Computational Physics 1968-1978

F. James

To be published in the Physics Bulletin on the occasion of the 10th Anniversary of the European Physical Society.
If the safest way to define physics is "that which physicists do", perhaps the only reasonable way to define computational physics is "that which physicists do with computers". Even in that very broad sense, computational physics is not very old, and if we restrict ourselves to the formal existence of computational physics, we might put its birth around the same time as that of the EPS. What we are chronicling here is therefore not so much the recent developments, but rather the earliest developments in the field.

Let us first consider some of the milestones of the early days of computational physics:

1966: First issue of the Journal of Computational Physics
1969: First issue of Computer Physics Communications
1969: First Computational Physics Conference, Institute of Physics
1970: First CERN summer school in computing
1971: Trieste School "Computing as a Language of Physics"
1972: First EPS Computational Physics Conference

The advent of these schools, conferences, and journals has been important, not only in advancing the techniques of computational physics, but also in helping to establish the identity of computational physics, and offering a forum for communication between physicists from different fields whose primary working tool is the computer.
As computers have evolved, our ways of using them have undergone major changes. Most of us have been able to take advantage of the advance in technology and the decline in prices to use progressively more powerful computing systems, and it is now the rule rather than the exception to have a dedicated minicomputer driving any major physics experiment or experimental apparatus. Punched cards have largely given way to terminals as a means of access to the computer, and the once-popular binary card has completely disappeared at most centres where permanent disk space is available for the storage of programs. High-speed data links, long-distance networks, dial-up terminal lines, improved terminals and displays, high-capacity disks, high-density magnetic tapes, microfiche output, modern graphics devices, and other hardware features have influenced the way many of us work.

On a less mechanical level, our attitudes toward computing are also changing. The availability of better compilers means less assembler coding and fewer low-level tricks to improve speed at the risk of total confusion. We are slowly learning not to re-invent the wheel, but rather to use tested routines from an existing program library. Our programs, although more complex, are more readable and more reliable. Improvements in text editors and other facilities for maintaining programs and documentation have relieved us of much of the drudgery of programming and allowed us to concentrate more on improved algorithms and better program organisation. Whereas at one time the physicist had to write everything except SQRT and COS, he can now count on his computer centre to supply him with a wide range
or tested software including special functions, linear and
differential equations solvers, function minimizers and
integrators, etc. Computer centres that do not maintain their
own program libraries can obtain good general-purpose scientific
libraries either commercially or from other laboratories, and
larger applications programs are now often published in journals
and made available through many different distribution channels.

FORTRAN has yet to be dethroned as the universal programming
language of physicists, in spite of the fact that we now seem to
be nearly the only ones who use it. ALGOL 60 has not been very
successful; ALGOL 68 and PASCAL are trying, but will probably be
defeated at the last moment by the new standard which will
finally make FORTRAN nearly acceptable even to language purists.
Exceptions to the dominance of FORTRAN are largely limited to
special-purpose languages, either non-numeric, interactive, or
machine-specific. A relatively recent phenomenon is the
proliferation of FORTRAN preprocessors which allow one to extend
the language, maintaining its better features (especially its
universal availability) while eliminating some of its
restrictions. The advantages of these preprocessors are
essentially limited only by the imagination of their authors; the
universal disadvantage is that there is one more level of
translation between source program and executing program, which
introduces another level of diagnostics, increases compilation
time, and can make debugging more complicated.

With a few notable exceptions, the communications gulf between
computational physicists and computer scientists is still very
wide. Where it would be unthinkable for a mathematical physicist
to have no formal training in mathematics, it is still common for
a computational physicist to have no formal training in computer science. Presumably this is only a temporary situation, as anyone coming out of a university today should have had some training in computer science. From the side of the computer scientists, I see indications that some of them also recognize the problem and would like to orient their work more toward practical aspects of the type faced by research physicists. The same thing is also happening to some extent in other related fields such as numerical analysis, statistics, and applied mathematics.

Turning now to the various individual areas of computational physics, the general picture is one of steady progress everywhere, with the computer becoming an indispensable tool in nearly every field. (Paradoxically, one of the fields which relies least on computing is solid state physics, which made modern computing possible!)

Of course the traditional areas where computers have been used heavily from the start are nuclear and particle physics. A high-energy physics experiment today relies heavily on computers in every phase, from the design and construction of accelerators and beams, through the data-taking to all phases of the data analysis, including the control and calibration of detectors and other major pieces of experimental equipment. The principal developments in the last ten years are probably: (1) The invasion of the experimental halls by increasing numbers of increasingly powerful minicomputers capable of performing a large part of the data analysis on-line, and (2) the gradual awareness that some of the ambitious projects for completely automatic pattern recognition are simply not feasible, and much can be gained by
using human guidance at strategic points in the data-gathering process, or better yet by planning the experiment in such a way that completely automatic data reduction is foreseen and feasible.

In fluid dynamics, plasma physics, quantum chemistry, atomic physics, and astronomy, the problem is not so much the complexity of experimental data, but rather predicting the behaviour of large systems whose elements obey known laws. The nature of these equations is often such that even three particles makes a "large" system, and two or three dimensions are already enough to make the numerical solution extremely complicated. It is often the case that even the grossest features of such systems cannot be understood without the use of the most powerful computers, programs, and graphical output devices. In these areas the most exciting recent developments concern two projects of the utmost practical importance for all of us: (1) the plasma simulations designed to harness nuclear fusion energy, and (2) the recent European project to devote one of the world's most powerful computing systems to making detailed medium-range weather forecasts.

Theoretical physics is of course the last refuge of the non-computerized physicist, and perhaps for good reason. After all, their work concerns neither the handling of large amounts of data nor the numerical solution of complicated systems. But even the most theoretical domains sometimes present such complexities as can only be handled by computers, for example unwieldy algebraic expressions which are now routinely evaluated and manipulated with computers. And more recently the famous four-color theorem of topology was finally proved with the apparently indispensable help of the computer.