The interplay of computer science and large scale scientific calculations

by KENT K. CURTIS
National Science Foundation
Washington, D.C.

The history of computing has been characterized by the statement that what the manufacturer makes, the customer takes. To the extent that this is true, it is not only an interesting and perceptive comment on history but also a testimonial to the remarkable nature of the sequential, stored program, electronic computer. If that invention had not fit so well from the beginning to a wide variety of interesting problems, the development of hardware and software systems could not have proceeded as independently of scientific motivation as it did. Of course, scientific requirements have influenced systems design, but the prominent parameters such as speed or memory size could be understood without detailed knowledge of scientific problems or programs and the parameters which might depend upon the structure of programs could be ignored.

It is fortunate that that was true. There were important problems to solve in physics, chemistry, meteorology, etc. We could work on those without delay and let systems design take care of itself, with ad hoc software adjustments when things became intolerable. Computers worked so well there was no apparent need for guidance by scientists of system design. Better design might make an incremental difference but that increment was not sorely missed and barely worth an incremental effort.

Meanwhile, computer development followed an internal dynamic of its own. It was influenced by the fact that scientific requirements provided a ready market for computers which were big and fast, but the general purpose computer was a useful paradigm and rapid advancement of electronics technology kept many problems at bay. Computer engineers and businessmen could do their own thing in their own way, with unusual freedom. The computers they built have reflected principally their own genius in taking advantage of technological opportunity, and that, happily, has been sufficient.

In making these comments, let me note that they refer to the machines which have until now dominated scientific calculation. Other designs which attempt to reflect some additional measure of understanding of scientific calculations have been considered and some are of substantial current interest. Pipe-line systems and the ILLIAC IV, are among those which come to mind, but none of them has yet had opportunity to prove its value. Even they, however, are the result of adding only one more global observation to the design criteria, the observation that many scientific calculations have the common property that they involve the redundant use of identical operation sequences on ordered sets of data. Hence, their appearance and history does not vitiate the force of the earlier observation. On the contrary, they extend its force to the near future.

They do, however, raise several interesting points. First, the principal ideas of architecture underlying each of the machines which are now being physically realized are approximately ten years old. Still, these machines are only now being delivered and experience suggests that it will be some time yet until the systems will have been stabilized sufficiently and used sufficiently to be able to make a definitive evaluation of them from experience. This is a long gestation period. It suggests that we should not expect breakthroughs in large scale scientific computing due to new ideas in machine architecture in a short time scale. The machines which are now coming to life may be substantial contributors to scientific computing ten years from now, if they prove out, but it is unlikely that machines which are still on the drawing board will contribute until later.

It also suggests that unless we believe we already have discovered the only important concepts in machine organization (which is possible but would be most remarkable, if true) we should be very interested in reducing the time span from idea to realization. Ideas in machine organization spring from an environment
of experience to meet a perceived need. They need testing to evaluate them. If the need is really important it will probably persist until an answer is found. But, on the other hand, if the need is important we would rather not wait ten years. This line of thought argues for research in such areas as description languages which can describe complex systems and be translated to a design and fabrication process and it argues for support by the scientific community for such research. I would like to return to this point later.

Second, a “paper machine” has very little weight. At an early stage in the development of the machine which is becoming ILLIAC IV, I spent some time with several people who were active in hydrodynamics and meteorological research to see what basis for estimating performance could be established from analyses of the actual codes which were of interest in those fields. My question was, to what extent does the parallelism which is available in this design offer an effective improvement in performance, considering the detailed structure of the computing algorithms, data management, and I/O processes that are used in these problems? I thought I understood how to do that kind of analysis, at least in part, including modifying the codes for parallel processing, but I did not know the codes. I hoped that someone who knew the codes could be motivated to undertake some analysis. I was impressed with the degree of interest in the machine and the lack of interest in attempting any analysis. I found a uniform response that if the machine existed, they would try it because it seemed intuitively attractive. No one was willing to attempt an analysis of his programs, however, with a view to justifying construction of the machine or influencing its design. In all fairness, they did not know how to do it and neither did I. The problem is harder and more subtle than I thought. The question remains open and, indeed, I still do not know how to find an answer with assurance except by extended experience. Responsible people still hold an honest difference of opinion. Fortunately, the continued development of that machine did not require the active support and involvement of the scientific community it was intended to serve. Technological opportunity supported by intuition and some analysis has been sufficient. The experience suggests, however, that there exists only a weak coupling between the community of natural scientists and the community of computer engineers. It also affirms the need to test new ideas in machine design by actual construction of machines and evaluation in a user environment.

Another interesting point which these machines have emphasized is that more than one schema may be applied to the solution of most problems and different computer architectures may favor different schema. This seems obvious but it required the actual development of parallel and pipe-line machines to stimulate such rethinking of problem formulation and algorithms. Now this offers a rich field of research for numerical analysts and computer scientists, where results may be of direct interest to scientists. I personally believe that this is a profoundly significant development which may portend the appearance of a new degree of influence by computers and computer science in our conceptualization of the scientific problems we attempt to solve. More immediately, however, it means that definitive evaluation of new ideas in computer architecture can only be made after good algorithms for those architectures are devised. Is this backwards? Should not the analysis of algorithms influence design? Is this not another example of weak coupling between communities with allied interests? Fortunately, we are beginning to make progress on the theory of algorithms which offers hope of benefit to both scientific computing and computer design.

Let us return, now, to the remarkable success of sequential computers in large scale scientific computing. If one considers the spectrum of activities which now comprises large scale scientific calculation one can observe a number of common features.

1. The conceptual theory upon which the computations are based was firmly established before the invention of computers. That theory had been well verified using experiments and calculations which could be and were conducted without computers, providing a solid basis for faith that large investments of effort and money in computing would not be wasted.

High energy physics might be considered to be one outstanding exception to this since conceptual theory in that field is being developed during the age of computers. If one considers the actual computing being done, however, the distinction disappears. The large scale computing is in analysis of data from experiments which are patterned in their design and analysis after particle physics concepts that were well established before computers were applied. Orders of magnitude less computation is used in theoretical investigation exploring the conceptual understanding of high energy physics. No theory commands sufficient faith to justify a larger investment of time and money.

2. The conceptual theories are all expressible in well formed mathematical or statistical models.

3. Based on the conceptual paradigms, it is possible to visualize, without using computers, interesting problems which can be solved with computers.
(4) Such problems can be solved, to a sufficiently
good approximation to be interesting, using
sequential machines operating for reasonable
lengths of time.

(5) These problems enjoy a high national priority
giving them access to the large amounts of
money and effort which have been required for
their study.

If you think about it, this is an interesting set of
properties. It suggests the roles which have been played
in fact by scientific motivation, technological oppor­
tunity, and public support in determining the
course of development of large scale scientific com­
cputing. It provides a basis for understanding why
scientific progress and the development of computer
technology could be so vigorous, so symbiotic, and yet
so independent. It provides a rationale for the observa­
tion that scientific inquiry has adopted the technology
provided rather than leading or strongly cooperating
in its development. At the same time, it points to
certain limits, which have been observed by the course
of events, in fact, on the regions of success in scientific
computing. If one may have faith in continuity, it
suggests some features which can be expected of de­
velopments in the near future.

One interesting observation which derives from this
set of characteristics is that although computing has
profoundly changed the style and methodology of
research in some fields, although it has opened new
questions for serious consideration, there has been no
departure from already established concepts. Com­
cputing has not yet altered our paradigms of nature in
a fundamental way. This is not surprising—the time
constant for developing fundamental physical theory
is much longer than the history of computers—but it
indicates that we should not expect the short time
scales associated with the development of technology
to be reflected in any obvious way, if at all, in the rate
we revise our fundamental understanding of nature. It
also invites the question of whether computing should
be expected to ever become an essential component of
conceptualization. Are there interesting phenomena
which cannot be described within the framework of
mathematical or statistical models? Are there impor­
tant questions to be posed which do not yield to
analysis by sequential procedures? What would it
mean to include a program operating in time as an
essential element in our description of a model, as we
now include time-dependent equations? It is an in­
teresting conjecture. Considerations of real-time sys­
tems are suggestive.

Before going on, let us examine some possible in­
ferences which may be drawn from the preceding
characterization of large scale scientific computing.
Certainly it is true that national priorities have strongly
influenced the choice of fields of investigation and
hence the visible progress. These priorities are changing
and efforts in large scale computation in fields other
than the physical sciences will receive more encour­
gement. Fields which have the property that they can
take advantage of existing technology, have well
posed questions based on well formulated mathe­
matical paradigms, and have a sufficiently large com­

From the collection of the Computer History Museum (www.computerhistory.org)
the physical sciences and by the fact they do not yet enjoy high national priority status. Technology notwithstanding, ten years is a short time in the development of human thought. Even quantum theory took much longer to evolve.

Another feature which emerges from considering this set of properties is the prevalence in successful scientific computing of certain mathematical or statistical models. Not only were the conceptual theories well established before computers entered the scene but the models used to describe those concepts and the approaches taken to solve them were also well established. For example, finite difference approximations of partial differential equations. We have always quickly exhausted the capacity of the machines we had to solve those problems in that way but we have not become frustrated because technology advanced rapidly enough to keep the game interesting. We have made substantial advances in discovering faster, more efficient algorithms for carrying out these processes and in understanding convergence and stability but basically we have worked, not only in the same conceptual but also in the same procedural framework that was common with desk calculators. (Monte Carlo techniques might be thought to be an exception. I think not. As with many things, computers made those already established techniques more effective.)

Now a new line of thought is beginning to emerge. I am aware of it in meteorology but it may be present elsewhere. Computers have enabled us to undertake "ab initio" calculations which could never be approached before. (Indeed, some chemists now believe that their theory is so sound and their computing techniques so accurate that they can place as much faith in the results of a calculation of a possible molecule as they can in an experiment. They have some substantial basis, as is evidenced by the story of the noble gas compounds but it reflects an interesting shift in attitude.) This ability has enabled us to undertake prediction with an accuracy never before possible and the long range weather forecasting which has become part of our daily lives is an outstanding result. But, perhaps we are beginning to push the limit of that ability using standard techniques. The predictive power of a given mathematical model rests ultimately upon the completeness and accuracy of the physical data we can provide even if the model is in principle perfect and we have infinite computing power. At some point, decreasing the size of the mesh or increasing the number of layers in a three dimensional atmospheric model multiplies the computational steps required without yielding a commensurate increase in useful results. It has been suggested that in atmospheric circulation problems, computers may be bringing that limit within sight. A couple of weeks may be a limit beyond which the model diverges from reality because of limitations in physics, not computing.

This observation is suggestive. The problem has been recognized before and some computer scientists have studied significance arithmetic schemes to provide continuous knowledge of the significance of quantities throughout a calculation. This work has not yet entered into practical application but interest may revive in it as we find ourselves pushing the limits of data accuracy more frequently. If true, it will offer a rich field for research relevant to computer users.

But the observation is also interesting in another dimension. It would be surprising if a conceptual theory can find mathematical expression in only one model or if that model can be solved in only one way. It would also be surprising if we have already chosen the best model for all situations. In atmospheric circulation, some of the questions one wants to raise may be answerable from a turbulence model which is statistical in nature and leads to a different computational problem. This could be a better way to obtain some results than solving the general field equations.

This phenomena is not new. Scientists have always reformulated models and changed approximations when they were unable to make further progress with one particular line of attack. Instead, the last twenty years have been unusual. Computers have increased the power of certain traditional lines of attack so much that we are only now approaching their limits. As we reach those limits, more attention may be paid once again to alternatives and to the real significance of the numbers we compute.

Finally, this set of characteristics of successful scientific computing can be helpful in anticipating the near term impact of new developments. They suggest that the most likely effect of technology will be to help scientists do better the things they already know how to do, perhaps, even stronger, the things they are already doing. Before discussing specifics here, however, it is necessary to make note of a new current which is beginning to influence the computing environment. While scientists and computer manufacturers each followed their own independent but mutually reinforcing interests, a third community, the community of computer scientists, was gradually emerging. This is the community whose imagination is captured by the intrinsic interest of complexity, systems, and computers and it is largely populated by bright young people who grew up with computers as a background. To the extent that computers have broken tradition, these people have the advantage of growing up with fewer ties to the old traditions. This community is now taking shape and beginning to find things to say which
have relevance to both scientific calculation and computer design.

Neither the computer manufacturers nor the computer users have had much interest in computer science. They have been preoccupied with technology and have strongly flavored the definition of the term "computer science" with their own interests. For this reason it may be necessary to define more precisely what I mean. To some degree, one man's science may be another man's technology and neither can live in good health without the other. This is especially true if one looks at the actual occupations of people who call themselves scientists, the things they really do with their time and energy. The term science, however, has a commonly accepted meaning which involves the discovery and formulation of generalizations or principles that can be expressed in models describing phenomena. Computing has focussed attention on phenomena which are as old as man but which have been taken for granted; phenomena concerning information, complexity, and the processes for organizing, describing, and analyzing. Computer science is addressing itself to these phenomena. Its success will have meaning for all who are concerned with computers.

The conduct of computer science research is also interesting. There is reason to believe that it must be, at least in part, essentially complex and involve the concerted effort of research teams. Professor Jack Schwartz, of New York University, has used an analogy which I find fruitful. He notes that mathematics is a solitary occupation which can be compared to diamond mining. One digs through large quantities of dross but in the end one finds a jewel which is small, beautiful, and can be easily admired by all who have eyes to see. Computer science, on the other hand, is comparable to coal mining. One handles comparable or greater quantities of material but none of it can be thrown away. Everything must be preserved for use and the process is intrinsically complex. This does not preclude generalization and the discovery of principles but it does say something about the process of finding them and about the description of them which gives them meaning.

With this preamble, let us return to the consideration of new technology. There are several things which come immediately to mind. They cannot really be separated—each influences the others—but you can construct your own feedback loops.

(1) Machine architecture

This has already been discussed. The new architectural ideas which may affect large scale scientific computation within ten years are already being built in hardware. The long lead times involved make it almost certain that the impact of other ideas will not be realized in that time frame. New contributions inspired by the machines now being built will increase rapidly as those machines actually become available, however, and will influence their use. These will be in the areas of problem formulation, analysis of algorithms, data management schemata, operating systems, languages, and file management. I hope these machines will be available for computer science research. I think they cannot be properly evaluated without it.

(2) Solid state memory

It appears likely that we will soon have memories available which are comparable in cost to present core memories but comparable in speed to logic. This will change one of the parameters of machine design which has been nearly constant throughout the history of computers, the ratio of memory time to logic time, from about 100:1 to about 1:1. It would be most surprising if this does not have some effect. Programming would seem to gain a new degree of freedom. One possible effect may be more effective specialization of machine function without specialization of machine design. Floating point arithmetic might be programmed again, not wired, for example, to give greater flexibility of word size and significance arithmetic may have a better chance for effective implementation.

On the other hand, the forces influencing change are numerous, subtle, and complex. It is very difficult to forecast but fascinating to watch as events unfold. The foregoing statements assume that it is easier to program than to build machines. But one can also imagine ways in which this same memory development, with proper inspiration and motivation, might lead to simplifying the fabrication process. Modular building blocks for machines and systems of machines might then become part of our standard repertoire.

(3) Algorithms

This is an area where computer science will give us an undisputed benefit. Theoretical limits on the speed of algorithms will provide measures against which to compare performance and will inspire improvement. Analysis and classification of algorithms will begin to give formal structure to programs and guides to optimize design of both programs and machines. From this area will come the indispensable tools for understanding the information processes which are possible for us to use in solving problems.
The benefit will be felt, not only in increased performance on our present problems, which may be significant or not, but in providing us with constructive procedures for thinking about the information processes appropriate for our problems. It may seem facetious to suggest that the science of algorithms will usurp the role of calculus and numerical analysis as the primary tools for description and problem solving. It certainly is premature. But I will be interested to see what our attitude is about this at the end of this decade.

(4) Languages

In the short history of large scale scientific computing we have relied primarily upon one language, FORTRAN, and have accepted its limitations in the interest of pursuing the scientific problems at hand. Again, it has been the case that other languages might provide an incremental gain but that has not been worth an incremental effort by scientists to learn, to use, or to develop. Improvements have been made and will continue. Computer scientists are developing automatic code optimization procedures which may help performance and, perhaps, debugging. Monitoring schemes and performance measurement studies are making it possible to do time analysis of programs to increase the efficiency of codes. Factors of two, three, ten, or sometimes fifty improvement result. But with respect to learning languages, the scientific community has been conservative.

Now we may be reaching another kind of limit. Programming is a human effort involving a long chain of people and events before a complete program can be made to work satisfactorily. Programming time and the irreducible error rate at each step of the process places a practical limit on the growth of complexity of programs, generally by making the cost or time required to achieve a certain degree of complexity greater than we can, or want to invest. In spite of large programming investments and outstanding successes we are all aware of these limitations, and have had the experience of thinking of interesting programs which we were unwilling or unable to write. More significantly, some of the programs we have already constructed are maximally complex for our resources. FORTRAN is becoming a restraint that can be felt.

It is clear that this restraint can be relaxed since this is an area in which computer science has made progress since FORTRAN was developed. Iverson's APL, for example, is indicative as is Schwartz' SETL. In still another direction, Hearn's REDUCE is building a record of accomplishment. Within a decade, scientists will be multilingual and have greater freedom for scientific research as a consequence.

Another problem in the general area of languages which computer science is attacking is the problem of transportability of programs from one computing environment to another. Gradually, the structure of programs, compilers, and operating systems is becoming clear. We are learning how to isolate the essentially machine dependent portions of codes, and to bootstrap a system with minimum reprogramming effort. The benefits of this development are already being felt in economy of programming effort when changing machines. Ultimately, it may make an important contribution to scientific communication by making it feasible to exchange codes.

(5) Communications

This may make the most significant change in our daily working environment and seems to have large momentum, with stimulation from many sources. The National Bureau for Economics Research Center, which I mentioned earlier, is being established with remote operation in mind from its inception. So is the CDC-7600 installation at Berkeley for high energy physics. ILLIAC IV, at Ames, will soon follow and it seems likely that other centers for specialized disciplines with planned remote operation will soon appear.

The trend seems natural and inevitable. Research computing involves not only a computing machine but also a corpus of appropriate programs and an operating policy favorable to the work. Finally, and perhaps most importantly, it involves a group of scientists and support staff which can work well together and communicate freely to develop the research methodology. The success of this organization for research has been demonstrated in the national laboratories. It is not surprising that it carries over to computing with communications providing the means.

At the same time, this mode of operation is a departure from the organization of research computing which developed during the fifties and sixties when each institution built the local, general purpose computing capability which would best satisfy its overall staff requirements within the limits of its resources. This change requires an adjustment by all parties concerned and the transition will take time to accomplish. It will be interesting to see what changes in our working environment and relationships result.

These are several technological changes that may affect scientific computing in the near future. Others should be included, also. The availability of large data files, improvement in man-machine interaction, progress in data representation (including the question of what data should be stored and what should be reconstructed), and others will have a significant impact.
Yet each of these seems likely to have effects which are rather specific to particular fields of research, in the near term, and little influence common to all fields. Color graphics may be an exception.

One other project, which NSF is undertaking, should also be mentioned. That is a cooperative effort among several universities and national laboratories to analyze, certify, and document standard subroutines. The resulting library should give us greater basis for confidence in results and less trauma in carrying out research programs involving computing.

Before closing, I would like to mention the problem of publication. Scientific computing has labored under a handicap because it has not been possible to publish programs. A number of factors are relevant but I am not sure of their causal ordering.

(1) **Attitude**

Programming has been considered necessary but menial even though it has been the principal occupation of many scientists in terms of time and energy.

(2) **Language**

Mathematics has a language which is concise and makes the intellectual content clear. Often, the best description of a FORTRAN program is the program itself which is neither concise nor clear.

(3) **Utility**

A mathematical analysis of a theory or experiment can be universally understood, generalized, and used. A program is typically obscure, specific, and almost universally useless.

The belittling attitude toward programming, which is another carry-over of tradition, has certainly inhibited any attempt to solve the problem but failure to overcome the other hurdles does not encourage a change in attitude. It is a classical example of an impasse, one which natural scientists will not solve within the framework of their own disciplines.

It may be that computer science will make an important contribution here by developing the basis for describing and communicating programs in a really useful and general way. If that results, the benefit to science of improved communication may be great. It is interesting, though, that the path to this accomplishment will have been through byways such as automata theory, formal languages, perhaps set theory, the study of algorithms, and the structure of machines which have almost no discernible relevance to physical science.

We have seen that although progress in large scale scientific computing has been surprisingly rapid, it has also been bound by tradition. We have also seen that there is a long time constant in developing and absorbing technology. Both of these facts tend to become obscured by the activity around us but together they will continue to shape the near term future. At the same time, a new influence is emerging through the development of computer science which may provide the means to relax the constraints of tradition and expand both our conceptual and procedural horizons. The time scale for fundamental change, if it occurs, can be expected to be long but a decade may reveal the strength of the trend. It will be interesting to watch.