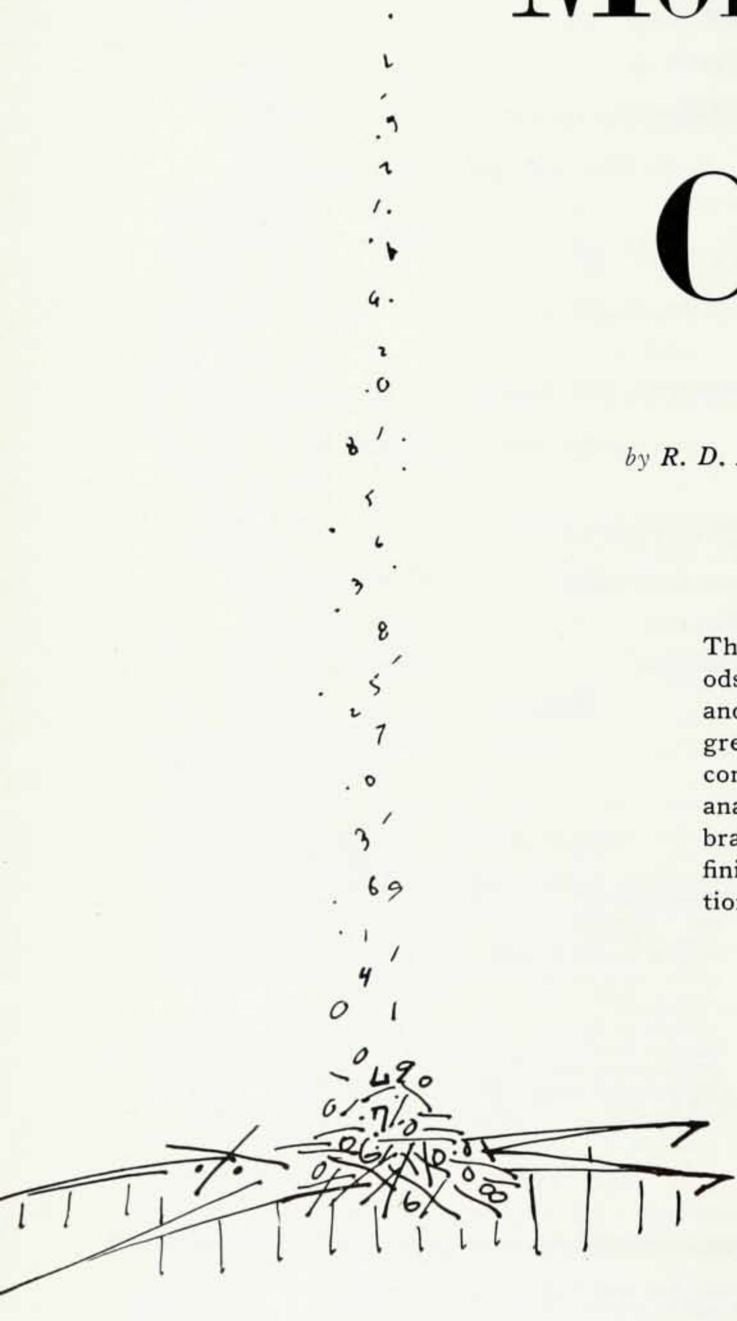
Modern Computing

by R. D. Richtmyer and N. C. Metropolis

The intricacies of automatic computing methods have been popularized by pictures, visual and verbal, of complicated wiring diagrams, great banks of electron tubes, and dramatic control boards, as well as by certain romantic analogies between the machines and the human brain. There remains, however, a need for defining the limits of computing machine operation, as well as its promise.



Sketches by Paul Bond

The physical sciences, notably physics and astronomy, have come in the past few decades to depend heavily upon the various devices intended to make the solving of mathematical problems less arduous and more rapid than in the days, and with the paper and pencil methods, of the ancients. The methods, of course, have themselves seen little change; yet machines capable of more and more speed in numerical computing make it possible to calculate problems that before would have been solved only approximately if at all.

And as with most sudden technological advances, this presents pitfalls in its use: Professor Courant, for instance, has mentioned the folly of thinking that if one knows merely the elementary laws of Newtonian mechanics, and has available a sufficiently large computing machine, he can (or should want to) calculate the motion of every drop of water in Niagara Falls. A machine will give dull answers if it is asked dull questions.

Engineering, Method, Logic

Automatic computing methods, during the last decade, have developed in several directions, with rapid progress involving engineering principles and techniques, mathematical methods, and the logic of automatic computation.

Quite generally there are two broad classes of computing machines. In one, the so-called analogue devices, a mathematical variable is represented by a physical quantity (voltage, current, angle of rotation of an axle, or the like) having a smooth and continuous range of variation. An example of this class is the differential analyzer. In the other class are the digital computers, where a mathematical variable is represented by a set of components, in each of which a physical quantity can assume only discrete values; the values assumed in the successive components of the set represent the corresponding digits of the variable. This class includes desk calculators and punch card equipment.

In what follows we shall be primarily concerned with digital computers and then primarily with the electronic (containing vacuum tube circuits, such as pulse generators, binary elements, memory organs, etc.) rather than the mechanical (which may contain electrically operated parts such as motors and relays). The development of electronic tubes and circuits now makes it possible to add two ten decimal-digit numbers in five to ten microseconds or to multiply them together in a few hundreds of microseconds. The result of this computation can then be stored, along with thousands of other similar numbers, in electrostatic storage tubes, where it will be available, on a few tens of microseconds' notice, for further computation. It is with electronic computers that the most startling advance in engineering principles and techniques has been made. For a discussion of computing machines the reader is referred to the extensive literature, and especially to the following: A. W. Burks, Proc. I.R.E. 35, 756 (1947); F. C. Williams, T. Kilburn, Proc. I.E.E., Part III, 96, p. 81 (1949); I. Auerbach, J. Eckert, R. Shaw, and C. Sheppard, Proc. I.R.E., 37, 855 (1949).

The basic vacuum tube circuit used in electronic computers is essentially an on-off device. In consequence the use of the binary rather than the conventional decimal system for representing numbers is regarded by many as both natural and advantageous. In the decimal system the digits run from 0 to 9 and are understood to be multiplied by appropriate powers of 10. In the binary system, the digits are 0 and 1 only, and are understood to be multiplied by successive powers of 2. For example the decimal number 19 is represented in binary notation by 10011, which stands for $1(16) + 0(8) + 0(4) + 1(2) + 1(1)$.

Such operations as multiplication and division are very much simplified when performed electronically in this system, and it is easy to convert numbers from one system to another, simply regarding the conversion as the initial and final steps of every problem. Thus decimal numbers may be introduced into the machine, and results got out in the same form, without the casual bystander ever knowing that the conventional decimal system has become passé. Since the number of numbers that have to be so converted is usually very small compared to the number of calculations performed, the extra trouble is negligible.

A second stream of development in automatic computing in the last decade has been in the purely mathematical problems of numerical computation. The importance of developments of this sort can be

R. D. Richtmyer and *N. C. Metropolis*, theoretical physicists at the Los Alamos Scientific Laboratory, have both been concerned with using automatic computors to solve complex problems in physics. Richtmyer, who studied at Göttingen, Cornell, and the Massachusetts Institute of Technology, has taught physics at MIT and Stanford and during the war was employed by the Navy Department and the OSRD. Since that time he has been at Los Alamos. Metropolis, formerly of the University of Chicago, has been associated with the atomic energy project since its early days.

shown by a few examples. Every high school algebra student knows how to solve n linear equations in n unknowns; and if n is not greater than, say, 4 or 5, the solution is readily performed. If n is as large as 50 or 100, the same methods apply in principle, but in practice not only is the amount of work involved very great, but some of the standard methods actually fail because of an unexpectedly disastrous accumulation of round-off errors, unless a prodigiously large number of digits is retained at each step of the calculation. Von Neumann and Goldstine have analyzed this problem in detail and have indicated numerical methods that are satisfactory under stated circumstances.

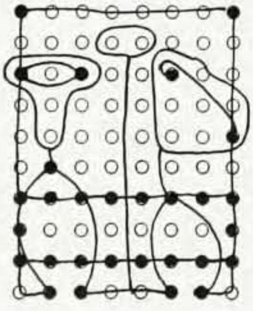
As a second example, the numerical solution of partial differential equations is frequently performed by assigning a certain small interval size to each of the independent variables, then replacing the derivatives by finite-difference quotients and solving the resulting algebraic equations. The idea is here, just as in the familiar stepwise solution of ordinary differential equations, that if the interval sizes are made small enough, the result should be a good approximation to the solution of the original differential equation. Courant, Friederichs, and Lewy showed in 1928 that for partial differential equations this is not generally the case, unless certain restrictions are imposed on the *relative* interval sizes for the several independent variables. If these restrictions are disregarded, the solution of the difference equations does not in general approach any limiting function whatever as the interval sizes tend to zero. This phenomenon is known as instability: research is continuing on the development of stable stepwise methods for the solution of partial differential equations.

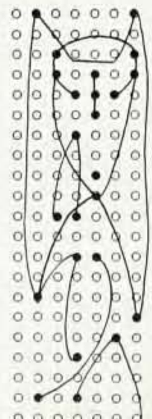
As a last example, certain partial differential and integro-differential equations can be solved approximately by the recently developed "Monte Carlo" method in cases where there are so many independent variables that solution by the normal stepwise methods would be impractical on any existing or contemplated machines because the number of numbers that would have to be stored and operated on increase exponentially with the number of independent variables.

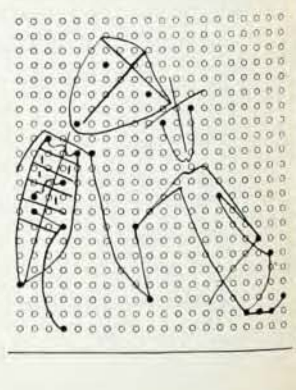
Between these two extremes of purely technical and purely mathematical developments are two important fields having to do with the logical principles of computations. One concerns the internal organization of the machine itself, especially its facilities for executing sequences of operations automatically. The other concerns the task of reducing a problem, once it has been formulated in suitable terms for machine calculation, to a set of operating instructions that can be fed into the machine. The former has been called the problem of the intelligence of the mechanical computer and the latter that of communication between the computer and the mathematician. We wish to discuss these two fields, especially the latter, at some length.

Internal Organization: Background Coding

One of the most basic advances to date in internal machine organization can be illustrated by reference to the ENIAC, built in Philadelphia and now located at the Aberdeen Proving Ground in Maryland. But first—all the computers we are considering are fully automatic: once the machine has been properly adjusted for doing a certain calculation, supplied with the necessary constants and starting data on a reel of magnetic wire or a stack of punch cards, dial settings, or the like, and the "start" button pushed, the machine then proceeds without human intervention (barring malfunction) until the calculation is finished. The ENIAC is such a machine. Its operations are primarily electronic (rather than mechanical) except that numerical data can be fed into it from a stack of punched cards and the answers punched out on another stack.







Among the various components of the ENIAC are twenty counters, each with a capacity of ten decimal digits and sign. Some of these counters are associated with the multiplying unit, others with the divider and square-rooter. Switches can be set manually to store whatever tabular functions may be needed in a given problem.

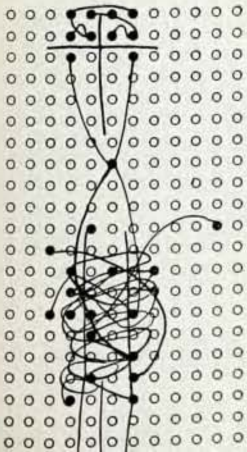
But the task of preparing the ENIAC (in its original form) to solve a particular problem turned out to be considerable. In each step of the sequence of operations two or more components of the machine would act in concert to effect the simple operations of arithmetic, the transfer of numbers from one part to another, the reading in of new data, etc. When a given step was completed, an electrical impulse was provided by one of the participating components, and this signal served to stimulate the various members involved in the succeeding step, and so on. The signals were transmitted along trunk lines, to which the various components were connected by short plug-in cables. The behavior of a component connected to a particular line could be further controlled by manual switches.

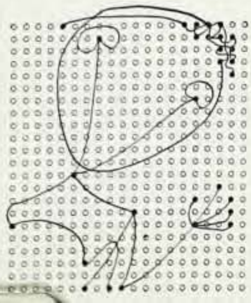
Now a component might be differently stimulated on several different occasions. The counters, for example, were provided with facilities for twelve distinct modes of operation. In many ways the whole arrangement was similar to a telephone communication system. The task of figuring out the wire connections for any but the simplest problems (to say nothing of stringing the wires, plugging them in, setting switches, and so forth) was a fantastic Chinese puzzle, and it is a tribute to the ingenuity and persistence of the persons connected with the ENIAC in those days that so much valuable work was done by its aid under that regime.

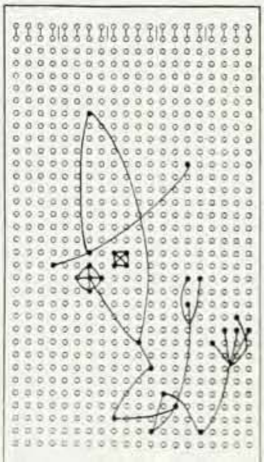
About two years ago, as a result of a suggestion by John von Neumann, a new approach to the method of operating the ENIAC was adopted. The idea was to simplify the preparation of problems for the machine as well as to reduce the actual time it took to set them up on the machine. The proposal was to wire up the machine, once and for all, so that it could understand and execute any one of a list of about sixty orders or instructions. Then, to make the computer solve a particular problem, it was only necessary to translate the problem into a sequence of instructions, chosen from that list, and supply the machine somehow with the sequence of instructions.

Any one of the orders on the list could be used any desired number of times, and they could be used in any desired arrangement. It can be proved that under these circumstances the original list of sixty orders sufficed for performing any finite calculation. A typical order is to take the number stored at one location, multiply it by the number stored at another location, and send the product to a third location. Other orders have to do with shifting numbers to the right or left with respect to the decimal point, transferring them from one storage location to another, adding, subtracting, dividing, extracting square-roots, reading numbers from punch cards or pre-set switches, etc. Two important orders, called "transfer orders," were included, whereby the machine could be made to repeat part of the sequence of instructions or to skip from one point in the sequence to another, depending on the outcome of calculations made up to a given point in the problem. With these latter orders the machine could perform mathematical iterations and decide which of several alternative procedures it should follow. Finally, the fact that the physical set-up remained essentially unchanged from problem to problem resulted in considerable simplification and ease in detecting malfunctions. As a consequence the efficiency of operation increased greatly.

To put the above proposal into effect, it was necessary to find some way of supplying the computer with the sequence of orders corresponding to a given

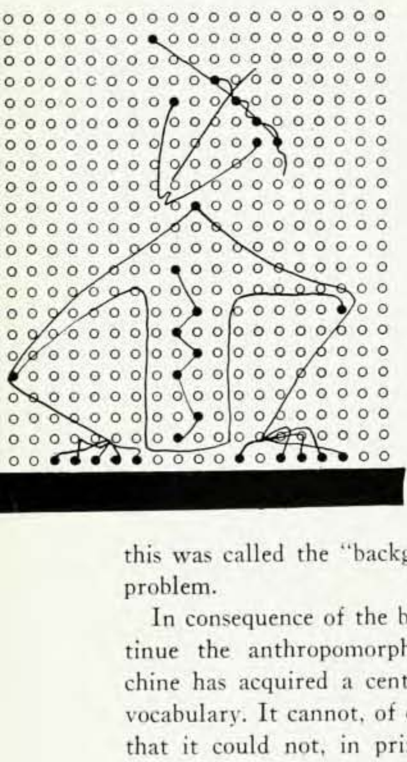






problem. Fortunately the ENIAC contained three components, called function tables, which possessed an enormous number of dial switches originally intended for holding the values of tabular functions, like logarithms, needed in a given problem. Experience gained up to that time showed that only a small fraction of these switches were ever used at once, and it was found possible to use the remainder for holding orders. Each order was expressed as a two- (in some cases four- or six-) digit number and the orders corresponding to any given problem could then be set up on these switches in sequences.

The problem of arranging wire connections and switches so that the ENIAC would read the orders, one by one, from the function tables, interpret them, and execute them, was solved in collaboration with workers from Princeton, Aberdeen, and Los Alamos:



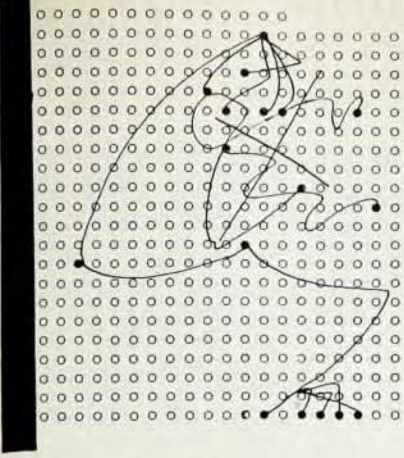
this was called the "background" coding or control problem.

In consequence of the background coding, to continue the anthropomorphic terminology, the machine has acquired a central nervous system and a vocabulary. It cannot, of course, solve any problems that it could not, in principle, have solved under the old regime, but the mathematician's problem of communicating with the machine has been enormously simplified.

Vocabularies

Virtually all digital computers now being built or designed employ the principle of a basic vocabulary in terms of which problems are to be presented to the machine. The choice of basic vocabulary is by no means unique, and it is worthwhile to indicate briefly some of the possible variations.

In the first place, one must decide which operations are to be regarded as fundamental, and which ones as derived. For example, some existing machines have built-in circuits for extracting square-roots in response to a single order, but in other machines this operation must be coded by the person planning the problem, i.e., translated into a subsequence utilizing only the basic arithmetical operations. The built-in circuit is, of course, more convenient and can be designed to operate more quickly and efficiently than the coded sequence, but it increases substantially the complexity of the machine. As a further example, some machines have automatic provision for a "floating decimal point," whereby every (nonzero) number is expressed (apart from its sign) as a decimal in the range from 0.1000... to 0.9999... times a power of ten (manifestly a similar arrangement is possible for numbers in the binary system) where both the decimal and the exponent of ten are stored in the machine. This simplifies problems in which quantities vary over enormous ranges, but it complicates the basic arithmetic circuits and slows the machine down needlessly in most problems. Even operations like multiplication and division could in principle be eliminated from the basic vocabulary, for they can be reduced to additions and subtractions, and these latter could be restricted to the addition and subtraction of one-digit numbers. To pursue this subject one step further, it is known that all mathematics can be reduced to the primitive logical operations of joint denial and quantification, and one mathematician has seriously suggested that machines could be used in the study of the algebra of classes; but anyone who thinks that very much arithmetic could be done in these terms alone should make the experiment of taking the definitions of the number 1 given in W. V. Quine's book on Mathematical Logic, and then expressing it in terms of the primitive symbols for joint denial and quantification, by reference to the preceding chapters of the book. The result easily fills several pages! It is clear that it is eminently worthwhile to retain the compounded arithmetic operations in the basic vocabulary of any machine intended for serious work in applied mathematics. But such derived features as square-root extraction and the floating decimal point are probably just about marginal in their desirability for an all-purpose machine.



The basic vocabularies differ also in their degree of inflection. Just as in some languages the meaning of a word can be altered by adding a variable ending or beginning, the meaning of an order can be modified by attaching variable parts. For example, a basic order might read, "take the number stored at location. . . . , multiply it by the number stored at location. . . . , and send the product to location. . . . ." The person coding the problem would then have the opportunity to insert in the blanks the numbers designating various storage locations in the machine. These designations are called addresses. The simplest feasible system appears to be that in which there is just one variable address in each order. In this case an order for multiplication would specify only the address for storing the product and would have to be preceded by two orders saying "take the number stored at location. . . . and send it to the standard location for the multiplier" and a similar order for the multiplicand. Some systems in use have as many as four variable addresses and other variable features in each order.

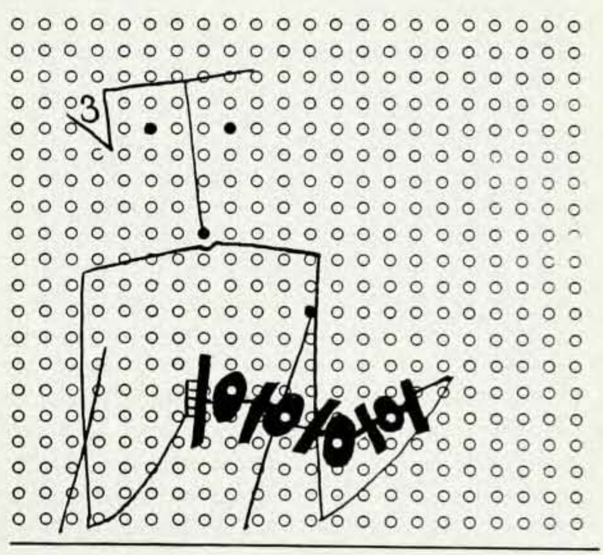
Problem Formulation

In any case it is clear that the basic operations that are wired into the machine will be restricted to fairly simple arithmetic, and this brings us to the other main aspect of the problem of communication between the machine and the mathematician—the art of formulating complex mathematical problems in terms of the basic vocabulary. This art is straightforward in one sense because all calculations in applied mathematics are in principle reducible to arithmetic, but it has different aspects, depending on how one looks at the computing machine.

By some, the computing machine is regarded as roughly the equivalent (except in the matter of speed) of so many square feet of human computers equipped with paper, pencils, desk calculators, and tables, and trained in elementary arithmetic. In this case, the task of preparing a problem consists of writing down a list of operations to be performed and indicating, after each operation, which operation in the list is to be performed next, or which of several operations is to be performed if the sequence depends on the results of an intermediate step.

In problems in which there are many repeated routines, sub-routines, sub-sub-routines, and so forth, which overlap in various degress, the task of preparing this list is difficult because it means that the person planning the problem must be able to visualize the entire course of the calculation at one time and always keep in mind what quantities are available as the result of earlier operations and in what storage location they are stored. This task is greatly simplified by use of a technique described by Burks, Goldstine and von Neumann in "Preliminary Discussion of the Logical Design of an Electronic Computing Machine," Institute for Advanced Study, 1946, and subsequent reports. One may draw a flow-diagram showing graphically the course of the calculation with specially-developed notational devices which make it possible to keep track of the quantities in storage, the status of incomplete inductions, etc.

This technique enables us to change slightly our point of view and suppose that the "equivalent" human computers can do more than carry out instructions on individual arithmetic operations: they can also remember large blocks of operations and perform such blocks upon receipt of single instruc-

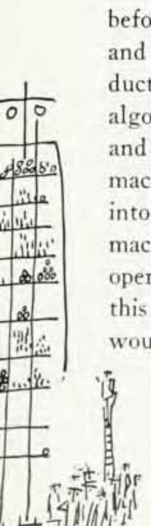


tions. For example, in a calculation requiring use of the exponential function, a sub-routine can be set up for calculating this function for whatever value of a variable is stored in a certain location. Then later, when the exponential function of any quantity is needed, this quantity is merely sent to the given location, and an indication given to go through the said sub-routine.

It can be visualized that, in a computing establishment built around an automatic machine, a large collection or library of such sub-routines will be accumulated, the corresponding sub-sequences of instructions being stored on punched paper tape, magnetized wire, or the like. There would be routines for computing special functions, evaluating integrals, solving differential equations of certain types, inverting matrices, etc. The gain can be likened to the development of interchangeable parts which eventually made possible our system of mass production.

Analysis

It is interesting to change our point of view still further and suppose that the "equivalent" human computers are capable of doing not only arithmetic, but also (after a certain amount of instruction) algebra and some kinds of higher mathematics. We may illustrate what is meant here by an example that is admittedly (and even intentionally) somewhat trivial. Suppose that in a problem it is necessary to evaluate, for many values of a variable x, a number of rational algebraic functions, fractions whose numerators and denominators may be polynomials of a high degree in x. Suppose, further, it is suspected that there are large factors that can be cancelled out of numerator and denominator in each case. *If* the coefficients of the polynomials were known in advance of the problem, each fraction could be reduced to lowest terms, by pencil-and-paper work, before the numerical work of computing numerators and denominators begins. However, the task of reduction to lowest terms (e.g., by the Euclidean algorithm) can in any case be coded as a sub-sequence and performed by the machine. In a sense, then, the machine has ventured out of the realm of arithmetic into that of algebra; although, in another sense, the machine is merely performing certain *arithmetical* operations on the coefficients of the polynomial—this is, of course, just what a human algebraist would do.



We can go further, and instruct the machine (again by suitable sub-sequences) to *differentiate* polynomials. We can instruct it to store, and operate with, functions defined by indefinite series, whose coefficients satisfy a recurrence relation, and so forth. Other operations in mathematical analysis can be similarly coded.

The extent to which the analytic methods of higher mathematics, as contrasted with arithmetic, can be usefully incorporated into computational problems by machine is hardly more than a matter for speculation at present. But it would seem reasonable to guess that it will be possible to think of macines as doing part of their work by analysis, in the not too distant future, and resorting to arithmetic only in the last stages of a problem. This is, of course, only a manner of thinking, but it may nevertheless have an important influence in the art of preparing problems.

Difficulties arise, just as for humans, in handling functions that are implicitly defined. For example, in analysis, one thinks of an indefinite integral as being "the function whose derivative is so-and-so" and because there are no routines which can be generally prescribed for finding out what function this is, one often simply looks it up in a table of integrals. The machine would also have to be supplied with a table of integrals (this would be part of the library mentioned above) through which it can search in a systematic manner when integration is required.

It is likely that before machines can be used efficiently for analytical purposes, the entire question of basic vocabulary will have to be re-examined and appropriate changes made in the basic machine operations. The orders or intructions now in use are really abbreviations for describing elaborate complexes of elementary operations, such as the steps involved in multiplying 635×20094. These complexes are highly efficient only in arithmetic, and it seems clear that other complexes will be needed for other work.

Practical Use of Machines

Nothing in the foregoing should be construed as meaning that it will soon be possible for mathematical physicists to dispense with thinking their problems through in more or less complete detail. In fact it becomes even more necessary to do this for machines than with human computers, who take a long enough time at their work so that the supervisor can develop and alter the methods used from time to time as the calculation progresses. In machine work one should cross all one's bridges before the calculation starts and have the complete strategy mapped out from the beginning. Automatic calculation is profitable at present only when all the operations, whether numerical or analytic, have been thoroughly worked out and tested in special cases. The function of the machine is to apply the same methods to a large number of similar cases. However with the advent of newer and more flexible computers in the laboratory and the ease contemplated in changing from one problem to another, it is not difficult to imagine that they will become a very useful experimental tool for testing methods as well as theories. Thus one can indeed make trial runs, relinquish the computer to the next person, and return to it after a period of examination and meditation.

In principle it might seem feasible to instruct a machine to solve problems of the following sort: given a complicated analytic expression and a certain differential equation, the question is: "does the given expression satisfy the given equation? Answer 'yes' or 'no'!" But in all such cases that one can readily think of, the labor involved in planning the machine calculation would seem to be greater than that of solving the problem by hand. At the other extreme one must avoid the temptation of thinking that fundamental progress can be made by the mere mass-production of numbers.

Probably the greatest promise for machines lies in those fields of physics, astronomy, chemistry, meteorology, etc. where the fundamental principles are well known but where their application to certain basic problems is beset with severe mathematical difficulties. The classical problem of this sort is of course the many-body problem of celestial mechanics. Other examples are the application of quantum mechanics to atomic and molecular structure, the dynamics of simple chemical reactions, properties of crystals, liquids, gases, and to low-temperature phenomena; applications of fluid dynamics to problems of turbulence and supersonic flow; applications of quantum physics to the problem of radiative transport in stars and the complex problems of stellar atmospheres; applications of mechanics, thermodynamics, and electromagnetism to problems in the dynamics of stars, especially variable stars and stellar models; calculation of particle orbits in proposed designs of high-energy accelerators for nuclear work; ray tracing in optical systems; the prediction of the properties of nuclear reactors; etc.

Applications to mathematics itself are perhaps not likely to be too numerous. But one problem which was put on the ENIAC involved searching through the integers, 1, 2, 3, 4, . . . , for integers having certain very rare properties of interest in number theory. It has been suggested that Riemann's conjecture concerning the zeros of the zeta function be put to test by detailed calculation of numerical values of this function in certain regions of the complex plane. In connection with mathematical problems one often hears the remark that you can never "prove" anything by mere calculation. As noted above, this is by no means true, although the fact probably remains that if one could prove a theorem by a machine one could in most cases prove it more easily without the machine, and this is likely to be a fundamental deterrent to most uses of machines in pure mathematics.

The most interesting applications in applied mathematics are likely to be those in which one does more than merely calculate with greater accuracy some already roughly-known result, and in which there is at least a possibility of surprise by way of basic new trends emerging, or in which one is exploring the properties of assumed models of physical, astronomical, chemical, or even perhaps biological systems. By their very nature, these applications are not easy to foresee, and perhaps, therefore, this is the point at which this discussion should close.