Mumenthaler reported at PC'94 that adding this simplified semi-empirical energy surface term to the standard conformational energy term and carrying out a minimization using Monte Carlo simulations gave folded structures with the correct three-dimensional topology. This is illustrated in the figure for a small test protein having three packed helices. One of the low-energy simulated structures (the structure on the left) closely resembles the structure on the right determined by NMR. So its seems that efforts to predict protein structures are moving in the right direction.

Teaching Computational Physics

Claudio Rebbi of Boston University, in arguing that the teaching of computational physics must become an integral part of physics education, offers some general principles as a guide.

COURSE PROJECTS IN COMPUTATIONAL PHYSICS

Calculate the thermodynamic properties of crystals.

Solve an equation for the internal energy of a crystal based on a quantized oscillator and the Debye approximation for the frequency of the normal modes of vibration. This introduces numerical integration. Going beyond the Debye approximation to use a more realistic dispersion formula demonstrates that the computational as opposed to the analytical perspective simplifies the calculation.

Calculate the trajectories of N-bodies subjected to mutual gravitational interaction.

A prototype project for the general class of molecular dynamic simulations. Introduces various algorithms for solving ordinary differential equations with initial value data. Leads to a discussion of stability and the need to employ more sophisticated techniques, such as the adaptive time step. With careful choices of initial data one can explore chaotic behaviour. Also introduces advanced computing technology (*e.g.*, implementation on massively parallel computers).

Calculate the energy of a superconducting vortex in the theory of Ginzburg, Landau and Abrikosov.

Minimize a function of the several variables that parameterize the field configuration. Introduces fundamental algorithms (*e.g.*, method of conjugate gradients).

Calculate the energy levels of a bound state of a heavy quark and antiquark with a potential that has a Coulombic component and a linear component.

Introduces boundary value ordinary differential equations. Goes beyond learning the algorithms to checking the predictions of the model against experimental data.

The majority of applications of computational physics involve numerical computations. The discipline must thus be taught by using a fair amount of numerical analysis. But there is much more to computational physics than the study of algorithms and numerical methods. Computational physics is the art of formulating and solving physics problems on the computer. It is this art which we must convey to students. To accomplish this we must keep the teaching of computational physics anchored to the treatment of actual physics problems. These can provide the background and motivation for the introduction of algorithms. The consideration of a concrete problem may, moreover, give the opportunity to illustrate the difference between an analytically minded approach and a computational approach, and how by changing one's perspective one can frequently achieve a simpler and more efficient solution. Finally, as code is developed for a teaching project, the students can refine programming skills and be exposed to advanced computing technology.

A Matrix

We therefore essentially have a framework for teaching computational physics consisting of a matrix of physics problems, numerical methods and computing technology. Such a framework provides an excellent educational platform, while also reflecting the integration of knowledge that plays a fundamental role