QUARTERLY OF APPLIED MATHEMATICS
APRIL, 1972

SPECIAL ISSUE: SYMPOSIUM ON
"THE FUTURE OF APPLIED MATHEMATICS"

MATHEMATICAL APPLICATIONS, COMPUTATION, AND COMPLEXITY

BY

HIRSH COHEN

IBM Thomas J. Watson Research Center, Yorktown Heights, New York

In this lecture I want to discuss some matters which are not themselves applied mathematics but are about the profession and practice of applied mathematics. In the trade, the measures of an article's quality are the sparsity of the prose, the swiftness of the mathematics, and the sweet connection to the real world. This one will be rather thick in prose and sparse in mathematics. I would have liked to have been able to respond to Prof. Davis's invitation to be philosophical, but that has always seemed to be a dangerous way to behave publicly. I offer these comments from the point of view of an industrial mathematician and as one of the Old Browns.

When I arrived at Brown nearly twenty-five years ago, the world looked a little different to a young applied mathematician than it does to a middle-aged applied mathematician now. It was clear that what was to be learned at Brown was continuum mechanics and the complex function and differential equation theory that was a part of it. Fluid mechanics, elasticity, plasticity—we were beginning to grapple with those problems that von Karman had said we were to in his 1940 Gibbs lecture [1]. We were really not yet "tooled up" for nonlinearity, as he urged in the first issue of the Quarterly of Applied Mathematics [2], but there was a realization of its importance. There was an atmosphere of the beginning of something at Brown in the late 1940s and, in fact, it was a beginning—a feeling that there were many, many problems to be solved in aerodynamics, flutter and aero-elasticity, viscous flow and boundary layers, plastic design, vibration problems. These were obviously good problems to solve because someone, somewhere needed the solutions, or they were hard, or they were the next problem in their field, or just because they were there.

We were (and, of course, I'm aware of that benevolent distortion that occurs in looking backward this way) an optimistic crew who had a great deal of confidence in the results, techniques, and styles of thought of several generations of European applied mathematicians. One can observe that for American science, applied mathematics was a rather new field then, not very populated or popular but with an apparent good potential for growth. Furthermore, when scientists thought of the uses of mathematics, they were used to thinking of the applications of classical analysis to the problems of the physical sciences.

In a corner of the house at 27 Brown Street there was a large room full of girls and machines—our computers. It was a small but busy and necessary enterprise. Other important locations were the softball field and the house on Benevolent Street where the Friday afternoon post-colloquium sessions were held.

Before nostalgia overcomes, what happened to all of that? Where are we?

What I think has happened is that the needs for and uses of mathematics have opened up very quickly and very dramatically during these twenty-five years. Science, in general, received a large amount of support; mathematics, in general, participated; and applied mathematics did too. Technological enterprise in government and industry bloomed, and so did the number of people involved in the applications of mathematics.

A great deal of this had to do with the mathematics of continuum mechanics and therefore led to grappling with the nonlinear problems that von Karman had pointed towards. But there were other mathematical activities coming into being as well. The new methods of discrete optimization, linear programming for example, were rapidly developed. A number of other applied combinatorial fields came into vogue—game theory, graph theory, queueing theory, with applications in various military, business, economic, and political operations. Control theory, in both its continuous and discrete form, was evolved mostly originally in response to aerospace needs but then applied in many industrial and finally even social processes. Numerical analysis, an old science, advanced practically and theoretically. Operations research and systems analysis, for better or worse, began to invade previously nonmathematized fields. As the number of people working with these techniques increased, so did the areas of application for them.

The largest effect on applied mathematics, however, came with the development of computers. In the late 40s, our computing center at Brown typically handled formulas that were the result of considerable analysis and manipulation. Gradually, as computers became faster, could handle more data in storage, and gained more input-output facilities, more problems were formulated so as to be done numerically at an earlier stage. We have learned now that the amount of analysis we do on a problem has to be carefully balanced against the ability to compute. There are classes of problems, for example in mathematical programming, in which the size makes it clear from the beginning that computer solution is necessary. In some modern physically-based problems that are formulated in terms of differential equations, the approximate methods of applied mathematics, in some cases, cannot make any headway at all. This may be because of the size, again, but is more likely because of strong nonlinearities or even linear interactions that are complex.

There are still other kinds of problems in which one is unsure of the formulation. The model itself may be in doubt. With the computer, we are now able to experiment with formulations, look at numerical solutions, and try new formulations in a selective fashion. There are interesting and useful "black box" problems in engineering, biology, ecology, and elsewhere, of this kind.

Finally, when one knows only that there is a problem and does not even have a mathematical model in terms of a set of equations, there is a collection of techniques known as simulation. These often have to do with particle motions or their counterparts (automobile movements, war personnel, or vehicles), and often models arise from such simulations.

All of these uses of the computer in the old and new realms of applied mathematics mean to me a gradual usurpation of some of the functions of the applied mathematician. There are some formulations, some analyses, and some solutions that are just more easily and more quickly accomplished with computation.

At this point, let me draw two observations from the description of change over twenty-five years. First, the application of mathematics to physical problems, and especially to problems in continuum mechanics, has continued to be an active and fruitful preoccupation for applied mathematicians. However, although it is undoubtedly growing, it has become a small fraction of the total uses of mathematics already and will probably become a smaller one. Second, the computer has usurped much of the work of the applied mathematician.

Both of these observations point to a change in what applied mathematicians do and, especially, in how we will train applied mathematicians for the future.

It seems clear to me that our reaction should be:

1. To broaden the training toward new areas of application. In effect, to recognize that the application of mathematics to the physical sciences now only represents a small fraction of the uses of mathematics.

2. To understand that computation itself should be studied as a major branch of applied mathematics; not only numerical analysis but a much larger view of the whole computational process.

1. New applications. The fact is that if one looks around a bit, it is easy to find mathematical formulations in almost every kind of endeavor. I am not speaking of data analysis although, in my mind, that topic represents one end of the spectrum of activities in mathematical formulation. There is, as I have pointed out elsewhere [3], a long history of mathematics in physiology. There are uses in other areas of biology and medicine. There is the famous marriage theory [4] in sociology, work in linguistics [5], economics, many areas of commerce and business, many topics covered by operations research and systems analysis-like formulations. There are, however, no successes like those of the physical sciences. Is it because mathematics and physics have grown up together? Is it because this was the part of the world men had most pressing need to understand and, therefore, gave the most effort toward? Or is it because on our space and time scale, these questions of physics are easier to put into mathematical terms and to solve? I suppose the fact that we have gotten by with linear descriptions of physical phenomena for such a long time may be one fact favoring the last view.

Whatever the reasons, in those areas where mathematics has had the greatest successes, one also finds the largest incursion of computing. There were elegantly appropriate uses and developments of eigenvalue methods that took place analytically in the early part of this century to find their place in quantum mechanics and many kinds of vibration and spectral theory. One now finds massive calculations undertaken for the solution of Schroedinger's equation or for many areas of continuum mechanics. In fact, calculations in these fields are now regularly part of design practices in aerodynamics, structures, nuclear energy, etc. In meteorology, oceanography, solid geophysics, hydrology, perturbation theories or other approximations show so little of the complex phenomena that people in these fields have turned to full- and large-scale computation.

If such is the case, that the classical realm of the applied mathematician has gone over very much toward computation, then, those who desire to continue using the beautiful approximation techniques of applied mathematics will have to look for new application areas. Similarly, for those whose creativity lies in formulation I am saying that I believe that all of those powerful techniques of approximation—regular and singular perturbations, integral approximations, comparison and bounding methods, etc.—must now be taken to more fertile fields.

How to move? Surely, it's not quite as simple as just picking up the bag of tricks and carrying it to the new subjects.

What I think is required is literally a mathematical immersion into new application areas to the same depths that were reached in the physical sciences. Applied mathematicians have come to know a great deal about fluid dynamics. From this have come precise, detailed explanations of very complicated phenomena. We observe how this in-depth mathematical knowledge is now being carried forward into oceanography, astrophysics, meteorology and other parts of the physical world that depend very strongly on fluid flows. On the other hand, I do not feel the same kind of impact is being made in applying fluid dynamical mathematical techniques to problems of blood flow or desalinization—yet. It will require more serious penetration into these areas by those who wish to make their mathematical contributions really meaningful.

For mathematics to become more effective in the social sciences and in social enterprises, I believe the same commitment is required. Instead of working on the fringes of fields, we shall have to learn as much about them as the social science specialists in each case. For example, colleagues of mine have prepared some tools, mostly computational schemes, for dealing with some aspects of urban housing [6]. To understand where this particular calculation (which has certainly shown itself to be useful to some of the parties interested in low-cost housing) lies in the whole spectrum of problems requires learning something of the sociological as well as the financial aspects of housing and probably a good deal of the structural and architectural factors.

If I'm on the right track, the implication may well be that training in applied mathematics for some must include serious excursions into social and political areas. This will, of course, be hard to come by in mathematics or applied mathematics departments and, as has so often occurred in applied mathematics, we will undoubtedly see the work beginning within the application fields themselves.

Even so, to encourage other areas to mathematize themselves we need to teach mathematics in a fashion that is sympathetic to their needs. There are already some beginnings of this. For example, as part of a biomathematics development at the Cornell Medical College and Sloan–Kettering Cancer Research Institute in New York, courses were given to graduate students in the medical sciences (Ph.D.'s in biochemistry, physiology, pharmacology, etc.). These took the form of problems from various parts of biology and medicine in which one could make very explicit gains by using differential equations, probability methods, matrix and eigenvalue methods, and so on. The gains were important to show; my own experience has been that medical students show very little sympathy for nonlinear partial differential equation solutions for nerve pulses when they can literally see the pulses in oscilloscopes.

Other courses of this type have begun in medical schools and universities. They will play a useful role. They won't replace the immersion I spoke of, however. I can see no other way than for young people in applied mathematics to learn as much as they can about the rest of the world.

There is another aspect of change in applied mathematics that I would like to mention. I would like to claim that we have so far mostly dealt with relatively simple systems. When we turn to sciences other than the physical sciences, we are confronted with greater complexity in structure, relationships, a larger number of variables, and often, it seems, a higher degree of interdependence and hence nonlinearity. I can't prove this. It may only seem to be the case because we have not yet perceived how to simplify and structure other areas as we have done in the physical ones.

It is clear that we do not even have very good measures of the difficulty of such structuring. With the mathematics of the physical sciences, there are far-reaching physical principles firmly tied to concepts of energy, entropy, work, etc., which enable us to look for optima, measure efficiency in the physical processes themselves and, in turn, in the methods of solving the problems. This is not the case in other sciences or in other nonscientific enterprises. We may want to create human good, avoid delay, raise profit, distribute goods more uniformly, increase democratic participation, make life easier, or survive. All valid objectives in their own context, but how do we examine the difficulties in solving problems? How close do we get to solutions? How efficient or effective are our solutions in these areas and how efficient are our methods of obtaining the solutions?

Let me take some examples from neurophysiology. As I have described elsewhere [3, 7], there is a good theory for how nerve pulses travel along nerve fiber membranes. This nonlinear system of partial differential equations uses, directly, physical relations and, empirically, chemical kinetic data and hypotheses. Now the familiar form of these Hodgkin–Huxley equations applies to nerve membrane that is continuous and continuously active along its length. In our own bodies and in most of the nerve systems of vertebrates, nerve membrane is interrupted along the length of the nerve fiber. In fact, the active energy input for the maintenance of the pulse propagation occurs only at the interruptions which are called nodes of Ranvier. The voltage across the membrane obeys a passive diffusion equation in between the nodes. The nonlinear system becomes an ordinary differential equation set at each node. The nodes may be 1μ in length and the regions between them at least 2 mm, so that the nodes can be looked upon as points along the line. Again, one would like to be able to calculate the nerve pulse propagation and, especially, how it is affected by changes in inter-node length, chemical environment, membrane deterioration, etc.

The equation that governs the voltage across the membrane, $v(x, t)$, between the nodes is

$$C(\partial v/\partial t) + (v/r_s) = (a/2r_i)(\partial^2 v/\partial x^2), \qquad x_i < x < x_{i+1}\,, \tag{1}$$

where the x_i are the node locations, C = membrane capacitance (fd/cm^2), r_s = cross membrane specific resistance (ohm cm^2), r_i = specific resistance of material inside the fiber (ohm cm) and a = radius of nerve fiber.

At the nodes $x = x_i$ the voltage $v = v_i(x_i)$ is given by

$$C_n(\partial v_i/\partial t) + I_m(v, m, n, h) = I_i\,,$$

$$I_m = m_i^3 h_i \bar{g}_{Na} v_i \frac{[\exp\,(v_i - \bar{v}_{Na}) - 1]}{\exp\,(v_i) - 1} + \bar{g}_K n_i^4 (v_i - \bar{v}_K) + I_L,$$

$$I_i = (\lim_{x \to x_i + 0} (\partial v/\partial x) - \lim_{x \to x_i - 0} (\partial v/\partial x))(1/r_n),$$

$$I_0 = (1/r_n)(r_n I_e(t) + 2 \lim_{x \to 0} (\partial v/\partial x)),$$

$$dm/dt = (1/\tau_m(v_i))(m - m_\infty(v_i)),$$

$$dh/dt = (1/\tau_h(v_i))(h - h_\infty(v_i)),$$

$$dn/dt = (1/\tau_n(v_i))(n - n_\infty(v_i)),$$

where C_n is node capacity, I_m is the transmembrane current, I_0 accounts for the stimulus I_e

at $x = 0$, r_n is the node specific resistivity, m, h, and n are variables that describe the behavior of sodium and potassium ion currents, $\bar{g}_{Na}$, $\bar{g}_K$, $\bar{v}_K$, $\bar{v}_{Na}$ are constants associated with these sodium and potassium flows, $\tau_m(v)$, $\tau_h(v)$, $\tau_n(v)$, $m_\infty(v)$, $h_\infty(v)$ and $n_\infty(v)$ are empirical functions that can be measured at the nodes for each type of nerve fiber, and I_L is a linear function of the voltage which accounts for the current of ions other than sodium and potassium.

Suppose the linear diffusion equation (1) is solved for the voltage in each internode region. The solution can be written down in terms of the boundary values for the voltage (unknown) at each nodal end of this inter node. From this expression, one can compute the derivatives at x_i^+ and x_i^- and use these to form I_i. If one then assumes that there is a pulse that propagates with velocity θ along the nodes, we can write

$$\frac{dv_i}{dt} + k\frac{C}{C_n} r_s I_m(v_i) = k\frac{C}{C_n}\frac{r_s}{r_n}\frac{2}{l}\sum_{n=1}^{\infty}\left(\frac{n\pi}{l}\right)^2 \int_0^t \exp\left(-k(t-\lambda)\right)$$
$$\cdot \exp\left(-\frac{n^2\pi^2}{l^2}(t-\lambda)\right)\left(2v_i(\lambda) - (-1)^n v_i\left(\lambda + \frac{l}{\theta}\right) - (-1)^n v_i\left(\lambda - \frac{l}{\theta}\right)\right) d\lambda$$

where $k = 2r_i/ar_s$ and l is the length of the internode. The problem is resolved into solving the system of ordinary differential equations to obtain the pulse shape as a function of time and to determine the value of θ that goes with this shape. Since I_m depends nonlinearly on v_i, it is a nonlinear eigenvalue problem.

This is not really an analytically easily tractable problem in this form; one can solve the ordinary differential equation numerically or return to the original equations and solve them numerically. One can also approximate by using a linear voltage change in the internode (assume the nodal voltage functions are slowly changing). It would be of interest to relate θ, the velocity of pulse propagation, analytically to the internode distance l and to the electrical and chemical parameters, but this has not yet been done. To me this is an example of an interesting problem where one probably should not carry analytical approximations very far.

Calculations of the original system have been carried out by FitzHugh [8] and Hardy [9]. Hardy is able to show the way in which speed of propagation is altered with chemical and electrical changes. Now the reason for mentioning the problem is that once such a calculation is in hand, one can begin to ask questions about the design of the nerve fiber. Why are the internodal spacings the way they are? Does it maximize pulse propagation speed? Or is it rather the total reliability of information transmitted in pulse form that is optimized? Is there a general principle of survival of organisms that applies in the small to a single nerve fiber the way equilibrium theorems in mechanics apply? Does survival depend on minimizing energy or does one need to observe the organism in relation to its environment and its internal energy needs to understand survival? And so on.

For all of these questions, one can see that the physics, chemistry, and measurable physiological function of single nerve fibers or active muscle membranes (which follow similar equations) can be treated rather well by classical differential equation methods. The information processing that the propagating pulses provide also seems to be tractable by communication theory methods of signal and noise processing [10]. Small collections of nerves and nerve-muscle interactions are harder [11], but one can see that sensory activities like vision [12] and hearing will be handled as we get more experimental data.

Large collections of nerve fibers that do higher neural functions for us are beyond us now. Can memory be studied mathematically? Can we connect behavior and neural response by a tight mathematical treatment? I think so, but I am not at all sure that the classical analysis that serves to do single nerve fibers will help. We will need new mathematical structures. Before this, however, we need to have mathematicians become very much at home inside fields like neurophysiology.

The mathematics of the physical sciences is well enough established and understood to allow for some measures of complexity of the physical phenomena. We have learned to construct sets of equations that follow natural or experimentally controlled behavior in a fashion that exhibits the degree of approximation involved. A very complete system of meteorological equations will include in the equations of conservation of mass, momentum, and energy, expressions for radiation and chemical exchange effects, ground and sea surface interactions of various kinds, gas-fluid mixtures, and phase transitions. Dimensional analysis and empirical knowledge of many of the factors allow for approximate systems to be developed. These may be approximations for different wave lengths of phenomena (large, global cyclones or small-scale behavior over crop fields), for different meteorological zones (hurricane behavior near the equator), or, say, for various degrees of vertical stratification (2-level models for gross behavior, 23-level models for greenhouse effect studies). In this and in other such subjects we have learned to understand such approximations eventually in terms of various kinds of expansions, perturbations, or asymptotic behavior, and so we have some measure of how complex the problem solved is compared to the whole problem. However, it is important to note that once we begin to operate computationally with machines on any level of such a problem, the full set of equations or one of the approximations, we do not yet have such measures of complexity and difficulty of the computational process itself. I will discuss this in the next section.

In a number of combinatorial problems there is also a measure of complexity. Problems of scheduling and distribution have a finite number of alternative modes to nominate for the optimal mode. Enumerative methods lead to a factorial number of possible trials, factorial, say, in the number of points n to be distributed to. In some cases with special structure, this may be reduced to polynomial growth in n. There is not a general theory for this kind of reduction. In other fields there are also measures that give indications of structure, complexity, or difficulty of solution. Information theory does this for certain modes of communication. For the real physical channels of electronic communication it was designed to model, it has proved to be useful. For other fields in which its application has been attempted, it seems to give only a very partial measure. For example, as Landauer [13] has pointed out, Shannon information theory deals with messages sent along a communication link; the logical systems in computers deal with streams of information which are made to interact nonlinearly.

I am suggesting that our profession, applied mathematics, must move in such a way that we learn mathematically as much about other sciences and nonscientific endeavors as we have about the physical sciences. I am also suggesting that we shall be moving into areas that are more complex than those we have already studied. We will need to understand something about the growth of this complexity for its own sake.

2. The applied mathematics of computation. The journal "Mathematical Tables and Aids to Computation" was probably appropriately titled and its contents were adequate when it was founded by Professor Archibald of Brown University and first appeared in 1943. Its first issues contained only information on new and old tables and met the needs of the relatively small number of people concerned. Gradually, some new algorithms began to appear, and later the journal became "Mathematics of Computation". There are now literally dozens of journals which deal with various aspects of computation. Few of them concentrate on numerical questions alone, and some make no reference to numerical methods at all. The readership is enormous, and so is the breadth of the computational problems discussed.

From a less literary point of view one might ask what is actually being computed in all of the computers. It turns out that about 20% of computer usage is for scientific problems. A greater amount of time is spent in sorting! There is a growing amount of computing that has to do with signal, image processing and other pattern and character recognition. However, most of the total computing capacity is used for filing and retrieving data for a variety of business transactions. In these, the amount of mathematical manipulation is small and not at all complex—some arithmetic in a payroll, for example—but there is a large-volume continuous stream of such data processing. Modern systems interleave programs within a batch and may, indeed, direct programs onto more than one central processor. The aggregate becomes a large information flow system—many activities all moving in discrete steps—and is indeed extremely complex when considered as a whole.

What this means is that there is a huge mathematical-like endeavor, involving many machines and people, and most important, growing at a very high rate. Growing, in fact, much faster than the polynomial-difference mesh-matrix manipulation-based science calculations. One could say that the science of computing of this general type is properly the realm of the computer sciences. There is a great deal of mathematics to be done, however, probably a great deal more than there is to do in the classical areas of mathematical application. It seems reasonable to suggest that applied mathematics should reach out and include some portion of this field. At the least, we should be very much aware of its growth rate and the amount and kind of mathematical work there is to be done, for the benefit of those we hope to train in applied mathematics.

In another article in this issue Prof. Perlis describes many of the problems of computation in the broad sense that I intend here. I want to add emphasis to several of these.

First, in moving a problem onto a computer, no matter whether it is a complicated nonlinear partial differential equation or a simple accounting or sorting procedure, one must have an algorithm. The study of algorithms for their own sake has gone through an abstract phase, but I would like to mention some aspects of the algorithms of real computational processes. There has been, in fact, a movement away from the study of algorithms on automata or Turing machines toward algorithms as they are actually performed. There always have been and will be algorithms made in an *ad hoc* fashion in order to carry out particular calculations. However, with much greater emphasis on efficiency we need measures of quality of real computational algorithms. As Knuth [14] has pointed out, there are two approaches: consider a very frequently used algorithm and analyze it in great detail in terms of arithmetic executions and storage and data requirements; or consider the entire class of algorithms that all succeed in doing a particular task and choose the "best" of these.

Knuth has given, as an example of the first approach, an analysis of reordering a set of indexed variables; i.e., replacing $(x_1, x_2, \cdots, x_n)$ by $(x_{p(1)}, x_{p(2)}, \cdots, x_{p(n)})$, where p is a permutation of $\{1, 2, \cdots, n\}$. p is given and auxiliary memory is finite. Such an algorithm might be required in transposing a rectangular matrix or unscrambling a finite Fourier transform. The measures of labor may be in terms of running time, or adds and multiplies, or compares, or storage usage. In this case, Knuth finds that the running time is of order $n \ln n$ operations on the average, but the worst case is of order n^2. Such familiar operations as iterative schemes [15], polynomial zero-finding [16] and matrix operations have been studied in similar fashion. These studies have attempted to remain analytical and often result in, as Knuth has remarked, interesting old-fashioned algebra problems. There is an empirical version of this single algorithm vein of analysis. One codes a problem, runs it, and times the result. It is, in fact, the standard way of testing not only algorithms but computing machines and systems.

The second approach has the difficulty that one must find the entire class of algorithms that does the task. For a mathematical operation of any degree of complexity this is hard to do. It has been accomplished in a few examples although, as I will note below, there are always some constraints on the generality. Before giving these examples, it is important to relate the ideas of efficiency we have mentioned with complexity, difficulty, and limits. We have been talking about explicitly measuring the work of a computation in terms of the number of basic arithmetic operations performed. We do not have any other measures except perhaps the time it takes to complete an algorithm. The state of this art seems to be to measure difficulty in terms of basic computer labor—arithmetic steps. The concept of "best" in these terms is then the minimum labor required. It may be a lower bound on the number of operations required and is, therefore, an ultimate limit. It may, on the other hand, be an upper bound, to show that one doesn't have to do more than a certain amount of work. By implication, on this level, complexity of an algorithm is also measured by the amount of work to be done. These are not very pretty definitions. They are also not related to the amount of real energy used in the computation. Such physical measurements of computing and the physical limitations on speed, power, size, etc., have been studied by Landauer [17], Keyes [18], and Freiser and Marcus [19].

There are more sophisticated notions of the complexity of functions related to their computability. There is a body of work that deals with the covering of functions and function spaces and uses the notion of ϵ-entropy to measure complexity [20, 21]. As mathematically attractive as these ideas are, they have not yet had a large impact on real computational algorithms.

The best algorithms in terms of number of operations can be related to the best times by lower bounds on the times of arithmetic operations that have been given by Winograd [22]. Formulas of this kind assume the physical switching time of each logic circuit is known and then develop the minimum number of logic stages required to do simple adding, carrying, multiplication, and comparing. For example, the minimum time for addition is given by $\tau_{\text{add}} \geq \Delta \log_r 2[\log_2 \alpha(N)]$ where r is the number of wires that enter any one of the logic circuits involved (only one wire exits), N is the word size, α is a function related to the primes of N, and Δ is the switching time of the circuits involved. This happens to be the special case where the logic is two-valued, but d-valued logic has also been done. There are also results for the other basic arithmetic operations. They are very general in that they hold for any encoding of the numbers.

The best algorithm for polynomial evaluation (including scalar product) of any degree polynomial is known [23, 24]; n multiplications are necessary for a polynomial of degree n. For a given polynomial, to be evaluated a number of times as in a special function subroutine, the method can be especially tailored to give even better results than the "best" for polynomials of that degree. The valuations must be done enough times to pay for the tailoring. An attempt to find a lower bound for matrix multiplication has not yet resulted in a best algorithm but has shown the surprising result that two $n \times n$ matrices can be multiplied in fewer than n^3 multiplications. Winograd [24] first found an algorithm that, for large n, requires $(n^3/2) + n^2 \cdots$ multiplies (the number of additions is increased from $n^3 + n^2$ to $\frac{3}{2}n^3 +$ lower order terms; addition is usually several times faster than multiplication in computers). Strassen [25] then discovered an algorithm in which the number of multiplications goes like $n^{\ln_2 7}$. This has not been shown to be best as yet, but the whole experience has been an interesting one once the right questions were asked. A study of best iterative methods has begun. Wolfe and Winograd [26] have considered a class of iterative schemes and established upper bounds on the speed of convergence. As for matrix inversion and linear equation solving, there are no very satisfactory results. It has been shown [27] that if one restricts the algorithms to those which use only row interchanges, then n linear equations require at least $\frac{1}{3}(n^3 + 3n^2 - n)$ multiplications. Winograd [24] has shown that this can be reduced to $(n^3/6) + \frac{3}{2}n^2 + (10n/3) - 8$. There are no extensive findings on these kinds of operations, however.

These have been a few results—actually, most of the results available—for numerical algorithms. On the nonnumerical side, there are results on best algorithms for sorting. Some years ago, A. Gleason found that one cannot sort a set of n items using comparison alone in less than $\log_2 (n!)$ compares. There are constructive proofs that show you can actually attain $\log_2 (n!) + 0.0861n$ compares. These bounds literally only count compares, however, and do not take into account the work required to rearrange data. There are other somewhat more practical limits associated with particular algorithms where the expected number of compares approaches $\log_2 (n!)$ as n becomes very large [28]. For finding the median of a set of n numerical values, it is known that the worst case will be of order n [29]. These sound like simple problems, but they are such heavily used algorithms that it is important to begin to understand them.

You might ask how one determines which are the important algorithms. There are no good answers to that question at present. There are algorithms that come into very heavy usage because of the way applications problems are formulated mathematically. Matrix algorithms are one example, sorting is another. There is another class which plays a dominant role in running the computer itself; for example, certain scheduling algorithms may be used over and over again in a multiprogrammed system. The operating system of a large computer system has to do a large amount of computing to run the system. Simpler manipulations that have to do with index updatings or data transferring from one storage level to another may have even heavier use. As far as I know, no one has carefully analyzed such patterns of usage.

The kind of approach to real algorithm analysis I have briefly described is only just beginning. It offers practical results for immediate use—the surprise when a new bound is better than old algorithms—and seems to offer theoretical insights into the computational process.

I've mentioned operating systems. These are very large engineering entities that require thousands of lines of code in their preparation. They must run with great reliability. There are really only a small number of these large programming systems that run now, but we will be producing them in the fashion that we now produce other major engineering projects—large bridges, major building programs, or transportation systems. They engage literally thousands of people and a great deal of money. I suspect that they are as complicated in their construction as the largest projects that men have built—including aeronautical and space systems. With all of this there is very little known about the mathematical structure of such systems. Some simulations have been accomplished, but it is hard to understand how accurate they are or what their predictive value is. Some particular parts of operating systems have been analyzed: the language compilers that translate from programmer to machine language, portions of the memory and storage functions. Even these theories are in their initial phases, and there are no overall models for design and optimization. It is an area in which there is much mathematics to do and in which there is a real entity to be modeled, explained, and improved.

I've also mentioned the very large amount of all computing that has to do with filing and retrieving information. This may be the raw data that are used in the applications themselves or it may be the computer's own operating system, filed as data in a portion of its memory, or some intermediate form of data bank. For any particular file, say a personnel file with a number of categories and an entry in each category—age, sex, marital status, job category, etc.—a clever programmer can analyze and probably optimize filing and retrieving in this file. However, if he has an ensemble of files which are coupled in their content, optimization becomes difficult. If, further, one faces the question of a very large collection of such ensembles in designing a file-handling system and then tries to understand its dynamic operation under a variable programming load, the problem is very hard. The world is full of files, and more and more of our activity will find us interacting with them. What mathematical structure do they have or can they be given? There has been a great deal of empirical construction in this field so that there are effective data base operations. But there has been very little mathematical structuring that is close to the real computational problem. Some attempts have been made to impose various aspects of graph theory, finite geometry methods, or combinatorial ideas on the structuring of data, but these have not been very good matches. It is an area that is growing faster than any other, but it is so far from classical mathematical applications that it has not yet received much attention.

Here, then, are three topics in computation—algorithmic complexity, operating system modeling, and data structuring—which could use much heavier mathematical participation than they now receive. Their requirements are far from the kind of approximative analysis that has been so successful in the physical sciences. But they will be large areas of endeavor with many people engaged in their engineering development and in their use—larger than most of the uses of mathematics we now have. I think we shall have to broaden our idea of what applied mathematics is to include these subjects.

I have said a bit about the complexity of certain algorithms, limitations on calculating, and also how little we know about the complexity of other mathematical structures and their applications in computing. I would like to turn briefly to the connections between the kinds of computation or information processing we do physiologically and what we can do electronically and mechanically.

First of all, just as one has fundamental physical constants that serve to set limits, one can ask if there are such limitations for computing. Winograd's limits, lower bounds on the time for arithmetic operations, are one kind. Physically, using the idea that the unit of energy required to switch from one state to another is $kT \ln_e 2$, Landauer [17], Marko [30], and others observe that there are something like 10^{93} bits available for information from all of the mass of the universe. The maximum rate at which these might be processed has been calculated by Bremermann [31] to be 10^{47} bits per gram per sec.

One would think these numbers give a great deal of leeway in terms of desired computing ability. Modern systems may have bit storage systems that go up to 10^{12} or 10^{14} bits. Bit-handling rates are only of the order of 10^7 or 10^8 bits per sec. A computer whose operating system can handle a million instructions per second may require input at a bit per instruction, which can be supplied. On the other hand, there are combinations of physically available entities that give numbers that are much larger. For example, Ashby [32] points out that a 20 × 20 grid of lamps can present 10^{120} different pictures, being only selectively on or off. The number of different possible variations of human genetic structure have been calculated by Bremermann to be $10^{2.4\times 10^9}$.

Physiologically, our sensory apparatus allows for bit rate entry not much different from computer storage transfer rates. It is estimated that the visual system can absorb of the order of 10^7 bits/sec. The other sense organs operate at smaller rates. There are, perhaps, 10^{10} neurons in the central nervous system, a part of which must represent storage capacity. But the real information-processing rates are very low. For reading and hearing for example, we may process 40 bits/sec. We may absorb numbers at 3 bits/sec. Obviously, our internal nervous system either carries out a formidable kind of data processing in order to do these tasks or we have a great deal of redundant processing power or we operate very far from peak capability. It would not be unexpected for our sensory and cerebral systems to have a kind of reserve that could handle much more data to survive in the face of surprise or danger. It makes one wonder, however, how we might get man to work normally closer to his higher capability.

This kind of discussion does not sound like applied mathematics. But I claim that to do the mathematics of neurophysiology, say, one must immerse himself in these kinds of data. I think it will be a great deal more important than the nonlinear differential equation models that I have mentioned before in connection with nerve fiber nodes.

I'm hopeful that the studies I've mentioned that have to do with limitations on computing algorithms and on physical limitations of computing will grow and that there will be counterparts in other new applications of mathematics. I have little doubt that more and more of the world will be mathematized and computed. I hope we, the applied mathematicians, can keep up.

References

[1] T. von Karman, *The engineer grapples with nonlinear problems,* Bull. A. M. S. **46**, 615 (1940)

[2] T. von Karman, *Tooling up mathematics for engineering,* Quart. Appl. Math. **1**, 1 (1943)

[3] H. Cohen, *Mathematics and the biomedical sciences,* in *The mathematical sciences: a collection of essays,* ed. National Research Council's Committee on Support of Research in the Mathematical Sciences (COSRIMS), The MIT Press, Cambridge, Mass., 1969, pp. 217–231

[4] A. Weyl, Appendix to C. B. Lévi-Strauss, *Les structures élémentaires de la parenté,* Presses universitaires de France, Paris, 1949

[5] N. Chomsky, *Formal properties of grammars,* in *Handbook of Math. Psych.,* ed. R. D. Luce, R. R. Bush, and E. Galanter, John Wiley, New York 1963, Vol. 2

[6] D. C. Gazis, *A computer model for the financial analysis of urban housing projects,* Socio-Econ. Plan. Science **5**, 125–144 (1971)

[7] H. Cohen, *Nonlinear diffusion problems,* in *MAA Studies in Mathematics, Vol. 7, Studies in Applied Mathematics,* ed. A. H. Taub, Prentice-Hall, Inc., Englewood Cliffs, N. J., 1971, pp. 27–64

[8] R. FitzHugh, *Computation of impulse initiation and saltatory conduction in a myelinated nerve fiber,* Biophys. J. **2**, 11–21 (1962)

[9] W. L. Hardy, *Propagation speed in myelinated nerve: dependence on external sodium,* Ph.D. dissertation, University of Washington (1969)

[10] R. Stein, *The role of spike trains in transmitting and distorting sensory signals,* in *The neurosciences second study program,* ed. F. O. Schmitt, Rockefeller University Press, 1970, pp. 597–604

[11] D. Kennedy, *Nerve cells and behavior,* Amer. Sci. **59**, 36–42 (1971)

[12] F. Ratliff, *Mach Bands,* Holden-Day, New York, 1965

[13] R. Landauer, *Cooperative effects, soft models, and fluctuations in data processing,* Ferroelectrics **2,** 1 (1971)

[14] D. Knuth, *Mathematical analysis of algorithms,* IFIP Congress, Ljubljana (1971)

[15] J. Traub, *Iterative methods for the solution of equations,* Prentice-Hall, Englewood Cliffs, N. J., 1964

[16] B. Dejon and P. Henrici, ed., *Constructive aspects of the fundamental theorem of algebra,* Proc. of Symp. at IBM Zurich, June 1967, Wiley-Interscience, New York, 1968

[17] R. Landauer, *The future evolution of the computer,* Physics Today **23**, 7, 22 (1970)

[18] R. Keyes, *Power dissipation in information processing,* Science **168**, 796–801 (1970)

[19] M. Freiser and P. Marcus, *A survey of some physical limitations on computer elements,* IEEE Trans. on Magnetics **MAG-5**, 82–90 (1969)

[20] A. G. Vitushkin, *Estimation of the complexity of the tabulation problem,* Fizmatgiz, Moscow, 1959 (Russian); translated: *Theory of the transmission and processing on information,* Pergamon Press, New York, 1961

[21] A. N. Kolmogorov, *On the representations of continuous functions of several variables in the form of superposition of continuous functions of one variable and addition,* Dokl. Akad. Nauk **114**, 679–681 (1957) (Russian)

[22] S. Winograd, *On the time required to perform addition,* J. Assoc. Comp. Mach. **12**, 277–285 (1965)

[23] V. Ya. Pan, *Methods of computing values of polynomials,* Uspehi. Mat. Nauk **21**, 103–134 (1966) (Russian); English translation in Russian Math. Surveys **21**, 105–136 (1966)

[24] S. Winograd, *On the number of multiplications necessary to compute certain functions,* Comm. Pure Appl. Math. **23**, 165–179 (1970)

[25] V. Strassen, *Gaussian elimination is not optimal,* Numer. Math. **13**, 354–356 (1969)

[26] S. Winograd and P. Wolfe, *On the rate of convergence of local iterative schemes,* in *Theory of Machines and Computations,* Academic Press, Inc., N. Y. and London, 1971, pp. 67–70

[27] V. V. Kljuev and N. I. Kokovkin-Shcherbak, *On the minimization of the number of arithmetic operations for the solution of linear algebraic systems of equations,* Zh. Vychisl. Mat. i Mat. Fiz. **5**, 21–23 (1965) (Russian); English translation in U.S.S.R. Comput. Math. and Math. Phys. **5**, 25–43 (1965). Also translated in Technical Report CS24, Department of Computer Science, Stanford University (June 1965)

[28] W. Frazer and A. McKellar, *Sample sort: a sampling approach to minimal storage-free sorting,* J. Assoc. Comp. Mach. **17**, 496–507 (1970)

[29] An as-yet unpublished result of M. Blum (Berkeley), R. Floyd (Stanford), R. Tarjens (Cornell), and others.

[30] H. Marko, *Physikalische und biologische Grenzen der Informationsübermittlung,* Kybernetik **2,** 274–284 (1965)

[31] H. J. Bremermann, *Optimization through evolution and recombination,* in *Self-organizing systems,* edited by M. C. Yovits *et al.*, Spartan Books, Washington, D. C., 1962, pp. 93–106

[32] W. Ashby, *Some consequences of Bremermann's limit for information-processing systems,* in *Cybernetic problems in bionics,* Bionics Symposium, 1966, ed. Hans L. Oestreicher, Gordon and Breach, N. Y., 1968, pp. 69–76