STATE OF THE ART IN SCIENTIFIC COMPUTING

R. W. Hamming
Bell Telephone Laboratories, Incorporated
Murray Hill, New Jersey

In order to understand the current state of an art it is often necessary to examine its history to see how it evolved. In examining the growth and present state of scientific computing, as contrasted to the other three fields of this session, machine design, software, and business applications, I find that there have been, and are, significant differences—in particular scientific computation has had a much more orderly growth than the others and is probably much more stable now, stable in the sense that the immediate future can be seen reasonably accurately. For example the comparatively ancient text by Whittaker and Robinson can still be used, but a book on coding of five years ago is hopelessly out of date.

Two reasons can be given for the more orderly growth of scientific computing. First, for practical purposes scientific computing has a much longer history of development. It is true that some of our earliest records are clay tablets recording commercial aspects of a civilization, and it is true that in the 1930's and 40's accounting machines of various types were adapted to scientific work, yet it seems fair to say that scientific computing was far more highly developed over most of the last 2000 years than was commercial computing. Furthermore most of the machines of World War II and the early post-war period were designed by and for men in science.

Second, behind much of scientific computing stands a highly developed and elaborated body of knowledge known as mathematics. This is very true for the field of numerical methods and to a somewhat lesser extent for many areas such as artificial intelligence which I suppose fall in the area of scientific computing. Scientific computing involves the application of engineering judgment to adapt the precise theorems of mathematics to the practical ends of computing, but the existence of mathematical theories is a great asset that none of the other three fields of this session can draw on to any similar extent. The fields of statistics and engineering also contribute to the orderly growth of scientific computing.

In saying that the growth of scientific computing has been, and probably will be, reasonably orderly, I do not want to give the impression of smugness and complacency—there is much that still needs to be done—but I do want to make clear the point that we have a very definite advantage in having a framework of known theorems, accepted notation, and form of publication, against which to judge our field.

I should also observe that both the hardware and the software fields have been generally developed for use in scientific computing before the use in business applications.

Let me now turn to an examination of some special areas of computing, taking them to some extent in the order of the amount of current machine time used. It may come as a surprise to some of the younger members of the computing fraternity to be told that many of the early relay machines as well as the ENIAC and probably the ORDVAC had their motivation in the demands of exterior ballistics—the solution of the ordinary differential equations of
the flight of a missile—in order to produce range tables. Even today many of the simulation problems we find on our computing machines involve the solution of ordinary differential equations of missiles.

A few people may have been aware of the stability problems that can arise in predictor-corrector methods for the solution of ordinary differential equations, but most of us at the end of the Second World War were blissfully ignorant of this fact. Not everything about stability is now known, but we seem to have the problem well under control—except in the so-called “stiff equations” where for the equation

\[ \frac{dy}{dx} = f(x, y) \]

we have

\[ \frac{df}{dy} \]

large and negative. Special cases of stiff equations have been handled, but a good general approach still seems to be lacking.

In the solution of ordinary differential equations it is usual in mathematical circles to speak of the step size—in engineering circles the sampling rate is a better way to describe the situation. In defense of my opinion that matters are well under control I will say that I would be greatly surprised if any new general methods will be found that would allow a significantly lower sampling rate unless they go to using many past data points; the sampling theorem of information theory is against it, more or less. However, for special systems of equations very little is known. For example, there seem to be no generally accepted methods for solving orbit calculations of space missiles in spite of vast sums of money spent for machine time to compare various methods. Nor has the theory so far been able to help much in the matter.

Data reduction is another activity which consumes much time, and which has had no great surprises—except to those who underestimated the amount of data that can be telemetered from a space missile in the course of its active life!

Turning to partial differential equations, we did know about stability in this case and were able to solve on primitive equipment rather difficult cases during the Second World War. The main advance seems to me to have been the discovery of implicit methods of solution which permit us to escape from the net size restrictions. Much work goes on in this area, and I would suspect much remains to be done.

Simultaneous linear algebraic equations have seen an enormous number of published papers, and I suspect the volume is due more to the mathematical elegance of the problem than to the intrinsic importance. In spite of all the research and proposed new methods I keep hearing the words Gauss, Seidel, Jacobi, relaxation—indicating to me that the older methods, somewhat modified are still widely used. Thus I am forced to say that comparatively little progress has been made. I suspect that Householder’s modification of Given’s modification of Jacobi’s method, and Wilkinson’s analysis of a slight modification of the Gauss elimination method are the most popular methods at the moment.

Again special cases have found special methods; for example systems with many zeros (so-called “sparse systems”) have great importance in many fields of applications. Another example is the special systems which arise in the implicit methods of solving partial differential equations.

Eigenvalues and eigenvectors of matrices have had a similar history; we have had many small advances, we by no means know as much as we wish we knew, but the fields have had no really great improvements, and matters seem to be in a state of orderly evolution.

Zeros of polynomials and more generally zeros of functions are further topics with long histories and no completely satisfactory methods of solution. When I recall that the great mathematician Gauss gave seven different proofs of the fundamental theorem of algebra and all of them are difficult to apply to practical computation, I do not expect to see a really easy method appear tomorrow—but I could be wrong.

Having described some of the more static parts of numerical analysis, let me turn to some of the more changing ones.

I would say that the use of Chebyshev (equal ripple) approximation has produced, and will
produce, great changes in our methods. It now tends to dominate completely the special function approximation area. A few cultivated mathematicians knew about Chebyshev polynomials many years ago, but their importance and central role seem not to have been appreciated before Lanczos rehabilitated them.

Along with the development of Chebyshev approximation I would also place the growth of nonpolynomial approximation in computing. Some methods, such as Prony’s method of approximation by sums of exponentials, were known and used, but they remained curiosities rather than regular tools of the trade. In particular I would call your attention to the growth of use of power spectral methods and band limited functions as a significant step forward. Both are as yet in their infancy and much work remains to be done to make their use better understood and more widely appreciated. The band limited function approach goes back at least to Gray and Rubinoﬀ at the University of Pennsylvania, while the name of Tukey is associated with the development of the power spectral approach. Both have their beginnings in electrical engineering practice. Indeed it should be evident that electrical engineering and information theory are two branches of knowledge from which computing can draw new inspiration and new methods.

Monte Carlo methods, meanings using random numbers in a computation, have received a great deal of publicity. However, I would hazard the opinion that the use of random processes, when the original problem does not have such a process, has not so far had the value most of us in the early days expected it would. It is in simulations, which currently occupy a lot of machine time, that random numbers are so often used to simulate various random processes, including noise. A rather large body of knowledge has been developed in this area, and a good textbook would be of tremendous help to those who are not specialists in the area. Indeed, I suspect such a book would stimulate the field itself. Several eminent men have started such books but were too busy to complete them. Thus the field is cursed with such an extensive “oral tradition”—von Neumann knew—it is in Herman Kahn’s notes—that the outsider cannot easily get started in the field.

Game theory is closely associated in my mind with Monte Carlo methods. The importance of the ideas of game theory is very great, but so far they have had comparatively little influence on current computing practice. Usually only comparatively trivial situations can be explicitly solved, and this may be the reason for the failure to influence computing practice in other areas such as integration. If so this would seem to be a fruitful field for investigation.

A couple of completely new fields are linear and dynamic programming. Both arose out of attempts at optimization, and are probably not the last fields that will arise from this fruitful source.

I have been surprised at the variety of problems which have been reduced to linear programming problems, especially those requiring integral solutions, but many times it seems to turn out that the general linear programming method is so expensive to use that the reduction is mainly of academic interest. As a result many special cases of the linear programming problem have been investigated and special methods for their solution found. I suspect that much remains to be done and that it will be economically valuable as well as intellectually satisfying.

I am gradually working my way towards such fields as scheduling, critical paths, and mechanization of certain design methods. All of these can, I believe, be summarized under the broad topic of find an “algorithm” for a problem or process that previously had been solved by guess, intuition, experience and dumb luck. This area has a bright future if you judge it by the money involved. Usually there is only a fragmentary mathematical background to draw upon, and this often in the difficult fields of integer solutions, and combinatorial mathematics. The history of mathematics seems to suggest that combinatorial mathematics and number theory are both difficult fields in which to find general methods, and hence we cannot expect to find a supporting structure such as numerical analysis has. The work, therefore, is likely to be fragmentary, irregular in development, and highly frustrating at times. Yet, let me repeat, it probably has a bright future in the sense that spectacular, unexpected results can be found in many special, important cases.
I would like to summarize a couple of points I have been developing. Optimization, and discrete, integer processes are two threads of scientific computing which have recently given rise to much new material, and should produce even more in the future. Both lack an adequate, practical mathematical theory for background, but both are of great economic importance, and support for work in these areas should be easy to find in the form of special situations of urgent importance to management.

I have avoided to a great extent discussing topics which have statistics as opposed to mathematics as the formal background, as well as the topic of statistics itself. Clearly statistics is required in the analyses of roundoff effects in long computations, in the design of experiment plan for examining situations involving many parameters, in sampling plans, in significance tests connected with "least squares" fitting, etc. Here again we have many topics awaiting further development, but for which much of the formal background information is reasonably well developed. In particular, in Roundoff Theory our knowledge is still in a sad state in spite of its obvious importance and extensive bibliography.

Statistics itself seems to be changing somewhat due to the impact of computers and the problems that are arising from the available data that can now be processed economically.

Topics in the general area known as "artificial intelligence" seem to fall in the domain of scientific computing, but unlike most of the topics so far discussed there is seldom a body of precise knowledge behind them. Most of them are in spirit like the other three areas in this session, subject to uneven growth, surprises, our inability to judge easily the significant result from the trivial and, even at times, false results.

I think this is the proper time for me to develop the themes of the advantages and disadvantages of having a body of precise knowledge like mathematics behind a field of activity. In trying to do this let me contrast research in numerical analysis to research in artificial intelligence. The vastness of the known relevant material in mathematics means that one cannot just sit down and write a paper with some new results in numerical analysis; rather a long apprenticeship is necessary. On the other hand, the ideas are available, the notation and style of presentation are well developed (one merely writes up the mathematical aspects and almost totally ignores the art and engineering judgment involved), so that it is in this sense easy to do research in numerical analysis. Just the opposite is true in artificial intelligence; almost anyone can sit down and imagine some new attack, but it is difficult to carry it out and especially difficult to present it in a written form suitable for publication. Furthermore, in the artificial intelligence area startlingly new results are of frequent occurrence and hence stimulating to further work. However, it is difficult to judge the significance of one's work and to know whether it is along a "good" or a "bad" path. In short there are advantages and disadvantages to having a well described body of knowledge as a background for activity.

I was asked to comment on what we did in the past that was wrong and what was right.

I believe one of the errors in numerical analysis is that we have too long tended to regard it as a branch of mathematics and to believe that the mathematical theorems were more relevant than we should have. Thus a corollary of the fundamental theorem of algebra states that 1, x, x^2, ..., x^n are linearly independent in any interval, but the Chebyshev polynomials show their dependence in the presence of noise; T_{21}(x)/2^{20} = x^{21} + ... and is less than 10^{-6} in -1 ≤ x ≤ 1. Of the right moves, we have been aggressive in opening new areas of thought like linear programming, Monte Carlo methods, etc. and if at times we have overestimated their importance that was far better than to have underestimated them. Another error we have made is to fail to produce textbooks in various areas such as Monte Carlo methods and hence the newcomer or outsider has difficulty in getting started easily.

I suppose I am expected at this point to take a grand view of scientific computing and briefly summarize years of development by thousands of workers in the world. It is an impossible task to do justice to, but here goes.

The heart of scientific computing, namely numerical analysis has had a fairly steady, reasonably controlled growth in the past decade. Three main classes of problems, often overlapping, namely, simulation, optimization, and combinatorial-integer problems, have given rise
to new disciplines and will probably spawn even more in the future. At present there are dozens of areas of specialization; in the near future there will probably be hundreds. The process of finding algorithms for areas previously considered as requiring thinking has also produced whole new fields such as theorem proving, drafting, language translation; more will arise. The problems of becoming acquainted with what is known is now hard, and it will only become harder in spite of all we manage to do with machines to help out. While the need for good, clear, simple presentations of known and new results is increasingly great, very few people seem willing to do much about it.